Interaction with Platform Games Using Smartwatches and Continuous Gesture Recognition: A Case Study

Thamer Horbylon Nascimento*†, Fabrizzio Alphonsus A. M. N. Soares*, Hugo A. D. Nascimento*,
Rogerio L. Salvini*, Mateus M. Luna*, Cristhiane Gonçalves* and Eduardo F. Souza*
*Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brazil
†Instituto Federal Goiano – Campus Iporá, Iporá, Brazil
Email: thamer.nascimento@ifgoiano.edu.br, fabrizzio@inf.ufg.br, hadn@inf.ufg.br, rogeriosalvini@inf.ufg.br,
mateus.m.luna@gmail.com, cristhiane.goncalves@ufg.br and eduardodx01@gmail.com

***Abstract*—This work proposes the development of a method for smartwatches that allows to control platform games using continuous recognition of gestures and conducts a case study as the game Super Mario World. Uses a set of gestures based on geometric shapes to send actions to the game. Gesture recognition is performed by the algorithm of continuous gesture recognition, as it is able to recognize a gesture before being finalized, allows an action to be performed quickly, improving feedback. The recognition process was paralleled to improve performance. A technique has been developed that allows the execution of several gestures in sequence, without the need for a signaling that a gesture has been finalized or initiated. It was also created a technique that allows the sending of special commands to the game using the pressure applied on the screen by the player. A prototype for smartwatches was developed that communicates with an emulation platform installed on a Raspberry PI 3. A user experiment was performed as well as usability and experience tests. The results show that the method has the potential to be used effectively and effectively by players.**

***Keywords*—Smartwatch, Wearable Devices, Continuos Gesture Recognition, Games, Platform Games, Raspberry PI 3, Super Mario.**

I. Introduction

The smartwatch is a smart wearable device in the shape of a watch, which can work in conjunction with a smartphone. It facilitates the daily life of people, because everything is available in a small screen attached to the user's wrist. In addition, there are several applications to these devices and their presence in the world population is increasing [10, 11, 12].

Because they are small devices, the interaction with these devices still presents a challenge, since their intention was to facilitate the life of the user. However, there are several surveys to improve interaction with these devices [1, 10, 11, 12].

Recently, surveys are being developed on using smartphones and smartwatches to control games and to collect data from the way people are playing [1, 3, 5, 6, 7, 13].

The platform games style has been around for many years, but there are many new games and various searches in this area. However older games can be run by emulators on a variety of platforms, such as PC and Raspberry PI 3 [2, 17].

Platform games are old games, but because of the popularization of emulators, they can run on a variety of platforms, such as PC and Raspberry PI 3 [2].

The game Super Mario World was released in 1990 for the Super Nintendo (SNES) console, however, its success was great that even after several years is still a very widespread game.

Several searches are also developed to collect data from players of platform games, such as the famous game Super Mario World. In the same way, researches are developed that create Artificial Intelligence (AI) techniques to play this same game. As well as, surveys that study the levels of difficulties in these games [2, 4, 9, 15, 17, 18].

Due to the popularity of the Super Mario Word game and the increasing use of smartwatches, this work proposes the development of a technique that allows the player to control platform games from smartwatches, using the game Super Mario World as a case study.

For this it will be necessary: to recognize gestures, to relate each gesture to a command of the game of intuitive way, to send the command to the game and, finally, to execute the command in the game.

In this way, this article presents the first results of the research for interaction with platform games using continuous recognition of gestures in smartwatches.

The following sections present contents that are relevant to the understanding of the results obtained. The next section discusses the related works, in sequence will be approached the emulation platform, as well as the prototype for developed smartwatch, later details on the realization of the experiment with users. Afterwards the results will be approached and, finally, the final considerations.

II. Related Work

The work of Birke et. al (2012) [3] presents a game to be used in places with crowds of people and uses smartphones as a control mechanism.

The work of Almeida and Alves (2017) [1] presents a mobile game with 3D concepts that stimulates physical activities and predicts in real time. For this it uses accelerometer data from a mobile device to control the game.

