Interactive Techniques for Entertainment Applications Using Mobile Devices

José Luis Gutiérrez Rivas[1,*], Pedro Cano Olivares[2], and Javier Díaz Alonso[1]

[1] Department of Computer Architecture
{jlgutierrez,jdiaz}@ugr.es
http://atc.ugr.es

[2] Software Engineering Department
pcano@ugr.es
http://lsi.ugr.es

Abstract. User interaction experience has been lately the main focus of the industry of entertainment. High rendered graphics and huge computing process have been lagging behind for paving the way to the user experience interaction in which the way the user gets involved with the system is a key point. For this reason, we present along with this paper an enhancement for video games and museum applications that is able to increase user experience by separating the display from the controller side, using mobile phones as tracking devices. For this purpose, we have implemented a system architecture based on a Bluetooth peer-to-peer model that establishes a strong connection between mobile phones and desktop applications. The utilization of mobile phones has been revealed as a fundamental element in user experience due to its ease of use and its widespread adoption among society, which makes possible to enter the competitive market of entertainment.

Keywords: Virtual Reality, Bluetooth, Accelerometers, Tracking, 3D models.

1 Introduction

It is clear that mobile technologies have experienced a considerable boom over the last few years. Part of this growth comes from the appearance of new advanced computing telephones that offers a bigger processing performance: smartphones [1]. The concept of these devices comes from a melting together of different apparatus, such as computers, telephones, music players, cameras, applications (software, web services) and networks (telephone and data), people use in their daily basis.

Smartphones offer the possibility to install software, which leads to a new business idea, the development of applications and the utilization of the best fit operating system on hardware devices. There are multiple different types of applications, which are aimed to utilities, entertainment, productivity, search

* Corresponding author.

I. Rojas, G. Joya, and J. Cabestany (Eds.): IWANN 2013, Part II, LNCS 7903, pp. 334–345, 2013.

tools, social networking, etc., but it is obvious that gaming applications and those which are related to the entertainment of users are creating a very competitive place for software companies products, such as Sony, Samsung, HTC, LG, Apple, Nokia, etc.

In the meantime, the video games and consoles industry has been suffering from a stagnation derived from the lack of innovation in their products [2]. While companies were putting their efforts in improving CPU speed and GPU enhancements, they realized that users were not interested in those features as much as before. For this reason video games companies started to invest in I+D and focus both software (SW) and hardware (HW) in new directions, where user experience and interaction were key points of the product. On that basis new consoles, such as Nintendo Wii [3], and peripherals, such as Microsoft Kinect [4] and Sony Move [5], have appeared on the scene giving rise to a new era in video games field, in which the most important feature of the product resides on the user interaction with the virtual reality (VR) environment [6].

Video games companies are not the only ones that can take profit of this situation. One of this market shares belongs to applications aiming the technical support for museums. It is true that museums are culture providers and exhibitors but it is mandatory to assume that the way they reach citizens has fallen behind in terms of technology and the coverage of the population [7]. Nowadays, young people are more closely related to technology than ever since they have grown up in a world in which every task in their daily basis is accomplished or accompanied by an electronic device. For this reason a way to grant the increase of the interest of young people in museums might be based on providing them with new entertainment and interaction methods that are closer to the new technology trends.

Moreover, not only young people can take profit of these enhancements. New technologies can play a significant role in supporting elderly people and allowing them to lead high-quality lives. One of the most difficult problems in adopting technology has been that of user interfaces which are often not well-suited for elderly users, as growing old inevitably changes the physical and cognitive capabilities of humans. Touchable interfaces and the utilization of larger icons have helped to the paradigm in the context of supporting elderly users in their everyday lives [8].

This paper shows the development of applications that are focused on these new trends, new interaction techniques for video games and new business ideas for museums by the utilization of mobile devices and their capabilities in terms of connectivity and tracking. Section 3 presents the design and Section 3.1 the implementation of the architecture used for this work. Section 3.2 describes the material used for the implementation of an Android video game, and Section 3.3 shows the development and components involved in the museum application. Finally we state in Section 4 the results and conclusions extracted from the described work as well as the future and ongoing work in Section 5.

