

SUPPORTING MOBILE MIXED-REALITY EXPERIENCES

by Martin Flintham, BSc, MSc

Thesis submitted to the University of Nottingham for
the degree of Doctor of Philosophy

December 2007

Abstract

Mobile mixed-reality experiences mix physical and digital spaces, enabling participants to simultaneously inhabit a shared environment online and on the streets. These experiences take the form of games, educational applications and new forms of performance and art, and engender new opportunities for interaction, collaboration and play. As mobile mixed-reality experiences move out of the laboratory and into more public settings they raise new challenges concerning how to support these experiences in the wild.

This thesis argues that mobile mixed-reality experiences in which artists retain creative control over the content and operation of each experience, particularly those that are deployed as theatrical performances, require dedicated support for content authoring and reactive orchestration tools and paradigms in order to be successfully and robustly operated in public settings. These requirements are examined in detail, drawing on the experience of supporting four publicly toured mobile mixed-reality experiences; *Can You See Me Now?*, *Uncle Roy All Around You*, *I Like Frank* in Adelaide and Savannah, which have provided a platform to practically develop, refine and evaluate new solutions to answer these challenges in the face of presenting the experiences to many thousands of participants over a four year period.

This thesis presents two significant supporting frameworks. The ColourMaps system enables designers to author location-based content by directly colouring over maps; providing a simple, familiar and yet highly flexible approach to matching location-triggers to complex physical spaces. It provides support for multiple and specialised content layers, and the ability to configure and manage other aspects of an experience, including filtering inaccurate position data and underpinning orchestration tools. Second, the Orchestration framework supports the day-to-day operation of public experiences; providing dedicated control-room tools for monitoring that reveal the content landscape and historical events, intervention and improvisation techniques for steering and shaping each participant's experience as it unfolds both physically and virtually, and processes to manage a constant flow of participants.

Selected Published Works

At the time of writing, sections of this thesis have appeared in several peer-reviewed publications as long papers, a selection of which are referenced here:

Flintham, M., R. Anastasi, S. Benford, A. Drozd, J. Mathrick, D. Rowland, N. Tandavanitj, M. Adams, J. Row-Farr and A. Oldroyd (2003) **Uncle Roy All Around You: Mixing Games and Theatre on the City Streets**. In *Proceedings of Level Up: The First International Conference of the Digital Games Research Association (DIGRA)*, Utrecht, Netherlands, November 2003.

Flintham, M., S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams and J. Row-Farr (2003): **Where on-line meets on the streets: experiences with mobile mixed reality games**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, Fort Lauderdale, Florida, April 2003.

Flintham, M. (2005) **Painting the town red: configuring location-based games by colouring maps**. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Valencia, Spain, June 2005.

Benford, S., R. Anastasi, M. Flintham, C. Greenhalgh, N. Tandavanitj, M. Adams and J. Row-Farr (2003) **Coping with uncertainty in a location-based game**. In *Pervasive Computing*, IEEE **2**(3): pp 34-41.

Benford, S., A. Crabtree, M. Flintham, D. Rowland, B. Gaver, M. Adams, J. Row-Farr, N. Tandavanitj, A. Oldroyd and J. Sutton (2004): **Provoking Reflection Through Artistic Games**. In *Proceedings of the 2004 CHI Conference on Human Factors in Computing Systems (Workshop 19. Social Learning through Gaming)*, Vienna, Austria, April 2004.

Benford, S., D. Rowland, M. Flintham, R. Hull, J. Reid, J. Morrison, K. Facer and B. Clayton (2004) **Savannah: Designing a location-based game simulating lion behaviour**. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Singapore, June 2004.

Benford, S., A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams and J. Row-Farr (2006) **Can you see me now?** In *ACM Transactions on Computer-Human Interaction (TOCHI)*, **13**(1): pp 100-133.

Acknowledgements

I gratefully acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC) through their support of the Equator project.

I would like to particularly thank my supervisor, Professor Steve Benford, and my collaborators and friends at Blast Theory; Matt Adams, Ju Row Farr and the inimitable Nick Tandavanitj. Without their vision and passion, and most importantly the faith they put in me to ‘get the job done’, none of this work would have been possible.

I would also like to thank the supporting cast I have had the opportunity to work with. Particularly, the performers; Paul Dungworth, Dicky Eton, Sheila Ghelani, Jamie Iddon, Caitlin Newton-Broad and Hannah Talbot, and the developers; Deyan Raykov and Jon Sutton. I could not have asked for a more dedicated and professional group of people to work with.

Thanks to my long-suffering colleagues and friends at the Mixed-Reality Lab past and present for putting up with me for so long, and their unwavering support both academically and socially; Alan Chamberlain, Adam Drozd, Chris Greenhalgh, Jan Humble, Pilar Herrero, Alex Irune, Dave Lloyd, Jim Purbrick, Stuart Reeves, Duncan Rowland, Holger Schnädelbach, Ian Taylor and far too many others to mention.

Finally I would like to thank my parents, who convinced me that I should do this in the first place, my brother, and my friends – especially Michelle for keeping me sane in spite of this occasionally being “a little bit rubbish”.

Table of Contents

ABSTRACT	I
SELECTED PUBLISHED WORKS	II
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	VIII
LIST OF TABLES	XI
LIST OF TABLES	XI
1 INTRODUCTION	1
1.1 INTRODUCTION.....	1
1.2 MOBILE MIXED-REALITY EXPERIENCES.....	2
1.3 METHODOLOGY.....	4
1.4 MOTIVATION AND FOCUS.....	6
1.5 THESIS STRUCTURE.....	9
2 A REVIEW OF MOBILE MIXED-REALITY EXPERIENCES	12
2.1 INTRODUCTION.....	12
2.2 RELATED WORK.....	12
2.2.1 <i>Pirates!</i>	12
2.2.2 <i>ARQuake</i>	14
2.2.3 <i>Human Pacman</i>	15
2.2.4 <i>Treasure</i>	17
2.2.5 <i>Pac-Manhattan</i>	19
2.2.6 <i>Riot! 1831</i>	20
2.2.7 <i>CitiTag</i>	22
2.2.8 <i>CatchBob!</i>	23
2.2.9 <i>Botfighters</i>	25
2.2.10 <i>Mogi</i>	26
2.2.11 <i>Urban Tapestries</i>	27
2.2.12 <i>GeoTracing</i>	29
2.2.13 <i>Real Tournament</i>	30
2.2.14 <i>GUIDE</i>	32
2.2.15 <i>Mobile Experience Engine</i>	33
2.2.16 <i>Desert Rain</i>	35
2.2.17 <i>Feeding Yoshi</i>	36
2.2.18 <i>Big Urban Games</i>	37

2.2.19	<i>GeoCaching</i>	39
2.3	ASSOCIATED WORK.....	40
2.3.1	<i>Alternate Reality Games</i>	40
2.3.2	<i>Indoor Augmented Reality Games</i>	40
2.4	SUMMARY OF REQUIREMENTS.....	41
2.5	CONCLUSIONS.....	44
3	FOUR FOUNDATION EXPERIENCES	45
3.1	INTRODUCTION.....	45
3.2	CAN YOU SEE ME NOW?.....	45
3.2.1	<i>Online Experience</i>	46
3.2.2	<i>Mobile Experience</i>	51
3.2.3	<i>Touring History</i>	53
3.3	UNCLE ROY ALL AROUND YOU	54
3.3.1	<i>Mobile Experience</i>	55
3.3.2	<i>Online Experience</i>	59
3.4	I LIKE FRANK IN ADELAIDE.....	63
3.4.1	<i>Mobile Experience</i>	64
3.4.2	<i>Online Experience</i>	67
3.5	SAVANNAH.....	71
3.5.1	<i>Mobile Experience</i>	72
3.5.2	<i>Classroom Experience</i>	74
3.6	SUMMARY OF REQUIREMENTS.....	76
3.7	CONCLUSIONS.....	76
4	A FRAMEWORK FOR SUPPORTING MOBILE MIXED-REALITY EXPERIENCES	78
4.1	INTRODUCTION.....	78
4.2	FORMS OF PARTICIPATION.....	78
4.3	SUPPORTING TASKS	84
4.3.1	<i>Authoring and Improvisation Tasks</i>	84
4.3.2	<i>Challenges Raised by the Framework</i>	88
4.4	CONCLUSIONS.....	90
5	TOOLS TO SUPPORT AUTHORIZING	91
5.1	INTRODUCTION.....	91
5.2	A REVIEW OF EXISTING AUTHORIZING TOOLS.....	91
5.2.1	<i>Geographic Information Systems</i>	92
5.2.2	<i>Game Engine Authoring Tools</i>	94
5.2.3	<i>Context Aware Systems</i>	98
5.2.4	<i>Topiary</i>	99

5.2.5	<i>Mediascapes</i>	100
5.2.6	<i>Activity Zones</i>	102
5.2.7	<i>Summary of Existing Tools</i>	102
5.3	COLOURMAPS	103
5.3.1	<i>Principles of ColourMaps</i>	103
5.3.2	<i>Start Positions</i>	106
5.3.3	<i>GPS Filtering</i>	107
5.3.4	<i>Clue Trails</i>	109
5.3.5	<i>Authoring Layers and Ideal Routes</i>	111
5.3.6	<i>Edges and Levels</i>	114
5.3.7	<i>Latching and Digging</i>	117
5.3.8	<i>Preparation and Validation Mechanisms</i>	119
5.4	SUMMARY OF COLOURMAP FEATURES	122
5.5	DISCUSSION	126
5.5.1	<i>Feedback</i>	126
5.5.2	<i>Limitations</i>	128
5.5.3	<i>Broader Reflections</i>	129
5.6	CONCLUSIONS	131
6	TOOLS TO SUPPORT ORCHESTRATION	133
6.1	INTRODUCTION	133
6.2	FUNDAMENTALS OF ORCHESTRATION	133
6.2.1	<i>Role-Playing and Improvisation</i>	134
6.2.2	<i>MUDs and MMORPGs</i>	135
6.2.3	<i>Static Mixed-Reality Performances</i>	136
6.2.4	<i>Control Rooms</i>	140
6.2.5	<i>Summary</i>	141
6.3	A FRAMEWORK OF ORCHESTRATION TOOLS	142
6.3.1	<i>Monitoring</i>	144
6.3.2	<i>Revealing History</i>	147
6.3.3	<i>Revealing Content</i>	151
6.3.4	<i>Intervention</i>	155
6.3.5	<i>Improvisation and Performance</i>	158
6.3.6	<i>Inducting Mobile Players</i>	161
6.3.7	<i>Inducting Online Players</i>	166
6.3.8	<i>Distributed Orchestration and Shared Awareness</i>	169
6.4	SUMMARY OF ORCHESTRATION FUNCTIONS AND TOOLS	173
6.4.1	<i>Functions and Tools</i>	174
6.4.2	<i>Broader Reflections</i>	177
6.5	CONCLUSIONS	180

7	IMPLEMENTATION.....	181
7.1	INTRODUCTION.....	181
7.2	TECHNOLOGIES.....	181
7.2.1	<i>Connectivity.....</i>	<i>181</i>
7.2.2	<i>Middleware.....</i>	<i>185</i>
7.2.3	<i>User Interfaces</i>	<i>186</i>
7.3	IMPLEMENTATION.....	187
7.3.1	<i>ColourMaps and the Game Engine.....</i>	<i>188</i>
7.3.2	<i>Supporting Varied Client Requirements</i>	<i>191</i>
7.3.3	<i>Integrating Orchestration Interfaces.....</i>	<i>195</i>
7.3.4	<i>Experiencing Disconnection</i>	<i>197</i>
7.3.5	<i>Supporting Disconnected Operation.....</i>	<i>199</i>
7.3.6	<i>Minimising Disconnection.....</i>	<i>203</i>
7.4	CONCLUSIONS.....	207
8	CONCLUSION	208
8.1	INTRODUCTION.....	208
8.2	SUMMARY	208
8.2.1	<i>Foundation Experiences and Related Works</i>	<i>209</i>
8.2.2	<i>Supporting Framework</i>	<i>211</i>
8.2.3	<i>Supporting Tools and Implementation</i>	<i>212</i>
8.3	RESEARCH CONTRIBUTIONS.....	216
8.3.1	<i>Supporting the third role.....</i>	<i>217</i>
8.3.2	<i>Methodology.....</i>	<i>219</i>
8.3.3	<i>Dissemination and Impact</i>	<i>221</i>
8.4	REFLECTIONS AND FUTURE WORK.....	225
	REFERENCES	228

List of Figures

FIGURE 1-1: A MOBILE PLAYER AND ONLINE PLAYERS IN CAN YOU SEE ME NOW?.....	4
FIGURE 2-1: PIRATES! INTERFACE.....	13
FIGURE 2-2: ARQUAKE HARDWARE (LEFT) AND IN-GAME VIEW (RIGHT).....	14
FIGURE 2-3: HUMAN PAC-MAN PLAYER AND VIEW, WITH ONLINE HELPER.....	16
FIGURE 2-4: TREASURE INTERFACE SHOWING NETWORK COVERAGE (LEFT) AND COINS (RIGHT).....	18
FIGURE 2-5: A PAC-MANHATTAN MOBILE PLAYER (LEFT) AND VIRTUAL GAME BOARD (RIGHT).....	20
FIGURE 2-6: A RIOT! 1831 PLAYER (LEFT) AND MEDIASCAPE AUTHORING TOOL (RIGHT).....	21
FIGURE 2-7: CITITAG HANDHELD COMPUTER INTERFACE.....	22
FIGURE 2-8: CATCHBOB! INTERFACE SHOWING USER ANNOTATIONS AND NETWORK CONNECTIVITY MONITOR.....	24
FIGURE 2-9: BOTH FIGHTERS ONLINE CONFIGURATION INTERFACE (LEFT) AND MOBILE PHONE INTERFACE (RIGHT).....	25
FIGURE 2-10: MOGI MOBILE PHONE INTERFACE (LEFT) AND ONLINE INTERFACE (RIGHT).....	26
FIGURE 2-11: URBAN TAPESTRIES MOBILE INTERFACE (LEFT) AND ONLINE MAP SHOWING TRAILS (RIGHT).....	28
FIGURE 2-12: GEOTRACING ONLINE INTERFACE SHOWING TRAILS AND LOCATION-BASED MEDIA	29
FIGURE 2-13: REAL TOURNAMENT GUN INTERFACE (LEFT) AND MAP (RIGHT).....	31
FIGURE 2-14: GUIDE INTERFACE SHOWING MAP OVERRIDE FUNCTION.....	32
FIGURE 2-15: TRICKSTER MOBILE INTERFACE.....	34
FIGURE 2-16: DESERT RAIN VIRTUAL ENVIRONMENT (LEFT) AND RAIN CURTAIN (RIGHT).....	35
FIGURE 2-17: FEEDING YOSHI MOBILE INTERFACE SHOWING A PLANTATION (LEFT) AND A YOSHI (RIGHT).....	37
FIGURE 2-18: CONQWEST PLAYERS (LEFT) AND THE GO GAME (RIGHT).....	38
FIGURE 2-19: A GEOCACHING CACHE (LEFT) AND PLAYERS USING A GPS RECEIVER (RIGHT).....	39
FIGURE 3-1: EARLY CAN YOU SEE ME NOW? ONLINE INTERFACE.....	47
FIGURE 3-2: ESCAPING FROM A MOBILE PLAYER.....	48
FIGURE 3-3: ZOOMED-OUT VIEW OF THE VIRTUAL CITY.....	48
FIGURE 3-4: AN ONLINE PLAYER IS SEEN.....	49
FIGURE 3-5: SIGHTINGS ARCHIVE.....	50
FIGURE 3-6: A MOBILE PLAYER RUNS WITH EQUIPMENT.....	51
FIGURE 3-7: MOBILE INTERFACE, LEFT, AN ONLINE PLAYER IS SEEN, RIGHT.....	52
FIGURE 3-8: A MOBILE PLAYER EXPLORES THE PARK.....	56
FIGURE 3-9: MAP INTERFACE AND RED MARKER, LEFT, A MESSAGE FROM UNCLE ROY, RIGHT.....	57
FIGURE 3-10: A MESSAGE FROM AN ONLINE PLAYER, LEFT, RECORDING A REPLY, RIGHT.....	57
FIGURE 3-11: A MOBILE PLAYER APPROACHES THE LIMOUSINE.....	59
FIGURE 3-12: SEARCHING PHOTOGRAPHS OF THE CITY.....	60
FIGURE 3-13: FINDING UNCLE ROY'S OFFICE ONLINE.....	61

FIGURE 3-14: DIRECTING A MOBILE PLAYER.....	61
FIGURE 3-15: MAKING A COMMITMENT IN THE OFFICE.....	62
FIGURE 3-16: VIEWING A COMPLETED POST CARD ONLINE.....	63
FIGURE 3-17: MOBILEPHONE INTERFACE, LEFT, 3G PHONE, RIGHT	64
FIGURE 3-18: A PERFORMER, RIGHT, MAKES A VIDEO CALL TO A MOBILE PLAYER, LEFT	66
FIGURE 3-19: SEARCHING FOR A POST CARD ONLINE	68
FIGURE 3-20: RECRUITING A MOBILE PLAYER TO HELP FIND A POST CARD.....	69
FIGURE 3-21: GUIDING A MOBILE PLAYER TO A POST CARD.....	69
FIGURE 3-22: FINDING FUTURELANDS ONLINE.....	71
FIGURE 3-23: SAVANNAH PDA INTERFACE	72
FIGURE 3-24: MANIPULATION USING THE DEN INTERFACE	74
FIGURE 3-25: REFLECTION USING THE DEN INTERFACE.....	75
FIGURE 4-1: HIGHLIGHTING FORMS OF PARTICIPATION.....	83
FIGURE 4-2: SUPPORTING TASKS.....	84
FIGURE 4-3: CHALLENGES RAISED BY THE FRAMEWORK.....	88
FIGURE 5-1: ARCGIS DESKTOP AUTHORIZING	93
FIGURE 5-2: EDITING A LEVEL IN VALVE HAMMER.....	95
FIGURE 5-3: XPILLOT, LEFT, EDITING A LEVEL WITH NOTEPAD, RIGHT	97
FIGURE 5-4: REPTON, LEFT, PAINTING A NEW LEVEL, RIGHT	97
FIGURE 5-5: THE ICAP EDITOR.....	99
FIGURE 5-6: TOPIARY AUTHORIZING MAP.....	100
FIGURE 5-7: AUTHORIZING REGIONS IN MEDIASCAPES	101
FIGURE 5-8: AUTHORIZING START POSITIONS	107
FIGURE 5-9: A GPS FILTERING COLOURMAP	108
FIGURE 5-10: SERVING CLUES WITH A COLOURMAP	110
FIGURE 5-11: GAME AREA AND START POSITIONS, TOP LEFT, IDEAL ROUTES, TOP RIGHT, ADDING REGIONS, BOTTOM LEFT, FINISHED COLOURMAP, BOTTOM RIGHT.....	112
FIGURE 5-12: A CLUE TRAIL COLOURMAP.....	114
FIGURE 5-13: A MULTI-LEVEL COLOURMAP	116
FIGURE 5-14: OCCUPYING SIGHT, SOUND AND SMELL COLOURMAPS IN SAVANNAH.....	117
FIGURE 5-15: DIGGING IN A COLOURMAP.....	118
FIGURE 5-16: COLOURMAP JPEG ARTEFACTS, TOP, AND EFFECTS OF A SOFT BRUSH, BOTTOM.....	120
FIGURE 5-17: VALIDATING A COLOURMAP	121
FIGURE 5-18: COLOURMAPS IN CONTEXT	123
FIGURE 5-19: ELEMENTS OF THE COLOURMAPS SYSTEM	124
FIGURE 6-1: ORCHESTRATING AVATAR FARM	137
FIGURE 6-2: ORCHESTRATING DESERT RAIN BEHIND-THE-SCENES.....	139
FIGURE 6-3: MONITORING COMPONENTS.....	145
FIGURE 6-4: CONNECTIVITY HISTORY, LEFT, MESSAGE HISTORY AND DETAILS, RIGHT	150
FIGURE 6-5: REVEALING A MOBILE PLAYER'S SPATIAL TRAJECTORY	150

FIGURE 6-6: REVEALING MAP ANNOTATIONS	153
FIGURE 6-7: REVEALING HOW CONTENT HAS BEEN EXPLORED IN SAVANNAH.....	154
FIGURE 6-8: INTERVENTION AND MESSAGE SENDING COMPONENTS FOR UNCLE ROY ALL AROUND YOU	156
FIGURE 6-9: INDUCTION PROCESS	163
FIGURE 6-10: CAPTURING PLAYER DATA FOR ORCHESTRATION	164
FIGURE 6-11: PREPARING THE DEVICE, LEFT, WHILE PLAYERS ARE BRIEFED, RIGHT	165
FIGURE 6-12: SUMMARY OF ORCHESTRATION FUNCTIONS	175
FIGURE 7-1: GAME-ENGINE ARCHITECTURE.....	189
FIGURE 7-2: DEDICATED PROXIES FOR ONLINE AND MOBILE CLIENTS.....	193
FIGURE 7-3: INTEGRATING THE ORCHESTRATION FRAMEWORK.....	196
FIGURE 7-4: REVISED ARCHITECTURE TO SUPPORT DISCONNECTED OPERATION	201
FIGURE 7-5: DISCONNECTED OFFICE CHALLENGE	203
FIGURE 7-6: DISCONNECTED INTERFACE, LEFT, CONNECTED INTERFACE, RIGHT.....	203
FIGURE 7-7: FOUR-LAYER COMMUNICATIONS PLATFORM.....	205

List of Tables

TABLE 2-1: SUMMARY OF EXPERIENCES AND THEIR REQUIREMENTS	43
TABLE 3-1 SUMMARY OF THE FOUR EXPERIENCES AND THEIR REQUIREMENTS.....	76
TABLE 4-1: DEFINING EXPERIENCES BY GENRE.....	79

1 Introduction

1.1 Introduction

Mobile technologies are more widespread than ever before. The last decade has seen the rise of the mobile phone to near ubiquity, alongside high-speed wireless internet access for other mobile devices and laptops, and location-sensing systems that are affordably available to the general public. Computing is no longer limited to the desktop, with researchers, the commercial sector and the general public embracing the ability to be connected to everyone, everywhere. Mobile phones have moved from being a medium that was used only to speak urgently with someone (de Souza e Silva, 2004) to being far more than this; with the advent and widespread adoption of text messaging, mobile internet browsing and multimedia messaging transforming urban environments into spaces where connectivity is the norm, rather than the exception.

These new connected spaces allow the convergence of physical and digital environments, and importantly these spaces are no longer solely the domain of the researcher. Artists, game designers, service providers and educators are all beginning to explore how to create **mobile mixed-reality experiences**, where people inhabit, communicate and play in both a physical space and a digital space simultaneously using mobile devices. However, as an emerging and uncharted new medium this raises the question of what new concepts, tools and practices are required to support practitioners as they create new experiences in the wild. The overall goal of this thesis is to examine the practical challenges of creating real-world mobile mixed-reality experiences, in an attempt to answer this question and to provide new frameworks, tools and platforms to support future mobile mixed-reality experiences.

This chapter begins by defining mobile mixed-reality experiences and describes their role within the research community and in wider society, before highlighting the background, context and methodology of this research.

This chapter then introduces the particular research focus of this thesis, before summarising its research goals and giving an overview of its structure with a brief summary of each chapter.

1.2 Mobile Mixed-Reality Experiences

While mobile technologies such as wireless communication and position sensing are continually evolving, their recent mass accessibility and affordability to the wider population means that there is a shift towards investigating *how* to use these technologies to support new and novel applications. Adoption of mobile technologies is already high, with the majority of mobile phones now supporting mobile internet access, basic location-sensing and in some cases built in GPS receivers allowing reporting of position. Wireless internet connections for laptops and PDAs, increasingly available in urban spaces, allow users to roam without being disconnected from their online lives. Similarly, standalone GPS receivers are cheaply available for hobbyists and outdoor pursuit enthusiasts. As a result, practitioners are able to both extend the online and digital world into the physical environment, as described by (Weiser, 1999) when stating how *ubiquitous computing is moving away from the current desktop personal computing era*, and also begin to consider completely new applications for the digitally enabled mobile domain.

Many developers are beginning to explore mobile technology for novel uses above and beyond simply moving existing desktop practices and applications into a mobile environment. Location-based services are growing as telecommunications operators seek new sources of revenue and new uses for increasingly sophisticated phones, for example by providing mobile maps and directions from a user's current location (Google, 2007), and targeted directories of places of interest. Game developers, while primarily focusing on providing ports of existing desktop and console games for mobile phones, are also beginning to consider *pervasive games* (IPerG, 2007), or how digital games can be given a physical aspect using location (Its Alive!, 2000). Researchers have been developing mobile applications for enhancing museums and the tourism industry, providing location-based tours and exhibit information to visitors equipped with location-sensitive devices (Cheverst et

al., 2000). These applications are targeted for use by the general public, using consumer-level rather than bleeding-edge technologies.

Significantly, artists are attracted to exploring this new mobile space, and are at the forefront of creating radical new kinds of experiences. (Popper, 1993) describes *communication art* as being interesting for artists as it has six main characteristics; it stages physical presence at a distance, it telescopes the immediate and the delayed, it focuses on the playfulness of interactivity, it combines memory and real time, it promotes planetary communication, and it encourages a detailed study of human social groupings. Mobile technologies provide an opportunity for artists to explore a new form of communication art in a setting that mixes the rich communication of the digital environment with the physicality of the real world, questioning a person's assumptions about this hybrid space and raising issues of trust, presence and acceptable behaviour. Most importantly for this thesis, mixing the physical and the digital allows artists to engage with researchers in order to create new kinds of artistic performance, while also providing a real-world platform for researchers to practically drive forward new ideas and publicly demonstrate results. Ultimately, these artistic projects can form the basis for, or feed into, more mainstream applications.

This thesis investigates a particular facet of the convergence of physical and digital environments; mobile mixed-reality experiences. (Milgram and Kishino, 1994) define a mixed-reality as *anywhere between the extrema of the virtuality continuum*, which extends from the physical environment through to the completely virtual, or digital, environment. Consequently, mobile mixed-reality experiences allow participants to inhabit both physical and virtual environments simultaneously, and where actions in one environment affect the presence of the participant in the other. Mobile participants, using devices enabled with position-sensing and wireless communication technologies, use their location in the physical environment to move within the virtual environment and to receive digital content accordingly, as shown in figure 1-1, a scene from the experience *Can You See Me Now?*. Conversely, other participants can primarily inhabit the virtual, digital environment online,

communicating with and accompanying their mobile counterparts as they move through a physical, urban space.

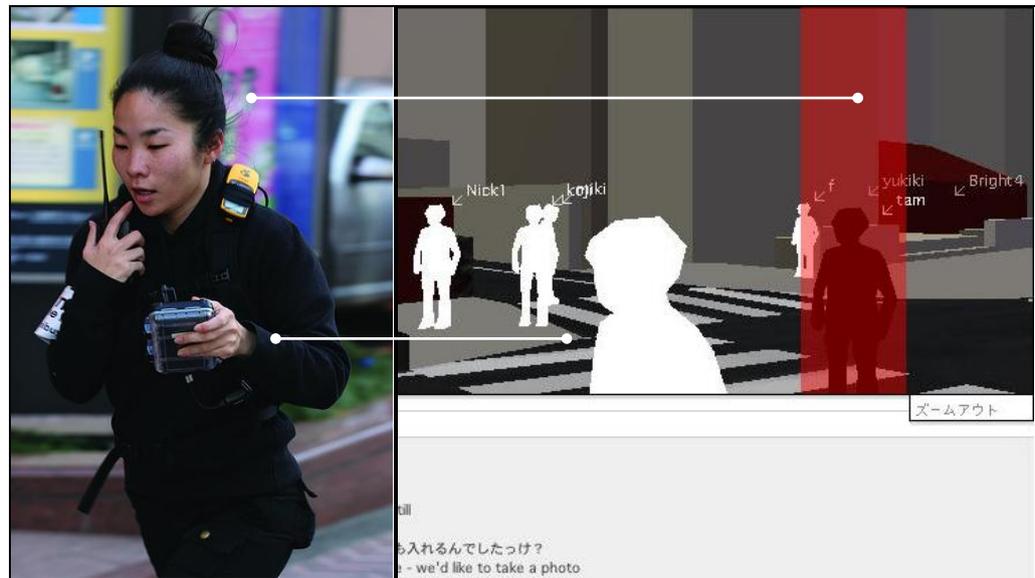


Figure 1-1: A mobile player and online players in Can You See Me Now?

Mobile mixed-reality experiences encompass a wide variety of operating paradigms, as will be highlighted in the next chapter, including those that are instigated and regulated by their participants, or involve community-driven participant generated content. However, this thesis examines experiences that encompass elements of artistic performance and theatre, in particular those that involve a third role alongside the virtual and mobile; that of the artist in shaping each experience as it is developed and as it unfolds. This third role becomes most important when considering mobile mixed-reality experiences in which the creator retains artistic control over the content and operation of each experience through authoring and orchestration, particularly those that are deployed as theatrical public performances. Consequently, this thesis explores how to support practitioners in this new hybrid mobile space through the creation of new tools and processes, particularly focusing on authoring for and orchestrating mobile mixed-reality experiences with a strong artistic element.

1.3 Methodology

This research has been largely conducted as part of the Equator project, an interdisciplinary research collaboration that aimed to understand how *digital and physical activities not only coexist but cooperate* (Equator IRC, 2007)

through a series of experience projects and investigations into new devices, software architectures and design and evaluation methods. The Equator experience project *Citywide Performance* involved a long-term collaboration with the artists group Blast Theory (Blast Theory, 2007), resulting in a series of highly public mobile mixed-reality experiences over a four year period; *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank*. As will be described in a later chapter, these three *Citywide* experiences have been developed for, and shown at, a variety of public art venues and new media festivals including the Institute of Contemporary Arts in London, the Adelaide Fringe Festival and the Dutch Electronic Arts Festival. This provided a unique motivation and opportunity for this thesis in that their development, together with a further educational experience, *Savannah*, required an iterative cycle of discovering the challenges involved and creating solutions in the field, pragmatically responding to their needs as they occurred.

Each experience emerged from a series of exploratory workshops with the artists and the construction of technology prototypes, which were used to define and test the feasibility of the core mechanics of the experience. This led to the first deployment of the experience, a process that involved close collaboration with the artists in order to both implement the supporting systems and to deploy the first versions of the tools that were needed to put the artists' ideas into practice, particularly highlighting the need for dedicated tools for authoring. Early deployments of the experiences highlighted the need for the artists to be able to shape an ongoing experience to provide the best experience for its participants, similarly highlighting the need for dedicated orchestration tools. The touring nature of the experiences meant that the iterative cycle was two-fold, with rapid iteration of supporting tools and technologies occurring with the artists in the lead-up to each deployment, and also a longer period of reflection, evaluation and refinement occurring afterwards, giving the opportunity to implement more substantial changes.

The methodology employed by this thesis, then, is to consider the lessons learnt from this real-world deployment of mobile mixed-reality experiences. It presents the culmination of four years of working closely with Blast Theory, listening to their requirements and accordingly developing solutions with

constant and immediate feedback, while providing an evolving platform for exploration and investigation, a process that Equator has characterised as working “in the wild”. This has allowed the tools and concepts presented in this thesis to be tested, refined and most importantly experienced by end-users, as part of the challenge of presenting these experiences to many thousands of participants, in a public setting on numerous occasions. This engagement with Blast Theory and with the general public has been described by the EPSRC’s final review of the Equator project as *a pre-eminent example of its kind anywhere in the world, especially folding practice-based work back into theory, policy and regulation* (Blackwell et al., 2007), has led to a large number of publications and related literature – listed in full in chapter 8, and gained wider recognition in the form of three BAFTA nominations and the Golden Nica for Interactive Art at Prix Ars Electronica.

The tools and concepts presented in this thesis have been subject to continuous evaluation as part of the practical iterative development cycle that working “in the wild” entails. This research is backed up by substantial ethnographic study in which the author has played a significant role, and the results of this evaluation have then been fed back into the iterative process. As a collaborative effort these studies are not included in this thesis, but are listed in full in chapter 8. It should be noted at this stage that, while the creation and deployment of the four experiences has also been a collaborative effort involving artists, developers and ethnographers, the solutions, tools and systems described in this thesis are solely the original work and contributions of the author, who was the primary software developer and systems architect of the experiences.

1.4 Motivation and Focus

The preceding sections have described the rise in interest in the use of mobile technologies to explore the interaction of physical and digital spaces, and particularly how this has led to the construction of a number of mobile mixed-reality experiences that have been presented to the public as artistic performances. However, as this thesis will show, these experiences raise new challenges that only become apparent when experiences move out of the

laboratory and into such real-world scenarios; including the need to support artists and authors in realising their visions through intuitive design and authoring processes, and careful management of public participants, both during day-to-day operation and when mobile technology is not the seamless enabler it is sometimes perceived to be.

With this in mind, the specific goal and research focus of this thesis can be defined as follows:

To develop a framework for supporting the design and public deployment of artist-led mobile mixed-reality experiences, and to create tools, techniques and guidelines to support experience authors in instantiating, populating and running such experiences with regard to factors unique to mobile technologies.

The term *framework* refers to an analysis and classification of the requirements and challenges of deploying a mobile mixed-reality experience in practice, to highlight areas that critically require support from dedicated tools.

The term *public deployment of artist-led mobile mixed-reality experiences* refers to an experience that is presented to the public as an artistic or theatrical performance, is undertaken by mobile participants exploring the physical environment using mobile, connected devices, and involves interaction and collaboration with online, virtual participants, particularly focusing on the four experiences highlighted in the previous section, and particularly focusing on those experiences in which a group of artists take a lead creative role.

The term *experience authors* refers to domain experts who wish to explore the mobile mixed-reality experiences as an extension of their normal practices; for example artists and educators, particularly drawing on the experience of collaborating with the artists group Blast Theory.

The terms *tools, techniques and guidelines* refers to new ways of supporting mobile mixed-reality experiences through the construction and real-world use

and evaluation of software and operating processes and paradigms, which can in turn provide general strategies that are applicable to the wider field.

Factors unique to mobile technologies refers to understanding how mobile technologies operate in practice on the ground, and importantly how participants react to and make use of them.

This thesis addresses the following specific issues:

- **Understanding the key requirements of mobile mixed-reality experiences:**
 - Identifying a framework that highlights the common requirements of recent mobile mixed-reality experiences from a general point of view.
 - Using this framework to demonstrate how key requirements should be investigated in order to successfully support a mobile mixed-reality experience.
- **Supporting authoring and orchestration:**
 - Identify the needs of authors in creating complex location-based content for mobile mixed-reality experiences as they are deployed in new locations.
 - Identify the need for the real-time orchestration and management of mobile mixed-reality experiences in the face of large numbers of members of the public using everyday mobile technologies.
- **Demonstrating Solutions:**
 - Show how the key requirements of the framework can be supported through the construction of dedicated authoring tools and a framework of orchestration tools and functions.

- Demonstrate how the tools and processes suggested by the framework have been successfully used and evaluated “in situ” to support a number of real-world mobile mixed-reality experiences.

By achieving these research goals it is intended that the foundations are laid for easing the process of creating a successful mobile mixed-reality experience by appreciating the key practical challenges of the task. By constructing a supporting framework and creating tools specifically for the mobile environment, it is hoped that artists and authors can continue to explore mobile mixed-reality experiences as large-scale, public events, with the framework in place to support the technological and practical challenges of implementing and operating them.

1.5 Thesis Structure

This chapter has presented a brief introduction to mobile mixed-reality experiences and to the general context of this thesis. It has introduced the motivation for and the focus of the research, and has outlined how the research is the culmination of a four-year process of deploying experiences in a real-world setting.

The overall structure of this thesis is comprised of three sections. First, it explores the current state of the art of mobile mixed-reality experiences and other related applications both in research and commercial endeavours, including the four experiences developed as part of this research. This leads on to an analysis of the general requirements of mobile mixed-reality experiences, which argues that the need for dedicated authoring tools and orchestration practices raises a critical challenge for practically supporting an experience. Finally, the thesis examines how to support each of these authoring and orchestration tasks in detail, drawing on the four experiences to demonstrate the key contributions of the research and the solutions it provides, before describing how the experiences are practically implemented.

Chapter 2 reviews a selection of mobile mixed-reality experiences and related works that have been developed within this rapidly growing area in order to place this research in a wider context. It includes a brief description of each

experience or application, its key requirements, functions and mechanics, and any other notable features. It concludes by defining a collection of common requirements drawn from these works, which also provides an introduction to the terminology used throughout this thesis.

Chapter 3 describes in detail the four specific mobile mixed-reality experiences that have been co-developed by the author and that form the basis of this research; *Can You See Me Now?*, *Uncle Roy All Around You*, *I Like Frank and Savannah*. It describes the end-user experience of each for both mobile and online participants, and gives a historical overview of the development and public deployment of the experiences. These descriptions are drawn upon throughout the remainder of this thesis to show how the experiences are supported by this research. Finally, the chapter concludes with a summary of the requirements of the experiences in the same vein as chapter 2.

Chapter 4 defines a framework for supporting mobile mixed-reality experiences. This is achieved by reflecting on the requirements of the four experiences described in chapter 3, and the related works reviewed in chapter 2, in order to show that supporting mobile mixed-reality experiences requires a combination of dedicated authoring tools and live orchestration. It argues that mobile mixed-reality experiences can be located on a spectrum of varying forms of participation, ranging from performance to game-play and in turn drawing on aspects of computer games, role-playing, street theatre and conventional theatre. It argues that a second dimension is required to encompass the authoring tasks that occur before an experience can begin, and improvisation tasks that occur during an experience. This in turn highlights new challenges for authoring and orchestration tools.

Chapter 5 drills down into how to support experiences through authoring, by focusing on a new tool to create location-based content in relation to the physical environment. It describes the development of the *ColourMaps* system, a tool that allows authors to create location-based content by colouring an image, and which facilitates the construction of complex narrative routes through a city environment, configuration of key positions for players and clear

GPS filtering. It goes on to show how the ColourMaps system has been used to support the central content-based game mechanics in the four experiences.

Chapter 6 explores how to support experiences with live orchestration, by demonstrating the need for, development and use of an orchestration framework that supports the various elements of a public experience in functioning together as a coherent performance. It shows how orchestration is necessary to manage the flow of participants through an experience from beginning to end through the use of dedicated control room tools and behind-the-scenes work practices. It also shows how pre-authored content can be usefully built upon and supplemented with live, improvised content.

Chapter 7 describes the implementation of the four experiences, to show how the tools presented in chapter 5 and chapter 6 are implemented within a larger supporting system. It examines the differing requirements of online and mobile clients, and how judicious system design can support the orchestration task with disconnected operation.

Chapter 8 concludes this thesis by summarising the major contributions and achievements of the research. It reflects on the implications of the research, its contributions to and dissemination and impact within the research and wider community, and avenues for possible future work.

2 A Review of Mobile Mixed-Reality Experiences

2.1 Introduction

This chapter presents a review of background work in order to set this research in the wider context. The next section reviews nineteen mobile mixed-reality experiences, mobile applications and related works that span a wide range of genres, offering differing motivations and functionality. The experiences range from those created by the research community to investigate a specific issue to those developed and deployed commercially, encompassing a variety of application domains from re-workings of classic computer games, tourist guide applications, wide-spread games and artist-led performances. They form only a subset of a rapidly growing domain, and as such there are many experiences that are not reviewed, although those presented are believed to represent the most notable.

The experiences in this chapter are examined to help to highlight common themes, requirements and challenges for supporting mobile mixed-reality experiences in general. For each experience a brief description is given, alongside a review of notable features and supporting requirements.

This chapter concludes with a summary of the functional requirements highlighted by the reviewed experiences. The next chapter will provide a similar review and summary of the four experiences that the author was involved in the development of, enabling chapter 4 to formulate a framework for supporting mobile mixed-reality experiences.

2.2 Related Work

2.2.1 Pirates!

Pirates! (Björk et al., 2001) is an indoor, multi-player, location-based game. The game involves up to eight players each playing the role of the captain of a ship sailing in a virtual fantasy archipelago. The objective of the game is to

complete a number of missions, to trade goods between the islands in the archipelago and to fight with other players' ships in battles.

Pirates! is a location-based game in which the physical indoor game environment is loosely coupled with the space of the virtual archipelago. Players visit islands within the archipelago by walking to a certain position within the physical game area, and meet other players to battle by walking to the same position as them, thus mapping the virtual space to the physical space. The virtual islands are indicated with physical landmarks in the physical space.



Figure 2-1: Pirates! interface

Players are equipped with a handheld computer that runs a game client application. Each handheld computer is connected to a wireless network and is equipped with a radio frequency based proximity sensor and beacon. The static islands are also equipped with a similar sensor and beacon. Each beacon regularly transmits a unique identifier that can be detected only at short range. Using these sensors, the game client can determine the player's position relative to the other game artefacts, islands and other players, and triggers the relevant game mechanic accordingly, for example arrival at an island. This is shown in figure 2-1.

The game client is a dumb graphics rendering terminal, with all game logic being performed by a central game server written in Prolog, allowing dynamic updates to game logic and providing a basic form of game management. Importantly, this enables the game authors to respond on the fly to any

unexpected issues that may arise over the course of the game. The game server also provides player login functions and a high score board.

The authors note that the decision to use relative positioning was deliberate, as it allowed rapid reconfiguration of the game space simply by moving the artefacts and sensors representing the islands, hence allowing greater portability. However, they note that the reliability of the proximity sensors was variable, and as a result players were given the choice of whether to accept each location-based decision made by the system, for example to land at an island or to ignore it.

2.2.2 ARQuake

ARQuake (Piekarski and Thomas, 2002) is an experimental single-player, indoor and outdoor augmented reality game. The aim of the game is to allow the player to play the first-person-shooter game Quake in a physical environment. The original game involves moving around a virtual environment, collecting objects and power-ups and shooting monsters that attempt to attack the player.

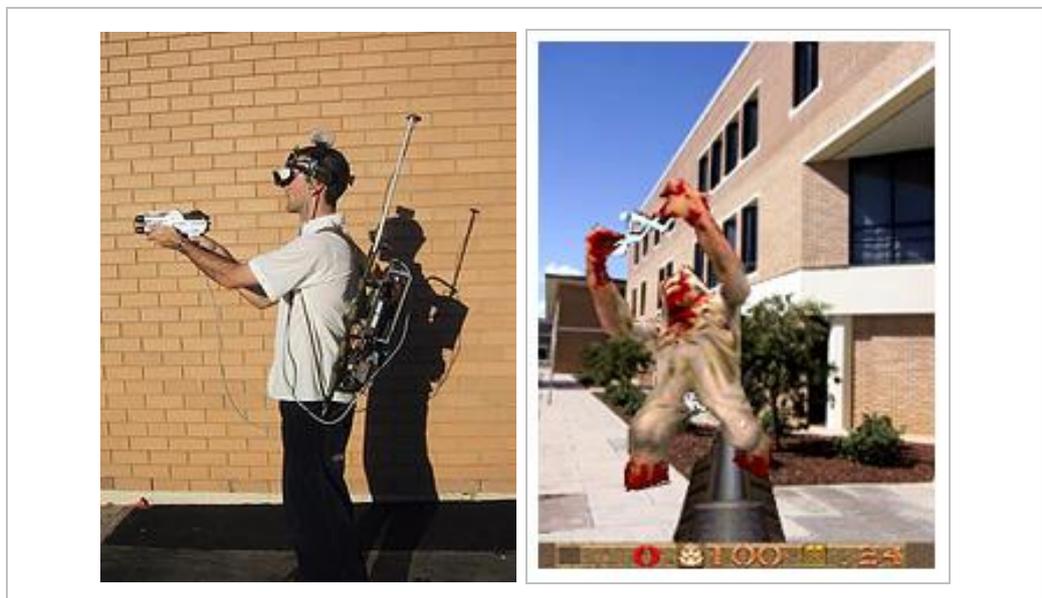


Figure 2-2: ARQuake hardware (left) and in-game view (right)

In ARQuake a representation of a physical space is modelled as a virtual Quake level. The player is equipped with a wearable computer platform, a haptic gun interface, and head mounted display, as shown in figure 2-2. The

system overlays a view of the Quake level from the player's physical point of view onto the display, with the walls of the level rendered transparent to allow the user to see the equivalent physical walls. The virtual walls occlude other virtual artefacts such as objects and monsters to create the impression that the artefacts inhabit the physical space, as is also shown in figure 2-2.

The player is placed in the virtual space using tracking data from a combination of a GPS receiver, digital compass and a fiducial tracking system using glyphs. The authors state that when a player was outdoors and more than 50m away from any buildings GPS and compass tracking provided satisfactory accuracy in that any errors were not immediately visible due to the distance. When close to a building or inside, however, these errors were very apparent and so the system instead had to rely on a large number of fiducial markers attached to the physical space to perform vision based tracking to overcome them.

ARQuake is implemented as a modification, or mod, to the original Quake platform, allowing much of the original game logic to be retained. However, the authors describe initial experiments with a multi-player version, having one player playing a conventional desktop version of the game collaboratively with the augmented player. They note the differing requirements of both players in the form of different renderings of the same shared space, for example the augmented player requiring walls to be rendered invisibly, and in textured form for the online desktop player.

As the authors note, coincidence of the physical and virtual spaces is essential for the game to maintain the fiction of elements of the virtual space appearing in the physical space. To this extent the space had to be modelled from detailed and accurate architectural drawings for each location that the game is played in.

2.2.3 Human Pacman

Human Pacman (Cheok et al., 2004) is a multiplayer mobile mixed-reality game that has many similarities with ARQuake. This time, Human Pacman extends the classic arcade game Pacman. Players must collect cookies and

other artefacts in the virtual environment while avoiding the ghosts, who can kill them.

In Human Pacman, as with ARQuake, players inhabit both a physical space and matching virtual space using tracked, wearable computer systems and head mounted augmented displays, shown in figure 2-3. As they move through the physical space, they also move through the virtual Pacman space. Some game artefacts, such as cookies, exist solely in the virtual space and players must reach the same location to collect them. Others, such as power-ups, also exist within the physical space in the form of small embedded devices that the player must physically collect.



Figure 2-3: Human Pac-Man player and view, with online helper

Players either take the role of Pacman or of a ghost. The game also adds a third group of players to the original game, known as helpers, who occupy the virtual space and play the game using a desktop interface. The helpers have access to additional information, such as the location of game artefacts that they may have seen while exploring the game space, which they may pass onto the augmented players using text messages. Helpers may also collaborate between one another online to achieve a goal.

Augmented mobile players connect to a central game server using a wireless network. The authors note that disconnections in communication often

interrupted the flow of the game, and that outdoor conditions often resulted in poor bandwidth and a high rate of errors on the wireless network. To minimise this, a small amount of system functionality was retained on the mobile player's device to reduce the effect of small network interruptions, mainly the functionality for viewing the static parts of the virtual environment.

The augmentation of the mobile players required very accurate position and orientation tracking. The authors attempted to overcome the tracking errors noted in the discussion of ARQuake by using a fusion of GPS and inertial sensor data to calculate the positions of other players and game artefacts relative to the player. This was necessary to provide a satisfactory augmented experience, which the authors see as being preferable to so-called primitive 2D handheld computer based interfaces, although despite this the augmented reality overlay was considered to be unbearable to play for more than a few minutes, enforcing a short game length.

The virtual space was authored, as in ARQuake, to coincide closely with the physical space. Also, to maximise the availability of the wireless network connection, the game area was carefully selected to be in an area of known, good, network connectivity.

2.2.4 Treasure

Treasure (Barkhuus et al., 2005) is a multi-player location-based game. Players must collect virtual coins that are scattered within an urban area by getting to that position, and place them in a virtual treasure chest. Players may also pickpocket other players to steal their coins before they have been placed in the treasure chest.

Each player is equipped with a handheld computer and a GPS receiver. The handheld computer displays a map of the game area with the position of the player as reported by the GPS receiver, and also the positions of the virtual coins. The handheld computer is connected to a wireless network that deliberately only provides partial coverage of the game area. As the player moves around the game area the network coverage of areas that they have

explored is rendered on the map, as either being disconnected, weakly connected or strongly connected, as shown in figure 2-4.

Coins are scattered randomly throughout the virtual game area. Players may collect coins while disconnected, but they may only be secured in the treasure chest while connected to the network. Similarly, players may only pickpocket other players while connected. The greater the strength of the network connection, the better chance a player has of successfully executing one of these actions. Two players can secure their coins at the same time to gain more points.



Figure 2-4: Treasure interface showing network coverage (left) and coins (right)

Treasure uses the partial wireless network coverage of the game area directly as a form of game play, and coins are automatically placed both in areas with coverage and those without. In this way the game is authored dynamically by the natural arrangement of the wireless network, although the authors made sure that the game area consisted of areas both with and without coverage. This can be said to be analogous to the physical authoring performed by the authors of Pirates! in that the authors may manipulate the placement of wireless access points in order to provide the desired spread of network coverage and non-networked areas.

The authors describe Treasure as a *seamful game*, in that it deliberately acknowledges and exploits the seams within the wireless network, meaning the boundaries between connected and disconnected areas, as a form of game play. This contrasts with other experiences where such seams are treated as flaws that need to be overcome.

The authors ran several trials of the game giving players the chance to play multiple games. They describe how players were observed to develop strategies and tactics over the course of the games to cope with inconsistencies in the technology, including lag in GPS updates delaying the virtual representation of a player's position, and inconsistencies between the on-screen display and a player's actual relationship to other players. With time, players became skilled at noticing these inconsistencies and modifying their game play with different movements and teamwork in order to capitalise upon it.

2.2.5 Pac-Manhattan

Pac-Manhattan (PacManhattan, 2004) is a multi-player mobile mixed-reality experience again based on the classic Pac-Man game. As with the original game, a player must collect all of the pills in the virtual game space, while avoiding four ghosts that also inhabit the space.

In Pac-Manhattan, players inhabit a large area of Manhattan and a matching virtual game area simultaneously. Pills are distributed evenly throughout the virtual game area, and the player, playing the role of Pac-Man, collects them by getting to the equivalent physical location in the real city. Similarly, the role of the ghosts is taken by four other players, also playing in the real city, who must get to the same location as the Pac-Man player in order to catch them. The virtual game space is broadcast over the internet for online spectators.

Pac-Manhattan uses basic technology to enable the mobile aspects of the game. Players are not directly tracked. Instead, each player is assigned a controller who mediates that player's interaction with the virtual space. As the player runs through the physical space, he or she informs the controller of his current location via a voice-call on a mobile phone, shown in figure 2-5, and the controller then updates the position in the virtual game space from a central

control room. Conversely, the controller informs the player of the current state of the virtual space. The virtual game area is a modified version of the original Pac-Man game board, matching the grid structure of a portion of the city to the horizontal and vertical movement of the original game, as is also shown in figure 2-5.

The authors state that they did not use GPS for tracking the players in the physical space as it did not work sufficiently well in the urban environment, being susceptible to multi-path errors. They also did not use a wireless network to serve data to the players, as they could not find an area of the city with consistent coverage over a large enough, area and lacked the resources to set up their own.



Figure 2-5: A Pac-Manhattan mobile player (left) and virtual game board (right)

The five player controllers are members of the game's production team. They mediate game play in the absence of tracking technology by judging a player's position and ensuring concurrency between the virtual game state and the player's view of the game.

2.2.6 Riot! 1831

Riot! 1831 (Reid et al., 2004) is a single-player location-based audio experience. The experience is an interactive historical play based on events that took place in a public square, and was presented to members of the public. As players move around the physical space where these events took place, they hear sounds designed to recreate the experience of these events from different locations within the square.

In Riot! 1831, each player is equipped with a handheld computer, a pair of headphones and a GPS receiver. The GPS receiver positions the player on a map of the square in an application running on the handheld computer. The application then plays sounds based on the player's location to the player using the headphones, creating a personal audio experience. The authors define this application as a *mediascape*, a mobile client that delivers digital media in response to contextual clues such as GPS location.

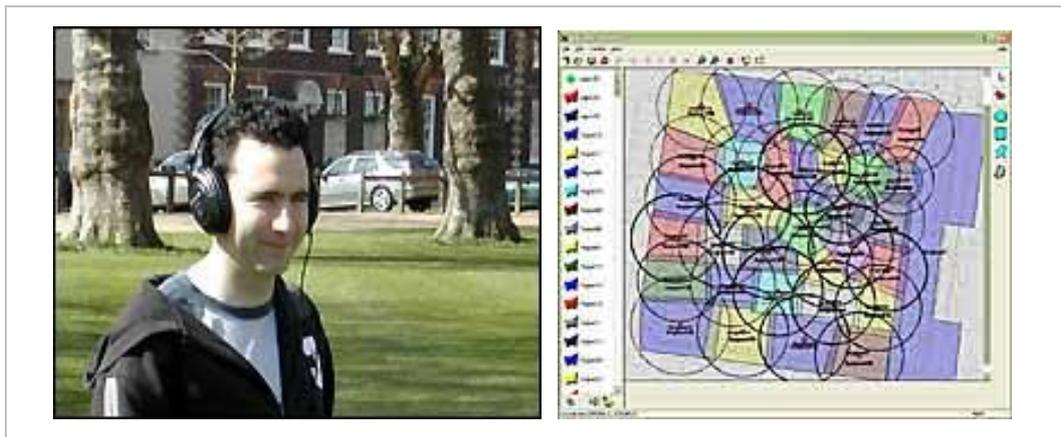


Figure 2-6: A Riot! 1831 player (left) and mediascape authoring tool (right)

Authoring Riot! 1831 involved assigning sound samples to specific physical locations using a tool developed by the authors, also called Mediascapes and part of the Mobile Bristol application framework, which is shown in figure 2-6. As a non-linear experience, the player is free to move throughout the space by his or her own route, with the audio content playing accordingly, sometimes for over an hour. The authors note that as the number of other people in the area changed, or the weather changed, people would either move randomly through the space or take a number of commonly used paths, and as such the mediascape had to be authored to support both non-linear and pseudo-linear content arrangements to some extent.

Members of the public took part in Riot! 1831 without intervention from the production team, meaning that once set up the authors did not play a direct role in directing a player's experience. However, the authors hint at problems encountered with using GPS for location tracking. High buildings and trees obstructed the GPS signal leading to poor accuracy, and therefore incorrect audio samples being played, or a lack of consistency between co-located

players who would hear different samples. On these occasions the game operators would intervene and ask the players to abandon the work and to return to play again at a later time when conditions were more favourable.

2.2.7 CitiTag

CitiTag (Vogiazou et al., 2005) is a multi-player location-based game. It is a team game played in a large area of a city. The game is a simple tag game; when a player gets close enough to a member of the opposing team they can tag them, trapping them virtually in their current location. Players can be untagged by a member of their team when in the same location. The game is won when all players of the opposing team have been tagged.

Each player is equipped with a handheld computer connected to a wireless network and a GPS receiver. An application running on the handheld computer relays the player's current location as reported by the GPS receiver to a central game server over the wireless network. The game server judges whether there are any other players within a given distance that can be tagged or untagged depending on which team they are a member of, and returns this data to the player's device where it triggers the relevant interface. Figure 2-7 shows the three interfaces; the default screen, being tagged and looking for a tagged member of the same team.



Figure 2-7: CitiTag handheld computer interface

The CitiTag system implements some simple functions to handle situations where a player's GPS or wireless network connection fails in some way. If the

GPS receiver does not have a valid fix, in that it cannot report its current location, then the application reports this fact and the game server flags the player as unavailable to be tagged. Upon disconnection from the wireless network, whether with a clean disconnection or due to lack of updates from the application, the game server again flags the player as unavailable. On disconnection the player is presented with a login screen forcing the player to manually attempt to reconnect to the game. Players cannot play while disconnected; on reconnection the current state of the game is sent to the application to update it and to continue playing.

