

Designing Game Controllers in a Mobile Device

Leonardo Torok, Mateus Pelegrino^(✉), Daniela Trevisan,
Anselmo Montenegro, and Esteban Clua

Computing Institute, Federal Fluminense University,
Av. Gal. Milton Tavares de Souza, s/n^o, São Domingos, Niterói,
RJ 24210-346, Brazil

{ltorok,daniela,anselmo,esteban}@ic.uff.br, mateuspelegrino@gmail.com
<http://www.ic.uff.br>

Abstract. The creative process behind new videogames always encouraged the development of innovative gameplay mechanics. However, the gamepad used to play is frequently overlooked, used only as a simple input device. This work proposes an improvement to an adaptive interface [12], using a smartphone as gamepad, with machine learning algorithms employed in real-time to tune the interface to the ergonomic needs of the current user. Now it includes an API that allows the game to change the interface elements anytime, creating a new gaming experience. Several statistics about the interaction with this interface were logged during test sections with 20 volunteers, with different levels of gaming experience. With these data, we seek to determine how to tune the interface in order to improve the experience, resulting in an iterative approach to controller design.

Keywords: Adaptive interface · Gamepad · Virtual gamepad · Touchscreen

1 Introduction

Innovation is at the heart of game development. Since the early days, where characters were represented by a few pixels, until the era of 4K gaming, more and more titles arrive in the market with groundbreaking innovations regarding level, character and sound design, gameplay mechanics and even introducing completely new game genres. However, even when a new game is revolutionary, the input device used to play it hardly is. Mouses and keyboards remained basically the same on the last few decades, only adding extra keys for shortcuts to allow quicker actions. The joystick, the most iconic interface used by consoles or PCs, also did not change that much. If one compares an original DualShock controller, launched in 1997 for the first PlayStation console, and a modern DualShock 4 [6], introduced in 2013 for the PlayStation 4, it is easy to see that they are strikingly similar. If we go as far as 1983 with the Nintendo Entertainment System, we can see that modern controllers only improved ergonomics and added more buttons [7].

Of course, it would not be accurate to claim that nobody tried to disrupt the joystick paradigm in the last decades. In 2006, Nintendo launched the Wii, a console that included a revolutionary controller: the Wiimote. Using accelerometers and infrared sensors to detect motion, the Wii allowed gamers to interact using gestures that were more intuitive and natural than simply pressing a button. The console became a resounding success, selling more than 100 million units [3]. As this trend was quickly replicated by its competitors, a new gaming audience appeared: casual gamers [3]. This market segment wanted simplified experiences, avoiding the complexities of regular games and gamepads that demanded quite a bit of dexterity from players to interact with multiple buttons simultaneously [5]. This trend continued with the rise of smartphones, making gaming more accessible than ever before.

In our previous work [12], we introduced a new type of gamepad. This new game controller is a mobile app running on a smartphone or tablet, displaying a virtual gamepad on its screen. This app communicates with a game running on a PC using sockets over a local Wi-fi network, sending all button presses to the computer. The traditional gamepad is replaced by a smartphone, while the game on the PC only has to implement a simple API to communicate with the new controller. This new gamepad is context-sensitive, showing only the buttons that the game needs in a simplified and more casual-friendly interface. It can also show GUI elements such as switches and dials, not possible on regular controllers, creating a completely new experience. This controller also includes several machine learning routines to correct and adjust its interface to the interaction patterns of a user, in real-time, avoiding interaction errors that occur on touch based interfaces. These corrections have the potential to significantly improve the experience [12], but they take some time to reach a better interface. The user has to, at least momentarily, deal with a sub-optimal UI design.

As we are dealing with a full touch interface, several data about the interaction between gamer and gamepad can be collected. While this was used before to fine tune the interface [13], this data will now be used to propose new designs for the adaptive game controller, using testing sections to determine troublesome UI decisions that will be fixed in a next version, proposing an iterative approach to controller design. These new and optimized layouts will then become the new default, potentially decreasing the time that the interface needs to adapt to a new user and solving the initial issue of playing with a sub-optimal UI. This process can then be repeated, until reaching the best layout for the gamepad.