2 Interactive Techniques for Entertainment Applications

One of the key points of our development was the integration of tracking capabilities as the main source of user experience that is the order of the day for video games. Tracking technologies include different HW elements such as accelerometers, gyroscopes, gravity and proximity sensors, which make mobile devices a complete development platform with a powerful tool-chain for user experience enhancements. Furthermore, tracking features in Android applications provide a huge versatility thanks to the possibility to develop a complete and customizable user interaction interface. For example, there are already devices that provide these features (i.e. Wiimote [9]) but, in addition to the expensive costs they are exposed to, their interface and logic are restricted and closed to the original idea they were developed for; in other words, they are not customizable.

On the other hand, museums also need to maintain citizens interest that leads to an increase of new and returning visits [7]. The idea of the second application is also related to the mobile business but applied to the interaction of real elements that people can find in museums such as sculptures, figures, paintings, furniture, etc. Most parts of these museum elements are not available for real public interaction because of wear, which means that many times, users are not able to walk around a sculpture or figure in order to see its details. Add in this fact the update mechanism for the information/legend that corresponds to these elements is a tedious process in which involves printing a new plaque with the new content. This leads inevitably to an expensive update process and it is where our business idea approach has a place for museums.

Our development consists of two applications. The first application consists of a mobile-mobile video game in which the display side and the controller side are separated into two different devices. One of them used as a rendering display device, and the other one as a tracking one, exchanging data through a wireless connection.

The other application maintains the same concept, but it is targeted to mobile-desktop applications for museums. It offers the same services as a tourist guide but including the possibility to interact with the elements that exist inside rooms. The idea is to download the application whenever you are to your mobile device. Your device could display 3D models that you are able to interact with or even to obtain information from them. When the user selects a 3D model, this is displayed on a screen of the room can interact with the model and also navigate in a virtual world.

In addition to this, another key point of our study regards to the utilization of own user devices that means a reduction in the maintenance costs of hardware elements for museums since they do not need to maintain any technical support service caused by the wear of these items.

The following Section 3 presents the implementation of both applications.

3 Framework Design

Our system presents an architecture to support the design and implementation of mobile games and museum applications using two intercommunicated devices, one for tracking user moves and control and another one as a display device for rendering 3D models.

First, we proposed a Nintendo Wiimote as the controller device, nevertheless, we experimented a considerable communication delay between this controller and Android devices. Since communication delays are not permissible in video games for a correct user interaction, we consider the utilization of other devices. After some tests, we obtained an unappreciable communication delay between two Android devices so that we decided to use them as the display and the controller device respectively.

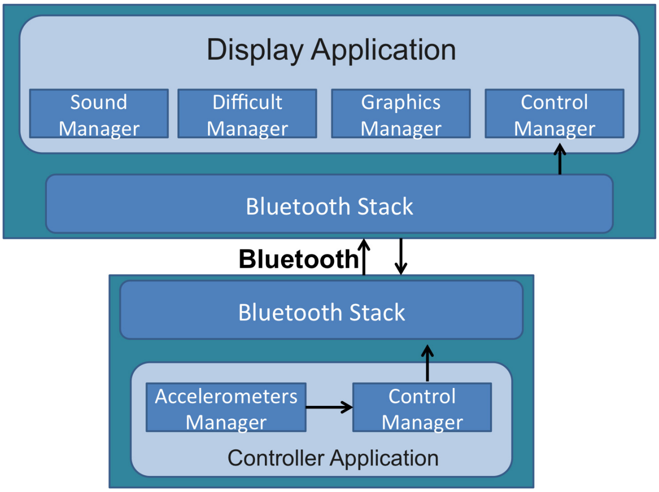

Fig. 1. System architecture model for mobile-mobile and mobile-desktop applications as controller and display devices with Bluetooth massage passing functionalities

The point of departure of this work is based on the client-server model architecture previously for desktop games using mobile devices [1]. Analogous to this architecture, our system architecture includes a communication protocol that follows the peer-to-peer (P2P) model using Bluetooth (BT) as Fig. 1 shows.

The display side application includes sound, graphical and control managers as well as a manager to handle BT events. These BT events are implemented in the BT stack which exchanges data between the display and controller applications. The same way the display application handles BT events, the controller application also includes a BT handler, with the exception of using it to send the accelerometers values and commands to the display side.