However, the work developed by Baldauf et. (2015) [2] "Investigating On-ScreenGamepad Designs for Smartphone-Controlled Video Games" conducts a study comparing four game controllers used in smart devices. The study was conducted with the games Pac-Man and Super Mario Bros.

Kurihara's (2015) [9] research already uses serious game principles and proposes a game called Tetris 3D Modeler that allows players to design 3D models and uses the game Super Mario Bros to players raise money for charity.

Another work that also uses the game Super Mario Bros is the one developed by Sørensen et. to (2016) [15]. This work presents an evolutionary computation technique that allows the user to guide the evolution of the players in the game.

Based on the games of the Mario franchise, the work of Joselli et. al (2012) [6] presents a technique that uses mobile devices to control existing games such as Mario using features like vibration, sound and voice recognition, for example.

With the focus on transmission technology, the work of the authors Tian et. al (2016) [16] uses wirelles and wearables devices to detect body movements and use them to control games by sending them through wireless technologies.

The work of Ramcharitar and Teather (2017) [14] performs a study that evaluates the Thumb-stick, Touchpad and Gyrosensor methods for game control.

As can be seen in this section, several researches are developed to develop new methods of interaction with platform games, such as the "Mario" franchise games. It can also be observed that there are searches using mobile and wearable devices as a mechanism to control games.

However, none of these studies addressed the use of continuous gesture recognition in smartwatches as a game control mechanism.

Considering that the use of simple gestures based on geometric shapes is intuitive to the players, this work proposes to use the method of gesture recognition for smartwatches proposed in our previous work [11], as well as wireless networks for communication between the smartwatch and the game and conduct a case study with the game Super Mario World.

In addition, a technique has been developed that allows the player to perform various gestures in sequence, without the need to remove the finger from the screen or perform a specific gesture to indicate the beginning of a new gesture, these steps will be exposed in the next section. Another technique developed was to check the pressure exerted on the screen by the player so a gesture could represent more than one action in the game.

III. Emulation Platform

The Operating System (OS) used was Ubuntu running on Raspberry PI 3 Model B.

The Raspberry PI 3 Model B is a mini computer and is the size of a credit card. It has a Quad-core processor of 1.2 GHz, 1 GB RAM and has connectivity Wirelles, Bluetooth. In this way, it is possible to install emulators of several consoles. In addition, it has a low cost, its official price is $35.00 or £29.99.

It used a Super Nintendo console emulator (SNES), which is responsible for running the game. In this way, it is necessary to send commands to the emulator for it to perform the actions.

One way to interact with an emulator is to use the keyboard. Thus, a service was developed using Java technology that simulates the actions of pressing and releasing keyboard keys.

A prototype for smartwatches has been developed that communicates with the emulation platform using sockets on a wirelles local area network (WLAN). The prototype will be detailed in the next subsection.

Only the prototype sends data and the Raspberry PI 3 with the emulation platform only receives. The data sent is codes that represent actions in the game. A code is represented integer, so only one integer is sent over the WLAN, thus allowing no transmission delay.

As the service developed simulates the actions of pressing and releasing keys, it can be used in several emulators, in this way, the developed method can be used to control games of several consoles, from the oldest ones like the NES to the young ones that already have emulator like Playstation 3, Playstation 4 and Xbox One, for example.

IV. Prototype for Smartwatch

A. *Method for Interaction with Smartwatches Using Continuous Gesture Recognition*

The technique developed in our previous work allows the entry of text in smartwatches using geometric forms without the user having to draw the whole letter. The text entry should be performed using a gesture of the set of predefined gestures that is composed straight and curved [11].

Our work uses a continuous gesture recognition algorithm proposed by [8]. Using this algorithm it is possible that when performing a gesture, it is recognized before being finalized, because it is able to predict them with high precision in several different datasets before being finalized [8].

For this, it uses a technique that consists of the authors in models and segments. Thus, each gesture is a model that has a set of segments describing, in an increasing way, the partial sections of the movement [8].