The CitiTag system provides an administration client. This connects directly to the central game server and provides general orchestration functions for the game authors, including starting, pausing and ending a game, sending a text message to all of the players' devices, modifying the tagging radius and displaying statistics regarding the current game state.

2.2.8 CatchBob!

CatchBob! (Nova et al., 2005) is an experimental location-based multi-player collaborative game. The game design is basic; three players must catch a fictitious virtual player known as Bob by positioning themselves in a triangular formation around him. Each player is equipped with a handheld computer that shows the current location of all three players on a map, and also allows the player to annotate the map with a suggestion of the direction that another player should take. These annotations are distributed to all of the other players. Figure 2-8 shows the handheld computer interface.

CatchBob! uses a wireless network to provide both connectivity and positioning for the handheld computer. Position is determined by triangulating wireless network signal strength based on the known positions of the network base-stations. This was used rather than GPS so that the system would provide positioning indoors, and the author states that the system provided position readings accurate to between 5 and 20 metres. To configure the game for each physical location the author had to survey the area, identifying and locating existing wireless network base stations on the map. The virtual position of Bob also had to be configured.

The system provided a logging function which enabled the author to study the results of the game after the event. The author notes that during the introduction of the players to the game and equipment, if the equipment was not demonstrated with absolute confidence then the players would not have absolute confidence in it either, and would be more likely to suspect it of failing - indicating that the author had to take the role of a performer to ensure the smooth running of the game from the outset.



Figure 2-8: CatchBob! interface showing user annotations and network connectivity monitor

Problems occurred when playing CatchBob! as players had the pre-conception that it provided very precise position data, and therefore were disturbed when their reported position on the map, or their reported proximity to Bob changed even when they were standing still. There was nothing on the interface to indicate that the position was not accurate. Regarding the wireless network connection, the author seems surprised at how, even when the network was also being used to provide positioning, the connection was not strong enough to send and receive data. Players also commented on this fact, and at some points commented that the application was broken as it did not reflect the state of the connection. The author responded by adding a network connectivity indicator to the interface.

2.2.9 Botfighters

Botfighters (Its Alive!, 2000) is a commercial multi-player location-based mobile phone game. The game involves shooting other players by sending a text message with a mobile phone. If the other player is within range, that is if the two players' phones are within a certain distance of one another, then a hit is scored. Botfighters also implements an online interface, shown in figure 2-9, where players can log on and configure or upgrade their characters. The specifics of a character, such as which weapon they are carrying, adjusts the likelihood of success when attacking another player.



Figure 2-9: Botfighters online configuration interface (left) and mobile phone interface (right)

Botfighters uses the ID of the mobile phone cell to which a player is connected to determine the distance between players. The first version of the game used SMS messages for all aspects of game communication. Players send a message to request information about other players in the vicinity, and so indicated their entrance to the game. They then sent a further message to shoot at a selected player. The use of SMS messages as a transport was at times susceptible to message delivery delay, effectively stopping the player from taking part in the game.

The second version of Botfighters attempted to rectify the problem of SMS message delay by introducing a game application that was installed on a player's phone. The application now communicated with the central game server using a GPRS connection, and introduced some of the configurable

content that was previously available on the website to the mobile phone application. Some criticisms of Botfighters, however, are that the latency of the connection is still noticeable when playing with players in close proximity and also that in this situation the positioning system is not accurate enough.

Botfighters does not require any location-based authoring as game play depends on the density and relative positions of players. Players are free to play whenever they wish.

2.2.10 Mogi

Mogi (Licoppe and Guillot, 2006) is a commercial mobile mixed-reality game. The game involves finding virtual artefacts within a large physical city area, then trading them with other players. The goal of the game is to complete collections of items through a combination of collection and trading. Players can communicate with one another via a global buddy system and instant text message.

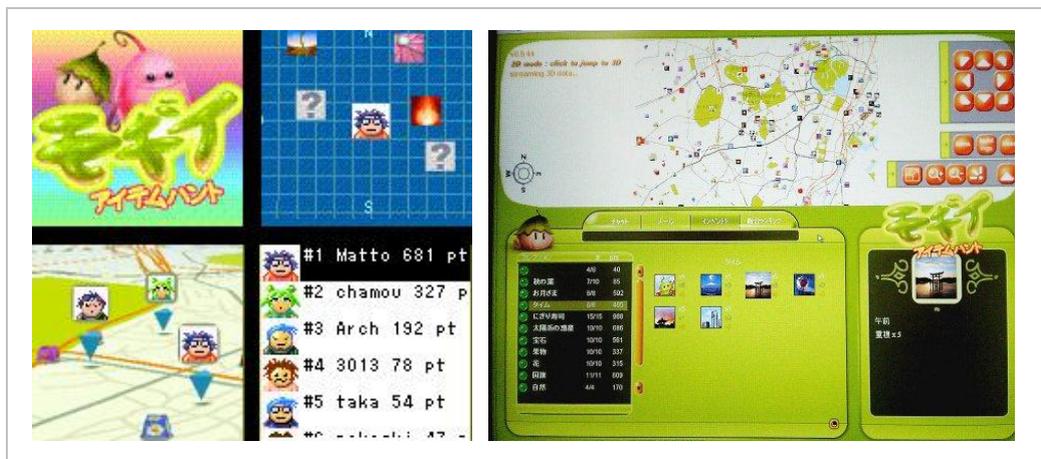


Figure 2-10: Mogi mobile phone interface (left) and online interface (right)

The game is played in Tokyo, where an application running on each player's phone shows a stylised grid map of their current area, other players and where there are items to collect. The position of the player is reported either by cell positioning as described in Botfighters, when only a crude position or orientation is required or by the mobile phone's built in GPS receiver when a player is trying to collect a specific item.

Mogi provides an online interface for players to use, shown in figure 2-10. However, unlike Botfighters, the Mogi online interface provides additional functionality that is not provided by the mobile phone application. This takes the form of a larger, more detailed map. Rather than being just an interface for accessing a player's personal data, the online interface also allows communication with all other players within the game, providing a shared mixed-reality environment.

The extended online interface allows online players and mobile players on the street to collaborate in completing their collections. Team play can then be facilitated with more dedicated players playing online, orchestrating a group of mobile players based on their wider knowledge of the game, enabling player-led micro-management.

As a long-term, publicly accessible game, Mogi players are able to play whenever they wish, and as such are largely responsible for managing their own play.

2.2.11 Urban Tapestries

Urban Tapestries (Lane, 2003) is a technological framework for creating mobile experiences. The framework aims to provide a system for the publicly creating, publishing and experiencing location-based content. Urban Tapestries also aims to be dynamically interactive, allowing its users to supplement existing content as they experience it, and to annotate the paths that people take through a city area.

During the development of the system the authors initiated what they describe as *bodystorming experiences*. These involved taking a map of a city area, and having users annotate the map with drawings and post-it-notes indicating areas of interest and regularly journeyed paths. This technique was used to gain ideas about what users might create with the framework, and how they reacted to the content created by others, without interference from any technological problems that may arise from a prototype. Users came up with a wide variety of users for the system, from trails defining the mood of the creator as they

walked to a description of a route traditionally taken by sheep drovers through the city.

Urban Tapestries ultimately aims to be technologically agnostic, in that its principles should be usable by a wide variety of different technologies. However, in the initial prototype users were equipped with a handheld computer with a wireless network connection and a GPS receiver to provide positioning. The handheld computer ran a custom application written in Flash to provide an interface. The authors state that future versions would use Bluetooth and GPRS or 3G to access content in areas not covered by the wireless network, indicating that almost half of the targeted city area was not covered; leaving large gaps between connected areas where the framework would not work.



Figure 2-11: Urban Tapestries mobile interface (left) and online map showing trails (right)

The framework allows a user to select a theme of content to follow, for example social threads linking a number of locations, as shown in figure 2-11. They then receive a map on their mobile device indicating the locations associated with it, generated from a GIS map-data repository. They may either follow a previous trail through the map or receive alerts when they reach a specific location, and see the associated multimedia content.

Urban Tapestries allows the user to upload their own path through the city, add extra locations by tagging their current location or upload new media to the system, although it is unclear how this is handled in areas without network

connectivity. This mechanism allows both individual application authors and subsequent users to author for the system as they play.

The framework provides an online, web-based interface that allows users to edit and share trails or location-based content that they have created. In this way, an online user could orchestrate or specify an experience for a mobile user. However, the onus is on individual application authors to specify and arrange this.

2.2.12 GeoTracing

GeoTracing (GeoTracing, 2005) is an application framework for creating location-based multimedia experiences. GeoTracing has similarities to the Urban Tapestries framework described above, and can be described as an instance of Urban Tapestries built using a different technology infrastructure. Experiences may be for single or multiple users, and are again based on a user's movement through a physical landscape with location-based media annotations.

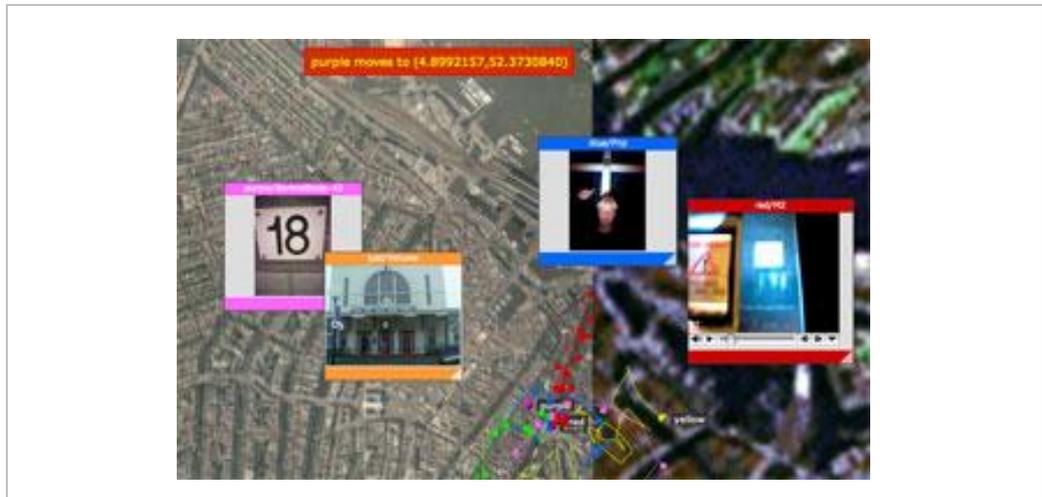


Figure 2-12: GeoTracing online interface showing trails and location-based media

Several applications have been developed using GeoTracing to date. (GeoSkating, 2005) involves skaters uploading photographs and information about the quality of road surface along a given route. (GeoSailing, 2005) and (GeoBiking, 2005) are two more applications with the same theme, but different modes of transport. Sense of the City (Broecke, 2006), as with Urban Tapestries, involves users annotating their everyday routes through a city.

The GeoTracing framework consists of a *MobiTracer*, which is a mobile phone with a GPS receiver. The mobile phone uploads position data and routes using a GPRS connection to a central server. There are two online components to the system. The *WebEditor* is a web interface that also allows the uploading of routes collected offline from standalone GPS receivers, and annotation using a map interface. The *WebViewer* is a web interface that displays maps with routes and annotations, and also allows live tracking of active mobile users, in a read-only mode.

Finally, the framework provides location-based content serving to mobile users based on existing annotations. When a user gets close enough to an annotated site, the relevant annotation or media is displayed. However, the system assumes that the user is connected to the central server constantly to access this functionality, and it is not discussed what happens if this is not possible or fails, and what support is given to users when the system fails. Several routes from the Sense of the City application, viewed with the WebViewer, appear to exhibit GPS errors, with one spurious update causing a route to deviate by a substantial distance. Unfortunately, it is not described if or how this affected users following the routes.

As with Urban Tapestries, GeoTracing allows both authors and users to author for different applications in the same way, by annotating physical space while using the application itself. Also, GeoTracing would allow the use of its online interfaces for game management and orchestration. However, this functionality was not inherently designed into the system as it does not directly support communication between users, or authors and users.

2.2.13 Real Tournament

Real Tournament (Mitchell et al., 2003) is a prototype location-based multi-player game. The authors describe the game as an augmented reality game, although it has a large amount in common with other location-based projects. Players inhabit a large, open, physical space. The aim of the game is to work with other players to shoot virtual monsters that inhabit a connected virtual space that corresponds to the same area. Some of the monsters require

collaboration between players to kill, by requiring several shots from different players.

Each player is equipped with a handheld computer housed in a large plastic gun casing, shown in figure 2-13. The gun is equipped with a GPS receiver and a digital compass to obtain the player's location to place them within the virtual space. The handheld computer shows a map of the virtual space centred on the player's reported position, with geographical surroundings, virtual monsters and the reported positions of other players. Players shoot monsters by facing in the appropriate direction and pulling the trigger on the gun.



Figure 2-13: Real Tournament gun interface (left) and map (right)

The gun connects to the game using a wireless network connection. If Wi-Fi is available in the particular area that the player is in then it is used, failing that the gun seamlessly switches to a slower, but hopefully more widespread, GPRS connection. The game replicates the state of all monsters and players within the game to all of the connected guns using a peer-to-peer system to maintain consistency.

The focus of Real Tournament is primarily on how to develop an application to use the new IPv6 infrastructure allowing transitions between different connectivity infrastructures, in this case Wi-Fi and GPRS, and the authors do not reflect on the issues raised by some of the other experiences, such as how the relationship between the game world and the physical world was authored, or how players responded to disconnections and subsequent network switching.

2.2.14 GUIDE

GUIDE (Cheverst, Davies et al., 2000) is a location-based electronic tourist guide application. The experience provides a context aware guide for tourists as they move around a city, giving location specific information on a variety of tourist attractions, in the form of web pages, text and images viewed on a handheld computer, as shown in figure 2-14.

Each user is equipped with a handheld computer, which connects to a wireless network. The network consists of a number of uniquely identifiable Wi-Fi base stations, which an application running on the handheld computer uses to provide coarse location awareness, for example reporting that the user is at a key area or building within the city.

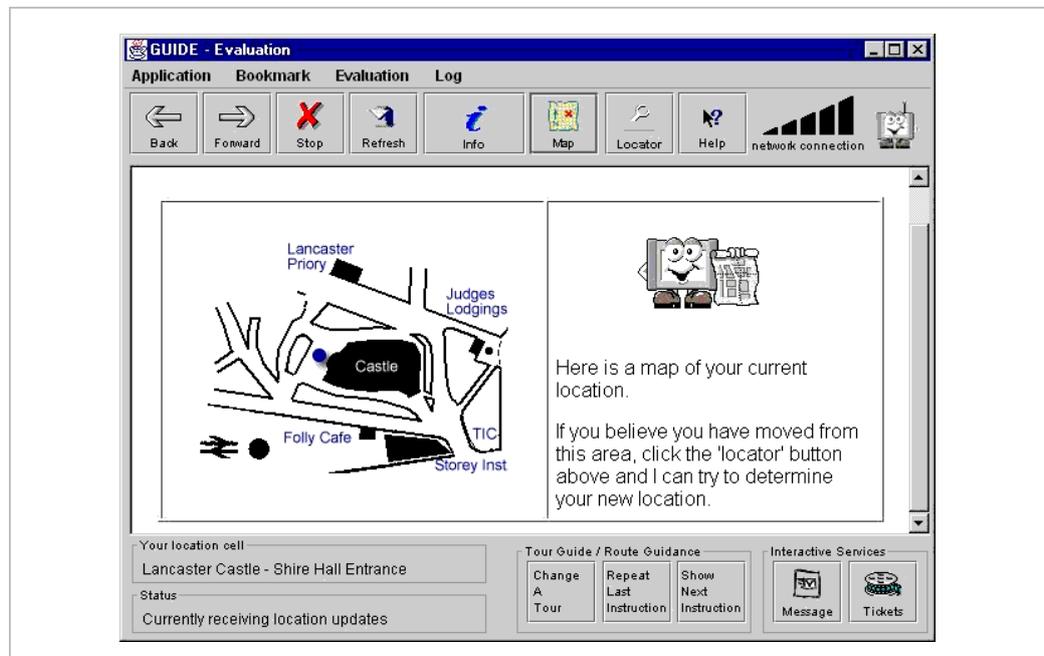


Figure 2-14: GUIDE interface showing map override function

GUIDE highlights some important design considerations regarding content delivery and connectivity. In terms of content, the range of possible applications and variety of content within these applications is large. Some of the content may need to be dynamic and change over time, or be interactive. Furthermore, the system is may be used by up to one hundred users at any one time.

Regarding connectivity, the authors of GUIDE were aware that there would be occasions when a user of the application moved into a location where there was little or no wireless network coverage. As GUIDE uses the wireless network to provide position reporting, the application would also not know where the user was, or be able to serve him or her connected content. Most interestingly, this issue was noted by the authors during the design phase of the prototype, rather than during deployment and being dismissed as a future problem to be overcome. Rather than stating that they required a better network connection and better positioning, the application was re-designed with these issues in mind, namely with the authors adding a function to enable the user to override the application's sensed location with a report made using an on-screen map in the case of loss of connection.

2.2.15 Mobile Experience Engine

The Mobile Experience Engine (Biswas et al., 2006a) is a framework for developing location-based new media experiences. It is a software toolkit that attempts to support the iterative development of context aware mobile multimedia applications. The authors focus, however, on both development of software by technologists and creative development by artists and designers, and supporting an iterative development cycle involving both. They note that the requirements of both parties will change during the development of an application and so the framework needs to allow for this, without one party holding up the workflow of the other.

The framework consists of a prototype generator and a mechanism for producing intermediate prototype visualisations for use by the application designer or artist. The software requirements of the application are formally described using the MEE content description language, which is then used to automatically generate source code and library templates. If the source code cannot be generated, or is unsuitable for the application, then it is written by a software engineer and added to the available code base of the prototype generator. While this is happening, the artist or designer of the application may work with the visualisation component. This also uses the formal description of the application, but produces a low fidelity prototype using scalable vector

graphics that can be used to test and refine the applications functions as a whole.



Figure 2-15: Trickster mobile interface

Applications are described, or authored, using the MEE content description language, which is used to describe an application's function using XML. An application consists of a number of contexts, and each of these is defined by transitions and resulting actions within the XML definition. For example, transitions, or entry into a particular context, may be driven by readings from position sensors, and the resulting action may be displaying a media clip.

The authors describe two example applications that have been created using the Mobile Experience Engine. The Trickster game (Biswas et al., 2006b) is a single player game that involves the player taking the role of different animals within a park, shown in figure 2-15. The player is equipped with an accelerometer to detect gestures, which is linked to a handheld computer running the application created using the framework. The application presents a number of contexts that involve the player carrying out various activities within their animal role, with transitions being triggered by the player making the correct animal gesture. The second example is Situated Editor (Donaldson and Gauthier, 2006), an application that allows a designer to create multimedia clips, annotate them with location data, and then add them to a central server.

The resulting location-based media environment can then be downloaded onto a location-aware mobile device to be experienced by a user. Both the authoring system and the location-based runtime triggering system are described and implemented using the MEE content description language.

These two examples are interesting as, unlike many other applications and frameworks; they highlight the rapid prototyping of mobile experiences whilst allowing for input from the designer in the iterative development process. However, the focus of the framework is still very much on the software engineering process, and it is unclear to what extent the needs of the designer are provided for during the deployment of the experience.

2.2.16 Desert Rain

Desert Rain is a multi-player mixed reality performance (Koleva et al., 2001). While it is not location-based, it is notable as a touring, heavily orchestrated piece mixed-reality application that requires the direct involvement of its authors to operate. Desert Rain is a new-media piece which is played by six players at a time in a game that lasts for twenty minutes. Players embark on a mission to find six targets within a virtual environment and then find an exit within the twenty minute experience. The virtual environment is projected onto six water curtains, one in front of each player.

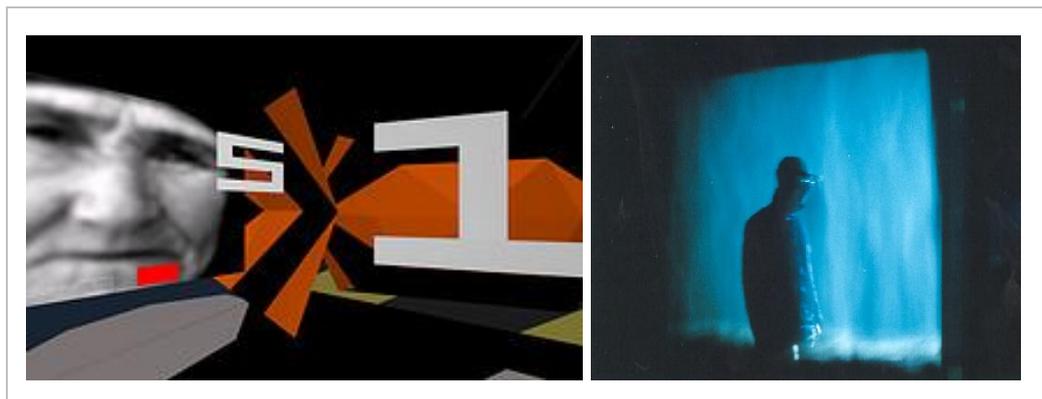


Figure 2-16: Desert Rain virtual environment (left) and rain curtain (right)

When a player finds his or her target, a performer steps through the water curtain and hands the player a swipe card. Once all players have found their targets, or their allowed time has elapsed, the players move through a corridor filled with sand to a recreation of a motel room. The room contains a swipe

card reader and a television; using their swipe card each player can view a video clip of an interview with their target.

Each player is embodied in a collaborative virtual environment, with their viewpoint projected onto the rain curtain, shown in figure 2-16. Players control their viewpoint within the virtual environment by standing on a footpad which acts as a joystick. Players can communicate with one another with an audio system that is spatially attenuated with distance in the virtual environment.

While Desert Rain is a touring piece of work, the content remains static, and as such does not require modifications for a new location. However, Desert Rain is not just a standalone computer system; the system and its use are embedded within a rehearsed performance piece. The authors note that it critically relies on a production team of both actors and unseen helpers to support the experience, through interventions, monitoring and encouraging players to help one another.

To support this, the Desert Rain system provides a number of integrated management tools. Each player's viewpoint is replicated on a screen in a control room for monitoring. An audio mixing desk allows performers to listen, connect players and join in to a player's conversation. Players can also be monitored visually from behind the water curtain.

2.2.17 Feeding Yoshi

Feeding Yoshi (Bell et al., 2006) is a multi-player location-based seamless game. In Feeding Yoshi, players are divided into teams of four. Each team must collect as many points as possible by feeding a Yoshi the fruit it desires. Players collect a seed of a certain type of fruit from a Yoshi by visiting a certain location, and then sow it in a plantation which will start to generate the fruit. The fruit can then be taken back to the Yoshi in exchange for points. Yoshi is played intermittently over a relatively long period of time, for example over the course of a week.

Yoshis and plantations are automatically generated from the identity of existing wireless networks scattered throughout the city. Each player is equipped with a handheld computer that scans for these wireless networks. The handheld

computer displays a map, and when a network is discovered the player can pin either the Yoshi or the plantation to its discovered location on the map. Finally, players upload their scores to a central game website, where they may also view the score of their and other teams.

Feeding Yoshi does not provide any explicit authoring functionality, as the game content is automatically generated from existing wireless networks. Once the teams and game were organised, players played independently and without intervention, except in situations where the application or handheld computer failed, in which cases the authors provided remote phone support.



Figure 2-17: Feeding Yoshi mobile interface showing a plantation (left) and a Yoshi (right)

2.2.18 Big Urban Games

There have been several large scale location-based city games that are usually mentioned in the context of mobile mixed-reality experiences. Big Urban Games are usually team based, and are location-based in that they involve players exploring, or operating in, a large city area. However, there is no overlapping virtual environment, and usually the position of the players does not form a fundamental part of the game.

The Go Game (Wink Back, 2000) is an example of a Big Urban Game. It is a team game, where each team is equipped with a web-enabled mobile phone

and a digital camera. The game lasts for a few hours, during which the teams receive challenges on their mobile phones. Challenges consist of finding clues hidden in the city to solve a puzzle, interacting with planted actors, or performing some kind of stunt often including a passer-by and then capturing and uploading the event with the camera. Finally, the teams gather to review the results of the game and to vote for a winner, based on who produced the most original or amusing results.

There are several variations of the Go Game in operation. *Navigate the Streets* (Shore, 2003) involves teams solving clues and taking photographs of the solutions, aided by being able to conduct research online using mobile devices. *Conqwest* (Lantz, 2003) involves several large teams manoeuvring large inflatable totems through the streets, while taking and uploading photographs of Semacodes (Woodside, 2003) containing unique URLs to declare their positions and capture territory using mobile phones.

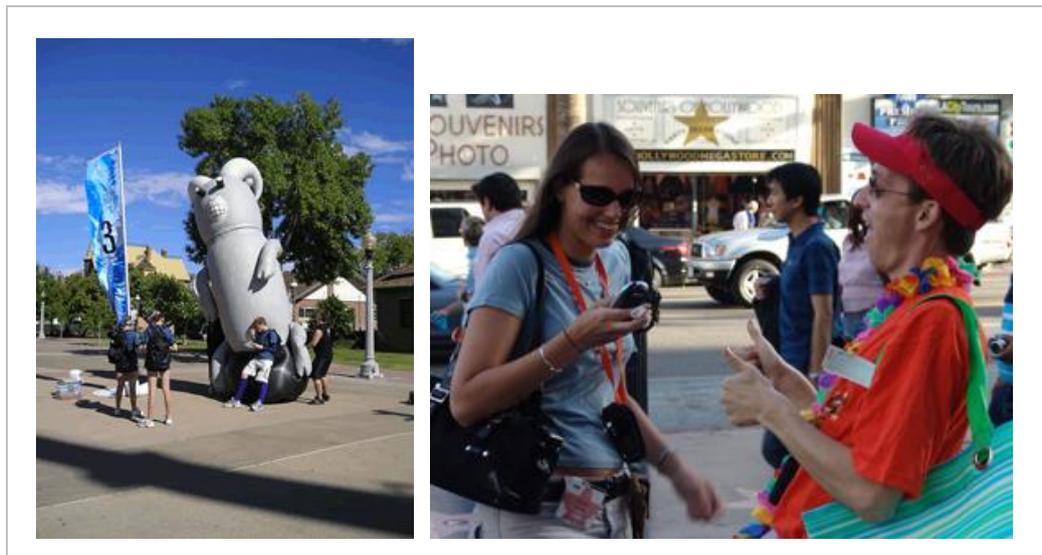


Figure 2-18: Conqwest players (left) and the Go Game (right)

The emphasis of these games is not, however, on technology or location sensing, but rather on encouraging the players to work as a team. Despite being promoted as making heavy use of wireless networking or location-based technology these games often focus on team building exercises, using a small amount of consumer technology, for example mobile phones or cameras.

The Go Game and its counterparts are organised on a per-game basis for its players, with the authors tailoring clues, actors and challenges to the location or specific requirements of their clients. While not being reliant on constant connectivity or position tracking, all of these games involve a central point of organisation, which in turn involves authoring and setting up the game for each location it is played in, and directly orchestrating and operating it while it is being played.

2.2.19 GeoCaching

GeoCaching (GeoCaching, 2000) is a large-scale multi-player location-based game. Fundamentally a treasure hunt game, mobile players place a cache anywhere in the world, which is a container containing a logbook and a number of small items. The location of the cache is posted on a central website in the form of a set of GPS coordinates and optionally a set of clues. Other players take the coordinates from the website and, using a GPS unit, attempt to find the cache. Upon finding the cache they note their visit in the logbook and exchange some of the items for some of their own, as shown in figure 2-20.



Figure 2-19: A GeoCaching cache (left) and players using a GPS receiver (right)

GeoCaching consists of a large community of players with GPS receivers, and a central online forum and database for sharing cache locations. Some players supplement their equipment with a handheld computer showing the location of the cache on a map, updated using the GPS receiver.

In GeoCaching, players are authors, creating caches for other players to find, as well as playing and managing their own game-play. It is not centrally managed

in that, as with other long-term games that have been mentioned, players play whenever they wish. It has a very large user-base consisting of many thousands of players.

2.3 Associated Work

2.3.1 Alternate Reality Games

Alternate Reality Games are commonly associated with mobile mixed-reality experiences and games. These are cross-media games that primarily involve online resources such as websites, and many involve real world artefacts and locations. Alternate Reality Games such as Perplex City (Mind Candy, 2005) or I Love Bees (42 Entertainment, 2004) are considered to be outside the scope of this literature review as while many of these involve mobile technologies or elements of performance, the main focus of the genre is significantly removed from mobile mixed-reality experiences.

2.3.2 Indoor Augmented Reality Games

There have been several notable augmented reality tabletop applications that attempt to merge physical and virtual game play. For example, RV-Border Guards (Ohshima et al., 1999) involves three players surrounding a physical game board equipped with head mounted displays, showing an overlay of invaders within a matching virtual environment and requiring the players to shoot the virtual invaders using a physical interface. Similarly, the STARS framework (Magerkurth et al., 2003) for developing pervasive applications provides enhancements to traditional tabletop board games or new tabletop games using a combination of handheld computers and augmented reality.

However, these applications are considered to be outside the scope of this literature review for a number of reasons. While introducing elements of mixed-reality, they are not mobile and so are subject to different constraints and technological issues. Secondly, these applications are generally deployed as laboratory technology demonstrators rather than primarily as playable works.

2.4 Summary of Requirements

This section concludes by presenting fifteen requirements that have been highlighted by the mobile mixed-reality experiences and applications reviewed in this chapter. These requirements also serve to provide terminology for referring to common supporting elements, and also to begin to consider which requirements are most critical for supporting an experience.

Players can participate in an experience in a variety of different ways:

- All of the experiences involve **mobile players**. This refers to a primary group of players who occupy a physical space, whose physical location is reported to and affects the experience, and who predominantly play using a handheld computer.
- Several of the experiences involve **online players**. These are players that inhabit an equivalent online, virtual space using a PC, and are not mobile.
- Several of the experiences are based around a **mixed reality environment**. Specifically, this refers to both online and mobile players inhabiting a shared representation of the same space, both physical and virtual.
- The majority of the experiences are **multiplayer**, in that they primarily involve interacting with other players to progress, as opposed to being single-player, where it is possible for a single player to complete the experience without interacting with other players.
- The majority of the experiences are **connected**. This means that players are connected to a central server, either through wired or wireless network connections. Some experiences that are not multiplayer may still require this to support centralised game mechanics and management.

Mobile players can use a range of interfaces to interact with an experience:

- Two of the experiences support a **3d overlay** as a primary display mechanism for mobile players. Mobile players see a representation of a virtual environment overlaid on their viewpoint using augmented reality techniques; usually using a head mounted display and associated tracking technologies.
- A number of the experiences support a **mediascape** as a primary display mechanism, for example Riot! 1831 and Urban Tapestries. Mobile players receive media including images and sound effects based on their changing position in the physical environment.
- The majority of the experiences support a **2d map** as a primary display mechanism. Mobile player see a representation of a mixed-reality environment presented as a map on a handheld mobile device, which commonly also shows their embodiment within the environment and is centred on the player's current location.

An experience may revolve around a variety of forms of content:

- The majority of the experiences involve **location-based content**. This is any content such as text, images or sound that is triggered by a mobile player's absolute position within the physical game space.
- Leading on from this, many of the experiences involve **terrain-based content**. This is location-based content that takes into account the physical features of the surrounding environment, for example the roads and building shown on a map or aerial photograph.
- Conversely, a minority of the experiences involve **relative position based content**. This is triggered based on a mobile player's position relative to another player or artefact, regardless of where they both are in the physical or virtual space.
- A minority of the experiences involve **user geo-annotation**. This involves mobile players creating their own location-based content within an experience by marking specific locations or annotating them

with fragments of media, which can then be distributed to and experienced by future players.

	Mobile Players	Online Players	Mixed- Reality	Multplayer	Connected	3D Overlay	Mediascape	2D Map	Location-Based	Terrain-Based	Relative Content	User Geo-Annotation	Publicly Deployed	Managed	Free Participation
Pirates!	X			X	X			X			X			X	
ARQuake	X	X	X	X	X	X			X	X				X	
Human Pac-Man	X	X	X	X	X	X			X	X				X	
Treasure	X			X	X			X	X	X				X	
Pac-Manhattan	X	X		X				X	X	X			X	X	
Riot! 1831	X						X		X	X			X	X	
CitiTag	X			X	X									X	
CatchBob	X			X	X			X	X	X				X	
Botfighters	X			X	X			X			X		X		X
Mogi	X	X	X	X	X			X	X	X			X		X
Urban Tapestries	X	X	X		X			X	X	X		X	X	X	X
GeoTracing	X	X	X		X			X	X	X		X	X	X	X
Real Tournament	X			X	X			X			X			X	
GUIDE	X				X		X			X	X		X	X	
MEE	X						X		X		X	X		X	
Desert Rain	X		X	X	X								X	X	
Yoshi	X			X	X			X			X		X		X
Big Urban Games	X			X						X			X	X	
GeoCaching	X	X							X			X	X		X

Table 2-1: Summary of experiences and their requirements

Participation in an experience can be managed in a variety of ways:

- Several of the experiences have been **publicly deployed**, in that members of the public can participate “in the wild”. This is an important distinction as many experiences are developed as experimental research projects that are not accessible outside of the

laboratory. For example, this is a key distinction between ARQuake and Botfighters.

- Several of the experiences require **managed participation**, and involve continuous management by their creators. This is usually a requirement for experiences that are operated as short-lived events, and where creators directly orchestrate the actions of players as a core part of the experience, for example with the use of performers in Desert Rain.
- Conversely, a minority of the experiences involve **free participation**, and where an experience can operate autonomously without direct input from its creators. An example of this is an experience that players may dip in and out of at times of their choosing, usually with their own apparatus, as seen in GeoCaching.

Table 2-1 shows how these requirements relate to each of the experiences reviewed in the previous sections.

2.5 Conclusions

This chapter has reviewed a selection of experiences that represent the state of the art of mobile mixed-reality experiences, location-based mobile applications and games, in order to set this research in the wider context. This chapter has shown that these experiences span a broad range of application areas, from laboratory demonstrations to mass-participation street games, encompassing a variety of interaction methods, displays, content and participation.

In light of this, however, this chapter has highlighted fifteen general requirements that are fundamental to the operation of many of these experiences. The next chapter describes the mobile mixed-reality experiences that have been developed by the author as part of this thesis.

3 Four Foundation Experiences

3.1 Introduction

This chapter introduces the four mobile mixed-reality experiences that have been supported by, and led to, the research presented in this thesis. Later chapters will focus on the key challenges and solutions arising from directly supporting these experiences; this chapter describes the end-user experience of each, giving a brief historical overview of the public deployment of the experiences. These experiences are heavily drawn upon in later chapters to provide concrete examples of the use and viability of the research to support mobile mixed-reality experiences in the wild.

This chapter describes four mobile mixed-reality experiences that have been co-developed by the author between 2001 and 2004. *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank* form a series of artistic performances that has resulted from a collaboration with the artists group Blast Theory as part of the Equator Citywide Performance project. *Savannah* is an exploratory educational experience for school children. It should be noted at this stage that the design and construction of the online interface components in *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank* were the result of a collaboration with Nick Tandavanitj of Blast Theory, and the mobile interface in *Uncle Roy All Around You* with Jon Sutton of BT Exact.

Throughout this chapter participants in the experiences are referred to as **online players**; if they take part in an experience primarily online, through a virtual environment, or **mobile players**; if they take part in an experience primarily in the physical environment using a mobile device.

This chapter concludes by giving a summary of the functional requirements highlighted by the four experiences, using the same categorisation as the previous chapter.

3.2 Can You See Me Now?

The first mobile mixed-reality experience in the series of artistic performances is a chase game called *Can You See Me Now?*. Up to twenty online players

inhabit a virtual model of a real city, through which three mobile players in the real city chase them. Both groups of players can communicate with one another via this mixed-reality environment. When a mobile player reaches a physical location in the real city that an online player occupies in the virtual city, then that online player has been 'seen' and leaves the game. The mobile players are professional performers, and the main objective of the experience is to engage and excite the online players by giving them a rich sense of the performers' experience of the city as they run through it, and to reveal how movements online affect their physical actions.

Can You See Me Now? was first exhibited in Sheffield, UK in 2001. Since then it has been refined cosmetically, but the underlying game design has remained the same.

3.2.1 Online Experience

The experience is typically run for two to three hour long performances, twice a day for up to a week. To participate, online players visit a website at **www.canyouseemenow.co.uk** where they can find out more information about the experience, the schedule of performances and join the experience if it is running. Online players participate using a Shockwave client embedded in a web-page, which is designed to be useable over a dial-up internet connection as well as broadband. The client asks the online player to enter their name, and also the name of a person that they have not seen for a long time, yet they still think of. The client then attempts to join the experience; if the experience already has twenty active online players, then the player is held in a queue until there is a space available. The client displays the player's position in the queue as an indicator of how long they will have to wait before a space becomes available.

The online experience consists of four main components; the virtual city, communication between players, game play and an offline archive.

Virtual City: On joining the experience the online player is dropped at a random position in a virtual model of the real city in which the game is hosted. The game is played in a relatively central area of the city, in a space of up to

one square kilometre that often consists of both open spaces and narrow streets lined with tall buildings. The virtual city is a 1:1 representation of this space, containing key features such as large trees, buildings and roads. In the first version of Can You See Me Now? the virtual city was presented as a two-dimensional schematic, with players represented by square icons, as shown in figure 3-1. Later versions replaced this with a more sophisticated three-dimensional model of the space, with players now represented by animated, walking avatars with an optional zoomed out view, as shown in figures 3-2 and 3-3. Mobile players appear as either orange icons, or later as black avatars highlighted in red.

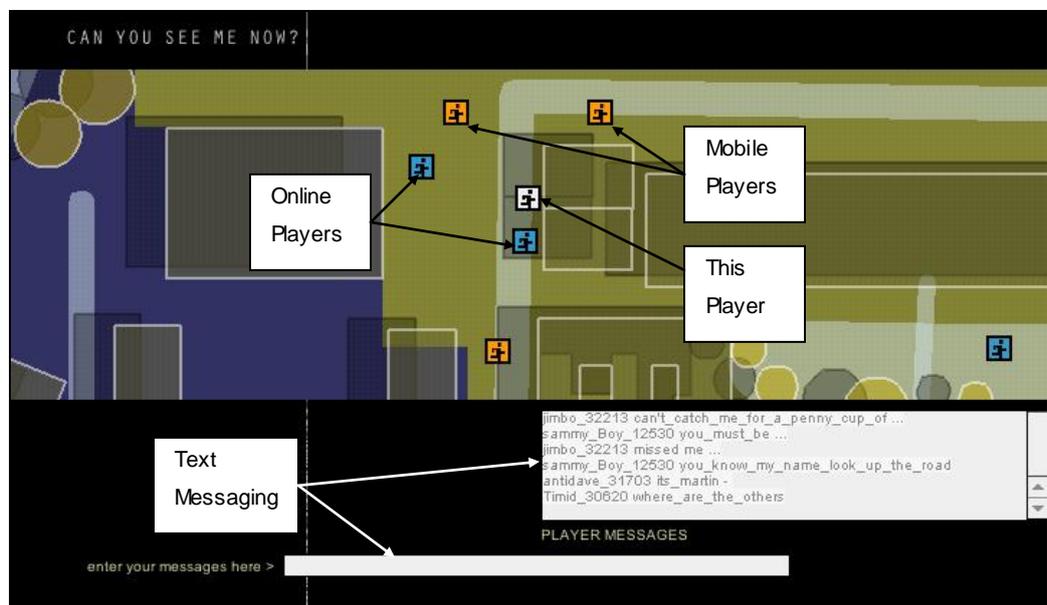


Figure 3-1: Early Can You See Me Now? online interface

Online players use the arrow keys to move their icon or avatar through the virtual city, at a speed equivalent to a slow jog within the real city. The features of the model prevent players from entering buildings, or crossing boundaries that are impassable in the real city.



Figure 3-2: Escaping from a mobile player

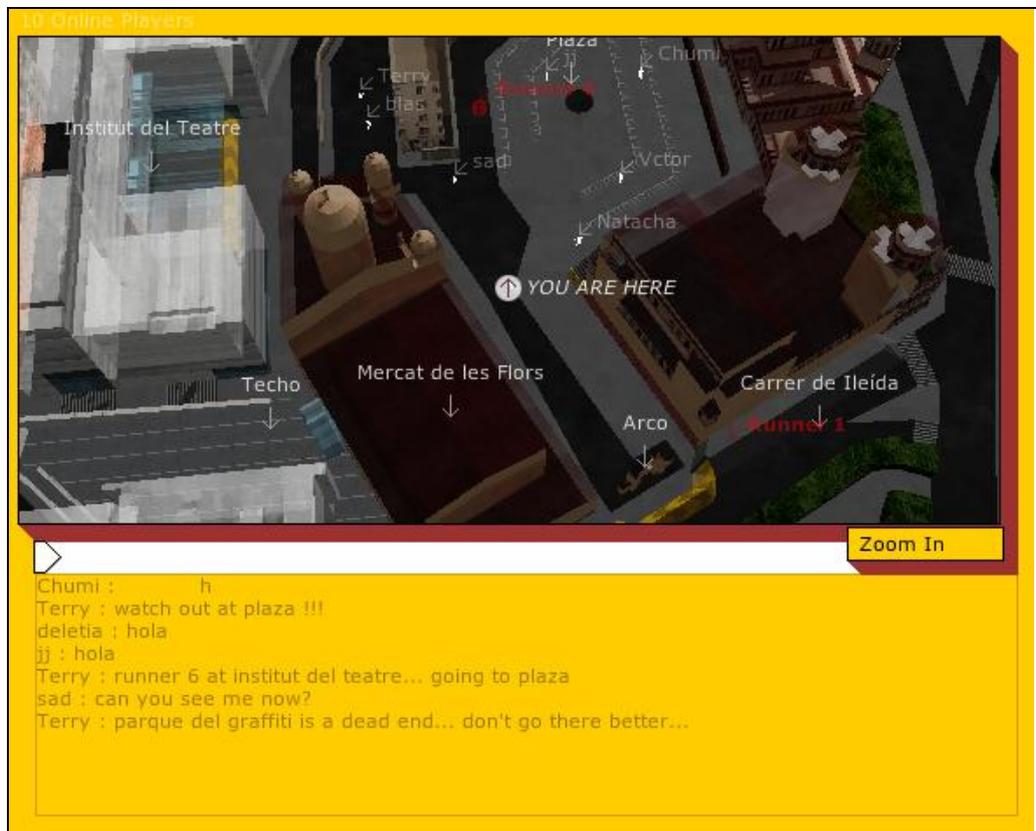


Figure 3-3: Zoomed-out view of the virtual city

Communication: The online interface provides a text input field, which allows players to send text messages to one another. All text messages are globally readable; all other players currently within the game, regardless of their location, receive whatever an online player sends, including mobile players. Online players hear a live audio stream from the mobile players in the city. Combined with the ability to send text messages, online players can engage in two-way communication with each other and with mobile players.

Online players exchange text messages to chat about tactics of how to avoid being seen by mobile players and where they currently are, while also taunting and goading the mobile players as they are heard chasing players across busy roads and up hills.



Figure 3-4: An online player is seen

Game play: When an online player enters the game, they are given a thirty second grace period in which to get their bearings and to familiarise themselves with the interface, during which time they are immune to being “seen” by mobile players. They are then chased down by mobile players, whose avatars

can be seen running towards the player as they run through the real city streets. When a mobile player gets to within five metres of the position of the online player, the player is ‘seen’ and the game is over. The camera in the virtual city zooms in and dramatically circles the online and mobile players, as shown in figure 3-4. They remain within the virtual city for twenty seconds, unable to move, while the mobile player announces over the live audio stream that they have seen the person that the online player named before joining, and have taken a photograph of them. The online player is then removed from the experience, freeing up a space for a new player waiting in the queue to join.

Online players may play as many times as they wish while the experience is running, however once they have been seen or have left the experience of their own accord, they must rejoin the queue to re-enter. A typical experience for an online player lasts from between a few minutes to over an hour.

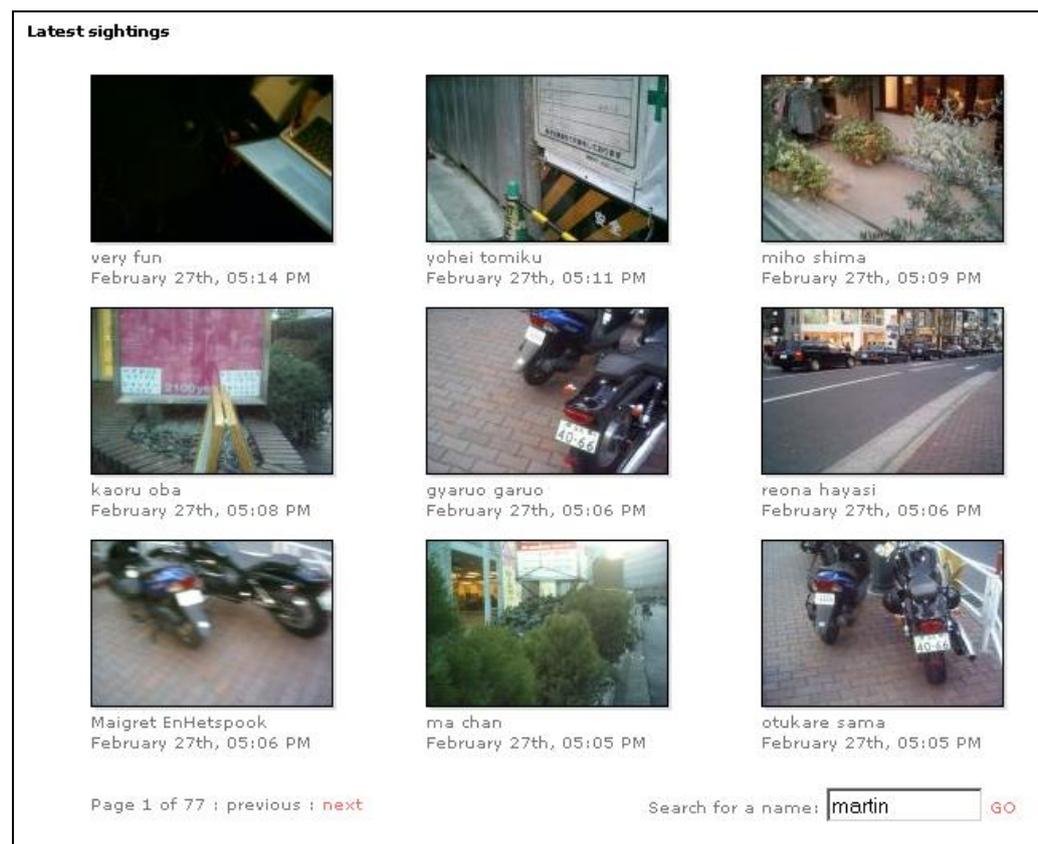


Figure 3-5: Sightings archive

Offline archive: After each game session, a photograph of each sighting of an online player made by the mobile players is uploaded to the experience’s web-

site, as shown in figure 3-5. Returning online players can search for the names of the people that they have not seen for a long time, and view photographs taken by the mobile players on the streets of the real city at the locations at which they were seen.

Finally, a scoreboard page on the web-site lists the three online players that have remained in the experience for the longest period of time without being seen. Some online players play repeatedly in order to try to obtain the top spot on the scoreboard.

3.2.2 Mobile Experience



Figure 3-6: A mobile player runs with equipment

Mobile players in *Can You See Me Now?* are professional performers, and inhabit the physical environment in order to catch online players. Each player carries a mobile device – a PDA which is connected using a wireless network, a walkie-talkie, a GPS receiver and a digital camera. The PDA is housed in a rugged, waterproof case, while the walkie-talkie and GPS receiver are mounted on the player's shoulder to aid reception, as shown in figure 3-6. The walkie-

talkie is fitted with a hands-free earpiece and microphone to allow ease of use on the move.

The GPS receiver reports the position of the mobile player once a second via the PDA and wireless network, placing the mobile player in the virtual city. The walkie-talkie is tuned to a specific channel that is shared by all of the mobile players, allowing them to communicate with one another. This channel is also streamed to the online players.

The PDA runs a software client that presents the mobile player with a live view of the virtual city, as shown in figure 3-7. Mirroring the online client, it shows a two-dimensional top-down schematic map view of the city and major roads, obstacles and features. Online players and their names are shown in red, while other mobile players are shown in blue. The three most recent text messages sent by online players are displayed at the bottom of the interface.

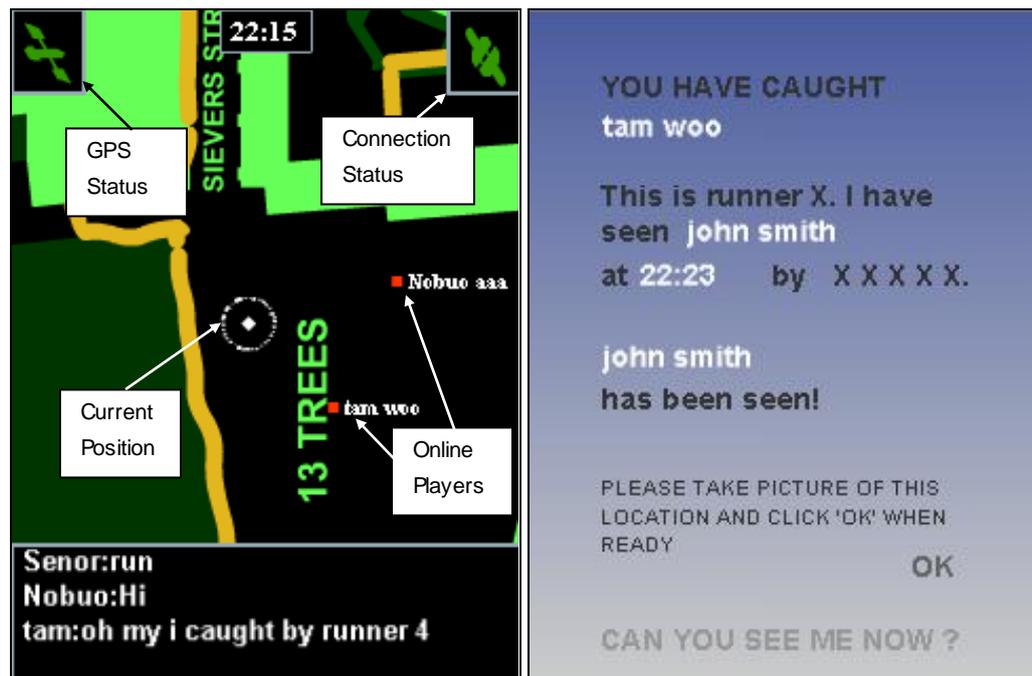


Figure 3-7: Mobile interface, left, an online Player is Seen, right

The map can be viewed in two modes, which the mobile player toggles between by tapping the screen. The zoomed-out mode shows the whole of the game area, allowing the mobile player to see where his or her colleagues are, and groups of online players if there are none in the immediate vicinity. The zoomed-in mode shows the immediate area that the player is in, with the map

continually updating with the mobile player's current reported position in the centre of the screen. This allows the mobile player to determine how to move in order to reach nearby online players.

The interface also indicates the reported accuracy of the GPS device, whether the device is currently connected to the wireless network and the rest of the experience, and how many online players are currently taking part in the experience.

When the mobile player's reported position is within five metres of that of an online player, the interface displays an alert message, as shown in figure 3-7. The message indicates that an online player has been caught, what their name is, what the name of the person that they have not seen for a long time is and the current time. It then instructs the mobile player to take a photograph of their immediate location with their digital camera. These photographs are usually taken of the floor, or any interesting landmarks, people or features in the surrounding area. Finally, the mobile player must tap the screen to clear the alert to be able to return to the experience, ensuring that they do not miss any sightings.

Mobile players perform for up to an hour before stopping for a break. There may be up to four or five performers on rotation, with three on the streets of the city at any one time. Part of the script that the performers adhere to is to continually describe their situation and environment over the walkie-talkie, to provide a rich experience for the online players. This includes describing frustrations with not being able to catch a particular player or the technology, responding to taunts or text messages, talking specifically to an online player, or simply describing the physical environment and the physical exertion of the performer. This is designed to enrich the online players' perception of the physical environment, heightening their engagement in the mixed-reality environment.

3.2.3 Touring History

Can You See Me Now? premiered in Sheffield in 2001 at the b.tv festival, before becoming a regularly touring performance, with each performance

involving over one thousand online plays. To date it has been shown in the following cities:

- Dutch Electronic Arts Festival, Rotterdam, 2003
- Edith Russ Site for Media Art, Oldenburg, 2003
- International Festival for Dance, Media and Performance, Köln, 2004
- Gardner Arts Centre, Brighton, 2004
- ArtFutura, Barcelona, 2004
- InterCommunication Centre, Tokyo, 2005
- The Junction, Cambridge, 2005
- Festival of Creative Technology, Cardiff, 2005
- Interactive Screen at the Banff Centre, Canada, 2006
- Museum of Contemporary Art, Chicago, 2006
- We Are Here 2.0, Dublin, 2007
- Donau Festival, Austria, 2007
- In Certain Places, Preston, 2007
- PICNIC, Amsterdam, 2007
- Machine-RAUM festival, Denmark, 2007

Can You See Me Now was winner of the Golden Nica for Interactive Art at the Prix Ars Electronica, Austria in 2003. It was also nominated for an Interactive Arts BAFTA award in 2002.

3.3 Uncle Roy All Around You

The second mobile mixed-reality experience in the series of artistic performances is Uncle Roy All Around You. It is significantly different from

Can You See Me Now? in that it places members of the public on the streets as mobile players, as well as online.

Mobile players undertake a journey through the streets of a real city, in search of an elusive character called Uncle Roy, while online players explore a parallel three-dimensional virtual city, again inhabiting a shared mixed-reality environment. Online players follow the progress of individual mobile players, communicating and collaborating with them, and choosing to help or hinder them. The main objective of the experience is to question a mobile player's trust in strangers and the anonymity of the digital world, whether they be online players, the mysterious figure of Uncle Roy or chance encounters with people on the streets of the real city.

Uncle Roy All Around You premiered at the Institute of Contemporary Arts, London in 2003. The experience was open to the public for ten days, for eight hours a day. During that time approximately 300 mobile players and 450 online players participated in the experience. Since then, the experience has been shown at The Cornerhouse, Manchester in May 2004, and The Public, West Bromwich in June 2004.

Uncle Roy All Around You was nominated for two BAFTA awards in the categories of Interactive Arts, and Technical and Social Innovation in 2004. It was also nominated for a Net Art Award at the Webby Awards, 2004.

3.3.1 Mobile Experience

Mobile players purchase a ticket for the experience, which lasts for a maximum of one hour, with up to twelve players participating at any one time. On arrival at the venue they hand over all of their personal possessions including bags, wallets, mobile phones and keys, in exchange for a mobile device – again a PDA, a ritual that is intended to increase their sense of anticipation, vulnerability, isolation and disconnection from the everyday experience of being in the city. It is also to increase their dependence on the voice of the game, Uncle Roy. Players have their photograph taken while a performer briefs them that their mission is to find Uncle Roy and explains how to use the

mobile device. They are then led out into the city and enter a nearby park, as shown in figure 3-8.



Figure 3-8: A mobile player explores the park

The mobile experience then consists of four main components; the clue trail, collaboration with online players, the office and the limousine.

Clue trail: The device displays a map of the area, with the mobile player's location indicated by an icon labelled "me". The first task is to find a red marker on this map, and to get to the physical location that it indicates. Once the player is there, they declare their position to Uncle Roy by using the stylus to drag the "me" icon to their current location on the map, and pressing the "I am here" button, as shown in figure 3-9.

