SoundPark: Exploring Ubiquitous Computing through a Mixed Reality Multi-player Game Experiment

Romain Pellerin * — Nicolas Bouillot ** — Tatiana Pietkiewicz * — Mike Wozniewski ** — Zack Settel * — Eric Gressier-Soudan * — Jeremy R. Cooperstock ****

* *Conservatoire National des Arts et Métiers, Paris, France*
** *McGill University, Montreal, QC, Canada*
*** *University of Montreal, Montreal, QC, Canada*

RÉSUMÉ. *Nous présentons une architecture ubiquitaire à travers le développement et la mise en œuvre de SoundPark, notre jeux audio en réalité augmentée. Les interactions ubiquitaires au sein de l'équipe sont encouragées par un objectif commun : collecter et assembler des sons virtuels disséminés dans un parc. Les technologies déployées permettent de supporter le jeu, ainsi que la navigation visuelle et sonore dans sa représentation virtuelle. Dans ce but nous avons intégré ensemble notre intergiciel pour les jeux mobiles et multi-joueurs, notre moteur de streaming audio basse latence, ainsi que notre moteur de rendu spatial audiovisuel.*

ABSTRACT. *We describe an ubiquitous computing architecture through a multi-player game application based on the objective of collecting audio clips and depositing them in a staging area. Central to the game are the themes of highly coupled interaction and communication between players with different roles and an engaging blend of interaction with both the physical and virtual worlds. To this end, numerous technologies including locative sensing, miniature computing, and portable displays had to be integrated with a game middleware and audio scene rendering engine. The result provides a compelling example of future distributed systems that this paper describes.*

MOTS-CLÉS : *Informatique ubiquitaire, réalité augmentée, mobilité, systèmes embarqués*
KEYWORDS: *Ubiquitous computing, augmented reality game, mobility, embedded systems*

1. Introduction

Ubiquitous computing architectures enable a wealth of applications that go far beyond the traditional realm of Distributed Systems. Taking advantage of the distinguishing features of ubiquitous computing architectures, namely, a distributed computational infrastructure, support for communication, location- and context-awareness, and critically, mobility of users, allows for the emergence of games that support rich interaction between multiple users and their environment. Traditionally, this has been done entirely in the space of a virtual world, with no obvious relationship to the user's surroundings. However, ubiquitous computing lends itself well to supporting mixed reality applications, thus leveraging the richness of the physical environment for more engaging interaction.

The research described here is intended to explore distributed system features for coupled interaction in the ubiquitous computing context, where real-time audio communications and frequent state updates are imperative to support the game activity. Our proof of concept application, *SoundPark*[1], is designed to let users navigate in a reasonably sized area, communicating with each other through high-fidelity, low-latency audio, and interacting with a rich, mixed-reality environment in which spatialized virtual audio sources are distributed throughout the physical space.

Section 2 describes different ubiquitous games and their main features. Section 3 is dedicated to SoundPark game design. Section 4 provides a description of the full game architecture while Section 5 deals with implementation issues. Section 6 includes a discussion of a fully distributed architecture and finally, Section 7 concludes the paper.

2. Related work

Technologies commonly found in large-scale pervasive computing games include mobile phones, PDAs, and portable game systems, in conjunction with a web server to publish information related to player

1. A sample video is available from `http://www.audioscape.org/twiki/bin/view/Audioscape/MobileAudioscape`.

missions or object collections. GPS technology is used extensively for geo-localization, in particular those based on mixed reality, either to provide players' locations or to collect virtual objects in the environment [NY07]. Examples include Botfighter [Sot02], CanYouSeeMe-Now [FBA+03], Mogi [Jof05], and CatchBob [GN06]. The remainder of this section surveys some of the relevant efforts involving locative technology, interaction with a virtual audio environment, and multi-player pervasive games involving networked communication between the participants.

Several applications of *locative media technology* focus on virtual audio overlays in large-scale spaces. In the simplest case, geographical locations of interest are tagged with sound. One of the earliest such efforts was the *[murmur]* project[2], in which the tags take the form of telephone numbers that can be dialed by other users on their mobile phones, in order to listen to audio recordings related to the locations. Another example, supporting continuous exploration of location-based audio, was the *Hear&There* project [RKD00], which allowed users to record audio at given GPS coordinates, while hearing a spatial rendering of other recordings as they navigate. This was limited to a single-person experience, with the scene state maintained by a single computer. These efforts established the basic concept, which has since been adopted by many other groups [Sch08, SHG+07], of selecting audio samples based on the user's physical location. The SoundPark project uses the same principles but address a multi-player experience.