It works with a vector of ordered points in relation to time, that is, a vector of ordered points related to the way the movement is to be produced. Thus, a gesture is segmented in several parts and in increasing movements. Fig. 1, exemplifies this technique.

By being able to predict partial gestures, the algorithm can recognize a gesture before it is complete, in this way, the instruction can be sent in a more agile way to the game thus improving the feedback between the user and the game.

In this way, this work will use the method of continuous recognition of gestures proposed by [8] and the method

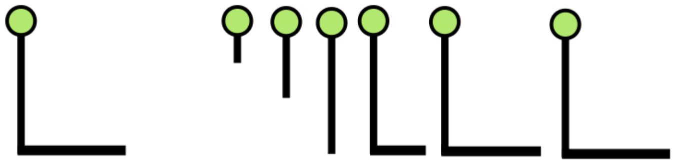

Fig. 1. Full model to the left and segments of the gesture to the right. Adapted from [8].

developed in our previous work [11], but without the purpose of entering text. It will use the technique as a basis to recognize gestures in smartwatches and relate each gesture to an action in the game and finally send and execute the action in the game.

B. Gesture Recogntion Parallelization

For the algorithm of continuous gesture recognition, each gesture represents a pattern to be recognized.

The algorithm of continuous gesture recognition calculates the probability of the gesture being performed belonging to all the patterns of its set. In this way, the time to calculate the probabilities depends on the amount of standards.

These calculations are done sequentially, that is, the probability is defined for each pattern one at a time in a loop.

As the proposed method proposes to control games using gestures, it is necessary that the recognition be fast and efficient.

Considering that the probability of a gesture belonging to a pattern does not depend on the probability of another pattern, a technique has been developed that creates a thread for each to calculate the probability of each pattern. In this way, it was possible to reduce the time needed to recognize a gesture and take better advantage of smartwatch resources.

Using the parallelization of the gesture recognition step, it was possible to reduce the recognition time by up to 4 times.

The VI section will show that the time for an action to take place in the game was an average of 80ms. However, without the parallelization the required time was on average 350ms which can detract from the player experience.

C. Prototype Developed

A smartwatch prototype has been developed that allows a player to control the game Super Mario World.

The developed prototype uses the technique for interaction with smartwatches based on gestures with geometric shapes and the algorithm of continuous recognition of gestures.

Since gesture recognition is continuous, it is not necessary for the player to finalize the gesture so that it is recognized. In this way, the prototypes were designed to send the action to the game when the player makes a gesture of at least 2cm and with the minimum recognition accuracy of 80%.

In this context, a set of gestures were created that allow the interaction of the user with the game. Thus, the player can interact with the game using simple and intuitive gestures. Fig. 2, presents the gestures and their respective actions in the game Super Mario World.

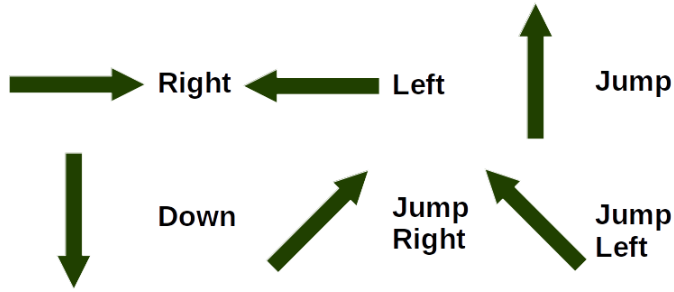

Fig. 2. Set of gestures and their respective actions in the game Super Mario World.

The arrows at the ends of the gestures shown in Fig. 2 are symbolic and represent their end. It should be emphasized that two gestures can have the same graphical representation, but they are different depending on their beginning and end, such as the gestures **right** and **left** for example.

As can be seen in Fig. 2,all the gestures represent a line, so they were generated from the reduced equation of the line, represented in the equation below:

$$y = mx + c \tag{1}$$

In this equation, x and y are the points belonging to the line and c represents the linear coefficient and m is the angular coefficient.

