# **Guidelines for Designing Dynamic Applications With Second Screen**

Bruno Pagno\*, Diogo Costa<sup>†</sup>, Leandro Guedes<sup>‡</sup>, Carla Dal Sasso Freitas<sup>§</sup> and Luciana Nedel<sup>¶</sup>

Instituto de Informática Universidade Federal do Rio Grande do Sul (UFRGS) Porto Alegre, Brazil Emails: {blpagno<sup>\*</sup>,diogo.costa<sup>†</sup>,lsguedes<sup>‡</sup>,carla<sup>§</sup>,nedel<sup>¶</sup>}@inf.ufrgs.br

Abstract—The concept of second screen became popular with the introduction of interactive TVs. In this context, while the user focuses on the TV screen, the exploration of additional content is possible through the use of a smartphone or tablet as a second screen. Lately, dynamic applications, e.g. video games, also started to use a second screen. Nintendo DS and Wii U are the game consoles that began to incorporate these ideas. Dynamic applications are based on real time action and interaction, and their implementation can be very complex specially because users have to change focus between the displays frequently. In this paper, we summarize the results found in a set of experimental studies we conducted to verify the usability of interaction techniques based on the use of a second auxiliary screen in dynamic applications. We developed a multiplayer game that employs one main screen shared by two players, each one also using a second (private) screen. From these studies, we elaborate a set of guidelines to help developers in the use of second screens. Although future case studies would improve these guidelines, our experiments show that they contribute with robust principles for developers who want to build multiscreen applications.

#### Keywords-second screen; interaction; guidelines.

# I. INTRODUCTION

The use of multiple displays is a powerful solution for presentation and interaction since they can be used in many applications as long as there is a user interacting with a software. Dynamic applications using multiple screens are gaining popularity because of Nintendo DS and Wii U consoles. Moreover, there are an increasing number of tablets and smartphones entering the home media environment and allowing interaction with television [1]. The combination of mobile devices and large displays, televisions and projectors make a perfect fit for multiple screen applications. New communication technologies and mobile operating systems also help to make this possible.

There are several studies reporting the integration of multiple screens and users, but interactive TV is the most common field [2][3][4][5][6]. Games, in particular, have a high multiuser/multiplayer characteristic making them suitable for such integration. Even in the most simple game, where objectives are quite simple, there are situations where the players can join and interact. Meeting software systems [7] have also been discussed as good examples of applications based on multiple screens.

In this paper, we present guidelines for building multiple screen applications. We base our guidelines on the analysis of some works related to second screen technology and also on our own experiments with an action game using



Figure 1. Space Crusher application being played by two users, each one using a smartphone as the second screen.

a main screen shared by the players and one smartphone for each player, which provide the secondary screen of the application (see Figure 1). The software developed for this work is called SpaceCrusher. It is a shoot'em up [8] multiplayer game, simple and easy to learn. We developed two versions of SpaceCrusher: a collaborative and a competitive version.

The objective of our work is to provide guidance for developers and researchers who wish to take advantage of best practices when designing multiple screens applications.

The rest of the paper is organized as follows. Section II presents related works on collaborative games, competitive games, and second screen usage. Then, in Section III we describe our game, providing both context of use and details about it. In Sections IV and V, we report the experiments that allowed us to derive different guidelines, which are presented in Section VI. Finally, in Section VII we draw some final comments about our results and discuss possibilities of future research.

#### **II. RELATED WORKS**

There are some works related to ours in different ways, all of them with pros and cons about the usability and interaction using second screen.

Stefik et al. [7] proposed a multiple screen software for meetings. In that work, they raised a few points very relevant to multiple screen applications. Firstly, they discuss the need for keeping all user's updates synchronized. Even though their software cannot be classified as a dynamic application, they noticed that delays can cause discomfort, thus having everything updated is essential. They also mention the use of private and shared information. Frequently, users make notes that they do not wish to share with others, so keeping some information private can be useful. Also, because of the technology available at that time (1987), they used connected PCs for their work. It is interesting to observe the evolution since then, and notice the impact of mobile devices available today.

Tsekleves et al. [9] discussed the idea of interacting with digital media at home via a second screen. They investigated how people use Interactive Television (iTV) services: their likes, dislikes, preferences and opinions. Then, they developed a second screen-based prototype in response to these findings and tested it with iTV users in their homes. The positive user responses led them to extend the scope of their previous research to look into other related aspects such as barriers to digital interactive media and personalization of digital interactive media at home.