In the area of multi-player mobile games, a seminal example is Cheok's *Human Pacman* [CGL+04], which inherits its design from the eponymous arcade game of the early 1980's, but has human players, wearing head-mounted displays (HMD), who take on the role of *pac-man* or *ghosts*. Although Human Pacman is an engaging, media-rich, multi-player game, the equipment users required was bulky and, in particular with regard to the HMD, awkward to wear on a long-term basis. Human Pacman has been extended by different projects, including Pac-Lan RFID [RBC+06] where both game objects (pills, super-pills, and ghosts) and human players have associated RFID tags. Such

2. `http://murmurtoronto.ca`

tags, increasingly popular in entertainment applications, are used for the purposes of location, identification and content delivery [Boh04, HGMJ07]. Our game is less equipment-intensive than Human Pacman, requiring only small form factor devices, such as gumstix and near-field communications (NFC)-enabled Nokia 5140 mobile phone, allowing for more user mobility in the gameplay. In addition to the common use of geo-localization, certain recently proposed games, such as MeetYourHeartTwin [FKS07], propose the additional integration of biosensors to establish social relationships between players. In this example, players are able to see others, on their PDA displays, sharing their heartbeat characteristics, e.g., in a similar frequency range.

SoundPark shares many characteristics of these earlier works. However, rather than a focus on 3D graphics (e.g., as in Human Pacman), our emphasis is on audio interaction, for communication between players, for rendering of game state, and for exploration of the virtual world. SoundPark also uses RFID tags, but only for content delivery of *clues*. This bias toward the auditory modality is motivated in part by observing that the physical world offers an environment that is enormously rich visually, and from which, we do not wish to isolate users. In this regard, we believe that an outdoor park can be augmented with digital artifacts such as sound loops and processing effects to offer a more compelling experience than could a strictly virtual environment.

3. Game design

The SoundPark game was designed for the Jeanne Mance park, an area covering 630x190 meters in the city of Montreal. The physical park environment was augmented with (physical) RFID tags and (virtual) audio objects, the latter which can be heard, acquired, and moved by players. A team is composed of three types of players, whose combined goal is to create a musical arrangement by discovering and collecting a number of compatible sound loops and assembling them in a staging area or home base, using walkie-talkie-like communication to coordinate their activities. The loops are placed initially at pre-determined locations, which are disclosed to one type of player, the *scout*, in the form of clues read from RFID-tagged bollards. These sound loops may only be discovered (via directed listening) and collected by another type of player, the *hunter*, when he is close to the corresponding location, once the associated clue has been discovered. Sound manipulation functions, required by the third player, the *D.J.*, are represented as virtual objects, discovered by the scout in a similar manner. The D.J. is responsible for organizing the collected sounds, using acquired capabilities of sound sequencing or mixing.

Consistent with the theme of a mixed reality environment, game objects typically exist as both real (physical) and virtual entities : Clues are represented by their associated bollards and RFID tags, but reveal their content graphically on the scout's mobile phone display. Although sound loops are only perceived in the virtual domain, i.e., heard through the players' headsets, they exist in association with specific physical locations in the real environment and can thus be moved from place to place. To increase the challenge of the game, clues indicate the location of individual sound loops, but not their musical properties, which can only be discovered by the hunters as they draw near. Furthermore, not all sound loops will fit with the theme of a target musical arrangement. For example, a baroque guitar loop cannot be part of a jazz arrangement. Attempts to deposit such a sound loop in the staging area fail with an appropriate feedback message. Inclusion of a modest number of these distractors throughout the environment adds a layer of complexity, fun, and opportunity for musical pedagogy to the game activity.

A team is composed of one scout, multiple hunters and one D.J. Activity is initiated by the scout, who discovers clues by reading the RFID tags distributed throughout the game area and observing the location information displayed on the mobile phone screen. At that point, the associated sound loop is unlocked, and can be heard by a hunter who passes within proximity of its physical location, typically guided by the scout using a wireless, low-latency audio communication protocol. The location of the hunter is updated in real time on the scout's mobile phone map display.

With guidance from the scout, the hunter must retrieve sound loops from their locations by pressing a button on a Wiimote, which communicates to the mobile computer worn by the player. Audio spatialization is used to guide the hunter toward the sound loop location, which becomes audible exclusively to the hunter and only in close proximity. Once acquired, the sound loop attaches to the hunter, and can be carried to the staging area, where it is deposited if part of the target musical arrangement. In this case, the other members of the team can hear the combined sound as an arrangement, which is ordered by the D.J. The game ends when the first team completes the assembly of their target musical arrangement or upon expiration of the game timer.

