GAME DESIGN FOR WIRELESS DEVICES

Börje Karlsson
Danielle Rousy D. da Silva
Alexandre Damasceno
Informatics Center (CIn)
Pernambuco Federal University (UFPE)
Av. Prof. Luiz Freire, s/n, CIn/CCEN/UFPE, Cidade Universitária,
Recife, Pernambuco, Brazil
50740-540
E-mail: {bffk,drds,algd}@cin.ufpe.br

KEYWORDS
Game design, wireless technologies, software engineering.

ABSTRACT
The present article describes the current status of wireless games and goes on to discuss conceptual aspects of wireless game design, showing design "rules" and real solutions to try to solve the several restrictions that the wireless devices and technology present to the game developer. This work is based on the experience of a Research & Development team of which the authors are part and that has already produced several development frameworks and more than 20 games for various wireless platforms.

INTRODUCTION
With the growing market for wireless devices and the increase in memory and processing power as well as the growth of the gaming market, gaming has become a clear target for many manufacturers and developers. The wireless technologies exhibit some characteristics very handy to game development, as connectivity, interactivity and mobility. The key attribute in this environment is that it allows for the users to play anywhere and at anytime. This, added to the fact that thousands of people have access to wireless devices, opens broad perspectives to the mobile game development market.

However, there are some restrictions regarding the wide range of user profiles and technologies and hardware platforms. And these restrictions need to be taken into account during the design and implementation stages of a gem development.

This article has as objective, the discussion of the conceptual design of games for wireless devices using the experience of a Research & Development team of which the authors are part and that has already produced several development frameworks and more than 20 games for various wireless platforms. Probably due to the fact that these technologies are maturing, little is dedicated in the available literature to the study of game design in this environment, which is a crucial area in game development.

WIRELESS GAME DESIGN
When the first games for mobile devices were developed, they were made specifically to a single limited device, what restricted the game so much that only very simple games were possible. With the market growth and some technology maturity, the device manufacturers started providing some infra-structure (J2ME, BREW, Mophun, etc.) to support game creation by third parties. And that allowed for richer games, however, most games were (and still are) fairly simple, finding their inspiration in games from the early '80s, like early Palm games (Spronck 2001).

With the complexity growth and the completely interdisciplinary nature, digital games require a complex development cycle. The conceptual project of a game (also called game design) became one of the most important stages of a game project life cycle. This stage covers the definition of the game feature set as storyline; characters (physical appearance and psychological profiles); game goals; ending sequence, scoring systems; object hierarchy; game world rules and several other aspects. The important thing is that, at the end of this stage, all the game properties have been well thought and defined, including the user interface project, graphics, sound effects and soundtrack.

In a wireless device environment, the game design must be adapted to the imposed restrictions, regarding both the technology and the device features.

Nowadays, mobile handsets are very different from each other in several aspects, among them: screen size and shape, number of colors available, input mechanisms, development platforms, operational system (when available), available memory for storage and execution and processing power. Besides that, games can be developed using one of several available technologies (such as WAP - Wireless Application Protocol, SMS - Short Message Service, Sun’s J2ME - Java 2 Micro Edition, Qualcomm’s BREW - Binary Runtime Environment for Wireless, Sony’s Mophun and several others).

WIRELESS DEVICES
One of the first considerations to be made during the game design stage is what is the target platform for the game, a
cell phone or a bigger device? If it is a cell phone game, which set of handsets will be targeted? For even in the same series of devices by the same manufacturer, there are huge differences. Some of which, are extremely relevant to game development, as the number of available colors, display size, frame rate, key placement, processing power and memory capacity. Also the availability or lack of features must be studied (like image transparency for instance, that is not supported on all devices) and some devices restrict the maximum application size even if enough more storage space is available.

Most of the issues presented here, are focused on cell phones or devices with small screens; but every ‘rule/idea’ can be applied to several different devices. Trying to reach a good level of portability, a solution to this problem has been to specify a common denominator to all handset features. However, this approach presents some disadvantages as not allowing the use of some features available in more powerful devices. Another approach is to develop different versions for the different devices. A better approach is to start developing to the common baseline, and then incorporate device specific features.