Whenever a mobile player declares their position in this way, they receive a short text message from Uncle Roy, the voice of the experience, which responds with a text clue as to where to go next. In this way mobile players undertake a journey through the city, following a trail of clues that lead them through the park and onto the city streets in search of Uncle Roy's office.

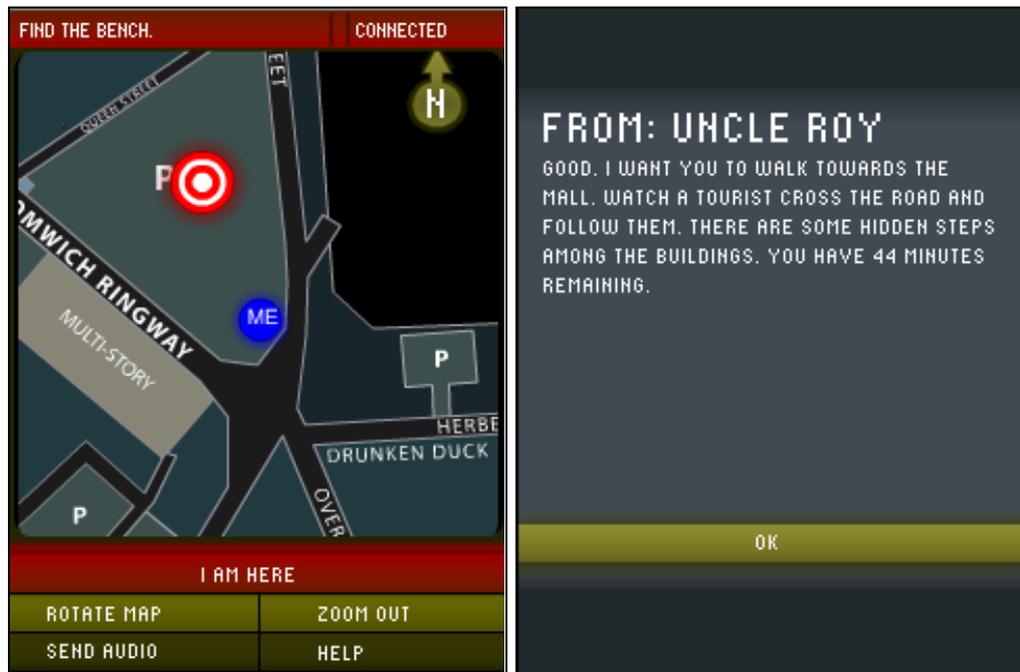


Figure 3-9: Map interface and red marker, left, a message from Uncle Roy, right

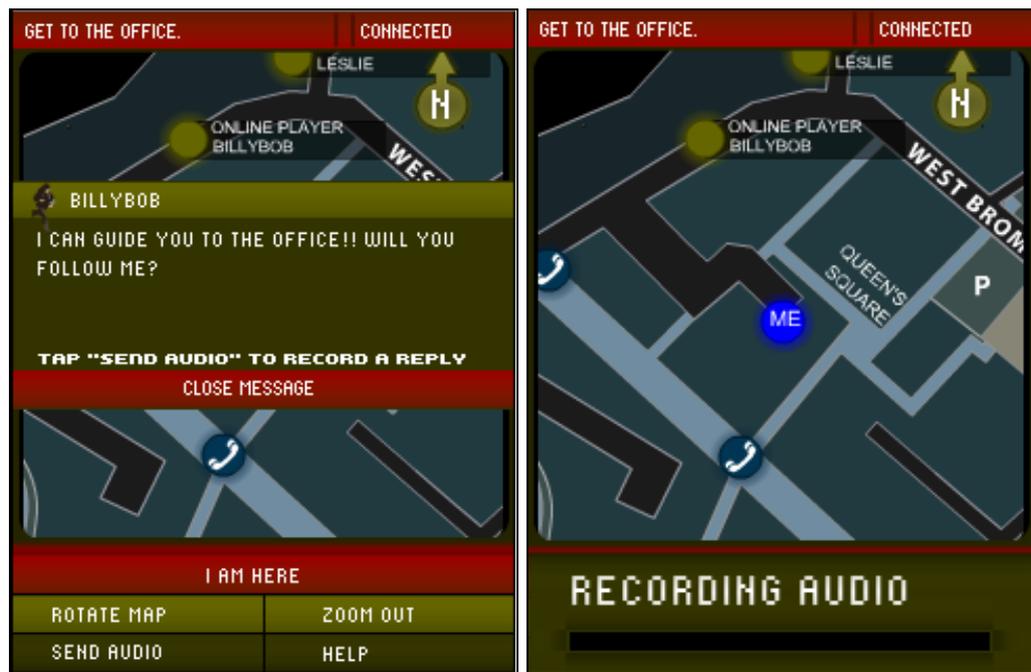


Figure 3-10: A message from an online player, left, recording a reply, right

A key feature of the experience is that the clues received from Uncle Roy are deliberately designed to be ambiguous. Some are direct and useful, while others are misleading to the point of being mischievous, encouraging mobile players to follow diversions, drawing on the history of the local environment, implicating passers-by in the experience, heightening the sense of being watched and also casting doubt on the intent and personality of Uncle Roy,

especially the extent to which he can be trusted. The majority of clues are location-specific, in that they make the most sense if the mobile player is in the location that they refer to. All of the clues remind the mobile player that they have a limited amount of time to find Uncle Roy and that this time is decreasing, as shown in figure 3-9. If a mobile player fails to find the office in the allotted time, then their device instructs them that their experience is over, and that they should return to the venue.

Online players: Once they have reached the red marker in the park, mobile players are able to see online players moving in the virtual city. They are indicated as icons on the map shown on the device. Online players can send text messages to a mobile player, which appear on the PDA as shown in figure 3-10. In return, mobile players can record and upload seven second audio messages for the online players to listen to. In this way a two-way communication channel between mobile players and online players is established, and the mobile player may attempt to enlist the help of an online player in finding Uncle Roy's office, if the online player knows where it is in the city.

The Office: Eventually the clues and text messages lead mobile players to Uncle Roy's office, which is a dressed set at the centre of the game area. At this point, the mobile device gives the mobile player a timed slideshow of instructions. They are asked to press a buzzer by the door to the office, and when this opens, are told to step into the deserted office and asked to look around. On a table in the office is a postcard with the question "when can you begin to trust a stranger?" printed on it. The mobile player is instructed to answer this question by writing on the postcard, then to sit in the nearby chair and to look up at a nearby camera and picture a stranger in their mind.

The limousine: After a few minutes the mobile player is asked to leave the building and, taking their completed postcard with them, to wait in the nearest phone box outside. After a few minutes the phone rings, and on answering it the mobile player hears a voice instructing them to walk around the corner and to get into a waiting limousine, shown in figure 3-11.



Figure 3-11: A mobile player approaches the limousine

A performer enters the limousine beside the player and the car pulls away to begin a circuit that will terminate close to the venue where the experience started. During the ride, the performer asks the mobile player a series of questions about trust in strangers, and tells them that somewhere else in the game another player is answering these same questions. The performer asks if the mobile player is willing to enter a year long contract to help this other player if ever called upon. If they agree, they are asked for their address and phone number. Finally, the car pulls up by a public post-box and the player is asked to post their postcard to seal the contract, or not, if they are unsure.

3.3.2 Online Experience

As in *Can You See Me Now?* online players participate in *Uncle Roy All Around You* through a three-dimensional virtual model of the same city area that mobile players are exploring. Online players visit a web site located at www.uncleroyallaroundyou.co.uk where they can find out more information and join the experience. Unlike mobile players, online players may play freely whenever they wish while the experience is running. As before, the online experience is limited to twenty players at any one time, and again if the virtual city is full then online players are held in a queue until a space becomes available.

The online experience consists of five main components; the virtual city, searching for the office, collaboration with mobile players, inside the office and an offline archive of commitments.

Virtual City: As in *Can You See Me Now?*, online players are dropped randomly into a three-dimensional model of the real city, through which they can navigate a walking avatar using their arrow keys. Again, they can chat with other online players by sending global text messages.

Searching for the Office: The first mission for online players is to find Uncle Roy's office in the virtual city. The environment is populated with photograph objects, which when approached display a photograph taken in the real city at the same location as the photograph object in the virtual city, as shown in figure 3-12. Players search these objects until they find one that is labelled as Uncle Roy's office, which reveals a photograph of the door to the office. Once found, online players are instructed that they should remember this location, as shown in figure 3-13, as this information may be useful.



Figure 3-12: Searching photographs of the city

Collaboration with Mobile Players: The online interface reveals a set of cards that give details of the mobile players currently in the experience, including their name, gender and the photograph that was taken at the start of

the experience. By selecting a card an online player can send private text messages to an individual mobile player, and listen to their most recently uploaded audio message, enabling two-way communication.



Figure 3-13: Finding Uncle Roy's office online



Figure 3-14: Directing a mobile player

As a mobile player moves the “me” icon on the mobile device, this estimated position is revealed as a pulsing red sphere in the virtual city, to indicate to online players where the mobile player is. Whenever a mobile player uses the

“I am here” button, this red sphere representation is dramatically enhanced – a large red sphere becomes visible and gradually shrinks down to the new position, and a sound is played. In later deployments of the experience, this was replaced by a walking black avatar illuminated in a red column of light, as shown in figure 3-14.

Using their knowledge of where the office is, and by following mobile players through the virtual city, online players can assist a mobile player in finding the door in the real city. However, they may also choose to deliberately mislead the mobile player, or the directions that they give may contradict the route that Uncle Roy’s clues on the mobile device are indicating.

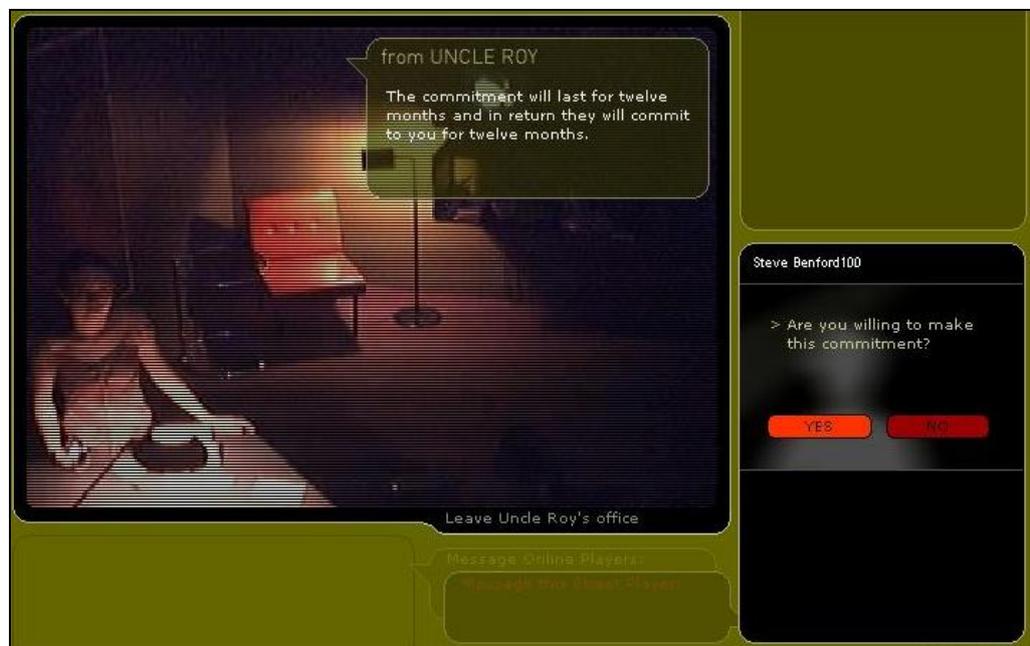


Figure 3-15: Making a commitment in the office

Inside the Office: When a mobile player successfully reaches Uncle Roy’s office, any online players that have located the office in the virtual city, and are in the immediate vicinity of the door are invited to join them. At this point the virtual city is hidden, and replaced by a live web-camera feed of the inside of the real office, enabling the online player to see the mobile player they have been assisting for the first time, as shown in figure 3-15. The online player is asked the same questions that the mobile player will be asked in the limousine via a series of dialogue boxes in the client. This includes the question of whether the player is willing to commit to help a stranger for the next year, in

which case they enter their personal contact details. After they have done this, the online player is given the choice of leaving the game, or returning to the virtual city to help another mobile player

Offline Archive: Having completed the experience, mobile players and online players who made the commitment to help one another are paired up and sent each other's contact details. The postcards filled in by mobile players in the office are made available on the experience's web-site, for both mobile players and online players to browse, as shown in figure 3-16.

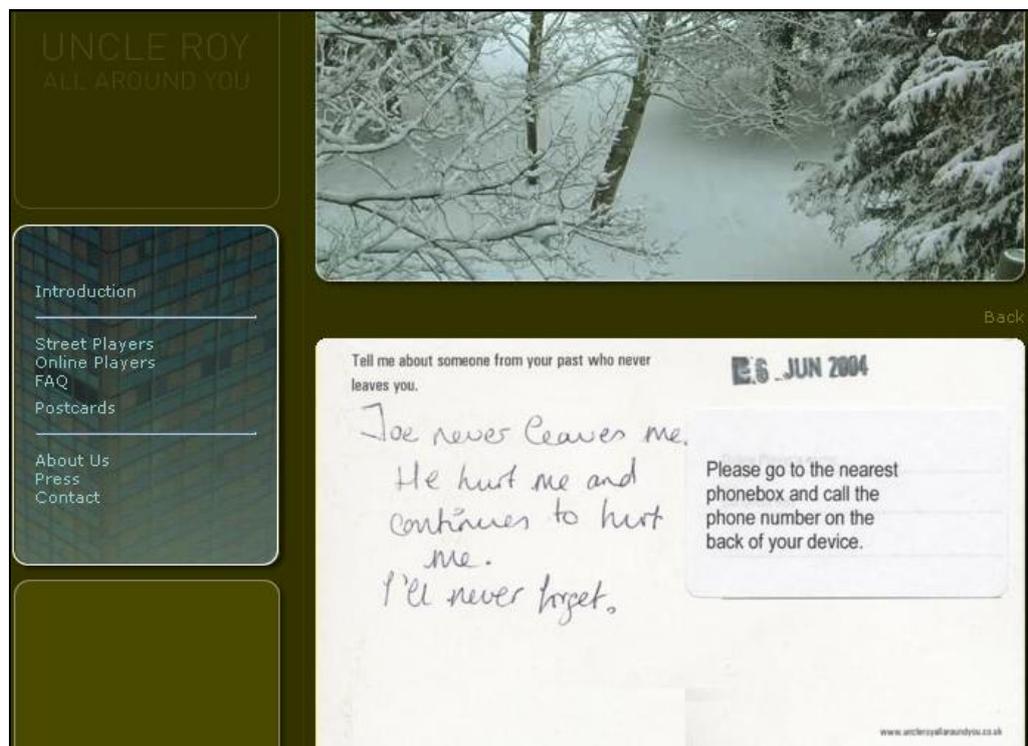


Figure 3-16: Viewing a completed postcard online

3.4 I Like Frank in Adelaide

The third and final mobile mixed-reality experience in the series of artistic performances is I Like Frank in Adelaide. Again, it places members of the public on the streets as mobile players and online. Drawing parallels with Uncle Roy All Around You, players are searching for a fictional character called Frank. The main objective of the experience is to engender interaction between the players, and artistically it draws upon themes of loss and absence, as Frank is never found; he is permanently missing.

I Like Frank in Adelaide was developed for and premiered at the Adelaide Fringe Festival, in March 2004, as part of the Adelaide Thinkers in Residence Programme. It was shown for ten days, during which the experience was available for eight hours a day. Approximately 250 mobile players and 1000 online players took part in the experience.

3.4.1 Mobile Experience

Up to fifteen mobile players can participate in I Like Frank at any one time. As with Uncle Roy All Around You, mobile players purchase a ticket for the experience, which lasts for up to an hour. On arrival at the venue they again hand over all of their personal possessions, but this time in exchange for a 3G mobile phone. A performer briefs the mobile player that their goal is to find Frank, and explains how to use the phone. They are then led out of the building, and the performer indicates the correct direction to head off in.

The mobile experience then consists of three main components; the clue-trail, finding postcards for online players and a final performance.



Figure 3-17: Mobile phone interface, left, 3G phone, right

Clue-trail: As in Uncle Roy All Around You, the mobile phone displays a map of the surrounding area, with cross-hairs showing the mobile player's last

reported position. As the player walks, they are instructed to use the phone's joystick to move the cross-hairs to represent their movements, and on releasing the joystick the map repositions itself to indicate their new position with the player once again at the centre, as shown in figure 3-17. A menu allows the player to zoom in or out of the map to see a wider area, and to declare their position to the system with an "I am here" button.

The first task for the mobile player is to find a red marker on the map, and to get to the physical location that it indicates. The red marker is on the edge of a university campus, leading into the centre of the city. As the player walks through the campus, they receive timed, introductory text messages from the voice of the experience, which is not identified, describing a relationship with Frank and how they need help from the mobile player in finding him:

"Welcome to Adelaide. I'm happy that you could make it. I want you to help me find Frank. He was here at one time, with me."

Having reached the red marker, the mobile player receives text clues as they declare their position, indicating where to go next. As in Uncle Roy All Around You, the mobile player undertakes a journey following a trail of location-based clues through the back streets of the city. Each time the player declares their position, they receive a new clue. Again, the clues are designed to be ambiguous, and open to a range of interpretations, describing the memory of a similar journey undertaken with Frank:

"Frank never took education seriously. Get off the campus."

"We toured the Grand Lodge, trying the handles of locked rooms."

Finding postcards for online players: Once the mobile player reaches the red marker, online players are revealed in the city as icons on the map, which move as the online players explore the virtual city, and the mobile player is revealed in the virtual city. Once within one hundred metres of one another, a mobile player and an online player can communicate using text messages and short audio recordings. When a text message is received from an online player, it appears as an alert on the mobile phone. The mobile player can ignore it or

reply with an audio message, which causes the phone to make a call to a voice mail message, allowing the player to leave a fifteen second recording before hanging up and returning to the map.



Figure 3-18: A performer, right, makes a video call to a mobile player, left

Online players approach the mobile player to enlist his or her help in obtaining a physical postcard that is hidden somewhere in the real city. The mobile player can choose to help in this task, but this may appear to be at odds with their task to find Frank as the clues may lead in a different direction, and they only have a limited amount of time. If the mobile player agrees to help an online player, the online player directs them to where they think a postcard is hidden – one of four locations; behind the bar in a pub, in a pool hall, in the saddlebags of two bikes chained at the end of an alleyway and outside a post office. Having found a postcard, the online player gives the mobile player their postal address, which the mobile player then writes on the postcard. The mobile player is now free to continue the search for Frank, carrying the postcard.

Final performance: The clue-trail eventually leads the mobile player south through the city, on a route that avoids the main thoroughfare and is peppered with memories of Frank. The clues lead toward a particular area of town; a secluded office complex called “Futurelands”. If the player does not reach Futurelands in the allotted time then an alert informs them that the experience is over, and that they should return to the venue. However, if the mobile player does reach Futurelands in time, they receive a video call on the mobile phone as they approach. Answering the phone the player can see themselves on the screen, filmed by somebody they cannot see, hidden on the other side of the road in Futurelands, as shown in figure 3-18. This video call is made by a performer, who instructs the mobile player to enter Futurelands, showing them where to go by pointing the camera on the calling mobile phone towards the entrance.

Still using the video call, the mobile player is instructed to sit down on a bench in a leafy, paved courtyard. Once there, the performer asks them to answer the question written on their postcard, or, if they did not collect a postcard during their journey, the question is asked personally. The question is one of the following:

“Is there someone who you saw briefly who you’ve never forgotten?”

“Who do you think of when you are alone?”

Having completed this interaction, the performer reads a script describing his last meeting with Frank, informing the mobile player that the performer is not Frank, but in fact the voice of the experience that has been sending clues. They are then asked to leave their postcard on the bench, and to return to the venue to return their mobile phone, as the experience has been completed.

3.4.2 Online Experience

Online players play in a virtual model of the city of Adelaide. As with previous experiences, this is a three-dimensional model of the city accessed via a webpage, this time at **www.ilikefrank.com**. Again, entry is controlled using a queuing system, and the number of online players that can play concurrently is restricted to around twenty.

The online experience consists of three main components; searching the virtual city, collaborating with mobile players to find a postcard and a virtual Futurelands sequence.

Searching the city: As in Can You See Me Now and Uncle Roy All Around You, online players navigate a walking avatar through the virtual city using their arrow keys. As before, online players can communicate with one another using text messages. Upon joining the experience, online players receive a timed set of text messages, analogous to the messages received by the mobile player at the beginning of their experience. These messages again explain that the online player should assist in the search for Frank; although this time the messages indicate that the online player should find a photograph in the virtual city that marks an important place to Frank, and to find a mobile player to help them find something at this location in the real city.



Figure 3-19: Searching for a postcard online

As in Uncle Roy All Around You, the virtual city is populated with photo objects, which the online player must search through as they explore the space. Entering one of these objects triggers the display of a photograph from the real city streets, taken at the same location as the object in the virtual city, as shown in figure 3-19. The online player must search through these photographs

looking for the photograph of the important location, which is randomly selected from the four pre-defined postcard locations. When this has been found, the online player is informed that this photograph shows a location in which a physical postcard is hidden in. Next, the player is instructed to find a mobile player to retrieve the postcard.



Figure 3-20: Recruiting a mobile player to help find a postcard



Figure 3-21: Guiding a mobile player to a postcard

Collaborating with mobile players: As in Uncle Roy All Around You, mobile players are revealed in the virtual city as walking avatars, which move when the mobile player updates their current position using the mobile phone interface. As before, cards displayed on the online interface provide information about mobile players currently taking part in the experience, and allow the online player to select one to communicate with. As mentioned previously, the online player has to be within sight of a mobile player to be able to send them a text message.

Having found their postcard in the virtual city, online players attempt to collaborate with a mobile player to find its physical equivalent in the real city. They do this by leading the mobile player to the correct location through their movements, which are mirrored on the mobile player's map, and by giving directions with text messages, as shown in figures 3-20 and 3-21. At the correct location, the online player describes the scene shown in the photograph, enabling the mobile player to retrieve the postcard. Once found the online player gives the mobile player their address to write on the postcard, which they will later receive in the post, complete with the mobile player's answer to the question written on it.

Virtual Futurelands: Having found a postcard, online players are free to attempt to find another postcard in a different location, or to help the mobile player that they have been collaborating with in their search for Frank, by offering advice and directions gleaned from the other photographs embedded in the virtual city.

When the mobile player reaches the end of their experience, their avatar leaves the virtual city, and the following online player is given the opportunity to enter a virtual equivalent of Futurelands, as shown in figure 3-22. Here they listen to a pre-recorded audio message that mirrors the text that the mobile player is hearing live from the performer, before leaving the experience.



Figure 3-22: Finding Futurelands online

3.5 Savannah

The final mobile mixed-reality experience is Savannah, an educational game for school children. The broad objective of the experience is to enable players to learn about lion behaviour by appreciating the various requirements and priorities that lions have for their day-to-day survival on a real savannah, including decisions regarding territory, shelter, food and water, hazards and changing seasonal conditions.

As with the previous experiences, mobile players inhabit a mixed-reality environment, exploring a virtual savannah area which is mapped to a real school playing field. However, Savannah explores the concept of an online player as a performer, analogous to *Can You See Me Now?*, in which play was directed by performers as mobile players. In Savannah, there is one online player, a school teacher who leads the mobile players through the experience from the classroom, with two specific missions interspersed with reflection periods in which players return to the classroom to review their earlier actions.

Savannah was developed in collaboration with Future Lab, the BBC Natural History Unit and Mobile Bristol. It was presented to a selected group of school children in Bristol, 2003.

3.5.1 Mobile Experience

Groups of six school children take part in Savannah at a time as mobile players. The experience takes place on an open rectangular area of approximately 100 by 400 metres in size, with no physical obstacles or markings. The boundaries of the game area, the four corners, are indicated with flags.

The mobile experience consists of three main components; inhabiting the savannah, an exploratory mission and a survival mission.



Figure 3-23: Savannah PDA interface

Inhabiting the savannah: Each child is equipped with a mobile device – a PDA that is connected to a wireless network, headphones to listen to audio media and a GPS receiver to track their movements within the game area. As players move within the physical environment, they also move within a virtual

savannah environment that replaces the virtual city from the previous experiences.

The virtual savannah is divided up into different spatial regions that reflect the different kinds of environment that may be found in a real savannah; grass, marsh, rocks or a river, together with hazards, and animals that could be prey for lions. As mobile players explore these regions, location-based content is displayed on their devices, representing a lion's sense of sound, sight and smell. Sounds are played through the headphones to represent the sense of sound, and images are displayed to represent sights and smells, as shown in figure 3-23. The device also displays the health of the lion, text messages received from the online performer, and a function key that allows the lion to scent mark or perform an attack.

Exploratory mission: The first mission is one of exploration and territory marking. The mobile players begin with no knowledge of the layout of the virtual savannah, and have ten minutes as a group to explore as much of the game area as possible, moving across the field and receiving content accordingly. During this mission, players are able to 'scent mark' each new area of content they encounter using the function key on the device.

Hunting mission: The second mission focuses on hunting and survival, with the mobile players collaborating to act as a pack of lions rather than as individuals. As they pass through the areas of content discovered in the previous mission, they encounter new content to indicate that different animals are now present, including a carcass, a young wildebeest, an old buffalo and an elephant. The function key on the device now allows the mobile player to attempt to 'attack' the animal that they have encountered.

An attack on these animals may or may not be successful. Carcasses or small animals only require one mobile player to eat, whereas larger animals require several of the players to attack at the same time to kill, and some animals or hazards may not be affected by any number of attacking players. After each attack, the mobile players involved receive a text message informing them of the result of the attack, and whether they have been successful.

The goal of this mission is to keep the lion's health level as high as possible, as indicated by the device. A successful attack results in the lion being able to feed, and their health is increased accordingly. An unsuccessful attack, or venturing into an area that contains a hazard, results in the lion losing health. In this way, mobile players are encouraged to collaborate and plan their attacks to make sure that they are successful, as well as thinking about their environment.

3.5.2 Classroom Experience

In Savannah, a common online interface functions as both a tool for the teacher to directly manipulate the mobile players' experience, and as an interface for reflection by the mobile players when they return to the classroom after each mission.

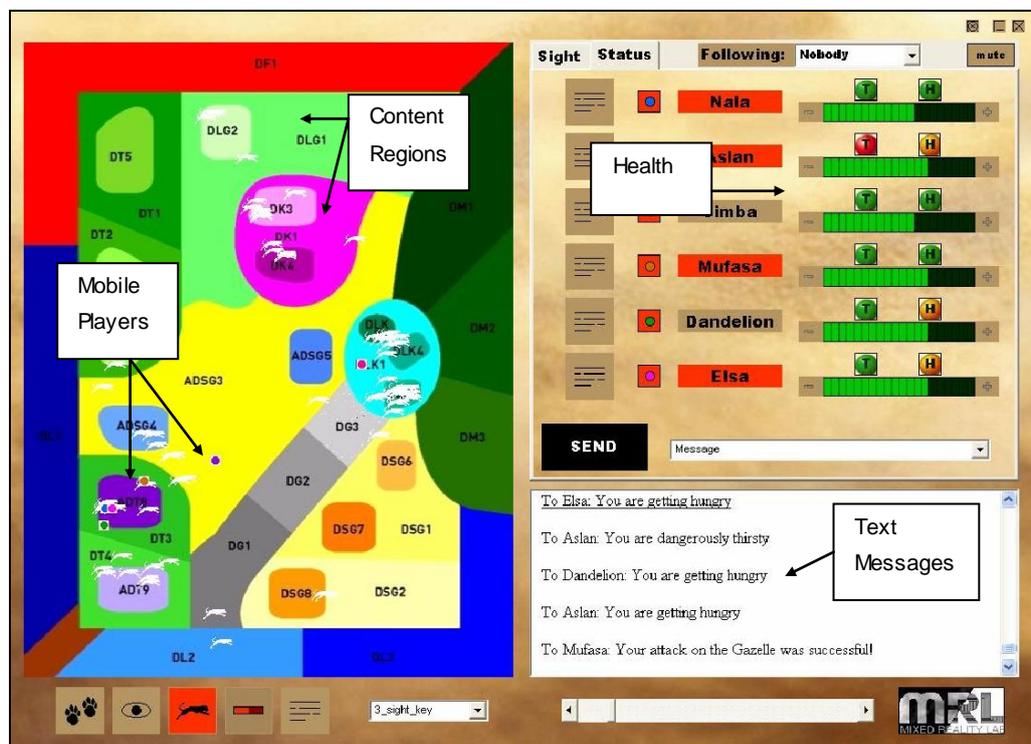


Figure 3-24: Manipulation using the Den interface

Online manipulation: while the mobile players are out on the field, the teacher guides the players through the experience. Unlike previous experiences, this one online player is not embodied in the virtual savannah; instead they observe the virtual environment using an interface known as the Den interface, which is displayed on a large interactive white-board. As shown in figure 3-24 the Den interface reveals the current location of the mobile

players in the virtual savannah, gives a history of text messages players have received and indicates the current health of each player.

The Den interface indicates when a mobile player has used the function key on their device, and provides a list of possible messages with which to respond to one, several or all of the mobile players. The online player can also construct a new message, which is then added to the list of available messages for future re-use. Finally, the interface allows the online player to manually adjust the health of a mobile player, in response to their actions.



Figure 3-25: Reflection using the Den interface

The online player uses the Den interface to construct scenarios for the mobile players out on the field, and to shape their experiences. This includes reducing the health of the lions and sending text messages to indicate that the lions are thirsty and need to find an area with water, rewarding the lions for successfully attacking a feasible prey with an increase in health, or punishing them for straying into the hazardous areas at the edge of the savannah with a reduction in health and an appropriate text message.

Classroom Reflection: After each mission, the mobile players return from the playing field for a period of reflection on their actions, led by the teacher. This is again facilitated by the Den interface, which now reveals the players’ actions in the previous mission, as shown in figure 3-25.

The Den interface shows historical trails of how the mobile players moved through the virtual savannah, indicating which content areas were visited. It also indicates key events; scent marking and attacking, and the mobile players are encouraged to rationalise their actions and how they believed they related to actual lion behaviour – for example why attacking a young wildebeest was realistic and therefore successful, and why attacking an elephant was unrealistic and consequently failed.

3.6 Summary of Requirements

Table 3-1 gives a summary of the key requirements of the four experiences described in this chapter, according to the fifteen categories presented in chapter 2.

	Mobile Players	Online Players	Mixed- Reality	Multiplayer	Connected	3D Overlay	Mediascape	2D Map	Location-Based	Terrain-Based	Relative-Content	User Geo-Annotation	Publicly Deployed	Managed	Free Participation
CYSMN?	X	X	X	X	X			X	X	X	X		X	X	
Uncle Roy	X	X	X	X	X			X	X	X			X	X	
I Like Frank	X	X	X	X	X			X	X	X			X	X	
Savannah	X	X	X	X	X		X	X	X	X	X			X	

Table 3-1 Summary of the four experiences and their requirements

3.7 Conclusions

This chapter has introduced the four mobile mixed-reality experiences that have been supported by the research presented in this thesis; Can You See Me Now?, Uncle Roy All Around You, I Like Frank in Adelaide and Savannah. For each experience, this chapter has described the end-user experience for both mobile and online players, and concluded with a summary of the

functional requirements of the experiences for comparison with those described in chapter 2.

At this stage it is apparent that supporting the four experiences raises significant challenges; both in terms of authoring and serving the location-based content required for each experience, and orchestrating the experiences in light of the emergent behaviour that can arise from engendering collaboration and interaction between the two groups of players, and their use of mobile technology out in the wild.

The next chapter continues with this theme in order to further consider how to best support mobile mixed-reality experiences in general. Later chapters will describe how supporting of these four experiences raises particular challenges that require the deployment of a number of dedicated tools and solutions.

4 A Framework for Supporting Mobile Mixed-Reality Experiences

4.1 Introduction

This chapter defines a framework for supporting mobile mixed-reality experiences. Previous chapters have reviewed a representative sample of mobile mixed-reality experiences, including four that have been developed by the author, and this chapter now uses this to draw out common requirements for supporting these experiences.

This chapter begins by classifying the reviewed mobile mixed-reality experiences by genre, showing how each can be classified along a dimension of participation that mixes performance and game play on the streets of a city, and contrasts them with more conventional forms of performance and play.

Next, this chapter defines an orthogonal, second dimension of supporting tasks – the activities required to support such experiences – that considers how experiences are authored before they begin, and how they are improvised while they are running. These two dimensions of participation and supporting tasks define a framework that highlights how mobile mixed-reality experiences draw in elements of street theatre, conventional theatre, computer games and role-playing, and in turn show how this suggests that experiences require particular support from new authoring and orchestration tools.

The next two chapters will examine these new challenges in detail, describing how supporting the deployment of the four experiences described in chapter 3 has led to the development of significant new tools and solutions to support authoring and orchestration. Chapter 7 describes the implementation of the four experiences, showing practically how support for authoring and orchestration fits into the wider system.

4.2 Forms of Participation

(Montola, 2005) considers how pervasive games attempt to break out of the traditional magic circle that specifies that games are played within a specified

space and time and a strict social setting (Salen and Zimmerman, 2003). He argues that pervasive games break these boundaries by being spatially expansive – where a game is played in a city where the player may be unaware of which locations constitute the game area, temporally expansive – where a game is played episodically or as part of everyday life, or socially expansive – where bystanders and spectators may make a difference to or be unknowingly involved in game play.

Genre	Experiences
Classic Games	ARQuake, Botfighters, Catch Bob!, CitiTag, Human Pac-Man, Mogi, Pac-Manhattan, Pirates, Real Tournament, Savannah
Big Urban Games	Conqwest, Go Game, Navigate the Streets, GeoCaching
Urban Explorations	GeoTracing, GUIDE, MEE, Urban Tapestries, Riot! 1831
Seamful Games	Treasure, Yoshi
Artistic Performances	Can You See Me Now? , Desert Rain, I Like Frank , Uncle Roy All Around You

Table 4-1: Defining experiences by genre

Mobile mixed-reality experiences fit into this classification by expanding onto the streets of the city in a number of ways, and creating new forms of participation. As demonstrated by *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank*, artists are exploring games as performance, which mix elements of mobile game-play with conventional performance and new-media art. Experiences also combine game-play with live role-playing to support free-play (Mandryk and Inkpen, 2001), giving players a wide agency to define their own actions on the streets, and blurring the boundaries between a fictional game and real life. By moving onto the streets of the city, the very nature of participation is changed, as players are thrust into a public setting, unwittingly performing for any spectators or bystanders, and potentially

crossing the boundaries of normal behaviour in a public setting (Benford et al., 2006).

For the purposes of this thesis, we consider mobile mixed-reality experiences can be grouped according to a few broad genres, as shown in table 4-1:

- **Classic Games:** These are experiences that recreate the game-play of an established computer game on the streets, as seen, for example, in ARQuake and Human PacMan, or traditional playground games, as seen in CitiTag and CatchBob!. These experiences involve mapping a virtual game area to an area of the physical environment, which is then inhabited simultaneously by multiple players, mediated by way of a two-dimensional map or by a three-dimensional augmented reality overlay. Participation in these experiences is highly influenced by the game-play of the original game and by its adaptation to a physical environment, typically relating movement in the physical environment to movement in the original game, or mixing role-play with technologically mediated game mechanics.
- **Big Urban Games:** These experiences are cross-media games that focus on player improvisation and performance in order to provoke interesting game-play which is orchestrated and judged by the organisers. Participation in Big Urban Games is often highly socially expansive and involves elements of performance, often requiring players to interact with bystanders to complete a task, as seen in the Go Game, or involving highly visible activity in the city, as seen in Conqwest with players dragging large inflatable totems through the city.
- **Urban Explorations:** These experiences focus on providing a single-player experience for a mobile player moving through a city. These experiences are largely driven by position-based game mechanics in that they are based around the exploration of pre-authored digital content based on the player's position in the physical environment. Participation in these experiences focuses on a player interacting with

static content rather than with other players, as seen in the walking tour provided by the GUIDE system, or the media-rich environment of Urban Tapestries.

- **Seamful Games:** These experiences involve game-play that directly exploits the seams in mobile infrastructures, notably wireless networking in experiences such as Treasure and Yoshi. Seamful games are more spatially expansive than classic games as, while the game areas may be selected for having a rich and varied infrastructure landscape, they can potentially be played anywhere, with natural variations in the infrastructure providing the main dynamic of the game as players move through the city.
- **Artistic Performances:** These experiences are primarily artistically driven, primarily taking the form of events and performances in order to be accessible to members of the public as engaging works of art. These experiences combine many aspects of the other genres. In Uncle Roy All Around You and I Like Frank, players explore the city as in urban explorations, but this is also combined with elements of game-play such as searching, collaborating or chasing, as seen in Can You See Me Now?, and performance elements in the form of professional performers on the streets, interaction with bystanders and static sets such as Uncle Roy's office, or those seen in Desert Rain. As a result, participation in these experiences is a hybrid mixture of pre-scripted game-play, and live, improvised performance.

Figure 4-1 shows how these genres and the experiences described in chapter 3 can be located along the dimension of 'forms of participation', and places them alongside more traditional forms for comparison. This dimension ranges from 'played' – experiences that are highly driven by game play, to 'performed' – experiences that fundamentally revolve around performance.

Moving from 'played' to 'performed', conventional computer games are placed at the played extreme of the dimension, as participation is fundamentally based on game-play, usually in a private setting. They do not involve performance,

and game-play is usually strictly limited to a prescribed and supported set of actions. Next, role-playing games are placed in the primarily played region of the participation dimension, but begin to introduce elements of performance; they primarily involve interacting with game-mechanics, but also involve players performing to one another to enhance a particular scenario, with some role-playing events spilling out onto the streets of the city, as seen in (Jonsson et al., 2006). Street theatre is placed in the primarily performed region of the participation dimension, as it is performed in a setting that is deliberately public, involving bystanders and passers-by, and dynamically negotiating participation with an ad hoc audience. Finally, conventional theatre is placed at the performed extreme of the participation dimension, as it rarely involves game-play for participants, and generally taking place in a highly prescribed, pre-scripted and deliberately and carefully framed manner, analogous to the well defined game-play of conventional computer games.

Figure 4-1 shows how the reviewed genres and the four experiences similarly occupy the dimension of forms of participation. Classic games, seamless games and urban exploration experiences are all placed in the played region of the participation dimension, as they are based upon strong game-play elements and interactions. Conversely, big urban games are placed towards the performed region of the dimension, as they are largely based upon the spectacle of performing unusual or unconventional acts in a public setting.

Artistic performances, particularly *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank*, have spanned the breadth of this dimension, as participation encompasses strong elements of both game-play and rich performance. For example, *Can You See Me Now?* is both played, mirroring the chasing game-play of a conventional computer game, and performed – professional actors drive the experience using their movements and vocal performance over the live audio stream. Similarly, *Uncle Roy All Around You* and *I Like Frank* are both played; requiring interaction and cooperation between players to search for and achieve common goals, and stretch further into the performed end of this dimension; professional actors perform rich, carefully framed interactions with individual players alongside dedicated sets and props, such as the office and the limousine.

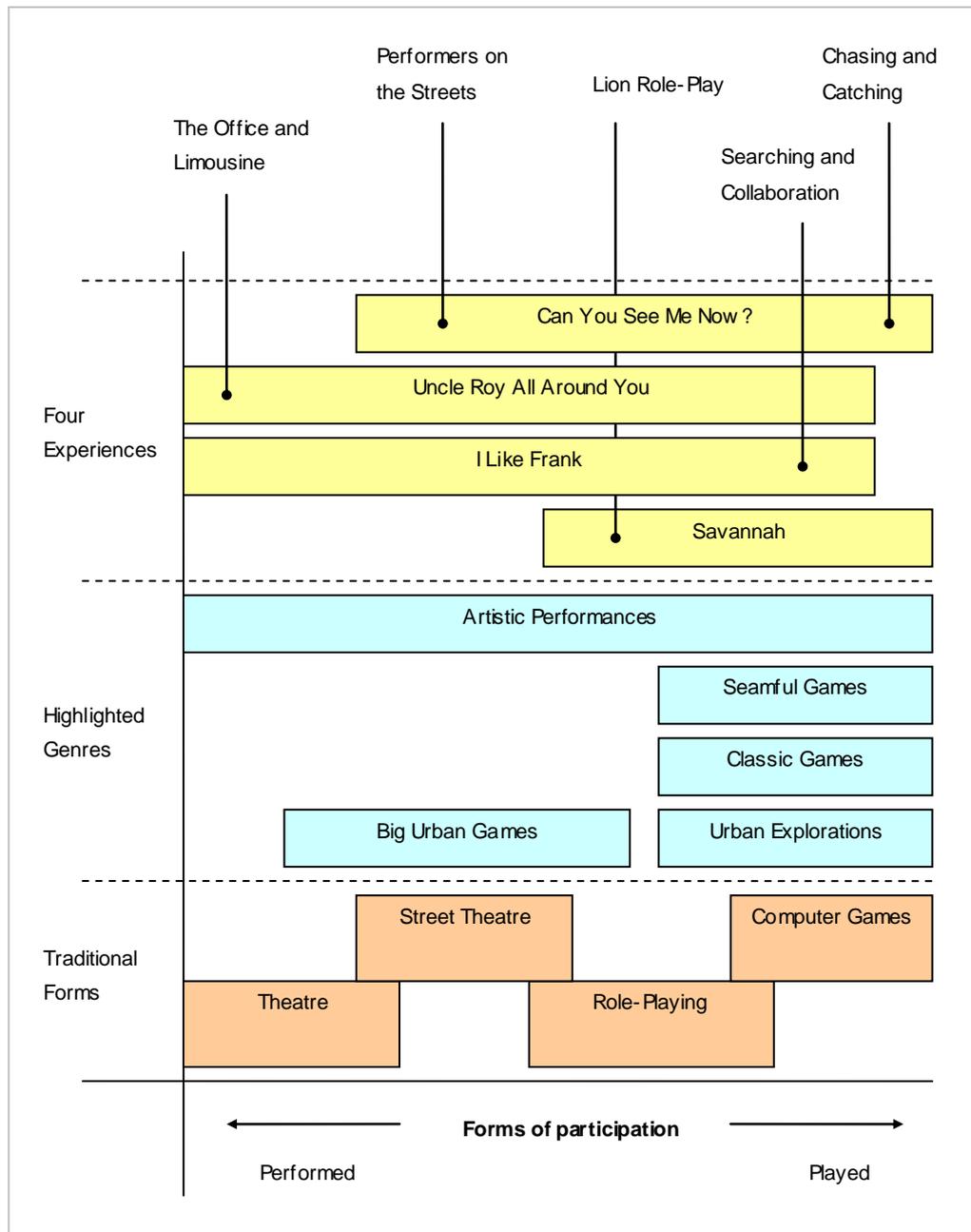


Figure 4-1: Highlighting forms of participation

Finally, in the centre of the participation dimension, artistic performances often use the ambiguous and volatile mixture of play and performance on the streets as a game dynamic in itself; for example by falsely implicating bystanders as performers or by having performers apparently appear out of nowhere. This has players questioning the exact nature of their participation in a performance, in turn creating heightened game-play for players by pushing the boundaries of their own behaviour in public – taking postcards, approaching strangers and

entering buildings they had no reason to be in, in the case of Uncle Roy All Around You.

4.3 Supporting Tasks

The previous section suggests that participation in mobile mixed-reality experiences fundamentally comprises of a variety of forms of play and performance, with artistic performances such as Can You See Me Now?, Uncle Roy All Around You and I Like Frank broadly spanning this dimension. However, in order to now consider how to support these various forms of participation our framework must now be extended to include a second dimension of tasks that must occur *prior* to a mobile mixed-reality experience taking place and those that must occur *during* a mobile mixed-reality experience.

4.3.1 Authoring and Improvisation Tasks

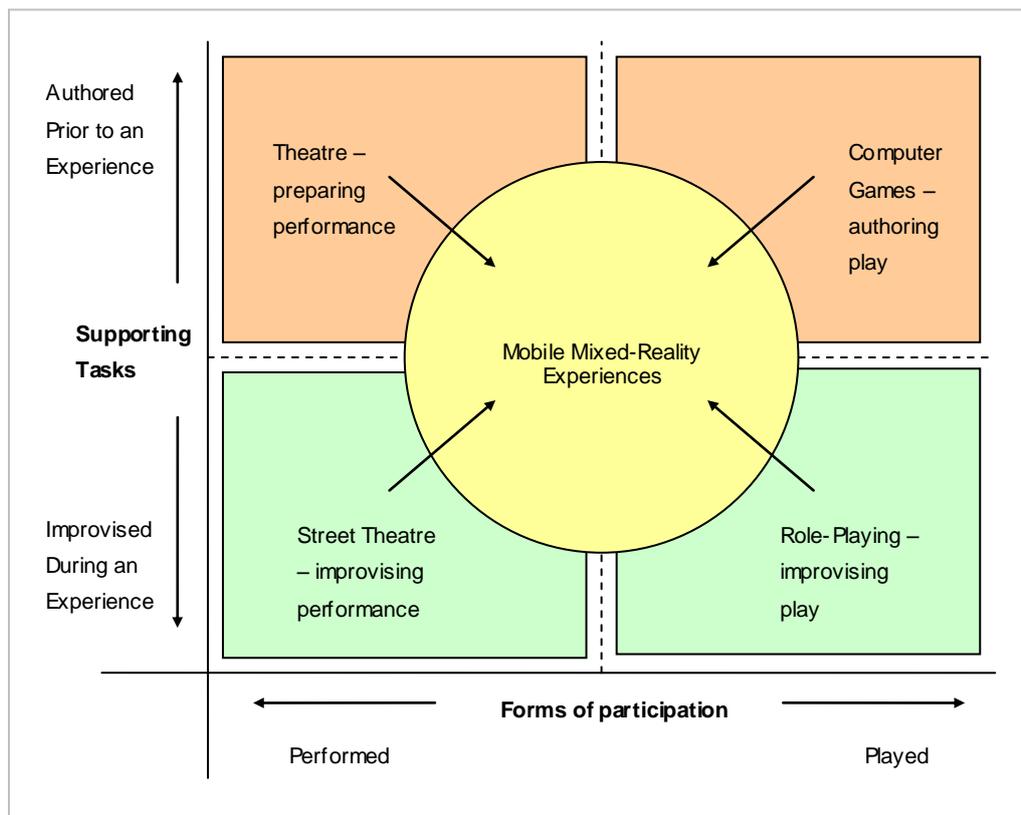


Figure 4-2: Supporting tasks

The orthogonal second dimension of the framework defines the tasks involved in creating and operating a mobile mixed-reality experience. This dimension

ranges from tasks that involve ‘authoring prior to an experience’ – which involves authoring content and framing the various components before it begins, to ‘improvised during an experience’ – which involves using improvisation as a means to react to events in an ongoing and dynamic experience. Figure 4-2 summarises how viewing this new dimension of tasks against the dimension of forms of participation suggests a matrix of four overlapping areas that require attention, and are each influenced by one of the four traditional forms highlighted earlier.

Before an experience can begin, its creators must consider two authoring tasks; preparing the performance elements of the experience, and authoring the game-play elements:

- **Authoring Play:** The first task is for experience creators to consider how to author the game-play elements of the experience. This involves authoring any location-based content that players interact with as they move through the city, for each physical space that the experience is deployed in. This content may create a sophisticated narrative for players to engage with, and also steers the movements and interactions of players as the experience unfolds, as, for example, seen in the trail of clues of *Uncle Roy All Around You* and *I Like Frank*. Finally, this task involves designing, implementing and configuring the underlying game mechanics of the experience, taking inspiration from conventional computer games. This involves defining how players interact with pre-authored content and how it is delivered to them, how players communicate and interact with one another; for example the chasing and catching mechanics of *Can You See Me Now?*, and how they inhabit the mixed-reality environment using a variety of devices and platforms; for example creating maps and virtual models of the physical space. This task both takes into account and is integrated with any theatrical performance elements of the experience.
- **Preparing Performance:** Next, the creators of a mobile mixed-reality experience must consider the performance elements of the experience, as influenced by conventional theatre. This includes scripting

performances by actors, for example the climactic interaction between a performer and a player in the limousine in *Uncle Roy All Around You* and at Futurelands in *I Like Frank*. It also includes constructing the supporting sets and props for these performances, for example Uncle Roy's office, and defining the processes that allow players to participate and move through them to a sustained and manageable schedule. Finally, this task involves defining and scripting how players are inducted into the experience at the beginning, to carefully frame the experience in the appropriate manner.

During an experience, its creators must consider the role of improvisation in shaping the experience as it unfolds, again by considering two tasks; the role of improvised performance and the role of improvised game-play:

- **Improvising Performance:** This task draws on street-theatre, allowing the creators of an experience to manipulate it by improvising elements of performance, in order to both heighten the experience for players and to respond to any unexpected occurrences that may threaten a player's engagement. Improvised performance on the streets may form a substantial creative element of an experience, as seen in *Can You See Me Now?*; where mobile performers continually improvise audio performance in response to the actions of online players and their experience of the changing physical environment. Conversely, and most importantly, improvisation allows performance around failure; intervening in a player's experience in the event of the player or the technology doing something unexpected. If a player becomes lost, or the mobile technology that they are using fails, or if they interpret pre-authored content in an unexpected manner, then performers can use improvisation as a tool to react to this, injecting new content and using live performance on-the-fly to get the player back on track. This is especially important in experiences such as *Uncle Roy All Around You* and *I Like Frank* that place players on the streets in an environment that is rich with ambiguity and choice – to adhere to a strict schedule of pre-authored performances and to keep players safe and engaged with pre-

authored content, performers must make constant use of improvisation to orchestrate the experience.

- **Improvising Play:** The final task is how improvisation is used as a mechanic to introduce dynamic game-play into an experience both by players and performers, and particularly how this is influenced by conventional role-playing games. A significant use for improvised game-play is as a creative tool in the hands of an experience's creator; in much the same way as a games-master in a conventional role-playing game can take a basic scenario and improvise a rich and immersive experience from it. Improvised game-play provides a mechanism for creators to explore new game-mechanics as an immediate response to the unfolding actions of players, as seen in Savannah; where the teacher improvised high-level scenarios that built upon the basic functionality of sending messages, receiving location-based content and a simple health scale. This may be part of a strategy to rapidly prototype game-play before it is finally implemented in code, as part of an iterative development process. Conversely, players in an experience role-play to a certain extent, and this is revealed in an individual's particular playing style. In Uncle Roy All Around You, online players can choose whether to play the role of a helper, and direct mobile players to achieve their goals, while others deliberately hinder. Others take it upon themselves to act as self-appointed 'experts' who explain how the experience works to new players. Mobile players are free to interpret content in a variety of ways, In Can You See Me Now?, online players form their own strategies, either revelling in the thrill of the chase and only surviving in the experience for a short time before being seen, or strategically moving through the city to obtain the highest score. This emergent and improvised play has implications for other tasks, particularly aiding or complicating the performance task during the experience and adding extra considerations for authoring.

4.3.2 Challenges Raised by the Framework

The framework of participation and supporting tasks shows that mobile mixed-reality experiences, particularly artistic performances, draw on elements of the four areas of pre-authored theatrical performance and game-play prior to an experience, and improvised live performance and dynamic game-play during an experience, as can be seen in each of the four experiences that underpin this thesis.

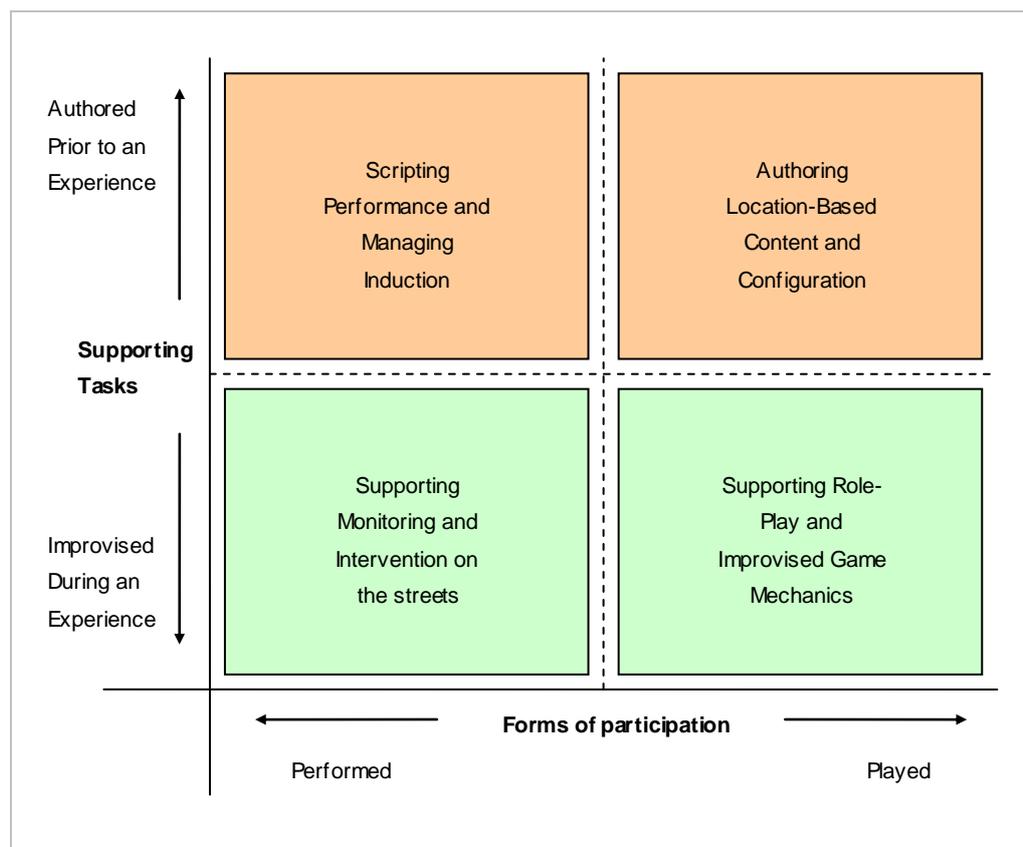


Figure 4-3: Challenges raised by the framework

Figure 4-3 summarises how this framework raises particular challenges for supporting mobile mixed-reality experiences with authoring and orchestration:

- **Authoring location-based content and configuration:** Chapter 2 showed that many mobile mixed-reality experiences involve location-based content that is triggered by the movement of mobile players through the city. This raises a significant challenge for authors, as they must support an experience with large amounts of location-based content that is then delivered to players, and configure the experience

for each physical space in which it takes place. This not only forms the basis of a suitable engaging and complex experience, but must also be structured to inherently steer the actions of players to complement the scripted performance in the experience.

- **Scripting performance and managing induction:** Experiences that are presented to the public as performances raise challenges for both authoring and orchestration. Performance must be scripted and managed in such a way that it can be integrated with the other content or technological elements of an experience, particularly when players are inducted into it. A particular challenge for orchestration is to create tools and processes that allow the flow of players through an experience to be managed, especially at critical performance junctures, to ensure that everything is working correctly and that everything is in place to receive a new player.
- **Supporting monitoring and intervention on the streets:** A significant challenge for orchestrating a mobile mixed-reality experience is to be able to improvise performance in response to events and possible failures on the streets and online. This requires the construction of tools and strategies to covertly monitor not only what players are doing, but also what they have been doing and what they are likely to do, particularly when players are moving in the physical environment and only a partial picture of unfolding events is available. Consequently, this requires supporting tools that are able to manipulate an individual player's experience and to intervene in a manner that minimises the possibility of the player losing engagement.
- **Supporting role-play and improvised game mechanics:** Finally, a challenge for orchestration is to support emergent game-play through improvisation and role-play. Again, this involves supporting tools to monitor and steer a player's experience in direct response to their actions, but also to create a framework that allows experience creators to manually operate the game mechanics of an experience, in order to explore and rapidly prototype new experience scenarios.

