

# ECE1778 FINAL REPORT - RUNPG

Heyse Li: 995156879  
Matthew Ma: 994639030  
Word Count: 1999

2012-04-12

# 1 Introduction

The goal of our application is to help users maintain a regular cardiovascular exercise regimen. Cardiovascular exercise is essential as it is one of the key factors in preventing cardiovascular disease. It is recommended that an individual should perform 30 minutes of vigorous exercise, five times a week. However, many people struggle with maintaining a consistent exercise routine because the rewards are not immediately tangible.

Our application, RunPG, was designed to overcome this problem by combining a role playing game (RPG) with running. People would be motivated to do more exercise as they constantly receive virtual rewards. The player runs to build up a bank of steps and spends these steps on in-game actions. The concept of the game revolves around exploring a game world to collect resources, upgrade the player character, and delivering resources to the player's village.

## 2 Overall Design

The overall design of the application is broken down into four major subsections as shown in Figure 1: the character manager, pedometer, game manager, and remote server.

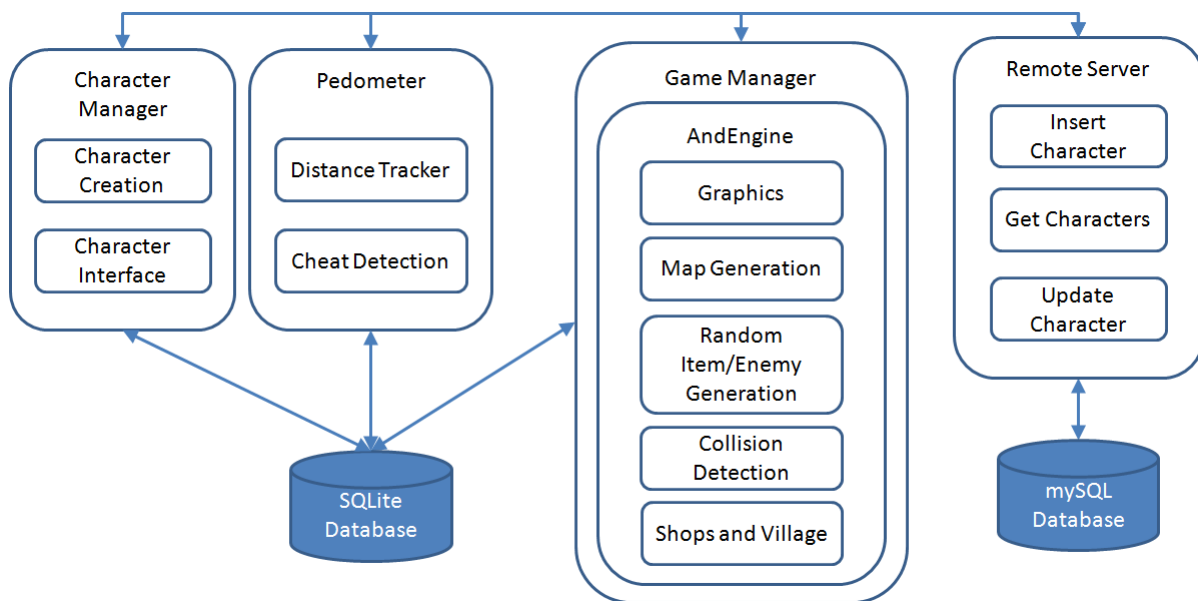


Figure 1: Block diagram of major code functionality

### 2.1 Character Manager

The character manager allows the player to create a new character and to manage their character. The basic information of name, age, weight, and height cannot be changed once

entered. While running or playing the game, the player can access their character to manage their attributes, inventory, equipment, view personal run statistics, and to view global player statistics.

## 2.2 Pedometer

When the user decides to go on a run, the Distance Tracker detects each step taken. It uses this information along with the estimated stride length to calculate the distance and speed. Also, it updates the number of steps in the steps bank and keeps track of the total taken since character creation. At the end of each run, levels or achievements may be granted based on the total steps and distance ran.

The distance tracker code utilizes code from an open source pedometer application, which was obtained from: <http://code.google.com/p/pedometer/> under the GNU General Public License v3. However, the code had some limitations; the step detection was not very accurate and was too sensitive. Therefore, it had to be modified to meet the requirements of our application.