GAME STYLES

Another relevant aspect is related to the game style, or game category, of the title that is going to be developed. Depending on the required resources and the resources available on the target devices (be they hardware or software), it could be impossible to implement the game maintaining a good gameplay. Our experience has shown that the best styles for cell phone games are: arcade style, as Space Invaders, Pacman, Breakout, Snake, Galaxian, Lode Runner, Seaquest, Frogger and others; or board games, like Checkers, Chess, Go; or card-games as Black Jack and Poker. The puzzle category is another one with a very good acceptance if the device restrictions are well handled. Good candidates are memory games, tic-tac-toe and the like. Figure 1 shows one classic game inspired game in Java 2 Micro Edition (J2ME) and Figure 2 shows two classic inspired games developed in J2ME and ported to BREW.

With the rise in processing power and display quality and size, it is also possible to enhance older games with new visuals and to develop larger and more complex games with levels and more features. Independent of the selected genre, it is always good to make the game as simple as possible. Figure 3 shows one such game, a racing game with several tracks and features, but still adequate to a wireless device.

DEVICE RESTRICTIONS

Some of the more important restriction imposed by the devices and that influence game design are as follows.

Controls

The available input mechanisms are a very important issue to be considered during the creation of a wireless game. Cell phones present a high variety of keypads and some even provide a stylus, as other wireless devices such as the Palm that have some similar problems (Spronck and Herik 2002).

One must remember that mobile phones were designed first and specially for voice telephony, and so were the controls on most of these devices (although that’s changing). Besides that, keypads are sometimes tiny and sometimes present a confusing layout; in some other cases, it is not possible to press more than one key at he same time. For example, create a Tetris like game where it’s necessary to move and rotate a piece at the same time (or a racing game where the player needs to press the accelerator and the steering wheel on a curve) can be very tricky.

In order to mitigate these issues, a good practice is to map some keys from the numeric pad to actions (as the 2, 4, 6 and 8 keys representing the directions). This practice came from the fact that some handsets had a very small size, and so, their directional keys were very close, what made it very difficult to users to control the characters on games. Another good approach is to not change the platform expected behaviour for a certain key (for instance, in BREW, the CLR key always cancels an action or erases a character on a text field).
One possible solution for games that require simultaneous key pressing (for example, moving the hero and shooting at the same time) is to specify one of the actions to happen automatically. As for example in PodRace, the player can choose if there will be automatic acceleration or if the player will have to press the “gas”; on the easier mode, the player most only control the vehicle wheel and the ship will break and accelerate automatically, but the user has the option to have manual control over breaking and accelerating if he wants that (making the game much harder) or if the device supports multiple keys pressed simultaneously.

This was also the approach used in Sea Hunter that follows the style of the classic SeaQuest, every time the diver gets in the confront area of the game, he starts to shoot automatically, freeing the player to focus on the diver movements.

During the development of a new game, a high priority concern is to try to find ways to simplify user interaction and improve gameplay. In the Tic-Tac-Toe game, for example, a mapping between the key pad and the game spots was made.

**Graphical User Interface**

Another problem found during the design of a game is the definitions of the images and graphical items to be used, especially because of size limitations for application storage, the display shape and size and the number of colors on each device palette. A good approach is modifying the graphics for different screen sizes, to help ensure an optimal user experience. Or in some cases, try to work around the limitations like, if all game action occurs in the center of the screen, try to center the image and clip the rest. For example, in GoldHunter, the action always happens near the screen corners, so the game level can be bigger than the screen in some devices and the screen scrolls, and the screen can be big enough for the whole map to fit on it at once.

An even more basic rule is to always focus on the device limitations. This means, don’t use too many or highly detailed images, and favor simplicity and clarity. Even when using color devices it is important to keep a small palette consistent between all the images.

Another problem is related to text, on some platforms, if you write a word to the screen with a certain font, this font can be different on another device, and the visual result may get compromised. Some times it’s better to have an image representing the text, but than this approach brings problems when one must have an application that supports several languages.