The D.J. orders the sound loops deposited in the staging area to create the musical arrangement, and plays a critical role in advising the hunter to ignore sound loops that are not part of the target score. To motivate communication between team members, the hunter must describe the sound loop, e.g., “I hear drums !” and the scout can request elaboration as necessary to determine whether the loop belongs in the arrangement. Since the distances between sound loops and the staging area may stretch across the full length of the park, the hunter may achieve significant time savings by discarding inappropriate sound loops right away rather than carrying them all the way to the base before discovering that they do not belong.

4. Architecture

The game design described above puts an emphasis on group interaction as a fundamental objective. In addition, it assumes the availability of a virtual environment that is continuously modeled based on sensor data, and displayed to users. At the application level, the success of the interaction depends considerably on the fidelity of users’ mutual awareness, which includes both player position and the high-fidelity audio transmitted between players themselves and from objects in the virtual environment. Support for the various interactions mandates the use of several technologies. These include DGPS locative sensing, RFID reading, audio input and output with spatialization capability, rendered individually for each player, real-time audio communications, and graphical map display. Integrating these components together is the responsibility of the middleware layer, which provides the game engine that maintains game state and logic in a highly mobile context.

The design of the computational infrastructure on which these components run requires integration of hetegoreneous hardware and software, and support for scalability, monitoring and mobility support. Although a fully distributed computing architecture would, in theory, be preferable, the demands of immediate practical use necessitated adoption of a centralized architecture for prototyping purposes. This further simplifies integration of the components, monitoring and debugging during experimentation and provides an initial step toward the next generation of interactive applications in the ubiquitous domain. Further challenges associated with a move to a distributed architecture are discussed in Section 6.

4.1. *Overview*

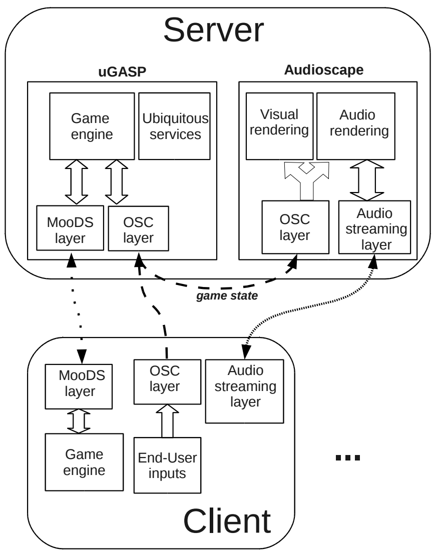

Figure 1 – SoundPark Middleware Architecture Overview.

Our architecture, shown in Figure 1, includes a server composed of two parts dedicated to computation of game state : the Audioscape server for real-time rendering of audio and 3D graphics and the uGASP server, dedicated to game management. Clients consist mainly of layers dedicated to I/O data transmission to the server, including audio and sensing. For instance, RFID reading capabilities are provided to the scout through an NFC-enabled Nokia 6131 mobile phone. Although RFID tags could contain game content on their own,[3] we simply use their Universal Identifiers (UID) to index into the game database, housed on a server.

Employing the use of an 802.11g wireless network, communication between software running on the servers and mobile devices uses three different protocols. The first one, the MooDS protocol [Pel07], is dedicated to object-oriented communication with mobile phones, suitable for game events dissemination. Then Open Sound Control[4] (OSC), a character-oriented data exchange protocol for real-time control of sound and other media processing that has gained considerable popularity in the distributed audio community. Finally, the high-fidelity audio streaming protocol, *nStream*, is used for bi-directional low-latency audio communication.

4.2. *uGASP*

According to disparate technologies employed, a middleware layer is required to manage the logic that links together all of these components and compute the game state. For SoundPark, this is done by the uGASP[5] [PGSS08] middleware, which implements the Open Mobile Alliance Games Services (OMA GS) working group specifications that deal with multiplayer game management on embedded devices. uGASP, implemented in Java J2ME, is based on the Open Services Gateway Initiative (OSGi) component framework, and more particularly on the iPOJO OSGi layer that provide services, modularity, dynamicity and configurability through the creation, deployment and call of a bundle[6]. Interestingly, this allows for deployment, optionally dynamical, of spe-

3. We are using Mifare 1k RFID tags, which provide 1 kB of storage capacity.
4. `http://opensoundcontrol.org`
5. uGASP is available under the L-GPL license from `http://gasp.objectweb.org/ubiquitous-osgi-middleware.html`
6. From OSGi terminology, a software component providing a specified service is defined as a bundle.

cific and optimized instances of the middleware, freeing binary from unwanted functionalities.