To reduce the occurrence of cheating, acceleration in the x,y, and z axes were analyzed for walking, running, and shaking motions. Through investigation and testing of various methods for detecting cheating, it was discovered that the amplitude of the acceleration vector could be used to roughly distinguish between these three motions.

$$\text{Amplitude} = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

The results were graphed, as shown in Figure 2, Figure 3, Figure 4. It can be observed that the shaking motion had the most fluctuation in amplitude, with points greater than  $22 \text{ m/s}^2$ . On the other hand, the points of running remained relatively constant at around  $19 \text{ m/s}^2$ . As a result, any amplitude values over the limit of  $19.7 \text{ m/s}^2$  would be considered as invalid and would not be accumulated in the steps bank. The Cheat Detection component also detects rapid shaking and shaking while the phone is parallel to the floor.

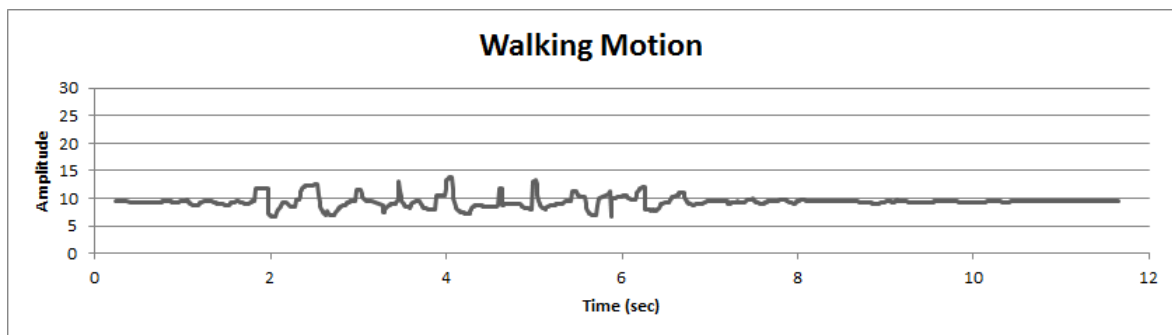


Figure 2: Walking motion

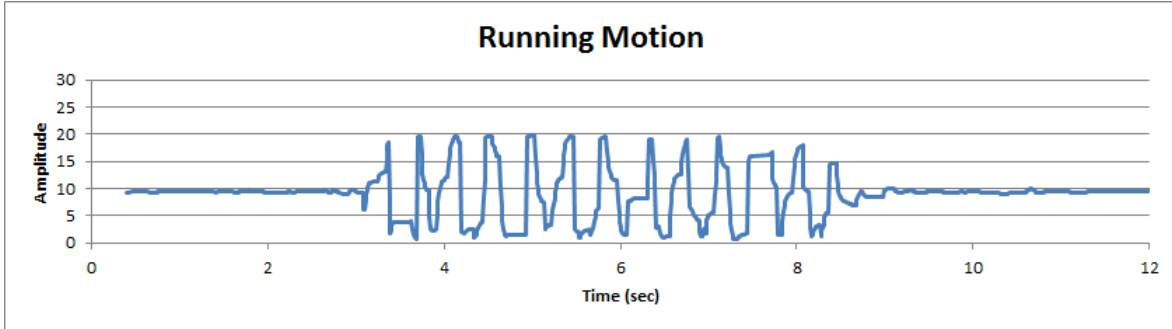


Figure 3: Running motion

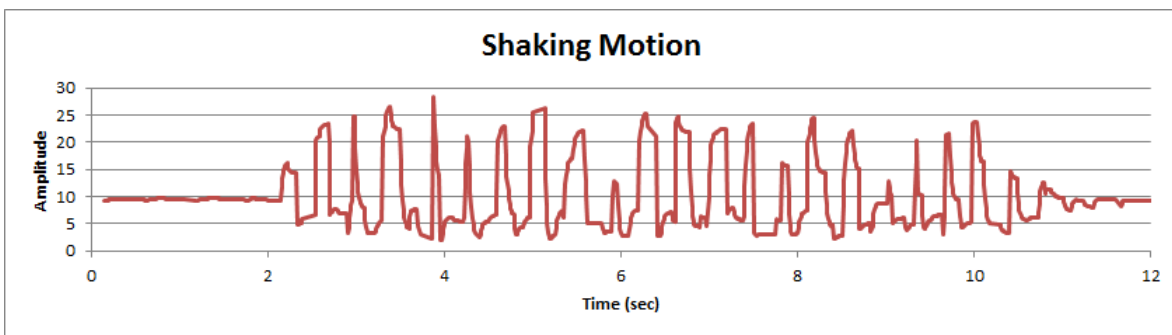


Figure 4: Shaking motion

## 2.3 Game Manager

The graphics portion of the code utilizes an open source 2d game engine library called AndEngine under the GNU Lesser General Public License. The engine is written on top of OpenGL ES2.0.