Regarding the game menus, one should avoid nested levels and complex navigation schemes. For example, in the PodRace game, the menu appears over the splash screen avoiding the presentation of another layer of screens and reducing the number of key presses for the user to start playing the game.

It is also good to emphasize that special care shall be taken to the user interface, providing the user a good degree of configuration and a consistent navigational model.

**Networking**

Networking and multiplayer gaming can add a lot to a game gameplay and replay value. It is possible to project a multi-user game for wireless devices (either using SMS messages or protocols such as HTTP) if the developers consider workarounds for issues as high latency and low data transfer rates, which if not addressed can have the opposite effect and lower the gameplay and replay values. Good games genres that could make use of networking and avoid these issues would be turn-based games and strategy games. Or yet, one could incorporate these usually undesired network behaviours into the game storyline/environment.

Another possibility to make use of the networking features available is to create servers that host score rankings or even allow the downloading of new levels or characters to a game.

**Sound**

Sound support is very different from device to device, and some (especially older devices) don’t even provide support for this feature, so it is another point that deserves special attention during development. Some handsets allow for one single sound, some for a sound track and some synthesized sound effects to play concurrently, and some even provide MP3, WAV and MIDI support. But the game developer must take care not to over use this feature, because repetitive sound can get boring and as the player can play games anywhere sound may be inconvenient in some situations, especially as some devices do not provide enough volume control.

**Interruptions**

As stated before, mobile phones (and most wireless devices) were designed first and foremost for voice telephony (or some sort of office assistants), so it is important to guarantee that the device can be used the way originally intended and that the game can handle interruptions gracefully, be they from a phone call, a message arriving or a simple alarm clock ringing to warn a user of a corporate meeting.

**GAME DESIGN RULES**

One should always remember that even with a quite complex game in mind, in this context presented here, it is targeted to a mobile device (usually a cell phone); consequently the game should be kept as “casual” as possible, where accessibility is favored instead of deepness and immersion; focusing on minimizing player frustration.

Game design is mainly focused on player actions, fundamentally, when playing a game; a player takes actions that cause changes in the game state; the game world receives these actions; and is updates accordingly. Games are a mix of struggle and effort. That is, a game that is too simple gets boring; and a game that is too difficult is frustrating to the player. Users usually enjoy games that challenge them with problems that they can
overcome. There are typically three types of challenges: physical, mental and opponents. The physical ones are the ones related to physical abilities, like, how fast someone can press a key several times, or how quick are his reflexes. The mental ones are the ones related to puzzles and memory. And the “opponents” category is the one provided but the game AI or by other players in case of a multi-player game.

A good and entertaining game usually consists of a combination of those kinds of challenges. The game should also be structured in a way that makes easy for the developer to adjust de difficulty level (or even, the game could have several difficulty levels) and change game design to balance and balance the game.

Apart from the game challenges, there are several “rules”, or rules of thumb, both collected via our experience or from sources as (Crawford 2003), that provide insights and guidelines for the design of new games (wireless or not). One of such rules and a very powerful one is to always try to build a community around your game, so that your game will benefit from the increased replay value; for example, a racing game like PodRace, could easily be extended to support a server that could provide new race courses and vehicles and that could store user created tracks allowing users to exchange files and “improve” their games, and this same server could host a ranking, what would improve competition on the game. This solution could also be faced as changing a single player game into some sort of single-player multi-player game, were even with single-player games, several players would “fight” each other to win, increasing the “opponent” category of challenges.

One other rule, more specific to the wireless games world, is that as cell phones are a platform for “free-time” gaming (gaming is not the main focus of the devices), the player will usually play for short amounts of time. This shall be taken into account in the game design, and the game should provide short term goals for small playing sessions and longer term goals for longer sessions and once these long term goals are implemented, it would be nice to have some way to store game progress so that the player does not have to begin everything again every time he wants to play, or every time the phone receives a voice call and suspends the game (this was implemented in PodRace, the player could choose to race one course in the Single Race option, or could choose to race the whole championship at once; and in this case, the game state would be saved between each race course).

