AdaptControl: An adaptive mobile touch control for games

Marcelo Zamith[*], Mark Joselli[†], José Silva Junior[‡], Mateus Pelegrino[‡], Evandro Mendona[‡] and Esteban Clua[‡]
[*]Federal University of Viçosa
Email: zamith.marcelo@gmail.com
[†]Escola Politécnica
Pontifícia Universidade Católica do Paraná - PUCPR
Email: mark.joselli@pucpr.br
[‡]UFF - IC - Medialab
Email: {josericardo.jr,mateuspelegrino,evandro.user}@gmail.com; esteban@ic.uff.br

***Abstract*—The immersion in games can come from different factors, but one of the most promising is the player interaction. Nowadays, a lot of devices are equipped with touch screen, such as Smart TVs, mobile phones, mobile tablets and even gamepads. The use of mobile keys in this kind of devices can lead to bore and stress feeling to the player, since there are no physical representation of the buttons and the player could touch the wrong place on the device. This work presents the AdaptControl, a new framework for building adaptiveness controls for games. The core of the proposed framework is based on reinforcement learning algorithms. These algorithms use the inputs' history and the game statistics to define new states for the virtual buttons, such as resize some buttons or reposition them, trying to achieve a better interaction for the players. In order to demonstrate its effectiveness, this work has conducted some usability tests that show promising results in the use of the AdaptControl. Additionally, these tests are responsible to evaluate how the proposed reinforcement learning algorithms performs.**

***Index Terms*—adaptive game control,mobile, game input, framework, reinforcement learn algorithms.**

I. Introduction

The immersion in the game can come from different factors, but one of the most promising is the player interaction. However, this immersion can be broken if the player input is acquired in a improper way. Due to this fact, game developers and designer have to start to look for new and a more reasonable ways for input devices. Knowing that many people avoid playing games because of the complexity of controllers devices, such as joysticks with many buttons, one recent trend is to provide news forms of users' interaction that avoids the usage of this kind of device, acting as more natural as possible. With this in mind, several devices have been developed to give a more natural interactivity between the human and the machine, such as guitars, drums, dance pads, Kinect, Wiimote and so on. The Fig. 1 illustrates some devices.

The simplicity of these devices provides people who have never played before to playing games as gamer due to the screen of mobile devices minimizes the difficulty of user interaction [1] and [2]. Additionally, some of these devices are also applied for improving health-related skills, enhancing self-esteem and self-efficacy, promoting social support, and ultimately motivating positive changes in health behaviors [3], [4].

Fig. 1. Different kinds of devices used to gather player input information for the game.

In most games, the tasks that need to be done for a real-time single player game can be divided in three general classes [5] [6]: data acquisition task, responsible for getting user commands from the various input devices, such as keyboard, mouse, joystick and mice; data processing tasks, also referred as the update stage, responsible for tasks that update the game state, like character animation, Physics simulation, Artificial Intelligence, game logic, and network data acquisition; and data presentation task, being responsible for presenting the results to the user according to its input. In games, this corresponds usually to rendering graphics and playing audio. This work concentrates its research on creating a new input framework for games.

Nowadays a lot of devices are equipped with touch screen, like Smart TVs, mobile phones and mobile tablets. Also, a lot of game controls have touch screen, such as the Wii U, PS Vita and GameBoy. This touch input requires some form of input study, since the buttons are not physically presented. This can leads to wrong inputs and may cause stress to the player. In order to minimize the problem, this work provides a framework that deals with control adaptation, performing the adaptation of controls buttons according to how the user plays the game. The use of this framework can help games to have adaptiveness controls and also help developers to have game controls prototyping before releasing the game.

Contributions: This work proposes and evaluates a framework which is able to construct and/or adapt the control of digital games on touch screen mobile dynamically. The framework is adaptable, allowing virtual keys to be proper adjusted in the device based on reinforcement learning algorithms in order to change dynamically some control attributes such as resize the button or define a new position for it. The game control and a game are implemented and evaluated using some participants with different experience using these kind of devices.

Paper summary: the remainder of this paper is organized as follows. Section II presents some related works on new form of user interaction and Section III describe the framework architecture. In Section IV presents proposed framework tests and results. Finally, in Section V, the conclusions and future topics to research are discussed.

II. Related Work

Mobile phones have specific hardware (camera, accelerometer, GPS, bluetooth and so on), lots of them different from traditional game platform, like the video games and PCs. For this reason, these devices bring us new forms of user interaction [7], and some of them will be detailed in this section.

Nowadays smartphones have reliable speech recognition, which can be used for game input. This type of voice recognition is still a processing expensive task in a mobile phone. Another form of speech recognition in mobile phones is thought the use of speech server using voiceXML. The use of voice in mobile games, and in games in general, has been widely explored. In [8] voice in mobile games is explored using a voiceXML server. Zyda et al. show the application of voice in different aspects of the game: for game command input, for chatting in a multiplayer game and for accessing exclusive content of a game.

