Mobile Context-Aware Game for the Next Generation

Maomao Wu, Keith Mitchell, Duncan McCaffery, Joe Finney, and Adrian Friday

Distributed Multimedia Research Group
Computing Department, Lancaster University, UK
{maomao, km, mccaffed, joe, adrian}@comp.lancs.ac.uk

Abstract

Traditional networked multiplayer games are typically confined to be played on desktop computers. With the recent advances in mobile networking, context-aware computing, and sensor-based computing, researchers and game designers are able to explore the potential of combining these new technologies together to develop mobile networked, context-aware, augmented reality multiplayer games.

As part of the new research collaboration between Lancaster University, Cisco Systems, Microsoft Research and Orange – MIPv6 Systems Research Lab (MSRL), such a mobile context-aware multiplayer game is proposed and explored. The proposed game, named Real Tournament, gathers real-time contextual information, e.g., physical location, orientation and other personal facets from the players and injects them into the game arena to generate game events. It is not only the interaction with the virtual game world that drives the game process, but the interaction with the physical world as well.

This paper introduces the game scenario that enabled us to explore the features in the aforementioned areas simultaneously. The proposed game architecture and components are described in detail, and the benefits of our architecture are showed by comparison with the existing approaches. The paper also describes the wireless overlay network infrastructure, and demonstrates how the advanced features in Mobile IPv6 are explored during the development of the game. Moreover, a security mechanism is designed at the network-level, and it can be easily utilized for other network services.

1. Introduction

Previous research experiences in next generation mobile networking [2], context-aware computing [3,4], and recent sensor-based computing [9] has made it possible for us to explore the impact of ubiquitous computing [10] on the lives of real people in Lancaster. One of the novel applications we are investigating as one of the MSRL's collaborative projects [1] is a mobile context-aware multiplayer team game, known as Real Tournament (RT), which is targeted for children aged 11-12 years old and containing a variety of educational, mental, physical and entertainment based challenges. The game is designed to be played in a real world environment using handheld personal computing devices augmented with an array of sensors, which are used to gather the game player's context, e.g., location, orientation, activity, and identity, etc. Different wireless communication technologies have been explored to create the underlying overlay network infrastructure, and internetworking is enabled by the Mobile IPv6 protocol. New features in the Mobile IPv6, e.g., auto-configuration, smooth handoff, extension headers, have been actively utilised during the process of the game design and implementation. An additional security mechanism has also been integrated into the network layer to protect both our open wireless network and the privacy of the end users.

The rest of the paper is structured as follows. We start introducing the previous research in Lancaster and motivations that impels us to develop this specific novel application. Then we describe the carefully designed game scenario that enables us to actively research the features in three different areas simultaneously. In section four, we demonstrate the benefits of our game architecture to the existing approaches, and enumerate the game components followed by their implementation status. The overlay network infrastructure is introduced in section five, as well as the integrated security mechanism. Related work is presented in section six, and finally we make our conclusions and propose some future work.

2. Background

The Mobile IPv6 protocol enables mobile computers to roam between different IPv6 sub-networks while maintaining location transparency to correspondent nodes. In the LandMARC project [2], we have implemented the mobile extension to the original IPv6 stack for Microsoft Windows operating system in order to support further research into mobile computing. The stack has different versions for various Windows operating systems, including 2K, XP, CE3.0, 4.1 and the Pocket PC 2002. The availability of Mobile IPv6 stack on Pocket PC allows us to use handheld devices, such as the Compaq iPAQ to do research experiments and user trials.

At Lancaster, we investigate the context-aware computing domain through the development and continued refinement of a context-aware tour guide system, called GUIDE [3,4]. Equipped with a GUIDE unit borrowed from local tourist information centre, the visitors to Lancaster will receive dynamic location-specific information while they are roaming in the city. Apart from location, the system also utilises other types of context, such as user's preferences, to tailor the information presented to the user. More recently, the second phase of the project GUIDE II [5] aimed to provide context-sensitive mobile services for the residents via a wireless network in the city of Lancaster. A secure access control mechanism [6,7] was designed in order to open up our wireless network access and context-sensitive services to authorised users only.

As part of the European research initiative The Disappearing Computer [8], the recent Smart-Its project [9] is interested in embedding small-scale computing devices into mundane everyday artefacts so that they will be able to enter into dynamic digital relationships. By augmenting these artefacts with sensing, perception, computation, and communication capabilities, they can become the building blocks of a generic software platform capable of context capture and distribution for large scale ubiquitous computing applications.