3.1 Implementation

For the architecture implementation, the platform and OS selected has been Android which offers a complete BT development framework based on Java, which allows compatibility with desktop applications. Moreover, Android offers OpenGL ES. as a 3D render engine. On the other hand, for the desktop architecture implementation we have developed the system using Java3D as the rendering system on the top of BlueCove layer, which allows communication between Android and desktop applications.

At first sight, it may initially seem that the need to use two mobile/tablet devices could represent a cost problem but actually, tablets and mobiles have become so prevalent that they are usually both present in any household. Moreover, we offer an all-in-one affordable solution that allows the customization of the device interface whilst big companies normally force users to invest in new peripherals to enhance playing experience. A clear example of this are the previously mentioned Microsoft Kinect and PlayStation Move. Tracking devices that are mandatory for playing several exclusive Microsoft and Sony games. In fact, users are not able to play them by using any other devices which represents an important barrier to the entertainment market.

Next Sections 3.2 and 3.3 present the utilization and implementation of the described development framework for a mobile-mobile game application and for a mobile-desktop museum application.

3.2 Mobile-Mobile Application

The first application scenario of this system architecture (Fig. 1) focuses the development of an Android video game application using two devices. One of the devices is used as a screen display and commander receiver while the other is used as a tracking device.

The game consists in a retro-style space ship adventure (Fig. 2) in which the user drives a space ship that must plough through the space crossing geometric figures to score points while avoiding meteors that are also flying all around the space and destroy the ship. The user has a limited number of lives that run out when a meteor impacts the space ship. This part of the video game is executed on the display application.

Regarding movements, the space ship is controlled by another application that runs on the controller application, a smartphone. It captures user motion from accelerometers and gyroscopes. The accelerometers compute the acceleration and direction in which the user has realized a movement and the gyroscope the user position respecting the coordinate axes. Every time the controller detects motion, it is captured by an event and sent to the display application using the BT communication.

In terms of user experience, the fact of separating the display and the controller functions into different devices improve visualization and usability. The problem is that by using the same device for both tasks causes a loss of vision when the display is over a certain vision angle [10]. This is translated into a

Fig. 2. Game concept of the mobile-mobile application. At the top, the display device rendering the graphic part of the game and receiving commands from the controller application through Bluetooth (at the bottom).

lack of usability and user experience. This problem has been lately identified by video game companies such as Sony, which has developed their last portable PS Vita [11] with touchable capabilities on the back side. This guarantees touchable functions even with just one screen without interfering visualization.

For this reason, we developed our video game on two devices: the controller and the display application. The following sections describes the implementation of both sides.

Controller Application Implementation. This application is responsible for sending commands and the accelerometer values to the display application. It does not require of a high performance HW since it is only focused on capturing motion and sending commands. For this reason almost any low-cost mobile device would be suitable for this purpose.

We have implemented a Java class as a global object to control the game status and store the values from the accelerometers. This class is called *RemoteDevice* and is composed of three types of attributes: accelerometer values, state of the game on controller side (playing, paused...) and the selected space ship model. Accelerometer values are capture as a coordinates vector [x,y,z] corresponding to x, y and z axes as Fig. 3 shows.

In order to ensure the capture of accelerometer values, we have created a function that captures the event that is launched when accelerometer's values change. This function is called *onSensorChanged*. After updating the accelerometer values, *bluetoothSendAccelerometers* function prepares a command buffer object composed of four bytes. The first byte is used to indicate the display

Fig. 3. Mobile-mobile game application. Left: The center image shows the 3D rendering of the display device running. The lower-right-hand corner presents the customized interface of the controller device. Right: Mobile accelerometers for [x,y,z] axes.

application the command that the user is executing on the controller application and must be executed. These command types regard to pausing, resuming and restarting the game and are interpreted later on on the display application.