This work will start discussing the related literature regarding touch interfaces and video game controllers. In the next section, the adaptive interface will be presented in details, showing how it can change the way we play videogames. We will then show the methodology used for our tests and the results of our testing sections, with the new proposed designs and the reasoning behind each changes.

2 Related Work

There are several works that try to address the impact that the lack of haptic feedback has on touchscreen interfaces, that normally results in more user mistakes. In this work, it will be measured as the success rate, the percentage of touches that correctly performed an action (in our case, the ones that hit any button instead of an empty screen area).

Rogers et al. [10] proposed models that treat touch input during the handover of control between user and system, trying to correct inputs when interacting with a map mobile application. Weir et al. [14] used machine learning to treat specific models of touch input, remapping incorrect touches to a different location on the screen, based on historical input data. A different approach was presented on Bi et al. [2], that improved touch accuracy using a Bayesian target selection criterion to treat input. Virtual keyboards are another important real-world application that has to deal with incorrect inputs. Solutions like key-target resizing [1] have been applied to increase the accuracy.

The approach on this work is the same used on Torok et al. [12], proposing machine learning routines to redesign the interface of a virtual gamepad in real-time, increasing the success rate. This solution will be described with more details in Sect. 3.2. When comparing this interface to a traditional gamepad, the first notable thing that it lacks is the haptic feedback that a physical button provides. Zaman and Mackenzie [15] compared a virtual gamepad with different physical controllers, showing that using a touchscreen for gaming decreases the in-game performance of users. While this application is different, since the game runs on a PC screen (avoiding the problem caused by the fingers concealing part of the screen), the lack of haptic feedback still is an issue. However, the tests performed in the previous work showed that the success rate of a simple virtual gamepad can be substantially improved using an adaptive interface with machine learning routines to improve the performance of a user.

While the lack of haptic feedback is clearly a disadvantage, this controller has a significant improvement when compared with the previous version. Now the game can configure the controller layout anytime, allowing the controller to display a simplified and casual-friendly interface that contains only the buttons that are necessary at the moment. A regular gamepad has to provide a large set of buttons since it is a generic interface used by a plethora of games, having to work with the simplest platforming game and with the most complex strategy game. When we reduce the amount of options displayed on the screen, a virtual interface on a mobile device can outperform a traditional controller due to demanding less mental effort [8]. We will discuss this aspect in Sect. 3.1

3 Adaptive Controller and Game

3.1 Game Context Adaptation

The adaptive controller, based on a full-touch interface on a mobile device, creates several new possibilities [9]. The first one is that it can display only the

buttons that are necessary in the current section of the game, while a traditional gamepad has dozens of buttons regardless of how many the current game actually needs, being more intimidating to novice players. The second improvement is the possibility to display GUI elements on the screen that are not possible on traditional controllers, like switches, dials and gesture areas. With these new elements, we can map in-game actions to a more natural interface, an approach similar to the one introduced by motion controls. With a simplified interface and less buttons, we will try to bring the simplicity of mobile gaming to PC and console games, creating a more attractive experience to a new generation of casual gamers.

As the game itself can determine every element of the controller layout, being able to radically change it anytime, this new gamepad is adapted to the game context. This means that the controller becomes part of the gameplay and of the game challenges. With this interface, the game designer now can develop the game and the controller used to play it at the same time, creating an entirely new experience. Figure 1 shows a gamer using the controller to play our game, illustrating how radically the interface can change during the game.

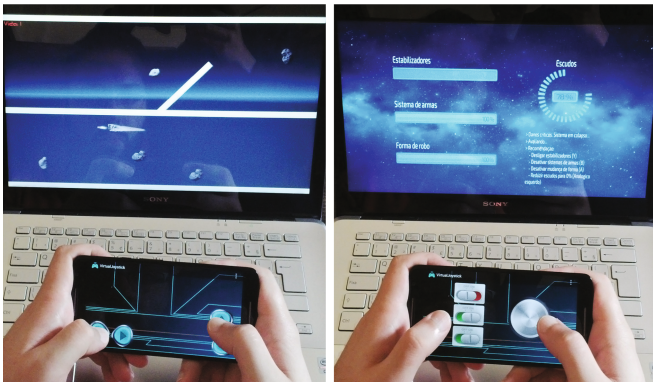


Fig. 1. A player using the adaptive controller to play our prototype game in two different moments.