Also proposing a solution for a secondary screen for TVs, Soskic et al. [6] describe a project based on an Android device, used as a remote control In an *app*, the Android user would access the TV guide, watch the channels mosaic (and select one of them). The user could also watch a different program from the one he is watching on TV. All these actions are performed independently from what is on the main TV, which keeps showing what is selected there. So the user ends up with more freedom to browse other channels without interrupting the program in the main screen.

In a recent article, Bernhaupt et al. [10] present the state-of-the-art of current types of second screen approaches and a classification attempt based on the dimensions of users' perceived synchronization and perceived interactivity. Based on the analysis of twenty second screen apps available in Europe, they performed a general task analysis of second screen applications. Subsequently, the tasks identified in the study were represented in paper prototypes and used for evaluation by a usability expert and also in experiments for assessing user experience. Results show that the presentation of information on two screens typically causes the user interface (UI) to be less efficient and effective, and that the higher the degree of interactivity, the less usable the UI is. In terms of user experience, they have identified esthetics, emotion, value/meaning and stimulation as well as social connectedness as critical dimensions. Courtois and D'heer [11] have investigated how tablet users incorporate multiple media into their television viewing experience. Three patterns were found: (a) only focusing on television, (b) confounding television viewing with other screen media (e.g. laptop, tablet) and (c) confounding television viewing with various media, including print and screen media. Their question was how the incorporation of screen media in

this experience affects the practice of engaging in digital comments about television content. The results, based on a sample of 260 users of tablet, indicated that there is only a modest uptake and interest in using secondary screens to share opinions digitally. However, only 15% of the participants had experience in working with both TV and screen media, which is a few users. So the interest in the simultaneous use of both devices is small.

Interactive mobile technologies have become part of audience experiences of live performances in terms of general media sharing and specific extra content. Barkhuus et al. [12] report a study based on a scenario with high resolution, panoramic image video stream, where audience members view on a tablet the very same live event they are watching. The video stream on the tablet is navigational and enables audience members to pan and zoom in the real-time video feed. They studied audience interaction and impressions in three performances of a dance and music show, and found distinct uses of the second screen video stream. Also, their study shows how working with perceived conflicts in technology can still open up design space for interactive technologies.

The use of a second screen is also present in game platforms such as Nintendo Wii U consoles [13]. The Nintendo Wii U uses a specific controller with a screen and buttons. The most interesting fact is that this controller provides private information for the user, letting other players only interact with the main screen. Nintendo also has a portable console called Nintendo 3DS [14]. It works with two screens – the main one and the touchscreen – letting the user play whenever they want because of their pocket size.

Multiple screens are also very common in desktop environments. One of the issues arised from this is how to use the cursor in such screens. There has been some studies made focusing on enhancing the mouse [15], [16] or replacing it for a different device [17]. They provide solutions that work on different sizes and compositions of screens. The solutions, though, are not very effective for our work, which is focused much more on auxiliary second screens.

As can be noticed, these works propose specific usages for a second screen, with emphasis on additional information for TV users. Wii U and Nintendo DS are proprietary solutions, serving only for games. Our work intends to provide generic guidelines for developing dynamic applications with a second screen, independently of its use.

#### III. MULTIPLE SCREENS IN SHOOT'EM UP GAMES

There are several shooting games available in the market but our aim was to have more control for testing the use of second screens. We used Space Invaders [18] as inspiration in the development of our game.

The success of any application using a second screen depends on a series of aspects. The choice of the device in which the software runs is very important. It should be evaluated according to the location it will be used and the popular choices for that kind of application [19]. Existing multiple display devices such as the Nintendo DS system are closed platforms, with very specific purposes, and limited resources available for developers. Thus, it is likely that devices such as smartphones and tablets become the main choice for a second screen when interacting with computers, TVs and projectors.

The placement of information is also a very important aspect in second screen applications. If the user has to keep changing focus too often for looking at specific information, the usability of the application will be compromised [10]. It is well known that the forms of feedback given to the user, be it visual or haptic, also play important role in the application usability.

In this section we describe the multiplayer game we developed as an example of the use of a second screen.