The uGASP internal architecture is composed of various families of services, including network communication handling, session management, game server logic handling, and system services. These bundles have been employed extensively in server and client implementation. For example, the uGASP location manager bundle is used to compute projections of player positions for both 2D and 3D display described later. Another bundle was developed for handling of RFID tags on mobile phones, and dedicated to hosting uGASP mobile software components. The middleware was also improved with a bidirectional network communication connector, supporting more efficient exchanges with the scout than were available with the pre-existing HTTP connector. As seen in Figure 4, this new connector provides the communication layer used between the mobile phone and Gumstix computer. Similarly, an OSC protocol bundle was developed for the purpose of communication with the Audioscape server and the mobile devices. This bundle is employed for data reception of GPS coordinates, Wii remote events and certain game events such as players joining the game. It is also used for communication with the audioscape server for game even ts that necessitate a particular rendering, for example, acquisition of virtual objects.

uGASP, in addition of being a central part of the SoundPark server, provides bundle for deployment of a game engine at client side, that supports 2D map display, as illustrated by Figure 2, to show location of other players and virtual object. As an example of uGASP modularity, we deployed the only necessary bundle on the mobile phone, providing optimized application memory footprint small.

4.3. ***Audio Communication***

As important features for our game, our high-fidelity audio streaming engine, *nStream*, allows for enabling necessary client/server audio communications. For the purpose of walkie-talkie-like communication and client-specific sonic world rendering, a bi-directional link have been established between mobile clients and the audioscape sound server.

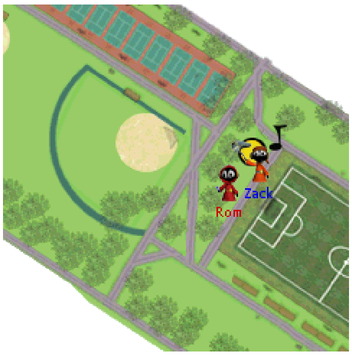

Figure 2 – 2D virtual world view, appearing on mobile phone display, illustrating two avatars, with a sound loop position indicated by the musical note, and Zack is approximately in the middle of the goal line of the football field (anchored at his feet). The yellow circle represents the staging area..

nStream supports both unicast and multicast communications, and permits reconfiguration of sender traffic by switching between different levels of PCM quantization (sample depth) and packet size, as well as number of audio channels interleaved in the stream. Dynamic reconfiguration of these parameters is enabled by specifying in the packet header the sample resolution, frequency, and number of channels. For a given protocol configuration, *nStream* provides a Constant Bit Rate (CBR) stream of uncompressed PCM audio data, using a playback buffer on the receiver end to maintain a constant latency and thus mask the effects of jitter and late arriving packets. The development of *nStream*[7] was motivated in large part by the high latencies associated with typical compressed audio formats,[8] and the unsuitability of non-WiFi solutions given their limited range, high power draw, or signal interference issues.

7. *nStream* is a Pure Data (Pd) component, also compatible with its fixed-point version, PDa. Source code is available with the Audioscape software from `http://www.audioscape.org`, distributed under the GNU General Public Licence.
8. A notable exception is the Fraunhofer Ultra-Low Delay codec [WSHK06], but this requires significant computational resources, well beyond those available on our platform.

4.4. *Audioscape*

To render audio for each player, the *Audioscape* platform [WSC06] was used to model the location of all sound sources in the environment. This was made possible by a server whose role is to maintain the state of a virtual 3D environment, and updates itself by listening to OSC messages from the uGASP game engine.

Despite its name, Audioscape is also used to provide a 3D *visual* rendering of the scene for the audience and game masters, as described below. This is accomplished with a customized graphics engine built with OpenSceneGraph[9], which represents soundNodes as arbitrary 3D models (created in programs such as Maya, 3DStudioMax, Blender, etc.). There are also several built-in graphical tools, including 3D sound level meters, and wireframe models of sound directivity. These allow the game designers to lay out sounds effectively. Audioscape stands also as the audio server, and provides real-time feedback of audio parameters during game play.

Figure 3 – 3D virtual world view, in which two players, the hunter and scout, can be seen, sounds are spread over the field.