Mobile phones equipped with accelerometer are getting popular since the release of the Wiimote game control. The use of accelerated motion on such devices is still very restricted. In mobile operation systems the accelerometer is only used for screen orientation, which is a very limited use of the accelerometer. Future mobile operation systems will probably explore the operation system interaction with the use of motion or gesture, like the work [9] show the control of a multi screen system using the wiimote motion.

The accelerated motion on mobile games can be used as orientation, such as controlling an aircraft in a simulator game, or a car in a race game, or gestures, to do like swords fights and bowling games. The use of motion as orientation is easy to be implemented in a mobile game but the use of gestures is more difficult. Most mobile games seem to use brute force when doing gesture recognition in a game. In [10] a gesture recognition framework for mobile devices is presented.

Mobile phone touch screen devices are also very common. Most devices that have such features possess few buttons, being almost all input made by touch. This way, mobile touch screen games must be designed to accept most of their input by touch. The touch in games can be used similar to mouse clicks, allowing developers to make different types of mobile games based on virtual buttons. Consequently, the screen can draw buttons and use it to simulate button input [11]. Another type of input is through touch gesture, as presented in a First Person Shooter mobile game [12].

Mobile phones have the unique characteristic, when compared to PC and game consoles: it can be used for location though the use of LBS (location based service) or GPS (global positioning system). The games that uses this unique feature allows players to know and use the others player position in the game. These types of game are commonly called LBMG (location-based mobile game) [13], [14].

Another built-in feature presented in almost phones is a camera. Camera can be used in mobile games to detect movement of the phone, by comparing the images taken by the camera, using this information as user input to the game. [15] shows the implementation of an API for the use of camera motion as the input for 2D or 3D games. Mobile phones equipped with accelerometer can present a similar game experience using low processing. Another feature that is becoming very common with camera is augmented reality, that allows the insertion of virtual objects in aligned and interacting with the real world [16].

Several researches in the field of human interaction have been developed to use the mobile devices as well as their sensors. As a result, game applications are employed to help rehabilitation or as a complementary therapy, instead of being used for fun [17] ,[18] [19] and [20]. In human interaction research field, two groups are well explored: sensors and computer vision. The former employed the input player from devices movement like Wii control [21], while the latter adopts computer vision so that it can recognize a pattern [22]. Both case are employed to input player.

Figueiredo et. al [23] developed a framework for allowing playing a guitar game without any device, based only in gesture recognition. For this, each finger is tracked using a different color, that must contrast with the background color for best results. Unfortunately, these kind of interaction does not behave so well as playing guitar in the air, without any device to interact with, is unnatural for most players.

CamSpace [24], a vision based project, can use almost any object for input during the simulation, using a computer vision algorithm. For its work, a pre-step calibration and recognition is necessary in order to configure the framework with the object that will be used for the simulation. Unfortunately, as many computer vision based approach, the player also needs

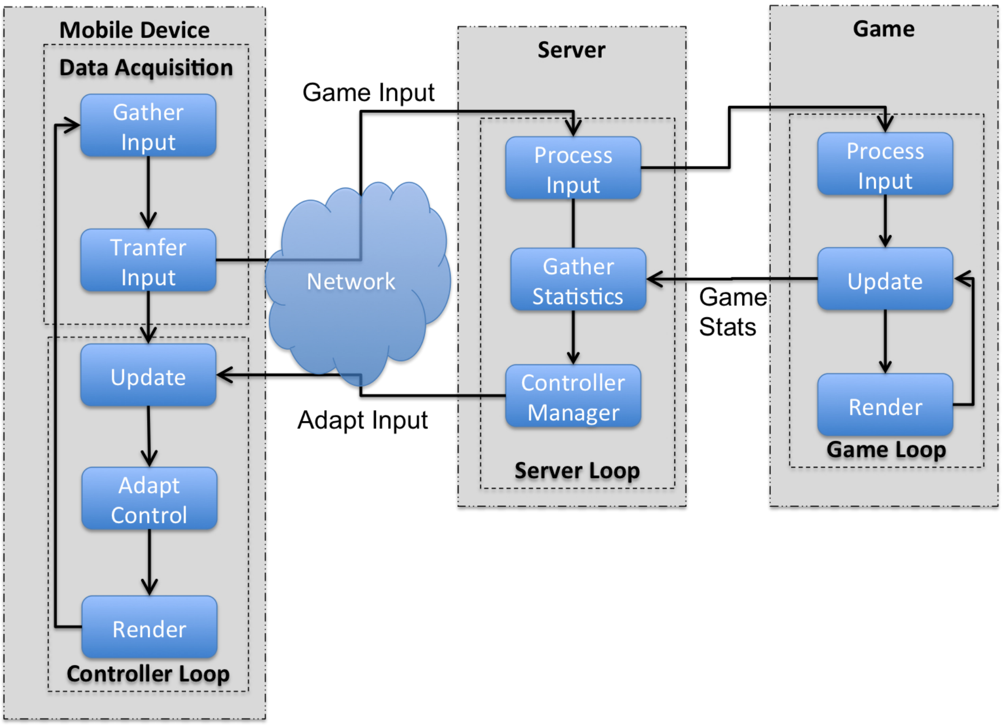

Fig. 2. Different kinds of devices used to gather player input information for the game.

to be in the focus of the camera, which could be a problem in cases where many peoples are going to play the game. In this case, the algorithm could get lost during the game or even takes much time for processing data. Another disadvantage of this approach is the the camera dependency for its proper work.

