# An Architecture for Using Smartphones as Interfaces for Computer Games

Thiago F. Costa, Ismael S. Silva, Glívia A. R. Barbosa, Flávio R. S. Coutinho
*Departamento de Computação*
*Centro Federal de Educação Tecnológica de Minas Gerais*
*Belo Horizonte, Brazil*
*thiagofigcosta@hotmail.com, {ismaelsantana, gliviabarbosa, fegemo}@cefetmg.br*

***Abstract*—Several issues with humans modern life, both in physical and psychological aspects, can be alleviated with the regular practice of physical activity. One way of encouraging those who are not too fond of the monotony of some exercises is by employing exergames. However, the technologies of motion detection are usually expensive and dependent on proprietary hardware from consoles. This paper proposes the development of a complex cross-platform communication architecture that links an app which transforms a mobile phone into a motion detection game controller for computer games. After an evaluation of the technologies that could be used in the implementation of the system, we came up with a proposal to make the best use of smartphones as motion detectors. A simple prototype was implemented and the results showed its effectiveness, efficiency and scalability in the exchange of messages between the various components in the system. Such proof of concept architectural model enables interesting uses of the smartphones sensors to control games in ways that are currently available only on home consoles like Xbox, Playstation and Wii/Switch. By bringing the possibility of playing exergames using more generally available and open hardware (smartphones and personal computers), the benefits provided by the use of exergames become more accessible and can be used in situations like teaching physical education in schools, helping with the treatment of diseases and rehabilitation, increasing the quality of life, and encouraging people to ingress on regular physical activities.**

***Keywords*-player-game interface; system architecture; mobile controller;**

## I. Introduction

More interactive games usually need to be accompanied by gadgets that are proprietary and expensive, which hinders its reach. Some controllers have features like force touch, second screen, vibration, gyroscope, accelerometer and other sensors, which can make games more interactive, tracking player movements [1], and allowing the use of exergames. Meanwhile computers, which are widely used for games, usually have only its standard input devices, such as a keyboard, a mouse, and sometimes a conventional joystick with no extra sensors.

More than 5 billion people have smartphones according to [2], meaning they are widely available. Even the simplest smartphones usually have accelerometer sensors that can detect acceleration in three dimensions and, when allied with a gyroscope it can greatly increase the accuracy of movement detection.

This article proposes to combine two popular and already existing technologies, smartphones and personal computers, by creating a communication architecture between them. It proposes three core applications: (a) a game controller app run on smartphones, which is basically an application to leverage its sensors, detecting movements and wirelessly sending control packets corresponding to the movements; (b) a second program receives the packets sent by the game controllers, handles them and then sends packets via data stream over process to (c) the third component, the game which is being displayed on a screen and controlled by the players' smartphones.

After surveying the possibilities of use for the technology we focused on exergames, but without ruling out other possibilities that have not been addressed in the text. Exergames are electronic games which use motion capture to detect the effort from physical activities as input into digital games. They are often applied in schools for physical education as detailed by [3], yielding benefits to youths' health and providing social and academic benefits such as those mentioned in [4] and [5].

The remainder of this work is organized as follows: Section II describes previous works which proposed a similar architecture but using different technology; Section III cites consoles that have controls with sensors and extra features, and how these features are exploited, in order to trace the purpose and form of this work; Section IV links the features mentioned above with the smartphone determining the corners of this adaptation; Section V where the final form of the project are presented; Section VI explains the need and possibilities acquired due to the use of this software, especially in the concept of generalization, so that any existing game can take ownership of this idea; Section VII the conclusion of the work and the many possibilities of improvements and products over this architecture;

## II. Related Works

The work published by [6] was inceptive to the idea developed in this paper, as it presents a motion detection architecture using older mobile phones, based on the principles from the Nintendo Wii Remote. Their aim was to provide motion capture hand-held controller which had sensors so the movement information was sent to a game being played

in a computer. A race game was developed to be played with Nokia phones by leveraging the features of accelerometer and bluetooth. Although even the author considers that what was developed was very simple in terms of graphics and complexity, it was a pioneering study that inspired other works like this.