When the game is loaded, the character's previous position, and revealed tiles are read from the SQLite database. The maps are read in from an XML file. The sprites used were obtained from <http://www.lostgarden.com/2007/05/dancs-miraculously-flexible-game.html> under a Creative Commons Attribution 3.0 license. The character is controlled via an on-screen digital joystick.

Random items and enemies are generated and placed around the map. The number of items and enemies are dynamically generated are based on the character's level. The difficulty of the enemies encountered will increase as the character grows more powerful.

Whenever the character moves the collision detection code will check what type of sprite the character sprite has come into contact with.

- If the sprite is Walkable, the character can move across it.
- If the sprite Blocked, the character cannot move past it.

- If the sprite is an Item, the character can pick it up.
- If the sprite is an Enemy, the character can attack it.

The player can visit shops to buy and sell items to earn money. The player can also return to their village to drop off any collected goods to feed their village and increase the population.

## 2.4 Remote Server

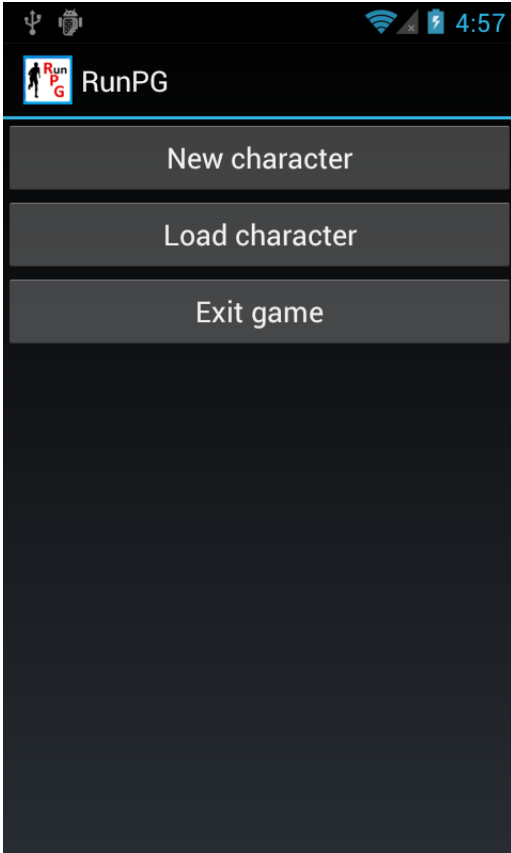
An Apache server running a MySQL database has been set up to host a global collection of player statistics. The server is contacted in the following situations:

- At character creation, the application will contact the server to check if the new character's name already exists. If it does not exist in the master database, the character will be created locally on the player's phone and also on the server.
- When the player begins the game, the application contacts the server to download the global list of players and their current level. This will allow the player to compare their progress to others.
- When the player finishes a run, their updated level and step count statistics are uploaded to the server.

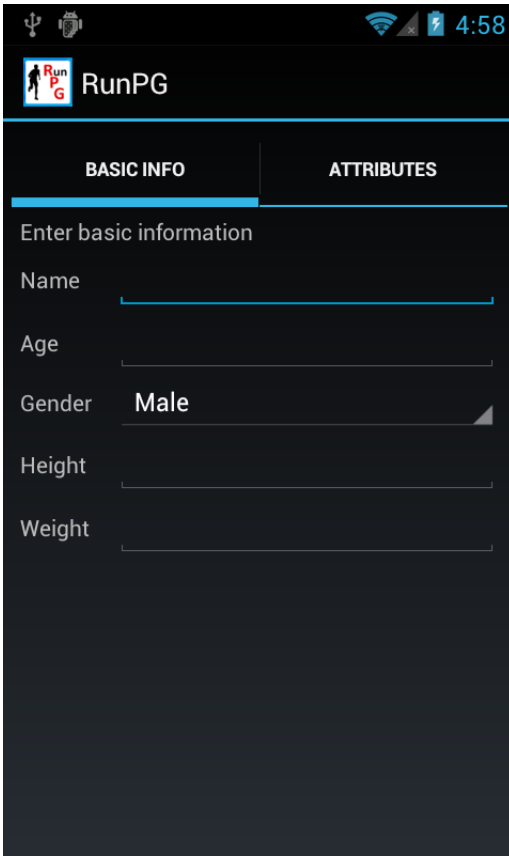
If it takes too long (10 sec) to contact the server, the operation will time out.

