

Research Article

Using a Mobile Phone as a “Wii-like” Controller for Playing Games on a Large Public Display

Tamas Vajk,¹ Paul Coulton,² Will Bamford,² and Reuben Edwards²

¹ Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Goldman Gyorgy ter 3. IV.em, H-1111 Budapest, Hungary

² Informatics, Infolab21, Lancaster University, Lancaster LA1 4WA, UK

Correspondence should be addressed to Paul Coulton, p.coulton@lancaster.ac.uk

Received 26 September 2007; Accepted 12 November 2007

Recommended by Kok Wai Wong

Undoubtedly the biggest success amongst the recent games console releases has been the launch of the Nintendo Wii. This is arguably due to its most innovative attribute—the wireless controller or “Wiimote.” The Wiimote can be used as a versatile game controller, able to detect motion and rotation in three dimensions which allows for very innovative game play. Prior to the Wii, and with much less furor, Nokia launched its 5500 model phone which contains 3D motion sensors. Using the Sensor API library available for the Symbian OS, this sensor data can be used by developers to create interesting new control schemes for mobile games. Whilst 3D motion can be utilized for ondevice games, in this paper we present a novel system that connects these phones to large public game screens via Bluetooth where it becomes a game controller for a multiplayer game. We illustrate the potential of this system through a multiplayer driving game using the Microsoft XNA framework and present preliminary feedback on the user experience from a public trial which highlights that these controls can be both intuitive and fun.

Copyright © 2008 Tamas Vajk et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Traditionally, the console game industry has divided potential players into two main categories: hardcore and casual gamers [1]. Of these categories it is the hardcore gamer that the industry has targeted itself towards as they exhibit features it wishes to exploit, such as [1] their tendency to purchase and play many games, their ability to enjoy longer play sessions, their ability to tolerate high levels of functionality in the user interface, their decision to play games as a lifestyle preference or priority. The result of the focus on this market has led to a console user demographic dominated by young white males and arguably the genres of first person shooter (FPS), sports and driving games [1].

However, there have been successful attempts to broaden the console user demographic, most notably by Nintendo through its DS console and innovative titles such as *Nintendogs*, *WarioWare*, and *BrainAge*, the latter of which created huge sales amongst previously nonconsole gamers.

The success of the DS led Nintendo to develop the Wii which has achieved phenomenal sales since its launch [2] by capturing the imagination of users not traditionally consid-

ered part of the current console gaming market [2]. From the perspective of both the game developer and game player, the most innovative feature of the Wii is the “Wiimote.” The Wiimote itself was one of the primary design aspects of the Wii, and in an interview by Kenji Hall for *Business Week* in November 2006, Shigeru Miyamoto describes part of Nintendo’s rationale:

“The classic controller was something we had become fond of and gamers had become comfortable with. It had many important elements. But it also had come to dictate a lot of what went into games—the way graphics were made, the way battles were fought in role-playing games, the arc of in-game stories. They were all being made to fit one standard. Creativity was being stifled, and the range of games was narrowing.”

Their solution was the Wiimote, a wireless controller which is able to sense both rotational orientation and translational acceleration along three-dimensional axes. It achieves this through the use of inbuilt accelerometers, together with a light sensor. This light sensor is used in conjunction with an array of light-emitting diodes centrally positioned above or below the console’s display [3] which allows for six degrees of freedom. The Wiimote can be augmented with additional

features, one of which is the “Nunchuk,” which features an accelerometer and a traditional analog joystick with two trigger buttons [3]. The overall result is a game interface capable of supporting a huge array of input possibilities that will enable new and exciting video game experiences.

Whilst mobile phones are in their relative infancy as gaming devices [4] compared to 7th generation consoles such as the Wii, their ubiquity and increasingly rich feature sets means they are equally capable of addressing the issue of widening the game player demographic [5]. Furthermore, as ubiquitous consumer devices they are ideal for interacting with urban computing environments and in particular large public displays [6]. Further, with the ever expanding feature set on handsets, mobile game developers can increasingly turn to innovative forms of user input such as RFID/NFC, cameras, microphone, and so forth [7].