While the distinction between authoring and orchestration may be blurred by making use of iterative authoring and participatory design processes during the development of an experience, by addressing these four challenges through the creation of tools to support authoring and orchestration for use both before and during an experience, experience creators should be able to create a rich, engaging and, most importantly, sustainable and robust mobile mixed-reality experience for all players.

4.4 Conclusions

This chapter has presented a framework for supporting mobile mixed-reality experiences, based on the experiences and related applications described in the previous two chapters.

By classifying experiences by genre, it has argued that mobile mixed-reality experience comprise a wide spectrum of forms of participation, including elements of both performance and game-play on the streets of the city, and placed these alongside more conventional forms such as street theatre, conventional theatre, role-playing and computer games.

Next, this chapter has defined an orthogonal dimension that comprises of the supporting tasks inherent in mobile mixed-reality experiences, particularly authoring tasks that occur before an experience and improvisational tasks that enable orchestration during an experience. This has allowed it to highlight particular challenges for supporting mobile mixed-reality experiences with authoring and orchestration; how to support the authoring of location-based content, how to integrate scripted performance, how to monitor and intervene on the streets, and how to support improvisation as a core game mechanic.

The next two chapters will describe the processes and tools for supporting the authoring and orchestration of mobile mixed-reality experiences in detail, and how the construction of our four experiences has led to the development of general solutions for each area. A final chapter describes the implementation of the four experiences, showing practically how authoring and orchestration fits in with, or imposes additional requirements on, the rest of each experience.

5 Tools to Support Authoring

5.1 Introduction

Chapter 4 argued that content authoring plays a significant role in creating a mobile mixed-reality experience. This chapter now focuses on how to support the challenge of associating different kinds of complex, location-based triggers with areas of the physical environment, and quickly configuring an experience for each new location in which it is deployed. Drilling down into this area, it demonstrates the development and use of a new solution to this challenge, before drawing out a set of general principles and common guidelines for authoring.

This chapter begins by reviewing existing authoring tools that support location-based content authoring; both those designed for mobile mixed-reality experiences and those from related areas, such as conventional computer game level and map design, and location-based services, in order to highlight their particular strengths and weaknesses.

In response to this, this chapter presents the ColourMaps system, a new tool for authoring and configuring location-based mobile mixed-reality experiences that has been developed to support the four experiences described in chapter 3. It examines the key innovations of the ColourMaps system, and shows how it supports the authoring challenges raised during the development of the four experiences and by other mobile mixed-reality experiences that revolve around location-based content.

Finally, this chapter summarises the design features, limitations and possible extensions of the ColourMaps system, and reflect on how the innovations it contributes fit can fit into a broader scope.

5.2 A Review of Existing Authoring Tools

As mobile mixed-reality experiences continue to grow in number, popularity and sophistication, there is arguably an equivalent increase in the need for more powerful dedicated tools to support their authoring and configuration. However, these tools should consider how to support complex narrative-based

experiences and a range of associated configuration requirements, rather than just location-based content.

Conventional computer game development tools are widely used to create complex experiences for desktops and consoles, mainly focusing on 3D games and addressing issues such as animation and modelling, artificial intelligence, scripting and level design. These tools are generally developed in conjunction with a specific platform or engine with its associated strengths and weaknesses in mind. Some mobile mixed-reality experience developers have, as seen in chapter 2, attempted to adopt these tools, for example by using level editors in order to configure their experiences. Similarly, there is widespread use of Geographic Information Systems to design, collate and use location-based data for mobile applications, although again these may not necessarily have been designed for the purpose of supporting mobile mixed-reality experiences.

While tools to support the authoring of mobile mixed-reality experiences may draw upon elements of the tools described above, they should be developed from the ground up to deal with support for these experiences; for example by not just focusing on creating location-based content, but also supporting rapid configuration, artist-led development and operation, and particular mobile traits. It is with this in mind that this section now examines a range of relevant authoring tools and solutions.

5.2.1 Geographic Information Systems

Geographic Information Systems (GIS) are primarily designed to manage data that is spatially or geographically referenced to the earth. They allow users to create, annotate, manage and present large amounts of data for use within many different fields. In the mobile field, GIS are commonly used in large-scale deployments of location-based services by mobile telephone operators. In terms of content provision to users, GIS provide access to a large amount of location-based data, for example satellite imagery and roads, combined with information such as street names, or the locations of amenities. This data is referenced geo-spatially, allowing a user to query the system with their current location and gain access to the relevant information, usually presented in a

lightweight format that can be displayed on a mobile device or within a web-browser.

The use of GIS in this area is becoming more widespread with the increase of sophisticated mobile devices and high-speed mobile networks, with many mobile operators providing services that allow their users to access useful information based on the automatically reported position of their mobile phone. Similarly, web-service providers such as Google support access to their online GIS suite from mobile browsers, providing map data combined with business and street directories and geo-coding functionality through Google Local (Google, 2007).

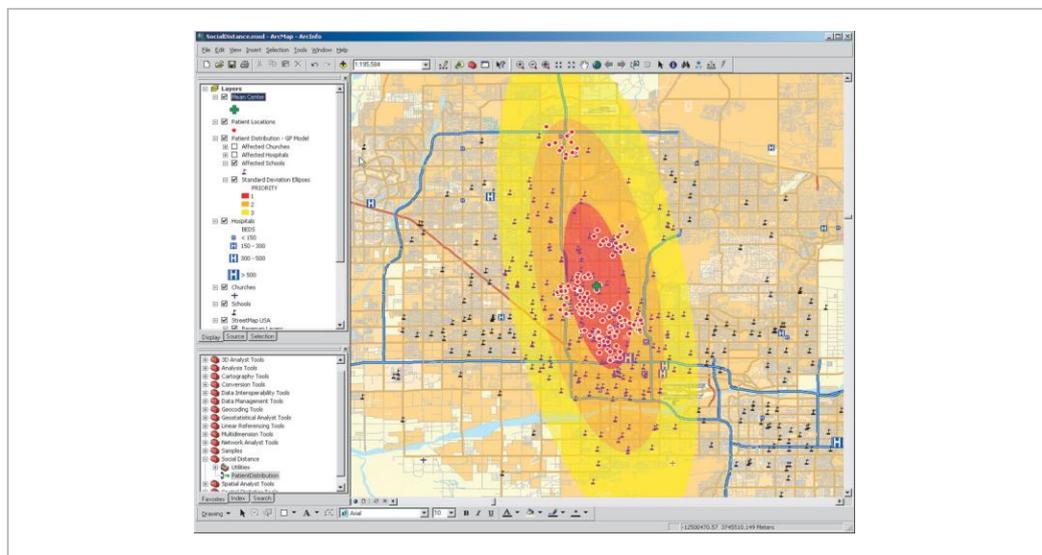


Figure 5-1: ArcGIS desktop authoring

For content or data authors, GIS provide suites of dedicated and powerful tools for inputting and manipulating the large amounts of data involved. ESRI, one of the more established commercial GIS developers, have produced the ArcGIS (ESRI, 2000) suite of tools that support data authoring and manipulation, as well as geo-statistical and topological analysis. As with many GIS, data may take the form of raster images such as maps or aerial photographs, or vector geometries representing roads or other boundaries, as shown in figure 5-1. GIS authoring systems aim to ease the time-consuming task of preparing data from a large number of sources for use within the application.

GIS are often developed around a geo-database, which presents vector or numerical data to an application in response to a spatial query for both authoring and runtime uses. For example, a spatially enabled database such as PostgreSQL (PostgreSQL, 1996) with PostGIS extensions (Refractions Research, 2005) may be queried for environmental data for a given location, which can then be rendered as part of an authoring or analysis application, or for an end-user's mobile device. High-end GIS such as ArcGIS server allow developers to extend the functionality of authoring tools programmatically in order to create new applications.

The fundamental functional requirements of a GIS have been categorised by (Rhind et al., 1988) as being data input and encoding, data manipulation and data management; and supporting retrieval, analysis, representation and presentation. However, while many GIS provide powerful support for these requirements, their complex nature is targeted specifically at the provision of location-based information. While the introduction of simple developer interfaces has allowed experience developers to make use of publicly accessible systems such as Google Earth and Google Maps in applications such as GeoTracing, GIS systems do not inherently support the more complex narrative structures of many mobile mixed-reality experiences, of which location-based content is just one part. GIS may form part of future suites of authoring tools, but they do not constitute tools for authoring mixed-reality experiences in and of themselves.

5.2.2 Game Engine Authoring Tools

Typically creating a computer game involves creating a game-engine, which delivers the experience to the player, and the subsequent population of the engine with supporting assets which describe the environment that the player inhabits, any artefacts that they may interact with and scripts that define the narrative of the game.

Early computer games involved much of this asset creation, or content authoring, being undertaken by the game's programmer. However, as games have grown in size and complexity game development companies have taken to creating dedicated tools to allow dedicated asset authors to populate a game

with terrain, models, animations, sound and scripts. As with GIS, it is interesting to review game authoring tools to see if there are any features or requirements that are relevant to mobile mixed-reality experiences.

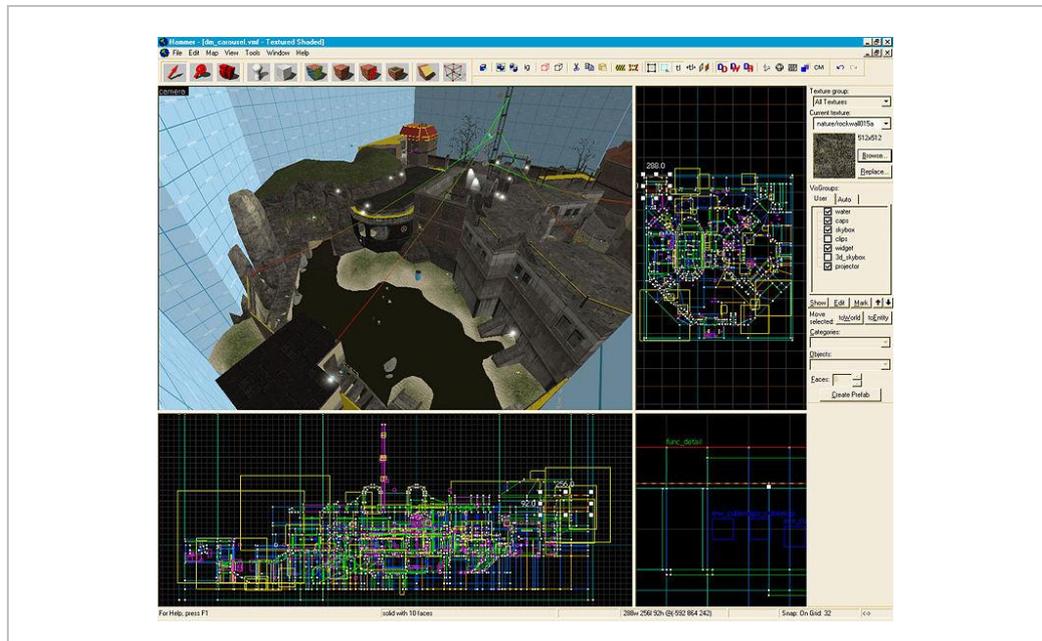


Figure 5-2: Editing a level in Valve Hammer

Half-Life 2 is one of several first-person-shooter (FPS) games built using the modern Source game engine (Valve Corporation, 2004). The creators of Half-Life 2 and Source released an editing package, Valve Hammer to allow the games community to construct their own levels for the standard game, and also to create new games by making modifications, or “mods”. The editing package allows users to create new levels by using a 3D level editor, as shown in figure 5-2, allowing the construction and texturing of the polygonal structures that define the terrain, placement of in-game objects and associated scripting to define how the objects behave. Valve Hammer follows a trend of game developers to release their in-house tools – ARQuake, described in the literature review, was a modification to the Quake engine (id Software, 1999), with a level created in Quake Radiant, the Quake equivalent of Valve Hammer. Such game engines also allow other assets such as character models and animations to be created using dedicated three-dimensional content authoring packages, for example 3ds Max (Autodesk, 2007), Maya (Autodesk, 2006), and Sketchup (@Last, 2000).

It takes a large amount of technical skill and knowledge to author an environment or game level using a tool as described above, with the interface taking time to understand and to become fluently used. Authors must understand the capabilities, strengths and weaknesses of the game engine that they are designing for in order to create a functioning and polished level; for example, by understanding that many complex structures, or some advanced game logic, may be beyond the engine's capabilities.

Attempting to appropriate these tools for authoring content for a mixed-reality experience may not be the most pragmatic choice. While the online component of a mixed-reality experience may draw many parallels with an FPS in terms of appearance, the mobile component has very different requirements. ARQuake was able to use Quake Radiant as the experience itself was based on the Quake engine, and was primarily visual in nature, taking full advantage of the existing engine. However, an experience that is fundamentally based on text narratives or other structures than a visual environment requires that these be the focus of the authoring tool – something that is a supporting aspect of a game engine rather than the core paradigm. Similarly, game engine tools are not designed to support the physicality of a mobile mixed-reality experience, including the irregular nature of physical terrain, and the need to rapidly and easily create new 'levels' for new locations.

In contrast, many of the mobile mixed-reality experiences that were reviewed in chapter 2 consider the playing environment to be a two-dimensional space, with most being played in an outdoor space that does not use the player's elevation as a degree of freedom. For this reason, it is interesting to consider older, two-dimensional computer game level design tools for their spatial authoring capabilities – in some respects these can be viewed as being the opposite of high-powered GIS tools as they provide simple, easy to use and lightweight authoring functionality. Secondly, some of the experiences described previously are recreations of classic, two-dimensional arcade games.

Classic tile-based games are often released with their own editing tools, or use simple, readable, text-based level formats that allow players to directly manipulate levels and rapidly review the results in the game.

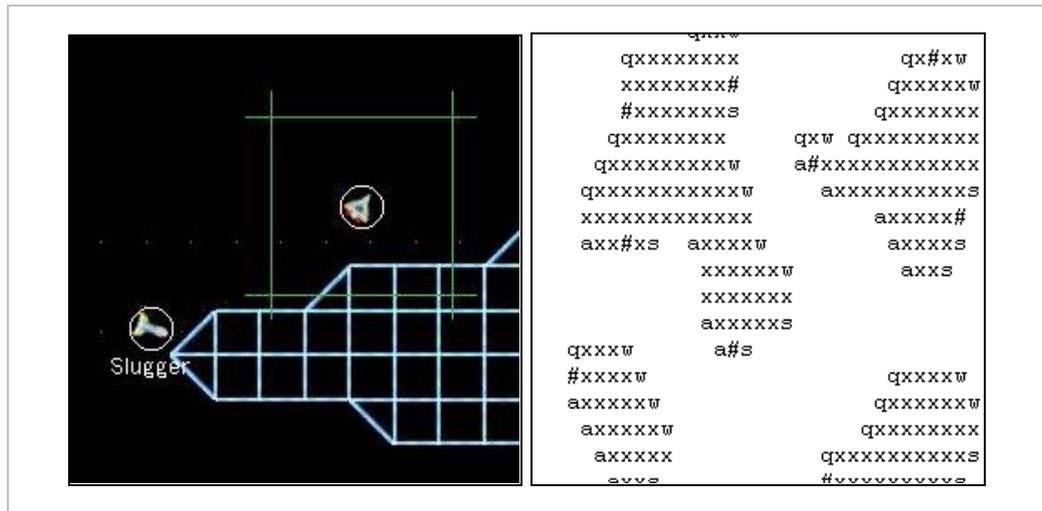


Figure 5-3: XPilot, left, editing a level with Notepad, right

Figure 5-3 shows XPilot (Stabell and Schouten, 1991), an example of a tile-based game. The player controls a space-ship's flight around a two-dimensional terrain built from tiles. Levels are saved in a readable text format, and as such can be viewed – and more importantly created and edited – using a plain-text editing application such as Notepad. Different text characters represent different tiles within the game, and the spatial layout of the characters in the file represents the spatial layout of the level itself. The text file also contains meta-information that is loaded with the level, such as the number of enemies that will also be present in the level.



Figure 5-4: Repton, left, painting a new level, right

Figure 5-4 shows Repton 3 (Superior Software, 1985), another early tile based game, that this time is distributed with a built in level editing application. All

tiles are the same size, representing the player, monsters, walls and other objects. The application allows the user to draw regions of tiles as with a brush in a paint package, although they are limited in the complexity of structures that can be built as all tiles have the same fixed dimension.

As with GIS tools, it can be seen that game-engine design tools are not suitable as the sole authoring tool for a mixed-reality experience. However some techniques, particularly the simple authoring techniques from two-dimensional tile based games, may prove useful in a mobile mixed-reality authoring tool.

5.2.3 Context Aware Systems

As described in chapter 2, there are several experiences that can be defined as making use of context aware systems, for example those that were developed using the Mobile Experience Engine framework. Context aware computing devices react to their changing environment in an intelligent way so as to enhance the computing environment of the user (Schilit et al., 1994). These systems are often used to support mobile experiences, although do not necessarily focus directly on the challenges of providing complex location-based content to users.

The Stick-e Note Architecture (Pascoe, 1997) allows users to leave electronic post-it notes tied to a certain location, the display of which is triggered when a user returns to that area within a given context, for example depending on what time it is, who they are with or what they are deemed to be doing. The architecture allows a variety of media to be associated with a note, including text and richer media such as video.

Authors create Stick-e Note applications using a hierarchical editor that defines individual notes as objects which are then linked graphically with other objects that represent trigger conditions, such as an object that checks who the triggering user is with. The authors describe an example application that displays relevant information as a user moves around a theme park, which operates by defining rectangular region objects and having the system trigger a piece of content when the user's location is reported as being within the

rectangle. If the author wishes, the interface can be extended programmatically to allow these rectangles to be authored in a more practical manner.

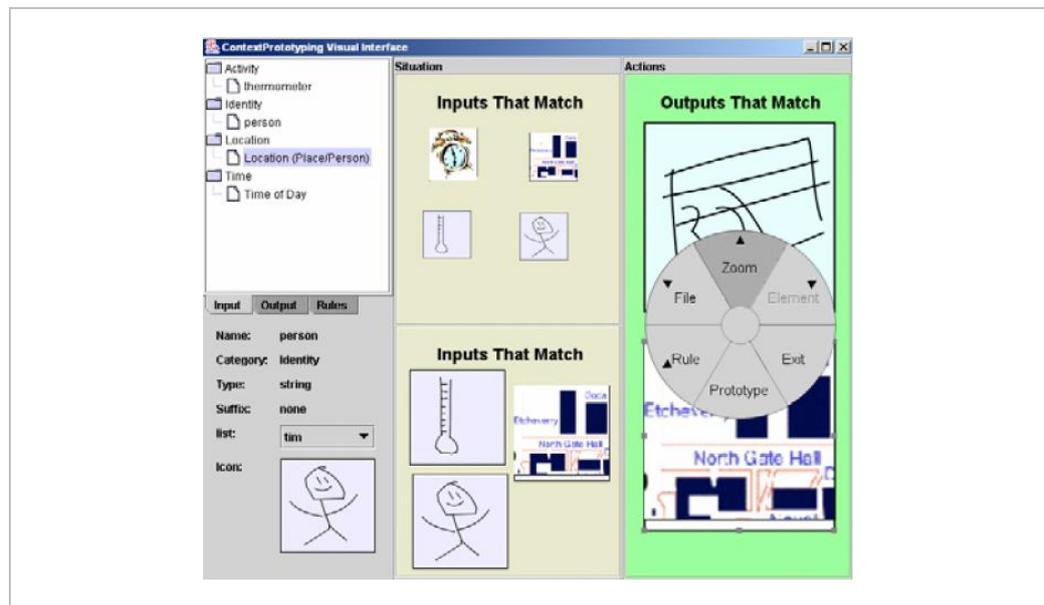


Figure 5-5: The iCAP editor

iCAP (Sohn and Dey, 2003) is a similar system that supports the rapid prototyping of context aware applications. The focus of the system is to support authors in creating applications without having to write code to support mobile devices and input sensors, instead using an editor to author the actions of the application. Figure 5-5 shows the iCAP editor. Using a sketching paradigm, authors instantiate input and output devices, and link them with simple interaction rules. Inputs are identified as being of one of four context types; activity, identity, location and time; whereas outputs may be a discrete binary or a continuous display.

Notably, iCAP provides a testing mode that simulates the output of real sensors within the editor, allowing the author to test rules and displays. While the sensor simulation may not truly reflect the potentially very variable output of real sensors, it allows the author to rapidly assess whether their construction is feasible or operating in approximately the correct manner.

5.2.4 Topiary

Topiary (Li et al., 2004) is a tool for prototyping location-based experiences. The authors state that it currently requires a high level of technical expertise to

build location-based experiences, in turn making it hard to prototype and test new designs.

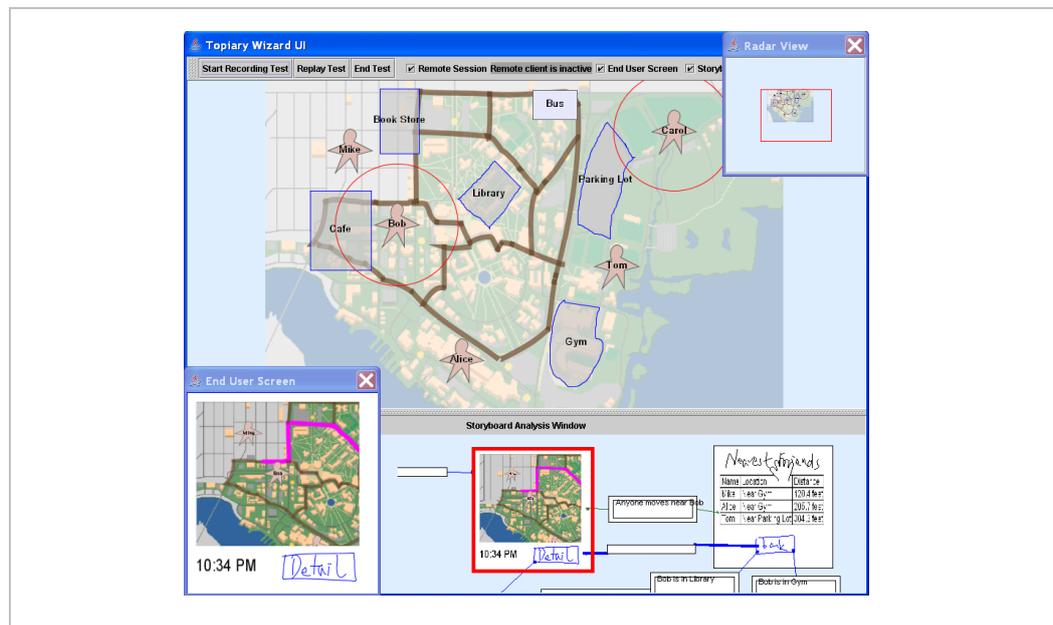


Figure 5-6: Topiary authoring map

Topiary provides a map interface that shows the location of tracked players and other artefacts, shown in figure 5-6. Authors then annotate the map to create location-contexts, for example a user being in a certain location with another user. These scenarios trigger other storyboarded content that can be displayed on a mobile device. Finally, as with iCAP, an application can be tested within Topiary using the testing interface which displays both the content that would be seen on the mobile device and a Wizard of Oz overview of the environment that allows the author to fake sensor input, such as position data.

5.2.5 Mediascapes

Mediascapes (Hull et al., 2004) is a framework for creating media-oriented, location-based mobile applications. Mediascapes provides a visual authoring environment for defining location-based media triggers supported by an extensible scripting language, and a run-time client. As described in chapter 2, the Mediascapes framework was used to create the Riot! 1831 experience.

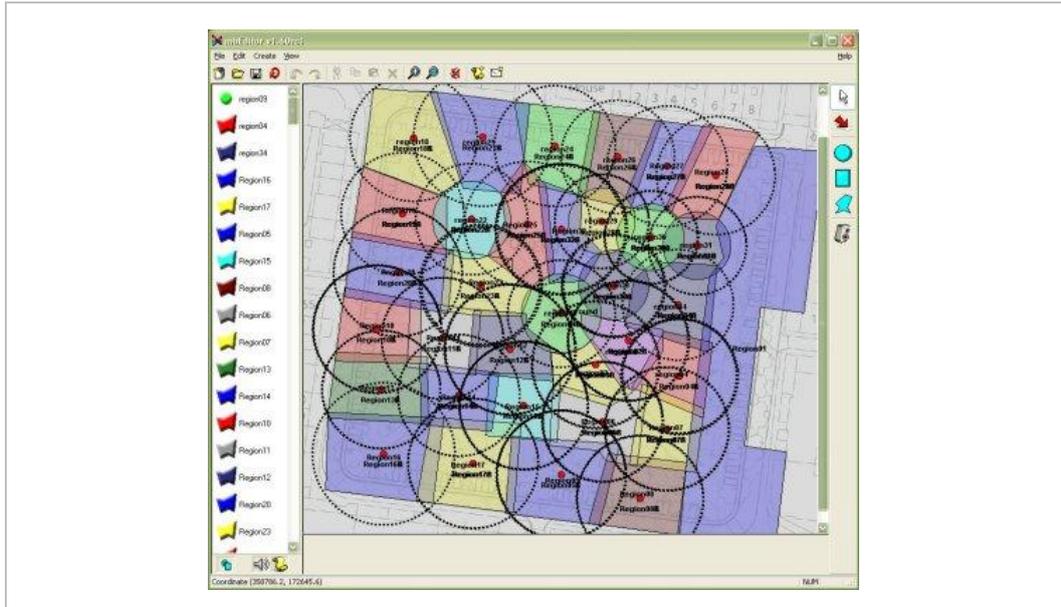


Figure 5-7: Authoring regions in Mediascapes

Figure 5-7 shows the Mediascapes authoring environment, which allows an author to define primitive regions that trigger media on a mobile client. Authors can specifically define the behaviour of how media is triggered on a mobile device as a player moves in and out of these regions. The Mediascape is exported as an XML file and can then be loaded by the run-time application on a mobile device – although the authors suggest that the file can be used by any application that can read the Mediascapes format.

The Mediascapes authoring environment imports a map image, which is used as a background for authoring against using vector shapes. The tool supports primitive shapes as trigger regions; circles, rectangles and freehand polygons may all be drawn, scaled and rotated, and shapes may overlap to allow two triggers to operate concurrently. However, it does not support more fine-grained raster authoring. Authors then use the scripting language to define how the trigger regions respond to position events generated by a player; the scripting language allows complex language to be embedded within a trigger and to call application-specific logic within the mobile client if it has been extended by the author. The authoring environment provides functionality that allows the author to test their work as they design it, by triggering regions as if with real position data.

Notably, a simplified version of the Mediascapes framework has been publicly released for non-commercial use to allow more users to create their own location-based experiences. The release consists of the authoring package and the basic mobile run-time application. Create-a-Scape (Futurelab, 2007) has been set up to encourage the education sector to make use of the system as part of learning experiences.

5.2.6 Activity Zones

Activity Zones (Koile et al., 2003) is a toolkit that has some similarities to the Mediascapes framework. Again, it is a toolkit for creating location-based experiences, based on marking physical regions within an environment. An Activity Zone is described as an area of specific activity, for example a room, arrangement of furniture or doorway, and the system is primarily focused on supporting domestic applications regarding these areas. These areas are identified using video analysis techniques on collected footage.

The Activity Zones tool requires the author to label and aggregate the reported regions by creating an encompassing primitive vector shape, as in Mediascapes. These shapes are saved as a set of coordinates within an XML file. Next, the author defines a set of context rules, using a simple pseudo-code language, that define how a set of output devices react to where users are within the environment and in relation to the regions; to this extent the system is similar to the other context aware systems described earlier.

5.2.7 Summary of Existing Tools

To summarise this section, the authoring tools and systems described each have their own associated strengths and weaknesses for use within a mixed-reality experience.

GIS are powerful, industry standard systems for managing large amounts of location-based data. Their ability to combine raster and vector modes is important for selecting the most appropriate format for a given data type. However, their primary focus is data management – there is little support for how this data can be used in a complex narrative experience.

Similarly, game engine design tools are specifically geared towards providing a rich, visual experience for the game player. While some game engines may be adapted by mixed-reality experience authors, this may not be the most appropriate solution. Two-dimensional or tile-based games, however, strongly support simple, intuitive, visual formats that can be readily used by players in creating their own game levels.

A common factor between context-aware and other mobile experience frameworks such as Activity Zones, Topiary and Mediascapes is how they seek to support authoring through defining spatial regions, or zones, and then associate content or media triggers with them. Experiences created with these tools historically appear to be relatively simple, small-scale or one-off installations. A possible criticism is that they have not yet been used to construct complex narrative experiences by non-developers – although the Mediascapes framework is moving in this direction. This potentially raises additional requirements in terms of the pragmatic usability and simplicity of the tools for mobile mixed-reality experience authors, rather than developers.

5.3 ColourMaps

This chapter presents the ColourMaps system, an authoring tool that aims to support the authoring and location-based content provision requirements of mobile mixed-reality experiences discussed in chapter 4. The ColourMaps system was developed and tested iteratively alongside our four experiences, and supports a critical element of each experience. It is a significant departure from other approaches as it aims to pragmatically support content authoring through the use of familiar, simple and flexible means.

This section will describe the key innovations of the ColourMaps system, and for each one describe how it successfully supported one or more experiences. The next section will then generalise these innovations and the lessons learnt.

5.3.1 Principles of ColourMaps

During the initial prototyping phases of the early experiences, location-based content was configured and served to players by defining simple circular triggers generated from a single position and an associated radius stored in a

text file. On entering a trigger, an associated piece of content was given to the player. While this functionality had the benefit of being implemented quickly, it became apparent that the artists had difficulty configuring content that accurately reflected their vision for the experience, particularly the inability to configure content triggers that encompassed more complex spaces, for example a street corner or a narrow alleyway, the inability to get a sense or overview of the configured content – as the definitions merely consisted of a list of GPS coordinates, and the inability to easily move content that was potentially placed incorrectly.

As the name suggests, the principle of ColourMaps is that a designer creates and configures a location-based experience by colouring over a map. This concept was inspired by the model of creating levels through painting as described for tile-based games in a previous section. This approach was originally chosen for the following reasons, highlighted during early prototypes:

- **Simplicity:** the need to provide experience authors with a straightforward approach to creating content that could be easily understood and applied by a non-developer.
- **Familiar skills and tools:** the need for designers, in our case specifically professional performance artists, to be able to employ their existing skills and to work with familiar and preferred tools, for example standard image editing packages such as Photoshop (Adobe Systems, 2007b).
- **Flexibility:** the need to enable maximum flexibility in terms of defining different kinds of content, and precisely matching this content to complex urban spaces, without restricting the author to preset shapes or levels of detail.

A ColourMap is an image where the colour of each pixel represents the identity of a piece of content to be used in a location-based experience. The image is geo-referenced so that pixels represent location, as on a map. Authors map

colours to content using a simple meta-data format, or make use of pre-programmed functionality within the system.

A ColourMap can be created using any paint package, and saved using a loss-less compression format, for example GIF or PNG, to preserve the original colours and shape definition. This gives the author direct control over the spatial layout of content within the system. ColourMaps allow the author to use a paint package with an intuitive paradigm that is familiar to them; the results of which can be fed directly into the system supporting the experience.

Once created, a ColourMap is loaded into the system supporting a mobile mixed-reality experience where it is used to directly serve content to players. As a player moves in the physical environment they also move in the ColourMap landscape. The ColourMap is then sampled to find the colour of the pixel at the player's location in a process analogous to the eye-dropper tool in Photoshop, retrieving a unique RGB value to be used as an index into lists of text or other media content to be triggered. The ColourMap system can also perform movement operations depending on the sample colour – it may search for the nearest pixel that is the same colour as the one it has found, or search for the nearest different colour. The location of this pixel can then be used as a content index as before.

As described in the previous section, many existing location-based or mobile authoring tools are based around the creation of vector primitives, whereas a standard image editing or drawing package allows the mixing of raster and vector functions when creating shapes, or in this case content regions. GIS, on the other hand, allow the combination of raster and vector authoring – as noted by (Winter, 1998) who argues that this combination enhances integration and data fusion. This is powerful, as vector shapes can be drawn and edited rapidly, while raster shapes allow the fine tuning of individual pixels. This latter feature is particularly important. When authoring game levels, it is common to draw polygonal, regular or predefined structures whereas freehand drawing is unusual. However, while some cities have a regular grid structure, the majority do not. Supporting freehand drawn raster shapes for triggers enables an author to precisely match content to the physical form of a game area, accurately

filling areas and avoiding problems with using regular shapes that do not properly tessellate or fit the features of the physical space.

Furthermore, when a vector shape or a trigger is created in a conventional map, it becomes a unique object in the system. By using a colour map it is possible to have disjoint areas representing the same piece of content by colouring the separated zones in the same colour. The pixels do not inherently form a group or an object unless the system is made to specifically look for adjacent pixels and process them accordingly.

Using a high colour depth gives the designer the opportunity for several million uniquely coloured pixels and thus an equivalent number of content regions. However, by using a smaller number of distinct colours it is possible to easily see the changes between different regions on the map. It may be hard for an author to visualize a list of numerical positions representing the positions of content regions without creating a dedicated visualization of where they lie. With a ColourMap, the visualization and the definition of the content are the same, and as such may be used for authoring, content provision at run-time, and orchestration tools.

5.3.2 Start Positions

A key requirement of mobile mixed-reality experiences is the rapid configuration of discrete positions in the game area. These can be used by the game engine to configure where objects, players or key positions are in relation to the physical space. These start positions are loaded from a ColourMap when the game engine is started.

Start positions are drawn on a physical map of the game area in a specific colour, as shown in figure 5-8, and all start positions on a given ColourMap represent the same feature within the system; for example positions where online players enter the virtual space are all coloured in blue. The system searches for all pixels that are this colour on the ColourMap and stores their coordinates. When a start position is requested the system selects one of these positions at random to be the starting position for that player.

Locations can be weighted to increase their probability of selection simply by colouring in a larger number of pixels in that area. If a start position turns out to be in a bad position, for example too far away from the main game area, it can be removed by erasing the relevant coloured pixels. This is significantly easier for the author to use than entering a list of coordinates as the distribution of positions can be immediately viewed. As only pixels of a certain colour are registered, the remainder of the map can remain intact as a visual reference for the author.

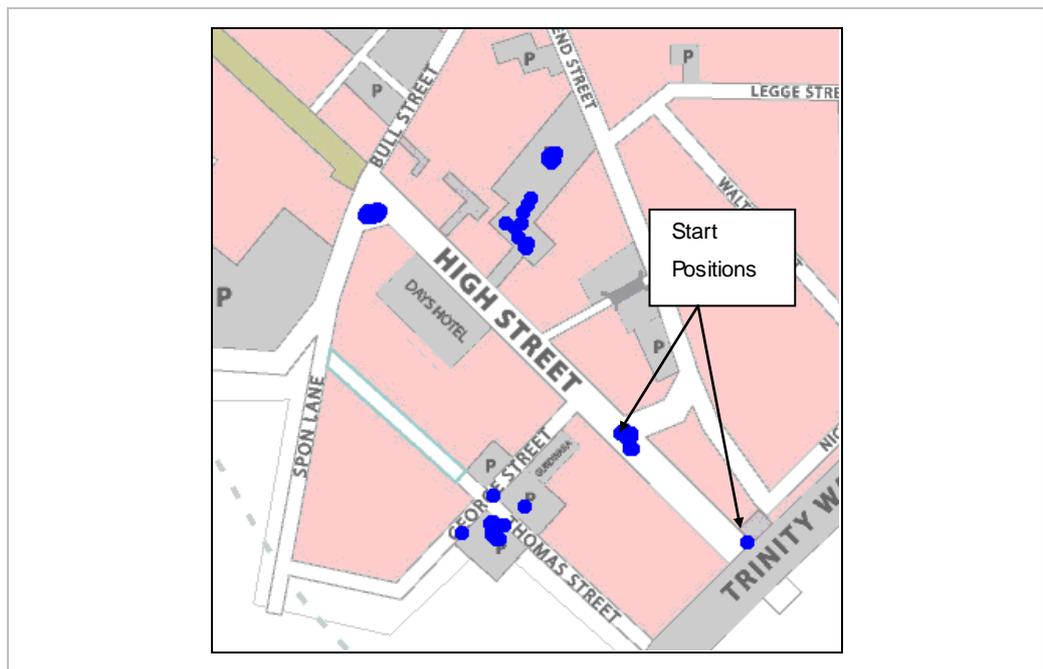


Figure 5-8: Authoring start positions

Start position ColourMaps are used to configure starting positions for online players in Can You See Me Now?, Uncle Roy All Around You and I Like Frank. They are also used to configure the positions of the ‘red markers’ in Uncle Roy. This feature has proved especially valuable in Can You See Me Now?, where online player start positions must be reconfigured for each new location, and potentially during the experience to achieve the desired spatial balance of new players.

5.3.3 GPS Filtering

ColourMaps provide a mechanism for filtering GPS position updates to remove invalid reports by finding the nearest pixel of a certain colour to a given position.

The author creates a layer of colour over the map of the physical game area to indicate areas where a GPS unit would not theoretically be able to report a valid position, for example inside buildings or on areas covered by water. When a position update is sent from a mobile device to the system, it looks up the colour of the pixel at that position on a ColourMap. If the colour is black (the default colour for an inaccessible region) then the system will select the nearest pixel that is any other colour. This new position is assumed to be a more valid position for the GPS unit to be in; as such it is reported to the rest of the game engine. Figure 5-9 shows an incorrectly reported position inside a building with the filtered, more plausible, location.

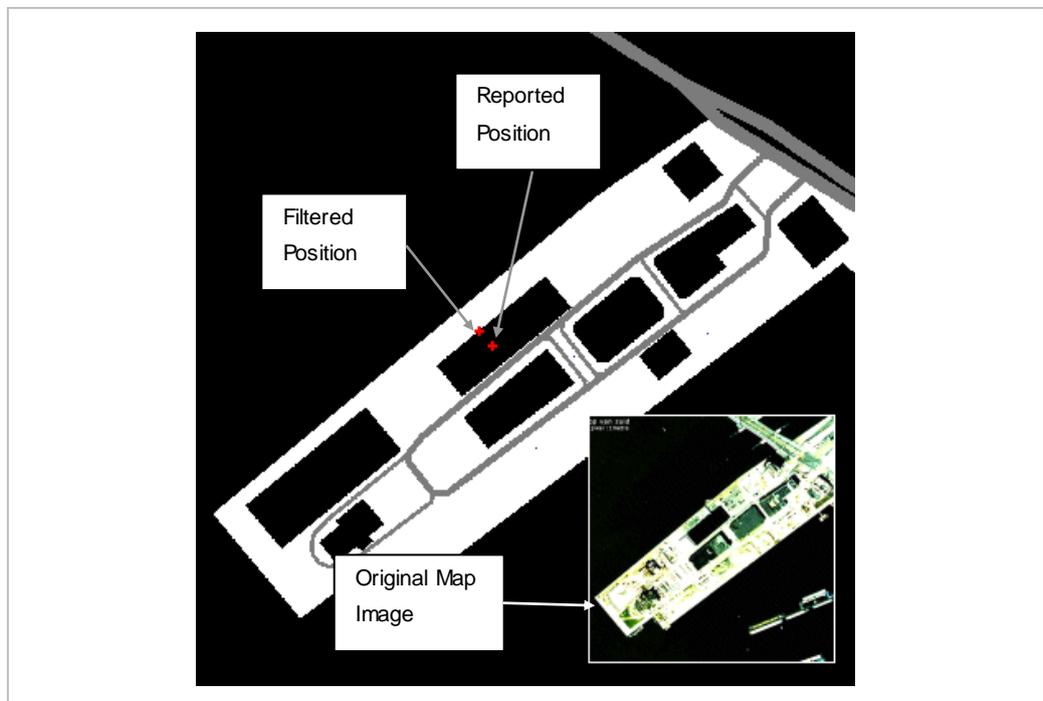


Figure 5-9: A GPS filtering ColourMap

This GPS ColourMap is produced by vector tracing an aerial photograph of the game area to include obstructing features and landmarks such as buildings, roads and water. Photoshop is then used to colour invalid GPS areas in black before the ColourMap is converted to a raster image and saved. The author may also fine tune the raster image to accurately map boundaries. Again, the ColourMap system in this case only registers black pixels meaning that the original map may remain intact on the final image, allowing faultfinding and further alterations.

ColourMaps are used heavily to filter GPS data in *Can You See Me Now?*. In this case, the GPS receiver provides a position update for the mobile player once a second. The avatar that represents the mobile player in the online city moves continuously between the positions reported by these updates, with an appropriate running animation. Invalid GPS data due to poor or reflected signals would cause the avatar to appear to run through solid buildings or across areas of water, breaking the fiction of the game for any online players that may view the event. The ColourMap provides a means to hide these problems from the online player, while both the original and filtered positions are revealed to the mobile player; allowing them to alter their movement accordingly, knowing that it is the filtered position that matters when catching an online player during a successful game.

Finally, as with start positions, the ColourMap can be quickly authored for each new location that *Can You See Me Now?* visits, and may be rapidly updated to exclude areas that provide poor GPS reception and should be considered “out of bounds” by the system.

5.3.4 Clue Trails

A core feature of the ColourMaps system is its ability to serve location-based content to mobile players in mobile mixed-reality experiences that involve non-linear narratives.

Mobile players report their positions using GPS as in *Can You See Me Now?* and *Savannah*, or self-reported positioning as in *Uncle Roy All Around You* and *I Like Frank*. In return each receives one of a number of pieces of content, either a text clue or the filename of a piece of media to be displayed. A major requirement of our mobile mixed-reality experiences is that an author is able to configure a specific and detailed clue trail, or set of text clues or media, for each city that the experience is deployed in. This requirement is fulfilled using ColourMaps.

A ColourMap is used to define a non-linear set of clues which are served to a mobile player based on his or her location. The clues are defined in a simple

XML format, with each clue consisting of several potential text strings and tags defining the RGB value of the colour that will trigger it.

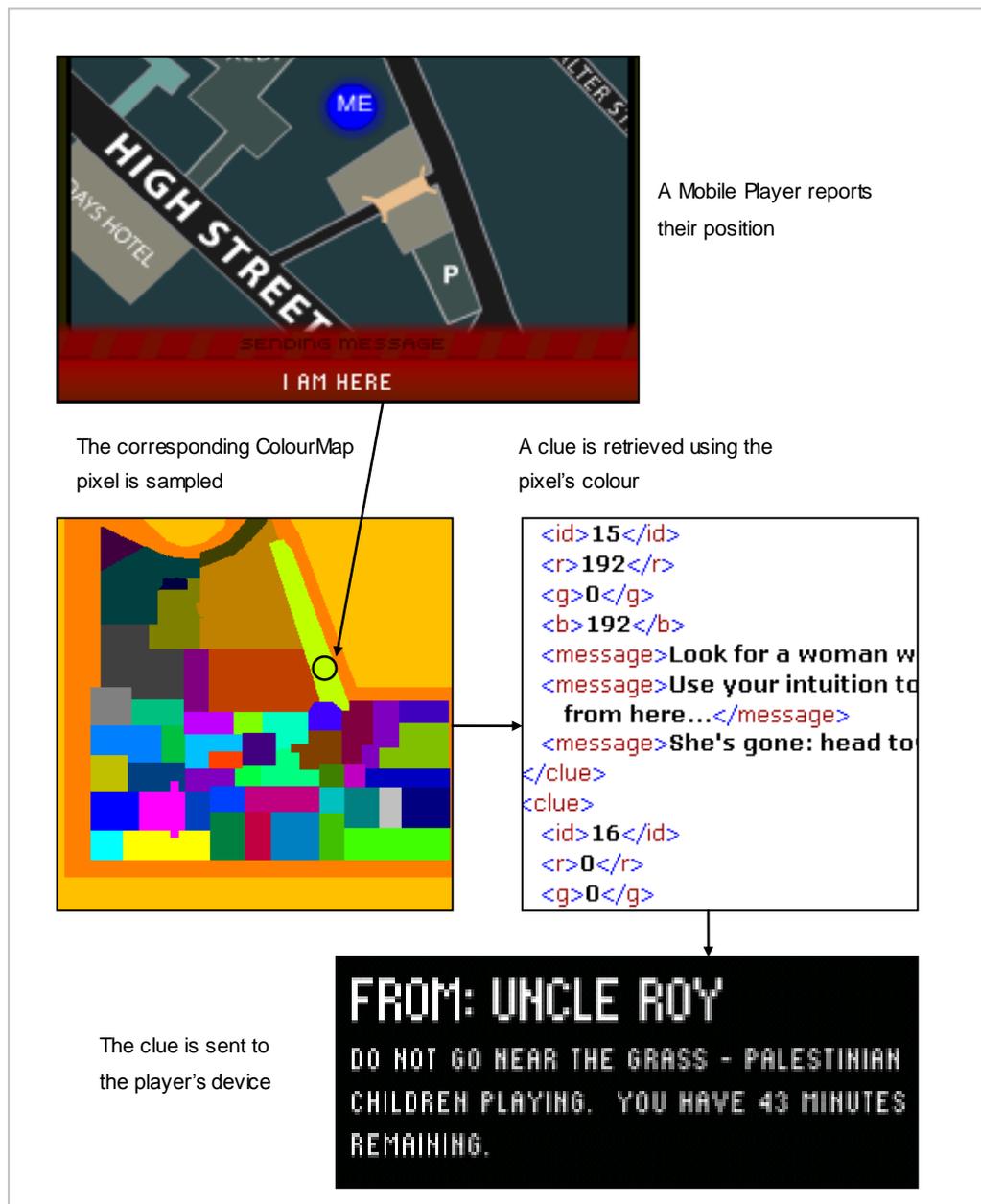


Figure 5-10: Serving clues with a ColourMap

As a mobile player moves their position is reported to the system which in turn places them appropriately on the ColourMap. As with the GPS filtering ColourMap, the system samples the colour of the pixel on the ColourMap corresponding to this reported position. This value is then used as a key to retrieve the set of clues defined for that particular colour region, and the correct clue is displayed on the mobile player's device. If a colour is not found, then a

sufficiently generic default clue is returned implying that the player is in an unknown area:

“I cannot guide you out here. You have got lost. Go back the way you came”

This process of triggering a trail of clues is shown in figure 5-10. In this way, the mobile player follows a trail of clues through the game area.

As with a normal map, a ColourMap can be based on an arbitrary transformation to the physical space it represents - it may represent a space a few hundred metres across or a few hundred kilometres across. Can You See Me Now?, Uncle Roy All Around You and I Like Frank use ColourMaps that are built on top of an aerial photograph where one pixel of the image represents one square metre of real space, and so there is an obvious mapping between the ColourMap space and physical space. However, Savannah is played on a school playing field with few physical landmarks. In this case, the supporting ColourMaps are designed with unit dimensions, and then a transform is applied to scale and rotate this to map to an arbitrary rectangular shape on the playing field, which is marked using flags. In this sense, ColourMaps can be created either using a physical map as a template for laying out the clues, or the physical space can be marked to represent the space that the ColourMap defines.

The following sections will describe how these concepts have been extended to create a more sophisticated system, and how authors use ColourMaps in this manner to create a mixed-reality experience.

5.3.5 Authoring Layers and Ideal Routes

When authoring the clue content for a new mixed-reality experience, or the redeployment of an existing system, the beginnings of the clues are written as the author walks the game area, based on their immediate impressions and thoughts. An initial version of the content is authored based on these walks, before being tested and refined as an iterative authoring process. The ColourMaps system was extended to support this process of first annotating the planned game area before beginning to author content against it.

The author begins to create the content system using a physical map of the game area as a base. They mark ideal start and end points for a mobile player's experience based on the location of the hosting venue; where mobile players begin, and the end of the game; for example "Uncle Roy's office". The next task is to draw several ideal routes through the city linking these points in an interesting manner. Ideally these routes will be separated to keep players apart from one another for the majority of the experience. Ideal routes and key points within the game area are drawn onto the map of the game area in Photoshop, with each new section of content added as a new layer.

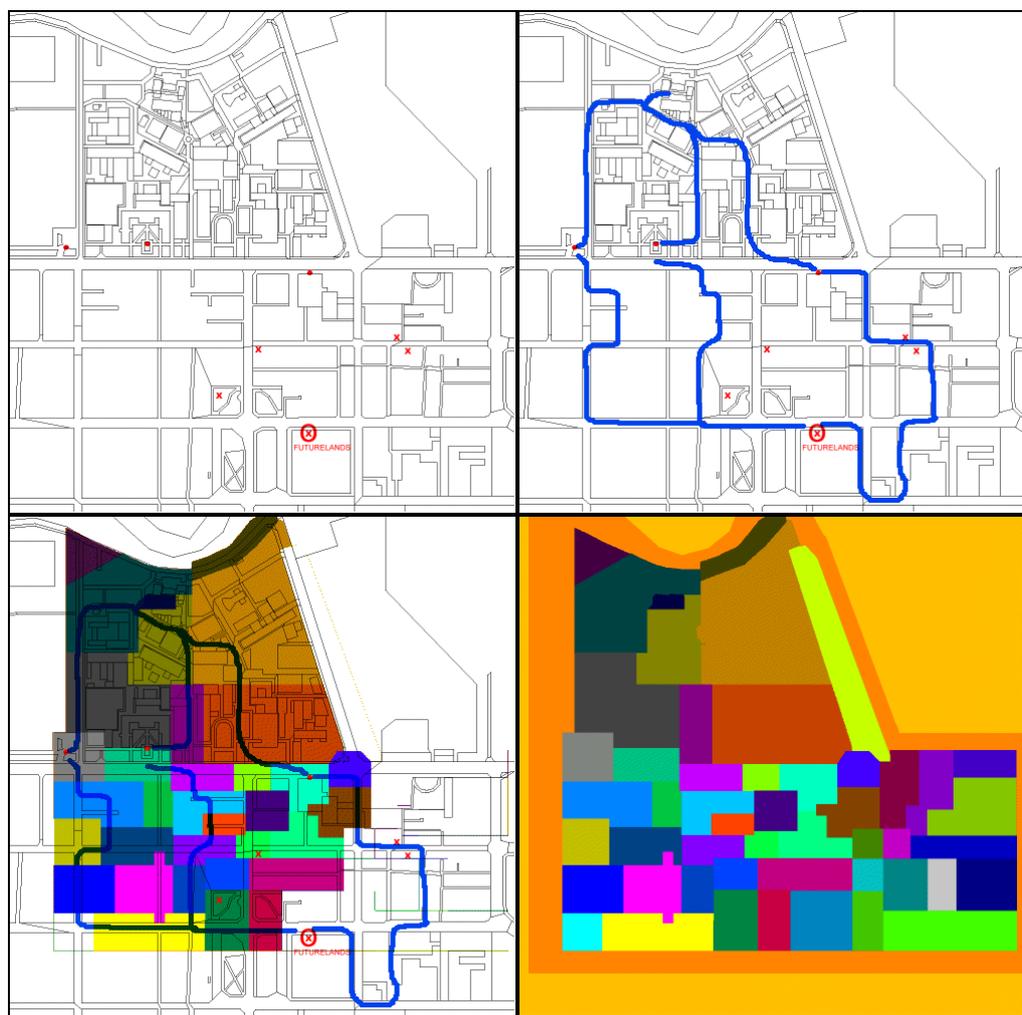


Figure 5-11: Game area and start positions, top left, ideal routes, top right, adding regions, bottom left, finished ColourMap, bottom right

The author can now work on painting the areas with colour that will trigger each clue. As each new colour is defined and used, they associate it with a clue in the XML file. The ideal route is painted with the colour of a clue that will

move the mobile player further along the route, with adjoining areas coloured to push the player back onto the route should they stray. Again, layers and transparency within Photoshop or a similar package allow the author to see other relevant content, such as the underlying map of the physical game area and the ideal routes.

Concentric rings of clues and colour are constructed towards the edges of the game area; as the mobile player moves further away from the desired area of play the clues are structured to become more forceful in driving the player back in the right direction:

“You have come the wrong way. Head towards Pall Mall.”

“The police officer was firm but polite: Not this way, today.”

Figure 5-11 shows the various layers being used during the construction of the ColourMap, in order to show as much useful data as possible. When the author has produced the ColourMap, the guide layers of the map, such as the ideal routes and physical map, are rendered invisible, leaving only the final colours visible. As described previously, Uncle Roy All Around You combined both a start position and a clue trail ColourMap. The start positions in this case defined key content points within the game, such as red-spot markers to be shown to the mobile player. The start positions are authored and stored in the same Photoshop working file as the clue trail ColourMap, but are exported as a separate image just containing the start positions and underlying map layer. This is to avoid leaving “dead spots” on the clue trail ColourMap where a pixel is taken up by a start position.

The ColourMap is then exported as a GIF file for integration into the game system, as shown in figure 5-12 – a ColourMap that was created for Uncle Roy All Around You staged in London. The ColourMap and associated XML file are then transferred onto the handheld computer, or reside on the central game engine. On running the system, the files are loaded and validated. The author can now take the device out into the city and try out the first cut of the new experience.

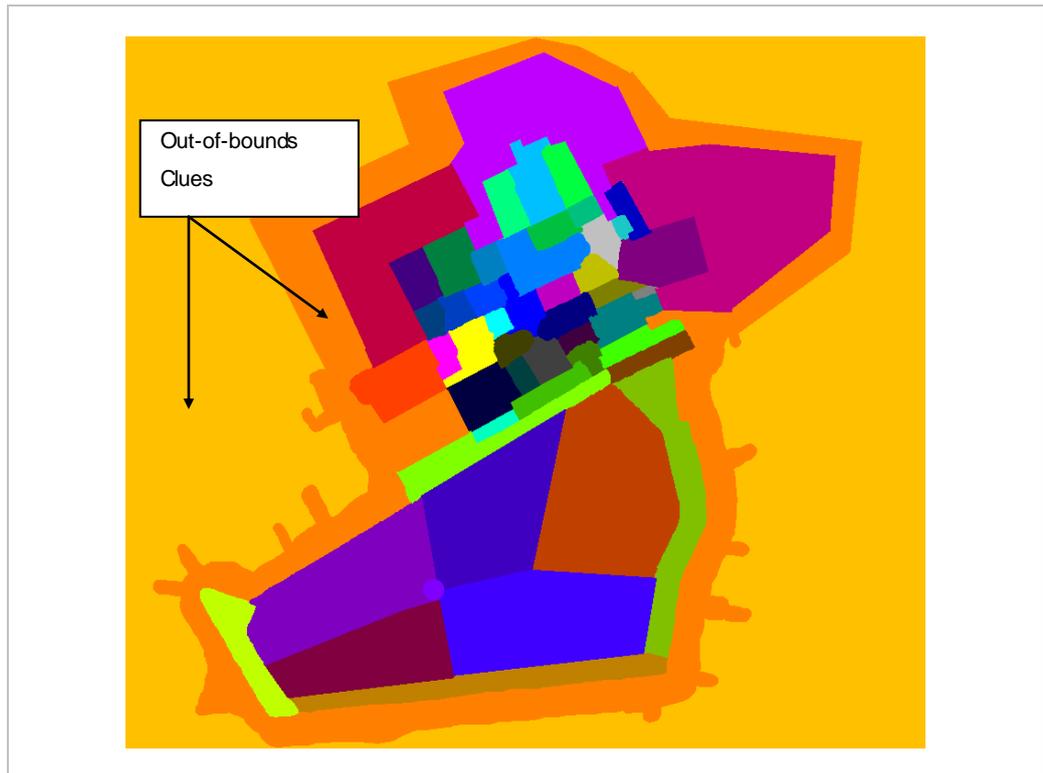


Figure 5-12: A clue trail ColourMap

During the creation of Uncle Roy All Around You and I Like Frank for different cities, it was seen that the construction of a clue trail ColourMap requires up to fifteen iteration, starting with large scale alterations of the shape of the game area and basic clue structure and finishing with the fine tuning of the wording of the clues and their punctuation, and fine alteration of the ColourMap pixels. Test participants are often invited to take part in the experience just before it opens to the public, and their feedback on specific clues or areas results in further changes. Finally, changes are even made on a day-to-day basis during the time that members of the public were taking part in the experience.