3. Motivation

MSRL has been established to build on our prior work within the areas of mobile networking, context-aware computing, and sensor-based computing. In our current research, we actively seek to explore the features in these specific areas simultaneously. The work involved in MSRL can by categorized into three vertical layers: network infrastructure, middleware or services, and applications. The higher two layers should actively make use of the new features in the next generation mobile networking protocol. We believe that only through the process of development, deployment, and evaluation can we have a better view of the impact of Mobile IPv6 on future network services and applications.

The traditional game industry has grown at an escalating speed in recent years. As the Internet becomes more popular and enters into the consumers' homes, the proliferation of online multiplayer games has increased dramatically. We believe that it is only a matter of time before such applications become prolific on mobile terminals too. It has been predicted that by the end of 2003, eighty percent of mobile telecom devices will have the capability to support server-based or downloadable games, a market which is expected to grow to cover nearly 200 million consumers and 6 billion US dollars by 2005 [11]. We therefore see an enormous potential in value added services such as multiplayer games beyond the current trend towards downloadable games.

As a result of previous successful research experiences in Lancaster and the keen insight into the networked game industry, we proposed and designed Real Tournament – a mobile networked, context-aware, augmented reality multiplayer game.

3. Game Scenario

Bearing the mind that we should actively explore the features in the aforementioned areas simultaneously, a proper game scenario needs to be carefully designed. Apart from the technical issues, the availability of the end system users is also required in order to evaluate our game by user trials. We finally decide to design a game full of educational, mental, physical and entertainment based challenges for children at 11-12 years old.

The game arena of Real Tournament is set in a public park close to the city centre of Lancaster – "Williamson Park". The park has a memorial (Ashton Memorial), a butterfly house and a mini-beasts museum housing lizards, spiders, snakes and a variety of other creepy-crawlies. The narrative for the game relates to a mad professor who has time-travelled back from the future, and invaded the mini-beasts museum and transformed the creatures into large monsters. These monsters have become the professor's accomplices and are to aid him in his quest to take over the world. The professor has released these virtual monsters into the wild causing disruption around the city and surrounding area.

The players are free to travel through a series of historical zones in the park, and collaborate to locate and capture virtual monsters that roam these zones in order to get highest points for their team. The difference of Real Tournament when compared with traditional multiplayer games is that it augments the virtual game world with real world artefacts. The most obvious example of real world artefacts is the player, whose contextual information, e.g., location, orientation, and status, is used to augment the virtual game environment of other users. The game players can monitor their virtual representation moving within the game arena through their handset while they are walking in the physical world. They can also use audio conferencing utilities to facilitate collaboration and synchronise movements in order to approach their target at sufficient distance before using a capturing facility.

Additional game points can also be obtained by solving a variety of mental and physical challenges in certain zones of the park. During a game session, players are able to collect both physical and virtual artefacts located around the game arena, such as utilities or power-ups. The game's narrative is supplemented by the use of video content streamed from game servers located at various vantage points around the game arena.

4. Game Architecture

There are two popular architectures for the existing networked multiplayer games: client-server and peer-to-peer. In the client-server model, there is a centralised server that stores shared data and relays messages between clients. More specifically, clients send user generated game events to the central server, where they are processed and transmitted back to all relevant clients. This approach guarantees data consistency and synchronises the game status between different players. But there are reliability and performance issues, since all computation and communication is processed on the centralised server. In the peer-to-peer model, there exists no central repository for game state, but each client needs to maintain a replica of the game state and update it in response to either locally generated events or remotely received game events. By localising the computation at the client side, this model reduces the network traffic but introduces consistency problems at the same time.

By trading off the pros and cons, we are thinking of adopting a hybrid solution for the Real Tournament game architecture, known as Mirrored Server Architecture (Figure 1) [12]. Instead of having a single central server, there exist several distributed servers for each game session. Each mobile game client initially connects to the geographically closest game server and communicates with it in the normal fashion. We currently make use of a simple UDP based discovery protocol to allow the mobile clients to obtain an IPv6 address for their local game server. The game servers are then interconnected over a fixed IPv6 low latency multicast network reserved for game traffic only. They exchange state using the peer-to-peer architecture, and each server maintains its own copy of the game arena.