The 3D graphics view, shown in Figure 3, which permits the game master and the audience to follow the actions of players, represented as dynamic 3D avatars, and the game status, on a virtual display. In order for this display to serve as an effective representation of the game

9. `http://www.openscenegraph.org`

state, each physical element (tennis courts, baseball diamond, walking paths, football posts, etc.) had to be modelled precisely with respect to its dimensions and physical location in the game area. Otherwise, the 3D display, and in particular, the placement of player avatars within it, would not correspond consistently to possible observation of player in the park.

5. Implementation issues

This section provides an overview of the devices used in our prototype, as well as related issues encountered during the implementation of our particular game.

5.1. *Device for mobile computing*

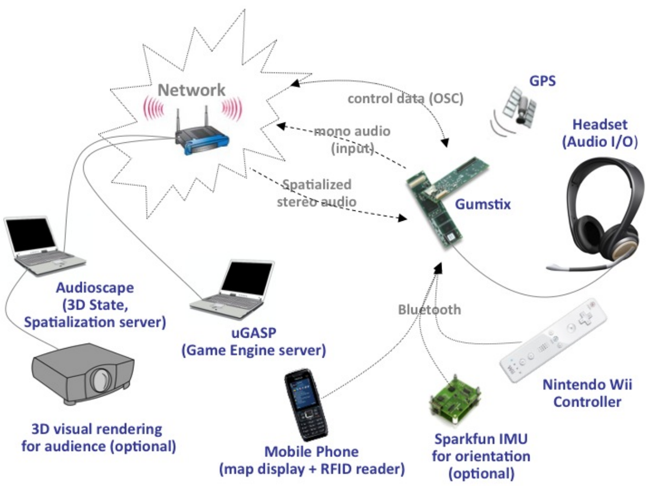

Figure 4 – SoundPark Devices Overview. Note that several of the components are unique to particular player roles (e.g., mobile phone for scouts, Wii controller for hunters).

Although game state and user-specific audio and graphic rendering are computed by centralized servers, the mobile devices support sensing of user and environment, and are used for local delivery of audio and graphics. As seen on Figure 4, we selected the Gumstix computer platform[10] for our prototyping effort. Our configuration, pictured in Figure 5, consists of a Verdex XM4-bt main board with expansion boards for memory, 802.11g (WiFi) communication, bidirectional high-quality audio, and differential GPS, Linux support, extremely small size and light weight, plus a portable USB power pack and antennas for Bluetooth and WiFi. Connectivity to other devices (e.g., mobile phone and Wii controller) is supported over Bluetooth.

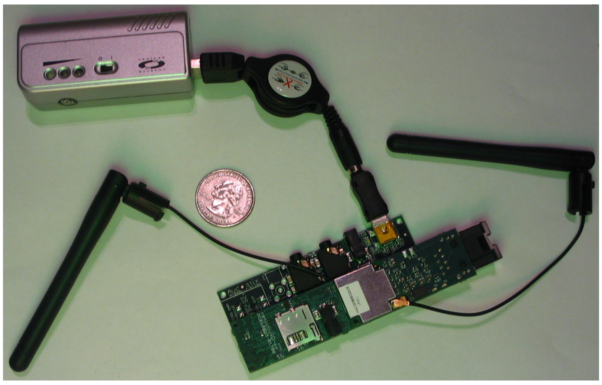

Figure 5 – Gumstix computer platform, with USB power pack and antennas, shown with a quarter (left of center), for reference purposes.

Software development for the ARM-based 400MHz Gumstix processor (Marvell PXA270) requires a cross-compilation tool-chain to generate binaries. Floating point operations are supported, but computed through software emulation. Thus, any significant DSP computations should be written to use fixed-point operations. Our software that runs on the Gumstix was implemented in C, as a set of *Pure Data* (Pd)[11] objects. We were able to adapt and cross-compile some pre-existing externals such as the OSC protocol but also had to develop platform-specific software for our real-time, dynamically reconfigurable, low-latency streaming protocol (see Section 4.3) as well as sensor acquisition code for the GPS receiver and Wii controller. The Gumstix is also

10. `http://www.gumstix.com`
11. Pure Data is a real-time graphical programming environment for audio. Because of the fixed-point restriction of this platform, we use *Pure Data anywhere* (PDa) [Gei03], a rewritten fixed-point version, based on Pd v0.37, intended for PDA devices.

used as a communication bridge between the mobile phone (using Bluetooth communications) and game logic server (over WiFi), as discussed in Section 4.2).

It is worth noting that our software is independent of the specific hardware described above. For example, the mobile phone can be replaced directly by any device providing a JME runtime, equipped with an RFID reader and Bluetooth interface. Similarly, our Gumstix computers can be replaced by another Linux-based small form-factor board providing similar I/O interfaces.