#### A. Overview

We developed a "shoot'em up" game based on a shared main screen and a smartphone for each player. In the game each player controls a ship that fires bullets and destroys asteroids. The main goal is to obtain the highest score by destroying as many asteroids as possible.

The main screen (Figure 2a) is where the game actually happens. It contains the ships, represented as colored rectangles at the bottom of the screen, and several elements that fall vertically from the top (Figure 3). The players can move the ship horizontally along the bottom of the screen, while it keeps firing automatically. Each time a bullet hits an asteroid (Fig.3a) it counts an amount of points to the respective player. If the player's ship touches an asteroid, he looses one life, and if the user touches another element he collects that item.

The second screen (Figs. 2b and 2c) is where the players interact with the application. It displays the controls of the game along with the score and remaining items/lives information. The elipse at the top reproduces the position of the ship in the main screen. It helps the player keep the context, while he changes focus from one screen to another.

We intended to keep the game as simple as possible and focus on multiple screen characteristics. Because of that we only developed one level in the game which can be played in different difficulty levels. As a result, the final application is intuitive and easy to play.

In the development process, we proposed two versions: one focused on collaboration and the other on competition. The second (competitive) version came as an evolution of the collaborative version. We replaced few features to test different aspects of multiple screen software. Both versions shared the core code and gameplay features.

1) Collaborative Version: This version of the game is focused on collaboration. Each player controls his/her ship and has an individual score. Still, their objective is to score the most points possible as a team.

In this version players can share resources such as lives and weapons in the game. One user can send the item to the other in order to help him achieve better individual results, and consequently help the team.





(b) Collaborative Score: 6

(c) Competitive

Figure 2. SpaceCrusher's interface: main (a), and second screen (b).

Another feature is the roulette (Fig. 4), shown when the player touches a lightning icon (Figure 3b). It is a special element that spins a few items and randomly adds lives/weapons to the player or asteroids to the screen. While the roulette is active the player's ship goes into a suspension mode where it becomes invincible, meaning that the ship is not affected by asteroids or any hazards. This allows the player to drive his attention to the second screen.

There is also another collectable item available in this version: the heart (Figure 3c), which add lives to the player.

The application also allows some set up configuration. When starting the game it is possible to set the dificulty level, which defines the amount of asteroids present in the game.

2) Competitive Version: The competitive version was built after the collaborative had been evaluated. It is an



Figure 3. Elements (items) of the Space Crusher game.



Figure 4. Roulette special item in Space Crusher.

adaptation meant to experiment different variables than the ones tested in our first evaluation, which is described in Section IV.

This version of the application forces the players to compete for the highest score between them. The final score is separated for each player, so they can check who gets the better result. The possibility of sharing items was removed from this version since it does not fit a competitive environment.

The collectable items changed a bit from the collaborative version. The lightning was replaced by the bomb (Figure 3d) and the coin (Figure 3e) was included. The coin adds a few points to the score of the player who collects it.

The bomb was designed to be used as a strategy to attack the opponent player. When the user activates the bomb, he/she goes into a "bomb placing mode", in which it must be decided where the bomb will fall. Like in the roulette, when the user is placing the bomb he/she becomes invulnerable. The user chooses where to drop the bomb by controlling the position of the elipse on the second screen (Figure 2c). We designed two different possibilities for this mode: the player's ship turns into a ghost (semi-transparent) or becomes completelly invisible on the main screen. In the first case, the opponent can see where the bomb is being placed. In the second case it becomes harder to know where exactly the bomb is going to fall. Also, to diferentiate the collectable bomb and the hazardous bomb we added a border to the one that can be



Figure 5. Overview of the application architecture

collected.

This version also features a vibrotactile feedback when the user is hit by any hazardous item.

The application configuration was expanded in the competitive version. We added options to select which bomb mode would be used, to show private information on the main screen, and to set the size of the buttons.

#### B. Implementation

The game was implemented as two applications, one for desktop and the other for mobile devices. We identify them as the main and the secondary screen, respectively. Both were developed using the Unity3D [20] game engine. For the communication between applications we used a classic client-server architecture (see Figure 5) with the help of Unity's networking API. The server runs in the desktop application, while the smartphones are the clients. Both applications present information but users interact only with the mobile version.