Mobile phones are already been proposed as input for simulations. The Poppet Framework [25] can be used as a simulation control that can send and receive data information to and from simulation, respectively. For this, the framework uses a bluetooth communication. Player's avatar movement is done by movement applied in mobile phone's accelerometer that is sent to a control central server.

Silvano et. al [26] propose BlueWave framework for using mobile phones as simulation input through bluetooth technology. BlueWave is implemented using Java technology, this way requiring a java enabled mobile phone. Unfortunately, the disadvantage of the proposed framework is that it uses a key's mobile phone device for control input, which can be completely different from a mobile phone to another. In this case, a layout key that could works well for a mobile phone could not work appropriately for another one, causing frustration for the player.

This work is a extension of the works [27] and [28] were a framework for building mobile game controls is presented.

III. The Proposed Framework

The framework proposed herein is based on three main components: the client, the server and the game. The client is responsible for data acquisition from the mobile devices. As the mobile devices is equipped with touch screen feature, the control is built dynamically. Each element (button) of control is viewed as a state machine, so that they can be updated according to the reinforcement learning algorithms. The server represents a layer between client and the game. This layer is responsible for synchronizing the new state of game, executing the reinforcement learning algorithms, and updating the control state. Finally, the game is connected with the server in order to update the current state and to inform its new state to the server. Fig. 2 illustrates the proposed framework.

The client is software that runs on the device used by the player to interact with the games. As proposed by this framework, it provides two connection between client and server: TCP/IP protocol and Bluetooth. In both connection is done in local network in order to maintain a data exchange rate of 30 frames per second. Otherwise, if the rate drop to lower than 30 frames per second, the game will fail due to low interactivity. Since the mobile device is able to play sound, vibration as well as render, these features are also employed by client component in order to provide feedbacks to the player.

The server component is responsible for processing the user connection requests and the data input that comes from the already connected users. It functions as a layer between the game and the game control. In this architecture, these data are received by the server and sent to the game. Moreover, the server make a synchronization among the components where the synchronize object employed is a barrier. Thus, for each frame, the client, server and the game are synchronized in order to update the new game state and the new control state. The server is able to process several connections simultaneously, which allows the user to play in a multiplayer mode.

The server also maintain statistic data from the input used and also from the game. This data can come from the client like pressed buttons and touch of the screen, or from the game, like scores, number of lives and currently level. This data is used to provide information for the reinforcement learning algorithms.

The Fig. 3 presents the sequence diagram of framework. In the first step, the server and client components initialize themselves. Following, the server enters an idle state and waits for client connection, while the client allows the player to create the control interface. After building dynamically the interface, the player selects which connection desired (TCP/IP or Bluetooth). While the user play the game, the server is recording every information about the control, each state of the buttons and the action of the player during the game.

The server uses the information previously recorded (game state, number of player mistakes in pushing any button, and so on) as input data to the reinforcement algorithms. Based on the these information, the reinforcement algorithms decides which is the new state of each element (button) of control. An example is when the player makes your avatar die due to failure in pushing jump button when the jump button is too small.

Another important feature of the framework is the player feedback. The mobile screen provides the player feedback. Considering the game dynamic, the server yields feedback to the client component, such as information to be rendered on the client (mobile device), sound played or vibrating the device. For instance, the device shows the player scores on its screen, as the Fig. 4 presents. Moreover, vibration feedback can be applied so that the player can be notified that something critical will happen in a few minutes inside the game.

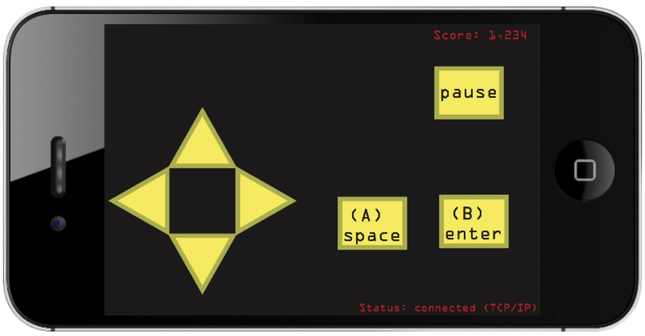

Fig. 4. The control initial state.

Furthermore, each button on interface is represented by finite-state machine (FSM). The Fig. 5 illustrates the button FSM. The initial state is current state, represented by letter C, from this state, two others one can be assumed (S and P states). S state represents the button scale, the server resizes the button on the screen, and P is the position state, where the button is reposition on the screen. S and P state are not final state, but C is. The reinforcement algorithms is responsible for changing the FSM of each button.