3.2 User Behavior Adaptation

According to Langley [4], an adaptive interface is an interactive software system that aims to improve its interaction with a user based on a partial experience. In our case, we will use the input data of the current interaction to perform adjustments to the controller layout. This fine tuning happens constantly, always using the latest data to improve the interface. The proposed controller is fully virtual and touch based, lacking any kind of haptic feedback (except for vibrating when a button is pressed). So, unlike what happens on a traditional gamepad, the user can miss buttons when trying to interact.

The adaptive controller employs several tactics to improve the interface. The first one is touch approximation: when the user misses a button, the touch is associated to the closest button. This guarantees that mistakes will be corrected. The second tactic is preventive instead of corrective [13], analyzing the user behavior to determine how to improve the interface. All touches are used as input for a K-means clusterization [11], where K is the amount of buttons on the screen that were used at least once. This results in K clusters, each one with a centroid. As the touches are normally located close to (or inside) a button, the clusters also end up close to their boundaries. The centroids will then be associated to the closest button, and the app will start moving the buttons towards its associated centroids. The size of each button is also changed: the 1/3 most used buttons will receive a slight size boost, becoming 50% bigger. This tactics are the same employed in [12].

3.3 Prototype Game

For the testing sections, a prototype game was developed for the adaptive controller, called Guardians of Eternity. The objective is to escape the planet with the main character. It's a simple game, with two stages. The player just has to reach the far right of the stage to finish it, avoiding getting hit by enemies or colliding with obstacles. Each hit decreases its health bar (or kills it instantly on level 2). If the player dies, he will return from the last checkpoint. In level 1, the player controls a robot and has to perform simple platforming tasks to defeat enemies and avoid hazards. The controller uses a simple layout with two directional arrows and a jump button, plus a transform button. Pressing the latter changes the character to the spaceship form. Now the gameplay is a classic dual-stick shooter and the controller layout changes to two analog sticks (left one to move the ship and right one to shoot at any direction). Figure 2 shows the game screens for this stage while Fig. 3 shows the respective controller layouts used to play.

After completing the level, the main character is too damaged to keep going and it is necessary to disable several of its weapons and powers. In Fig. 2 we can see the minigame screen, where the player has to turn off the switches on the controller and turn down the power using the dial. This is an example of the kind of direct and natural interaction that this controller provides (as shown on Fig. 3), mapping a real-world action to an interface that is way more similar to real devices than mere buttons on a gamepad. The player, now stuck in the spaceship form, proceeds to enter a maze, where he has to avoid colliding with walls or asteroids. The weakened guardian has troubles to fight the effects of gravity. The player must constantly press “up” to avoid falling to the ground. The controller (as seen in Fig. 3) has buttons to control the vertical and horizontal movements, separated in the opposing sides of the interface in an attempt to provide a more ergonomic alternative to a regular d-pad.

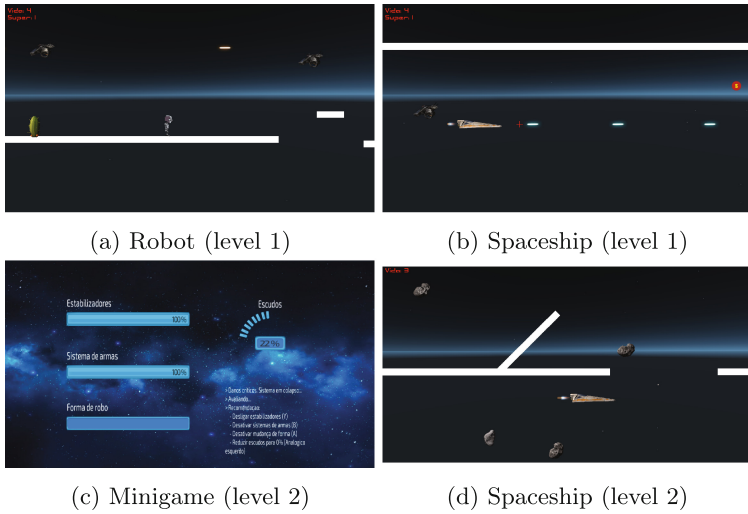


Fig. 2. Game screens for both levels. Level 1 has two gameplay modes: platforming as robot (a) and dual-stick shooter with the spaceship (b). After finishing this stage, the player enters a minigame (c) that precedes the second level (d).