The ColourMap system proved valuable during these periods of rapid content iteration. Once the author had a first version of a ColourMap created it could be tweaked, exported and loaded onto a test device within a few minutes as they were already very familiar with the tool used to create it, Photoshop.

5.3.6 Edges and Levels

This section describes how the basic functionality of the ColourMap system described above is extended to enable the creation of more sophisticated

mixed-reality experiences by extending how a player occupies a ColourMap space and triggers content.

Edges define the behaviour of the system if a mobile player reports that their position is outside the borders of the ColourMap image. A number of different behaviours may operate in this situation; the position can be interpreted as undefined, in which case the default clue is returned; the mobile player is moved onto an adjacent ColourMap that has been defined as being next to the current one, allowing large areas to be tiled with multiple ColourMaps; the position can be snapped to the defined edge of the ColourMap, in which case the nearest valid pixel defines the clue that is received thus giving the impression that the borders stretch indefinitely in all directions. This edge snapping behaviour was used in Savannah, where thin colour borders on the edges of the ColourMap represented hazards that extended outwards and that would harm the mobile players – encouraging players to stay within the central game area.

ColourMap levels introduce the concept of a player inhabiting one of several ColourMaps, with each one representing a stage, or level, within the mobile mixed-reality experience. A mobile player can move to the next level by reaching a certain position or by achieving some other experience-specific goal, allowing two mobile players to be in the same physical position but to receive different content or clues as they are at different stages in the experience. This multi-level structure is conceptually based on the idea of a three-dimensional ColourMap space consisting of plateaux, each one representing a level, and each one constructed using the now familiar two dimensional ColourMap. The mobile player can be thought of as starting on level one, the uppermost level in the three-dimensional space. As they move around the level, the clues drive them towards ‘holes’ or specific target areas, which the player falls down to achieve the next level.

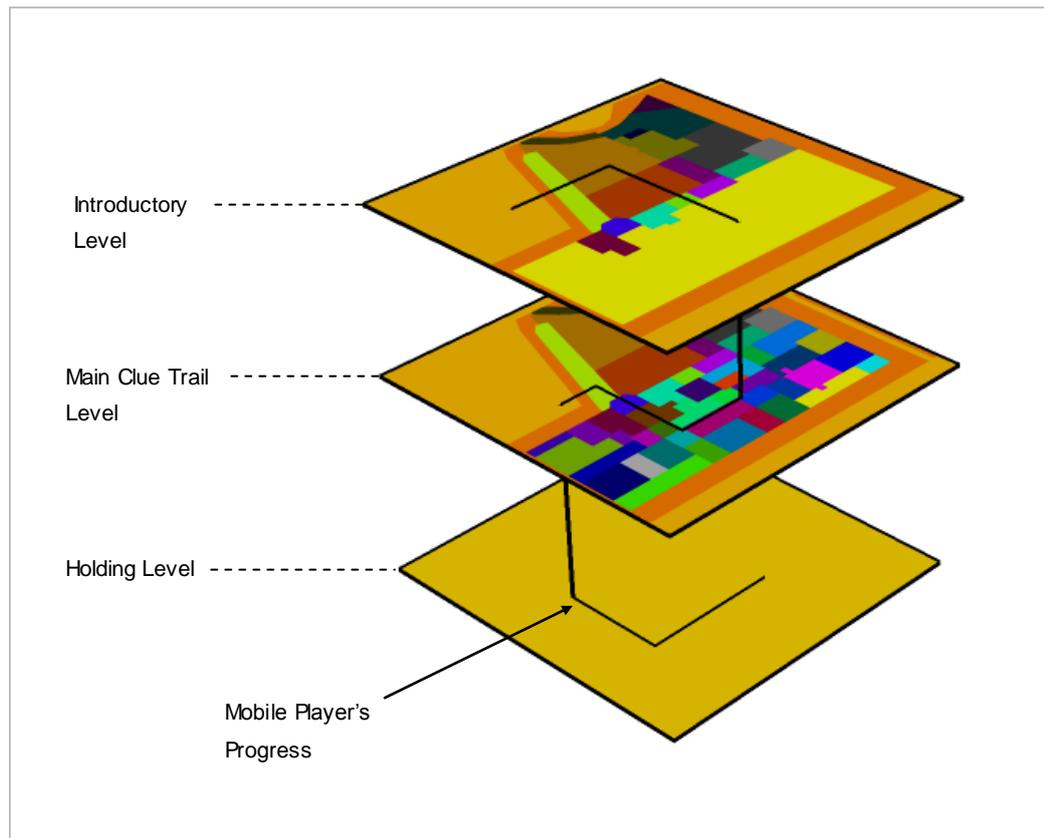


Figure 5-13: A multi-level ColourMap

Levels are used to give I Like Frank three levels, each with its own ColourMap and clue trail, as shown in figure 5-13. The first level is a training level, with the ColourMap consisting of straightforward clues to introduce the mobile player to the basic concepts of the game as they move through the relatively safe environments of a university campus. If they reach the target red spot, they fall through the hole onto the next level, whereas if they move too far into the city, the clues attempt to steer them back on track. The second level takes up the majority of the player's time in the game, involving a more complex clue trail, as in Uncle Roy All Around You. The third and final level contains one colour that triggers simple clues to keep the player waiting at a specific location while the end-game performance is prepared.

Multiple ColourMap levels may be occupied concurrently to provide different content-triggering functionality. In *Can You See Me Now?* a ColourMap was used to provide start positions for online players while a second ColourMap was used to provide GPS filtering. In *Savannah*, however, each mobile player inhabits three ColourMaps simultaneously, with each one representing one of

the senses of sight, sound and smell and triggering the appropriate media accordingly. These three ColourMaps are shown in figure 5-14, showing how, for instance, a mobile player can “smell” and “hear” an elephant on the savannah before they can see it.

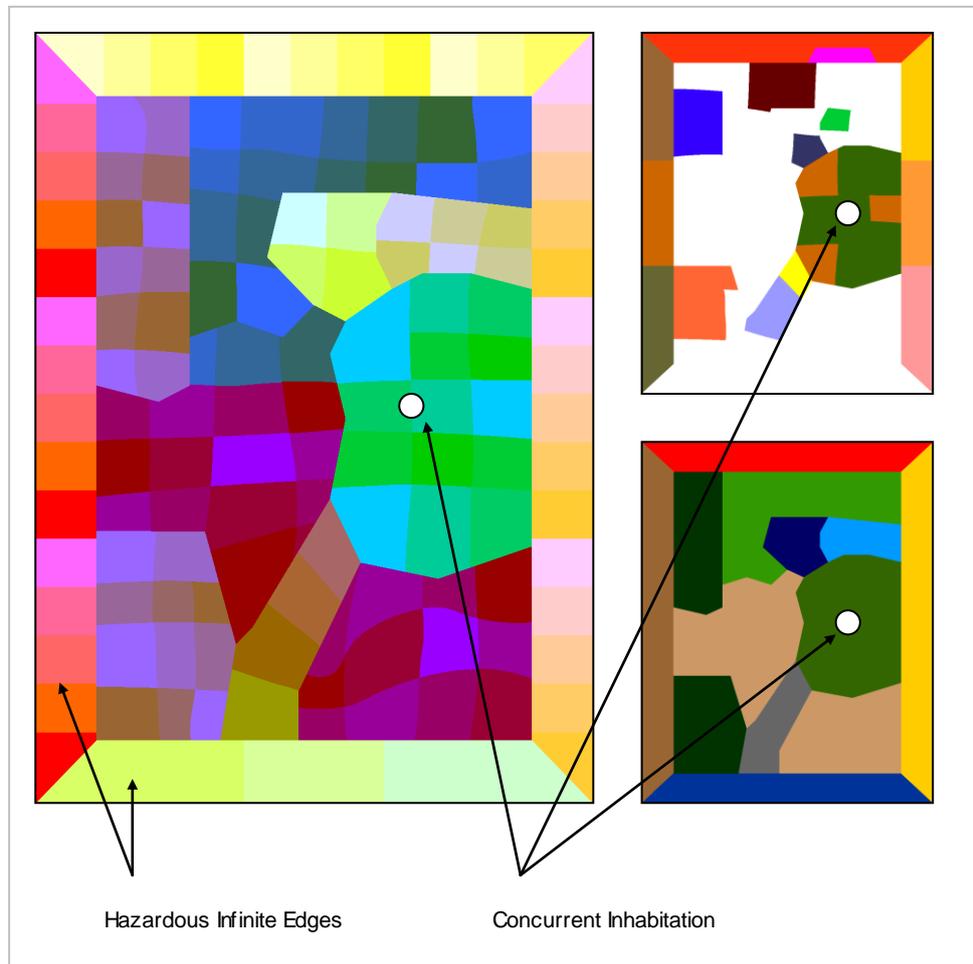


Figure 5-14: Occupying sight, sound and smell ColourMaps in Savannah

5.3.7 Latching and Digging

The final functional extension to the system defines how a mobile player’s movement between coloured regions on the ColourMap triggers a variety of content and clues.

Latching requires that a player leave their current region on the ColourMap and re-enter it before that region is triggered again, namely by moving into a region of a different colour. This behaviour is especially useful when position updates are sent regularly by a GPS receiver, to avoid the same content region being re-

triggered on each update. For this reason latching was used in the Savannah experience, which also used GPS receivers for positioning mobile players.

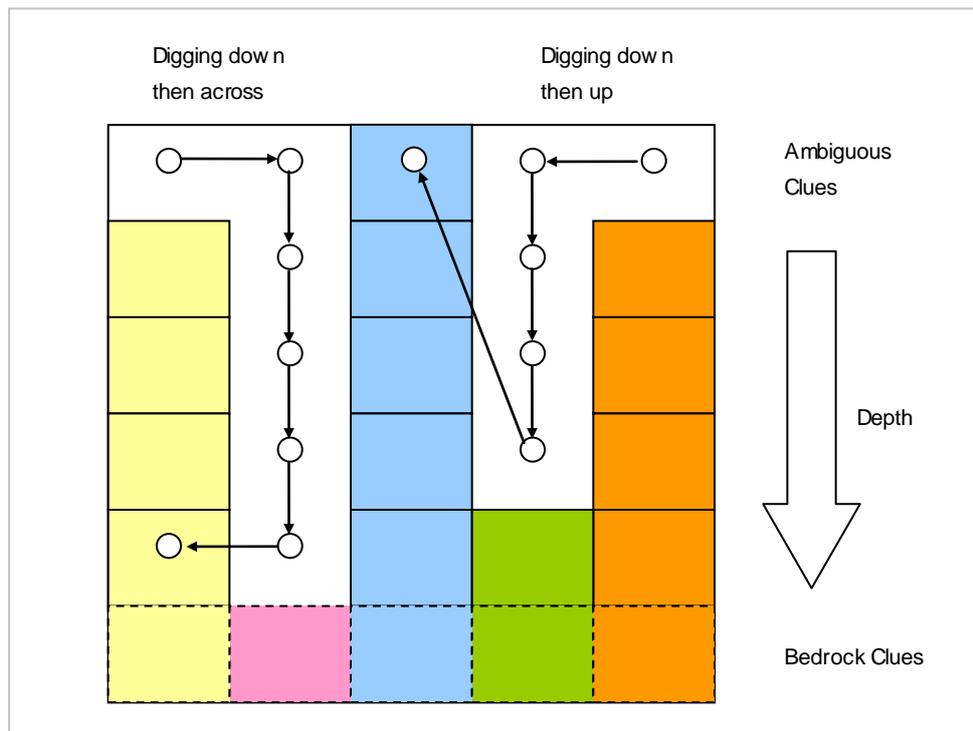


Figure 5-15: Digging in a ColourMap

Digging in ColourMaps introduces the concept that a ColourMap, rather than being a flat-two dimensional image, can be imagined as having a certain thickness, and that multiple clues or content can be associated with each coloured region, which in turn has a certain depth – with each stratum of colour linking to a specific clue.

When a mobile player requests a clue in a coloured region, the system serves the uppermost clue in the stack, and then removes that clue so it is no longer available. When the mobile player next requests a clue in this particular region, they dig deep into the colour strata to find the next clue, as shown in figure 5-15. The last clue can be thought of as bedrock that can not be removed – instead this clue is repeated while the player remains in the region. Once a player has dug down, they can either remain at their current depth when they move into an adjacent colour region, or begin to dig from the topmost layer again.

There are multiple reasons for digging. One concern about Uncle Roy All Around You and I Like Frank was that the voice of the game, or the text clues received, were obviously automated. One of the reasons for this was repetition of clues when a region was revisited. Digging allows authors to create significantly more content for each coloured region. Most notably, digging allows authors to vary the difficulty or ambiguity of clues according to their depth in the ColourMap.

The topmost clue in the stack was often quite ambiguous; encouraging players to follow diversions, drawing on the history of the local environment:

“Wait for someone coming in the opposite direction. Gently turn and follow them.”

Whereas if a player repeatedly requests clues in the same colour region they dig past the more obtuse clues and into clues that become more precise and geographically specific in an attempt to help them to move on:

“Leave the lake behind, they went North towards the Mall. Crossover the road and look for some steps.”

In Uncle Roy All Around You each colour region typically had four or five clues associated with it, with the bottommost clue being very specific, although it was unlikely that the majority of mobile players would dig this far down.

5.3.8 Preparation and Validation Mechanisms

The process of creating ColourMaps involves some stages of preparation and validation as well as authoring.

While the creation of content within Photoshop was quickly realised by the authors, there were early problems with using ColourMaps in an experience. The initial implementation of the software required a particular colour depth of image to be loaded on the device, 8 bits. If the ColourMap had not been created with this colour depth, then the number of colours had to be reduced before export. Without a predefined palette, a significant number of colours would be slightly changed. There was no tolerance in the system for ranges of colours,

each clue being identified with a specific RGB value, so a slight variation would result in the clue lookup failing, and a default clue returned. A similar problem occurred when a soft brush was used in Photoshop to define an area, as shown in figure 5-16. While not immediately visible to the author, the edge of the region, or areas within the region during painting, would have subtle variations in colour. In this case the majority of the area would trigger the correct clue, with small regions triggering the default clue.

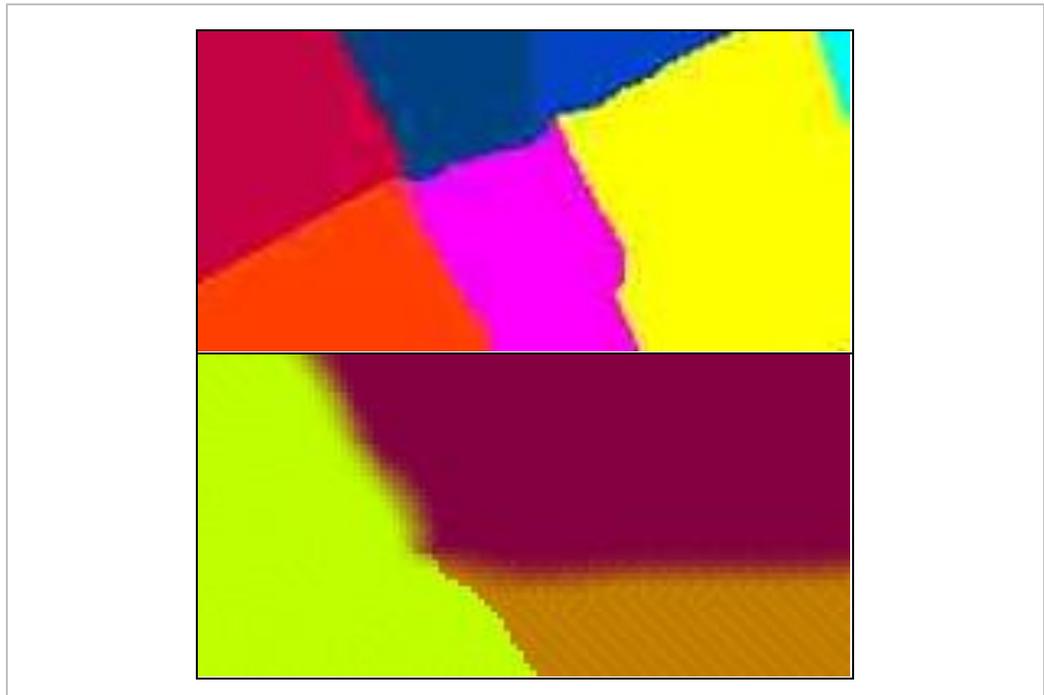


Figure 5-16: ColourMap JPEG artefacts, top, and effects of a soft brush, bottom

Secondly, the concept of loss-less compression had to be impressed on the authors. During the creation process, the ColourMap was exported as a JPEG file, introducing pixels of an incorrect colour as the compression algorithm blurred the edges of regions, again as shown in figure 5-16. These minor problems are one of the disadvantages of using non-dedicated tools, although were overcome after discussions with the authors.



Figure 5-17: Validating a ColourMap

A second issue was in creating the XML file containing clues. This required the author to understand the concept of XML, and to define it by hand. The ColourMap was intended to avoid the designer having to learn and use extra packages of languages, but XML was considered to be the easiest way of adding structured extra data to the ColourMap. It was discovered that the artists knew of no simple way of creating XML on a Macintosh computer – their text editor of choice. Microsoft Word would add hidden or operating system specific characters to the text. A stage of validation using Internet Explorer's XML parser therefore had to be introduced to check the file.

Initially these problems were identified and the authors made aware of how to avoid them, with the errors discovered by trial and error during the testing of the clue trail. However, it soon became apparent that a set of processes to validate the ColourMap and associated content would make the overall process easier.

The validation tool, shown in figure 5-17 emulates the loading of a ColourMap and associated clue XML file with the addition of extended debug output to identify the location of errors which would be difficult to find once the files

were loaded onto a mobile device. The XML file containing the clues was validated by attempting parse it, and illegal characters and missing tags were identified with locations of the errors within the file. The ColourMap is searched for unique colours, and solitary pixels. Each colour is cross referenced with the XML file for the existence of a clue, the colour and location of unassigned pixels is displayed, allowing quick repair, as these pixels are likely to have been created in error.

Finally, a palette of around one hundred colours was created based on a previous ColourMap after a suggestion by one of the authors. When the ColourMap was initially defined, this palette was applied to it, with two advantages. Firstly, the designer did not have to define unique colours; they can just select the next one to use from the palette. Secondly, if the colour depth of the image has to be changed, Photoshop can be forced to maintain this palette rather than arbitrarily modifying colours in the image.

5.4 Summary of ColourMap Features

This section now presents a summary of the key design features and innovations of the ColourMaps system, combining the lessons learnt from its integration and use in supporting our four experiences.

The ColourMap system supports the core requirements of the authoring process of a mobile mixed-reality experience, run-time content provision and aspects of the operation and orchestration process, as shown in figure 5-18. Familiar and generic image colouring packages are combined with text content, and used to create ColourMaps and associated data files to be loaded into the system. Validation tools, such as those mentioned earlier, are used to debug and validate the ColourMaps. Finally, ColourMaps are loaded into a run-time system for testing and to serve content to both mobile and online players in the mobile mixed reality experience, while also providing a visual overview of content for orchestration interfaces.

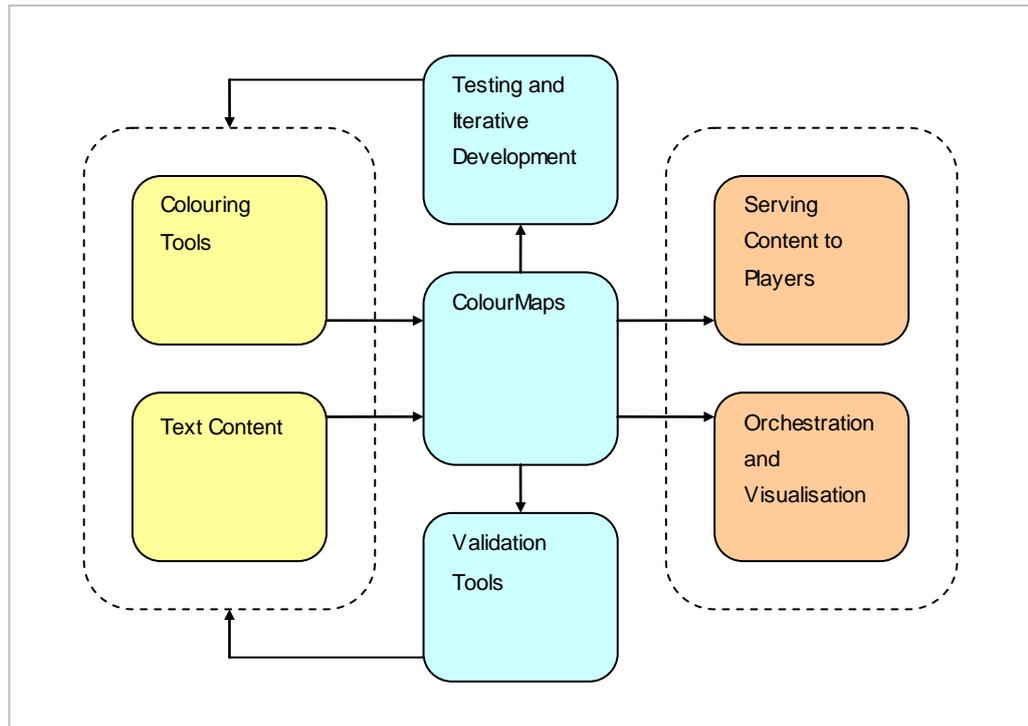


Figure 5-18: ColourMaps in context

Figure 5-19 summarises the key components of the ColourMaps system. An experience consists of **multiple levels**; several ColourMaps that are linked logically within the mobile mixed-reality experience to create a content space. Levels are accessed sequentially, as in I Like Frank, or concurrently to serve a variety of content, as in Savannah. The construction of each level involves **multiple layers**, each of which supports a different function:

Physical map: this layer is a geographical map or aerial photograph of the physical area in which the experience will be played. It is used as a guide when drawing content regions, although as demonstrated in Savannah may not be required. It is not used in the run-time system, though may be left on the image for use in orchestration interfaces and reviewing the content without interfering with other functions.

Start positions: this layer contains individual pixels that determine where specific entities are placed in the experience, for example where online players enter the game, or markers for mobile players. Assuming that start positions are selected randomly, an area can be weighted to become more likely by adding more pixels. This layer is exported to the run-time system.

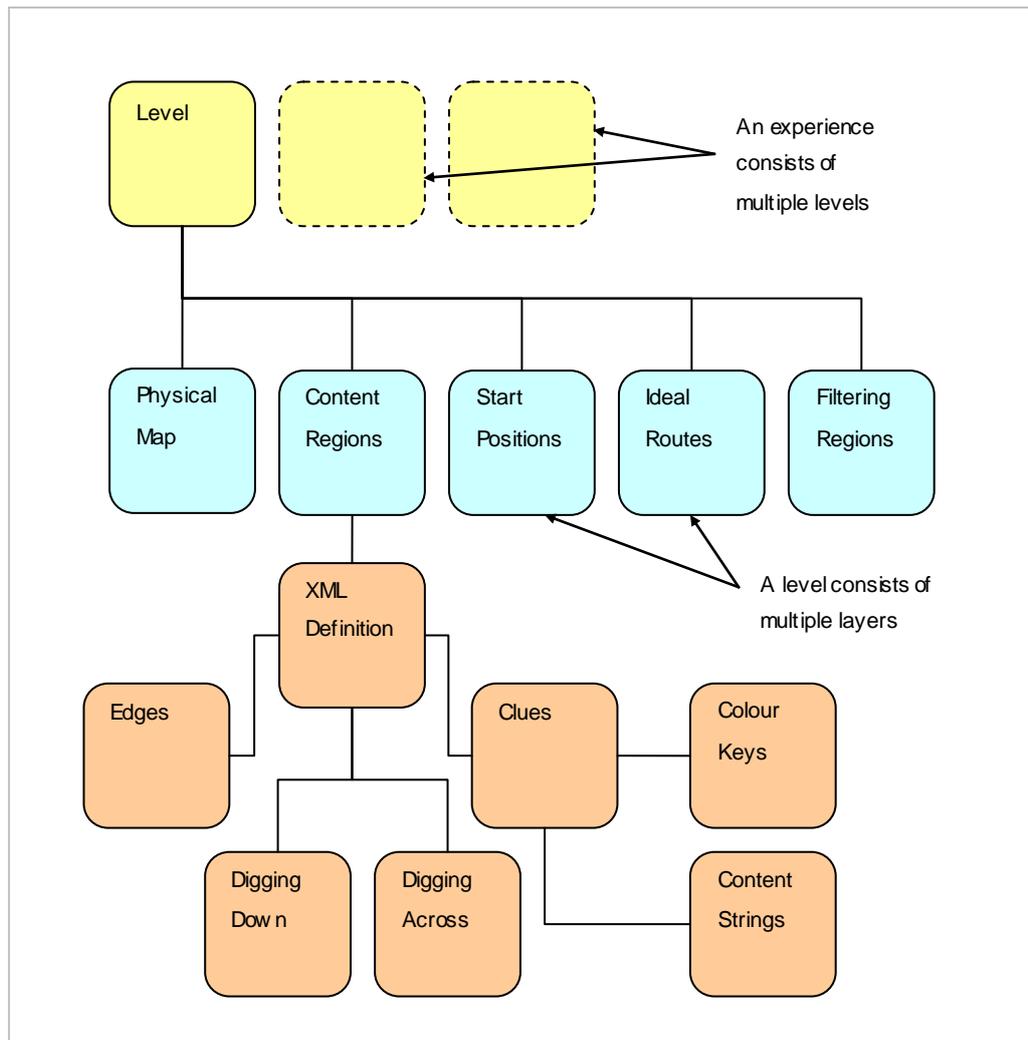


Figure 5-19: Elements of the ColourMaps system

Ideal routes: this layer is a visual guide used by the author to determine the best route through the physical space, for example several trails through the city in Uncle Roy All Around You and I Like Frank. Surrounding coloured regions should drive the player towards one of these routes. These are not exported to the runtime system, but may usefully be revealed on an image used for orchestration purposes, which will be discussed in the next chapter.

Filtering regions: the coloured regions in this layer are used for filtering positional data, for example correcting GPS readings in Can You See Me Now?. They are drawn over the physical map colouring in buildings and areas of water that mobile players cannot enter. This layer is exported to the run-time system which applies a position correction algorithm, for example mapping any point within a forbidden region to the nearest point outside of it.

Content regions: the coloured regions in this layer define the core content of the experience, that is the text (Uncle Roy All Around You and I Like Frank), images and sounds (Savannah) and possibly other content that is associated with different locations. If levels are occupied concurrently, then several content layers can be active at the same time, triggering different media, as we saw in Savannah.

The content layer defines behaviours that define how a mobile player triggers content while moving through the ColourMap space:

- **Edge behaviour:** A rule that defines how the ColourMaps system reacts when a player moves outside the defined game area. Their position can be snapped to the edge of the map, as seen in Savannah, moved to a different level if one exists, or considered to be undefined and return a default clue.
- **Latching behaviour:** A rule that defines what happens when a player moves between adjacent colour regions or remains in the same region. They can either trigger the region for each update, or the region can be latched until they move to a different region and back again.
- **Digging behaviour:** A rule that defines the content triggered when a mobile player repeatedly requests content for the same colour region when latching is disabled. The author may choose between digging down then stopping at the bottommost content, or digging down and returning to the top when the player moves to an adjacent colour region. Authors define the associated content to become simpler as the player digs down to help them move on.

The content region layer consists of a set of clues that are associated with the coloured content regions of the ColourMap. Clues consist of:

- **Colour keys:** Each clue contains an RGB value that links it to a particular colour of pixel on the ColourMap image.

- **Content strings:** The content to be triggered in the form of text, or media filename. There may be several if digging is to be used.
- **Default clue:** The content region layer may have a default clue that is triggered if no content is associated with the colour region the mobile player occupies, or if they move outside of the defined game area.

5.5 Discussion

ColourMaps have proved to be a useful tool for authors in creating and configuring content for mixed-reality experiences. We have seen how the ColourMaps system has been successfully deployed as the core content support system for our four experiences over the course of numerous iterations and public deployments.

5.5.1 Feedback

As described in chapter 1, the ColourMaps system is the result of an intensive and iterative development process “in the wild”, or adapting the system to meet the real-time requirements of creating our four experiences through constant testing, rapid feedback and incremental changes. The ColourMaps system was designed from experience, rather than the conventional software engineering process of design then implementation. As such, the ColourMaps system has been continuously evaluated during the construction of the four experiences, with this chapter describing the most recent iteration of the tool. This continuous evaluation has taken place over the course of several years and a total of approximately fifteen public outings of the four experiences. However, feedback from some of the authors who used the system is interesting to examine as it highlights some of the early problems they had with it, leading to a number of the design innovations described:

“I found making the colourmaps for Savannah quite straight forward. I used Adobe Illustrator to make them, then exported with anti-aliasing turned off (I didn't at first - that's the graphic designer in me - until I realised that different hues of the colours was a bad thing - duh...).”

In the feedback above two authors describe one of the problems described previously that led to the creation of the validation tool.

“I think colourmaps are a very practical and easy mechanism for creating content. They're logical and anybody can understand them - which means that everybody can be involved in their development. I'm not sure the process could be much easier for me than it was - perhaps a predefined palette of colours within Illustrator specifically optimised for colourmap use.”

“- a comprehensive checker: are RGB values correct? is XML correct? The ultimate checker would have a search type facility: show me any colour that has less than two clues etc”

Similarly, this feedback led to the creation of a pre-defined palette of colours to make the process of creating ColourMaps less prone to errors. A different author describes how they understood the system to work:

“In Uncle Roy All Around You we used colourmaps to grid the map of the chosen area (West Bromwich). To create segments, dividing up the larger area into smaller sections. A clue trail for the game area was created alongside the colourmap, where the two work hand in hand. The clues are given colour codes positioning them to their exact position on the map. ”

And also the process of making and testing revisions, and initial problems:

“About 10 revisions, as it was tested repeatedly and amendments were always needed. [...] Overlap of colours, whereby clues would blurred or be missing. We had errors where we had used the same colour twice resulting in mixed messages to the participants. How could the process be made easier? Understanding its importance and difficulties (or problems that may arise when creating a colourmap) so that

when in the process you can try and avoid making the mistakes that results in more amendments to the map.”

Finally, an author describes their process of constructing the clue trail using ideal routes and concentric rings of clues:

“The first step was to walk through the game area with a paper map dividing the area into small sections in pencil. I marked the key decision points that anyone navigating through the area would use such as junctions and turnings. Then I sized areas according to their relative importance and the speed with which we intended players to travel through them. Larger areas were of less importance. As players neared their destination the areas become increasingly small to give more and more precise clues to allow them to find a specific location in the city. Then I filled in areas in between the important areas to ensure that wherever a player clicked in the game area they would receive a clue. Finally I bounded the game area with two concentric rings: one to give a warning that a player was heading in the wrong direction, the second to instruct a player to turn around and head back into the game zone.”

However, the same author found the process of creating the XML file difficult and the most time consuming part of the process:

“The relationship between the XML file and the colour map is too remote. I had to have them side by side and cross refer very often. But because their primary function is to allow artists and/or non technical people to use them then a clean, intuitive interface that takes you through each stage would transform the technique”

5.5.2 Limitations

The most obvious criticism of the system is that the ColourMap image and the associated XML definitions of the content are too far removed, in that while the authoring paradigm of painting content proved to be very powerful, the

authors struggled somewhat with creating the relevant linkage to the XML file. A useful addition, therefore, would be to create a plug-in for an image editing package such as Photoshop that automated this file generation, for example generating the colour keys automatically from the palette of colours used within the image.

In our four experiences there was a one-to-one mapping between colours and the content triggered. A more subtly created image could support more sophisticated or fuzzy triggering by, for example, averaging the colour values in a given area to produce a weighted trigger. This would be useful if a ColourMap dynamically depicted changing environmental conditions, such as variations in GPS or wireless network coverage – environmental features that are sometimes visually displayed on a map in other applications – for example contour maps built from large set of spatially distributed scalar data, often found in GIS.

Finally, the two modes currently implemented allow digging through repeated triggering of the same region, and latching so that nothing occurs until a different region is entered. It may be useful to adopt a more generic event model for triggering the content, for example by including enter, exit and time-based events when travelling through a region, as suggested by the Mediascapes tool.

5.5.3 Broader Reflections

As the core authoring and content system for our four experiences, the ColourMaps system has enabled their deployment to large numbers of members of the public in a variety of cities. By working directly with a number of authors, it has been shown that the approach is sufficiently simple and familiar, and yet flexible and powerful enough to be able to support these experiences, and this has helped to refine and generalise the approach. There are some important lessons to be learnt from the development of the ColourMaps system, and this section concludes by identifying some common solutions that can be applied to tools with similar requirements.

Raster AND Vector: Many conventional game-engine design tools focus on vector-based authoring, for example supporting the placement and Boolean modelling of regular three-dimensional shapes in a modelling package, or the placement of tiles. GIS support storage of both raster and vector data, yet the majority of authoring is concerned with vector visualisations of data sets. By using a drawing package such as Photoshop, rapid prototyping can be first done using vectors, or primitive shapes, and then converted to a raster layer for fine tuning of shapes to fit specific terrain, for example using raster brush tools. Mixed-reality experiences are often deployed in complex urban spaces, and should therefore support arbitrary content region shapes without resorting to complex modelling.

Design for authors, not technicians: Any tool for designing content needs to be easily used by the end user, the artists and authors. In experiences such as Uncle Roy All Around You, I Like Frank and Savannah, the narrative and flow of the experience is entirely dependent on the nuances of the content. It is important to be able to cut out the technical team and allow the author to work directly with the content, using a tool that they are confident with. It has been an important feature of creating our four experiences that the authors can use tools that, to them, are readily available and familiar.

Human Readable Formats: It is accepted that there is scope for further implementation work in automating more of the authoring process of ColourMaps, such as creating the XML data file; however, it is clear that a visual medium such as painting maps provides a rich paradigm when laying out location-based content in this way. While a three-dimensional format might give more scope for elaborate and sophisticated triggers, choosing a medium that can be rapidly viewed and internalised by the author is important, especially considering the fact that the author may not be a computer scientist.

Rapid Configuration and Loading: The content should be easily deployable within the system to allow for iterative development and testing by the author, rather than by the technical team. As has been shown, designing a mixed-reality experience from scratch requires many iterations and refinements; over

a limited timescale it is vital that the author can begin work as soon and as efficiently as possible.

Flexibility for multiple uses: we have seen that the ColourMaps system has been used for a wide variety of purposes within our four experiences; defining location-based content triggers for different media, GPS filtering, and setting start positions. Ultimately, this flexibility requires generalising ColourMaps to include a multi-level, multi-layered structure where different layers can be associated with different design choices (e.g., digging, latching and edge handling mechanisms). At the same time, it is important to maintain conceptual simplicity; otherwise the original purpose will be lost. The intention is that basic single-level and single-layer ColourMaps can be used for creating simple experiences, while appropriate metaphors such as levels, layers and digging gradually introduce more complex functionality in a way that is consistent with the overall approach.

5.6 Conclusions

This chapter has investigated how to support the authoring and content requirements of mobile mixed-reality experiences as suggested in chapter 4, and has shown how these challenges have been overcome during the development of our four mixed-reality experiences.

This chapter has reviewed a selection of other related tools, from commercial GIS and game-engine design tools, to research-based context-aware systems and tools to support mobile and location-based applications, highlighting the strengths, weaknesses and novel features of each.

In response to this it has described the development of the ColourMaps system, a tool designed specifically to support the authoring and content requirements of mobile mixed-reality experiences, and giving examples of how each key feature of the ColourMaps system has been successfully used to support a critical aspect within our four experiences. It has shown how the ColourMaps system has supported collaborating artists in creating a rich narrative landscape for players to explore.

Finally, this chapter has summarised the unique innovations of ColourMaps, giving an overview of the complete system. It has reflected on how the authoring tools should exploit an author's knowledge of familiar tools to cut out the technical team, and introduce support for rapid configuration and loading to increase productivity, alongside flexibility to support a wide range of functionality and types of content.

6 Tools to Support Orchestration

6.1 Introduction

Chapter 4 argued that orchestration is an essential aspect of mobile mixed-reality experiences in that it provides the ability to mix pre-authored content with dynamic improvisation, and live performance with interactive play, and that this requires a general solution. This chapter argues that it is not sufficient to merely allow players to take part in an experience, and that support for orchestration should be a primary concern for experience creators. It demonstrates the development and use of a general orchestration framework that allows the various elements of an experience to function together as a coherent production.

This chapter begins by examining elements of orchestration from a number of previous projects, in order to show how the orchestration of an experience by authors, performers, and sometimes participants themselves plays a critical role in its smooth and ongoing operation. It highlights key requirements, techniques and practices, but this also raises the question of what tools are required to orchestrate a mobile mixed-reality experience.

In response to this, this chapter presents a framework of tools dedicated to supporting the orchestration of mobile mixed-reality experiences, based on the practical, real-world deployment of the four experiences described in chapter 3. These tools demonstrate the large number of orchestration functions that need to be supported in the day-to-day operation of an experience.

Finally, this chapter summarises the key functions of the orchestration framework, and reflects on how it can influence future mobile mixed-reality experiences.

6.2 Fundamentals of Orchestration

(Laurel, 1991) proposes that computers should be considered to be a form of theatre, rather than tools for a particular application. She also argues that the focus of design should be on engaging users with content, rather than technology, and that behind-the-scenes activities are required, as in theatrical

productions, to manage and maintain participants' engagement with an experience. We can define orchestration as the art of guiding and shaping an experience as it unfolds, to ensure smooth operation and enjoyable, engaging and constant participation.

With this in mind, this section describes how orchestration plays a key role in experiences and activities within a wide range of disciplines, from the dedicated monitoring of automated systems and behind-the-scenes management of performances, to player-managed orchestration in role-playing games through improvisation. The common factor in all of these examples is that orchestration needs to shape participation to adapt to shifting or unexpected needs, or to mould it on a case-by-case basis to ensure the most engaging experience. Orchestration may take a variety of forms; perhaps playing a fundamental role in the experience through improvisation or interpretation of a loose set of rules; ensuring that time critical events occur on time through monitoring and marshalling; moderation of a public environment through responding to shifting player behaviour and unexpected events.

6.2.1 Role-Playing and Improvisation

Improvisation is a fundamental aspect of many varieties of role-playing game – table-top, face-to-face and to some extent modern online variations – in order for players to shape their own participation to allow for greater engagement.

Traditional table-top role-playing games, involving cards, figures or purely vocal in nature, involve a *game master* who orchestrates the game by describing the events that happen within the narrative of the game, based on the actions of each player. (Fine, 2002) describes the game master as follows: *He is God in that he creates the world in which his players must survive; he maintains ultimate interpretive authority.* The game master interprets the game rules, and acts as an arbitrator between the rules and the players. Importantly, the game rules only form the basis of the experience for the players, which are then interpreted and extended by the game master in order to create the experience proper; they improvise descriptions of what is happening to the players within the space loosely defined by the rules. In a table-top role-playing game, this improvisation and operation is performed vocally by the

game master – while in other disciplines it may be a behind-the-scenes action that is made visible in a different manner.

Live-action role-playing, or LARP, involves players acting out their actions within the game in the physical realm, and is now often the focus of mobile mixed-reality experiences (Jonsson, Montola et al., 2006). Role-playing in this form can be thought of as being similar to improvisational theatre – players act out their characters vocally and physically in response to a loosely defined script. As with conventional role-playing, a game master enables the game through the improvised interpretation of a narrative and basic rule-set, however players themselves are encouraged to remain in character as much as possible for a prolonged period of time, sometimes up to several days. During this time players continually interpret and improvise their own roles, acting out many concurrent scenes as part of a distributed narrative, in order to maximise all players' engagement with the game.

6.2.2 MUDs and MMORPGs

MUDs, or Multi-User Dungeons, Domains or Dimensions, are *network-accessible, multi-participant, user-extensible virtual realities whose user interface is entirely textual* (Curtis, 1997). MUDs combine elements of role-playing and social chat in a shared environment of rooms, objects and non-player-characters. MUDs are largely automated, with descriptions and events being generated for players based on an underlying system of rules and content database. As such, responses to actions such as arriving in a new room, or using an object, are determined by the system – in this case the system acts as the game master.

MUDs allow certain users to be raised to a privileged administrator role, known as a *wizard* or an *immortal*. Wizards are responsible for the management or orchestration of the system, administrating by creating and enforcing rules through moderation and potentially punishment. Wizards can manipulate the database and system that the MUD operates on, by creating new rooms, objects and behaviours, and placing them in the game for players to interact with. They can also add new system capabilities and rules, and make bug fixes. A newer variation of the MUD is the MOO, or MUD Object

Orientated, in which wizards perform object-orientated programming on the server itself whilst in-game, in order to add new functionality or content.

Wizards in a MUD also have the power to monitor and manipulate game-play at the player level – they can change the scores and attributes of individual players, and destroy or banish players from the system completely. Monitoring powers allow wizards to see what players type and what they see, as well as to become invisible to normal players. As (Muramatsu and Ackerman, 1998) note, normal players are free to act and interact as they please, but only within the bounds specified by the wizards and the system.

More recent online role-playing games such as World of Warcraft (Blizzard Entertainment, 2004), commonly known as MMORPGs or Massively Multiplayer Online Role-Playing Games, similarly employ a number of privileged game masters to monitor and moderate player behaviour. These game masters have access to a historical record of player chat, and again can make themselves visible or invisible in order to discreetly monitor player actions. Game masters monitor the game for signs of cheating or *griefing* – deliberately disrupting the game for other players (Foo and Koivisto, 2004) – and can respond by muting or restricting players. They can also manipulate certain aspects of the game as necessary, for example the giving or removing of objects, although they cannot add completely new content, as with MUDs.

6.2.3 Static Mixed-Reality Performances

Public performances and static mixed-reality installations often rely on dedicated behind-the-scenes production crews to manage and orchestrate an event. This approach is directly derived from more traditional media such as television and theatre, in which the crew are directly responsible for operating, monitoring and subsequent intervention in the production.

Out of this World (Greenhalgh et al., 1999) was a performance in which professional actors and members of the public took part in a game-show within a virtual environment, which was then broadcast live to an audience in a theatre. The authors describe the key technical innovation of Out of this World to be the development of dedicated orchestration tools for the production,

which allowed the control of various aspects of the virtual environment, including scene and avatar changes, and the control of the virtual cameras that created the view seen by the audience. The tools were used by behind-the-scenes staff to forcibly move participants to key positions in the virtual environment, to ensure that the show followed a tightly defined schedule.

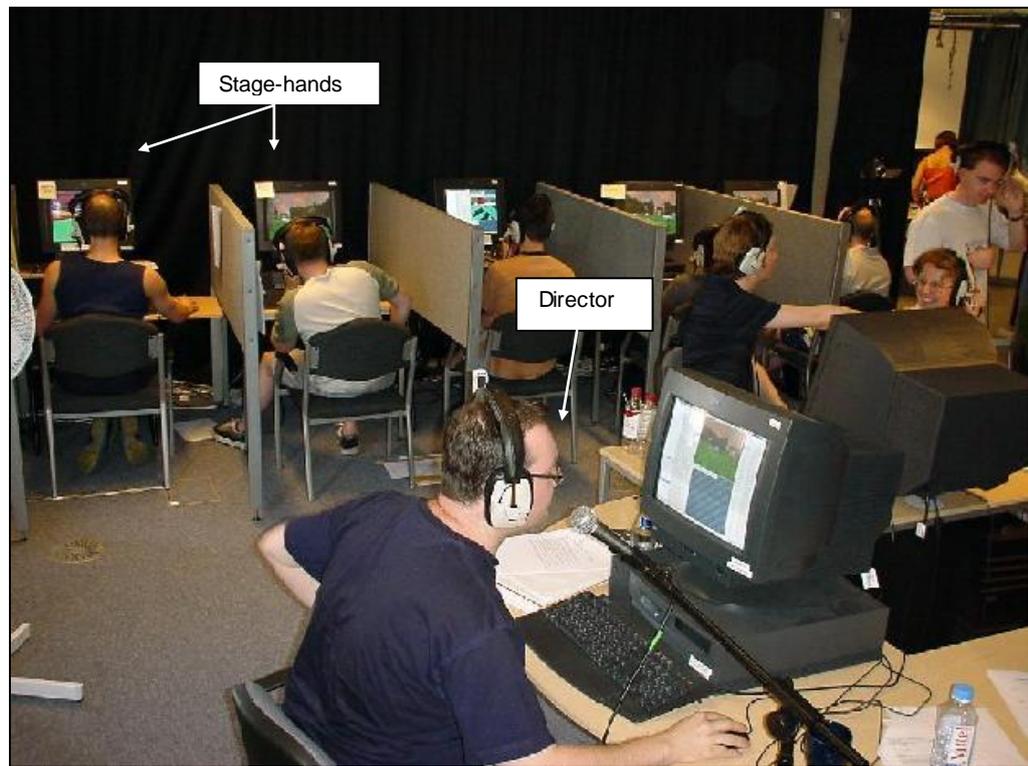


Figure 6-1: Orchestrating Avatar Farm

Avatar Farm (Drozd et al., 2001) was a similar example of a mixed-reality performance, that this time was constructed around an improvised drama within a virtual environment and then broadcast, live, onto the web; involving four members of the public, seven professional actors and an extensive behind-the-scenes production crew. The virtual environment was made up of four distinct, concurrently running virtual worlds inhabited by the players, who engaged in a partially scripted, partially improvised story that lasted for two hours. The story made extensive use of props and objects placed in the virtual environment as part of its narrative, and pre-recorded scenes were injected into the environment in a relatively complex temporal structure. The resulting story was then filmed from within the virtual environment by multiple virtual cameras, mixed and broadcast to the spectators viewing on the web.

As with *Out of this World*, *Avatar Farm* employed a number of behind-the-scenes orchestration tools to allow the production team to act as stage-hands by manipulating the virtual environment in order to create aspects of the performance, shown in figure 6-1. Invisible to the viewing public, the production team opened and closed portals between worlds and arranged props and other in-game entities into the correct positions in response to instructions from the director, rather than relying on a large amount of pre-scripted content or automated behaviours. Other members of the production team controlled the invisible virtual cameras that are mixed into the live feed.

The production of *Avatar Farm* was heavily *socially* rather than *technologically* mediated, in that the production crew and the director used their orchestration tools to create the unfolding narrative, which as an improvised story was not automatically maintained. Similarly, the professional performers in the virtual environment used improvised narrative in the context of the game, and in character, to explain away errors and mistakes on the part of the stage-hands, or unexpected occurrences. The performers also used these techniques to shepherd players into particular places for the ongoing story by using delaying or hurrying tactics, to which the production crew also had to respond. This required both the production crew and the performers to quickly react to unplanned or novel events in the performance, while still maintaining the engagement of the players and viewers. To this end, the supporting orchestration tools were required to not be restrictive; instead allowing scope for wide agency and improvisation.

Desert Rain was a touring mixed-reality performance and installation that mixed a virtual environment with live performers (Koleva, Taylor et al., 2001). Six public players spent twenty minutes searching for human targets in the virtual environment - played by performers that step through a water curtain upon which the virtual environment was projected on in order to interact with the players. At the end of the twenty minutes, the players moved to a final physical room where the identities of their targets were revealed. As with *Out of this World* and *Avatar Farm*, *Desert Rain* relied on a behind-the-scenes production crew, as well as front-of-house performers, to manage the performance as much as it relied upon a complex technical architecture. A

steady throughput of players through the experience was maintained by making significant use of orchestration tools and efficient and constant management.



Figure 6-2: Orchestrating Desert Rain behind-the-scenes

Desert Rain highlights a number of key instances in which interaction with players was meticulously pre-planned and rehearsed, in order to be able to set and maintain expectations of what was happening during the performance – the initial introduction to the experience, face-to-face interactions with performers through the water curtain, and being marshalled into the final room. Each of these instances was carefully managed to build a back-story to the performance, and to ensure that the correct timing of each aspect of the performance was met.

Figure 6-2 shows the Desert Rain control room which, as with the previous performances, consisted of a number of dedicated orchestration tools. Players were monitored constantly; both through their actions in the physical space via video cameras and their movements and communication in the virtual environment. The production team had a number of strategies with which to respond to situations in which a player was not progressing through the experience or having difficulties – interventions that were performed in such a

way as to maintain engagement. Players were given direct instructions via an audio link; invisible interventions involved a stage-hand overriding a player's movements without their knowledge if they were stuck; face-to-face interactions were used as a last resort to give practical advice to players. Expected problems and strategies to resolve them were planned and rehearsed by the production team beforehand.

6.2.4 Control Rooms

As a final note, control rooms provide many examples of practical orchestration, and form a central part of the operation of many disciplines in the way they support operators in monitoring and day-to-day work activities, as well as responding to unexpected occurrences. As we have seen, live mixed-reality performances as well as more conventional television and theatre productions use behind-the-scenes control room technologies to manipulate the stage according to unfolding events or to a pre-defined schedule. Control rooms are found in tube systems, traffic monitoring systems, air traffic control and the handling of emergency calls, amongst others. Although the specific details and requirements of these control rooms vary greatly depending on the particular application, they all support centralised orchestration systems for monitoring, intervention and trouble-shooting.

Control rooms have often been studied in an attempt to understand the collaborative nature of human-computer interaction and computer supported cooperative work. However, studies have revealed that control rooms are only one part of a wider division of labour that includes supporting work that occurs outside of the control room, potentially in a mobile situation – (Luff and Heath, 1998) assert that understanding this mobility is critical to understanding collaboration in work. Similarly, (Suchman, 1995) proposes that while control rooms play a key role in the operation and orchestration of systems, there is also a need to develop a decentralised perspective that takes into account the perspective of mobile members of the team, and that the control room should not necessarily be considered to be the authoritative point of view.

6.2.5 Summary

To summarise, this section has reviewed a number of projects that highlight fundamental principles of orchestration, and described how they are used and often required in order to operate particular applications.

Examining role-playing, we have seen how improvisation plays a key role in keeping an experience engaging and interesting, by allowing the real-time interpretation and extension of a set of game rules by both the game-master and players themselves. The rules of the game are not rigidly enforced; instead they provide a framework that participants may use and adapt as they wish.

MUDs and other online role-playing environments suggest that players' abilities need to be tempered – that they should not be given complete freedom to be able to do what they want in the environment, and to completely define their own game. This is enforced by administrators who invisibly monitor and moderate the game, while also extending and refining it.

Static mixed-reality performances rely on an extensive behind-the-scenes production effort, as with conventional television and theatre. Dedicated tools allow the production crew to manipulate the unfolding action as it happens, and this practice may form a substantial part of the experience. This is combined with planned and rehearsed interactions in order to maintain player engagement and expectations throughout.

Finally, we have seen how control rooms provide centralised orchestration in a variety of disciplines. However, a control room forms only one part of collaborative orchestration work – orchestration should also take into account the needs of mobile operatives outside of the control room.

This section has described practices from a variety of, arguably, established disciplines that can be thought of as *orchestration* – the applications presented fundamentally rely on these practices as part of their day-to-day operation. They raise the question of how orchestration should be supported in the emerging discipline of *mobile* mixed-reality experiences, which adopts many aspects of these disciplines; are there any existing techniques that should be

transferred or adopted, and what new orchestration tools and practices are needed?

6.3 A Framework of Orchestration Tools

Chapter 4 proposed that mobile mixed-reality experiences, especially those that cross into the domain of public performance, are a combination of pre-authored and improvised content, and performance and game play. Orchestration is therefore required to support these dynamic aspects of experiences. Similarly, the previous section has shown how orchestration is an important part of many applications. Rather than leaving participants to fend for themselves, and assuming that systems are operating correctly, dedicated tools and practices are required to support orchestration; to both operate and troubleshoot an application, while also enabling rich and engaging content.

This chapter now presents a framework of orchestration tools that specifically targets mobile mixed-reality experiences. It draws out a comprehensive list of orchestration functionality, and shows how this can be supported within an experience. As defined in chapter 4, the framework views orchestration as one of the fundamental building blocks of a mobile mixed-reality experience, rather than as a supporting tool or add-on – it shows how orchestration is used within and alongside the automated system and pre-authored content. It treats our mobile mixed-reality experiences as dynamic performances that require continuous and extensive behind-the-scenes production effort in order to operate smoothly and professionally. As with the ColourMaps system described in the previous chapter, the orchestration framework evolved iteratively over through its use as a fundamental part of the production of our four experiences, and these are drawn upon to provide real-world examples of the concepts presented within it.

The orchestration framework supports a variety of orchestration functions that are used by the production crew both behind-the-scenes and for face-to-face interactions with players, via several independent software tools and supporting practices. It aims to manage each player's experience from beginning to end, whilst aiming for maximum engagement with minimal disruption. This includes managing and monitoring players and core systems, responding to key

events that will occur during the experience both as part of its mundane, planned operation and otherwise, and intervening both subtly and grossly. The framework aims to support the following high-level orchestration functions:

- **Monitoring:** allowing the production crew to review what players are doing, what they have been doing in the past, and what they are likely to be doing next. This provides confirmation that the experience is working correctly, and allows appropriate decisions to be made as to whether any action needs to be taken.
- **Intervention:** dynamically manipulating a player's experience on a number of levels, either face-to-face or behind-the-scenes to respond to problems as they occur.
- **Improvisation:** supplementing a player's experience with personalised content performed live by a member of the production crew in order to engage a player with rich, reactive content or to rapidly prototype a range of game mechanics in a new experience.
- **Induction:** managing the every-day work of players entering the experience; ensuring that they are introduced to the mechanics and technology of the experience, and also allowing the production crew to moderate and troubleshoot the experience and associated systems as players begin.
- **Distributed Orchestration and Shared Awareness:** allowing orchestration to move beyond the control room. Each core aspect of a mobile mixed-reality experience is supported by dedicated production crew members, who work collaboratively to identify the changing needs of players and to react accordingly, either in the control room or on the streets.

This section now describes how the orchestration framework supports each of these requirements in turn, highlighting particular functions within each high-level requirement and demonstrating how each one has been developed and consequently used within one or more of our experiences. The next section will

then summarise these functions and reflect on the lessons learnt, showing how orchestration plays a useful and necessary role in a successful mobile mixed-reality experience.

6.3.1 Monitoring

The first deployment of Can You See Me Now?, as the deployment of any of the experiences, did not include any dedicated support for orchestration. The connectivity of mobile players, the state and quality of their GPS position reports, and their interactions with online players were largely opaque to the control-room operators, and this information could be gleaned only with some difficulty by studying the command-line output from the various server processes. Failures or temporary outages, such as loss of connection, GPS signal or software issues were therefore difficult to diagnose from generic statements from players such as ‘Why can’t I catch this online player?’ and ‘I don’t seem to be moving’.

Consequently, the orchestration framework begins in the control room; a critical element of the behind-the-scenes production of an experience. The orchestration of each experience begins with the continual and discreet monitoring of players’ interactions with the system, particularly the three core aspects of a mobile mixed-reality experience: **who** is currently taking part in the experience and how they are connected to it; **where** they are in the mixed-reality environment, both physical and online; and **what** they are experiencing in regards to the experience’s content. The production team must continually monitor these aspects of the experience for each player, and watch for particular failings or events regarding each one that may adversely affect the experience for one or more players; players’ mobile devices should be *connected* to the system, they should be *moving* smoothly and within the designed *time schedule*, and they should be receiving the appropriate *content*. Disconnection, lack of movement or conversely extreme or erratic movement, being in the wrong place at the wrong time or receiving incorrect content are all indications that a player may be struggling and require that the production crew intervene to rectify the situation. Conversely, an absence of these serves as confirmation that everything is likely to be working appropriately.