Later, the authors of [7] proposed the use of mobile phones to establish a connection with a computer to control their multiplayer games on it using older mobile devices, which supported the Java Micro Edition platform. The results were very promising at the time, as it was possible to use in several devices (not only Nokia's), corroborating the value of the idea.

The main difference of this work from its previous is the focus on exergames and the technology used to achieve the objective, as this paper proposes an architecture using Android smartphones and wireless communication through WiFi, while the others had narrower reach (either Nokia phones or Java Micro Edition-enabled ones) and used Bluetooth for their communication protocol.

## III. Popular controllers with motion detection

One of the first popular and successful controllers was the Wii Remote, the Nintendo controller that was responsible for the popularity increase of motion detection controllers, through the use of accelerometers, gyroscope, buttons and a speaker. This controller allowed people to play games using their motion and the showcase game of the console was Wii Sports [8], which featured several sports like boxing, bowling, tennis, baseball and golf. Such games were played by performing the typical movements of each sport while holding the Wii Remote. In response to the motion controlled gaming experience enabled by the Nintendo Wii, Sony introduced PlayStation Move, which was similar to the Wii Remote but more precise, and it used Computer Vision techniques for motion detection. In a similar vein, Microsoft created Kinect, which used infrared sensors to detect movements, dispensing the use of a controller in hand.

Other interesting functionality came out with newer controllers like a second screen on tablet-like controllers, expanding the game design horizon.

In the personal computer field some controllers have been proposed like Leap Motion [9], which is a hardware that detects very sensitive hand motion. It is a sensor that must be positioned on a table pointing upwards, thus detecting individually the fingers of the hand and their positions. To better take advantage of its features there is a game store and applications with sensor support. The problem with such technology is that it has not been widely spread yet.

## IV. Smartphones as interfaces for computer games

Considering the proprietary ownership of the aforementioned controllers, we propose in this paper an architecture that leverages the popularity of smartphones. After all, because of its inherent portability, such devices have many interesting features that can be availed like motion sensors, touchscreen, speakers and WiFi board. The union of those resources enable the use of smartphones as a controller to play games on computers.

We propose the development of a mobile app to detect movements and to allow players to interact with the games. The main idea is to develop games exclusively for the platform, so they could better use the available resources. But there is also the possibility of creating either a middleware or a software adapter to allow existing computer games to be controlled (at least partially) by the motion detection performed by the mobile app.

In any case, to expose an interface for interaction with the platform through software, an application programming interface (API) was created in order to be included in game projects by the developers.

In addition to taking advantage of motion detection, games can also use other features such as touch buttons as input, show info and scenes on the screen, play sounds on the speakers and vibrate.

Most smartphones have accelerometers that are sufficient by themselves to detect movements. But to achieve better accuracy it is necessary to use a sensor not so frequently present in older or low-end smarphones: the gyroscope. While the accelerometers identify the acceleration of the device on each of the three axes, X, Y, and Z, the gyroscope identifies the rotation of the devices on those axes. The best approach is to use the accelerometer for detection and if the smartphone contains a gyroscope it could improve the accuracy in the motion detection algorithm. In the case of the most common operating system among smartphones, Android, there are also two more common sensors called StepDetector, which detects a step as soon as it is "completed", and StepCounter, which detects steps with greater precision while trading off a higher latency.

## V. System Architecture

The proposed system architecture is comprised of three main parts: (a) the controller app, run in smartphones, (b) computer games and (c) the mediator software, which executes on a personal computer and controls the communication between the controllers and a running game. We developed a prototype of such parts in Java, as it is multiplatform and can also be used to create Android applications. An overview of the architecture can be seen on Figure 1.

The main function of the controller application is to collect data from the movement sensors of the smartphone and the screen touches and display the game's graphical user interface (GUI) or the screen sent by the mediator. However, the controller has other functions such as starting the game, signing up for a game in a waiting foom and