### 3 Statement of Functionality

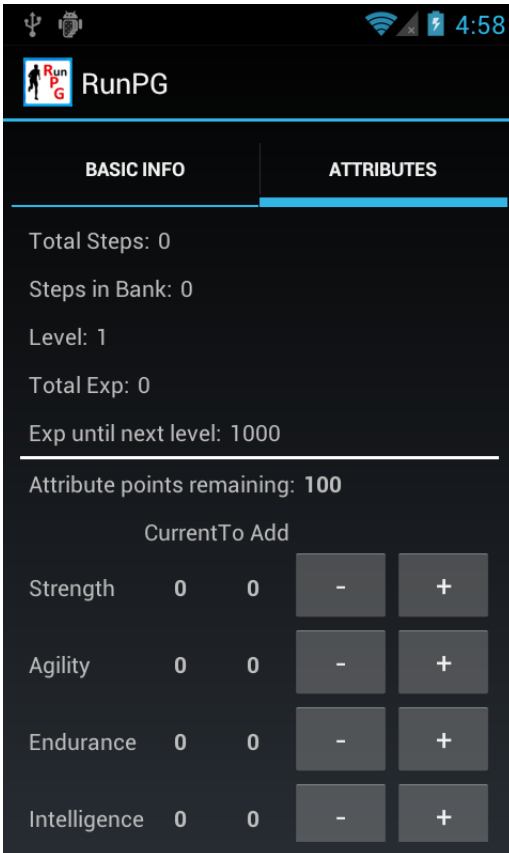
#### 3.1 Functioning Elements



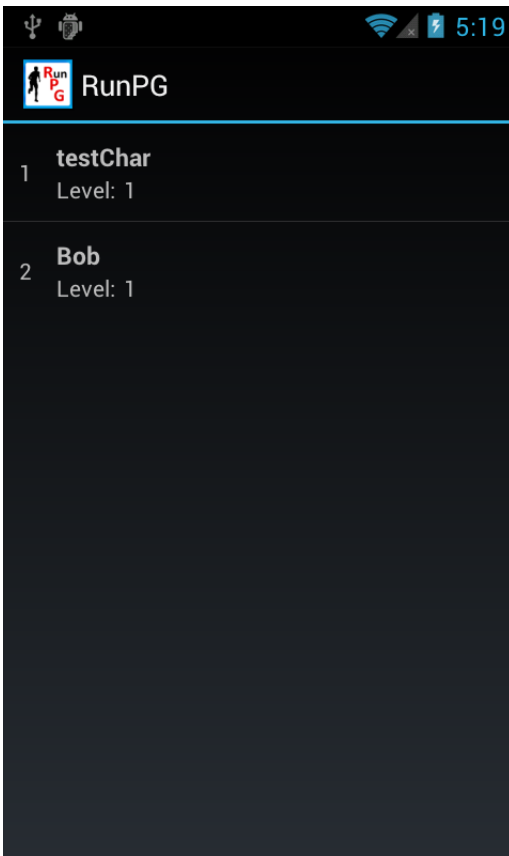
Upon opening the application, the player is presented with options to create a new character, load an existing character, or exit the game.