1) Server Application: The server application should run on a shared screen visible to all players. It also controls all the internal mechanics of the game, such as the amount of items and lives each player has. This way the game depends minimally on external intelligence to work correctly.

Since the communication of the application was built using Unity's networking API, we set up a local master server<sup>1</sup>. The master server is an application provided by Unity to be used with their networking API. As the API has several limitations, specially when working with smartphone's operating systems, we had to implement the communication between the server and client through RPC. Even though this is not an optimal setup regarding performance, during our tests the communication occurred without any problem.

The initial setup of the application was set on the server and sent to each client when they connected to the server.

The server application also records log files with information about the players when each match finishes.

2) *Client Application:* The client application is where users actually interact (Figure 2b). It serves as the controller for the game that is running on the server, but it also presents information, serving as a second screen. It was developed for handheld devices and, because of

 $<sup>^1\</sup>mathrm{Master}$  server application is accessible at http://unity3d.com/masterserver/

that, its interface is based on touchscreen features. We also prepared a desktop version for a test as explained in Section IV. The desktop adaptation was controlled with the keyboard.

Like the server, the client application uses Unity's networking API and RPC for communication. Whenever the player executes an action, the client sends a message to the server, which executes the action, giving the client a reply or not. As explained before, RPC is not very efficient, but it was enough for our experiments.

It is possible to connect multiple clients to the server, each one being given a different color so each player can recognize its own ship on the main screen.

# C. User Interface

We have built two slightly different interfaces for this project. Still, our objective, during every step of the development was to keep the interface simple, with the minimum amount of elements possible.

The main screen (Figure 2a) shows an interface with few elements. During a match the only items visible are the ships represented as rectangles at the bottom of the screen, the asteroids and collectable items (Figure 3) and the bullets. We added player's score information to the main interface in some tests to verify the best placement for that information. This clean interface makes the software very easy to understand and use.

The second screen (Figs. 2b and 2c) is also as clean as possible, containing only the buttons needed for interaction and the private information regarding each player. We also added an icon (the ellipse) to show the position of the ship on the mobile too. This way, if the player has to change focus and look at the mobile for any reason, she/he will not loose the tracking of his position in the game.

We restructured the interface of the mobile application for the second version. Mobile hardware is highly diverse, so it was necessary to make the software more adaptable. We changed the buttons to have dynamic sizes, so the interface looks the same in all kinds of devices.

With the idea of using the resources available on the second screen in mind, we designed the item sharing feature based on touch gestures. The player could send an item or life to the partner dragging it for a minimal distance in any direction. In the desktop adaptation, we designed specific buttons for each item.

# IV. USER TEST I - COLLABORATIVE

The first experiment was applied to the collaborative version of the Space Crusher game. It was performed with a sample of 14 subjects, divided into 7 pairs, and lasted approximately 20 minutes each. It was applied in four rounds of two minutes each. These rounds were divided into two difficulty levels and two different devices: PC and smartphone. The first pair started the test using the smartphone, the second with the PC, and so on. We collected qualitative and quantitative data using characterization and opinion questionnaires, and logging the matches played. Illustrative pictures taken during the experiment can be seen in Figure 6.





(b)

Figure 6. Users testing the game with different devices: notebooks (a) and smartphones (b).

Except for the order, the task given to all the users was the same. They should connect the client application to the server and play the four designed matches. They were also asked to cooperate to get the highest sum of points they could using the resources given.

#### A. Variables

We have established as independent variables the device for the client application (smartphone or notebook) and the two difficulty levels.

The dependent variables recorded in our experiment were: satisfaction with the game, fun, and the performance measured by the application and registered in the logs.

# B. Hypotheses

With the results obtained in this test we intended to prove the following hypotheses:

- *H1*: The collaboration facilitates the gaming experience.
- *H2*: Using a mobile device as secondary screen is better than using a computer.
- *H3*: The presence of information in two screens enhances the gaming experience.

# C. Results

Thirteen males and one female tested the collaborative version. The mean age was 26.5 years old. They used to play electronic games (more frequently computer games): 64% of them have played action games on a mobile device previously, while only 21% have played games with a second screen (e.g. Wii U, using two monitors, or Nintendo DS).

Using the data collected during the matches, we were able to find some insightful results. In general, the performance of each pair was better using the desktop version of the software, but when using the smartphone, teams were still able to complete every task successfully.