5.2. *Implementation feedback*

The mobility requirement imposed hard limits regarding weight, size, and power autonomy. Solutions had to be found to extend battery life among the various components, ranging from the Gumstix to the WiFi network and field servers, while maximizing power. Protocols for OSC-based data exchange between the different devices and services used had to be developed.

In terms of logistics, the equipment *worn* by players should be mounted solidly but with sufficient ventilation so as to not overheat. Our prototyping experience quickly indicated that sealed boxes of electronics operating on a hot summer day do not mix well. Ideally, the gear should be fixed securely to each player in a manner that nonetheless allows for rapid putting on and taking off. For the next experiments we have to cope with constraints from wearable computing [PT02].

During our experiment, the game space was covered entirely by a wireless network configured in managed mode. Robustness was sufficient for supporting the traffic generated by our application. These preliminary results are encouraging and motivate us to investigate in the near term a more dynamic configuration that uses an ad-hoc networking strategy. To meet this new challenge, we are investigating adaptive protocols for periodic exchange of sensor data, possibly at non-trivial bandwidths, and that supports network resource variations and disconnections.

The dynamic nature of GPS performance in the game area should be observed and taken into account : GPS signal loss conditions must be integrated into the user interface to alert players and audience members to the loss of accuracy. It is also important to note that the general inaccuracies of GPS affect both the process of map calibration during development of game content as well as the players' sense of their positions with respect to the displayed content. Although our differential GPS sensors provide reasonable precision for most areas of the game environment, the extracted positions from GIS maps (whether from Google or Yahoo !) lead to correspondence errors with our own readings. This, in turn, frustrates our manual process of calibrating between the 2D (mobile phone display), 3D (Audioscape representation), and GPS coordinates, as needed for a consistent rendering of the players and their environment.

6. Toward a fully distributed architecture

A fully distributed game architecture requires that each device maintains its own game state, updated by interactions with its peers. This necessitates availability of several services, as described in the remainder of this section. In particular, our experience with SoundPark motivates us to emphasize low-latency group communication, distributed game state management and mobile computing capability.

6.1. *Group communication*

Low-latency group streaming over a wireless network remains a major challenge, in particular for audio. In this regard, a distributed architecture offers advantages of scalability and lower latency than a centralized architecture, as this would support direct, peer-to-peer communication rather than requiring such communication to route through a server. Although our present implementation remains centralized, we consider here some of the design decisions that might be considered as we move toward such a distributed architecture. For example, we must take into account the need to ensure fair access to the network not only for audio transmission but also for other data exchange. Dujovne and Turletti

[DT06] conducted IP multicasting experiments over 802.11b networks in infrastructure mode, concluding that Constant Bit Rate (CBR) multicast streams reduce the opportunities for unicast transmitters to acquire network access at the MAC layer, which results in potentially unfair sharing of bandwidth.

Network topology is another important question. For mobile ad-hoc networks (MANET), Benslimane noted that*mesh* approaches support multiple paths between any source and receiver pair and are thus preferred over *trees* where network topology may change frequently [BFH07]. Explicit Multicast approaches support Single Source Multicast (SSM), which requires the sender to specify a list of receivers. Unfortunately, there has been little experimentation with these methods outside of laboratory settings. Further investigation is required to study the quality-of-service issues related to SoundPark : reliability of event dissemination, audio streaming latency and support for dynamic membership.

6.2. *Game state management*

While distributed data management, including storage and consistency aspects, has been widely studied, distributed real-time games require additional attention to temporal constraints such as application responsiveness, event synchronization, and prevention of ordering conflicts [BGS04]. Use of a consistency protocol combined with dead reckoning has been proposed as one possible solution [Bou05]. However, further investigation is required to determine how well this copes with possible disconnections, as are frequently encountered in wireless network conditions.

Contextual data can be stored entirely in a mobile device or distributed among several such devices. While local storage is an easy-to-implement solution, dynamic fetching should prove to be more scalable, but requires content allocation among distributed (mobile) hosts. This can be facilitated by contextual metadata descriptions. Options for collaborative and dynamic data management include the use of a Delaunay triangulation variant [BA08], which determines the connectivity graph between hosts based on their geographic location, and its corresponding dual structure, the Voronoi diagram, which may delineate the extent of data exchange.