Whilst some of the aforementioned input mechanisms have been the subject of a variety of research projects [7], the use of 2D accelerometers on mobile devices has been the subject of few studies [8] and the potential for using 3D accelerometers on phones appears completely unexplored [7]. This is principally due to the fact that the first 3D accelerometers have only just been integrated into mobile phones and these phones have yet to become widespread. The Nokia 5500 was one of the first such equipped phones having been targeted at sports users, utilizing the built-in motion sensors as a pedometer and speed/distance tracker for various exercising purposes. Other phones with motion sensors include Samsung’s SCH-S310 (claimed to be the world’s first 3D motion sensing mobile), NTT’s N702, and more recently the Nokia N95.

Although there is an obvious potential for innovation in game play on mobile phones themselves, in this paper we are concerned with the potential use of a phone as a controller for games (or indeed a variety of applications) running on large public displays. In the following sections, we present the generic design and implementation of such an interactive system together with issues related to the implementation on this particular phone. Furthermore, we highlight the potential for the general use of accelerometers on phones in a variety of control interfaces. To illustrate the opportunities of this interface, we highlight its operation through the development of a novel multiplayer car racing game using an analogue control mechanism. This extends upon our previous work [9] based on the old arcade classic *Tron Light Cycles* in which a digital control scheme was employed. We then present the user experience from participants playing the game at a public event in Budapest, before drawing our overall conclusions.

2. PHONE CONTROLLER SYSTEM

Whilst there are obviously possibilities for creating innovative interfaces using 3D motion as the input mechanism for mobile games, in this project we explore using the phone as the games controller for multiplayer games shown on external displays and, in particular, large public screens. Using a large screen offers a number of benefits: it frees the games developer from constraints of the limited graphics capabil-



FIGURE 1: Poppet system.

ities of the mobile screen [10], enables a greater amount of movement to the participating players, provides a rich social atmosphere [6] and affords an opportunity for rich social interaction [11] in a variety of urban landscapes. However, we do acknowledge the practical deployment of such displays is often fraught with difficulty [12].

The system developed is part of a generic framework, which we have termed Poppet,¹ for utilizing on-board phone sensors such as cameras, accelerometers, radio frequency identification/near field communications (RFID)/(NFC), which can be linked to games running on large public displays via Bluetooth as shown in Figure 1.

Although Poppet is capable of addressing a range of devices, in the next section, we specifically describe the design challenges involved in producing a mobile client for the Nokia 5500 together with its on-board accelerometers. However, the game server design is sufficiently generic to allow interaction with a wide range of phones using Bluetooth as the means of communication.

2.1. Mobile client design

Accessing the 3D motion sensors requires the use of the Symbian Sensor API which is similar in function to J2ME’s Mobile Sensor API (JSR-256). Both of these APIs provide the potential to access a wide range of sensors such as accelerometers, thermometers, barometers, and humidity monitors, in fact any type of sensor designed to be incorporated in a mobile phone, or those accessible via Bluetooth [13]. Sensors need only be supported by the API library to be usable. The Symbian Sensor API is available from Nokia and requires the use of the Symbian S60 3rd Edition SDK. Whilst there is support for the Symbian Sensor API on several mobile devices, there are currently no mobile phones that include JSR-256 [13].

¹ In folk-magic or witchcraft, a “Poppet” is a doll made to represent a person, for casting spells on that person. The intention is that whatever are the performed actions, the doll is transferred to the subject. These dolls are often incorrectly referred to as Voodoo dolls.



FIGURE 2: Nokia 5500 accelerometer axes.