We noticed that collaboration in this game tended to happen based on specific needs. This means that pairs of players with better individual skills did not feel the need for collaboration, and ended up not collaborating much. Players that exchanged items when near to being defeated extended their time in the game. In situations where one player would die and leave the partner alone, the other would readily give one of his "lives" to the dying player, continuing the round until time ran out.

1) Questionnaires and Logs: Players feedback were very positive in relation to the use of mobiles. 65% of them were satisfied while 28% were neutral about the mobile version. When asked about the fun factor, 71% really enjoyed the experience, while 28% were still neutral.

On the other hand, the desktop version did not receive such a positive evaluation. It had 50% satisfaction and fun approval, with 21% of the feedbacks as neutral.

As user experience in general, 85% of the players felt they had a good experience using multiple screens, and 78% approved and enjoyed the collaboration.

Based on the logs, it was possible to infer a direct relation between the user's score and the device used, with p < 0.0299 in the Analysis of Variance's (ANOVA) test. Also, the mean value for the amount of collaboration iterations indicates that players collaborated similarly on both devices.

2) Verifying the Hypotheses: Regarding the hypotheses stated for this experiment, we can conclude the following:

- *H1: The collaboration facilitates the gaming experience.* We consider H1 confirmed based on the fact that players actually collaborated when needed. When the game presented hard situations, they instantly exchanged their items to try to achieve a better score. However, we should conduct more tests with and without collaboration in the future to be fully safe to argument this.
- *H2: Using a mobile device as another screen is better than using a PC.* H2 can also be considered true based on the questionnaires' feedback. Most users preferred the mobile rather than the desktop interaction.
- *H3: The presence of information in two screens enhances the gaming experience.* According to players' feedback, this seems to be very positive. Several users mentioned the advantage of having their private

information hidden as they felt it was useful strategy to play the game. Separating user's private information and public information seems to be a reasonable solution to local multiplayer games.

# V. USER TEST II - COMPETITIVE

After performing the first experiment, we noticed that the interface of the game had some elements that could be more thoroughly analyzed. Based on the elements we wanted to evaluate, we decided to develop the competitive version of the application. However, since we have a considerable amount of variables to verify and, in order to avoid making new tests too lengthy and potentially confusing, we isolated two variables (buttons size and vibration in the smartphone used as second screen) and made an informal "pre-test" with them.

In this pre-test, we observed 10 users playing in single player mode. They were asked to try to get the highest score they could achieve. The setup involved buttons of three different sizes (small, medium, and big) combined with the use or not of vibrotactile feedback. The test result in six matches with different configurations each. After the tests we asked the users to answer which was their favorite setup.

Based on their answers, we could notice that the vibration feedback is very positive. Ten of the ten users found that the experience was better with vibration even if it was not necessary for the game. The largest button was preference of eight users while two chose the medium button as their favorites. No user liked the smallest button.

From the analysis of the logs generated by each match in this pre-test, it was possible to notice that players missclicked far more frequently with smaller buttons. Nevertheless, the difference in missclicks between mediumsized and the biggest buttons was not large.

The last set of experiments was performed with the competitive version of the game. It was applied to 30 subjects divided into 15 pairs. Again, the experiment was divided into four rounds of two minutes each. This time we tested the display of private information (score and lives) on the main screen and the bomb mode used (ghost or invisible). As in the first experiment, we applied characterization and opinion questionaries, and also logged users' actions. Figure 7 shows one pair of users during the experiment.

We asked each user to play four matches of the game trying to perform better than the opponent. At the end of each match they compared scores.

#### A. Variables

For each test we alternated between displaying private information on the smartphone and on the main screen, or just on the smartphone. We also alternated between the bomb mode: showing the ghost ship (Figure 7) or becoming completely invisible.

# B. Hypotheses

With the results obtained in this experiment, we intended to prove the following hypotheses:



Figure 7. Users testing the competitive version with the ghost bomb mode on.

- *H4*: A second screen allows the use of personal and confidential information and this affects the user's strategy in the game.
- *H5*: The use of additional interaction resources that are not available on the main screen enhances the gaming experience.