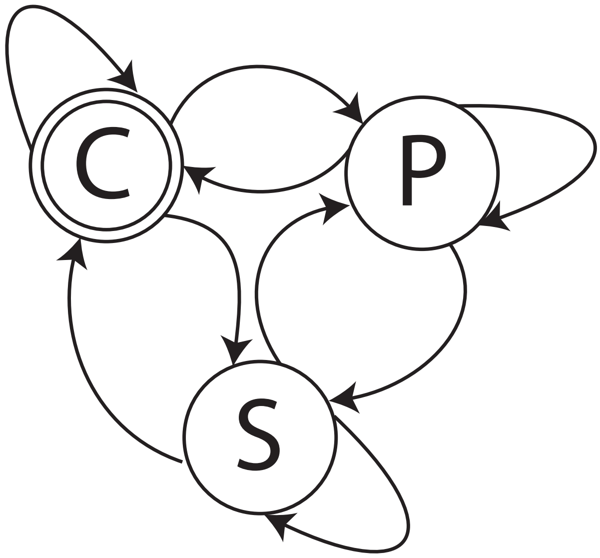

Fig. 5. Button state machine.

The Fig. 6 shows four different mobile interface configuration: the first is initial state (1). The second represents a new state where the button A is repositioned (2). In 3 is illustrated the case of increasing the size of button A; and the last state shows the state where directional button is scaled (increased its size) (4).

In the case of scaling a button, we have adopted, in our preliminary investigation, three different size multiplicators based on default size of each button: normal or default where the size has factor 1. Medium size is defined with factor 2 and the factor 3 for the big size. In relation to reposition button, we consider maximum distant from a default position of 50 pixels.

A. Adaptation Algorithms

One of the major contribution made by this paper is the possibility offered to the user about a better control layout. For example, the server together based on the game statics, does some algorithm processing, which can be based on statistics about some button that have to be pressed or have been pressed after the time or have been pressed in the wrong area. Additionally, a control could use other type of input, like the accelerometer. An example would be if the player always shake his device when trying to jump or rotating the device when playing a car race game. The algorithm could detect these patterns and allow this kind of input.

The algorithms can receive a number of statical data, divided in two groups: input data and game data. Input data can be any touch that are made in the control, by pressing the rendered keys or even input in the areas that do not have keys. Also the