The prototype for smartwatch was developed for the Android Wear system and uses the set of gestures exposed in Fig. 2 for the player to send the shares to the game. Fig. 3 presents the smartwatch prototype interface running on the *Asus ZenWatch 2* and *Motorola Moto 360* smartwatches, respectively.

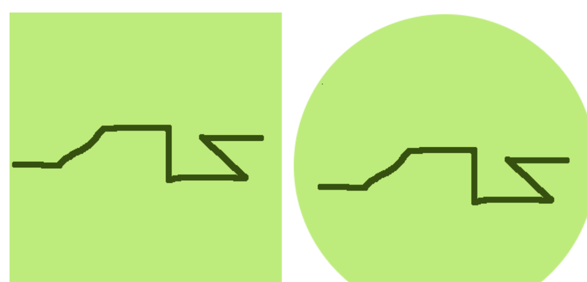

Fig. 3. Prototype interface on *Asus ZenWatch 2* and in the *Motorola Moto 360*, both running the sequence of actions: right, jump right, right, down, right, jump left and right.

Thus, when the player makes a gesture, the first step is to recognize it with the algorithm of continuous gesture recognition. In sequence, the action that it represents is identified. Each gesture has an action in the game corresponding to it, as can be observed in Fig. 1.

The action is sent by the prototype to the emulation platform on the Raspberry PI 3 using a WLAN.

The emulation platform service that simulates actions on the keyboard receives the data and sends it to the emulator running the game Super Mario World.

The steps from the performance of the gesture by the player to his execution in the game are illustrated in Fig. 4.

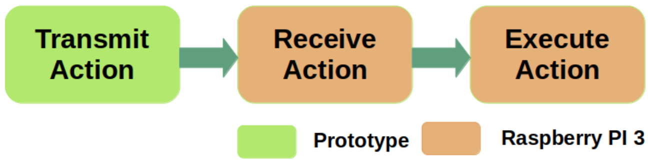

Fig. 4. Steps to send action to the game from the gesture performed by the player.

After the gesture recognition, the action is performed in the game until the user makes a new move or removes the finger from the screen. If you remove your finger from the screen, all the keys that are pressed will be released.

However, if the user starts a new gesture, all the currently-pressed keys will be released and the keys for that gesture will be pressed.

For this, it is constantly verified the gesture that the user is performing, if it is identified the beginning of a new gesture this action happens. As shown in Fig. 5.

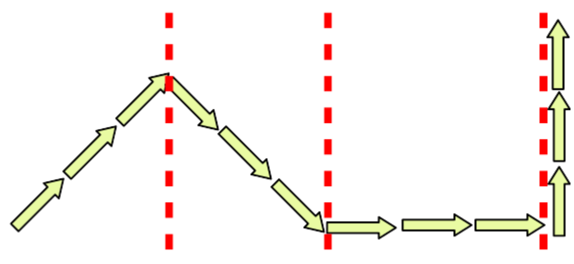

Fig. 5. Detection point of the gesture change performed by the user.

It can be seen in Fig. 5 that a gesture is represented by several arrows, each arrow represents the recognition of a gesture. After the gesture is recognized, its previous points are discarded, in this way, when recognizing a new gesture the method considers only the movements made after the last recognition. It is also possible to check in Fig. 5 a straight line dotted with red, this straight represents the beginning of a new gesture.

After the user removes the finger from the screen, the gestures performed are deleted from the screen present on the screen is deleted, this way, when the user initiates a new gesture, only it will be exposed by the interface of the prototype.

The developed prototype was used to carry out an experiment with users, explained in the next section.

D. Special Commands

There are other actions that a player can send to the Super Mario World game in addition to those shown in Fig. 2. For example, in addition to walking to the right or left, the player can run to the right and left. Therefore, a technique was used to verify the pressure applied on the screen by the player to allow the sending of the commands to run to the right or left.

As the player increases finger pressure, the size of the area he presses also increases. In this way, when the player performs a **right** or **left** move and increases the pressure on the screen, the moves will be modified for **run right** and **run left**, respectively.