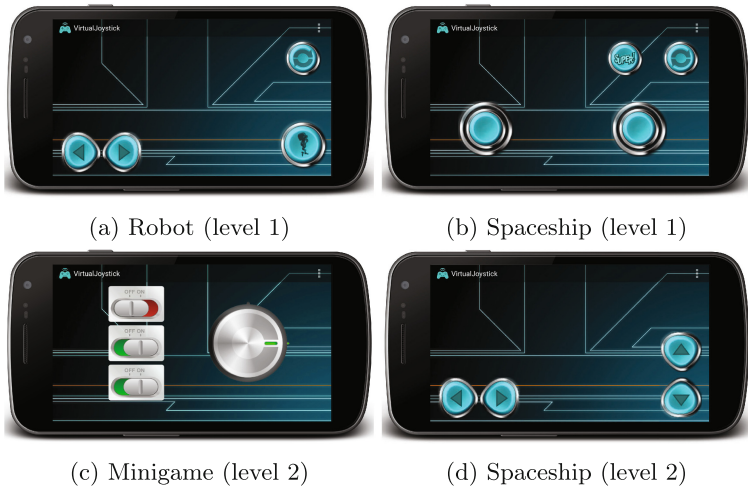


Fig. 3. The layout displayed by the adaptive controller during the different stages of the game.

4 Experiment

The tests were performed with 2 groups of 10 users each. The first one represented the advanced players: people that play videogames as a hobby, several hours per week, and usually expend a fair amount of money in gaming hardware

(consoles or GPUs for PCs) and games. The second group was the less experienced users, or casual players. These volunteers rarely play games and never expend any cash on gaming hardware, usually playing only in their smartphones.

All users had a single task: to finish the *Guardians of Eternity* game using the adaptive controller. Each level was preceded by screens explaining the plot of the game and teaching how to control the characters. In addition, as soon as a level started, we explained briefly the objective and gameplay mechanics to the users. A 20 min hard time limit was also defined in case any user failed to complete the game. However, as it was pretty short, no user failed to complete the task.

The mobile app collected a large variety of logs about the interaction, like the number of touches on the screen and its coordinates, the current active controller layout and several in-game events, like deaths and checkpoints. These data will be used for the evaluations in the next section.

The smartphone used as gamepad was a Motorola Moto X Play, running Android 6.0.1. It has a multitouch 5.5' full HD display. The game was running on a laptop with a Core i7 3537U CPU, 6 GB of RAM and a Nvidia GeForce GT735M GPU. The operating system was Ubuntu 16.04.

5 Results

With all testing sections performed, all the log files were gathered and we started evaluating which kind of improvements to the interface could be made. This analysis started by grouping the touches for all users of each group, creating a heatmap that showed which points of the screen were being touched more frequently. In the same heatmaps, the positions that the buttons reached during their real-time optimization were also mapped. These two information combined will be used to improve the layout of the controller. Heatmaps were created for all layouts, except the ones used on menus or in the minigame, since these screens only demanded a few touches in the screen. The second part of our analysis will be a detailed evaluation of the data in the log files, verifying how the changes in the controller interfered with the success rate (that is the percentage of touches that correctly hit any button and performed an in-game action).

