

'Balls of Steel'

Group 4

John Laurin

Joakim Åkerlund

Milan Ivanovic

Daniel Öberg

Christoffer Lundell Johansson

Project Overview Document

1. *Intended Users and Specification of Purpose*

The intended demographic of users for this project are men and women between ages 10 and 30 (approximate), which have access to a cellular or mobile phone supporting the Java Micro Edition Platform. The goal of the project is to produce a mobile phone game which is to provide a recreational distraction relieving tediousness. The game is aimed at casual players and is intended to be easy to access and quick to start playing, requiring no long start up or configuration.

2. *System Use Specification*

The game shall take the form of an arcade-style scrolling platform game on mobile phones supporting Java Micro Edition. Your avatar in the game takes the shape of a steel ball, which is in a perpetual state of motion. The default state of the ball is that it is in a perpetual bounce with no lateral motion whatsoever. The balls momentum does not decrease as it bounces.

The player can induce lateral motion by using the default left and right buttons on the phone, as defined by the manufacturer. This motion is imparted on the ball in the form of spin in the appropriate direction. The amount of spin changes the vector of the ball, as well as the height of the bounce.

The goal of the game is to survive for as long as possible. The game ends when a global countdown reaches zero. Time can be gained by collecting power-ups throughout the levels. A level is completed when the player collects a number of keys distributed across the level and reaches the predetermined exit point. On completing a level, an amount of time is added to the global countdown.

In addition to the keys, there are a variety of enemies across the map. These can be both static and dynamic, with the unifying factor that if the player hits one of these, the level restarts and an amount of time from the global countdown is subtracted.

In addition, there are objects placed around the map that alters the position, trajectory and other aspects of the ball, often being combined to create puzzles for the player to solve.

When the countdown reaches zero, the game is over, and the total time played is displayed. The 10 highest times are saved on the High-Score, the viewing of which is a menu option.

Use-Cases:

1:

Daniel takes the subway to school. This journey takes him approximately 15 minutes. On this day however, there are major disturbances in the subway-system, resulting in the trains being delayed by 30 minutes or worse.

After 14 minutes of waiting, Daniel finds himself with nothing to do for another 20 minutes. He therefore decides to play some 'Balls of Steel', which he recently obtained for his mobile phone. He starts up the game, arriving at the main menu. Since he hasn't played the game before, there is no option to continue a previous game. So he then selects 'New Game'.

Daniel performs rather well, and the train arrives before the game ends. He then presses the 'back'-button, sending him back to the menu, whereby he selects 'Quit', and the game shuts down. He is not worried, because he will be able to resume the game at another time.

2:

Daniel is now in school. He is now at a lecture which is very tedious. At the break he decides to continue playing 'Balls of Steel'. He starts up the game, arriving at the main menu.

He has his game from earlier that day, so he decides to continue playing that. Upon selecting the option 'Continue', the game starts in a paused state, allowing Daniel to appraise the situation prior to starting. He then presses the left key to start the game proper.

However, due to the lecturers dry presentation, Daniel's waking state is somewhat precarious. This lowers his performance in the game. Within seconds he hits an enemy, throwing him back to the beginning of the level, and subtracting time from the countdown. This occurs several times, and after a couple of minutes he hits an enemy that results in his countdown reaching zero, whereby the text 'Game Over' appears, and his total time played. This time is not high enough to enter the high-score, and he then returns to the main menu. He doesn't feel like playing anymore, and selects 'Quit', whereby the program exits.

3. *Contextual and Environmental Specification*

The game is meant as a recreational and fun way to spend some extra time and, since the game is to be run on the mobile platform one of the greatest advantages is that it can be accessed from basically anywhere. The only requirement is a mobile phone with support for Java ME (which the majority of the mobile phones on the market have support for).

4. *Scope of the System*

Since the system we are developing is a game intended for the mobile platform the scope will be fairly limited. The core purpose is to entertain the user of the game, and the game will be quite simple since it is for mobile use and also we don't have that much experience nor time.

<u>Topic</u>	<u>In</u>	<u>Out</u>
Support single-player	X	
Have graphics	X	
Have Sound	X	
3D-graphics		X
Support for multiplayer		X
Support saving system/high score	X	
Platform independent (Java ME)	X	
Different resolutions	X	

5. *Design and Building Considerations*

'*Balls of Steel*' is a game developed for the mobile platform using Java ME. Its primary function is to entertain its users by providing correct output (graphics, sound) on the user's phone and to be able to correctly take input that the user makes (key-presses) and implement it as stated in the requirements. It will be easy for a user to learn about the rules of the game and start playing without spending too much time reading the documentation. It will support a wide range of mobile phones that can run Java programs and will be able to run efficiently on them without compromising the phone's other functions. Factors that should be taken into account when developing such a system are ;

- What happens if the game is forcefully exited? When you start the game again you should continue from the last save, but the forceful exit will in all likelihood have bypassed the saving procedure. Therefore the game should not overwrite or remove the save-file when it is loaded.
- Support for different resolutions for different mobiles. Ideally the game should offer an adaptive environment based on the aspect ratio of the current mobile phone.
- Easy to start and learn how to play. Easy to quit and continue later.
- If someone calls while you are playing the game it should pause and minimize (if minimize doesn't work it should save the state and then exit)
- Handling of telephone events that are not immediately critical, such as the arrival of an SMS, a battery warning etc. The game should not be exited in such cases, and should simply pause while the user decides what to do about them.
- The game must be efficient as the system resources on mobile phones may be very limited. Graphics is the primary consideration in this regard.

6. *Technologies and Risks*

The tools we will use are Bazaar as our distributed version control system, Netbeans as our integrated development environment, 3D Studio Max and Adobe Photoshop for our visual content.

The technologies we will use are all included in Java ME (or more specific JSR 118/MIDP 2.0). The risks are first and foremost that almost no one in the group is familiar with Bazaar (and other version control systems for that matter). Not understanding the version control system can lead to delays and in the worst case lost work/code. In addition, since we are using a private server to store the code, there is a risk of data-loss if that server goes down. Although, as the option to store new versions locally is available, the risk of a total data-loss is significantly reduced.

We are all familiar with Netbeans, so no significant training-time should be necessary.

The most difficult item will be learning about Java 2D API and the physics that will be used in the game. Java 2D is somewhat notorious for its inefficiency, and if it is not implemented well enough it could lead to the game being unplayable on certain mobiles. The physics component are simply a series of algorithms which should be relatively simple in theory. Implementation on a practical level using as few resources as possible may be an issue.