Fig. 6 is the pressure exerted by the player in smarwatch screen, it can be seen from this Fig. that lower pressures are represented in yellow and higher pressures by the color red, and intermediate pressures for orange.

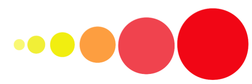

Fig. 6. Graphical representation of the pressure exerted by the player on the screen of the smartwatch.

As people's fingers have several sizes and each person can do different types of pressure on the screen. At the start of the prototype, the player performs a system calibration, where he performs gestures for **right** and **left**, as well as for actions **run right** and **run left**.

In this way, one can define the pressure difference that the player applies on the screen to perform the **right** actions and **left** for the pressure applied to the actions **run right** and **run left**.

V. User Study

The experiment was conducted with 20 participants aged between 16 and 40 years with the mean of 24 years. Of these, ten are undergraduate students, five postgraduate students and five professors, 15 participants from the computing area and the other 5 from another area. Four participants have smartwatch.

Regarding the experience with smartwatches, 5 stated they were very experienced, 8 said they had regular experience, 5 said they had little experience and the other 2 said they had no experience.

//. In order to validate the developed method the prototype was installed in the *Motorola Moto 360* and *Asus ZenWatch 2*.

Before starting the experiment, the prototype and its functionalities were exposed to the experiment participants. The set of gestures available to be used that are demonstrated in Fig. 2 was exposed to the participants and they could observe them throughout the experiment. For the experiment, the prototypes shown in Fig. 3 were used.

In the experiment, participants played the game Super Mario World for 20 minutes. Thus, the experiment provided a total of 400 minutes of experimentation with the proposed method.

The results of this experiment can be seen in the section VI, which will present the results of this research.

A. Usability and Experience Test

In order to evaluate the usability and experience of the developed prototypes, a questionnaire was applied to the participants of the experiment who answered the questions related to usability and experience to verify the efficiency and effectiveness of the proposed method and questions related to the participants' experience.

The affirmatives were answered by the participants using the Likert scale. The affirmations applied to the participants were:

1) The proposed method is intuitive.
2) It was easy to control the game in the first few minutes using the prototype.
3) After a few minutes using the prototype you felt familiar.
4) All actions are sent correctly to the game.
5) The actions are performed quickly and satisfactorily in the game.
6) It is nice to control the game with the proposed method.
7) I would use the proposed method to play in my everyday life.

To verify user experience and usability with respect to the special commands for the run right actions and **run left**, the following questionnaire was applied:

1) The actions **run right** and **run left** are easy to execute.
2) The actions **run right** and **run left** are executed with precision in the game.
3) The actions **run right** and **run left** are executed quickly and satisfactorily in the game.
4) The technique of increasing the pressure on the screen to execute the actions **run right** and **run left** is appropriate.

The next section will discuss the usability and experience tests.

VI. Results

The prototype was used for the validation of the proposed method and the accomplishment of a test of usability and experience proposed with participants. The test was designed to validate the implementation of the developed emulation platform, as well as, the prototype's functioning.

One must remember that the purpose of this work is to allow a player to control platform games from his smartwatch using gestures. And carry out a case study with the game Super Mario World.

Fig. 7 shows the responses of the experiment participants to the affirmatives of the first questionnaire, regarding the usability test and user experience with the proposed method.

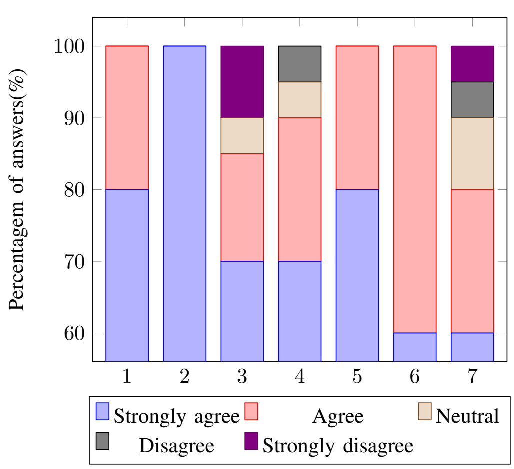

Fig. 7. Experiment participants' responses to the questionnaire applied to experience and usability.