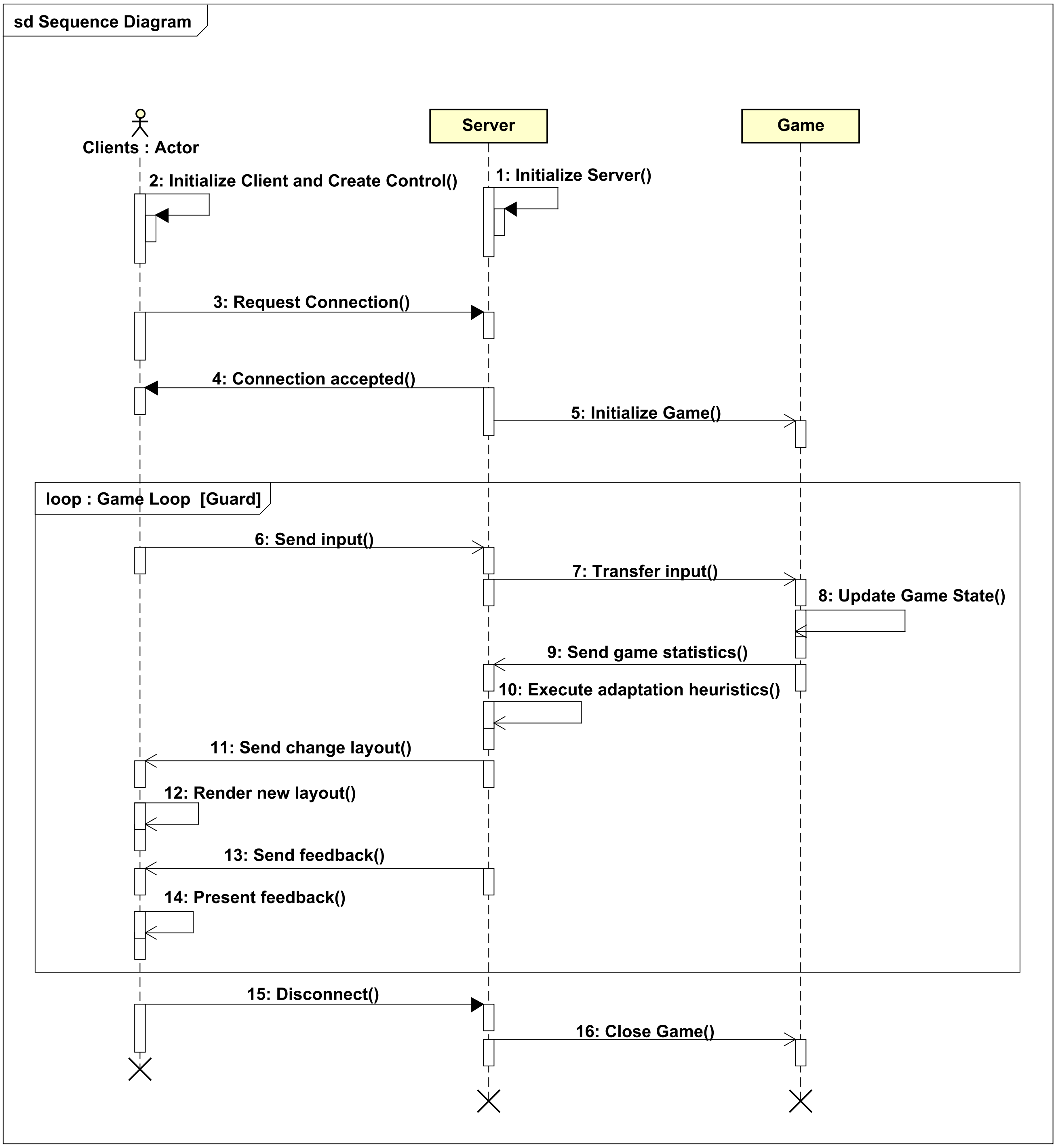

Fig. 3. Sequence of the tasks performed by the system.

accelerometer data is gathered and can be used as input for the game as well as determine new adaptive inputs. The control buttons are changed in the game according to the use. After the server has received these data, the algorithm does some analysis, which is based on statistics about the buttons pressed and errors made by pressing close to target button. According to these analysis, if any mistake is made several times, or the button is pressed more frequently than others, the game and control send this information to the server, which analyzes it and decide what is the best way to help the user. Sometimes,