5.1 Heatmaps

Figure 4 shows the heatmaps for the layout used to play as the robot on level 1. The scale goes from blue (lower concentration of touches) to red (higher concentration of touches), with a yellow cloud showing the positions occupied by the center of the buttons when trying to adapt to the interaction patterns. In this case, we can see that the more experienced users had a more consistent pattern of touches, focused on a smaller area, indicating that they are generally more precise when hitting the screen. In both groups, we saw a tendency to approximate the right-side buttons, allowing the player to reach both buttons with less effort and more agility. This is a case where the original interface was not very

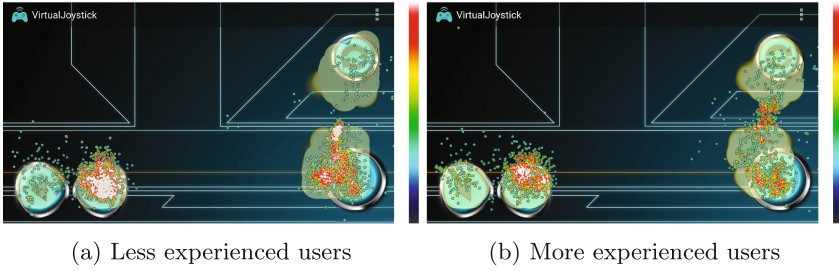


Fig. 4. Heatmaps for the layout used to play as robot, for both groups. The yellow clouds show the areas that the button occupied when adapting its position, while the colored regions show the heatmaps for the touches of the users. (Color figure online)

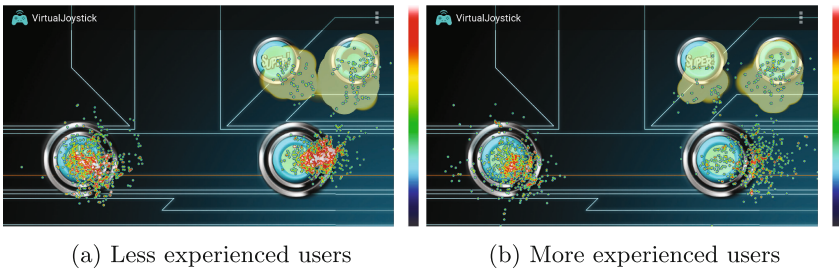


Fig. 5. Heatmaps for the layout used to play as the level 1 spaceship, for both groups. The yellow clouds show the areas that the button occupied when adapting its position, while the colored regions show the heatmaps for the touches of the users. (Color figure online)

well designed and the controller automatically fixed this mistake and reached a more ergonomic position. Also, as all tests were performed with a 5.5" phone, we could theorize that on a smaller device this effect would not be so aggressive since the buttons would not be so far apart. Anyway, it shows that this adaptation can also be useful when dealing with devices with different screen sizes.

The layout for the level 1 spaceship resulted in the heatmaps displayed on Fig. 5. Both groups showed again a tendency to bring the upper side buttons closer to the right analog stick, decreasing once more the effort to reach multiple buttons. The more experienced users, however, maintained the distance between the transform and super buttons, while the less experienced group tried to get them closer. A stark contrast is on the right analog stick, with the less experienced users showing way more touches. This was caused by a big difficulty in understanding how to play a dual stick shooter among the less experienced users. While the expert players already knew how to play such type of game and simply pointed the right analog stick towards the enemies, the less experienced players tried to press the right stick as if it was a button each time they intended