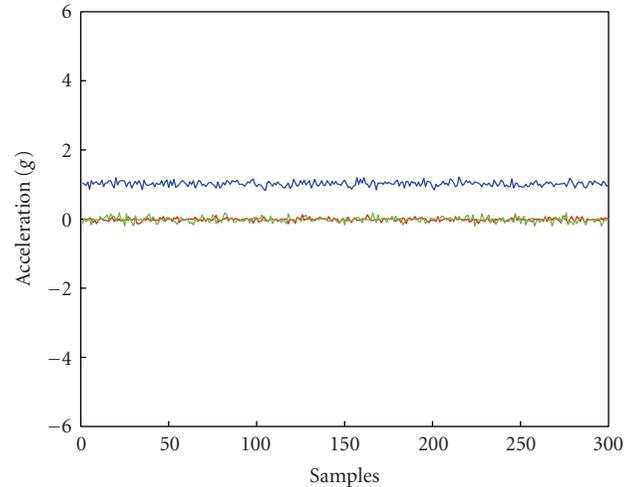
The general Poppet framework uses J2ME, as this is currently the most widely deployed mobile platform. Although the sensor data is not directly accessible from J2ME, because of the lack of JSR-256 as previously highlighted, the problem can be overcome by using a socket connection on the mobile phone to allow access to native services. The general solution for accessing native services from J2ME on Symbian S60 phones is by opening a low level socket connection in a Symbian C++ application then connecting to the defined port on the loop-back address from the J2ME application [14]. Thus the Symbian C++ application can retrieve sensor/other data from the phone and then forward this to any J2ME applications that may be listening. This solution has been applied in our simple mobile client to allow J2ME applications to access the 3D sensor data.

The connection between the game client and server is based on Bluetooth, which creates a reasonably high bandwidth (data rates can vary between 1 Mbps and a few Kbps depending upon the type of transfer mode initiated) between the devices. In order to allow a device to become discoverable by others, it is necessary to advertise at least one Bluetooth service. In the case of Poppet, the mobile client implementation uses the official Java Bluetooth API (JSR-82) to alleviate porting issues.

The Nokia 5500 utilizes a 6g accelerometer, that is, it can detect acceleration forces with a magnitude of up to six times that of earth's gravity. The accelerometer outputs three 12-bit signed data values at a frequency of around 37 Hz. These outputs correspond to the three phone axes (x , y , z) as shown in Figure 2.

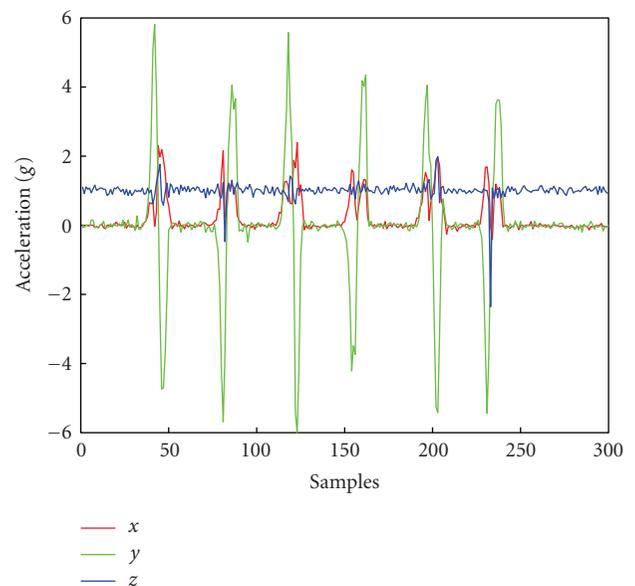
In terms of illustrating the opportunities for creating novel interaction methods using phones with inbuilt accelerometers, it is worth considering the following three Figures (3, 4, 5) which show the recorded output from the sensors on the Nokia 5500 in three different scenarios. Note that output has been expressed as acceleration forces in g for clarity.

Figure 3 shows the three accelerometer outputs when the phone was statically placed upon a desk (representing a horizontal plane). It can be seen that outputs x and y are approximately zero (although some sensor noise is evident) and



— x
— y
— z

FIGURE 3: Nokia 5500 accelerometer data (phone at rest on a table).



— x
— y
— z

FIGURE 4: Nokia 5500 accelerometer data (lateral movement of phone across the table).

output z is showing a positive 1g force which is the effect of gravity on the device. Note, although sensor noise could be reduced by the application of a digital filter on the output values, in this case we felt showing the raw data was more informative. Overall, this figure illustrates that gravity provides a means of deducing the orientation of the phone which can then be utilized to provide a “tilt-based” controller.