Control room activity often revolves around a central management interface, which provides an immediate summary view of the experience from the point of view of the system, and also provides mechanisms for intervention that will be discussed in detail in a later section. The management interface sits in the centre of the run-time system, and subscribes to and displays incoming data from players' clients and other supporting systems, and to outgoing data published by the game engine. This allows it to monitor and manipulate the flow of data through the underlying system, and if necessary allows multiple instances to operate concurrently. The management interface is modified cosmetically for each experience, although all versions implement this common core functionality.

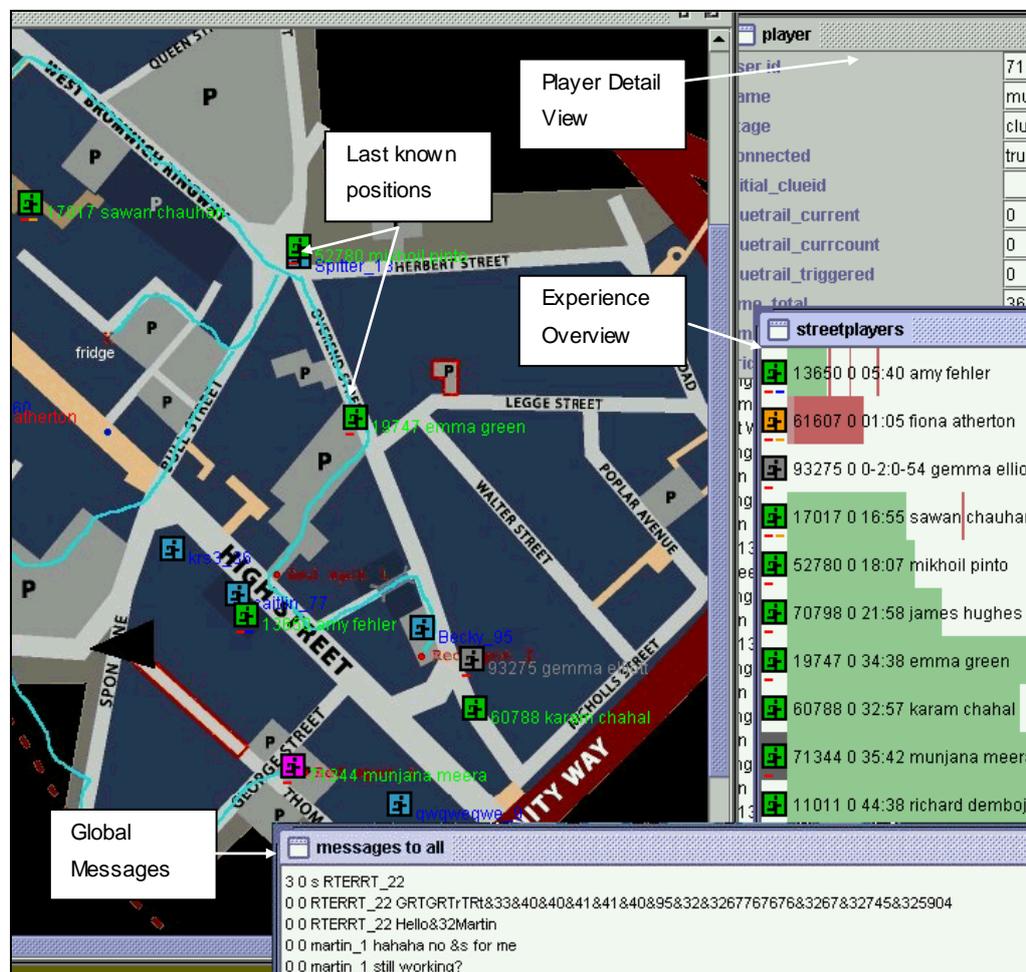


Figure 6-3: Monitoring components

The management interface consists of multiple components for viewing the state of players taking part in the experience, and also for intervention. Figure 6-3 shows the management interface as it was used during Uncle Roy All

Around You with monitoring components visible; the map component, experience overview component, player detail component and global message component.

- **Map:** this component includes a map of the game area to show where players are, with the **last known positions** of all players indicated with icons that are updated as the player moves. Current state is indicated by the colour of a player's icon – **online players** are shown in blue, while **mobile players** for whom the system is operating correctly are shown in green. Mobile players who are **disconnected** are orange, yellow if they are **requesting assistance** and grey if they have **run out of time** but have yet to be officially signed-out of the experience. Thus, it is important to show status as well as location.
- **Experience overview:** this component shows a list of players currently taking part in the experience, giving their **name**, a graphical indication of **how long** they have been playing for and which **stage** of the experience they are on. The solid green bar for each player in the overview gives a **visual representation of time remaining**; allowing the operator to view which players may be running out of time and may need assistance to hurry them to the end. The list also replicates the coloured icons of the map component. Selecting a player from the list highlights the player on the map, and brings up further information about that player's experience to date in the player detail component. Players currently taking part are always shown in the list whether their device is active and functioning or not – they are added to this list when they register, and removed when they return at the end of the experience.
- **Player detail:** this component gives in-depth information about the selected player, including numerical representations of connectivity and position, and specific information regarding their experience progress, such as **how many clues** they have received and **how much time** has been spent on the current stage. It also shows **all messages** that the player has received.

- **Global messages:** this component displays messages that are broadcast to all players, including **public player chat** and other **global system messages**.

The combination of these components gives the experience operator a variety of views of the unfolding experience. Glancing at the experience overview gives an immediate view of how many players are in the experience, and the distribution of their progress and state. If, for example, all mobile players are shown as orange icons then the operator can infer that there may be a global connectivity problem and should check on the relevant system, whereas if just one player is disconnected the operator should potentially assist them. Similarly, the map component shows the physical distribution of players and alerts the operator if a player is heading towards the edge of the game area, or the area of the final end-game performance and they should intervene or alert other members of the production crew. Once the operator has identified a player who warrants closer attention or a response, they can monitor the player detail component for a constantly updated view of the player's actions as they occur.

The monitoring elements of the management interface therefore play an important role in orchestrating an experience – to this end these were used in each one of our four experiences. At a high level they present an overview of players' actions within the system, allowing the operator to immediately see if the experience is grossly functioning. Next, the interface highlights those actions or events that are detrimental to the experience; disconnection, moving inappropriately or irregularly, running out of time or not receiving content, allowing the operator to drill down into that player's experience and monitor them more closely to determine what, if any, intervention is required.

6.3.2 Revealing History

As we have seen, a key task for the control room production crew is to remotely monitor players' actions within the experience, and this forms the basis of the orchestration process. However, it is not practical to expect the production crew to constantly monitor all aspects of every player's experience, and we have seen in the previous section how the management interface

highlights players for whom problems have occurred or players who have requested help.

The state of the experience revealed by the management interface is the current view, and often an operator will be examining a player *after* a problem has occurred in order to determine what has happened and what action to take, which will potentially involve finding the player and intervening in some way to get them back on track. Early experiences of Uncle Roy All Around You highlighted the difficulty of diagnosing a wide range of possibly concurrent but unrelated problems, using an interface that only reveals the current and constantly changing state of the experience. To properly judge the player's situation the operator not only needs knowledge of this current state, which may be incomplete as far as the management interface is concerned due to potential disconnection or other factors – the player may have moved or have incorrectly reported their position – but also what the player was doing up to the time the problem was recognised or the player requested help.

The management interface was consequently updated to respond to this requirement by providing a view not only of the current state of the players taking part in the experience, but additionally a **historical view** for each monitoring component. While it is common for text-based displays to show past messages, notably each monitoring component of the management interface also reveals an appropriate historical view of what it displayed previously, interleaved with the live display.

- **Connectivity:** the experience overview component, shown close up in figure 6-4, displays a historical record of when mobile player have been disconnected. For each player the green bar that indicates time remaining to complete the experience is combined with a graph that shows **periods of network disconnection**, with the width of each red bar representing **how long** the player was disconnected for. Short periods of disconnection are a regular occurrence when moving around the city and the player should generally be expected to reconnect without incident. However longer periods, such as those represented by

a wide red bar, indicate that something has gone wrong and the device is unable to reconnect, and this player warrants further attention.

- **Message history:** the player detail component, also shown in figure 6-4, maintains a historical **list of all messages and content** that the selected player has received ordered by time, alongside the detailed representation of the player's current state. This allows an operator to determine how the player has been interacting with the experience by reviewing recent entries; what communication they have had with other players or conversely if they have failed to respond to other players and whether they have repeatedly received a particular piece of content which may indicate that they are struggling to understand it.
- **Spatial trajectory:** the final historical addition is to the map component, which now reveals **where players have been** alongside where they currently are, as shown in figure 6-5. When a player is connected, their icon on the map is green, and its position is updated when the player moves. When the player is disconnected, their icon is changed to orange and remains in their last known position, rather than being removed from the map, providing a record of where they were. When a player is selected, the map component displays **a trail connecting the player's last movement updates**, showing where they have been. This historical movement trail allows an operator to see where a player is, whether they have been there for an amount of time – shown by a compact trail – or, if they are moving an indication of their **speed** and **direction**, potentially indicating where they might be headed.

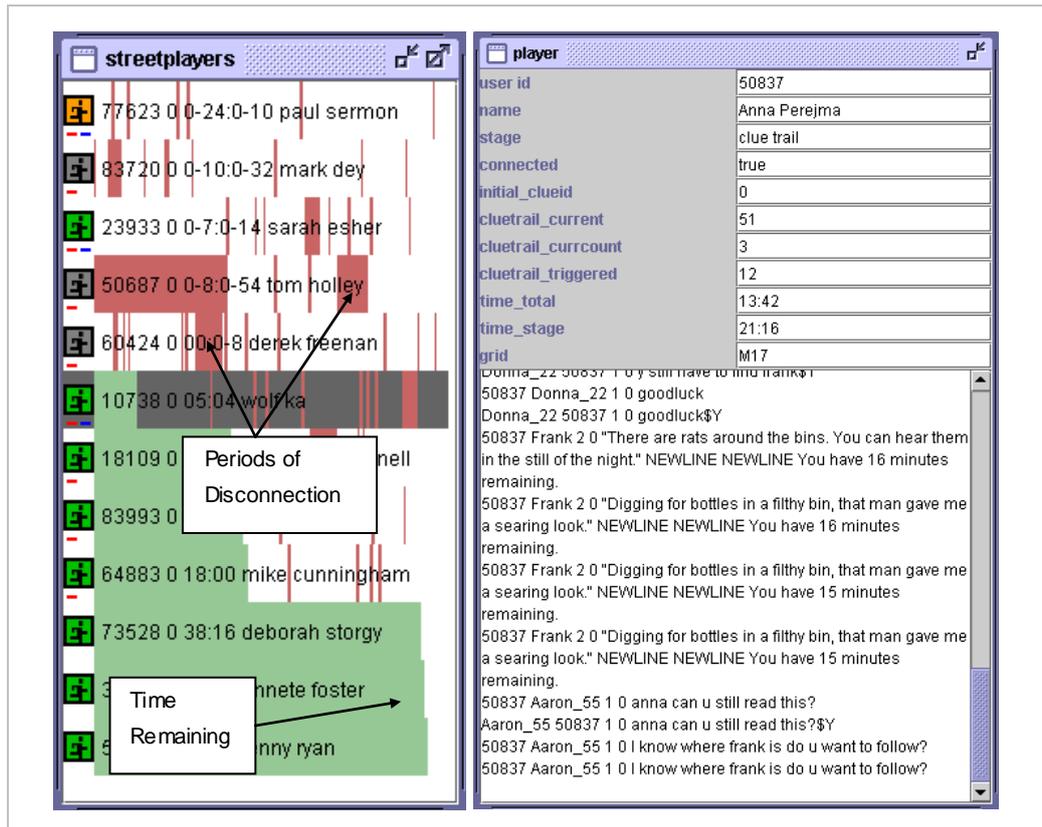


Figure 6-4: Connectivity history, left, message history and details, right

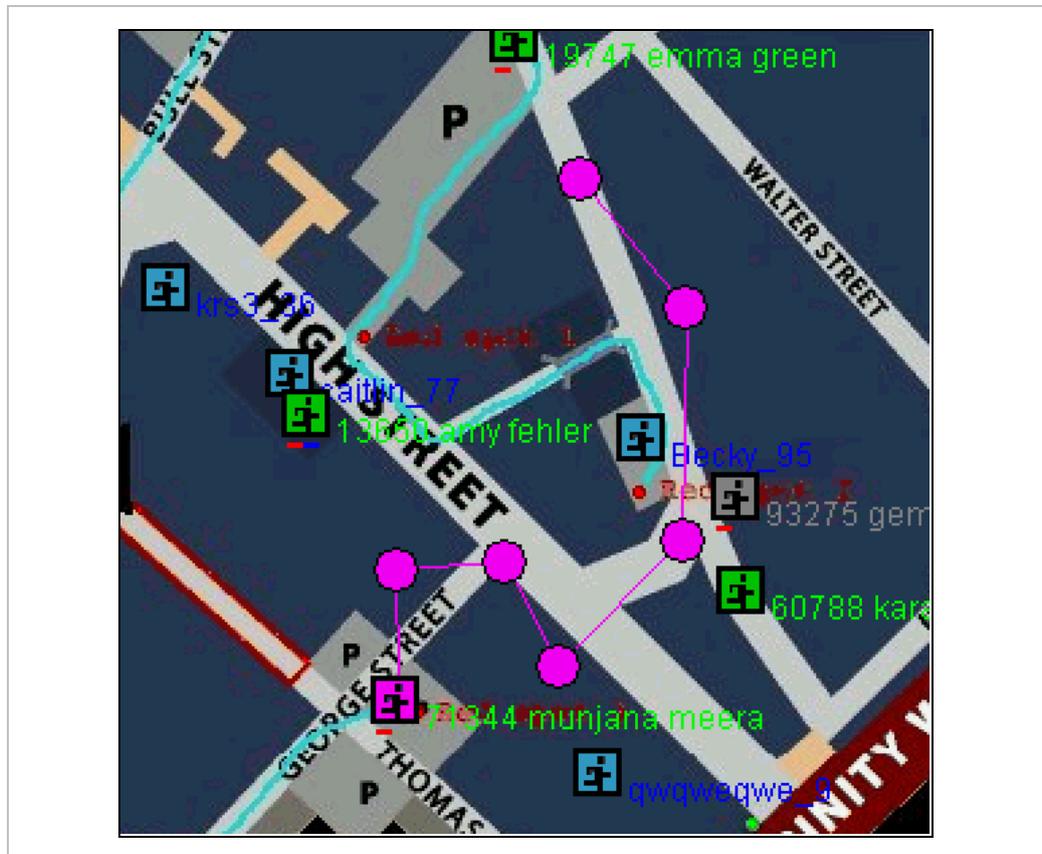


Figure 6-5: Revealing a mobile player's spatial trajectory

By reviewing the historical information displayed in the monitoring components alongside the current live view, the production crew can begin to build a more accurate picture of what is happening in the experience out on the streets. As has been described, this is used to respond to problems as they occur, in order to decide how to assist a player. However, this is conversely also used to periodically check that a player's experience is progressing smoothly; that they are receiving content, moving regularly and disconnecting and reconnecting as expected. In the orchestration of Uncle Roy All Around You and I Like Frank, these historical additions were used heavily in order to estimate whether a confused player was having problems and to what extent, and in Savannah they were extended to allow players to see where they had been and what actions they had performed.

6.3.3 Revealing Content

The first deployment of Uncle Roy All Around You revealed control-room operators using their anecdotal knowledge of the authored content of the system in order to predict the actions of mobile players, for example by knowing the locations of the initial red targets in the experience, an operator often inferred where the route that they thought a mobile player was likely to follow upon leaving the starting venue. Similarly, operators were occasionally seen attempting to steer players who were lost by considering which content they would receive if they were sent in a certain direction.

The final addition, then, to the monitoring components of the management interface was to reveal the content landscape that players inhabit and trigger during their experience. This completes the production crew's control room view of a player's experience by contrasting what they are likely to experience and where they are likely or should go next with what they have already experienced. It allows the production crew to immediately view whether a player is deviating from an ideal route through the experience, and to judge whether they should intervene in order to shepherd the player into a particular area, or how to get back on track. The management interface therefore reveals the content of an experience spatially and temporally on the map component and the experience overview component.

- **Indicating temporal milestones:** As described earlier, the experience overview component graphically shows how much time a player has left to complete the experience as a green bar. In Uncle Roy All Around You and I Like Frank, when this time has elapsed a player should have reached the end-game performance. As all players start the experience at different times and proceed through the experience at different rates, it is important for the production crew to be aware of players who may be moving slowly and need extra assistance to reach this stage in time. To indicate that the player should be nearing this stage with **five minutes remaining**, the bar changes colour to alert the production crew that potentially the player should be hurried, otherwise they may run the risk of running out of time.
- **Map annotations:** the map component is annotated with aspects of the content landscape as well as the physical environment, as shown in figure 6-6. Chapter 5 described how the ColourMap layers used to define the content landscape that players inhabit can be merged and exported as a single image for orchestration, retaining useful information of how players will take part in the experience. The image retains the physical map of the game area, but may also include where **content regions** are, **start positions** for mobile and online players, **ideal routes** through the experience and the GPS **filtering map** if this is used. It also marks **key physical locations**, such as the location mobile players are registered and the end-game performance.

Using this information the control room operator can infer details about where a player is likely to be and what they should be experiencing in terms of experience content, even if the player is disconnected and this information is not directly or incorrectly reported to the system. This becomes useful when coordinating other members of the production crew in order to find a lost player, or to notify performers that the player may be approaching soon. For example in Uncle Roy All Around You and I Like Frank, knowing that a player is on the clue trail and was last seen moving along an ideal route leading to the office, the operator can alert performers to look out for the disconnected

player. Similarly, knowing which starting position a player has been assigned as they leave the venue, and where the position is on the map, the operator can inform other crew members where the best locations are from which to monitor the new player, and which direction they are most likely to have headed in. In *Can You See Me Now?*, the production crew can view whether start positions or the GPS filtering map are correctly defined.



Figure 6-6: Revealing map annotations

This depiction of the content landscape plays a fundamental role in the orchestration of Savannah. As described in chapter 3, the Savannah Den interface serves a dual purpose; allowing the experience operator, in this case a teacher, to monitor and manipulate the experience while it is running, and secondly to reflect and review on the experience with the players after the event. It displays the current positions of players, combined with historical trails of their previous movements and where they have performed actions in

the experience. During the exploration periods of Savannah, players move around the physical game area receiving automated content based on their positions. Higher level game-play is performed manually by the operator by manipulating health and sending messages. By revealing the content on the map the operator can quickly see where players are and what content is around them, in order to manipulate the experience.

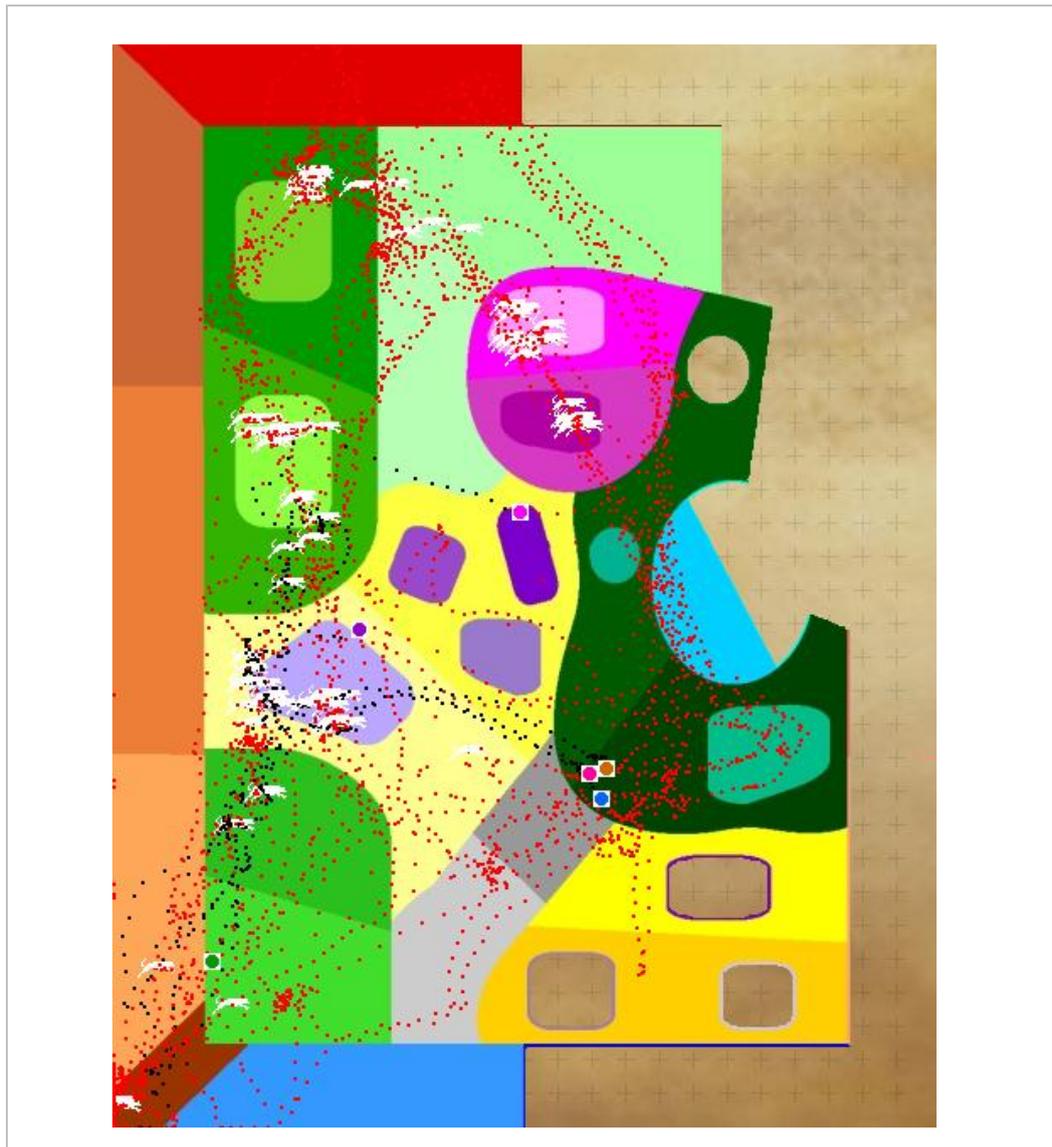


Figure 6-7: Revealing how content has been explored in Savannah

During the reflection period of Savannah, players review how much of the game area they have explored. This is revealed by the management interface, which shows those content areas of the ColourMap that the players have visited, while hiding those that have yet to be explored, as shown in figure 6-7. Based on this, players then revisit the game area to explore the remaining

areas, once again with the operator governing the game play. Similarly, by viewing how players are interacting with the content the production crew can modify the layout of the content for the next outing of the experience, as part of an ongoing development process.

Revealing the content landscape of an experience aids an operator in orchestrating the experience by showing the content that a player is likely to receive, allowing them to judge how improvised content or impromptu interventions should be balanced with the displayed pre-authored content. Moving beyond just displaying a map of where players are physically, it allows operators to make assumptions about where they may be going or, as we have seen in Savannah, allows players themselves to decide where to go next. Combined with monitoring current events and the historical views of past events described in the previous section, an indication of where a player will be going and what they are expected to do *in the future* gives the production crew a powerful tool to monitor players, and to make judgments of how and when to intervene in a player's experience.

6.3.4 Intervention

As we have seen, a large proportion of each of our experiences revolves around automation, in that the player receives content based on pre-defined game rules, pre-authored content and scripted events within the system. This is tailored to how a player is expected to progress through the experience, and also aims to encompass a range of situations and scenarios such as moving off route, or operating while disconnected. However, chapter 4 proposes that a mobile mixed-reality experience requires the ability to mix pre-authored “on-rails” content with live performance in order to respond to unexpected events, in order to dynamically orchestrate the experience for the player. This was especially apparent during the first deployment of Uncle Roy All Around You, where the operators were regularly faced with scenarios in which the pre-authored content of the experience was not sufficient to cope with a particular situation, and ultimately required gross human intervention – for example physically directing a lost player, or quickly manually manipulating a new

device to the same state as one that had failed, but in full view of the player and potentially disrupting their experience.

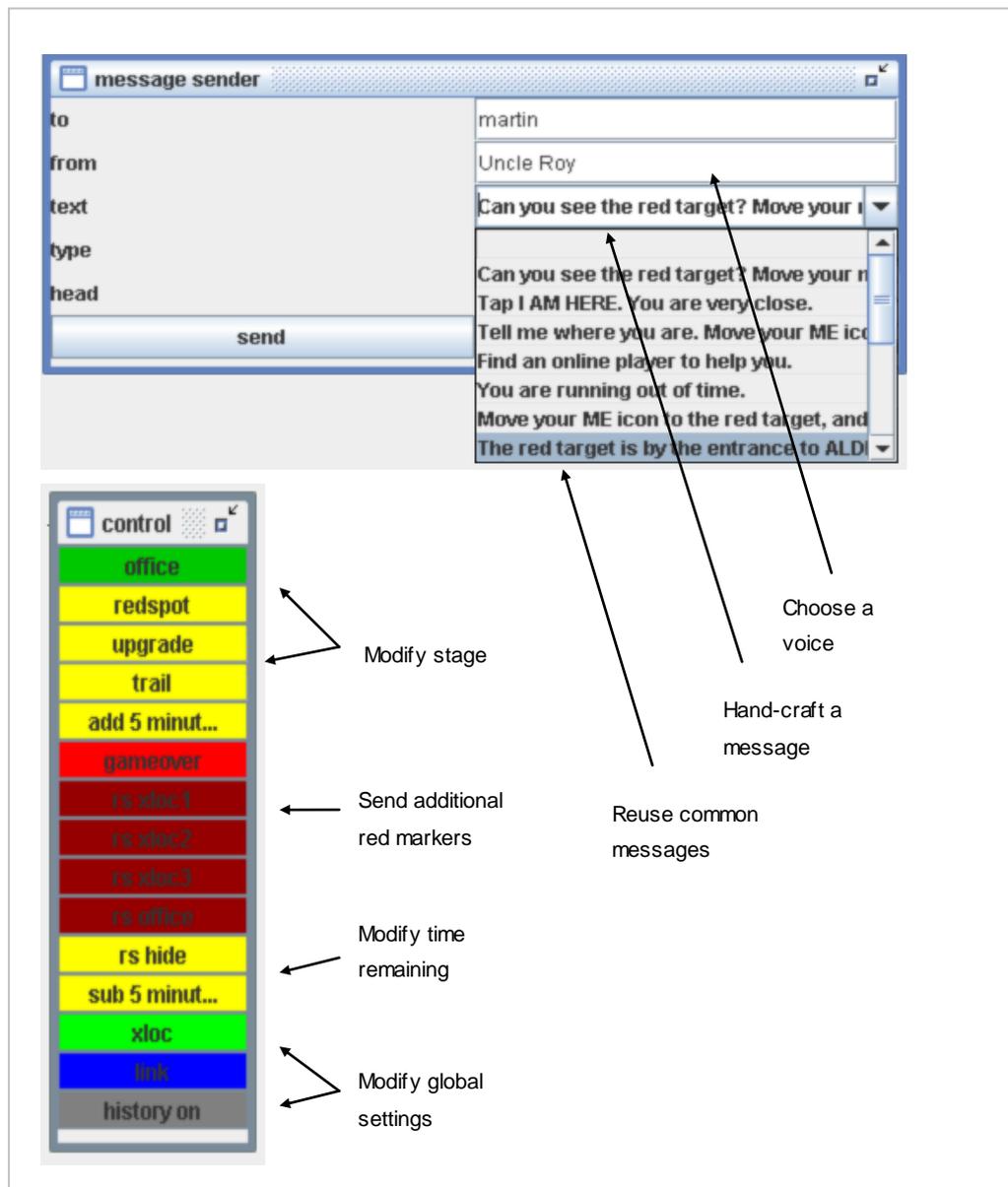


Figure 6-8: Intervention and message sending components for Uncle Roy All Around You

A requirement, therefore, for the orchestration framework was that the automated system be malleable – that the production crew can intervene in order to override, modify and supplement the system to manipulate a player’s experience; bringing it back on track or responding to unforeseen events in a planned and organized manner within the context of the experience. Without mechanisms to intervene within the system as the experience unfolds, the production crew can only grossly intervene when something goes wrong, potentially damaging a player’s engagement with the experience.

In response to this requirement the management interface was modified to provide a second set of components that, sitting alongside the monitoring components, allow an operator to modify all aspects of the player's experience; by remotely manipulating the state of the player's game-state and by supplementing or replacing automated content as they see fit. In relation to chapter 4, this shifts the experience away from being a completely pre-authored game by allowing the operator to inject additional unscripted content in order to respond to an individual player's needs or circumstances – analogous to the behind-the-scenes production of static performances described earlier. The management interface supports this with two intervention components as shown in figure 6-8.

- **Control panel:** this component allows the operator to remotely **change the system state** of any players within the experience. If the selected player is connected these changes are immediately reflected on the player's device, or if they are disconnected when they reconnect. The functions made available by the component vary due to the differing requirements of each experience, but in general allow the operator to modify any aspect of a player's state within the system. In Uncle Roy All Around You and I Like Frank the operator can increase or decrease the amount of **time** the player has left, perhaps adding time that they have lost through hardware or software failures, or giving extra time to reach the final performance if they are very close. They can manually override the **stage** of the experience the player is currently on, perhaps if they are struggling to perform a required action themselves. They can also manipulate the **target** that the player is heading for, if they are moving in the wrong direction and it is easier for them to achieve a different one, or to indicate the correct direction. In Savannah, the operator can increase or decrease the player's **health**, causing a corresponding change on the player's device. Finally in Can You See Me Now? the operator can manipulate **global variables** that affect all players, such as whether the capture function is operational, or what the capture radius is.

- **Messaging:** this component allows the operator to **send messages** to one or all of the players using any of the **multiple “voices”** of the system, and have them displayed to the players as if they had been sent by the system as usual. In this way the operator can **perform many roles** within the experience; to online players as an online player themselves, as the voice of the game – Frank or Uncle Roy, or to all players as a game master acting in an administrative capacity. Most importantly, this component allows the operator to **hand-craft** messages to players that need particular and personal assistance, while lessening the risk of the player becoming disengaged from the experience by having the message displayed in the usual manner.

These two intervention components give the production crew the ability to intervene technologically on two levels. The control panel component gives the operator a “hand of god” interface that allows *gratuitous intervention* (Laurel, 1991) by overriding the normal, planned operation of the system in a way that may potentially be obvious to the player and lessen engagement – they might suddenly have more time than they did, or their device may display an altered interface. This would normally be used in response to similarly disruptive events, for example a player's device crashing. The messaging component is a more subtle, hidden mechanism for weaving in additional content – an action that the player would preferably not be aware of thus maintaining engagement. Both of these components have been used extensively in all four of our experiences.

6.3.5 Improvisation and Performance

Chapter 4 proposes that mobile mixed-reality experiences should mix automated, pre-authored content with dynamic, improvised content, and participation through play with performance. This is a common feature in role-playing games, as described earlier, although overly rigid or pre-scripted experiences may hamper a participant's sense of autonomy, whereas an experience with too little direction may result in a weak storyline or with participants not fully knowing how to take part – as described in (Jonsson et al., 2007).

This section describes how the monitoring and intervention functionality provided by the management interface can be used in conjunction with pre-authored content to support improvisation, where high-level game mechanics are performed by the production crew as part of the orchestration task as the experience progresses. This allows the operator to rapidly explore new techniques and immediately respond to player actions, which can then potentially be implemented as automated rules in later versions of the experience, allowing immediate play-testing through rapid prototyping.

- **Supplementing content:** experiences such as Uncle Roy All Around You are based around a large amount of pre-authored content, however as we have seen in the previous section an ongoing orchestration task for the production crew is to perform additional content by **manipulating** a player's experience and sending **additional** messages, in order to steer the player when the system breaks down or where the pre-authored content is lacking. Similarly, in Can You See Me Now? the mechanics of the game are implemented as pre-authored game rules, however the performers continuously supplement the experience with improvised chat over the live audio stream, **performing around failures** or idiosyncrasies in the mobile technology in order to maintain an engaging experience for the players.
- **Rapidly prototyping content:** a second approach is to use improvisation and performance as an **alternative** rather than an addition to pre-authored content. This involves mixing basic pre-authored content with game rules that are improvised on-the-fly and performed live by the production crew. **Manually operating** the “rules engine” in this way provides the operator with maximum flexibility when exploring **new experience scenarios**; what happens to players while they inhabit the underlying content terrain, the tasks they have to perform and their outcomes. This “low-tech” prototyping is used as a tool to support participatory design.

Savannah uses improvised performance to support rapid-prototyping of the experience, to discover which scenarios are the most engaging before they are

implemented as automated game rules. As a result it does not contain any high-level game logic. As players move around the field their devices display sounds and images based on location, giving a general impression of the virtual Savannah by representing the sights, sounds and scents of various animals and the nature of the terrain. Each player's device also displays their "health" – a value that can be changed by the experience operator but which has no automated effect within the experience. The game-play in the experience is based on improvised text messages sent to players which are displayed on their devices, for example telling them the outcome of an action or movement with perhaps a resulting change in health, or instructing them what to do next.

The Savannah management interface again implements the control panel and message sending components described previously. The operator can choose to select a message to one or more players at a time, and messages that have been sent are added to a drop-down list in the messaging component where they can be selected and sent again if required. The map component again continually shows where players are, again with historical trails showing where they have been, and where they have performed actions such as attacking or scenting – this allows the operator to immediately respond to player actions.

During Savannah, the operator uses the management interface to improvise scenarios for players out on the field, and to respond to the content regions they occupy. For example, in Savannah the edges of the game area are seen as hazards that can harm the players and reduce health, so as to force the players back towards the central playing area. This scenario is achieved by sending players in this area a message saying that they are in a dangerous area and if they remain in this area the operator decrements their health, sending them a message to explain why. However, if the operator believed that the player has accidentally strayed into this area, they may choose to be lenient and not alter health. Other scenarios are constructed by the operator in order to stimulate exploration or to pose a specific challenge; for example by informing players that they are thirsty and need to find water and gradually decreasing health until they do. The location-based content merely suggests what may occur, for example that the players are in an area where elephants may be – it is up to the operator to create the scenario of a particular elephant, and then respond to the

players' possible attacks with descriptive messages detailing the reasons for success or failure and appropriate change in health.

This section shows how an operator can use the management interface to act as a game master to *improvise* an experience, as in a role-playing game. It provides a feedback loop that allows the operator to *perform* a rich and reactive experience to players from the control room, in which players respond to the operator's scenarios and the operator responds to the players' actions. Although these techniques are not immediately scalable to many players, they provide valuable insights into the early stages of the design of the experience, without being hampered by having to hard-code it before being able to test it. They also free the operator from the constraints of a completely automated experience, providing a balance between pre-authored and improvised content, and performance and play as defined in chapter 4.

6.3.6 Inducting Mobile Players

So far we have been concerned with the orchestration of an ongoing experience – the behind-the-scenes production of the experiences that involves managing players as they progress. As public performances, the next challenge for orchestrating a mobile mixed-reality experience is managing the entry of mobile players into the experience for a number of reasons:

- **Tickets and timeliness:** public experiences may involve a player purchasing a ticket to take part at a certain time, and as a result the time they should have to wait before beginning should be minimised.
- **Limited resources:** a limited number of mobile devices, or set performance pieces in the experiences that ideally only involve one player, set an upper limit on the number of players who can take part at any one time. Delays in a player starting on the experience quickly introduce knock-on delays for future players.
- **Introductory briefing:** the experience may involve players being briefed in how to use the technology, safety advice and an overview of

how to play so that they can begin confidently without losing any of their allocated time.

- **Minimising disruption:** ensuring that all aspects of the system are ready to receive the player, and that any failures are noted and fixed quickly in order to minimise any potential loss of confidence in the experience.
- **Maximising engagement:** the experience begins for the player even before they have exited the venue onto the streets with their mobile device. As the first point of interaction with the experience, any introduction has to carefully manage player expectations in order to maximise engagement.

The orchestration framework responds to this requirement by introducing a procedure that manages mobile players up to and including the time that they step out onto the streets, viewing it as critical as the rest of the experience. This first point of interaction with the player is theatrically referred to as *front of house*. The induction process attempts to combine the system requirements of successfully and repeatedly registering and logging players into the experience with the performance requirements of introducing players to the experience, and providing support and monitoring as the player leaves the venue.

The induction process is used to support the front of house process in Uncle Roy All Around You and I Like Frank, both of which require a steady flow of players through the experience. A similar approach is taken in Can You See Me Now? - although the experience did not involve public mobile players, the street performers require a rehearsed and established process to ensure that everything was working before the experience opens, so that the *performance* can begin on time and any technology failures are hidden from public players.

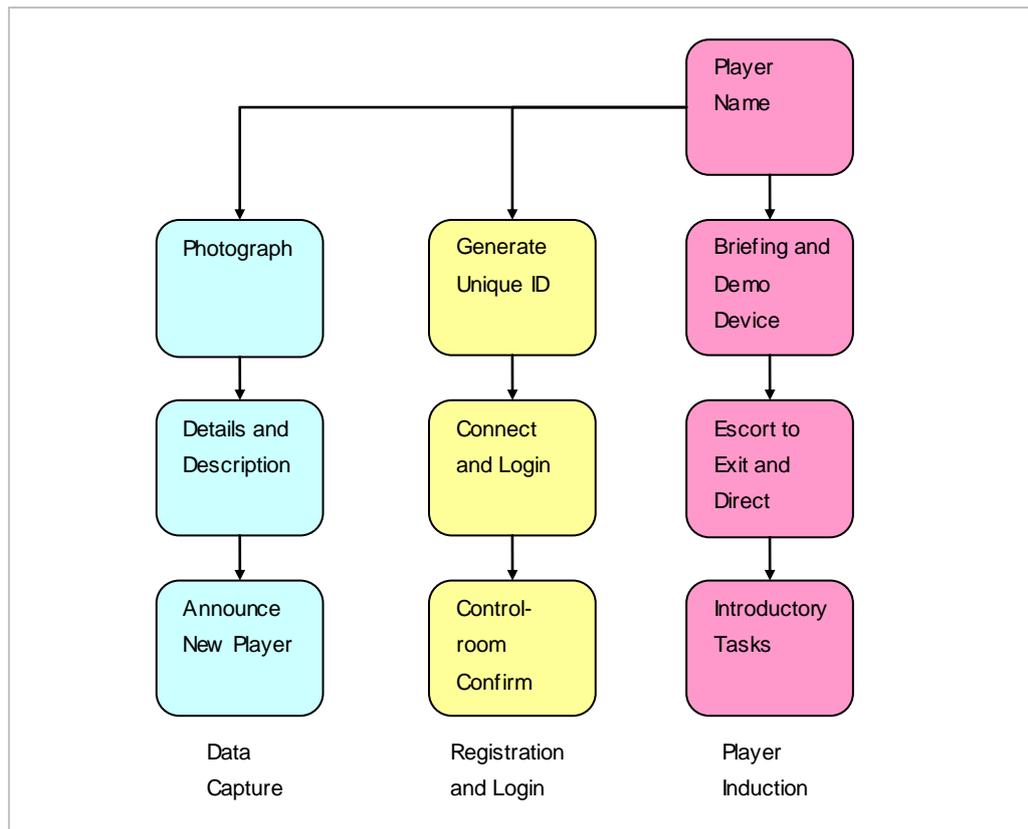


Figure 6-9: Induction process

Figure 6-9 shows how the induction process is split into a number of functions:

- **Data capture:** front of house staff capture two different kinds of data, that which is used as part of the experience such as the player's **name** and **photograph**, and that which can be used to aid orchestration such as the player's **description** and any **special requirements** that other members of the production crew should be aware of. The fact that a new player has successfully entered the experience is **announced** to the rest of the production crew.
- **Registration and login:** a **unique ID** is created to identify the player within the system, which is used to **connect and login** the player on a fresh device. The control-room **confirms** that the device is connected and that it is ready for the player to use.
- **Player Induction:** front-of-house staff **brief** the player, introducing the player to the experience and demonstrating how the technology is used using a dedicated **demonstration device**. The player is **escorted to the**

exit and shown which direction to head off in. They are prompted to perform a series of **introductory tasks** using the mobile device to teach them the basic skills that they will need to use throughout the experience.

46683		16:01:47	ian sheperd	male, short blond spiky hair, black tshirt, blue-green jeans	PDA: 12	Log Out	Edit Details
19292		15:59:44	tim enright	male, black short hair, white tshirt with orange short sleeves, blue jeans	PDA: 1	Log Out	Edit Details
76895		15:59:03	gary carr	male, short black spiky hair, yellow tshirt, blue jeans	PDA: 3	Log Out	Edit Details
52234		15:44:49	andrew bond	male, large, black short hair, dark grey topblack trousers	PDA: 4	Log Out	Edit Details

Figure 6-10: Capturing player data for orchestration

A front of house performer uses a terminal with a web browser to perform the registration and data capture tasks. A web interface creates a new profile for the player in the system, and the performer uploads a photograph of the player and enters their name and a short description of what they are wearing and other distinguishing features, to be used to identify the player visually on the streets, as shown in figure 6-10. Once the profile has been created the player is active within the system, and as such can be seen on the management interface until the player is logged out again using the same interface at the end of the experience.

The performer next prepares the mobile device that the player will use and logs it into the system. The devices are all installed with an identical software image, allowing a device to be immediately replaced with another if it fails. On creating a new player profile, the registration web interface generates a five-digit, non-sequential unique ID that is used to identify the player throughout the system. Non-sequential numbers are used to make it easier to distinguish between the numbers in any of the interfaces, and all of the generated numbers are wholly divisible by 91, which is used as a simple checksum to prevent errors in entering. The performer enters the player's ID into the device, which

then attempts to connect to the system. The ID associates connections from the particular device with the player, and if the device fails at any point the player's ID can be entered into a second device and the player can continue. The performer and the control room operator collaborate to monitor this process, to ensure that the player and the device have successfully registered and connected and are ready to begin.



Figure 6-11: Preparing the device, left, while players are briefed, right

While registration is taking place, a second performer introduces the player to the experience, giving a short, rehearsed script of the main tasks in the experience and what the player will have to do, as shown in figure 6-11. They introduce the player to the fundamentals of the mobile technology using the demonstration device, indicating how to use the map, how to report their position and how to get help if things go wrong. The player has a chance to ask questions, although the script is designed to give them enough information that they can confidently begin the experience.

Once the registration and briefing has been completed, and the control room has confirmed that the player's device is ready to go, the device is handed to

the player. The player is escorted to the exit of the venue, and the performer indicates the direction to take as they begin the experience. The first stage of the experience for mobile players is deliberately designed to build upon the initial introduction at front of house, teaching each new player the basics of interaction with the system, during which time they are monitored by the production team. Indicating the direction to take initially helps the player to take their first steps with the device, and from then on a series of timed instructions helps the player to try out the device's functions while moving towards a location explicitly marked on the map – recording an audio message, reading messages from the system. Once this stage is completed the experience properly begins; the mobile player is revealed in the online environment and enters the main clue trail.

The induction process provides a structured mechanism for introducing players to the experience in a timely and repeatable manner, and it proved to be essential in the continuous operation of the experiences. It benefits the orchestration task by enabling a regular flow of players into the experience, collecting data that can be used throughout the experience and gives the production crew the opportunity to notice difficulties early on.

6.3.7 Inducting Online Players

Our mobile mixed-reality experiences require a balanced ratio of online players to mobile players. The number of mobile players is often limited by the number of devices available, or the number of performers and how many players they can support at any one time. Similarly, there is a minimum and maximum number of online players taking part to ensure the best experience for both groups. This is especially true if, as with our four experiences, fundamental elements of the experience revolve around personal interactions between members of both groups of players.

The absolute maximum number of online players is obviously limited by the software capability and bandwidth of the system as a whole, although there is also an experience specific ideal maximum. The number of online players attempting to take part in the experience at any given time is also largely unpredictable and may fluctuate wildly, depending on how much publicity the

experience has had, and also based on the fact that players may log in from any part of the world and from a variety of time zones. Whilst access to the mobile experience can be regulated by tickets sales by the hosting venue, too few online players may make for an unsatisfying experience for mobile players, and too many online players may leave some not able to join. The next requirement for the orchestration framework, then, is to manage and regulate the flow of online players into the experience.

The orchestration framework presents a reusable queuing system for online players, while to some extent mirrors the philosophy of the front-of-house induction process – to allow prepare players for entry into the experience by supporting the following functions:

- **Accumulating players:** allowing online players to **accumulate before the experience begins** while the production crew is making final preparations, so that when the experience does start there is a pool of online players for mobile players to interact with straight away.
- **Limiting numbers: maintaining the desired ratio of online to mobile players,** while also limiting the number of online players to within the tolerances of the system.
- **Restricting access:** providing a structured mechanism for **stopping players from entering** the experience in the case of technical problems while maintaining their connections, rather than refusing connections which would imply that the experience had suffered severe failures.
- **Moderating access:** preferential access allows **selected online players** to begin the experience ahead of others. Conversely it allows **fair access** through a natural turnaround of available slots in the experience, so that all players in the queue are able to play.

To join the experience, online players visit the associated website. Around fifteen minutes before the experience is due to begin the website is updated, allowing players to download and activate the browser-based online client, which connects to the queue. The queuing system is implemented as a server to

which the online client connects, and which maintains an ordered list of connected clients waiting to join the experience, with the first to connect being the next to join. Regular updates are sent to all connected clients indicating their place in the queue, so that each client can display an estimate of how long the player will have to wait before joining.

The server monitors data published by the game engine that indicates whether the experience is open or not, how many online players are currently in the experience and what the maximum number of players is. This data may be dynamically updated by the operator at any point during the experience and the queuing system reacts to it by allowing players to join or restricting their entry. Once there is a slot available in the experience, the queuing system sends the online client at the head of the queue the address of the main online game server; allowing the online player to disconnect from the queue and connect to the main server, thus joining the experience.

A simple interface displays the name and IP address of connected clients, and allows selected players to be “upgraded” and enter the experience even if it is full or technically closed. This allows operators to test the experience before it is fully opened, or to allow local visitors to take part ahead of other online players. Conversely, the interface allows an operator to kick online players from the queue by dropping their connections, or even to permanently ban the IP addresses of online players who persistently cause problems in the experience – stopping them from connecting to the experience at all. At the end of the experience - for example after the online player has been caught in *Can You See Me Now?*, the player’s online client is disconnected from the server, and they are redirected to the experience’s web-page. They are welcome to play again, but must join the back of the queue as before – ensuring a constant turnover of new online players.

The queuing system has been used to regulate the many thousands of online players taking part in *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank*. In this way, the online queuing system supports the experiences as performances – players to wait for the experience to begin, and how many players can concurrently take part is regulated in order to produce the best

experience for both groups of players. At the same time, it provides a mechanism for the production crew to orchestrate the online game as the need arises.

6.3.8 Distributed Orchestration and Shared Awareness

So far we have largely focused on applications, processes and techniques for the control room that support the orchestration of mobile mixed-reality experiences. These mechanisms allow the behind-the-scenes production crew to operate within the bounds of the system to orchestrate player actions through monitoring, intervention and improvisation. However, this only accounts for half of a mobile mixed-reality experience. *Mobile* mixed-reality experiences place players on the streets; the majority of their actions occur in the physical environment and fall outside of the controlled scope of the mobile device and supporting automated systems.

Consequently, the control room remains largely unaware of what mobile players are *actually* experiencing – has a player stopped moving because they are confused and do not know where to go next, or are they just waiting to cross a road? In *Uncle Roy All Around You*, the mobile device may have instructed a player to leave the office and head towards the limousine, but have they actually done so? The management interface cannot tell the operator if a player is disconnected but continuing with the experience while their device attempts to reconnect, or if it has crashed and they are standing still not knowing how to proceed. Similarly, the control room operator cannot remotely reset a device or reconnect a disconnected network connection – they can only intervene within the scope of the operating system. While in an experience such as *Can You See Me Now?*, in which mobile players are professionals who perform for an online audience, mobile players can exploit this ambiguity to create a rich experience, for a public mobile player these circumstances can potentially be problematic.

Based on these regularly occurring events in early experiences, the orchestration framework was modified to respond to this by considering the control room as only one part of a combined orchestration effort. Just as a mobile mixed-reality experience revolves around players' actions on the streets

and simultaneously within a virtual environment, orchestration takes place virtually – within the system via the control room – and on the streets via members of the production crew distributed deliberately throughout the physical environment as *street performers*. The orchestration task is distributed between the control room and the street performers; mirroring the control room management interface by monitoring and intervening in mobile players' experiences *in the physical environment* by discretely watching players and with face-to-face interventions if necessary.

This distributed orchestration evolved from an informal practice to a defined strategy that included all members of the production crew, monitoring players from their respective points of view, and then broadcasting this information to other members in order to collaborate to create a combined perspective of what players are actually doing, both on the streets and within the system. This shared awareness is then drawn upon in order to make a decision as to whether any action, such as intervention, is required, and if so which member of the crew is best situated to perform it. Distributed, collaborative orchestration work serves to glue the potentially disparate elements of the experiences together, from the initial induction of players, trouble-shooting unexpected events and managing players through the final performance, and general monitoring to make sure everything is running smoothly from all points of view.

Practically, the orchestration framework supports distributed orchestration and shared awareness by implementing the following functions:

- **Constant communication:** each member of the production crew continually listens to a **shared walkie-talkie channel** for information that may affect their particular role, and also broadcasts information regarding their activities or sightings of players.
- **Shared map references:** each member of the production crew is equipped with a **map** showing the game area overlaid with a **grid-reference scheme**, allowing them to quickly report their own position or that of a player, and conversely to pin-point positions reported by other members of the crew.

- **Mobile street performers:** the team of street performers is spread across **key vantage points** in the game area in order to both **covertly monitor players** and relay this information to the rest of the production crew, and to **perform face-to-face interventions** if necessary.
- **Shared database access:** the control room, front of house and the street performers all have **access to the database** of player descriptions and photographs captured during the induction process via a web-interface. Street performers can access the interface while on the streets using a PDA in order to aid in the **identification of specific players**.

The use of distributed orchestration is a core task in the day-to-day work of operating both Uncle Roy All Around You and I Like Frank. It allows the production crew to directly contrast the actions of players on the streets with their actions within the system as viewed in the control room, in order to expose potential problems or just to confirm that a player is in fact progressing as they should. This can be seen from the following walkie-talkie dialogue between a street performer roaming the streets and the control room during Uncle Roy All Around You:

Runner: OK just to warn everyone, there's 2 guys just around the edge of China Town, one's got a pony tail and they're really legging it to try and get to the office by the look of it.

The controller selects the descriptions list, searching for the player described. He looks at photos of 2 players and selects one for a closer look then closes the photo.

Control room: OK, that's Johnny and Francis. They've both got time left. They've got like 25 minutes.

Runner: Oh yeah, but they're really really sprinting.

The above shows how the street performer recognises unusual player behaviour – two mobile players running. Listening in, the control room operator checks how long they have left and realises that further intervention is not required as they have plenty of time based on where they are. Notably, a street performer

failing to spot a mobile player in the area that they have been described as being in by the control room is just as important as spotting a player – the runner can alert the control room that the player is missing, and the two can then collaborate to see if there is a problem.

Street performers may intervene in a mobile player's experience if they are either having technical difficulties that cannot be resolved remotely, or interpretive difficulties where they are unable to relate the pre-authored content of the experience to their current physical environment. In these cases the performer liaises with other members of the production crew to confirm the player's identity and to identify what assistance they need before intervening face-to-face, perhaps by resetting their device to restart the client, or just by pointing them in the right direction. Finally, both the performer and the control room confirm that the player is back on track. In the dialogue below, the control room operator is informing a street performer of a technical fault that a player is experiencing, and that their device needs to be restarted:

Control: Yes, his PDA just needs resetting. He's connected but it doesn't think it is, so he can't send audio. He's a young male, he's got short brown hair, dark blue coat, and blue jeans.

Caitlin: Is he at the phonebox that's there?

Control: Yep. I told him to wait at the phonebox. He just needs the PDA resetting.

Caitlin: Caitlin to control, can you confirm that Craig is now connected and in the game?

Control looks at connectivity interface. Craig's player icon is highlighted orange and disconnection is visible

Control: He's not connected at the moment. Make sure he's connected.

Caitlin: I've just put Craig back in the game. He's on the clue trail heading for the (inaudible).

Control looks at Craig's connectivity and can see that he is connected

In this way the control room and the performer collaborate in order to fix the problem, and to make sure that the player is ready to continue before sending them on their way.

As we saw earlier, front-of-house performers and the control room operator collaborate to ensure that each new mobile player is registered correctly, and their device has connected to the network in order to begin the experience. Using the management interface, the control room operator has a system-centric view as to whether the device is ready to go, and this information is broadcast over the walkie-talkie channel. Similarly, if there are any delays in the induction process, they are broadcast by the front-of-house performers, and they and the control room will work together to resolve them. Front-of-house announces that a new player has left the building, and the control-room announces in which direction the pre-authored content will send them. This alerts the street performers, particularly those in the vicinity of the venue, to be on the look out for the new player and to watch to make sure they are heading confidently in the right direction, or if they look like they are struggling. A similar process is in place to recognise players as they approach the end of the experience.

This constant, distributed orchestration of the experience enables the production crew to respond when things do not go according to plan; street based monitoring is critical if players are not to be stranded through technological or interpretive failings. However, it also allows the production team to blend face-to-face interventions into the experience by responding quickly and confidently so as to minimise any disruption of a player's engagement.

6.4 Summary of Orchestration Functions and Tools

This section now summarises the key functions and tools of the orchestration framework, reflecting on its role in supporting our four experiences and considering general lessons that can be learnt from the framework.

6.4.1 Functions and Tools

The orchestration framework encompasses a set of principles that the production crew employ in order to operate each experience, consisting of supporting tools, strategies and processes rather than one monolithic application. Figure 6-12 shows the union of functions that are involved in orchestrating a particular experience, and although each experience has a different implementation of the orchestration framework, and not all of the detailed functions may be applicable or relevant to all experiences, the experiences all make use of each high level function.

Monitoring the ongoing experience is a core requirement of orchestration, and all four of our experiences require this. The orchestration framework supports monitoring via a control room management interface that allows the production crew to monitor multiple aspects of the experience:

- **Monitoring the present:** a variety of components within the management interface allow the operator to immediately view the current state of the experience, to judge whether it is functioning, if particular players need closer attention and the distribution of players across multiple aspects of the experience. A **map** displays each player's last known position in the mixed-reality environment. The **experience overview** shows a list of all players, how much time they have remaining, their current state, for example whether they are connected (Uncle Roy All Around You) or attacking (Savannah). The **player detail view** reveals more detail about the state of a specified player.
- **Revealing history:** each component within the management interface displays a historical record of what players have been doing, to allow an operator to make a judgement as to whether a player is repeatedly having difficulties, or to reflect on exactly how players have been taking part in the experience. This includes where they have been via their **spatial trajectory**, when they have been disconnected and for how long via the **connectivity graph**, and what content they have been receiving, via the **message history**.