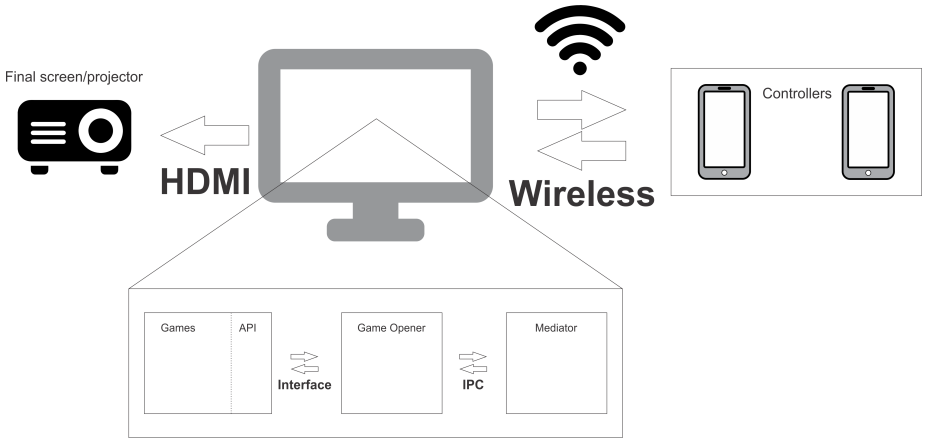


Figure 1. Architecture overview

controlling settings, a strategy similar to the communication between the client and the server used in [10]. We chose the TCP/IP stack of communication protocols due to its stability, reliability and support for all types of operating systems and hardware. The messages are sent wirelessly to a local network with the mobile phone connected through WiFi in the same network as the personal computer which is running the mediator software.

Games must implement an interface of the API which connect to the mediator. Game developers must implement this interface through the API, so that the games can communicate with the mediator to receive the movement data, players for the match and all other control messages. The API is a package of patterns, methods and classes which must be implemented by the game. The interfaces contains functions to be extended that handles the receiving of new players, starting a match, restarting a game, notification of movements, pressing some predefined key, among other interactions. The only implementation code contained on the API are on some classes which handle screen and sound capture, synchronism sending messages back and connection aspects to make sure that everything between mediator and the game is clear and thread-safe.

The mediator has a game state machine which synchronizes with the game to ensure that the mediator knows in which state the game currently is. It also implements a message queue that ensures that the messages sent by the mediator to the game arrive in their original order, that they are sent in a game state in which it can handle them, and if it fails to be interpreted correctly, it tries to send the same message again. One of the responsibilities of the mediator is to open and manage a game, which was implemented in a piece of software called “game opener”: a mediator module responsible for opening the games in a new process, different from the one of the mediator itself. That was necessary to prevent failures that could happen to the game not to affect the mediator execution health, as if they were on the same process, an unhandled exception in one part could cease the execution of the other.

An application-level communication protocol was created because of the various processes involved in the gameplay and it works through a data stream. The mediator opens the game opener by calling it in a virtual shell that returns its new process giving to it the game executable path as parameter, so the game opener opens the game in a thread on the same process by locating and running its main class inside the code. As the process of the game and the opener is the same, they share the same system input and output stream, so the mediator catches its standard input and output and when it needs to deliver a message between the processes the mediator prints on the opener standard input. On the game opener side, there is a thread which is always listening to the standard system output, and another thread which is always asking for the game state. The messages sent are strings which have to be parsed. When the mediator needs to call an API function inside the game it prints on the game opener process the name of the function and its parameters and the game opener parses and executes it.

## VI. Mediator

To link the mobile controller and the game, a mediator application was developed. By analogy with videogames, the mediator is the console and the smartphone is the joystick. It is certain that smartphones are very popular, but considering everybody’s mobile phones we can see that they have a huge discrepancy in terms of resources, sensors and processing power, given their fast evolution rate. So it is important for the smartphone application to be optimal and lightweight, as it has to pick up and process the information of the sensors to detect movements and it also has to exchange network messages constantly. So the mediator links inputs on the smartphone with a command of the game.

The side of the application which handles the communication with the game is very important because without that there is no game to be played, because will not have a way to control it. The mediator needs to be aware of every game state to know how to handle it, for example, if the game is on the main menu the mediator can not send commands to control a character, it needs to send commands to select the game and phase. Hence, the mediator must keep a state machine that needs to be synchronized every time with the game by asking it about its condition or recognizing it using key elements of the game.