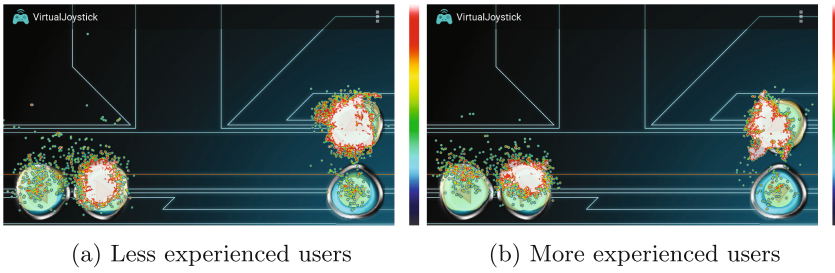


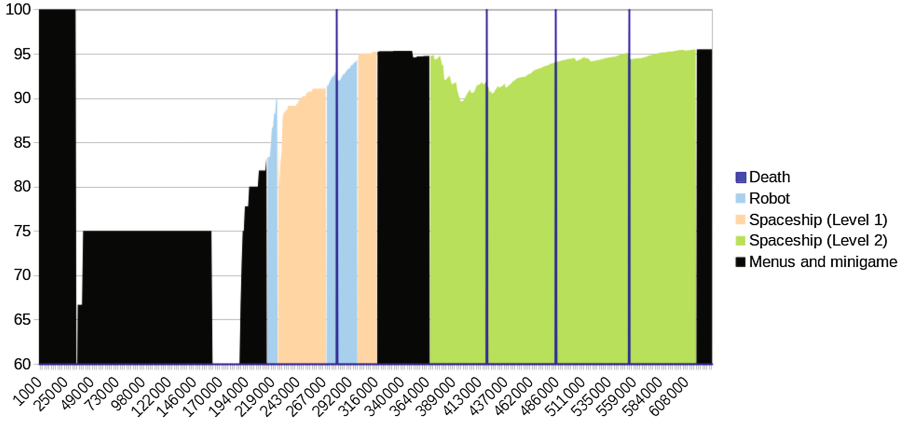
Fig. 6. Heatmaps for the layout used to play as the level 2 spaceship, for both groups. The yellow clouds show the areas that the button occupied when adapting its position, while the colored regions show the heatmaps for the touches of the users. (Color figure online)

to fire the main weapon, even after being informed about the correct way to play several times. Their lack of gaming skill and experience made this kind of control too complex. This showed that maybe the less experienced players should receive a different layout, simplified, with a normal button to simply fire forward instead of the more advanced gameplay mechanics of dual stick games. Providing different interfaces for more casual or more experienced gamers is one of the possibilities that the adaptive controller brings to gaming.

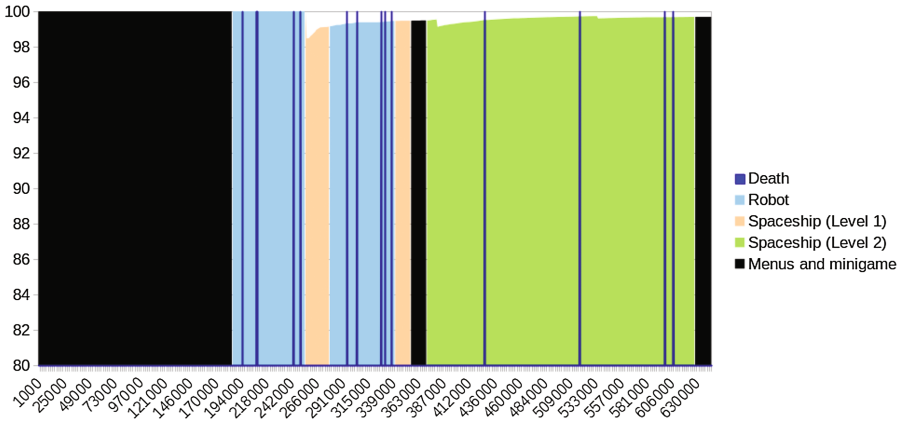
The level 2 ship controller resulted on the pattern indicated by Fig. 6. As the buttons are already close to each other, we did not have a lot of variation in the positions. The more experienced players rarely used the down button, while the less experienced players relied on it more constantly. Indeed, it is not mandatory to use this button, since the stage already has a gravity effect so the ship is always gently going down. The experienced players usually only moved up moderately and calculated their movements so most of the times they had to lower the position of the ship, it could be done with the natural force of gravity (showing superior skills). The less experienced players usually lacked this kind of control and had to compensate constantly their less gracious movements to avoid colliding to the walls.

5.2 Detailed Log Analysis

We also investigated if changes in the controller layout are influencing the game performance. One user from each group was selected. The more experienced user chosen for this analysis (Fig. 7a) had the lowest success rate between his group and the third lowest overall, being a special case. The initial black area is the time on menus, where the user was experimenting with the interface and missed some keys. As soon as he starts playing, the success rate constantly increases until he changes the interaction form (from robot to spaceship). From this moment on he starts missing buttons until the interfaces adapts to his behavior and improves his performance, a tendency that remains mainly stable until the next level. However, he misses a button when playing as robot almost at the same time



(a) More experienced user



(b) Less experienced user

Fig. 7. Complete history for the test session of one of the more experienced users (a) and one of the less experienced users (b). The X-axis is the time, in milliseconds, while the Y-axis is the success rate at that time. The color of the plot area shows which layout was active, while the dark blue horizontal lines mark each time the user died in the game. (Color figure online)

that he dies, indicating that these events may be correlated. On stage 2, the controller keeps improving its success rate, but this time we have 2 deaths close to moments where he missed buttons. This shows that, for some users, it still is necessary to adjust the layouts to reach a more optimal performance and the results of the heatmaps can be a first step to improve these layouts.