The main layer is composed of several icon-like buttons as Fig. 3 shows. The game interface includes a BT button, which is used to establish the BT communication between both devices. When it is pushed, the device lists all the BT devices that are in its BT coverage. By clicking on a device, the application attempts to connects. There is also a cross-like button. It is actually composed by four buttons: top, down, left and right. These button are used in the selection screen so that the user choses different space ships. In addition to this, a confirmation button is used to confirm options such as space ship selection, and also to stop and resume the game. A try again button is used when the user runs all his lives out the game stops and shows a *Game Over* screen. Finally, an auxiliary button (Help Button) shows a help message when pressed. The help message depends on the status of the game. When there is no BT connection establish, it shows instructions to establish the connection, otherwise the message is related to the game instructions.

It is important to remark that this screen interface is totally customizable, making it possible to add or remove buttons and features when required.

Display Application Implementation. The display application runs the graphical engine of the video game and executes the commands sent by the controller. The graphic engine has been implemented with the Android version of OpenGL ES that Android NDK provides.

The same way the controller application instantiates a *remoteDevice* object, the display application has its own *remoteDevice*. Both objects have as attributes the x, y and z coordinates, game status (selecting space ship, playing and dead), the BT connection status and the chosen space ship model but in addition to this, the display application has more attributes related to the global variables of the game. This refers to lives number, score, game features, difficulty level, timers that controls the progress of the game to change the difficulty level, the OpenGL renderer and several *MediaPlayer* objects that plays different sounds of the game.

When the application is initialized, a BT connection is launched waiting for incoming connections. This BT connection has been performed as paired by using the Bluetooth object from the Android SDK. The BT connection is implemented as a listener that performs six functions: *bluetoothWrite()*, *bluetoothRead()*, *onConnecting()*, *onConnected()*, *onConnectionFailed()* and *onConnectionLost()*. These functions runs as the following state diagram (Fig. 4) shows:

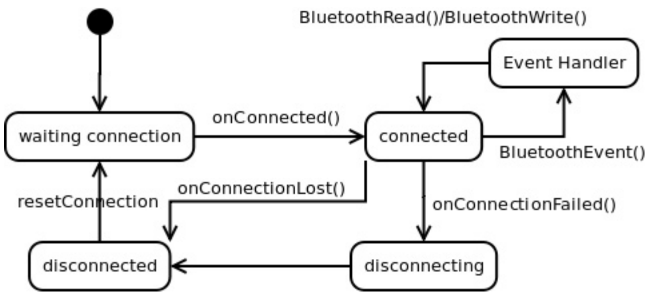

Fig. 4. State diagram of the Bluetooth connection between devices

The initial state of the display application remains waiting for incoming BT requests. When the controller application establishes a connection with the display, in both sides *onConnected()* is executed and a BT paired is performed. After pairing, the display application runs the OpenGL render and the game starts by driving the user into the selection screen. Among *Connected* state, both applications are able to exchange BT packages by using *bluetoothWrite()* and *bluetoothRead()* when the BT event handler is triggered.

When the user exits the controller application, the Android *OnDestroy()* function evokes the unpairing between both devices. The display side receives this request and performs *onConnectionLost()* that resets the BT connection. After the BT reboot, the display application is ready again to receive incoming BT requests. *onConnectionFailed()* works in the same way as *onConnectionLost()* except that it is executed after expiring a BT idle timer.

3.3 Museum Application

The museum application consists also in two applications. The first application is a desktop application running Java3D and displaying 3D models. The other one is an interactive application to control the displayed 3D models and virtual scenario. On this occasion, however, both of them are implemented using different architecture devices. The display application has been implemented in Java and runs on a desktop computer and the controller application has been implemented for Android devices. Android mobiles allow users to interact with the virtual environment by using their own devices. This makes possible an immediate availability of the interaction resources that are available at the museum, otherwise user would have to wait for their availability.

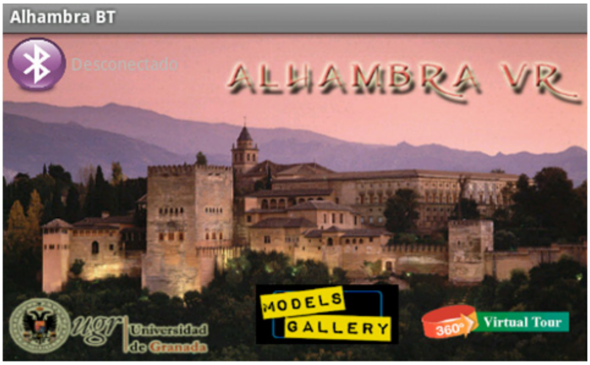

Fig. 5. Museum controller application. User side application that allows the interaction with a list of 3D models on the desktop application. Buttons available: Bluetooth connection, 3D models gallery and Virtual Tour.