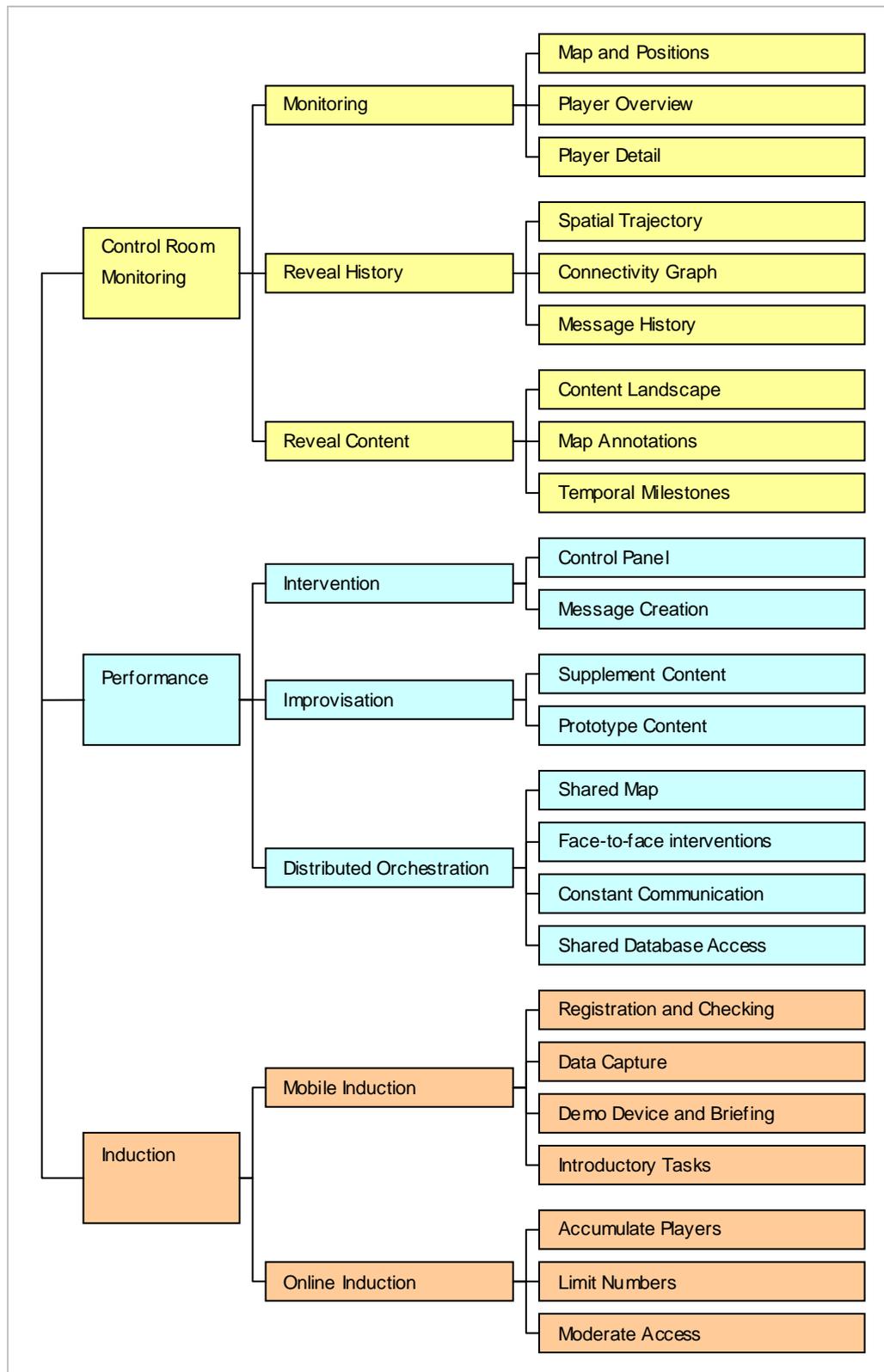


Figure 6-12: Summary of orchestration functions

- **Revealing content:** The **content landscape** that players inhabit is revealed throughout the management interface, including key **temporal milestones** and **map annotations**. This allows the operator to

potentially direct players to areas that they have yet to explore but plan to (Savannah), or give an indication of likely routes through the content that a player is likely to take (Uncle Roy All Around You and I Like Frank). This also allows the operator to make judgements about where a player is likely to have moved to if they are disconnected and this information is not directly available, or where a player *should* be at a given point in the experience in terms of both position and progress.

The orchestration framework supports the production crew in intervening and improvising in order to perform elements of an experience, to respond to unexpected events, tailor it to a specific player's needs, or to add additional, dynamic content. Each of our four experiences involves performance as part of the orchestration process:

- **Intervention:** The intervention components of the management interface allow the control room operator to manipulate a player's experience, within the supported context of the system. This involves overriding or altering automated functionality and content using the **control panel**, for example by adding extra time or changing the stage of the experience a player is on, or sending additional messages using the **messaging component**, as if from any of the experience's voices; the automated system or other players.
- **Improvisation:** The monitoring and intervention functions allow the operator to **supplement** existing content and **prototype** additional content. As a tool for rapidly prototyping game new game mechanics and exploring new scenarios, the operator can improvise new gameplay, rules and content on the fly before it is implemented as an automated system, as seen in Savannah. The operator can also respond to a player's actions, sending additional messages, performing around failures and manipulating the experience in general, as seen in Uncle Roy All Around You.
- **Distributed orchestration:** orchestration moves out of the control room through the use of dedicated street performers. **Constant**

communication ensures that all members of the production crew are aware of where players are, what they are doing and when critical events occur. A **shared map** and **player information database** allow performers to recognise players in the streets, and **face-to-face** interventions help to minimise loss of a player's engagement if something goes wrong.

Finally, experiences that involve a constant flow of public players taking part require support from the orchestration framework in order to manage entry. This is a core requirement for mobile and online players in Uncle Roy All Around You and I Like Frank, and for online players in Can You See Me Now?:

- **Induction of mobile players:** mobile players are inducted into the experience during a registration process that both introduces players to the experience and to the technology. **Registration and login** ensures that the system is operating correctly before players begin in order to maximise engagement and minimise delays. **Data capture** provides profile information for both the system and for future orchestration purposes. **Demonstration** ensures that players are confident in the use of the technology, and **introductory tasks** gently introduce players to the experience.
- **Inducting online players:** online players enter the experience via a dedicated queuing system that allows an operator to manage the potentially variable flow of online players entering the experience. Online players **accumulate** in the queue before the experience begins, and numbers can be **restricted** to ensure an appropriate balance of online to mobile players is maintained.

6.4.2 Broader Reflections

The orchestration framework deliberately provides more than an interface for monitoring the technology of a mobile mixed-reality experience; it provides a variety of tools that support the combination of pre-authored content and pre-defined game mechanics with the dynamic performance that is required to

operate an experience “in the wild”. The significance of the orchestration task and the solutions provided by the orchestration framework have been demonstrated through the successful deployment of our four experiences to a large number of members of the public, and through the variety of challenges that this has raised. This section now concludes by reflecting on the lessons that can be learnt from the orchestration framework, and identifies common solutions that can be used by other authors of mobile mixed-reality experiences and in the wider field.

Design for performance: mobile mixed-reality experiences should combine pre-authored content and live performance, and orchestration should be carefully considered to support this from the outset. While this places additional demands on the production crew, if large numbers of players are to take part in an experience each of its elements should be repeatable and sustainable, including those elements in which the production crew intervene to manipulate the experience for a particular player.

Perform around failures: regardless of how sophisticated pre-authored content may be, the production crew should, through orchestration, be able to adapt the experience in response to unforeseen situations, technology failures and a variety of player behaviours. With a large number of players, network disconnection, application crashes or misinterpretation of content should all be regarded as inevitable, and as such orchestration should allow the production crew to respond in the context of the experience as much as possible, so as to minimise loss of engagement for the player. To this end, orchestration must support the modification of existing content and game mechanics, and the seamless addition of new, improvised content.

Reveal more than just the map: many control systems and game management interfaces support orchestration by displaying where players are using a map. However, mobile mixed-reality experiences are fundamentally much more than just where players are in the city - orchestration should reveal what players are actually doing, what they have done and what they are likely to do next. Orchestration should attempt to reveal the *bigger picture*, by showing where players are in relation to key content elements, how they have

been interacting with other players and their physical actions through direct observation.

Experiences begin and end: while considerable effort may be expended in realising the core mechanics of an experience for an individual player, authors must also consider the experience as a complete production that many players take part in, and as such orchestration should plan how players enter and leave the experience – an experience that begins as soon as players arrive at the venue, both mobile and online. It is important to both introduce the system and technology to players through briefings, demonstrations and introductory tasks, and to introduce the players to the system through data capture, sanity checking and ensuring that all systems and performers are ready to receive the new player.

Players are mobile: while a large number of orchestration tasks can be performed from the control room, it is important to remember that mobile mixed-reality experiences take place on the streets of the city as well as within the mixed-reality environment. Events that occur outside of the immediate scope of the system, such as software or connectivity failures will, similarly, be largely outside of a control room operator's point of view. Gross physical actions, as well as the finer points of a player's behaviour, may be unobservable from this view point. Orchestration should be planned to take place in the same domain as the players – on the streets through physical monitoring and face-to-face interventions. A possible extension of this practice is to use other remote monitoring methods, such as CCTV, or to piggy-back on to existing web-camera feeds, such as the office view in Uncle Roy All Around You.

Share information: any distributed system requires communication between its constituent parts, and the orchestration of a mobile mixed-reality experience is no exception. Simple, reliable communication using walkie-talkies creates a collaborative awareness that allows the production crew to share a common picture of what is happening in all aspects of the experience; online, on the streets and at front-of-house, and to quickly respond if necessary in the most appropriate manner, which may be via technical performance or face-to-face interventions

6.5 Conclusions

This chapter has presented a framework of tools that support the orchestration of mixed-reality experiences, based on the deployment of the four experiences described in chapter 3. It has shown how orchestration presents a significant challenge for the authors of experiences that are presented in a real-world scenario.

It has reviewed the use of orchestration in a variety of areas, from role-playing games to public performances and control rooms, to show how orchestration is a combination of monitoring, intervention, improvisation and performance, and to raise the question as to how orchestration can play a useful role in mobile mixed-reality experiences.

In response to this, this chapter has described a framework of orchestration tools and functions that allows a behind-the-scenes production crew to manage the operation of our four experiences, by combining dynamic performance and pre-authored content through the use of dedicated control room tools and work practices. It has shown how the orchestration framework has played a critical role in the day-to-day operation of the experiences as well as when things go wrong, as highlighted by the general framework proposed in chapter 4.

Finally, this chapter has reflected on how the orchestration framework has highlighted a number of more general solutions for the orchestration of mobile mixed-reality experiences.

7 Implementation

7.1 Introduction

This chapter describes the implementation of our four experiences. It shows how the authoring and orchestration tools described in the previous two chapters are implemented within a larger supporting system, and in turn how the challenges of authoring and orchestration influence the design of this system.

The chapter begins by reviewing the core technologies that have been used to implement our four experiences; focusing on connectivity, middleware and user interfaces, and describing the strengths and weaknesses of each in order to highlight the technical challenges that these raise for each implementation; how connectivity adds additional challenges for orchestration, how middleware should be used to implement a platform that supports orchestration, and how content is presented to players in a professional manner.

The chapter then goes on to describe a component-based system architecture and the primary functions that each mobile mixed-reality experience supports, before focusing on how the architecture has been implemented, particularly the differing requirements of online and mobile clients, supporting remote orchestration and disconnected operation.

7.2 Technologies

This section reviews those technologies that have been used to implement our four experiences, and that are also often used to implement other, similar experiences. They can be divided into three areas; networking technologies used to connect mobile devices, middleware that is used to connect system components and applications, and multimedia authoring technologies used to create and deploy player-facing user-interfaces.

7.2.1 Connectivity

In our four experiences, players inhabit a shared, mobile mixed-reality environment, in which they receive location-based content and are aware of the

positions and actions of other players. In order to support this, the experiences make extensive use of wireless networks to connect mobile players' devices to a central server as they move through the streets, providing a real-time view of the shared mixed-reality environment. This wireless connectivity can be provided by one of a number of technologies – often Wi-Fi, GPRS or 3G, with each presenting its own particular strengths and weaknesses.

Wi-Fi, or wireless LAN, is commonly used to provide wireless connectivity for mobile devices over a small area, and is popularly used to support the majority of small scale mobile experiences, including Savannah and early deployments of *Can You See Me Now?*, due to its ease of deployment and use. A Wi-Fi network transparently provides the same functionality as a wired LAN connection in terms of network addressing and routing, combined with relatively low latency and high speeds when compared to other wireless technologies. However, the quality of a Wi-Fi connection depends on a variety of factors, such as radio interference from other wireless networks and microwave sources, the number and power of access points and obstacles between the access points and mobile devices, for example buildings or other urban obstructions.

While there are a large number of publicly accessible Wi-Fi networks, coverage by one network across a large urban area is rare, and existing access points often may not be placed to support this explicitly, and a connection in any given area cannot be guaranteed to any degree. Conversely, as a technology readily available off-the-shelf to the consumer, it is possible for third parties to set up their own Wi-Fi network as required, and this has the advantage that the experience author has complete control of the placement of access points in order to provide the best coverage, and also that the network is not subject to any operator imposed restrictions such as port blocking or billing. However, the time and effort required to set up a Wi-Fi network to cover a reasonable urban area for a mobile mixed-reality experience is large. In Savannah, one access point was sufficient to reliably provide wireless connectivity across the school playing field, whereas to provide wireless connectivity across a built-up urban area of approximately one square kilometre in *Can You See Me Now?* required several roof-mounted access

points, with each in turn requiring power and connectivity back to a central network hub. Any Wi-Fi network cannot be expected to provide seamless coverage of the desired area; even with careful planning of access point locations there may be areas that are not covered or provide less coverage than expected due to transient interference, leading to dropped or marginal connections, and ultimately regular disconnection from the mixed-reality environment whilst mobile.

GPRS, or General Packet Radio Service is a data service that uses the GSM network to provide internet access for mobile phones and other GSM compatible devices, for example a PDA with a suitable modem. GPRS was used to provide wireless connectivity for Uncle Roy All Around You and later versions of Can You See Me Now?. A GPRS data connection provides a speed similar to a conventional dial up connection with a typical latency of 500 to 1000ms. The most significant advantage of using GPRS is its use of the near ubiquitous GSM network, meaning that the service may be used by a mobile device in most locations without the overhead of setting up a dedicated infrastructure as with Wi-Fi, however this use of the GSM network also leads to a number of disadvantages which compound the relatively slow speed of the connection when compared to Wi-Fi.

The GSM network infrastructure is made up of many base stations, analogous to Wi-Fi access points, and each can handle a limited number of concurrent transmission channels. As GPRS is a packet switched system, several GPRS sessions may concurrently use one transmission channel of those not being used for regular mobile voice traffic. However, network operators often prioritise voice over GPRS traffic, meaning that when attached to a busy base station multiple GPRS clients must compete for a limited amount of the base station's capacity. As a result, speed and reception may be poor due to this contention. As with other wireless protocols, the quality of a GPRS connection drops logarithmically with distance from the base station that a device is attached to. In a densely populated area with many base stations this problem is less apparent, however in an area with only a few base stations the performance of the connection may degrade significantly based on location, especially if a base station is not operational. Finally, network operators often place

restrictions on the traffic that may flow over a GPRS connection; for example by employing transparent caches to heavily compress some media, or requiring that all traffic is routed through a gateway using Network Address Translation and so limiting true end-to-end connectivity, in turn introducing restrictions on an application using stateless network protocols. GPRS provides convenient and widespread wireless connectivity, however again the availability and quality of the connection cannot be guaranteed, potentially leading to repeated disconnections from the mixed-reality environment.

3G refers to third-generation mobile phone networks, and encompasses a number of rival technology standards that provide fast wireless connectivity in comparison to 2.5G networks, a colloquialism for networks that support GPRS. 3G was used to provide wireless connectivity to mobile players in *I Like Frank*, in which players used 3G phones to inhabit the mixed-reality environment. The main technological advantage of 3G over 2.5G is support for video telephony, and its ability to handle greater numbers of simultaneous voice and data customers, and data transfer at speeds that approach those of home broadband connections. In practice, 3G operates in a similar manner to GPRS, with video telephony requiring dedicated transmission channels in the same way as voice telephony, therefore often taking priority over data transmission. As a relatively new technology, 3G coverage tends to be limited to urban areas due to the service requiring a new infrastructure roll-out by network operators. As a result 3G networks tend to be hybrids that support both 3G and 2.5G standards, switching down in an area with little or no 3G coverage. In terms of contention, routing, and to a larger extent, coverage, 3G suffers from similar problems to GPRS in being unable to guarantee that wireless connectivity will be available consistently in all areas.

For mobile players to be able to inhabit the mixed-reality environment, then, the supporting system for each experience must take into account the properties of the wireless network that it relies upon, primarily regarding temporary disconnection due to fluctuations in signal strength, and longer disconnections due to contention and lack of coverage.

7.2.2 Middleware

Our four experiences make extensive use of two existing middleware platforms – software to connect a variety of software components - in order to simplify the task of creating a distributed system to support the mobile mixed-reality environment. *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank* are all built using *Equip*, and *Savannah* is built using *Elvin*, both of which provide similar data-sharing functionality.

Equip (Greenhalgh, 2002) is a real-time middleware platform that supports data-sharing between distributed applications, and is designed to support the convergence of real and virtual worlds by allowing a diverse set of devices to communicate. This enables the creation of a shared mixed-reality environment by sharing the state of a master environment between mobile devices, online clients and a component-based central server.

Equip enables data sharing through the use of a number of named “dataspaces”, referenced by a URL-like naming scheme. Applications connect transparently to one or more of these dataspaces, either locally or over a network, which act as tuple spaces allowing data objects to be shared. *Equip* implements the tuple space paradigm by allowing applications to act as producers by adding, updating or deleting items within a dataspace, or as consumers by adding a template to subscribe to events regarding a particular data object that they are interested in. This subscription model drives data replication through *Equip*, with the dataspace informing the relevant application that something of interest has occurred. The replication of events between the potentially remote dataspace and the application-level API is handled internally by *Equip*'s object serialisation protocol over either TCP or UDP.

Elvin (Fitzpatrick et al., 1999) is a lightweight message based notification system designed to facilitate communication between distributed applications. Applications use *Elvin* in a similar manner to *Equip*, using the *Elvin* API to connect to one or more *Elvin* routers – analogous to dataspaces - and to publish and subscribe to particular messages. Replication between processes is again

driven by subscription and templates, with applications registering their interest in a set of messages using wild cards or specific values.

Equip dataspace and Elvin routers can be used as a central data bus that forms the backbone of the supporting system, which can then be built from a number of potentially reusable, logical components, with the middleware platform handling the communication between these components. In some cases it may not be appropriate to use the internal serialisation and network communication protocols of the middleware system to connect remote clients – for example over a high-latency GPRS network with port restrictions, or to an internet based online player, and in these instances a proxy application using a more dedicated protocol must be created to route events between the middleware system and the application.

7.2.3 User Interfaces

In our four experiences, the user interfaces that players use to access the mixed-reality environment and the content provided within it are created using industry standard interface authoring packages, for both ease of deployment and to allow the creation of a polished interface for players to interact with. Secondly, separating the core functionality of the supporting system from its user interfaces allows the task of constructing the interface to be farmed out to a specialist designer.

Flash is a multimedia authoring system created by Macromedia (Adobe Systems, 2007a), which is commonly used to create games, animations and other interactive media content to be embedded in web pages. Flash applications, or movies, are authored in a dedicated IDE, and support both animation and scripting using Action Script. Movies are exported as byte-code which is run in the Flash Player virtual machine, usually embedded as a plug-in within a web-browser. Flash movies may be run on mobile devices and increasingly on mobile phones, making Flash an increasingly popular choice for creating interfaces for mobile experiences.

A Flash movie can communicate in real time with a remote application over a network using the TCP protocol, which is exposed as an XMLSocket object

within Flash. However, this wrapping only exposes limited functions of the underlying socket, with operations limited to instructing the socket to connect and receiving call-backs on disconnection and data transfer. Secondly, and perhaps more importantly, a complex Flash movie may struggle to run at a reasonable speed on a resource-limited device such as a PDA, and this combined with limited network support means that a Flash interface is probably best supported on a mobile device by a native host application.

Shockwave is a second multimedia authoring system created by Macromedia. While it can also be used for creating animations and interfaces, it is often used for creating online games. As with Flash, Shockwave movies are developed using the dedicated Director IDE (Adobe Systems, 2004a), which are then played with the Shockwave virtual machine plug-in. Shockwave provides a more sophisticated platform than Flash in that it supports extensible network connectivity, audio streaming, and provides access to hardware accelerated 3D graphics, allowing authors to embed rich 3D content into movies; however this sophistication means that it is largely limited to standard PC web-browsers, rather than mobile devices. The Shockwave plug-in is said to be installed on the computers of 59 percent of web users (Adobe Systems, 2007c), however it may also be downloaded and installed via a web-browser as required, making it an ideal platform for the deployment of the online components of a mobile mixed-reality experience.

An important addition to Shockwave is the Multi-User Server Xtra (Adobe Systems, 2004b), which allows real-time network communication via TCP or UDP using a proprietary wire protocol to a remote server; functionality which is typically used to create multi-user applications such as chat-rooms or online games. The Multi-User Server communication protocol is message based, with clients and server sending and receiving messages that contain serialised collections of Lingo types – the scripting language used within Shockwave.

7.3 Implementation

This section now describes the implementation of our four experiences. It begins by describing the common game-engine architecture that supports the core functions of the experiences and integrates the ColourMaps system, and

upon which all four experiences are based. It then describes the implementation of particular components to support different types of players, orchestration tools and especially disconnected mobile players.

7.3.1 ColourMaps and the Game Engine

The system architecture, shown in figure 7-1, is based on a conventional client-server model, with a central server maintaining the state of the experience and to which a variety of clients connect, including player clients and supporting orchestration tools. A peer-to-peer architecture was not considered for the experiences due to their short nature, and the need to support synchronous communication between clients that would not usually be expected to peer over the course of an experience – however, this architecture would perhaps be more appropriate for a longer experience that lasted for days, or weeks rather than just one hour, and timely message exchange was not a critical requirement.

Each client publishes updates over the network to the central server regarding its current activities – primarily position updates, audio and messages for players, and each client subscribes to data from the server regarding the evolving state of the mixed-reality environment and to receive content. Clients render the state of the environment and display content in the appropriate manner; as a three-dimensional model or two-dimensional map for online players or typically a two dimensional map for mobile players. As players inhabit the mixed-reality environment their clients share position, text messages, audio, content and other game mechanics via the game-engine, which sits between the **nonValidated** and **validated dataspace**s and filters and moderates these events:

Position: Each player client publishes a position object into the nonValidated dataspace, which is updated with their current position as it becomes available, for example a mobile player's position as reported by GPS or self-reported positioning, or the position of an online player's avatar as it is moved in the virtual environment. Subscribing to updates to this object, the game-engine converts these positions into a common coordinate frame as required, updates its own internal state of where the player is and then publishes a corresponding

position object in the validated dataspace where it is available for other clients to subscribe to. The game-engine publishes a corresponding position object for each individual player, and also an aggregate object that is updated once every few seconds containing positions of all of the players of a particular type, for mobile clients that have limited bandwidth or difficulty handling large numbers of updates. Using this system, all clients subscribe to the positions of all other clients, and as a result can display the positions of players in the mixed-reality environment.

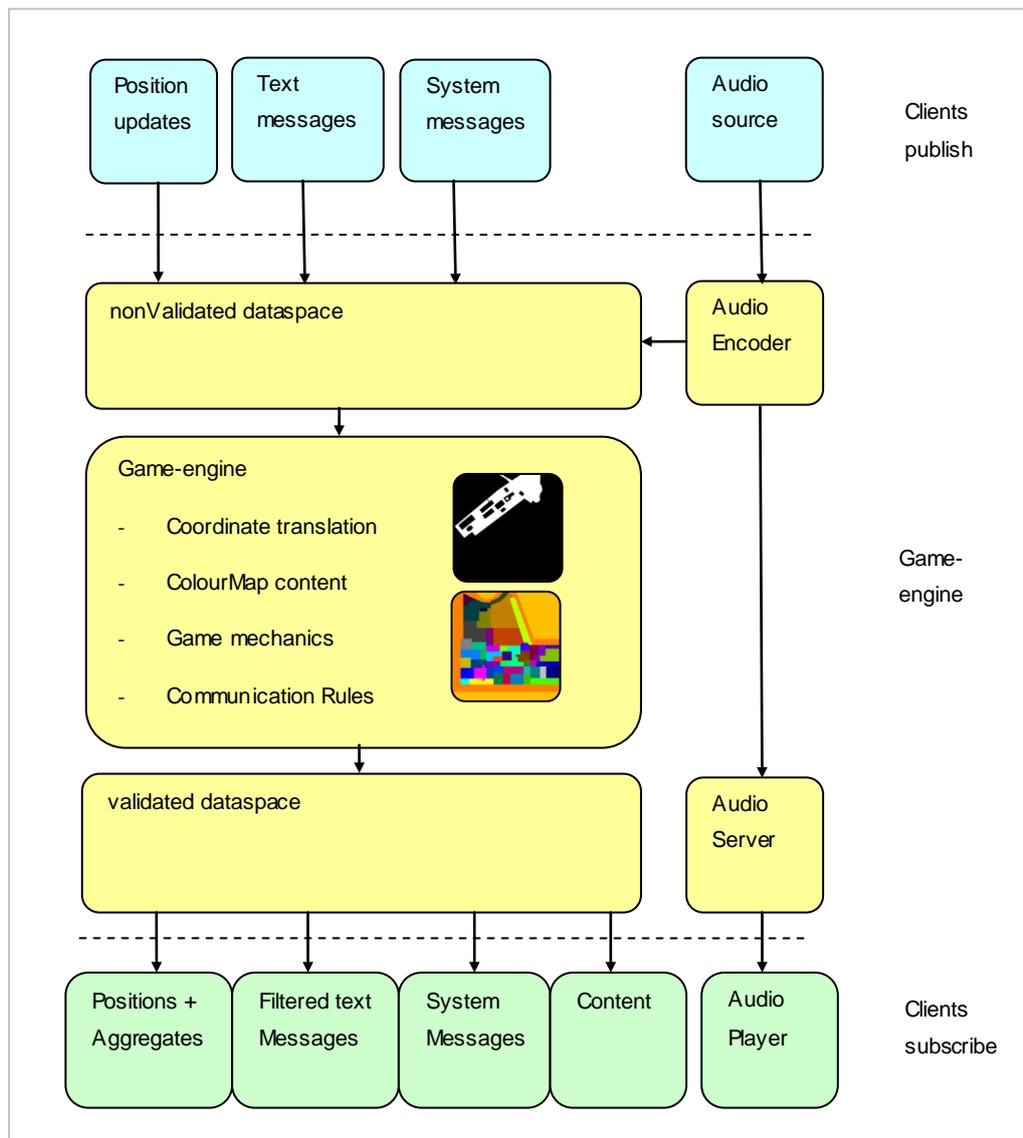


Figure 7-1: Game-engine architecture

Messaging: Each client publishes a messaging object into the nonValidated dataspace which it updates in order to pass messages to the game-engine. The game-engine publishes a matching message object into the validated dataspace

which the client subscribes to, allowing the game-engine to send messages specifically to that client. The game-engine also publishes its own messaging object, allowing it to broadcast messages to all or a subset of the connected clients. This architecture provides a transparent messaging framework allowing the clients and the game-engine to communicate which abstracts the handling of individual connections to clients to other parts of the system, allowing the game-engine to be easily extended regardless of how clients are actually connected. The messaging framework is used to send system notifications, text chat and other experience specific events to and from the game-engine.

Text Messages: Clients use the messaging framework to send text messages to the game-engine, which then distributes them to other clients as is appropriate for the experience; public chat messages are broadcast using the game-engine's messaging object to all players, whereas private text messages are sent to the specified player using that player's own messaging object.

Audio: Mobile players can communicate with online players using a continuous audio stream in *Can You See Me Now?*, and with short audio messages in *Uncle Roy All Around You* and *I Like Frank*. A dedicated audio process receives, encodes and broadcasts audio for online clients, and notifies clients through the game-engine using the messaging framework when audio is available to listen to. In *Can You See Me Now?*, mobile players communicate over a shared walkie-talkie channel which is captured and dynamically encoded as MP3, before being continuously streamed to online clients from a dedicated servlet. In *Uncle Roy All Around You*, audio is recorded locally on the mobile player's device and encoded using standard AMR (3GPP, 1998) to reduce size. The audio file is uploaded to the audio process before being re-encoded as MP3 and made available on the web-server for clients to download and play. Finally, in *I Like Frank* the mobile player's phone makes a phone call to one of a dedicated array of phones in the control room which records a short audio clip before hanging up. Again, this audio file is re-encoded as MP3 and is made available on the web-server. In *Can You See Me Now* the audio process introduces a delay of approximately five seconds, and in both *Uncle Roy All Around You* and *I Like Frank* there is a delay of approximately ten

seconds from a player finishing recording a message to it being heard by online players.

ColourMaps and Game Mechanics: The game-engine manages experience-specific game mechanics, interactions between players and serving location-based content to players. Content and other global game-state variables are loaded from ColourMaps, as described in chapter 5. When an online player connects, the game-engine assigns a random start position using a ColourMap, and broadcasts the event to all existing clients. In *Can You See Me Now?*, when a position update is received from a mobile player, the game-engine uses the GPS ColourMap to correct the position if necessary before checking to see if the mobile player is close enough to an online player to have caught them, and if so sends messages accordingly and begins a timer to disconnect the player. Likewise, in *Uncle Roy All Around You* and *I Like Frank* the game-engine determines whether each mobile player is at the correct stage in the experience to be visible to online players, and if so whether online players are close enough to a mobile player to be able to send text messages to them, and it modifies the position and messaging objects accordingly. In *Uncle Roy All Around You*, *I Like Frank* and *Savannah*, the game-engine manages time and location-based content, using the translated position updates with the appropriate content ColourMap to send content events back to the client using the messaging framework.

7.3.2 Supporting Varied Client Requirements

As we have seen, clients connect to the central game-engine in order to inhabit the mixed-reality environment of an experience. Online players use a Shockwave client in *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank*, and mobile players use a hybrid native and Flash client running on a PDA in *Can You See Me Now?*, *Uncle Roy All Around You* and *Savannah*, and a Java Midlet running on a phone in *I Like Frank*.

The online client is a self contained Shockwave movie embedded in a web-page and hosted on the web-server, which enables deployment to a large number of online players without the need to install dedicated software. The client contains a three-dimensional model of the city that mirrors the maps and

ColourMaps used elsewhere in the system, and the player moves their avatar through the model using a matching physics model that enforces a certain speed and collisions with relevant physical boundaries. The client publishes the position of its avatar once a second, and listens for position updates for other players published by the game-engine; the positions of their avatars are then smoothly interpolated using these updates and rendered using a walking animation. Dynamic audio from mobile players is played using Shockwave's built-in streaming audio player, mixed with static sound effects that are played as necessary to alert the player to key events. Finally, the client listens for messages from the messaging framework and performs the appropriate action; displaying text messages in a variety of text fields, or changing to a different mode, for example the end-of-game scene when the player is caught in *Can You See Me Now?*, or the office scene in *Uncle Roy All Around You*.

The mobile client consists of a Flash interface combined with a native C++ application that runs locally on the player's device. The Flash movie acts as a thin rendering client for displaying the map, online players and other content, and passes interface events, for example position reports and requests for audio recording in *Uncle Roy All Around You*, to the native application via a local socket connection. The native application maintains communication with the central game-engine, forwarding events and messages to the game-engine or handling them locally as appropriate, for example recording and uploading audio messages as requested by the Flash interface and reading and decoding GPS position reports if the experience requires it. It passes rendering instructions to the Flash interface regarding incoming messages, content or the positions of other players.

As described previously, distributed middleware systems such as Equip and Elvin greatly simplify the task of creating a component based game-engine, and when using a high-speed wireless network in an open area, such as the playing field in *Savannah*, can be used to directly connect clients to the central server. However, when using a potentially patchy, low-bandwidth wireless network in an urban environment their internal communications protocols may not be best suited for connecting clients. Similarly, the networking capabilities of a user interface platform may be limited in their abilities to cope with such a

network – as in the case of Flash – or specifically designed to communicate with a server platform produced by the same vendor – as in the case of Shockwave. For this reason connections from clients are further abstracted from the game-engine dataspace using dedicated proxies which handle connections from particular clients in a more appropriate manner, as shown in figure 7-2.

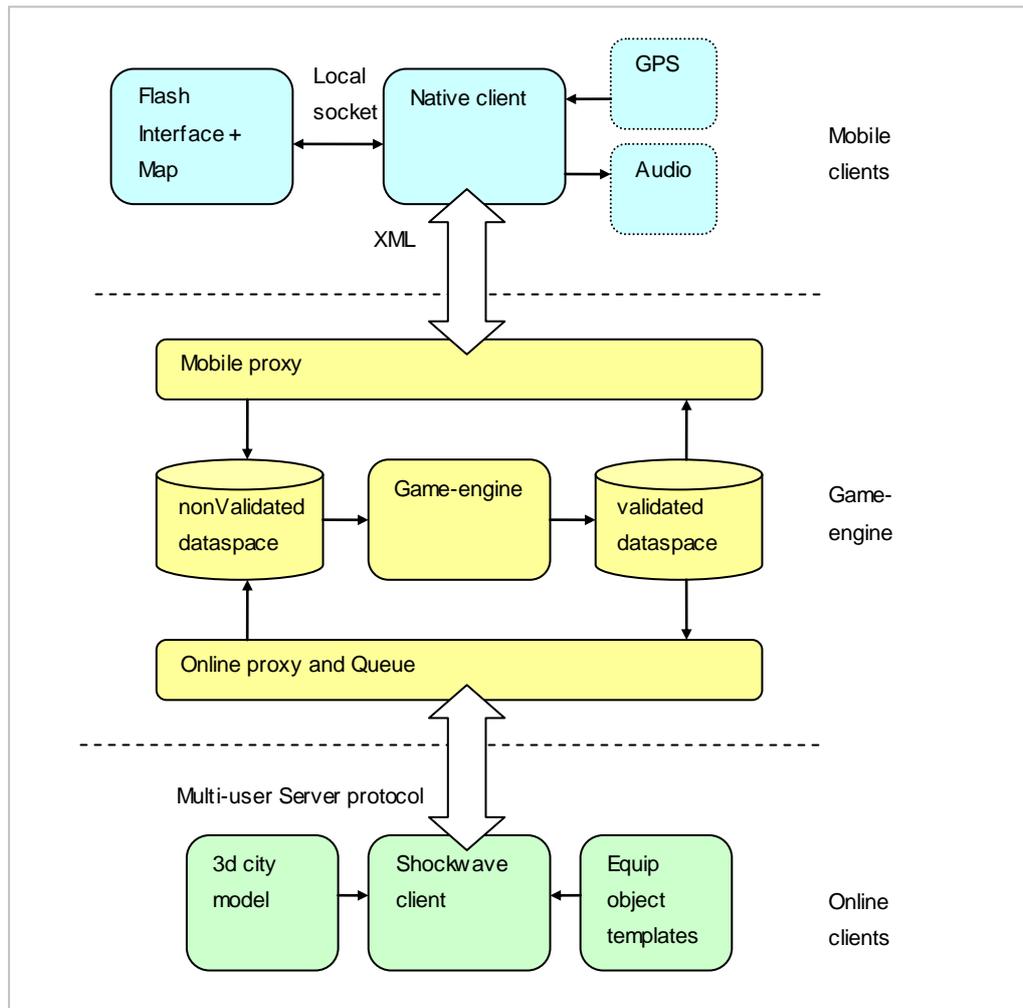


Figure 7-2: Dedicated proxies for online and mobile clients

For online clients a reusable proxy provides a mechanism that allows Shockwave to connect to Equip dataspace using the standard Multi-User Server Xtra, and is used to support both the queue and access to the game-engine. The proxy provides both generic Equip and Multi-User Server functionality, by presenting an implementation of the Multi-User Server messaging protocol while also making use of the Equip API, enabling Shockwave to interact with Equip and vice-versa. The proxy is a multi-

threaded TCP server application written in Java that runs alongside the game-engine, and one instance supports connections from multiple Shockwave clients. It implements the Shockwave protocol and basic types, allowing it to read and write the binary message format used by clients.

The online proxy uses introspection to construct nested Lingo property lists from Equip objects so that they can be passed to Shockwave, using meta-data to maintain class type information as well as field names. Conversely, Lingo structures constructed by the Shockwave client are de-serialised using the stored meta-data to inform the Java class loader, enabling the client to create Equip objects. The proxy connects to the nonValidated and validated dataspace, and Shockwave clients access Equip functionality by specifying the required function in the subject line of an incoming Multi-User Server message; adding, updating or removing a listener template to receive events from the validated dataspace, or adding, updating and removing an item from the nonValidated dataspace. By defining template Lingo structures based on Equip object IDL definitions and constructing the appropriate objects, the client can access the dataspace as normal.

Finally, the proxy publishes information about its current status into the dataspace, including how many remote clients are connected and active, allowing the game-engine or other orchestration tools to monitor it, or to control it by publishing certain control messages using the messaging framework.

For mobile clients, a similar proxy provides access to the game-engine dataspace using a light-weight XML protocol in order to support high-latency, limited bandwidth or firewalled networks – particularly GPRS and 3G. As before, the mobile proxy is a multi-threaded TCP server application written in Java that runs alongside the game-engine, and one instance supports all of the mobile clients in connecting to the dataspace. In this case, however, rather than providing completely generic access to the dataspace the mobile proxy handles the task of subscribing to particular events and publishing and updating objects, to minimise complexity and network communication for clients. When a client connects to the proxy, the proxy creates position and messaging objects

for the client, and subscribes to the relevant updates on the client's behalf. Updates and messages are marshalled as XML and sent to the relevant client, and similarly XML updates from clients are un-marshalled and the relevant Equip objects updated.

7.3.3 Integrating Orchestration Interfaces

As the game-engine is abstracted from the clients' connections to the two core dataspace, and all data is published into the appropriate dataspace rather than directly sent to the game-engine and the clients respectively, orchestration clients can connect to the dataspace to subscribe to and manipulate data in the system without the need to create separate interfaces to the game-engine and to clients. Figure 7-3 shows how the orchestration tools are supported in the system.

The control-room orchestration interface described in chapter 6 subscribes to the validated and nonValidated dataspace as an observer in order to directly monitor incoming updates from clients and outgoing events generated by the game-engine. Each of the supporting proxies publishes information regarding current connections into the nonValidated dataspace, which the orchestration interface monitors in order to display the connection status and connectivity history of each player. Similarly, the orchestration interface monitors the validated dataspace for filtered and translated positions of players, in order to display the map and associated spatial trajectories. It also monitors all messages published by the game-engine, in order to display the message and content history for each player.

The orchestration interface publishes generic position and messaging objects to both dataspace, which it can manipulate in the nonValidated dataspace to emulate any given player or system in order to insert updates to be handled by the game-engine, or in the validated dataspace to bypass the game-engine and send events directly to connected clients. While game-engine variables such as the capture radius and immunity time for online players in Can You See Me Now? are initially loaded from a configuration file, the game-engine listens for special control messages published by the orchestration interface, allowing these variables to be modified while the game-engine is running.

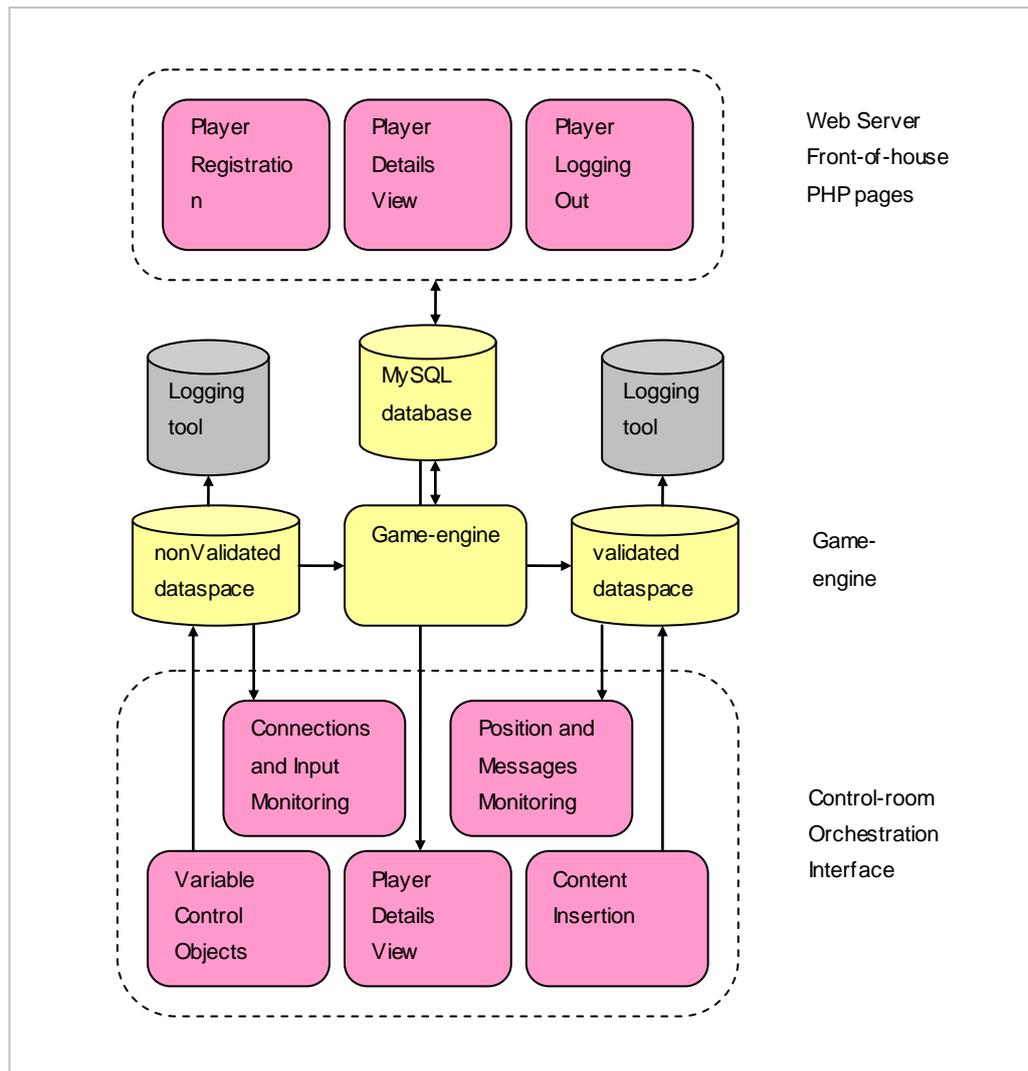


Figure 7-3: Integrating the orchestration framework

The front-of-house orchestration interface is presented by a web-server and backed by a MySQL database. A set of dynamic PHP web-pages hosted on the web-server allow the front-of-house performers to create player records in the database with associated descriptions, to view currently active players and to log out players who have returned. These pages are also available remotely for street-performers. When a mobile client connects and identifies the player, the game-engine queries the database to check that it is a correctly registered player, and to retrieve the player's details and game state. Similarly, when a player is logged out at front-of-house the database is updated to reflect this, and the game-engine removes the player from its active state.

Finally, a logging tool connects to each of the two dataspace and captures all of the events that occur within each dataspace to a file, which can be parsed at a later date for analysis or to completely replay an experience.

7.3.4 Experiencing Disconnection

One of the most significant challenges that arose during the first outings of our four experiences was the transient nature of mobile players' connections to the central game-engine, which manifested itself as patchy or intermittent connectivity while using wireless networking, GPRS and 3G. As a fundamental element of the experiences, loss of connectivity for mobile players is a serious concern.

In *Can You See Me Now?*, as mobile players moved through the city they were often frequently disconnected completely from the wireless network, or the signal strength dropped to a level that maintaining a connection between the mobile client and the game-engine was difficult. This resulted in a mobile player being unable to take part in the game until they moved into an area with better wireless coverage – the performers learnt anecdotally where good and bad areas were. These periods of disconnection were not critical to the experience as a whole, as the mobile player could perform around the outage using audio chat while moving and there were multiple mobile players to continue the chase, however online players experienced the disconnection as the avatars of mobile players freezing, disappearing or suddenly appearing in a new location upon reconnection.

Savannah was relatively unhampered by network disconnections, as the open nature of the playing field upon which it was played gave direct line-of-sight visibility to the wireless access points. Although the experience required mobile players to be connected to receive content, the mobile client only displays new content as the player moves between regions, meaning that a temporary disconnection was unlikely to be observed by the player as the game-engine would send the correct content on reconnection and receiving the latest position update.

The first outing of Uncle Roy All Around You was by far the most susceptible to disconnection due to its initial software design and implementation. Each mobile client required a constant connection to the game-engine to receive content and enable communication with online players. Although GPRS coverage was considered to be ubiquitous across the area – central London – with numerous base-stations in the immediate vicinity, it was common for a client's connection to fail in a variety of ways; the TCP connection to the game-engine would stall for several minutes due to heavy packet loss; the TCP connection would fail and yet the client would not be aware of this fact due to long TCP time-outs; or the underlying GPRS modem connection would disconnect due to poor signal strength and require manual redialling and reconnection. I Like Frank suffered from similar connectivity issues while using a 3G phone. Disconnections were compounded by the 3G infrastructure, which required existing data connections to be completely torn down in order to make voice or video calls, and often failed to be able to reconnect afterwards, requiring the phone to be reset before the player could continue receiving content from the game-engine.

The majority of mobile players in Uncle Roy All Around You and I Like Frank experienced network disconnection that disrupted their experiences. Analysis of the game-engine logs for the first outing of Uncle Roy All Around You showed that of the 272 mobile players that took part, over a third suffered a disconnection that necessitated intervention by a performer in order to restart the mobile client. On average one of these more serious disconnections would result in the mobile player spending 10 minutes disconnected from the experience. On average each mobile player would suffer from three or four disconnections that did not require intervention, but that rendered the player disconnected for a few minutes each time while client attempted to reconnect or repair the connection.

The effect of these disconnections on a mobile player's experience was severe. Given that the player required a connection to the game-engine for the core functions of the mixed-reality environment and to receive content, disconnection left them stranded with no recourse; player feedback indicated that this was one of the most frustrating parts of the experience as affected

players felt that the system had failed or was subsequently viewed as unstable. Furthermore, the disconnections added greatly to the orchestration task in identifying, managing and responding to these disconnections, both for street performers in finding and intercepting players, and for the control room in bringing manipulating content for affected players to recover from the disconnection.

(Kureshy, 2004), writing on how to develop a mobile application for use in a business environment, states that: *total coverage isn't a guarantee. Users are going to need to use applications when their modem or network card is unavailable [...] Connectivity can be the best-case scenario for an application but should not be required for using the application.* This assertion is equally applicable to mobile mixed-reality experiences – it is clear that when an individual player's experience critically depends on maintaining constant communication with the game-engine, and yet this cannot be guaranteed, that the implementation should be redesigned with this in mind. To this end, after network disconnections led to severe disruptions for many players in the first outing of Uncle Roy All Around You, the implementation of the experience was changed to explicitly support disconnected operation, to reduce the effort required to orchestrate the experience and to minimise disruptions for players.

7.3.5 Supporting Disconnected Operation

Many computer games, and for that matter the first implementations of our four experiences, take a centralised approach in providing a connected experience, by maintaining the core logic of the game within a central game-engine to which remote clients send update requests in order to receive subsequent events, for reasons of concurrency and to minimise cheating by players. As we have seen, this approach can similarly be adopted by mobile mixed-reality experiences, as a thin client poses fewer demands on a resource limited device such as a phone or a PDA, and a central game-engine is easier to maintain and orchestrate. In online games, disconnection from the central game-engine is considered to be a rare occurrence due to comparatively reliable wired networks, and as such developers focus on minimising network traffic and delays. From our early experiences it is apparent that this view

should not be adopted by mobile mixed-reality experiences, and that disconnections from the wireless network during an experience cannot be treated in the same way.

Supporting disconnected operation involved a threefold redesign of the implementation of Uncle Roy All Around You, in order to firstly accept disconnection and to design the implementation of the experience to support disconnected operation from the ground up, to deliberately reveal the effects of disconnection to players during the experience so that they could continue confidently, and to harden the implementation to minimise disconnection should it occur.

The second implementation of Uncle Roy All Around You assumes that disconnection from the wireless network is a normal and anticipated aspect of its operation. The experience consists of two parts; a disconnected single-player experience and a connected multi-player experience which are combined when the wireless network allows connectivity. The richest, and most desired, experience available for a mobile player requires the client to be connected to the game-engine, which is necessary to communicate with online players and to receive improvised content through the orchestration process. However the other fundamental parts of the experience, receiving location-based content from Uncle Roy, navigating the clue-trail and experiencing the office, do not require the client to be connected. The migration of the ColourMaps system from the central game-engine to the mobile client allows the mobile player to continue with the experience, albeit in a reduced form. Furthermore, as the ColourMaps system and content still facilitate the movement of the player through the city, if a disconnection has been caused by poor signal strength in a particular location then the player can continue to move, potentially to a location where reconnection may be more forthcoming.

Figure 7-4 shows the new architecture for the mobile client and game-engine. The single-player functionality of the experience that was previously implemented in the central game-engine is moved to the native C++ half of the mobile client, which is now responsible for serving location-based content and time-based narrative to the Flash interface using the content ColourMaps and

associated XML data-files, which are now hosted locally on the device. When the mobile player reports their position using the map interface, the position report is sent locally to the native client, which responds with the appropriate messages from Uncle Roy, including all time-based content and introductory red targets. The client stores the state of the player's experience locally using the registry, so that if the device needs to be restarted then the player can continue where they left off.

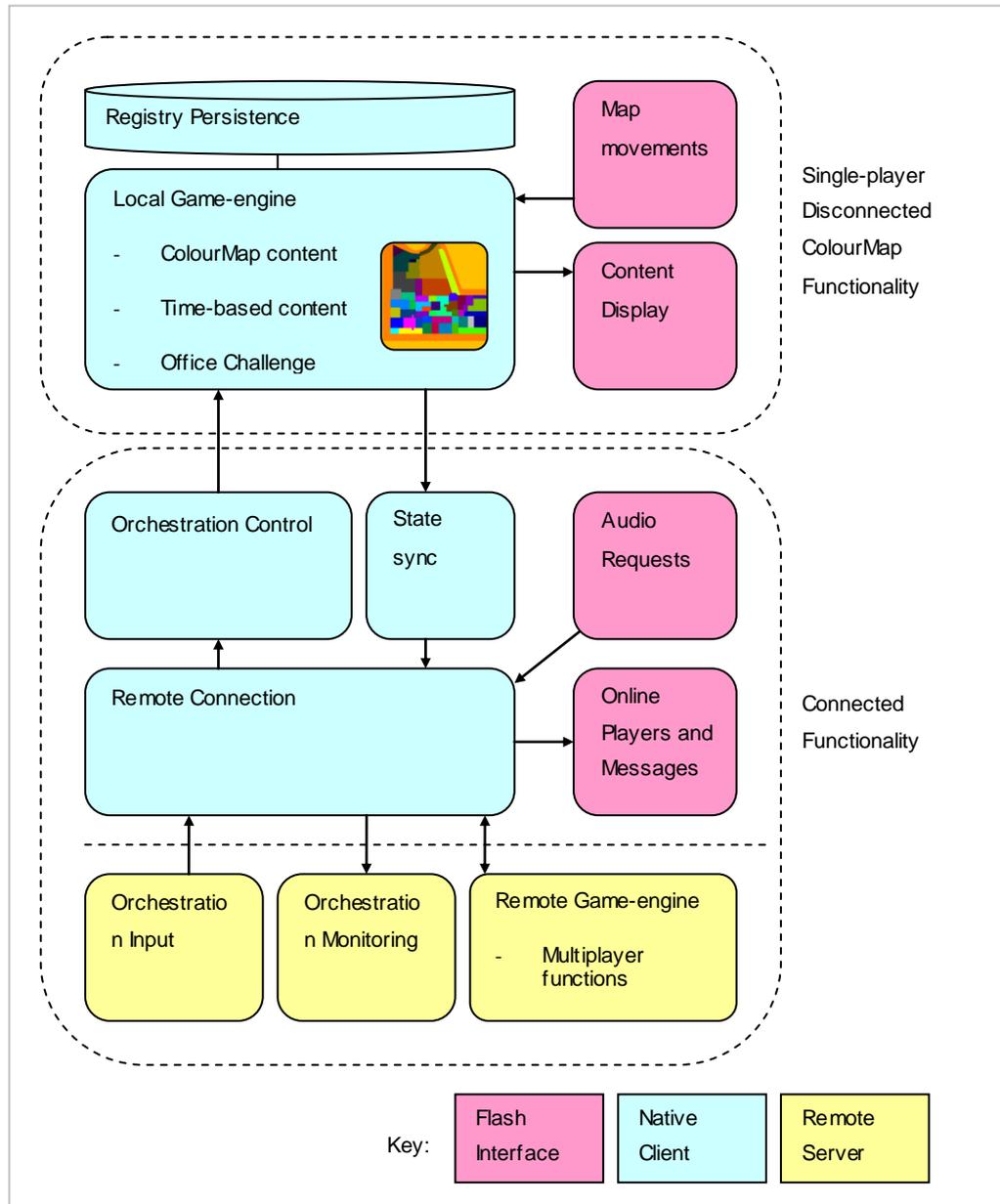


Figure 7-4: Revised architecture to support disconnected operation

In the original implementation, when mobile players arrived at the end of the experience, Uncle Roy's office, a member of the control-room staff would

trigger the timed slide-show sequence of instructions on the player's interface remotely using the orchestration interface, once the player had been positively identified by a performer. However, it now has to be assumed that the mobile client may not be connected when the player arrives, rendering such a remote “upgrade” impossible.

To overcome this, a challenge sequence is introduced to the interface to allow a mobile player to upgrade themselves, allowing all aspects of the mobile experience to be completed whilst disconnected. The sequence, shown in figure 7-5, is triggered when the player declares at the location of the office, and prompts them to choose a particular figure in a photograph, the answer to the challenge being physically displayed at the entrance to the office – ensuring that the player cannot prematurely upgrade before they are in the correct location.

The interface is modified to explicitly expose the state of the network connection to the mobile player, and to indicate those functions that are not available while disconnected, as shown in figure 7-6. A status bar indicates whether the client is connected, disconnected or currently attempting to connect. When disconnected, the button used to record an audio message is disabled and labelled as such, and online players are removed from the map, to ensure that the mobile player does not attempt to use a function that will fail due to disconnection, or to communicate with online players who may have moved.

Finally, connected multi-player functionality is layered on to the single-player experience when available. After a period of disconnection, the game-engine's state is synchronized with that of the mobile client. When connected, the client relays its actions to the game-engine, including position updates, content as it is triggered and the player's progress through the experience. Similarly, remote orchestration tools can be used as normal to directly supplement content and to manipulate the player's experience.

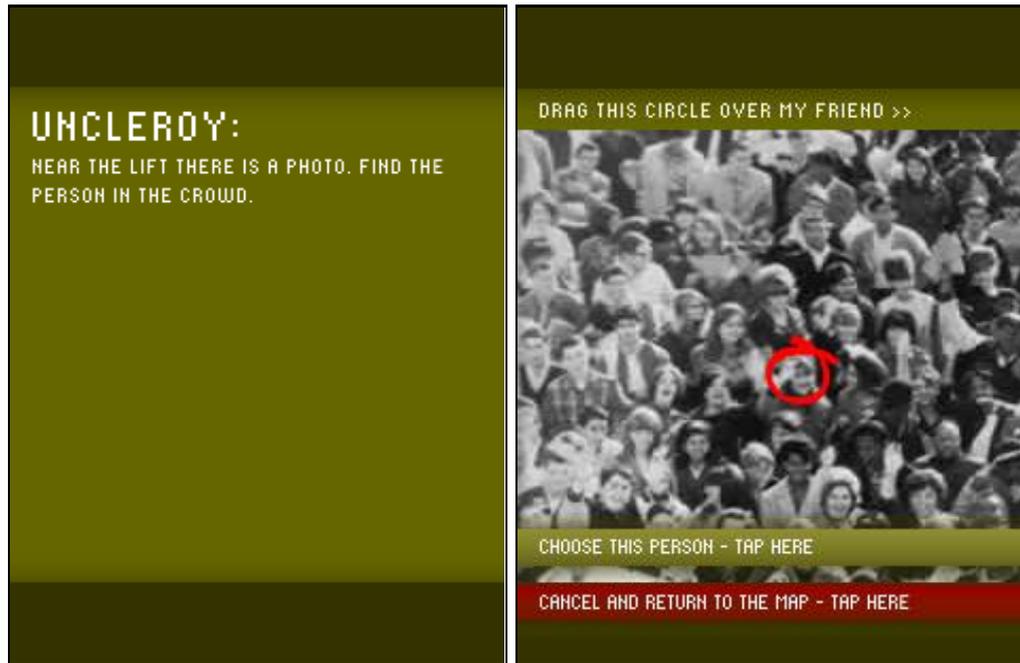


Figure 7-5: Disconnected office challenge



Figure 7-6: Disconnected interface, left, connected interface, right

7.3.6 Minimising Disconnection

The final stage of the new implementation is to minimise network disconnection through the architecture of the supporting software platform, as the most desirable state is to have mobile players connected so that they can

inhabit the mixed-reality environment and interact with online players, and be available for remote orchestration.