6.3. *Mobile computing*

We recognized early in our implementation effort that the current generation of tiny mobile platforms[12] were inadequate to perform most of the necessary computation locally [WBSC08], in particular with respect to managing computation of game state and audio rendering. We thus decided to rely on such devices exclusively as a conduit for input and output. As mobile devices constantly increase in power, these should soon be capable of supporting the necessary local computation of audio spatialization and high-level communications, simultaneously with game state management and display.

7. Conclusion

SoundPark illustrates the possibilities of an ubiquitous computing architecture as applied to the highly engaging social activity of a multi-player mobile game. Taking advantage of high-precision location tracking, reasonably powerful computational resources, and low-latency audio interaction, we have developed an application that is at once demanding of the underlying technologies but also enjoyable for its users.

Our research effort combines multiple worlds, following a mixed reality approach : the physical reality of the Jeanne Mance park, a virtual 2D view, seen on the scout's mobile phone, a virtual 3D view, seen by the audience, and a rich spatialized audio environment, accessible to all : players, audience, and game masters. Whereas most mobile games today rely on proximity to target zones to trigger game events, our application takes place in a fully continuous and physically modelled environment with audiovisual content.

12. We refer specifically to devices such as the Gumstix. Ultra-mobile PCs are becoming increasingly capable in computational power but remain somewhat awkward in size and weight for effective mobile game-play. Regardless, the present effort should be understood as a proof of concept implementation.

SoundPark represents an early attempt to build highly interactive, ubiquitous games, which integrate a variety of technologies. Our experience presented us with many of the challenges one expects in building modern applications, attempting to develop a modestly complex, ubiquitous multi-player game. Requirements included maintaining lossless, low-latency communications across a heterogeneous mix of processors and programming languages, rendering 2D/3D mixed reality views of the environment, and supporting GPS localization, RFIDs and other sensors on resource-constrained embedded processors. The development effort was conducted in only three months–a very short time, especially considering that this was the first occasion in which the group members worked together on the same project. An important factor in the success of this effort, was the use of a component-oriented middleware (uGASP), which facilitated the integration of heterogeneous computing units.

Références

[BA08] Eliya Buyukkaya and Maha Abdallah. Efficient triangulation for p2p networked virtual environments. In *NetGames '08 : Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, pages 34–39, New York, NY, USA, 2008. ACM.

[BFH07] Abderrahim Benslimane, Cédric Ferraris, and Abdelhakim Hafid. A comparative analysis of multicast protocols for small manet groups. In *ADHOC-NOW*, volume 4686 of *Lecture Notes in Computer Science*, pages 213–225. Springer, 2007.

[BGS04] Nicolas Bouillot and Eric Gressier-Soudan. Consistency models for distributed interactive multimedia applications. *SIGOPS Oper. Syst. Rev.*, 38(4) :20–32, 2004.

[Boh04] J. Bohn. The smart jigsaw puzzle assistant : Using RFID technology for building augmented real-world games. In *Workshop on Gaming Applications in Pervasive Computing Environments*, 2004.

[Bou05] N. Bouillot. Fast event ordering and perceptive consistency in time sensitive distributed multiplayer games. In *CGAMES'2005 7th International Conference on Computer Games*, pages p.146–152, Angouleme, Nov. 2005.

[CGL+04] Adrian David Cheok, Kok Hwee Goh, Wei Liu, Farzam Farbiz, Siew Wan Fong, Sze Lee Teo, Yu Li, and Xubo Yang. Human pacman : a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal Ubiquitous Comput.*, 8(2) :71–81, 2004.

[DT06] Diego Dujovne and Thierry Turletti. Multicast in 802.11 WLANs : an experimental study. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 130–138, 2006.

[FBA+03] Martin Flintham, Steve Benford, Rob Anastasi, Terry Hemmings, Andy Crabtree, Chris Greenhalgh, Nick Tandavanitj, Matt Adams, and Ju Row-Farr. Where on-line meets on the streets : experiences with mobile mixed reality games. In *CHI '03, Proceedings of the SIGCHI conference on Human factors in computing systems*, pages pp. 569–576, New York, USA, 2003. ACM.

[FKS07] I. Flammer, W. Ka, and R. Skraba. Meet your heartbeat twin. In *Proceedings of 4th International Symposium on Pervasive Gaming Applications PerGames 2007*, pages page 157–158, Salzburg, Austria, June 2007.

[Gei03] Gunter Geiger. PDa : Real time signal processing and sound generation on handheld devices. In *International Computer Music Conference*, 2003.

[GN06] F. Girardin and N Nova. Getting Real with Ubiquitous Computing : the Impact of Discrepancies on Collaboration. *eMinds*, 1, 2006.