A very interesting project that deals with “game design rules” is the 400 Project (Falstein 2003), where a set o 400 game design rules is being collected. Although still at the beginning, the project already offers some insights into game design issues. Some of the rules that most fit a wireless gaming environment (and our experience interpretation based on our experience) are:

- Turn Constants into Variables: like the physical modeling in a game does not have to be realistic, one can use constants and several simplifications, but the user must think it is real.

- Fight Player Fatigue: especially true in the wireless world, the game must always present challenges to the players but without frustrating them.

- Maintain Suspension of Disbelief: true for any game, even in mobile devices were it is difficult to have player immersion; the games must focus the player attention and games simplifications or platform restriction should not harm the gameplay.

- Make Sub-games: accomplished by providing the ability to play just a little game, like racing a single race in PodRace.

- Provide Clear Short-Term Goals: somehow intersecting with the above one, but not equal. Present even on a single race, passing a opponent is a shorter term goal.

- Let the Player Turn the Game Off: Both to be able to save game state and allow the player to resume the game and to be easy to exit the application.

- Identify Constraints: this one is the main focus of the article; identify the constraints and work with and around them.

- Provide an Enticing Long Term Goal: this one complements the short term goals one.

- Make the First Player Action Painfully Obvious: also especially true for the wireless gaming world, the game must not require that the player read a manual in order to start playing. It may be that if the player reads the manual he will have access to more features, but it must be obvious how to start playing.

- Keep the Interface Consistent: already commented on the User Interface restrictions.

- Create AI in the Mind of the Player: same trick as the Turn Constants into Variables one, one does not need a real AI to make the game fun; little tricks can do the magic (as real AI can also).

- Don’t Penalize the Player: could be stated as Reward the Player, the user is playing to have fun, so give him an entertaining experience.

- Make the Game Fun for the Player, not the Designer or Computer: this one works closely with the previous one; of course it is fun to develop games and developers want their favorite features into games. But the end user is the game target and especially with the different profiles of wireless gamers, their interests will probably be different from the developer ones.

An important artifact and guide through the game development process is the Game Design Document. Often neglected, especially in smaller games like the ones commented here, it is essential to record design decisions and provide a unified view of the project to the whole team. It should start as general design thoughts and be refined over and over again until a good foundation is reached. A good way to start would be to discuss what could be present on the game (the possible), what must be present (the essential) and what the developers would like to see present (the desirable).
CONCLUSION

Wireless games were once a niche, but are now becoming a huge growing market, and constitute a great area to be explored by both commercial developers and academics. Offering great opportunities in technology experimentation and game development techniques, this kind of game requires an additional effort to take into account the different wireless devices user profiles and different device restrictions. By one side, more intuitive games (but with high game play) are required. And from the other side, the range of target audiences grows, as for example, games for company executives.

Several rules of thumb used in order to help on the design of new games for this environment and the importance of a game design document documenting the decisions made, were presented.

But the strongest issue to be concerned with is that wireless games can run a variety of platforms and devices with different and restricted resources, each one having to be considered. And also an important point is that every platform require constant tests on the devices and not only on the development kit emulators, in order to guarantee that your design decisions work across several devices.

REFERENCES


AUTHOR BIOGRAPHY

BÖRJE KARLSSON is a Student Researcher at Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE) in the field of Artificial Intelligence and is currently working in projects at CIn/UFPE and C.E.S.A.R, related to development and testing for mobile devices. Börje has a BSc in Computer Science and is finishing a specialization in Software Engineering at CIn/UFPE.

DANIELLE SILVA is a PHD Student at Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE) in the field of Artificial Intelligence and is currently working in projects at CIn/UFPE and C.E.S.A.R, related to development and testing for mobile devices. Danielle has a BSc and a Msc in Computer Science.

ALEXANDRE DAMASCENO is a Masters Student at Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE) in the field of Artificial Intelligence and is currently working in projects at CIn/UFPE and C.E.S.A.R, related to development and testing for mobile devices. Alexandre has a BSc and is finishing his Masters degree, both in Computer Science.