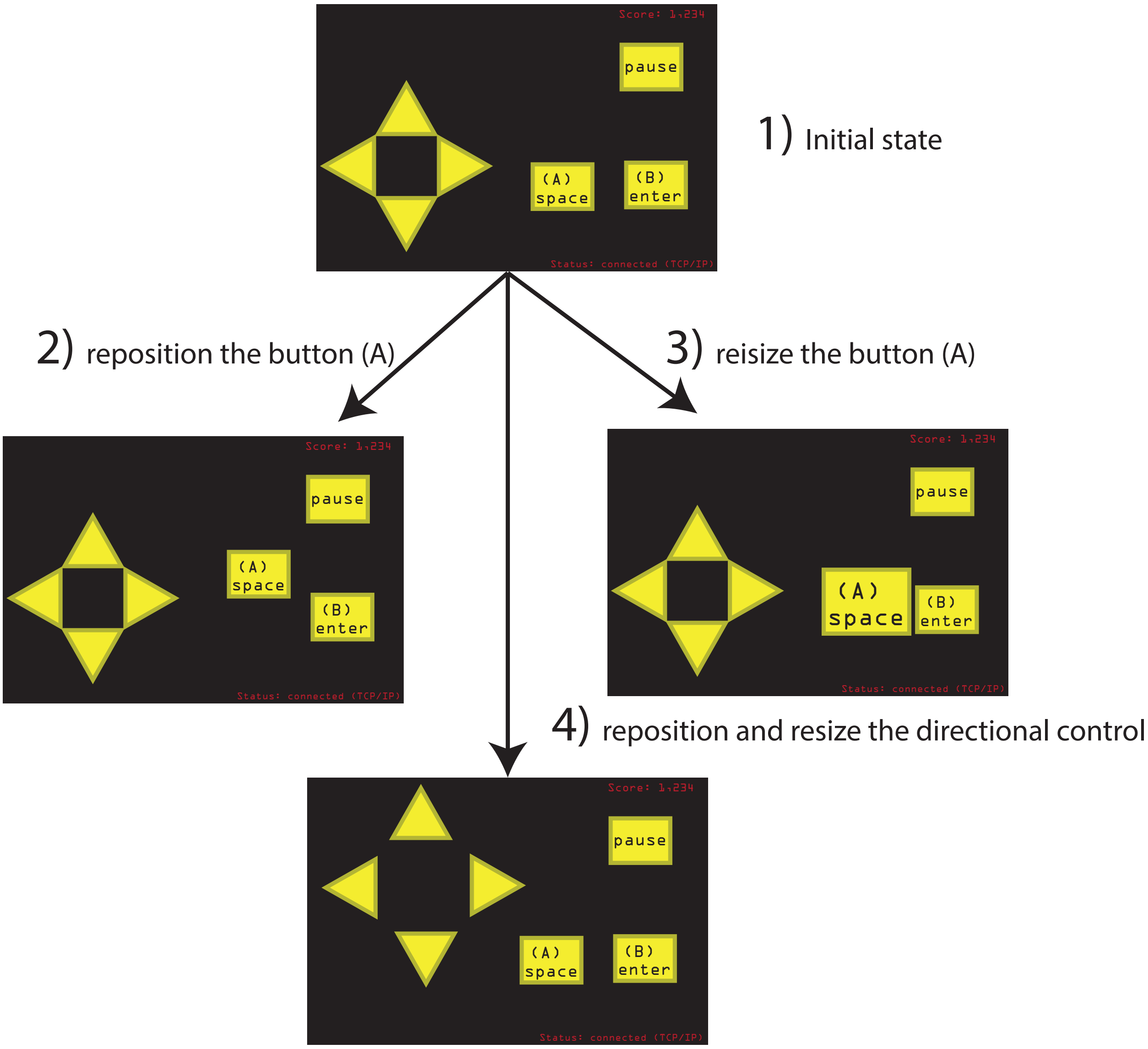

Fig. 6. States of control.

the server decides to increase the size of the button or change the position of it.

This work implement two types of algorithms, one based on the history and another based on the necessity. History adapts the control by analyzing the number of clicks, which can help the user to perform a click without a huge difficult, as the most clicked area is increased. Necessity, is the adaption of control by analyzing the mistakes that the user has made. The mistakes are divided by: ”wrong area” or ”time missed”. The ”wrong area” is committed when the user performed a click near the button, but in an area which does not have button. The ”time missed” occur when the user clicks in the button in the wrong time, such as for the case where an enemy is coming and the user has to use a shield.

IV. Tests and Results

In order to evaluate the framework properly, an application was developed for devices based on IOS SDK. The game was made using the academic framework for games JPlay [1], based on Java that grants access for all the built-in hardware resources. In the mobile application the Cocoa Touch was used to make the interface. A screenshot of the application can be seen on Fig. 7.

A game for desktop computers was made to evaluate the architecture. This game is a 2D prototype action, where the

[1] Available for free download at http://www2.ic.uff.br/jplay/index_en.html

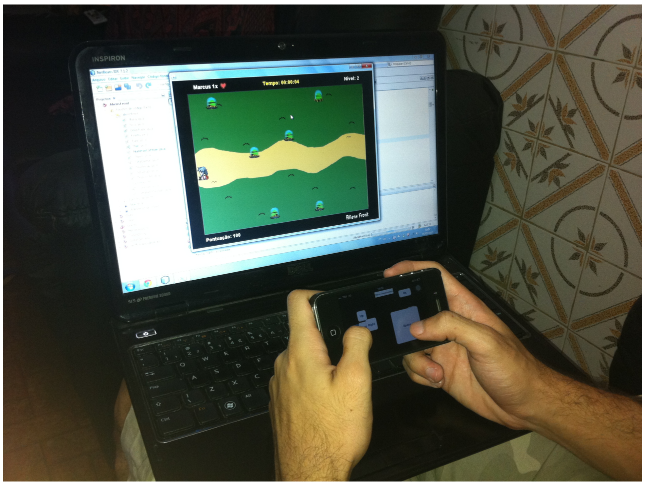

Fig. 7. Photo of the AdaptControl in action.

main enhancement is that it uses a mobile device to gather the user input and the disposition of the buttons inside the mobile can be adapted, allowing a unique new kind of interaction with a game.

The game play is very simple: the player acts as a warrior inside different planets, where he needs to destroy enemy aliens by hitting it with his sword, or defend himself using a shield. Every time the warrior make physical contact with an enemy without the use of his shield, he loses life. The objective is to kill at least the required number of enemies to achieve points to go to the next level. The game has five levels, and each level has a specific number of enemies. In the last level, the player must defeat the boss, which has life and attacks the warrior wherever he go. After killing the boss, the warrior saves the princess and the game ends.

For this game, two possible types of input methods were implemented, for emphasizing what the framework proposes. One input method allows the user to design the control and modify it, but it stays static while the game is played. And the other that the user design the control allows the AdaptControl to better design the layout with the use of adaptation algorithms.

A. *Usability Evaluation*

For the pilot, usability tests with the following characteristics were addressed to evaluate the framework and the new form of interaction:

- Ergonomic: the player tested the device and rated its ergonomic comfort level on a scale ranging from 0 (very uncomfortable) to 100 (very comfortable) of how he feels about it;
- Fun factor: the player tested the game with the different devices and rated how fun it was on a scale from 0 (very boring) to 100 (very funny) the fun he had with it;
- Player choice: the player rated each scheme overall compared to others, ranging from 0 to 100;
- Feedback: the user rated the quality of feedback he had with the device from 0 (bad) to 100 (great);
- Ease of learning: the observer rated how difficult it was to the player learn the input device from 0 (very difficult) to 100 (very easy);
- Player score: the game was scored according to the points made, and the players points were summed across all sessions with the same input device; the player rated how natural was to use, on a scale from 0 (very different) to 100 (usual).