The mobile client implements a four-layer communications platform to enable connectivity with the central game-engine:

- **GPRS:** The GPRS layer controls the GPRS modem in the PDA, which connects to the mobile phone network and obtains an IP address. It is controlled by the mobile client, which can request it to dial and hang-up the connection, and can also give the status of the GPRS connection.
- **TCP:** The TCP layer creates a connection to the mobile proxy on the central game-engine using a TCP socket. If the underlying GPRS layer is not connected, then the socket creation operation will fail immediately due to the PDA not having an IP address. While TCP is often suspected to operate poorly over wireless networks (Meyer, 1999), as described previously it is often the only protocol available for use over GPRS without resorting to more aggressive methods such as NAT punching (Ford et al., 2005). When the socket connect operation succeeds it is detected by both the mobile client and the remote mobile proxy.
- **Application:** The application layer dictates which messages are sent to and from the mobile client over the TCP socket, using a proprietary XML format. This layer also identifies the mobile client to the proxy using its unique ID (described in chapter 6), and the state of the mobile player's experience, as described earlier.
- **Interface:** The interface layer alerts the player as to the status of the connection, and modifies other features of the interface accordingly, as described in the previous section.

Figure 7-7 shows how the four layers of the communications platform interact. Each layer can be connected, disconnected or attempting to connect. Each layer senses its own status and automatically attempts to repair itself if it detects a failure in the connection. Each layer also explicitly monitors the status of the

layer below it, and so the status of the connection as a whole is propagated towards the interface layer where it is revealed to the player.

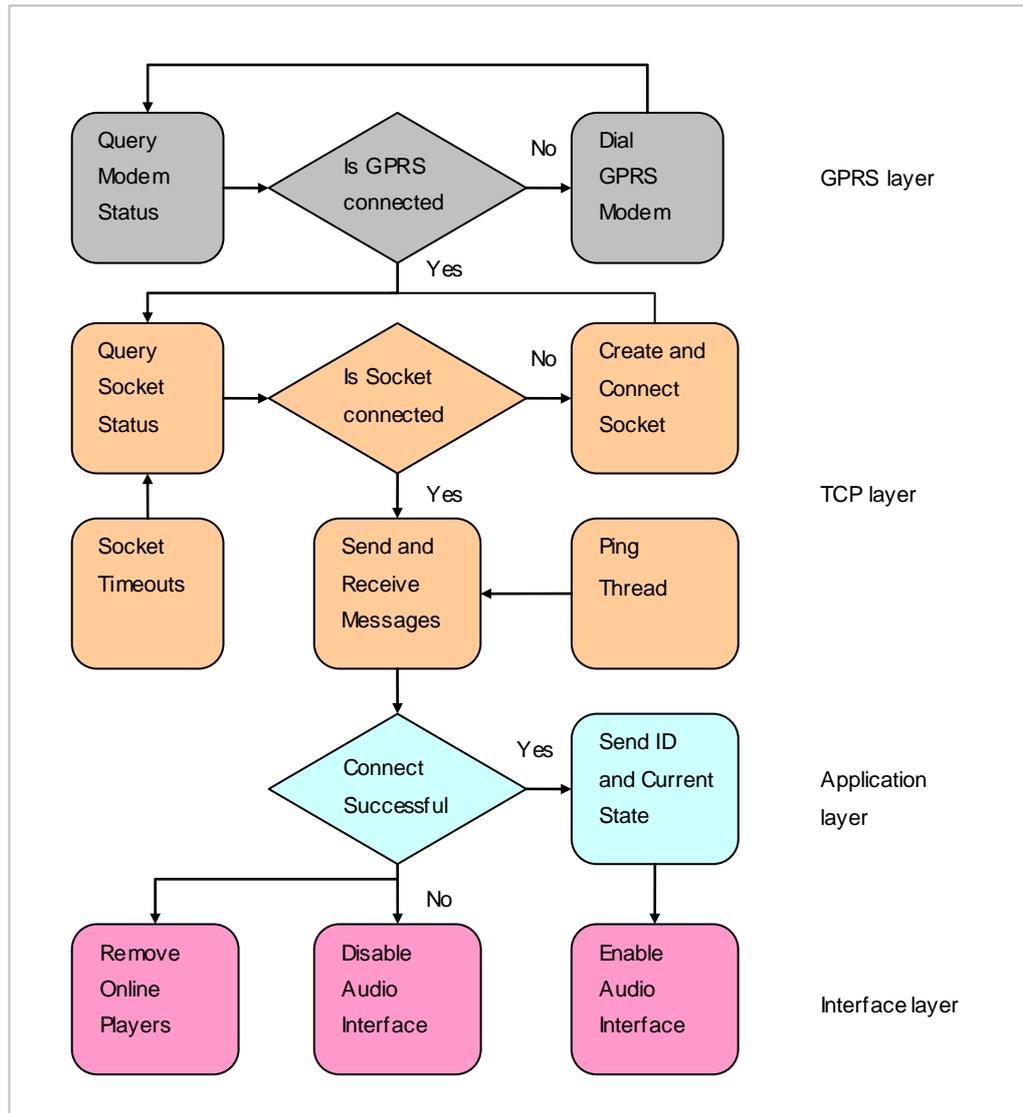


Figure 7-7: Four-Layer communications platform

The bottom layer, GPRS, constantly attempts to maintain a connection to the GPRS network by directly controlling the GPRS modem. If the modem is connected, then the layer identifies itself as connected and merely polls the modem at regular intervals to confirm this, whereas if the modem is disconnected or an operation has failed, then the layer resets and attempts to reconnect the modem. Similarly, the TCP layer constantly attempts to maintain a socket connection with the remote mobile proxy. As mentioned above, if the GPRS layer is not connected, then this action will fail immediately, whereas if the GPRS connection is weak and the socket takes a long time to connect, then

aggressive time-outs abandon the socket and retry, so as to immediately notify higher layers of the poor connection. Secondly, aggressive time-outs force the layer to assume the worst case scenario if no messages have been received recently and indicate quickly that the connection has been broken.

The remote mobile proxy works in conjunction with the application layer to maintain an accurate knowledge of the usability of the connection, by sending regular “ping” messages to each mobile client, to which the application layer responds in kind. These messages inform both parties that the connection is still alive and functional, and also keeps the TCP layer from timing out. Failure to receive and respond to the ping messages causes the mobile proxy to assume that the socket is experiencing packet loss and to close it to force a reconnection, and on the mobile client force the TCP layer to assume the same. Similarly, if a mobile client connects and identifies itself while the mobile proxy already has a connection open for the client, the original connection is destroyed as it can be assumed to have timed out.

The new software implementation supported further deployments of Uncle Roy All Around You and Can You See Me Now?. Response to the implementation was positive from both players and the production crew, in that responding to disconnections was no longer the critical focus of orchestrating the experiences, and performers could now spend time assisting mobile players with other problems, such as navigating content, or more severe software and hardware failures.

Analysis of system logs from both of these later outings of Can You See Me Now? and Uncle Roy All Around You showed that disconnection still occurred regularly, with a mobile player experiencing on average ten short disconnections over an hour long experience, each lasting a few seconds. It was also not uncommon that a player would experience a few much longer disconnections lasting for several minutes before the mobile client managed to successfully reconnect. This was especially true during Uncle Roy All Around You in West Bromwich, where an outage disabled one of the two mobile phone cells covering the game area, leaving only one cell for mobile clients to connect to. However, it is testament to the disconnected design strategy that

this did not appear to adversely affect the experience - one mobile player remained disconnected for 55 minutes, and yet remarked that they had a very enjoyable experience despite not being able to communicate with online players for most of the time.

7.4 Conclusions

This chapter has described the implementation of our four experiences, to show how the experiences have been constructed and how the authoring and orchestration tools described in the previous chapters are placed in the larger system in the light of the practical challenges of implementing an experience.

This chapter has described the implementation of the game-engine that forms the core of our four experiences, with the use of nonValidated and validated dataspace enabling it to support moderated position and message passing, audio communication and content distribution using ColourMaps. It has shown how the experiences support the use of industry-standard multi-media authoring packages for user-interfaces using dedicated proxies, which are also tailored to the various connection requirements of online and mobile players. It has shown how the architecture has been designed to support the integration of both control-room and front-of-house orchestration interfaces.

Finally, this chapter has discussed how wireless connectivity affects the design of the system, and how lack of connectivity can disrupt an experience for players while also complicating the orchestration task. It has shown how judicious placement of the ColourMaps system within the supporting architecture can provide a single-player, ColourMaps-based experience when disconnected, and how the connectivity subsystem can be made more robust in order to maximise the connected, multi-player experience.

8 Conclusion

8.1 Introduction

This final chapter presents a summary of the research undertaken over the course of this thesis. It reviews the significant contributions made to research into mobile mixed-reality experiences, reflects upon the work that has been accomplished and points to areas for future research and development, and concludes with a discussion of how this thesis relates to the wider community.

This chapter begins by restating the overall research goal of the thesis, and summarises the work presented chapter by chapter to show how these goals have been achieved.

The chapter then summarises the key research contributions and innovations of the work from three points of view; contributions made to the research community by the framework and tools presented in this thesis, the methodology used throughout this research in supporting our four mobile mixed-reality experiences, and its subsequent dissemination and impact.

Finally, this chapter reflects on some of the limitations and implications of the research, including possible directions for future work.

8.2 Summary

This thesis has investigated how to support mobile mixed-reality experiences in the wild, where members of the public take part in artistic performances, exploring a mixed-reality environment as mobile players using connected devices, or interacting and collaborating as virtual, online participants. These experiences spanned games, educational applications and performance-based new-media works, and have raised new challenges and opportunities for tools to support their creation and operation.

The original goal of the research is restated below:

To develop a framework for supporting the design and public deployment of artist-led mobile mixed-reality

experiences, and to create tools, techniques and guidelines to support experience authors in instantiating, populating and running such experiences with regard to factors unique to mobile technologies.

This goal has been accomplished through several steps, traceable through the previous chapters, and enabled by the practical experience of creating four real-world experiences that have had many thousands of public participants. These steps can be broadly divided into three parts:

- Placing the research in context by examining related work and the four experiences that form its foundation.
- Formulating a framework that highlights the need for authoring and orchestration in artist-led mobile mixed-reality experiences, based on the iterative development of our four experiences.
- Exploring the implementation and use of dedicated tools to support the authoring and orchestration of mobile mixed-reality experiences, developed iteratively alongside our four experiences.

8.2.1 Foundation Experiences and Related Works

The first section of the thesis began by reviewing selected related research in order to give the reader an overview of the field. Chapter 2 presented a selection of mobile mixed-reality experiences and other applications that have been developed in this area. It showed that many of the experiences make use of location-based content that is authored to match the physical environment, with some providing dedicated tools for authoring, or mechanisms for rapidly configuring an experience for each new location. Several of the authors of these experiences make note of the fact that unexpected occurrences regarding technology could be highly detrimental to a player's experience, and yet do not describe how this could be rectified within the context of the experience. In a similar vein, the majority do not discuss how to actively manage their participants, raising the question of how these experiences can move into a more public setting. The chapter concluded by summarising the key

requirements and functions of the experiences, highlighting the fact that, although they span a variety of genres and applications, many experiences share common features:

- Experiences involve multiple mobile, position-tracked players.
- Experiences involve simultaneous inhabitation of physical and virtual environments, presented to mobile players via a two-dimensional map.
- Players receive location-based content as they move.
- Experiences are generally short and event-based, and are managed directly by their authors.

Chapter 2 also showed that these features are not fundamental requirements of all mobile mixed-reality experience – with some involving non-spatial context rather than position-tracking, others based on history or trajectory, and some that last for longer periods of time and are largely uncontrolled by their authors.

Chapter 3 continued by describing the four mobile mixed-reality experiences in which the author was directly involved in the conception and development of – Can You See Me Now?, Uncle Roy All Around You, I Like Frank as part of an extensive collaboration with the artists group Blast Theory, and Savannah. The chapter described the key experiential components and the history of each experience, showing how these experiences formed an important set of staging posts for this research – firstly by giving a first-hand insight into challenges as they arose, secondly by providing an environment in which to rapidly prototype and iteratively refine solutions to these challenges, and thirdly by providing a demanding real-world scenario that allowed new ideas to be tested in experiences that involved many thousands of public participants in the wild. As a result of these four experiences, the research presented in this thesis could be identified, implemented and refined *in situ*, responding directly and immediately to the needs of the experience, the participants and the collaborating artists. The chapter concluded by summarising the requirements

of the experiences, showing how they share the common features presented in chapter 2.

8.2.2 Supporting Framework

Chapter 2 showed that many mobile mixed-reality experiences share similar functional requirements, and chapter 3 described a series of mobile mixed-reality experiences that provided a unique platform for investigating how to support these requirements. In response to this, and derived from the practical experience of developing the experiences described in chapter 3, chapter 4 presented a framework for supporting mobile mixed-reality experiences that considers how dedicated support for authoring and orchestration tools can play a useful role.

By classifying experiences by genre, it has argued that mobile mixed-reality experience can be located along a dimension of varying ‘forms of participation’, which ranges from ‘played’ – including experiences that are highly driven by game-play, to ‘performed’ – including experiences that revolve around theatrical performance. This has been used to demonstrate how artistic performances in particular are influenced by computer games; being highly driven by game-play, role-playing games; involving players taking particular roles, street-theatre; involving performers on the streets, and conventional theatre; including highly theatrical, pre-scripted performances.

Next, chapter 4 extended this framework to include an orthogonal, second dimension of the supporting tasks inherent in creating a mobile mixed-reality experience. This dimension ranges from tasks that involve ‘authoring prior to an experience’ – authoring content and framing the various performance elements of an experience, to those ‘improvised during an experience’ – using improvisation as a means to react to unfolding events. Consequently, the framework identified the following four tasks that required particular support during our four experiences: authoring play; creating and delivering location-based content and associated narrative, defining game-mechanics and interactions between players, preparing performance; scripting performances and constructing props and sets, and defining schedules and induction processes, improvising performance; performing around failures and

orchestrating the experience on the streets, and improvising play; rapidly prototyping new game-mechanics, testing new scenarios and responding to particular player actions.

The framework highlights the four challenges for supporting mobile mixed-reality experiences with authoring and orchestration, from the development of the four experiences:

- **Authoring location-based content and configuration**
- **Scripting performance and managing induction**
- **Supporting monitoring and intervention on the streets**
- **Supporting role-play and improvised game mechanics**

Chapter 4 concluded by arguing that these four challenges should be addressed through the creation of dedicated tools to support authoring and orchestration for use both before and during an experience.

8.2.3 Supporting Tools and Implementation

Chapters 5 and 6 examined the challenges of authoring and orchestration raised by the framework in detail, drawing on the development of the four experiences described in chapter 3 to show the supporting solutions that were put in place and to assess their impact on the experiences.

Chapter 5 focused on the challenge of supporting experience authors in authoring location-based content and serving this content to players during an experience, by charting the development of a tool that supported the core content requirements of Uncle Roy All Around You, I Like Frank and Savannah, and the configuration of Can You See Me Now?. The ColourMaps tool allows authors to create location-based content by painting colour regions onto a map of the physical environment, and this image is then used to serve content to players based on the colour of the pixel they occupy on the map.

The ColourMaps tool has been shown to provide flexibility for multiple uses within an experience; beginning with the physical map, authors mark key

starting positions for online and mobile players, coloured filtering regions to demarcate areas that are inaccessible to players for filtering incoming GPS data, and annotate several ideal routes to steer players through the environment. Layers of coloured content regions are added to provide multiple levels of content that is triggered by players as they move through the environment. By configuring edge, latching and digging behaviours to modify how a player interacts with these content regions, authors can construct more sophisticated content landscapes.

The ColourMaps tool proved to be invaluable for supporting the authoring requirements of our four experiences, with authors using it to construct the complex narrative clue-trails of *Uncle Roy All Around You* and *I Like Frank*, the sight, sound and smell environments of Savannah, and configuring start positions for online players and GPS filtering for mobile players in *Can You See Me Now?*. These elements formed the very core of each experience, and as such required authoring support that was sufficiently simple and familiar, and yet flexible and powerful enough for authors to easily be able to realise and refine their designs. The ColourMaps tool evolved to support these requirements by allowing the use of familiar image editing packages for creating ColourMaps, encompassing an intuitive and easily readable graphical format, and rapid configuration and loading.

Chapter 5 highlights the importance of the authoring challenges of the framework in supporting mobile mixed-reality experiences and the ColourMaps tool as a solution, demonstrating not only the need for dedicated tools to support the authoring task, but also tools that are designed with the particular needs of the authors in mind.

Chapter 6 focused on the challenges of orchestrating mobile mixed-reality experiences highlighted by the framework, and described the development of a framework of orchestration tools that were used to support the orchestration of our four experiences by enabling monitoring, intervention, improvisation and performance by behind-the-scenes staff. This framework aimed to support the day-to-day operation of each experience as a whole, combined with the micro-management of individual players both as part of normal operation and in the

light of unforeseen occurrences such as unexpected interpretation of content or technological failures.

The orchestration framework provides a control-room interface for monitoring an experience. The interface reveals the current state of all players, both online and mobile, allowing the operator to quickly view the state of the experience and then to drill down to examine a particular player if necessary. Importantly, the monitoring interface also reveals historical information for each player, including their previous movements, content that has been triggered and connectivity history, helping the operator to better understand the context of a given situation. Similarly, the interface reveals the content landscape that players inhabit, revealing where players should be, where they are likely to be going and areas that they potentially should explore next.

The orchestration framework provides support for intervention, improvisation and performance, which as the framework described in chapter 4 suggested, formed a significant part of the task of orchestrating our experiences. The orchestration framework allows the control room operator to override or modify the automated content of an experience, or to insert additional content on-the-fly. In conjunction with the monitoring interface, this supports improvisation by allowing the operator to respond to the actions of a player as they occur, or to reduce disruption after a technological failure by giving the player tailored content to get them back on track. The orchestration framework introduces the concept of orchestrating mobile players in situ, with a team of street performers identifying and monitoring players in the physical environment to provide a contrasting view to the control room, and to perform face-to-face interventions if necessary.

Finally, the orchestration framework views mobile mixed-reality experiences as performances, and as such provides dedicated support for inducting online and mobile players into an experience. A queuing system moderates and limits access for online players, and mobile players are inducted in a process that both introduces the experience and technology to each player and ensures that the experience and performers are ready to receive the player, while capturing additional data for orchestration further down the line.

The orchestration framework provided critical support for the operation of our four experiences, with behind-the-scenes staff relying on it to ensure each player taking part had the a good experience. Its support for improvisation allowed Savannah to explore new experience scenarios through rapid prototyping. Its support for monitoring, and control room and face-to-face interventions allowed players in Uncle Roy All Around You and I Like Frank to explore the city on their own terms while maintaining a strict performance schedule, responding to the needs of players as they arose while reducing disruption or loss of engagement.

Chapter 6 has demonstrated the importance of support for orchestration as raised by the framework described in chapter 4, and shown how the orchestration framework presents an integrated solution. It has shown that support for the orchestration of mobile mixed-reality experiences in the wild, as highlighted by our four experiences, is a core task that is essential for the successful deployment of an experience, rather than as a subsidiary task or an afterthought.

Finally, chapter 7 has described the implementation of our four experiences, showing how the experiences have been supported technologically, and placing the authoring and orchestration tools described in chapters 5 and 6 in context. It demonstrated how our four experiences made use of a common, component-based game-engine architecture to support position, communication, content distribution and remote orchestration, which also supported the use of industry-standard multi-media authoring packages for user interfaces, and catered to the different connection requirements of online and mobile players using dedicated proxies.

Chapter 7 discussed how the mobile players of early iterations of the experiences were particularly susceptible to disconnection from the wireless network, causing severe disruptions and complicating the orchestration task. It continued by describing how later versions of the experiences responded to this by explicitly supporting disconnected operation as an expected state of a mobile player's experience, placing content locally to support a single-player

experience when disconnected, and strengthening wireless connections when available to maximise connected, multi-player operation.

8.3 Research Contributions

This thesis shows how mobile mixed-reality experiences require dedicated support for authoring and orchestration, both of which present a number of identifiable requirements and challenges. This shifts the focus of the research from presenting mobile mixed-reality experiences as novel applications in and of themselves, to understanding these challenges in detail when supporting experiences in the wild. This has been shown to lead to a number of practical solutions, tools and guidelines that support the deployment of new experiences by deliberately supporting authoring and orchestration tasks. This thesis argues that only by embracing these issues can mobile mixed-reality experiences reach their full potential when deployed in the wild.

Authoring: A major contribution of this thesis is the investigation into how to support the challenge of authoring content for a sophisticated mobile mixed-reality experience. A new tool, the *ColourMaps system*, allows experience authors to create the location-based content that participants experience as they explore a mixed-reality environment, and configure how it is served to them along with other aspects of an experience that are related to the physical terrain. It shows how collaborating authors can maximise the potential of an experience by painting content directly onto a map of the physical environment, exploiting their knowledge of familiar tools to cut out the technical team and introducing support for rapid configuration and loading to increase productivity, alongside flexibility for multiple uses and types of content.

Orchestration: Another major contribution is the investigation into how to support the challenge of orchestrating a mobile mixed-reality experience that is presented in the wild to large numbers of public players. The *Orchestration framework* combines several novel techniques for managing players in a variety of situations that arise during an experience. Players are routinely and sustainably brought into their experience using dedicated queuing and induction tools. Control room operators make use of a management interface

that reveals what players are doing, what they have done through spatial trajectory, connectivity and message history views, and what they are likely to do by revealing the content landscape and authoring annotations. Operators and performers are given great agency to manipulate the experience of each player in a minimally disruptive manner, through intervention, improvisation and collaborative, distributed orchestration processes. This provides the operators of a mobile mixed-reality experience with the means to deal with a wide range of potential issues should they, and as they regularly do, occur.

8.3.1 Supporting the third role

This thesis has focused upon how to explicitly support a third role in the construction of mobile mixed-reality experiences – the role of the author alongside mobile and online players. The fundamental nature of the four experiences that have formed the foundation of this thesis, despite their iterative development, has been dictated primarily by the creative vision of the artists, and consequently this has driven the development of the supporting tools to a large extent. The artists retained strict creative control over each experience, from the core mechanics of the experience – for example receiving location-based content rather than a different content-triggering mechanism – to the fine details – the precise wording of the script that was performed to each player during the induction process in *Uncle Roy All Around You*. This creative vision has both focused and, to some extent limited, the scope of this research and the choices made in a number of respects.

The iterative development of the tools towards supporting specific types of mobile mixed-reality experiences has similarly been driven by the creative needs of the artists. The ColourMaps system focuses on authoring location-based content due to each experience particularly revolving around location-based content triggering. A limitation of the research is that it has not explored the potential of using the system to support other, not necessarily spatial, content authoring paradigms – for example supporting an experience that involves temporally, historically or contextually driven content triggering. Arguably the ColourMaps system presented in this thesis did not explore these alternatives in detail as the artists were not creating an experience that required

them, not that location-based content is by definition the most appropriate choice for a mobile mixed-reality experience.

Similarly, the creative vision of the artists in creating a mobile mixed-reality experience that, in their view, was primarily to be seen as much as a theatrical performance as a game or any other experience genre, led to the orchestration framework being developed to focus on ensuring that each player had a good experience *theatrically*, and one that was as close to the artists' vision as possible. As such, and similarly to the ColourMaps system, the orchestration framework focuses on supporting the micro-management of an experience by the artists, as this was their desired objective. This research consequently did not explore how the orchestration framework could be appropriated to support mobile mixed-reality experiences with different, less strict, and non-theatrical requirements, particularly those that are unmanaged, socially embedded, and last for longer periods of time.

The role of the artists also significantly influenced how the two tools described in this thesis logically relate to one another. As has been described in chapters 5 and 6, elements of the content created using the ColourMaps system are subsequently reused in the management interface of the orchestration framework, particularly in the use of annotated maps revealing the content landscape. However, the orchestration framework does not specifically complete the iterative loop by feeding information about experiences that have taken place, and the actions that needed to be taken to maintain them within the orchestration framework, back into the ColourMaps system, enabling content to be refined in order improve the next iteration of an experience. Arguably, each experience should take into account where players went, which content they explored too rapidly, and which content delayed them because it was too ambiguous, and the experience should be updated accordingly. However, once fundamental errors in the content had been resolved, the artists did not wish to further iteratively develop the content in this manner, and this is perhaps indicative of the creative control over the experience that they retained – once the experience and associated content had been developed to sufficiently match their particular creative vision, reacting to subsequent players' actions became less important, or may even have been contradictory, to this vision.

This focus on dedicated support for the work of the author and the orchestrator of an experience raises the question of how it might be applied to experiences that specifically do not involve such roles. While certain aspects of the ColourMaps system and of the orchestration framework could be redeployed in a more autonomous system – automatically generating ColourMaps from measured data, or queuing and marshalling players – a clear limitation of this research is its reliance on the artist.

8.3.2 Methodology

One of the most significant contributions of this research has been its use in supporting four publicly deployed mobile mixed-reality experiences; *Can You See Me Now?* - which continues to tour several cities a year to date, *Uncle Roy All Around You* and *I Like Frank*, and the educational experience *Savannah*. Critically, this research has emerged from the process of developing and supporting these experiences “in the wild” - collaborating with real-world artists and domain experts, and presenting the experiences to large numbers of members of the public as complete, professional performances.

This methodology of directly supporting the public performance of mobile mixed-reality experiences has provided a driving force for the development of this research, by presenting realistic challenges for which solutions had to be found in the form of supporting authoring and orchestration tools and systems. It presented an arena in which to demonstrate, refine and prove the validity of the tools and framework in response to direct feedback from the collaborating artists while authoring, and from the wide variety of orchestration tasks that arise when deploying such experiences in such a real world setting. Importantly, it has also provided the opportunity to evaluate the tools and concepts presented in this thesis, both as part of the practical iterative development cycle that working in the wild entails and that is documented here, and as part of a larger, collaborative, ethnographic study of the experiences. It has demonstrated the successful realisation and integration of the tools as part of the technical implementation of a component-based game-engine architecture that supports the four experiences. It has shown that while these tools are not an end in themselves, they prove provide crucial support to

artists and other domain experts in exploring these new avenues of mobile and related technologies.

This methodology allowed the work presented in this thesis to be particularly successful. The Equator Citywide Performance and Savannah projects, and most notably the collaboration with Blast Theory, provided the opportunity for creating four experiences that allowed this research to flourish, and provided a rigorous, demanding and often stressful environment for testing and iterative development. Conversely, the experiences would arguably not have been as successful as they have been without this research, and this is born out by the global recognition that they have received.

However, this methodology of rapidly prototyping tools and systems to directly support the four experiences is, sometimes, a double-edged sword, as was the fact that *Can You See Me Now?*, *Uncle Roy All Around You* and *I Like Frank* were publicly accessible performances, with all of the theatrical deadlines, additional requirements and public pressures that this entails. Given the relatively short production time allowed for the experiences, both the artistic design and supporting tools and technologies evolved as part of a tightly collaborative process between the author and Blast Theory. This was often akin to a shifting set of goal posts, with it sometimes being difficult to rationalise the needs of the artists – whose main motivation was to create an artistically valid experience, with the research requirements of the parent research project and this thesis. However, in hindsight it is clear that these tensions and associated deadlines were a catalyst to focus only on what was strictly necessary to accommodate both, and to resist the urge to pursue other superfluous goals. Much of this work has been inspired by collaborating closely with Blast Theory, understanding their needs and sharing their frustrations in both understanding the intricacies and nuances of mobile technologies while simultaneously creating a ground-breaking performance with it.

At times the demands of creating and refining the experiences and supporting tools, even while they were running and being used, can be frustrating due to the pressure of time and the knowledge that each player's experience was

something not to be thrown away lightly. However, while it may seem like it would have been ideal to have had a longer period of time to make sure everything was “just right”, arguably this is greatly superseded by the unique opportunity to practically support experiences in such a public and demanding arena. It is believed that this is what makes this research a significant contribution to the field; crucially supporting some of the most pioneering and publicly visible deployments of mobile mixed-reality experiences to date, while also refining the approach of working in the wild by demonstrating how to focus on the very pragmatic needs of collaborating artists and participating players alike.

8.3.3 Dissemination and Impact

This research has contributed to the general philosophy of supporting mobile mixed-reality experiences in a number of ways. Primarily, and as stated previously, it has directly and critically supported the creation of four experiences that have both successfully demonstrated the use of the framework and tools presented here, and that have also been critically acclaimed as novel and ground-breaking experiences by computer science and arts communities alike.

The author has been actively involved in the subsequent dissemination of this, and related research, through the following publications.

Details of the ColourMaps system have been published as:

- Flintham, M. (2005) **Painting the town red: configuring location-based games by colouring maps**. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Valencia, Spain, June 2005.
- Benford, S., D. Rowland, M. Flintham, R. Hull, J. Reid, J. Morrison, K. Facer and B. Clayton (2004) **Savannah: Designing a location-based game simulating lion behaviour**. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Singapore, June 2004.

Selected details of the Orchestration framework have been published in:

- Benford, S., A. Crabtree, S. Reeves, J. Sheridan, A. Dix, M. Flintham and A. Drozd (2006) **Designing for the opportunities and risks of staging digital experiences in**

public settings. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, Montreal, Quebec, April 2006.

The author has been directly involved in the evaluation and ethnographic study of this research and of the four experiences, details of which have been published as:

- Crabtree, A., S. Benford, T. Rodden, C. Greenhalgh, M. Flintham, R. Anastasi, A. Drozd, M. Adams, J. Row-Farr and N. Tandavanitj (2004) **Orchestrating a mixed reality game 'on the ground'**. In *Proceedings of the 2004 SIGCHI conference on Human factors in computing systems*, Vienna, Austria, April 2004.
- Benford, S., A. Crabtree, M. Flintham, D. Rowland, B. Gaver, M. Adams, J. Row-Farr, N. Tandavanitj, A. Oldroyd and J. Sutton (2004) **Provoking Reflection Through Artistic Games**. In *Proceedings of the 2004 SIGCHI Conference on Human Factors in Computing Systems (Workshop 19. Social Learning through Gaming)*, Vienna, Austria, April 2004.
- Benford, S., W. Seager, M. Flintham, R. Anastasi, D. Rowland, J. Humble, D. Stanton, J. Bowers, N. Tandavanitj and M. Adams (2004) **The Error of Our Ways: The Experience of Self-Reported Position in a Location-Based Game**. In *Proceedings of UbiComp 2004: Ubiquitous Computing: 6th International Conference*, Nottingham, UK, September 2004.
- Benford, S., D. Rowland, M. Flintham, A. Drozd, R. Hull, J. Reid, J. Morrison and K. Facer (2005) **Life on the edge: supporting collaboration in location-based experiences**. In *Proceedings of the 2004 SIGCHI Conference on Human Factors in Computing Systems*, Portland, Oregon, April 2005.
- Benford, S., A. Crabtree, S. Reeves, J. Sheridan, A. Dix, M. Flintham and A. Drozd (2006) **Designing for the opportunities and risks of staging digital experiences in public settings**. In *Proceedings of the 2006 SIGCHI conference on Human Factors in computing systems*, Montreal, Quebec, April 2006.

Details of the four experiences that this research has supported, with selected elements of this thesis, have been widely published by the author and regularly cited by other researchers, most notably in the following publications:

- Flintham, M., S. Benford, R. Anastasi, T. Hemmings, A. Crabtree, C. Greenhalgh, N. Tandavanitj, M. Adams and J. Row-Farr (2003) **Where on-line meets on the streets: experiences with mobile mixed reality games**. In *Proceedings of the 2006 ACM*

SIGCHI Conference on Human Factors in Computing Systems, Fort Lauderdale, Florida, April 2003.

- Flintham, M., R. Anastasi, S. Benford, A. Drozd, J. Mathrick, D. Rowland, N. Tandavanitj, M. Adams, J. Row-Farr and A. Oldroyd (2003) **Uncle Roy All Around You: Mixing Games and Theatre on the City Streets**. In *Proceedings of Level Up: The First International Conference of the Digital Games Research Association (DIGRA)*, Utrecht, Netherlands, November 2003.
- Benford, S., R. Anastasi, M. Flintham, C. Greenhalgh, N. Tandavanitj, M. Adams and J. Row-Farr (2003) **Coping with uncertainty in a location-based game**. In *Pervasive Computing*, IEEE **2**(3): pp 34-41.
- Benford, S., M. Flintham, A. Drozd, R. Anastasi, D. Rowland, N. Tandavanitj, M. Adams, J. Row-Farr, A. Oldroyd and J. Sutton (2004) **Uncle Roy All Around You: Implicating the City in a Location-Based Performance**. *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Singapore, June 2004.
- Benford, S., A. Crabtree, M. Flintham, A. Drozd, R. Anastasi, M. Paxton, N. Tandavanitj, M. Adams and J. Row-Farr (2006) **Can you see me now?** In *ACM Transactions on Computer-Human Interaction (TOCHI)* **13**(1): pp 100-133.
- Benford, S., A. Crabtree, S. Reeves, M. Flintham, A. Drozd, J. Sheridan and A. Dix (2006) **The Frame of the Game: Blurring the Boundary between Fiction and Reality in Mobile Experiences**. In *Proceedings of the 2006 ACM SIGCHI Conference on Human Factors in Computing Systems*, Montreal, Quebec, April 2006.

This research has directly contributed to the Equator IRC (Equator IRC, 2007), and aspects of the work have been actively used and extended within IPerG (IPerG, 2007)- the Integrated Project on Pervasive Gaming, particularly the use of the ColourMaps system in supporting Day of the Figurines, a new, long running experience using SMS (Flintham et al., 2007).

The author has presented elements of this research at a number of invited addresses to other communities, including the May You Live In Interesting Times Festival of Creative Technology (Flintham, 2005a), and the Screenplay Festival, (Flintham, 2005b).

The author has been involved in the construction of an interactive digital replay-based installation of Can You See Me Now?, which has been shown at

the Ars Electronica, Linz, Austria, and the Edith Russ Site for Media Art, Oldenburg, Germany, in 2003.

This research has been used as the basis for two artists' master-classes in collaboration with Blast Theory, at the Amsterdam-Maastricht Summer University, Amsterdam, 2001, and at the Technology School of the Future, Adelaide, 2004.

Can You See Me Now?, Uncle Roy All Around You and I Like Frank have received favourable critical reviews from the mainstream press, notably in the Metro (Murphy, 2003), and the Guardian (Wilson, 2003). As described in chapter 2, Can You See Me Now was nominated for an Interactive Arts BAFTA award in 2002, and was winner of the Golden Nica for Interactive Art at the Prix Ars Electronic, Austria in 2003. Uncle Roy All Around You was nominated for two BAFTA awards in the categories of Interactive Arts, and Technical and Social Innovation in 2004 and was nominated for a Net Art Award at the Webby Awards, also in 2004.

Other academics have also begun to write favourably about the four experiences, as revealed in the following excerpt from (Dixon, 2007), on participating in Uncle Roy All Around You as a mobile player:

Go home, re-evaluate yourself, the one you once loved and lost, the nature of memory, time's winged chariot, cities and surveillance, "virtual" realities, the fallibility of computers, the boundaries of bodies and space, the nature of life and its relationship to performance, the meaning of art. Cry like a baby. Realise, again, just how new and unprecedented such work is, and how timeless and humbling is the experience of great art.

Finally, particularly due to the touring longevity of Can You See Me Now?, it is estimated that this research has supported upwards of 30,000 members of the public in participating in these experiences.

8.4 Reflections and Future Work

This final section presents a number of reflections on both the limitations and implications of this research, and potential avenues for future work.

The research presented in this thesis has largely focused on supporting the mobile player in a mobile mixed-reality experience, with the role of the online player often somewhat relegated to a secondary concern. While *Can You See Me Now?* was solely targeted at online players and supported by mobile performers, in *Uncle Roy All Around You* and *I Like Frank* the research focused on supporting the authoring of location-based content primarily for mobile players, and primarily supporting the orchestration of mobile players as they explored the physical environment. This was perhaps practically justified, and to some extent politically motivated. Firstly, supporting online play is a well established and understood field, and the nature of mobile play arguably *requires* a greater, or at least different, level of support. Secondly, online players can take part in the experiences at no cost to themselves and, unlike mobile players, they are not required to make the same level of commitment to an experience - purchasing a ticket and turning up at the venue on time, and it is perhaps then not surprising that the authoring and orchestration effort was dominated by the need to make sure that the mobile side of the experiences was safe, smoothly run and enjoyable. It would have been useful to investigate how these tools could have played a more substantial role in the online experience.

Similarly, it is apparent that some of the everyday occurrences for mobile players, such as network disconnection, software or hardware failure or simply just getting lost, were detrimental to the online experience, with online players often failing to understand what was actually happening on the streets. It would have been useful to have investigated how this can be effectively revealed to online players, potentially through orchestration, so that they could adopt their own play accordingly.

It is believed that the ColourMaps system has not yet been explored to its full potential. While it has been successfully used to provide the core content functionality for all of the experiences, the more advanced characteristics of digging, levels, their light-weight nature and intuitive authoring principle have

not been fully exploited by giving the system to a wider variety of authors. In the interests of further supporting the rapid prototyping of both mobile and mobile mixed-reality experiences, it would be interesting to provide and distribute a simple set-up on a single device that allowed authors to construct clue-driven location-based experiences using ColourMaps and self-reported positioning. This would provide a standalone platform for use by artists groups, schools, or any other groups that were interested in creating experiences in a variety of areas, in a similar vein to Create-a-Scape (Futurelab, 2007).

As a highly visual medium, ColourMaps can provide a dynamic representation of other aspects of the mobile infrastructure, for example GPS reception or connectivity strength. This could be used to allow authors to create content in relation to both the physical environment and this mobile infrastructure that took into account whether a player was likely to be connected or have access to reliable position reports, or to even trigger content directly based on the measured infrastructure as it evolves. This work is currently being taken forward, by examining how authoring tools can benefit from an infrastructure visualisation layer (Oppermann et al., 2006). As described in the previous section, ColourMaps are not restricted to authoring spatially distributed content, and could be used to author content to be triggered by change in any two discrete inputs – for example using time to place a player on the ColourMap, or data collected from a sensor.

A particular challenge for orchestration was to respond to the needs of players with minimal disruption to their experiences and subsequent loss of engagement. To this end, approaching a mobile player on the streets was considered a last resort for orchestration, and also the greatest strain on the resources of supporting performers in recognising and locating errant players. To minimise the time taken during this intervention a strategy for getting a player's device back on track, even if they were merely disconnected, was to “turn it off and on again”. A potential avenue for future research is how to better support distributed orchestration on the ground, with the development of more sophisticated mobile interfaces that allow street performers to perform the same orchestration functionality as provided by the control room,

potentially remotely orchestrating globally disconnected players locally through short-lived peer-to-peer networks.

The mobile mixed-reality experiences described, compared to large-scale commercial deployments of location-based services and games, have been relatively small, short-lived performances, despite their touring longevity and the involvement of a not insignificant number of members of the public. The experiences were the result of intensive periods of rapid prototyping and development, to meet the inherent and concrete deadlines of presenting the experiences to the public at the allotted time, compared with the more structured, longer-term roll-outs that one might expect from a more 'polished' product. In some senses the experiences are sophisticated prototypes, and while they have provided a platform to consider the research presented in this thesis and other areas, it is an open question as to how this research might evolve to support a much larger production. Similarly, it would be interesting to explore how this research could be applied to experiences in which an artist did not play such a fundamental creative role as the author and orchestrator, and that were less theatrical or performance-based in nature.

References

3GPP (1998) **AMR Specification**, Available from <http://www.3gpp.org/specs/specs.htm>

42 Entertainment (2004) **I Love Bees**, <http://www.ilovebees.com>

@Last (2000) **SketchUp**, <http://www.sketchup.com>

Adobe Systems (2004a) **Director**, <http://www.adobe.com/products/director>

Adobe Systems (2004b) **Director Multi-User Server**,
<http://www.adobe.com/products/director/multiuser>

Adobe Systems (2007a) **Flash**, <http://www.adobe.com/products/flash>

Adobe Systems (2007b) **Photoshop**, <http://www.adobe.com/products/photoshop>

Adobe Systems (2007c) **Shockwave Player Adoption Statistics**,
http://www.adobe.com/products/player_census/shockwaveplayer

Autodesk (2006) **Maya**, <http://www.autodesk.com/maya>

Autodesk (2007) **3D Studio Max**, <http://www.autodesk.com/3dsmax>

Barkhuus, L., M. Chalmers, P. Tennent, M. Hall, M. Bell, S. Sherwood and B. Brown (2005) **Picking Pockets on the Lawn: The Development of Tactics and Strategies in a Mobile Game**. In *Proceedings of UbiComp 2005: Ubiquitous Computing: 7th International Conference*, Tokyo, Japan, September 2005, Springer London.

Bell, M., M. Chalmers, L. Barkhuus, M. Hall, S. Sherwood, P. Tennent, B. Brown, D. Rowland and S. Benford (2006) **Interweaving mobile games with everyday life**. In *Proceedings of the 2006 SIGCHI conference on Human Factors in computing systems*, Montreal, Quebec, Canada, ACM.

Benford, S., A. Crabtree, S. Reeves, J. Sheridan, A. Dix, M. Flintham and A. Drozd (2006) **Designing for the opportunities and risks of staging digital experiences in public settings**. In *Proceedings of the 2006 SIGCHI conference on Human Factors in computing systems*, Montreal, Quebec, April 2006.

Biswas, A., T. Donaldson, J. Singh, S. Diamond, D. Gauthier and M. Longford (2006a) **Assessment of mobile experience engine, the development toolkit for context aware mobile applications**. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, Hollywood, California, ACM.

- Biswas, A., T. Donaldson, J. Singh, S. Diamond, D. Gauthier and M. Longford. (2006b) **Trickster**, <http://trickster.banff.org>
- Björk, S., J. Falk, R. Hansson and P. Ljungstrand (2001) **Pirates!-Using the Physical World as a Game Board**. In *Proceedings of the 8th TC13 IFIP International Conference on Human-Computer Interaction*, Tokyo, Japan, July 2001.
- Blackwell, A., W. Mackay, L. E. Holmquist, S. Wright and N. Guernion (2007) **The EQUATOR Interdisciplinary Research Collaboration Final Review**, Mixed Reality Lab, University of Nottingham, May 2007.
- Blast Theory. (2007) **Blast Theory**, <http://www.blasttheory.co.uk>
- Blizzard Entertainment (2004) **World of Warcraft**, <http://www.worldofwarcraft.com>
- Broecke, J. (2006) Sense of the City, <http://www.senseofthecity.nl>
- Cheok, A. D., K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo, Y. Li and X. Yang (2004) **Human Pacman: a mobile, wide-area entertainment system based on physical, social and ubiquitous computing**, In *Personal and Ubiquitous Computing* **8**(2): pp 71-81.
- Cheverst, K., N. Davies, K. Mitchell and A. Friday (2000) **Experiences of developing and deploying a context-aware tourist guide: the GUIDE project**. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, Massachusetts, United States, ACM.
- Curtis, P. (1997) MUDDING: Social Phenomena in Text-Based Virtual Realities. **Culture of the Internet**. S. Kiesler. Mahwah, NJ, 1997, Lawrence Erlbaum Associates.
- de Souza e Silva, A. (2004) **Art by Telephone: From Static to Mobile Interfaces**, In *Leonardo Electronic Almanac*, **12**(10).
- Dixon, S. (2007) **Digital Performance: A History of New Media in Theater, Dance, Performance Art, and Installation**. Cambridge, Massachusetts, 2007, MIT Press.
- Donaldson, T. and D. Gauthier (2006) **Source code and MEE libraries for Trickster, Dear & Bear and Situated Editor**. Montreal, Feb 2006.
- Drozd, A., J. Bowers, S. Benford, C. Greenhalgh and M. Fraser (2001) **Collaboratively Improvising Magic: An Approach to Managing Participation in an Online Drama**. In *Proceedings of 7th Intl European Conference on Computer Supported Cooperative Work (ECSCW)*, Bonn, Germany, September 2001, Springer Netherlands.
- Equator IRC. (2007) **Equator IRC**, <http://www.equator.ac.uk>

- ESRI. (2000) **ArcGIS**, <http://www.esri.com>
- Fine, G. A. (2002) **Shared Fantasy: Role Playing Games as Social Worlds**, University of Chicago Press.
- Fitzpatrick, G., T. Mansfield, S. Kaplan, D. Arnold, T. Phelps and B. Segall (1999) **Augmenting the workaday world with Elvin**. In *Proceedings of the sixth conference on European Conference on Computer Supported Cooperative Work*, Copenhagen, Denmark, 1999, Kluwer Academic Publishers.
- Flintham, M. (2005a) **Designing for Uncertainty and Failure**. Presented at *May You Live In Interesting Times Festival of Creative Technology*, Cardiff, October 2005.
- Flintham, M. (2005b) **Mobile Mixed-Reality Experiences**. Presented at *Screenplay 2005*, Nottingham, 2005.
- Flintham, M., K. Smith, S. Benford, M. Capra, J. Green, C. Greenhalgh, M. Wright, M. Adams, N. Tandavanitj, J. Row-Farr and I. Lindt (2007) **Day of the Figurines: A Slow Narrative-Driven Game for Mobile Phones Using Text Messaging**, In *Proceedings of the 4th International Symposium on Pervasive Gaming Applications*, Salzburg, Austria, June 2007.
- Foo, C. Y. and E. Koivisto (2004) **Defining grief play in MMORPGs: player and developer perceptions**. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, Singapore, June 2004, ACM.
- Ford, B., P. Srisuresh and D. Kegel (2005) **Peer-to-peer communication across network address translators**. In *Proceedings of the 2005 USENIX Annual Technical Conference*, 2005.
- Futurelab (2007) **Create-a-Scape**, <http://www.createascape.org.uk>
- GeoBiking. (2005) **GeoBiking**, <http://www.geobiking.com>
- GeoCaching. (2000) **GeoCaching**, <http://www.geocaching.com>
- GeoSailing. (2005) **GeoSailing**, <http://www.geosailing.com>
- GeoSkating. (2005) **GeoSkating**, <http://www.geoskating.com>
- GeoTracing. (2005) **GeoTracing**, <http://www.geotracing.com>
- Google. (2007) **Google Maps**, <http://maps.google.com>
- Greenhalgh, C. (2002) **EQUIP: An extensible platform for distributed collaboration**. In *The Second Workshop on Advanced Collaborative Environments*, July 2002.

Greenhalgh, C., J. Bowers, G. Walker, J. Wyver, S. Benford and I. Taylor (1999) **Creating a live broadcast from a virtual environment**. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, Los Angeles, CA, USA, August 1999, ACM Press/Addison-Wesley Publishing Co.

Hull, R., B. Clayton and T. Melamecr (2004) **Rapid Authoring of Mediascapes**. In *Proceedings of UbiComp 2004: Ubiquitous Computing: 6th International Conference*, Nottingham, UK, September 2004.

id Software (1999) **Quake 3 Arena**, <http://www.idsoftware.com>

IPerG. (2007) **Integrated Project on Pervasive Gaming**, <http://www.pervasive-gaming.org>

Its Alive! (2000) **Botfighters**, <http://www.botfighters.com>

Jonsson, S., M. Montola, A. Waern and M. Ericsson (2006) **Prosopopeia: experiences from a pervasive Larp**. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, Hollywood, California, June 2006.

Jonsson, S., A. Waern, M. Montola and J. Stenros (2007) **Game Mastering a Pervasive Larp: Experiences from Momentum**. In *Proceedings of the 4th International Symposium on Pervasive Gaming Applications*, Salzburg, Austria, June 2007.

Koile, K., K. Tollmar, D. Demirdjian, H. Shrobe and T. Darrell (2003) **Activity Zones for Context-Aware Computing**. In *Proceedings of Ubicomp 2003: Ubiquitous Computing: 5th International Conference*, Seattle, Washington, October 2003.

Koleva, B., I. Taylor, S. Benford, M. Fraser, C. Greenhalgh, H. Schnadelbach, D. Lehn, C. Heath, J. Row-Farr and M. Adams (2001) **Orchestrating a mixed reality performance**. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, Seattle, Washington, United States, ACM.

Kureshy, A. (2004) **Architecting Disconnected Mobile Applications Using a Service Oriented Architecture**. *MSDN Library*, September 2004: pp.

Lane, G. (2003) **Urban Tapestries: Wireless networking, public authoring and social knowledge**, In *Personal and Ubiquitous Computing* 7(3-4): pp 169-175.

Lantz, F. (2003) **Conqwest**, <http://homepages.nyu.edu/~dc788/conqwest/>

Laurel, B. (1991) **Computers as Theatre**, Reading, MA: Addison-Wesley.

- Li, Y., J. Hong and J. Landay (2004) **Topiary: a tool for prototyping location-enhanced applications**. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, Santa Fe, NM, USA, ACM.
- Licoppe, C. and R. Guillot (2006) **ICTs and the Engineering of Encounters: A Case Study of the Development of a Mobile Game Based on the Geolocation of Terminals**. In *Mobile Technologies of the City*. M. Sheller and J. Urry. New York, NY, Routledge.
- Luff, P. and C. Heath (1998) **Mobility in collaboration**. In *Proceedings of the ACM conference on Computer supported cooperative work (CSCW)*, Seattle, Washington, 1998.
- Magerkurth, C., R. Stenzel and T. Prante (2003) **STARS - A Ubiquitous Computing Platform for Computer Augmented Tabletop Games**. In *Proceedings of the Fifth International Conference on Ubiquitous Computing (UBICOMP '03)*, Seattle, Washington, USA, October 2003.
- Mandryk, R. L. and K. M. Inkpen (2001) **Supporting free play in ubiquitous computer games**. In *Proceedings of Workshop on Designing Ubiquitous Computer Games, UbiComp*. 2001.
- Meyer, M. (1999) **TCP performance over GPRS**. In *Proceedings of IEEE Wireless Communications and Networking Conference*. 1999.
- Milgram, P. and F. Kishino (1994) **A Taxonomy of Mixed-Reality Visual Displays**. In *IEICE Transactions on Information Systems* **E77-D**(12).
- Mind Candy (2005) **Perplex City**, <http://www.perplexcity.com>
- Mitchell, K., D. McCaffery, G. Metaxas, J. Finney, S. Schmid and A. Scott (2003) **Six in the city: introducing Real Tournament - a mobile IPv6 based context-aware multiplayer game**. In *Proceedings of the 2nd workshop on Network and system support for games*, Redwood City, California, May 2003, ACM.
- Montola, M. (2005) **Exploring the Edge of the Magic Circle: Defining Pervasive Games**. In *Proceedings Of Digital Experience: Design, Aesthetics, Practice conference*, Copenhagen, 2005.
- Muramatsu, J. and M. S. Ackerman (1998) **Computing, Social Activity, and Entertainment: A Field Study of a Game MUD** In *Computer Supported Cooperative Work (CSCW)* **7**(1-2): pp 87-122.
- Murphy, S. (2003) **Theatre Review**. Metro. London, 30th May 2003.

- Nova, N., F. Girardin and P. Dillenbourg (2005) "**Location is not enough!**": an **Empirical Study of Location-Awareness in Mobile Collaboration**. In *IEEE International Workshop on Wireless and Mobile Technologies in Education*. Japan, November 2005.
- Ohshima, T., K. Satoh, H. Yamamoto and H. Tamura (1999) **RV-Border Guards: A Multiplayer Entertainment in Mixed Reality Space**. In *Proceedings of the 2nd Int'l Workshop on Augmented Reality*, San Francisco, October 1999.
- Oppermann, L., G. Broll and S. Benford (2006) **Exploiting Seams in Mobile Phone Games**. In *Proceedings of the 3rd International Workshop on Pervasive Gaming Applications (PerGames2006)*: pp 4-10.
- PacManhattan. (2004) **PacManhattan**, <http://www.pacmanhattan.com>
- Pascoe, J. (1997) **The stick-e note architecture: extending the interface beyond the user**. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, Orlando, Florida, United States, ACM.
- Piekarski, W. and B. Thomas (2002) **ARQuake: the outdoor augmented reality gaming system** In *Communications of the ACM 45(Special Issue: Game engines in scientific research)*: pp 36-38.
- Popper, F. (1993) **Art of the Electronic Age**. New York, Harry N. Abrams, Inc.
- PostgreSQL. (1996) **PostgreSQL**, <http://www.postgresql.org>
- Refractions Research. (2005) **PostGIS**, <http://postgis.refractions.net>
- Reid, J., R. Hull, K. Cater and B. Clayton (2004) **Riot! 1831: The design of a location based audio drama**. In *Proceedings of UK-UbiNet*, Cambridge, May 2004.
- Rhind, D., S. Openshaw and N. Green (1988) **The Analysis of Geographical Data: Data Rich, Technology Adequate, Theory Poor**. In *Proceedings of the 4th International Working Conference on Statistical and Scientific Database Management*, Springer Verlag.
- Salen, K. and E. Zimmerman (2003) **Rules of play: game design fundamentals**, MIT Press.
- Schilit, B. N., N. Adams and R. Want (1994) **Context-aware Computing Applications**, Xerox Corp., Palo Alto Research Center.
- Shore, T. (2003) **Navigate the Streets**, <http://www.navigatethestreets.com>
- Sohn, T. and A. Dey (2003) **iCAP: an informal tool for interactive prototyping of context-aware applications**. In *CHI '03 extended abstracts on Human factors in computing systems*, Ft. Lauderdale, Florida, USA, ACM.

- Stabell, B. and K. R. Schouten (1991) **XPilot**, <http://www.xpilot.org>
- Suchman, L. (1995) **Making work visible**, In *Communications of the ACM* **38**(9).
- Superior Software (1985) **Repton 3**, <http://www.superiorinteractive.com>
- Valve Corporation (2004) **Half-Life 2**, <http://www.valvesoftware.com>
- Vogiazou, Y., M. Eisenstadt, M. Dzubor and K. Komzak (2005) **From buddies pace to CitiTag: large-scale symbolic presence for community building and spontaneous play**. In *Proceedings of the 2005 ACM symposium on Applied computing*, Santa Fe, New Mexico, ACM.
- Weiser, M. (1999) **The computer for the 21st century** In *ACM SIGMOBILE Mobile Computing and Communications Review* **3**(3): pp 3-11.
- Wilson, H. (2003) **One of the team**. Office Hours, The Guardian, September 29th, 2003.
- Wink Back. (2000) **The Go Game**, <http://www.thegogame.com>
- Winter, S. (1998) **Bridging vector and raster representation in GIS**. In *Proceedings of the 6th ACM international symposium on Advances in geographic information systems*, Washington, D.C., United States.
- Woodside, S. (2003) **Semacode**, <http://semacode.org>