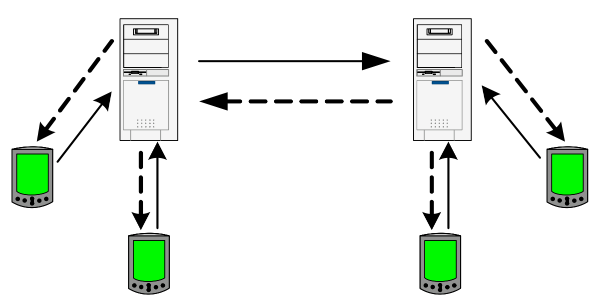

Figure 1: Mirrored Server Architecture.

The existence of multiple servers removes the bottleneck and single point of failure found within traditional client-server architectures. This also means that the complexity in terms of communication exists primarily between servers (where bandwidth is cheap and reliable) instead of the clients. However, we are actively looking for ways to extend and refine this architecture to make it more suitable for our own game requirements.

4.1. Game Client

The Real Tournament handset (Figure 2) is a modified Super Soaker XP70 water cannon augmented with a Trimble Lassen™ SQ GPS module, a electronic compass, two micro-switches (used for firing and utility selection), a Smart-Its sensor board [13], a Bluetooth enabled Compaq iPAQ Pocket PC with Compact Flash (CF) expansion jacket and a low power wireless 802.11b network card. The Smart-Its sensor board includes a touch sensor, infra-red light sensor and an accelerometer, although not all of these have been incorporated into the game yet.

Figure 2: (a) A complete handset with iPAQ; (b) Cut-away view of a Super Soaker XP70 showing GPS, compass, trigger and battery.

All these components communicate via the standard RS232 serial interface to the iPAQ Pocket PC. For this to work, the GPS device, the digital compass and the triggers are multiplexed via a PIC microcontroller on the RS232 interface of the iPAQ. The data is transferred in plaintext ASCII format over a simplex 9600 baud RS232 connection. Data items are sent line delimited by CR/LF and in no particular order, just as the data becomes available from each hardware device. GPS data is received and sent across the serial line once per second, orientation information is processed 10 times per second and the trigger status is received on change.

Client side software consists of a variety of components that can be categorized into five types according to their different functionalities: a serial port communication processing module that gathers raw sensory data from the hardware sensors; a parsing and translation module that transforms the raw sensory data into software events; a communication module that sends software events to the game server and receives updates from it; a module that renders graphics and plays sounds to the end user; and audio conferencing utility client software.

These software modules are written in a combination of languages including Microsoft C#.NET, C++, and Embedded C++. C#.NET enables us to rapidly prototype and maintain client functionality while C++ allows us to make use of lower level functionality currently not supported by the .NET Compact Framework [14], such as serial port access, Mobile IPv6 communications and high speed graphics rendering. A number of third party gaming libraries were also utilised to provide graphics and sound support.

4.2. Game Server

A series of mirrored game servers maintain all game logic and state, and they are responsible for disseminating updates to each player's mobile device. More specifically, the game servers maintain a series of data structures representing each player's identity, location, playing status, and the status of all the virtual artefacts and monsters in the game. They are responsible for ensuring that all the mobile clients have a consistent view of the state of the game arena.

A player's physical location and orientation within the real world is mapped to their virtual location and orientation within the Real Tournament game arena. Therefore physical movements and changes in direction generate events which are mediated by the system and mapped in real time to each user's handset and display. The system is mediating and controlling all system wide events although in most cases these are generated initially by the player's themselves. The game servers are able to filter incoming events from the mobile clients and disseminate them to the relevant parties based on context such as location and player status. The approach taken in ensuring consistency of state between servers and clients is beyond the scope of this. In addition to that, we are also trying to develop a generic approach to distributing contextual events to a large number of users.

In the current implementation, the game servers run on standard Windows XP PCs and are located inside the Ashton Memorial in the centre of the Williamson Park. One of the game servers is connected to a large plasma display which is used to display the overall game arena and allow spectators (including parents, developers and system administrators) to watch the game session unfold. The software running on game servers are implemented in a combination of C#.NET and C++ in the Microsoft .NET environment.

5. Network Infrastructure

As one of the main tasks of the MSRL, we are currently in the process of deploying a Mobile IPv6 overlay network infrastructure designed to facilitate ubiquitous access to information and services throughout the city of Lancaster and the University campus. We have already deployed several Wireless LAN (802.11b) base stations that cover the whole area of the Williamson Park, where the game takes place. Furthermore, the use of GPRS [20], as an "umbrella" for ubiquitous coverage across the whole area, will provide the project with a real-world testbed that enables experimentation with standard wireless overlay networks.