In Figure 4, we see the accelerometer outputs resulting from several rapid lateral movements of the phone on a desk. At the start of the graph, we see outputs in the same state as in Figure 3 and then very large acceleration forces on axis y produced by the movement. Note that the output rapidly

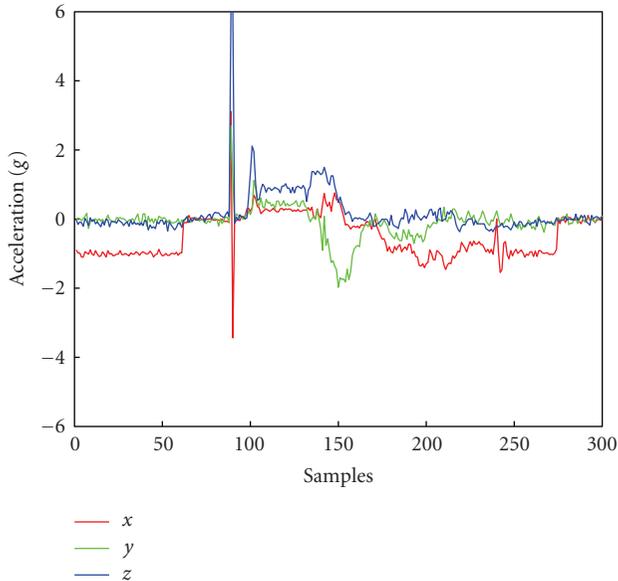


FIGURE 5: Nokia 5500 accelerometer data (phone dropped).

alternates between positive, then negative data values indicating the transitions from acceleration to deceleration and vice versa.

There are some smaller residual artifacts from this movement, seen in both x and z , which result from testing by hand rather than using a fixed jig (as it is difficult to isolate a movement completely under such conditions). Figure 4 depicts the potential for utilizing “hand gestures” for the control of events within games [15] and without the problems associated with using the camera to obtain the movement [15]. However, gesture recognition requires careful study as the variation in how the user holds the phone could produce anomalous outputs and there is no method of obtaining the phone’s physical position within actual space as provided by the “sensor bar” for the Wii. This is because both rotation and translational acceleration affect the accelerometer’s output (essentially they can produce the same internal forces on the accelerometer).

Figure 5 visualizes the output after the phone has been dropped. The trace shows that initially the phone is held upright with the screen held at the top with gravity acting on x . When the phone is dropped, the effect of gravity is overcome and all three accelerometers output values approximating zero, before the phone hits the floor with a jolt. The aim of this test was to show that the phone could be used to measure activity of a player within a game, such as jumping or running, whereby the phone would simply be worn rather than held and directly controlled.

2.2. Game server design

Using Bluetooth on a PC requires the implementation of a Bluetooth stack. Furthermore, different models of USB Bluetooth dongle have different stacks that may or may not be compatible with Microsoft Windows’ Bluetooth stack. To alleviate some of these compatibility issues, the game server

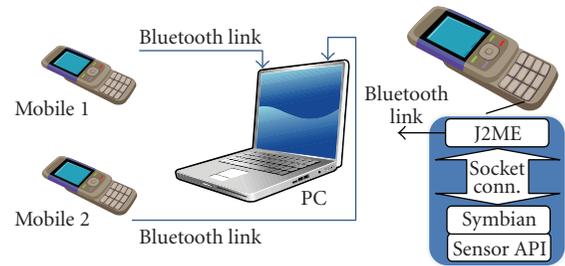


FIGURE 6: Phone game controller communication architecture.

was implemented using the Franson Bluetooth SDK for C#. This SDK supports the two most common Bluetooth stacks, the Windows stack and the WIDCOMM stack, which should ensure that the Poppet game server can be used with most Bluetooth devices. The basic communication architecture is shown in Figure 6 which also illustrates the on-phone socket communication used to transfer accelerometer data between the native application and a J2ME application.

The game server performs two main tasks: it locates/connects the Bluetooth phone controllers and it also processes the data to be used within the game. The first requires device discovery, which looks for the previously advertised service number on each phone controller in range of the PC. Once a phone controller is found, the PC tries to establish a connection with it. If connecting to the mobile is successful, both sides have all the necessary information to send and receive data between each other. The communication is based on streams at both ends.