# C. Results

The experiment was performed by 30 persons (27 men and 3 women) with a mean age of 24.4 years old. All of them were undergraduate students in computer science. They reported to have a good experience with tablets and smartphones, and use to play games sometimes with these devices. The frequency they played on other platforms like consoles and computers was similar. Only eight of the users had experience with applications that make use of a second screen, highlighting Wii U and Nintendo DS and a racing simulator which uses multiple screens. All of them had a positive experience with this kind of applications.

In general, the performance of the users during the experiment was better with public information (i.e. information displayed on the main screen) and the ghost bomb mode.

1) Questionnaires and Logs: Since the size of the buttons were previously evaluated, we set them to the largest size, but still asked the users' opinion about it. In this competitive version, 57% of the users found the size of the buttons comfortable, but 43% think they could be even larger.

We asked the users about how natural, comfortable and satisfying is the use of the second screen (Figure 8). Considering 1 and 2 as bad results, 3, neutral, and 4 and 5 as good, we concluded that: (a) 23% thought that the use of the second screen was not natural, 33% of the users were neutral, and 44% thought it was natural; (b) 33% of the users thought that is comfortable to use the second screen, 40% had a neutral opinion, and 27% found it uncomfortable; (c) 47% concluded that it was satisfying (good) to use the second screen, 37% had an average opinion, and only 16% concluded that it was bad.

Scores were highly influenced by the game mode, private or public information and ghost or invisible bomb



Figure 8. Users' evaluation of how natural, comfortable, and satisfying is the use of the second screen: 1 (blue) is the worst result and 5 (brown) the best.

(see Figure 9). Preference for public information was reported by 84% of the users, mostly because they wanted to focus on the main screen's challenges. Private information was preferred by only 13% of the users, mainly because they could hide the score and number of lives from the other user, increasing the competition. Only 3% of the users were neutral, finding positive and negative aspects in both modes.



Figure 9. Scores obtained by players using public and private information.

The final scores confirm what the users thought about private and public scores, even without a significant result when looking at the p value (p = 0.8239). Public information resulted in a higher score in most individual cases and 4.3% better score in general.

Additionally 53% of the users thought that using a touchscreen and vibration enhanced the gaming experience, 27% had a neutral experience, while 20% did not feel that touchscreen and vibration enhanced the gaming experience (Figure 10).

Bomb mode also influenced the game results. When we asked for the user which visualization was preferred for the bomb mode, 43% of the them preferred the ghost, and the reason most cited was that they knew their location on the main screen, making it easier to throw the bomb; 30% were neutral; and 27% preferred the invisible mode, because in this case, the opponent would not know their position, making the game funnier. Additionally, when we asked users which visualization they preferred for



Figure 10. Evaluation of the improvement in user experience employing the resources available in the smartphone: 1 means no improvement while 5 is the maximum of improvement.

the opponent, 33% preferred the ghost bomb, 13% the invisible bomb, and 54% had no specific preference. The main reason pointed out is that they did not notice any significant change in the final result.

2) Verifying the Hypotheses:

- *H4: A second screen allows the use of personal and confidential information and this affects to the user's strategy in the game.* H4 was not actually confirmed. Most of the users preferred public information because they could check things easily, and this means that they did not care about sharing their information. However, users that preferred using private information instead of public did so to hide information from the opponent, because that surely would contribute to their game strategy.
- *H5: Use additional interaction features that are not available in the main screen enhances the gaming experience.* We considered H5 true. Since the pre-test, we realized that users enjoyed the non-conventional interaction features (vibration and touchscreen). Moreover, from the second test, most of the users liked both features. Vibration gave feedback to users' realize when they died or were hit by a bomb, and the touchscreen made the users control and interact with different game scenes. However, we had negative feedback about finding the controls. Since they played with a flat touchscreen, a lot of missclicks happened, and sometimes they had to stop paying attention to the main screen to look at the smartphone and check what they were pressing.

# VI. GUIDELINES FOR THE DESIGN OF SECOND SCREEN INTERFACES

Some important lessons have been learned from the study of previous works involving the use of second screen in dynamic applications, the development of a multiple screen video game (see Section III), and the application of a set of user experiments. In this section we propose a set of guidelines for the design of dynamic interactive applications using the concept of second screen.