On the other hand, the less experienced user shown on Fig. 7b keeps its success rate at 100% until changing form to the spaceship. As the layout changed, he

gets momentarily lost. As he adapted to the interface and the controller adapted itself to his behavior, he regained his full success rate. The only other radical decrease is on the beginning of level 2, showing a consistent pattern. He missed buttons when the interface changed, but his performance quickly improved due to his own learning of a new interface and the controller adaptations. Also, we did not have any situation where he missed buttons and died shortly afterwards, indicating that the interface was not an issue for this user.

In both cases, we can notice that the success rate tends to fall when a new interface is introduced. After the machine learning algorithms fixed the layout, the success rate starts to improve until stabilizing at a high value. This provides an interesting motivation to create a new and fine-tuned initial layout for the adaptive controller, decreasing the initial frustration with a new interface by providing a semi-optimized version as default, that is the exact concept of this work and the objective of the final section.

6 Conclusion and Future Work

Using the touch pattern displayed by the heatmaps (Sect. 5.1), improved versions of our controller layouts were created. The layout used to play in the second stage will remain unchanged, since the proximity of its buttons already resulted in a heatmap without any significant patterns to indicate potential improvements.

When playing as the robot in level 1, the players showed a tendency to bring both action buttons together, allowing faster access to both buttons. Figure 8 shows the result of the first correction proposed in this new iterative process. The tendency of decreasing the distance between buttons is also apparent for the spaceship layout, but the clear difficulty in using a second analog stick to shoot for the less experienced users made us create two separate layouts, one for each group, as seen on Fig. 9.

In both cases, we approximated all action buttons following the pattern displayed on the heatmap. As the groups positioned the action buttons similarly, we decided to use the same position for both cases, keeping the two layouts as similar as possible. While the more experienced group will keep the dual stick mechanics, we decided to replace the second stick with a single “Fire” button that shoots straight forwards for casual players. Of course, this layout removes



Fig. 8. New controller layout for the robot on the first level.

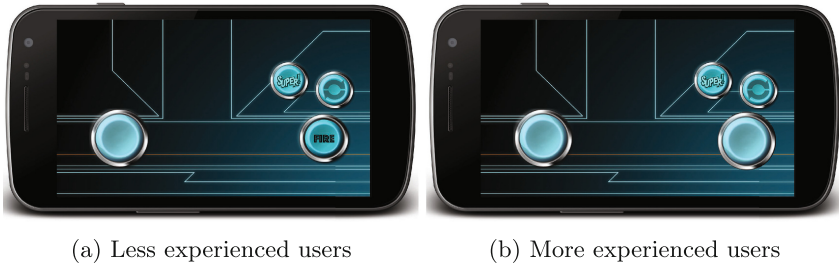


Fig. 9. New controller layouts for the spaceship on the first level.

some extra shooting features, but simplifies the task of hitting enemies by letting the less experienced players focus only on moving the ship instead of dealing with complex mechanics.

The game can show a screen before level 1 allowing the player to choose how he wants to play or simply relate these layouts to its difficulty settings. For instance, when a user plays on “Easy”, the game could use the simplified layout, assuming that a casual user will play. Selecting “Normal” or “Hard” would also select the dual stick layout, assuming that we now have a more skilled player. The latter approach will be used in the next version of the testing game.

This new iterative approach to controller design intends to improve an interface using data obtained directly from usability testings. By using all input patterns of these gameplay sessions, new interfaces that match more accurately the real behavior of the players can be designed. These results will become the new default layouts for the controller. While the machine learning routines will still optimize it for each user, the starting point will be much closer to an optimum value, shortening the time it takes to become optimized to a user and avoiding the initial decrease in the success rate when interacting with a new layout (Sect. 5.2). This process can be repeated after applying the changes, until the controller reaches the best default layout for a comfortable gaming experience.