The remaining task of the game server is to process the received data. As data triplets are arriving at the PC approximately 37 times per second which is fine in this case although, there might not be enough time to process every triplet and maintain the operation of the game in real-time. Therefore, if a real-time game is created, the developer has three options: they can drop packets, interpolate missing accelerometer samples, or assume that all samples can be processed before the next change in game state.

3. GAME IMPLEMENTATION

Even though the project presents some interesting technical challenges, it is principally aimed at developing a theoretical understanding of the user’s experience in utilizing this type of tangible interface in conjunction with a game played on a public display. This understanding will be achieved by collecting empirical data from game prototypes played by various groups. Our first prototype was the game we called *Mobi-Tron* [9] which operated with a simple digital steering control and was tested amongst staff and students at Lancaster University.

When questioned on how to improve the experience, a number of players expressed a wish that the degree of tilt of the phone should correlate to the speed of the light cycle, as the current game only allows for changes in direction but not speed [9]. Other comments suggested incorporating sounds which the current version lacks and one person suggested

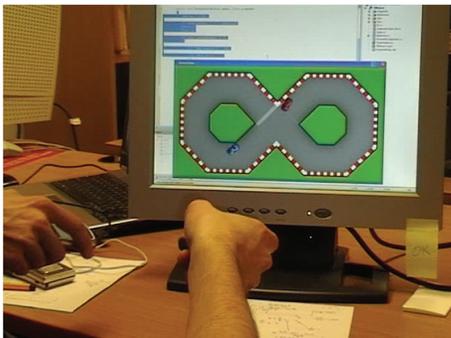


FIGURE 7: Screenshot of TiltRacer version 1.



FIGURE 9: Users playing TiltRacer.



FIGURE 8: TiltRacer version 2 on a large public display.

that rather than simply turning at right angles, the full 360 degrees of movement should be used [9]. This has been subsequently incorporated in our new game called TiltRacer which is presented in this paper and shown in Figure 7.

TiltRacer utilizes the Poppet framework previously described and the actual driving game was developed in C# using Microsoft's XNA game framework. The initial version of the game was a simple figure of eight shaped track shown in the previous figure, although this was expanded to a more complex track, shown in Figure 8, for the user trial as our initial in-house feedback considered it too easy to master. The game itself is a simple first to complete a selected number of tracks and can be played by up to 4 players simultaneously although in the trials we limited this to two.

4. USER EXPERIENCE

In this section, we present the results of testing the system amongst participants of the Forum Nokia Technical Days and European Mobile Innovation Competition² held between the 6th and 8th June 2007 in Budapest, Hungary. The game was displayed on a large screen at the evening reception for the event is shown in Figure 8 and participants were invited to try the game out. To ascertain aspects of the user experience

² The game represented the UK entry for the innovation completion, having previously won through the UK heat, and was the ultimate winner of the event.

as a whole we decided to adopt an ethnographic approach [16] because of the nature of the event which combined video and still image recording by three researchers of the 30–35 individuals who tried the game.

The general response to the interface was that it was fun, as highlighted by some of the players' facial reactions in Figure 9 and that it was also intuitive, as participants quickly gained an understanding of how their movements affected game avatars with little or no instruction from the research team or preceding players. Further, as is evidenced by Figure 9, the players were concentrating solely on the screen and not looking at the interface which aided their immersion in the experience. Whilst this observation is interesting in relation to research concerning the display of public information on a large screen against private information on a small screen it should be noted that this largely influenced by the nature of the game-play and something like a card game would provide more interesting analysis of this area.

The overall response to the game is probably best summed up by Harri Lehmuskallio, a user experience specialist for Idean who was part of the judging panel,

"The application expands the interaction from mobile to the physical world. The game concept is simple, but it demonstrates that casual games can be played this way in public spaces. The fact, that TiltRacer was created, opens new possibilities for the industry as it shows that innovating is fun and doable."

5. CONCLUSIONS

The Wii console has captured the imagination of users who previously have not participated in console-based gaming. Whilst the particular game genres chosen are a factor in user acceptance, it is undoubtedly its innovative controller functionality that has proved its most attractive feature.