The game was tested with each type of control ten times by a group of five different users. The pilot test group consisted of three male users (participants A, C and E) and two female user (participant B and D) with ages ranging from 12 to 29. None of the participants was physically disabled and all of them have already played console games. One participant had major experience with a touch mobile device (participant A), two had a medium experience with that mobile devices(participants C and D), one had minor experience with such mobile devices (participant B), and one had no experience with touch/accelerometer mobile phone devices (participant E). One participant is in the pre-adolescent age (participant E). All subjects were trained for five minutes by watching the observer play the game and showing them the way of interaction.

Table I presents the results of two phases of testing: the first test consisted of playing the game using the control without the AdaptControl; and the second using the AdaptControl.

In both phases, it is possible to see that players who had more experience with key touch input achieved a higher score in relation to the players who had minor experience with it. Noticeable that several players achieved more points using the AdaptControl control than using the control without adaptiveness, excluding just the participant E, which had no experience on touch inputs.

V. Conclusion and Future Works

New forms of user's input are being researched by industry in order to attract more players and grows up the immersion during the game play. Many times, people avoid playing games due complexity observed in many input devices, pushing them away from video games. Using a mobile phone as a game data input gives the opportunity to people that are resistant to these complexity input devices to try and possibly enjoy playing games using a device that they are used to use daily-by-daily, minimizing the *Time to Learn* necessary to start enjoying a game. This could, in many situations, eliminate the enter barrier found in novice game players.

In order to demonstrate this fact, a framework was developed and a simple game to be played using the AdaptControl. This adaptiveness control allows player to have adaptability of the control with his use, fact that is not possible in many others related works or even in conventional devices. Observing the results, we could confirm that users with almost any experience prefer the adaptive control.

TABLE I
USABILITY EVALUATION

Control without AdaptControl							
User	Ergonomic	Fun Factor	Feedback	Time to learn	Player Escore	Naturality	Player Choice
A	70	80	100	350	350	100	80
B	50	100	100	150	150	100	50
C	80	70	100	110	110	100	80
D	20	85	100	80	80	100	30
E	90	100	100	100	100	100	100
Control with AdaptControl							
User	Ergonomic	Fun Factor	Feedback	Time to learn	Player Escore	Naturality	Player Choice
A	100	100	100	100	450	100	100
B	100	100	100	100	180	100	100
C	100	100	100	100	300	100	100
D	80	90	100	70	250	100	100
E	100	100	100	50	100	100	100

For future topics, the authors will investigate others numerical classifiers and machine learning functions in order to better arrange the mobile screen, providing a better and comfortable interface to the player, and also minimizing the time processing. Furthermore, the authors will consider extending the test to a larger group with more variation of age, sex and the experience with mobile phones.

Additionally, the authors point the study of how to substitute the key-mouse input for different users input in traditional games and make a case study of how this change affects the gameplay and immersion during the game.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge CNPq, CAPES, FINEP and FAPERJ for the financial support of this work.

REFERENCES

[1] B. Bonifacio, H. Oliveira, and T. Conte, "Avaliacao de usabilidade de aplicacoes em dispositivos moveis," in *Proceedings of the IX Symposium on Human Factors in Computing Systems*, ser. IHC '10. Porto Alegre, Brazil, Brazil: Brazilian Computer Society, 2010, pp. 269–270. [Online]. Available: http://dl.acm.org/citation.cfm?id=1999593.1999642

[2] A. M. Mol and L. Ishitani, "Avaliacao de interface de um aplicativo para uso em telefone celular e voltado para a terceira idade," in *Proceedings of the IX Symposium on Human Factors in Computing Systems*, ser. IHC '10. Porto Alegre, Brazil, Brazil: Brazilian Computer Society, 2010, pp. 1–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1999593.1999595

[3] Z. Zhang, "Microsoft kinect sensor and its effect," *Multimedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[4] S. Ganesan and L. Anthony, "Using the kinect to encourage older adults to exercise: a prototype," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2012, pp. 2297–2302.

[5] M. Joselli, M. Zamith, E. W. G. Clua, A. Montenegro, R. C. P. Leal-Toledo, L. Valente, and B. Feijó, "An architecture with automatic load balancing and distribution for digital games," in *Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*. IEEE, 2010, pp. 59–70.

[6] M. Zamith, M. Joselli, E. W. G. Clua, A. Montenegro, R. C. P. Leal-Toledo, L. Valente, and B. Feijo, "A distributed architecture for mobile digital games based on cloud computing," in *Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on*. IEEE, 2011, pp. 79–88.

[7] M. Joselli, J. R. S. Junior, M. Zamith, E. Clua, and E. Soluri, "A content adaptation architecture for games," in *SBGames*. SBC, 2012.

[8] M. Zyda, D. Thukral, S. Jakatdar, J. Engelsma, J. Ferrans, M. Hans, L. Shi, F. Kitson, and V. Vasudevan, "Educating the next generation of mobile game developers," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 96, 92–95, 2007.

[9] J. Delgado, M. Joselli, S. Stanzani, S. M. Sadjadi, E. Clua, and H. Alvarez, "A learning and collaboration platform based on sage," in *Proceedings of the 14th Western Canadian Conference on Computing Education*. ACM, 2009, pp. 70–76.