It can be seen in item 1 of Fig. 7 that all participants stated that the method is intuitive, therefore, 80% checked the option I fully agree and 20% the option I agree.

In relation to learning the method, it can be observed in item 2 of Fig. 7 that it has easy learning, since 100% of participants marked the option totally agree.

Checking the answers of item 3, it is observed that 85% of participants stated that it is easy and quick to control the game with the developed method.

The participants' responses to item 4 of the questionnaire show that all participants stated that the actions are sent correctly to the game.

Item 5 of the questionnaire refers to the response time of the system, it can be seen in Fig. 7 that all participants in the experiment stated that the actions were sent quickly to the game. And during the tests, it was observed that the time to perform an action in the game after it was performed by the user was on average 80ms, thus allowing good experience and feedback to players.

It can also be observed that the participants stated that it is pleasant to use the proposed method and would use it in their daily lives, as shown in items 6 and 7 of Fig. 7.

It is observed in Fig. 7, that the majority of the participants marked the option totally agree or agree to all the affirmatives. With this, it is possible to verify that the method developed has potential to be used to control games.

Fig. 8 shows participants responses to their experiences with special commands for actions **run right** and **run left**.

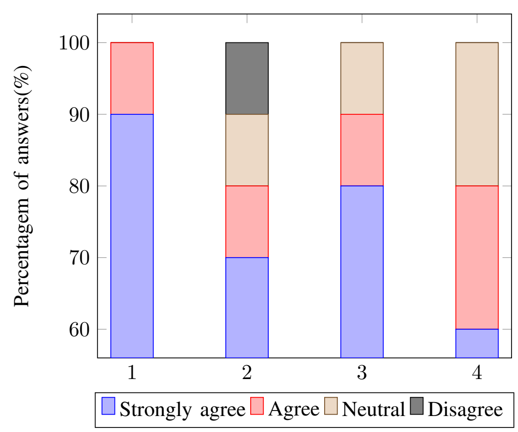

Fig. 8. Experiment participants' responses to the questionnaire applied regarding experience and usability with execution of actions run right and run left.

The responses of the experiment participants generally demonstrate that they have had good experience with the technique to perform the actions **run right** and **run left**, as can be seen in Fig. 8.

It can be seen in item 1 of Fig. 8 that all participants stated that actions **run right** and **run left** are easy to execute because 90% checked the option agreed fully and 10% the agreed option.

By observing the participants' responses to item 2, one can conclude that the actions **run right** and **run left** were accurately sent to the game.

The answers in item 3 of Fig. 8 indicate that the actions were sent to the game quickly. Already the answers to item 4 show that 80% of users stated that the technique of increasing the pressure on the screen is appropriate to execute the actions **run right** and **run left**.

In this way, it is possible to conclude that the technique of increasing the pressure on the screen to send the actions **run right** and **run left** were well accepted by the participants of the experiment, as well as the whole method.

It was possible to observe during the experiment that the participants moved the smartwatch in the same sense in which they perform the gestures.

As mentioned above, the action was executed in the game on average 80ms after the accomplishment of the gesture by the user. During the experiment, it was noted that the *Motorola Moto 360* smartwatch performed less than *Asus ZenWatch 2*, however our tests demonstrate that this was imperceptible to players.

VII. Final Considerations

This work presented first prototypes and experiments for research on interaction with platform games using continuous recognition of gestures in smartwatches.

A method was developed to provide to game players control over platform games using continuous recognition of gestures on smartwatches. Experiments were conducted withe Super Mario World as case study. Thus, we developed prototypes for smartwatches to interact to a game emulator system. As main steps were recognizing gestures, map them to a game command or action as most intuitively as possible, send these commands to emulator which finally performs it in the game.

The continuous gestures recognition algorithm performed actions very well, and it was not necessary a gesture being fully completed to be recognized, giving users freedom during game playing and providing faster feedback to the user.