We are using Cisco Systems' 350 series base stations to provide 802.11b wireless coverage around Williamson Park. The antennae used with these base stations are 80 degrees diversity patch antennae from Cisco Systems, which have two separate transmit/receive elements in them. This can be advantageous as they can improve reception by detecting which element has the best signal at any instant in time, and rapidly switch to using that element. The antennae are located to the North, South, East and West sides of the building and are tilted downwards at an angle of 10 degrees to provide coverage as near to the base of the Ashton Memorial as possible. Signal tests around the Park have so far been very positive with each area of coverage having a slight overlap with the neighbouring cell. Data link and network level handoffs have also been tested and work well.

5.1. Network Security

The open nature of the network infrastructure requires a certain degree of protection both for the network itself and users of the network from accidental or malicious misuse of the system. The security mechanism we designed is based on IP level packet marking, encryption and filtering, and it is integrated into the Mobile IPv6 stack on Windows platforms (2K, XP, and CE). The components of our public access wireless network infrastructure for the game are illustrated in Figure 3. The access control mechanism we designed is so generic that it can also be used for other network services, e.g., an Internet access service.

Each wireless cell in the diagram is controlled by a base station, which is a link layer entity that operates transparently from the perspective of IP layer. Every base station is associated with an *Access Router (AR)* that controls the access to the campus network and the Internet. The game servers are collocated with the base stations, and they are put behind the Access Routers. Access Routers block the traffic originating from unauthorised end devices based on network-level packet filtering, and each AR forms a single routable IP sub-network, named a *District*. In order to gain access to the services, mobile devices need to authenticate themselves with the *Authentication Server (AS)* first. Upon successful authentication and authorisation of a user, the AS issues a limited lifetime access token to the user, which provides the basis for the client to do the packet marking. In essence, our packet marking scheme is a hybrid way of digitally signing the whole IP packet, including the headers and payload, and this is placed in an IPv6 extension header. Optional encryption of the IP payload can also be realised for confidential communications using the same session key.

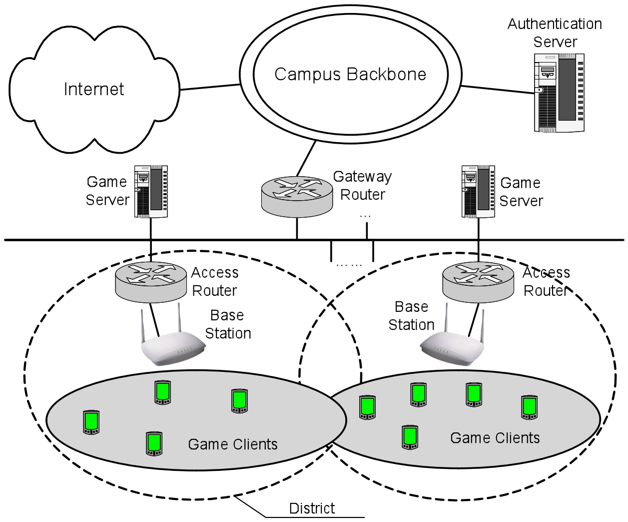

Figure 3: Secure Network Infrastructure.

In order to prevent malicious users stealing the token, a secure token and session key distribution protocol is designed, and its lifetime is also restricted, known as *Refresh Time*. Each AR maintains an access control list, which accommodates all the information required to validate the digital signature in the packet, namely the MAC address, IPv6 address, access token, and session key. This information is securely distributed to the AR by the AS after successful authentication of a mobile user. When a packet comes, the AR looks up the access control list using MAC address as an index. If an entry exists, the AR verifies the validity of the signature in the packet by using the values in the entry. When successful, the access control extension header is either stripped off or padded with a certain value and the packet is passed on; otherwise, the packet will be dropped. One exception to this rule is that when a client first turns up in a cell, it is allowed to contact certain well-known IPv6 addresses, e.g., authentication server address.

The mobile IPv6 protocol supports roaming from a network-level point of view (i.e., location transparency and fast network handoffs). When a mobile node moves to a new sub-network, the network access will be controlled by a new AR, which may have no knowledge of previous authorisations for the mobile node. We solved this problem by using a proactive approach: the AS not only sends the token triple to the current AR, but distributes it to the neighbouring ARs as well after successful authentication of a mobile user. When the user roams to a neighbouring network, the new AR already has the necessary information to verify the signature by the mobile node.