## VII. Final Remarks and Future Work

The proposed architecture aims to be widespread since it takes advantage of already popular technologies, computers and smartphones, to simulate a popular solution already in the consoles, which are games that leverage motion detection, however with high cost attributed. A proof of concept prototype of this architecture was successfully developed by splitting the system in three parts: the controller located in the smartphone that takes care of the inputs and outputs, the mediator in the computer that takes care of the connection between the game and the control, and the game itself.

The implementation of the project still needs to be evaluated regarding its efficiency and optimality, such as: who will process the data to detect the movements - the smartphone or the computer? Will motion detection require the help of extra sensors when available? In addition, we still have to evaluate the efficiency of the platform in the case of real exergames, which comprises the main motivation of this proposal.

The developed prototype was successful in maintaining the active connection with multiple players and it was possible to play two games developed on the API using an emulated control.

This architecture, when fully developed, can be very useful in lots of projects, for example a community library with the adaptions of the game for the platform shared by their own users. The project can be adapted to be used at schools to teach physical education as previously mentioned, and the medicinal uses on physiotherapy and rehabilitation as in [11] and [12]. And like [13], it is possible to implement the system on a gym to motivate people to take better care of life while they play games.

The development of the project continues through the mentioned things that needs to be implemented in future work like the motion capture system, video and audio streaming, connection speed, GUI rendering on the mobile, capture screen touches, send vibration to the phone etc. And everything else can be evaluated to make sure there is no room for improvement and, if there is, it should be improved. Very new extensions can be added as well, for example an extension of the connection between the computer and the mobile, extending over the Internet instead of intranet, expanding the platform's reach.

## Acknowledgment

The authors would like to thank the institution CEFET-MG for the financial support of the project and all the students and teachers involved who could make it happen.

## References

[1] M. B. Del Rosario, S. J. Redmond, and N. H. Lovell, "Tracking the evolution of smartphone sensing for monitoring human movement," *Sensors*, vol. 15, no. 8, pp. 18 901–18 933, 2015.

[2] L. Agrela, "5 bilhões de pessoas têm smartphones," *Exame*, 2017. [Online]. Available: https://exame.abril.com.br/tecnologia/5-bilhoes-de-pessoas-tem-smartphones/

[3] M. D. Finco and A. B. Fraga, "Rompendo fronteiras na educação física através dos videogames com interação corporal," *Motriz: Rev Educ Fís*, vol. 18, no. 3, pp. 533–41, 2012.

[4] A. E. Staiano and S. L. Calvert, "Exergames for physical education courses: Physical, social, and cognitive benefits," *Child development perspectives*, vol. 5, no. 2, pp. 93–98, 2011.

[5] J. Nitz, S. Kuys, R. Isles, and S. Fu, "Is the wii fit™ a new-generation tool for improving balance, health and well-being? a pilot study," *Climacteric*, vol. 13, no. 5, pp. 487–491, 2010.

[6] T. Vajk, P. Coulton, W. Bamford, and R. Edwards, "Using a mobile phone as a "wii-like" controller for playing games on a large public display," *International Journal of Computer Games Technology*, vol. 2008, 2008.

[7] S. M. Malfatti, F. F. Dos Santos, and S. R. Dos Santos, "Using mobile phones to control desktop multiplayer games," in *Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*. IEEE, 2010, pp. 230–238.

[8] Nintendo, "Wii Sports," 2006.

[9] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[10] M. Zamith, M. Joselli, J. Siva Junior, M. Pelegrino, E. Mendonça, and E. Clua, "Adaptcontrol: An adaptive mobile touch control for games," *SBGames*, vol. 137, p. 145, 2013.

[11] J. D. Smeddinck, M. Herrlich, and R. Malaka, "Exergames for physiotherapy and rehabilitation: a medium-term situated study of motivational aspects and impact on functional reach," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 4143–4146.

[12] G. Barry, B. Galna, and L. Rochester, "The role of exergaming in parkinson's disease rehabilitation: a systematic review of the evidence," *Journal of neuroengineering and rehabilitation*, vol. 11, no. 1, p. 33, 2014.

[13] V.-M. Nurkkala, J. Kalermo, and T. Jarvilehto, "Development of exergaming simulator for gym training, exercise testing and rehabilitation," *Journal of Communication and Computer*, vol. 11, pp. 403–411, 2014.