[HGMJ07] G. Heumer, F. Gommlich, A. Müller, and B. Jung. Via mineralia - a pervasive museum exploration game. In *4th International Symposium on Pervasive Gaming Applications, PerGames*, pages pp. 159–160, 2007.

[Jof05] Benjamin Joffe. Mogi, Location and presence in a pervasive community game. In *Proceedings of the Seventh International Conference on Ubiquitous Computing, Ubicomp. Ubiquitous Computing, Entertainment, and Games Worshop*, Tokyo, September 2005.

[NY07] S. Natkin and C. Yan. Adaptive narration in multiplayer ubiquitous games. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 2007. IGI Publishing.

[Pel07] R. Pellerin. The MooDS protocol : a J2ME object-oriented communication protocol. In *Mobility '07 : Proceedings of the 4th international conference on mobile technology, applications, and systems*, pages 8–15. ACM, 2007.

[PGSS08] R. Pellerin, E. Gressier-Soudan, and M. Simatic. uGASP : an OSGi based middleware enabling multiplayer ubiquitous gaming. In *International Conference on Pervasive Services, ICPS 2008 Demonstration Workshop*, Sorento, Italy, July 2008.

[PT02] Wayne Piekarski and Bruce Thomas. Arquake : the outdoor augmented reality gaming system. *Commun. ACM*, 45(1) :36–38, 2002.

[RBC+06] Omer Rashid, Will Bamford, Paul Coulton, Reuben Edwards, and Jurgen Scheible. PAC-LAN : mixed-reality gaming with RFID-enabled mobile phones. *Comput. Entertain.*, 2006.

[RKD00] Joseph Rozier, Karrie Karahalios, and Judith Donath. Hear & There : An augmented reality system of linked audio. In *Intl. Conference on Auditory Display*, 2000.

[Sch08] Jan C. Schacher. Davos Soundscape, a location based interactive composition. In *New Interfaces for Musical Expression*, Genova, Italy, 2008.

[SHG+07] S.P. Stenton, R. Hull, P.M. Goddi, J.E. Reid, B.J. Clayton, T.J. Melamed, and S. Wee. Mediascapes : Context-aware multimedia experiences. *Multimedia, IEEE*, 14(3) :98–105, July-Sept. 2007.

[Sot02] O. Sotamaa. All the world's a botfighter stage : Notes on location-based multi-user gaming. In Edited by Frans Mäyrä. Tampere, editor, *Proceedings of Computer Games and Digital Cultures Conference* CDGC'02, Finland, June 2002.

[WBSC08] Mike Wozniewski, Nicolas Bouillot, Zack Settel, and Jeremy R. Cooperstock. Large-scale mobile audio environments for collaborative musical interaction. In *Intl. Conference on New Interfaces for Musical Expression*, Genova, Italy, 2008.

[WSC06] Mike Wozniewski, Zack Settel, and Jeremy R. Cooperstock. A paradigm for physical interaction with sound in 3-D audio space. In *Proceedings of International Computer Music Conference (ICMC)*, 2006.

[WSHK06] Stefan Wabnik, Gerald Schuller, Jens Hirschfeld, and Ulrich Krämer. Reduced bit rate ultra low delay audio coding. In *Proceedings of the 120th AES Convention*, May 2006.

ANNEXE POUR LA FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE

1. ARTICLE POUR LA REVUE :
   Studia Informatica Universalis.

2. AUTEURS :
   *Romain Pellerin * — Nicolas Bouillot ** — Tatiana Pietkiewicz **
   *— Mike Wozniewski ** — Zack Settel ****
   *— Eric Gressier-Soudan * — Jeremy R. Cooperstock ***

3. TITRE DE L'ARTICLE :
   SoundPark: Exploring Ubiquitous Computing through a Mixed Reality Multi-player Game Experiment

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :
   SoundPark

5. DATE DE CETTE VERSION :
   14 juillet 2010

6. COORDONNÉES DES AUTEURS :

– adresse postale :
  * Conservatoire National des Arts et Métiers, Paris, France
  ** McGill University, Montreal, QC, Canada
  *** University of Montreal, Montreal, QC, Canada

– téléphone : +1 514-398-5992
– télécopie : +1 514-398-7348
– e-mail : nicolas@cim.mcgill.ca

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :
   LaTeX, avec le fichier de style `studia-Hermann.cls`,
   version 1.2 du 03/12/2007.

SERVICE ÉDITORIAL – STUDIA UNIVERSALIS
41 rue Gay Lussac, 75005 Paris
Tél : 01-44-10-84/83
mel : contact@complexica.net
Serveur web :