We also put a *Gateway Router* between the public network and campus intranet in order to prevent someone masquerading our ARs. This gateway router will only accept traffic originating from the access routers whose MAC addresses have been registered in its access control list.

In order to reduce the number of the access routers, we exploit the Virtual LAN (VLAN) technology. Working with a VLAN capable switch, a single router equipped with a VLAN capable network interface card can have many virtual network interfaces so that it can route packets from different sub-networks. The access router is based on off-the-shelf PCs running Windows .NET Server 2003, and it is equipped with a 3COM Gigabit Server Network Interface Card (3C996B-T), which supports up to 64 virtual networks on a .NET server platform. For the switching of the virtual LAN networks, we use Gigabit switches from Extreme Networks (models 5i and 48i switches) and Cisco (model 3550-48 switches). The gateway router is based on a Cisco 7206VXR router running IOS version 12.2. The authentication server is hosted on a special service network connected via the gateway router.

6. Related Work

Research within the field of virtual and augmented reality has recently moved into outdoor environments and aims to combine the virtual world of games with the physical world in new and engaging ways. However, support for mobile users within a wireless heterogeneous networked environment does not yet exist. This section provides a brief overview of the current systems available and outlines their research aims and objectives and how they differ to our work on Real Tournament.

“Pirates!” [15] is a multiplayer indoor game for handheld devices connected to a single wireless LAN cell. Each player’s device is equipped with a proximity sensor used to sense relative proximity to other players and locations in the real world. Players’ movement between locations triggers game events and allows them to engage in a variety of activities such as exploring islands and taking part in sea battles [16]. Proximity was used to explore the use of social interaction to provide a richer game experience in a social setting. However, it was beyond the scope of “Pirates!” to consider scalability issues. Neither the hardware infrastructure or software architecture would provide adequate performance given a large user base.

“Can You See Me Now?” [17] is a mobile mixed reality game in which online players were chased across a map of a city by players physically located on its streets. Online players accessed a map over the Internet to move their characters across the city. Runners equipped with handheld wireless devices augmented with GPS receivers chased them by running through the city streets. Online players were able to communicate with each other via text messaging, and had the ability to listen to audio streamed from the runners' walkie-talkies. The focus of this work was based around ethnography, player interviews and examining logs of game activity which revealed issues pertinent to citywide mixed reality games [18]. While much can be learned from their experiences, our work is more focused on systems level issues of deploying and utilising MIPv6 wireless overlay networks.

"Bot-Fighters" [19] utilises cellular phone features such as cellular positioning and SMS to provide a location based combat game. Bot-Fighters users are able to go to the game's website and locate other players in their area. They may then send an SMS (or use a WAP phone) to check their target's physical location. If the target is within range, the player can shoot by sending a "fire" SMS. Players receive SMS messages regarding who is in their vicinity and their attack status. By replying to SMS messages they are able to virtually retaliate or flee from attack. Although this application is highly scalable, there are no real-time constraints and the user interface is extremely limited. We intend to make use of rich multimedia content to provide an engaging experience to our users.

7. Conclusions and Future Work

This paper has introduced a mobile-networked, context-aware, augmented reality multiplayer game named Real Tournament. We believe Real Tournament distinguishes itself from the other networked multiplayer games by a number of important features, namely:

- A carefully designed game scenario. It enables us to actively explore the features in three different research areas simultaneously. It also makes it easy to evaluate our game through user trials by targeting the game players to the children at age of 11-12.
- Custom designed and built handsets. By making the user interface presented to game players customized to the task at hand, this greatly improves the players' gaming experience.
- Real-time communications. By adding real-time voice streaming between players, this adds an additional element to the game play. (By allowing players to collaborate more effectively).
- Context-awareness in the game. By injecting real-world contextual information (such as the players' physical location) into the game, the way of interacting with the computing device becomes more natural, and the game players will feel more comfortable to concentrate on the game content.
- An overlay network infrastructure based on Mobile IPv6. The advanced features in Mobile IPv6 assist the players' devices in maintaining a consistent view of the gaming arena, by allowing devices to communicate over any available network medium.
- A hybrid game architecture. By replicating game servers and interconnecting them by low latency multicast network, it not only eases to maintain the consistency and synchronization, but also increases the availability and performance.
- A network-level security mechanism. Our security mechanism, which is integrated into the Mobile IPv6 stack, allows game operators to regulate access to the wireless networks used to support the game. The mechanism is also generic enough that we can easily use it for other network services.