We also paralleled the stage of gesture recognition, and we developed a technique that has also been developed to identify the player's gesture changes, even if he continues to draw without removing his finger from the screen. Just like, a technique that checks the pressure exerted on the screen to perform different actions with the same gesture.

The usability and experience test results shown that our developed method has potential to provide a enthusiastic experience to users and enable interaction with platform games using mobile devices and wearable efficiently and effectively. In this way, the next step of this research will be to develop a method that uses the accelerometer and the gyroscope of the devices so that it is possible for the players to control the games just by moving the device.

Acknowledgment

The authors are grateful to the Goiano Federal Institute – Campus Iporá for supporting the realization of the first author's PhD.

References

[1] Alexandre Almeida and Ana Alves. "Activity Recognition for Movement-based Interaction in Mobile Games". In: *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '17. Vienna, Austria, 2017.

[2] Matthias Baldauf et al. "Investigating On-Screen Gamepad Designs for Smartphone-Controlled Video Games". In: *ACM Trans. Multimedia Comput. Commun. Appl.* (Oct. 2015).

[3] Alexander Birke, Henrik Schoenau-Fog, and Lars Reng. "Space Bugz!: A Smartphone-controlled Crowd Game". In: *Proceeding of the 16th International Academic MindTrek Conference*. MindTrek '12. Tampere, Finland: ACM, 2012.

[4] N. C. Hou et al. "Infinite Mario Bross AI using Genetic Algorithm". In: *2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT)*. Oct. 2011.

[5] L. B. Jacob et al. "A Non-intrusive Approach for 2D Platform Game Design Analysis Based on Provenance Data Extracted from Game Streaming". In: *2014 Brazilian Symposium on Computer Games and Digital Entertainment*. Nov. 2014.

[6] M. Joselli et al. "An architecture for game interaction using mobile". In: *2012 IEEE International Games Innovation Conference*. Sept. 2012.

[7] M. Joshi et al. "Object-Oriented Representation and Hierarchical Reinforcement Learning in Infinite Mario". In: *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. Nov. 2012.

[8] P. O. Kristensson and L. C. Denby. "Continuous Recognition and Visualization of Pen Strokes and Touch-screen Gestures". In: *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*. SBIM '11. Vancouver, British Columbia, Canada, 2011.

[9] Kazutaka Kurihara. "Toolification of Games: Achieving Non-game Purposes in the Redundant Spaces of Existing Games". In: *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*. ACE '15. Iskandar, Malaysia: ACM, 2015.

[10] Huy Viet Le et al. "A Smartphone Prototype for Touch Interaction on the Whole Device Surface". In: *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '17. Vienna, Austria: ACM, 2017.

[11] T. H. Nascimento et al. "Method for Text Entry in Smartwatches Using Continuous Gesture Recognition". In: *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. July 2017. DOI: 10.1109/COMPSAC.2017.168.

[12] Ian Oakley and Doyoung Lee. "Interaction on the Edge: Offset Sensing for Small Devices". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014.

[13] Taiwoo Park et al. "Human Factors of Speed-based Exergame Controllers". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014.

[14] Adrian Ramcharitar and Robert J. Teather. "A Fitts' Law Evaluation of Video Game Controllers: Thumbstick, Touchpad and Gyrosensor". In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: ACM, 2017.

[15] Patrikk D. Sørensen, Jeppeh M. Olsen, and Sebastian Risi. "Interactive Super Mario Bros Evolution". In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. GECCO '16 Companion. Denver, Colorado, USA, 2016.

[16] J. Tian et al. "Wireless transmission system for motion sensing game controller based on low power Bluetooth technology". In: *2016 IEEE 13th International Conference on Signal Processing (ICSP)*. Nov. 2016.

[17] Rina R. Wehbe et al. "Testing Incremental Difficulty Design in Platformer Games". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA, 2017.

[18] A. Zafar and H. Mujtaba. "Identifying Catastrophic Failures in Offline Level Generation for Mario". In: *2012 10th International Conference on Frontiers of Information Technology*. Dec. 2012.