Whilst we are certainly not suggesting that mobile phones could rival the Wii, the inclusion of 3D accelerometers opens up the possibility for the same type of innovation for games running on mobile phones. Further, they also provide users with an innovative game interface that they already carry around as part of their everyday day lives. This could allow them to interact in novel ways with games forming part of the environment. In particular, the increasing presence in our cities of large public displays is making this hybridization of

virtual and real space available to the mass market and thus brings new opportunities for games.

Although this game can be considered a fairly simple example in terms of graphics and complexity, and there is certainly a requirement for further study using other types of game, it does highlight that a novel interaction mechanism coupled with a fun group activity can provide an enjoyable social experience, with high levels of user interaction.

ACKNOWLEDGMENT

The authors thank Nokia for the provision of software and hardware to the Mobile Radicals research group which were used in the implementation of this project.

REFERENCES

- [1] C. Bateman and R. Boon, *21st Century Game Design*, Charles River Media, Rockland, Mass, USA, 2005.
- [2] J. Brightman, "Merrill Lynch: 30% of U.S. Households to Own Wii by 2011," <http://biz.gamedaily.com/industry/feature/?id=15309>.
- [3] Nintendo, "Nintendo Insider Technical Forums," <http://forums.nintendo.com/nintendo/>.
- [4] R. Tercek, "The First Decade of Mobile Games," Keynote at GDC Mobile, San Francisco, Calif, USA, March 2005, www.roberttercek.com.
- [5] P. Coulton, R. Edwards, W. Bamford, F. Chehimi, P. Gilbertson, and M. Rashid, *Mobile Games: Challenges and Opportunities, Advances in Computers*, Elsevier Press, Amsterdam, The Netherlands, 2007.
- [6] J. Leikas, H. Stromberg, V. Ikonen, R. Suomela, and J. Heinila, "Multi-user mobile applications and a public display: novel ways for social interaction," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '06)*, pp. 66–70, Pisa, Italy, March 2006.
- [7] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The smart phone: a ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 70–71, 2006.
- [8] J. F. Bartlett, "Rock 'n' scroll is here to stay," *IEEE Computer Graphics and Applications*, vol. 20, no. 3, pp. 40–45, 2000.
- [9] T. Vajk, W. Bamford, P. Coulton, and R. Edwards, "Using a mobile phone as a 'Wii like' controller," in *Proceedings of the 3rd International Conference on Games Research and Development*, Manchester, UK, September 2007.
- [10] P. Coulton, O. Rashid, R. Edwards, and R. Thompson, "Creating entertainment applications for cellular phones," *ACM Computers in Entertainment*, vol. 3, no. 3, 2005.
- [11] J. Reid, J. Hyams, K. Shaw, and M. Lipson, "Fancy a Schmink?: a novel networked game in a café," *ACM Computers in Entertainment*, vol. 2, no. 3, p. 11, 2004.
- [12] O. Storz, A. Friday, N. Davies, J. Finney, C. Sas, and J. G. Sheridan, "Public ubiquitous computing systems: lessons from the e-campus display deployments," *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 40–47, 2006.
- [13] P. Coulton, W. Bamford, F. Chehimi, P. Gilbertson, and O. Rashid, "Using in-built RFID/NFC, cameras, and 3D accelerometers as mobile phone sensors," in *Mobile Phone Programming: Application to Wireless Networking*, F. H. P. Fitzek and F. Reichert, Eds., pp. 381–396, Springer, New York, NY, USA, July 2007.
- [14] A. Gupta and M. de Jode, "Extending the reach of MIDlets: how MIDlets can access native services," Symbian Technical Paper, version 1.1, June 2005.
- [15] S. P. Walz, R. Ballagas, J. Borchers, et al., "Cell spell-casting: designing a locative and gesture recognition multiplayer smart-phone game for tourists," in *Proceedings of PerGames*, pp. 149–156, Dublin, Ireland, May 2006.
- [16] A. Crabtree, S. Benford, C. Greenhalgh, P. Tennent, M. Chalmers, and B. Brown, "Supporting ethnographic studies of ubiquitous computing in the wild," in *Proceedings of the Conference on Designing Interactive Systems (DIS '06)*, vol. 2006, pp. 60–69, University Park, Pa, USA, June 2006.