In the future work, we intend to further develop and evaluate the hybrid mirrored server model described in this paper, and investigate its role in extremely wide area applications such as maintaining context information for multimedia messaging applications. We also intend to design a context manager for efficient, scalable support for context management of both real time and non-real time data, and to make it generic enough to support other context based services. Other latest wireless communication technologies, such as Bluetooth, GPRS, will also be explored to extend the underlying network infrastructure and to realise the Mobile IPv6 overlay network.

References

[1] Mobile IPv6 Research Laboratory (MSRL), Collaboration with Cisco Systems, Microsoft Research and Orange, Lancaster University, available via the Internet at http://www.mobileipv6.net/testbed, February 2001.

[2] The LandMARC Project Lancaster and Microsoft Active Research Collaboration, available via the Internet at http://www.LandMARC.net, October 1999.

[3] N. Davies, K. Cheverst, K. Mitchell and A. Friday, "Caches in the Air: Disseminating Information in the Guide System", in Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), New Orleans, February 1999.

[4] K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences", in Proceedings of CHI'00, Netherlands, pp 17-24, April 2000.

[5] "The Guide II Project: Context-Sensitive Mobile Services for City Residents", Research Project, Lancaster University,

EPSRC Grant GR/M82394, available via the Internet at http://www.comp.lancs.ac.uk/computing/users/maomao/G2/, 2000.

[6] S. Schmid, J. Finney, M. Wu, A. Friday, A.C. Scott, and W.D. Shepherd, “An Access Control Architecture for Microcellular Wireless IPv6 Networks”, in Proceedings of 26th Annual IEEE Conference on Local Computer Networks (LCN2001), Tampa, Florida, November 2001.

[7] A. Friday, M. Wu, J. Finney, S. Schmid, K. Cheverst, and N. Davies, “Network Layer Access Control for Context-Aware IPv6 Applications”, Accepted to ACM Baltzer Wireless Networks WINET Special Issue, 2002.

[8] The Disappearing Computer Initiative, an EU-funded proactive initiative of the Future and Emerging Technologies (FET) activity of the Information Society Technologies (IST) research program, available via the Internet at: http://www.disappearing-computer.net/, 2001.

[9] “Smart-Its”, Collaboration of Lancaster University, ETH Zurich, University of Karlsruhe, Interactive Institute and VTT, available via the Internet at http://www.smart-its.org, 2001.

[10] Mark Weiser, "The Computer for the Twenty-First Century", Scientific American, pp. 94-10, September 1991.

[11] Shankar, B., Brown, A., MacMillan, D., ODonovan, P. and Wood, B., "Games and Entertainment within the Mobile Phone Industry", also available via the Internet at: www.gartner.com, April 2002.

[12] Cronin, E., Filstrup, B., Kurc, A.R. and Jamin, S., “An Efficient Synchronization Mechanism for Mirrored Game Architectures”, in Proceedings of NetGames2002, April 2002.

[13] Holmquist, L., E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H., W., “Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts”, in Proceedings of UBICOMP 2001, Atlanta, GA, USA, Sept. 2001.

[14] The Microsoft Mobile Developer Conference, MDC 2002, London Hilton Metropole, available via the Internet at: http://www.microsoft.com/europe/mobdevcon/, April 2002.

[15] S. Björk, J. Falk, R. Hansson, and P. Ljungstrand, “Pirates! - Using the Physical World as a Game Board”, in Proceedings of Interact 2001, IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, July 2001.

[16] J. Falk, P. Ljungstrand, S. Björk, and R. Hansson, “Pirates: Proximity-Triggered Interaction in a Multi-Player Game”, in Extended Abstracts of Computer-Human Interaction (CHI) 2001 (Interactive Poster), ACM Press, 2001, pp. 119-120, April 2001.

[17] “Can You See Me Now?” project, available via the Internet at: http://www.canyouseemenow.co.uk/, 2001.

[18] S. Izadi, M. Fraser, S. Benford, M. Flintham, C. Greenhalgh, T. Rodden, H. Schadelbach, "Citywide: supporting interactive digital experiences across physical space", in Dunlop and Brewster, editors, Personal and Ubiquitous Computing Journal, Volume 6, 2002.

[19] “BotFighters”, homepage available via the Internet at: http://www.teliamobile.se/botfighters/.

[20] Nokia, “General Packet Radio Service - GPRS - Nokia's vision for a service platform supporting packet switched applications”, White Paper, 1998.