The main concept of the application is based on a controller application running on an Android device to interact with the elements of a museum in a 3D virtual scenario. The museum provides a screen that is connected to a desktop computer, which is running the display application. This application is playing a custom video while it is waiting for incoming BT connections. The BT connection has been implemented using the Java BlueCove library. When the connection is established, the video stops and the application starts to show the 3D models that users can interact with.

For the controller application, the museum offers a QR code [12] that is available for visitors to download the application. Then the user runs the application (Fig. 5), its interface is displayed to the user showing all the interacting possibilities with the 3D models. The controller interface is exposed in Fig. 5.

The interface offers to the user two main options apart from the BT connector: a model gallery for model interaction and a virtual tour. The model gallery shows up a list of the available 3D models than can be displayed and then interacted by the user on the screen of the display application (Fig. 6). This interaction includes zooming, color change, rotation, translation and the possibility of showing information of the models on the controller device.

Moreover, the virtual tour offers the possibility of a third-person navigation through a virtual room like the room in which the user lies. Along with this room the objects that can be rendered are distributed all along the scene. The user is able to approximate to these objects using the device accelerometers

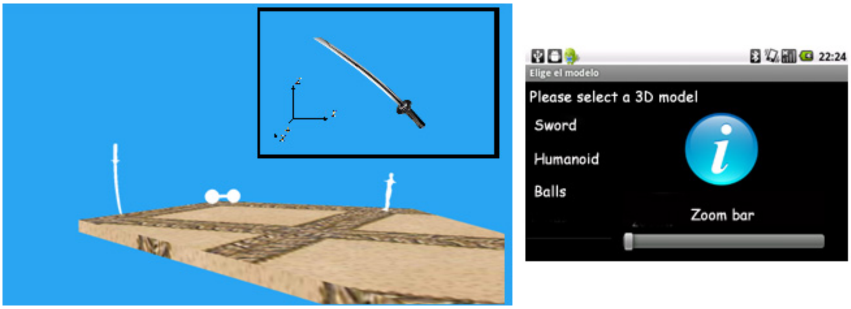

Fig. 6. Left: The center picture shows the navigation through the Virtual Tour, and the top-right-hand square shows the closer object to the user position. Right: User interface of the controller application. Shows 3D model selection menu, zooming bar and the information button.

that are sent to the desktop application the same way the previous described video game in section 3.2 describes. By this the user can rotate and translate along the room looking for different 3D models. When the user collides with a model, the Java3D render of the desktop application runs the viewer of the model and the controller application interface changes in order to provide the user the interaction possibilities previously described. When the user stops viewing a model, he/she is returned to the virtual tour.

Whenever the user exits the application, the initial demo video runs again waiting for a new incoming BT connection.

4 Conclusions

We have implemented a system architecture to use mobiles as powerful and customizable devices to control entertainment applications using a BT P2P communication model. This approach has been applied to two different examples, a mobile-mobile video game with tracking capabilities and a mobile-desktop application to interact with 3D models in a virtual environment for museums.

Regarding the mobile-mobile application, we have developed an Android Virtual Reality video game with a 3D render, OpenGL ES. The game consists in two applications that run in two different devices. One of the application is considered as a display device using OpenGL ES, and the other one has been designed to be running on an Android smartphone. We have included the possibility of exchanging tracking data and user commands between applications through a BT connection. This has created a solid communication system between both devices. By this, we made possible to transfer tracking values and commands from the controller application to the display device.

A key point of the development of the controller application is the utilization the device accelerometers and the Android Sensors API to add user tracking capabilities [9]. This application is registering the user tracking every time there is a change in its status by capturing a sensor event. Thanks to this, we have improved the user experience hence the user interacts with the game with his arm's move instead of the typical button-interface. This makes the user get immersed into a VR environment. In addition to this, using different devices for controlling and displaying the game ensures a good user interaction experience since, the touchable screen and the torsion angle derived from the tracking movement, does not interfere with the display of the game as occurs in single screen devices and consoles that joins control and display. This also lead us the opportunity to focus only on the specific tasks of each application, avoiding any device overhead and preventing the display application of any graphic lack or slow-down produced by motion/control capture.