[10] M. Joselli and E. Clua, "grmobile: A framework for touch and accelerometer gesture recognition for mobile games," in *Proceedings of the 2009 VIII Brazilian Symposium on Games and Digital Entertainment*, ser. SBGAMES '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 141–150. [Online]. Available: http://dx.doi.org/10.1109/SBGAMES.2009.24

[11] M. Joselli, J. R. da Silva Junior, , E. Clua, and E. Soluri, "Mobilewars: A mobile gpgpu game," *Lecture Notes in Computer Science*, vol. 8215, 2013.

[12] C. Wei, G. Marsden, and J. Gain, "Novel interface for first person shooting games on pdas," in *OZCHI '08: Proceedings of the 20th Australasian Conference on Computer-Human Interaction*, OZCHI. New York, NY, USA: ACM, 2008, pp. 113–121.

[13] M1ndLab, "Alien revolt: Location-based massive-multiplayer online rpg." Avalible at: `http://www.alienrevolt.com`, 2007.

[14] A. De Souza E Silva, "Hybrid reality and location-based gaming: Redefining mobility and game spaces in urban environments," *Simul. Gaming*, vol. 40, no. 3, pp. 404–424, 2009.

[15] K. N. Tran and Z. Huang, "Design and implementation of a built-in camera based user interface for mobile games," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, ser. GRAPHITE '07. New York, NY, USA: ACM, 2007, pp. 25–31. [Online]. Available: http://doi.acm.org/10.1145/1321261.1321266

[16] M. Joselli, J. S. Junior, and E. Clua, "Realidade aumentada em games com uso da unity3d," in *SBGames 2013 - Tutoriais*, São Paulo, SP, oct 2013.

[17] M. R. Golomb, B. C. McDonald, S. J. Warden, J. Yonkman, A. J. Saykin, B. Shirley, M. Huber, B. Rabin, M. AbdelBaky, M. E. Nwosu, M. Barkat-Masih, and G. C. Burdea, "In-home virtual reality videogame telerehabilitation in adolescents with

hemiplegic cerebral palsy," *Archives of Physical Medicine and Rehabilitation*, vol. 91, no. 1, pp. 1 – 8.e1, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000399930900817X

[18] J. W. Burke, M. McNeill, D. Charles, P. Morrow, J. Crosbie, and S. McDonough, "Serious games for upper limb rehabilitation following stroke," in *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications*, ser. VS-GAMES '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 103–110. [Online]. Available: http://dx.doi.org/10.1109/VS-GAMES.2009.17

[19] A. Laikari, "Exergaming - gaming for health: A bridge between real world and virtual communities," in *Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on*, may 2009, pp. 665 –668.

[20] A. Brandao, D. Trevisan, L. Brandao, B. Moreira, G. Nascimento, C. Vasconcelos, E. Clua, and P. Mourao, "Semiotic inspection of a game for children with down syndrome," in *Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*, nov. 2010, pp. 199 –210.

[21] V. Teichrieb and L. S. Figueiredo, "Interao em realidade virtual e realidade aumentada," in *Livros de minicursos do Brazilian Symposium on Virtual and Augmented Reality 2010*, 2010, pp. 9–22.

[22] B. Stenger, T. Woodley, and R. Cipolla, "A vision-based remote control," in *Computer Vision: Detection, Recognition and Reconstruction*, 2010, pp. 233–262.

[23] L. S. Figueiredo, J. a. M. X. N. Teixeira, A. S. Cavalcanti, V. Teichrieb, and J. Kelner, "An open-source framework for air guitar games," in *Proceedings of the 2009 VIII Brazilian Symposium on Games and Digital Entertainment*, ser. SBGAMES '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 74–82. [Online]. Available: http://dx.doi.org/10.1109/SBGAMES.2009.17

[24] C.-T. Tech, "Camspace from cam-trax technologies," 2011, [Online; accessed 15-July-2011]. [Online]. Available: http://www.camspace.com/

[25] 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," *Int. J. Comput. Games Technol.*, vol. 2008, pp. 4:1–4:6, January 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/539078

[26] S. M. Malfatti, F. F. dos Santos, and S. R. dos Santos, "Using mobile phones to control desktop multiplayer games," in *Proceedings of the 2010 VIII Brazilian Symposium on Games and Digital Entertainment*, ser. SBGAMES '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 74–82.

[27] J. R. da Silva Junior, M. Joselli, E. Clua, M. Pelegrino, and E. Mendonça, "An architecture for new ways of game user interaction using mobile devices," in *SBGames*. SBC, 2011.

[28] M. Joselli, J. R. SILVA JUNIOR, M. Zamith, E. Soluri, E. Mendonca, M. Pelegrino, and E. W. G. Clua, "An architecture for game interaction using mobile," in *Games Innovation Conference (IGIC), 2012 IEEE International*, August 2012, pp. 73–77.