Future developments of this research include the application of this approach in different types and genres of games, evaluating how it can impact the development of controller interfaces. Another interesting research is to apply the concept of having different controller layouts for specific user profiles, such as simplified interfaces for casual users, evaluating how it impacts their performance and enjoyment of different games.

Acknowledgments. We would like to thank our volunteers for their participation in our testing sessions. We are also grateful to CAPES, CNPQ and FAPERJ for their financial support.

References

1. Baldwin, T., Chai, J.: Towards online adaptation and personalization of key-target resizing for mobile devices. In: Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, pp. 11–20. ACM (2012)
2. Bi, X., Zhai, S.: Bayesian touch: a statistical criterion of target selection with finger touch. In: Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, pp. 51–60. ACM (2013)
3. Hollensen, S.: The blue ocean that disappeared—the case of nintendo wii. *J. Bus. Strategy* **34**(5), 25–35 (2013)
4. Langley, P.: Machine learning for adaptive user interfaces. In: Brewka, G., Habel, C., Nebel, B. (eds.) *KI 1997. LNCS*, vol. 1303, pp. 53–62. Springer, Heidelberg (1997). doi:[10.1007/3540634932_3](https://doi.org/10.1007/3540634932_3)
5. Marshall, D., Ward, T., McLoone, S.: From chasing dots to reading minds: the past, present, and future of video game interaction. *Crossroads* **13**(2), 10 (2006)
6. Merdenyan, B., Petrie, H.: User reviews of gamepad controllers: a source of user requirements and user experience. In: Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play, pp. 643–648. ACM (2015)
7. Mott, T.: *1001 Video Games You Must Play Before You Die: You Must Play Before You Die*. Hachette, UK (2011)
8. Oshita, M., Ishikawa, H.: Gamepad vs. touchscreen: a comparison of action selection interfaces in computer games. In: Proceedings of the Workshop at SIGGRAPH Asia, pp. 27–31. ACM (2012)
9. Pelegrino, M., Torok, L., Trevisan, D., Clua, E.: Creating and designing customized and dynamic game interfaces using smartphones and touchscreen. In: 2014 Brazilian Symposium on Computer Games and Digital Entertainment, pp. 133–139, November 2014
10. Rogers, S., Williamson, J., Stewart, C., Murray-Smith, R.: Fingercloud: uncertainty and autonomy handover incapacitive sensing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 577–580. ACM (2010)
11. Smola, A., Vishwanathan, S.V.N.: Introduction to machine learning, vol. 32, p. 34. Cambridge University, UK (2008)
12. Torok, L., Pelegrino, M., Lessa, J., Trevisan, D.G., Vasconcelos, C.N., Clua, E., Montenegro, A.: Evaluating and customizing user interaction in an adaptive game controller. In: Marcus, A. (ed.) *DUXU 2015. LNCS*, vol. 9188, pp. 315–326. Springer, Cham (2015). doi:[10.1007/978-3-319-20889-3_30](https://doi.org/10.1007/978-3-319-20889-3_30)
13. Torok, L., Pelegrino, M., Trevisan, D.G., Clua, E., Montenegro, A.: A mobile game controller adapted to the gameplay and user’s behavior using machine learning. In: Chorianopoulos, K., Divitini, M., Hauge, J.B., Jaccheri, L., Malaka, R. (eds.) *ICEC 2015. LNCS*, vol. 9353, pp. 3–16. Springer, Cham (2015). doi:[10.1007/978-3-319-24589-8_1](https://doi.org/10.1007/978-3-319-24589-8_1)
14. Weir, D., Rogers, S., Murray-Smith, R., Löchtefeld, M.: A user-specific machine learning approach for improving touch accuracy on mobile devices. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, pp. 465–476. ACM (2012)
15. Zaman, L., MacKenzie, I.S.: Evaluation of nano-stick, foam buttons, and other input methods for gameplay on touchscreen phones. In: International Conference on Multimedia and Human-Computer Interaction-MHCI 2013, p. 69-1 (2013)