On the other hand, the utilization of an Android device as a tracking controller has made possible the creation of a suitable interacting interface that is completely customizable. Unlikely to other HW tracking devices, such as Wiimote, Kinect, PS Move, our controller is an all-in-one device designed for our system requirements in terms of interface, software and communication protocols. This provides the system with a high level of versatility as opposed to its competitors. Moreover, we made possible the utilization of a mobile device, which is nowadays a daily basis of human beings.

In addition to this, we have developed a new business idea application for museums using a desktop computer and an Android device. This concept includes Java3D for graphic rendering and BlueCove to provide communication between devices. This application increases considerably the user experience of the traditional museum technical support. Moreover, it is a new business idea that guarantees a considerable savings in terms of hardware since the device itself is provided by the user and not by the museum that can be subject to physical wear.

5 Future Work

We have planned to enhance the graphic aspects of the video game as well as to add multiuser features to our mobile-mobile and mobile-desktop applications. Currently, both applications allow only one connection at the same time. The video game would aim to a multi-player system in which one of the devices acts as a display screen, and more than one user connect to the display device at the same time with their mobiles. Users' mobiles would be the controllers again and the game type could be a collaborative game as well as a multi-player adventure. Moreover, adding the possibility for the user to modify the controller interface in real-time would provide a new interface interaction concept to the controller application. Equally, the addition of multiuser features for the mobile-desktop application would make possible several users to interact with the desktop application and the 3D models.

In terms of technology, we are planning to integrate the Qualcomm Augmented Reality library. The idea is to provide and improve the game interaction with a higher user experience derived from the overlapping of additional virtual information to the display of the game. This would make possible to use the game in wide-open spaces, becoming the real wold the new virtual environment.

To end with, we are planning to invest in increasing our scope aiming to video surveillance and applications for the automotive domain. Our efforts are focusing in designing applications for these domains that could take the advantage of using mobile devices and their sensors and the lower price of its components since these are very competitive markets and business savings are very significant.

Acknowledgement. This work has been supported by the FP7 project TOMSY: Topology Based Motion Synthesis for Dexterous Manipulation (Grant Agreement number FP7-270436).

References

1. Malfatti, S., dos Santos, F.F., dos Santos, S.R.: Using mobile phones to control desktop multiplayer games. In: 2010 Brazilian Symposium on Games and Digital Entertainment, SBGAMES, pp. 230–238 (November 2010)
2. Dominic, A.: Video game genre, evolution and innovation. Journal for Computer Game Culture 2 (2009)
3. Wingrave, C.A., Williamson, B., Varcholik, P.D., Rose, J., Miller, A., Charbonneau, E., Bott, J., LaViola Jr., J.J.: The wiimote and beyond: Spatially convenient devices for 3d user interfaces. IEEE Computer Graphics and Applications 30, 71–85 (2010)
4. Corporation, M.: Microsoft kinect (February 2013)
5. Entertainment, S.C.: Playstation move motion capture (February 2013)
6. Burdea, G.C., Coiffet, P.: Virtual reality technology. In: Virtual Reality Technology (2003)
7. Squire, K.: Rethinking the role of educational and social science research in digital gaming. Journal of Computer Game 2 (2002)
8. Häikiö, J., Wallin, A., Isomursu, M., Ailisto, H., Matinmikko, T., Huomo, T.: Touch-based user interface for elderly users. In: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI 2007, pp. 289–296. ACM, New York (2007)
9. Vajk, T., Bamford, W., Coulton, P., Edwards, R.: Using a mobile phone as a wii-like controller for playing games on a large public display. Int. J. Comput. Games Technol. 2008, 4:1–4:6 (2008)
10. Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., Chin, J.: An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In: Dourish, P., Friday, A. (eds.) UbiComp 2006. LNCS, vol. 4206, pp. 87–104. Springer, Heidelberg (2006)
11. S. C. Entertainment, Playstation vita (February 2013)
12. D. W. Incorporated, Qr code essentials (April 2013)