Avoid the change of the user's focus between the screens. The way and frequency the user changes focus from one screen to the other can impair the usability [10] of the interface or even prevent the task to be performed [21]. So, applications that use multiple screens should always preserve focus on one screen at a time. In our experiments, several users complained about having to

look to the smartphone to know their score and remaining lives and, in consequence, loosing lives or opportunities of increasing their score. When comparing the score from matches with private and public information, we found out that tests with public information had a higher score mean, though the variance was not statistically meaningful (p = 0.8239).

Keep the second screen near the main screen in dynamic applications. When playing a multiple screen game like Space Crusher, players have to keep watching the main screen. Their attention cannot be constantly divided between both screens. For dynamic applications, it is recommended that both screens are kept in the users' field of view, as occurs in the Nintendo DS console. In doing so, the focus of the user is not lost due to the change of screens.

**Replicate the content of the "action" window.** Even though replicating the "action" content may seem to be a waste of processing or make unnecessary the second screen, we noticed that it can be a solution to certain situations where one absolutely need to change the focus to the other screen. Players of SpaceCrusher had problems to know where to put the bomb in the invisible bomb mode. If the screen was replicated we could add individual information on the private screen enhancing the application.

Make use of private and public information. When the players performed the experiments, there was a large preference for having the score and remaining lives information displayed on the main screen. Even if the information could be considered to be individual and be used as a game strategy, players preferred it on the main screen because of the change in focus. Based on that, we concluded that individual information for players should not impact the user experience. The impact of changing focus is bigger than the gain from having private information. Still, if the application is not real time and action-based, the necessary private information can be kept on the second screen. Another option would be to let the user control the information he/she wants to keep private [22].

Use physical buttons, sensors or actuators instead of virtual controllers. The lack of haptic feedback in the virtual controllers can be a hard problem for users. When the focus is in the primary screen, players often loose the location of the buttons and start tapping on wrong places on the smartphone screen. In our experiments, even with large buttons there was a considerable number of "missclicks". However, when considering the preference for using smartphones instead of a notebook as the second screen, the best solution is likely to use the resources available within the mobile device, such as accelerometer/gyroscope or vibration, as a feedback when a button is pressed.

Use standard technology. When dealing with multiple, different devices as a second screen and different operating systems, developing an application can be quite complicated. There are a few software solutions that make it easier, though. The Unity3d game engine is a good example since an application made in it can be deployed on many different platforms. Of course, there are several details of implementation such as screen resolution or performance issues, but with software that helps the deployment to many platforms and the easy communication between the different devices, everything becomes much easier.

**Favor the use of easy to learn controls.** Where there is much action in a software, the controls should be easy to understand. The user should not be forced to switch to another interface to keep remembering which control does what. Bernhaupt et al. [10] reinforce that the usability in applications using multiple screens can be impaired because of inconsistent use of labels.

Keep all the screens instantaniously synchronized. In environments where the second screen also controls the application like SpaceCrusher and the works of Bernhaupt et al. [10] and Tsekleves et al. [9], the answer to the user's action must be immediate. Even in applications that are not action-based the delay between commands and the interface reaction will cause confusion and usability problems.

# VII. CONCLUSIONS

In this paper we presented a set of guidelines aiming at helping researchers and developers to build better multiple screen applications. The guidelines were devised based both on related work and lessons learned from experiments performed with a multiple screen game we developed within this project.

We tested our application with several users and obtained a positive feedback about the software, as well as the use of a mobile device as the second screen and input device at the same time.

For future work, we aim at improving the application by adding the use of the gyroscope as an option for control, instead of touchscreen virtual buttons. We also planned to print a mask to use over the mobile phone screen to give tactile feedback and facilitate finding the game controls. There are also improvements in the interface that could be done, like better status bars on the main screen for the players to check their scores.

## **ACKNOWLEDGMENTS**

The authors would like to thank the UFRGS Graphics, Visualization and Interaction lab, all the users that helped with testing, and specially CNPq for making this paper possible.

#### REFERENCES

[1] E. D'heer, C. Courtois, and S. Paulussen, "Everyday life in (front of) the screen: The consumption of multiple screen technologies in the living room context," in *Proceedings of the 10th European Conference on Interactive Tv and Video*, ser. EuroiTV '12. New York, NY, USA: ACM, 2012, pp. 195–198. [Online]. Available: http://doi.acm.org/10.1145/2325616.2325654

- [2] R. Mendes Costa Segundo and C. Saibel Santos, "Second screen event flow synchronization," in *Broadband Multimedia Systems and Broadcasting (BMSB)*, 2013 IEEE International Symposium on, June 2013, pp. 1–7.
- [3] C. Albuquerque Rodrigues, E. de Lima Reis, and V. Ferreira de Lucena Junior, "Second screen prototype for broadcasted digital tv users in isdb-tb standard," in *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference* on, Oct 2014, pp. 266–267.
- [4] P. Cesar, D. Bulterman, and J. Jansen, "Leveraging user impact: an architecture for secondary screens usage in interactive television," *Multimedia Systems*, vol. 15, no. 3, pp. 127–142, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00530-009-0159-z
- [5] L. Cruickshank, E. Tsekleves, R. Whitham, A. Hill, and K. Kondo, "Making Interactive TV Easier to Use: Interface Design for a Second Screen Approach," *Design Journal*, vol. 10, pp. 41–53, 2007.
- [6] N. Soskic, D. P. Tapavicki, D. Rapic, S. Medic, and N. Kuzmanovic, "Remote control android application for second screen dtv environment," in *Embedded Computing* (*MECO*), 2014 3rd Mediterranean Conference on, June 2014, pp. 272–274.
- [7] M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar, "Wysiwis revised: Early experiences with multiuser interfaces," *ACM Trans. Inf. Syst.*, vol. 5, no. 2, pp. 147–167, Apr. 1987. [Online]. Available: http://doi.acm.org/10.1145/27636.28056
- [8] F. Muliawan, "Enemy speed control on shoot em' up game with fuzzy takagi sugeno method," in *Student Project Conference (ICT-ISPC)*, 2014 Third ICT International, March 2014, pp. 87–90.
- [9] E. Tsekleves, L. Cruickshank, A. Hill, K. Kondo, and R. Whitham, "Interacting with digital media at home via a second screen," in *Multimedia Workshops, 2007. ISMW* '07. Ninth IEEE International Symposium on, Dec 2007, pp. 201–206.
- [10] R. Bernhaupt, M. Pirker, and B. Gatellier, "Identification of user experience and usability dimensions for second screen applications: Results from an expert evaluation using generic task models," *Proceedings of International Broadcasting Convention Conference - IBC 2013*, 2013.
- [11] C. Courtois and E. D'heer, "Second screen applications and tablet users: Constellation, awareness, experience, and interest," in *Proceedings of the 10th European Conference* on Interactive Tv and Video, ser. EuroiTV '12. New York, NY, USA: ACM, 2012, pp. 153–156. [Online]. Available: http://doi.acm.org/10.1145/2325616.2325646
- [12] L. Barkhuus, A. Engström, and G. Zoric, "Watching the footwork: Second screen interaction at a dance and music performance," in *Proceedings of the* 32Nd Annual ACM Conference on Human Factors in Computing Systems, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 1305–1314. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557369
- [13] Nintendo. (2015, Mar.). [Online]. Available: http://www.nintendo.com/wiiu
- [14] —. (2015, Mar.). [Online]. Available: http://www.nintendo.com/3ds

- [15] H. Benko and S. Feiner, "Pointer warping in heterogeneous multi-monitor environments," 2007.
- [16] —, "Multi-monitor mouse," in CHI '05 Extended Abstracts on Human Factors in Computing Systems, ser. CHI EA '05. New York, NY, USA: ACM, 2005, pp. 1208–1211. [Online]. Available: http://doi.acm.org/10.1145/1056808.1056878
- [17] H. Debarba, L. Nedel, and A. Maciel, "Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision," in *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, March 2012, pp. 125–132.
- [18] T. Nishikado. (2015, Mar.). [Online]. Available: http://www.space-invaders.com
- [19] D. Dearman and J. S. Pierce, "It's on my other computer!: Computing with multiple devices," in *Proceedings* of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 767–776. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357177
- [20] Unity3D. (2015, Mar.). [Online]. Available: http://unity3d.com/
- [21] M. E. Holmes, S. Josephson, and R. E. Carney, "Visual attention to television programs with a second-screen application," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, ser. ETRA '12. New York, NY, USA: ACM, 2012, pp. 397–400. [Online]. Available: http://doi.acm.org/10.1145/2168556.2168646
- [22] J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen, and M. Czerwinski, "Impromptu: A new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development," in *Proceedings* of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 939–948. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357200