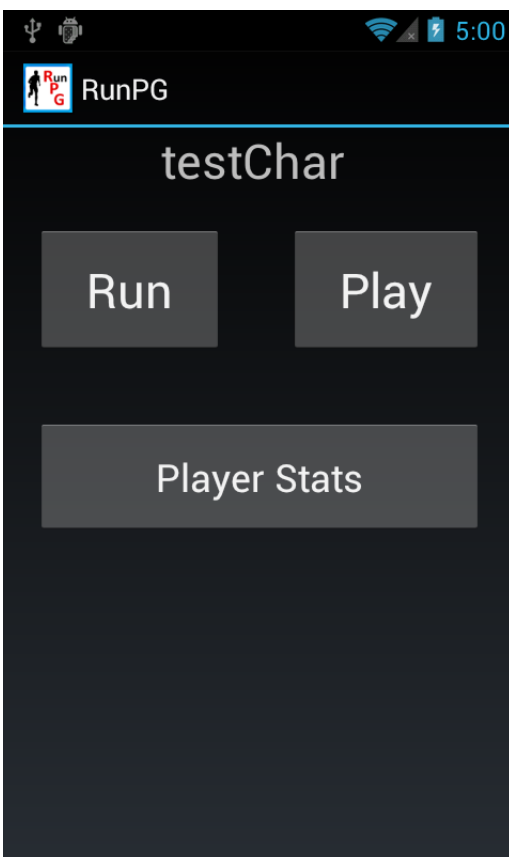
If the player chooses to create a new character they are prompted to enter the basic information of their character. Age, height, and gender are used to calculate stride length based on population statistics.



The player can decide to allocate attribute points to their character during creation, or they can do it any time during the game. Strength allows the user to deal more damage to enemies and also spend less energy (banked steps) during combat. Agility increases the number of banked steps the player can use. Endurance allows the player to carry more items in their inventory. Intelligence allows the player to extend their sight radius and also gives them better deals at the shop.

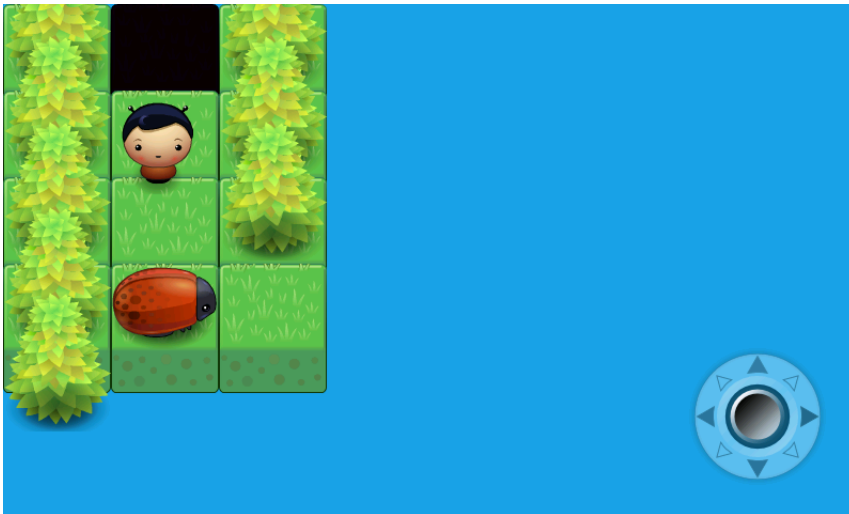


The player can select an existing character to load.

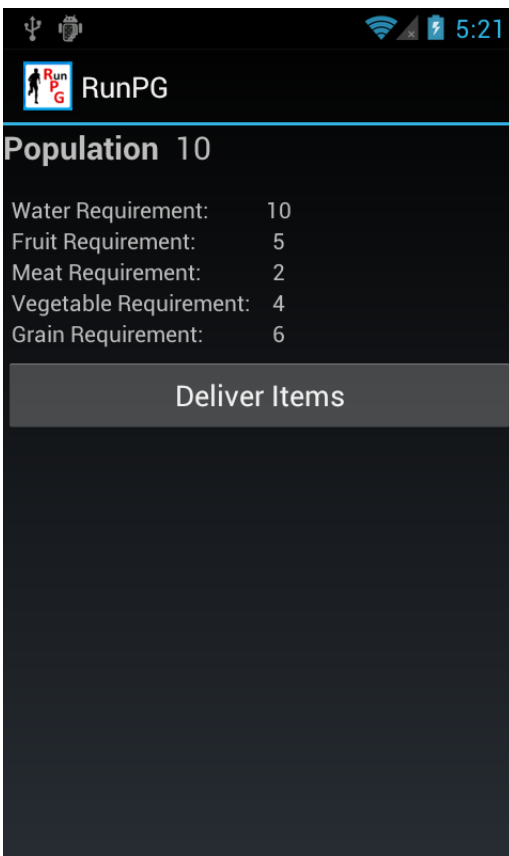


After creating a new character or loading an existing character, the player can select to go on a run and increase their banked steps which will allow them to move their character and attack enemies in game. The player can also select to play the game or manage their character.

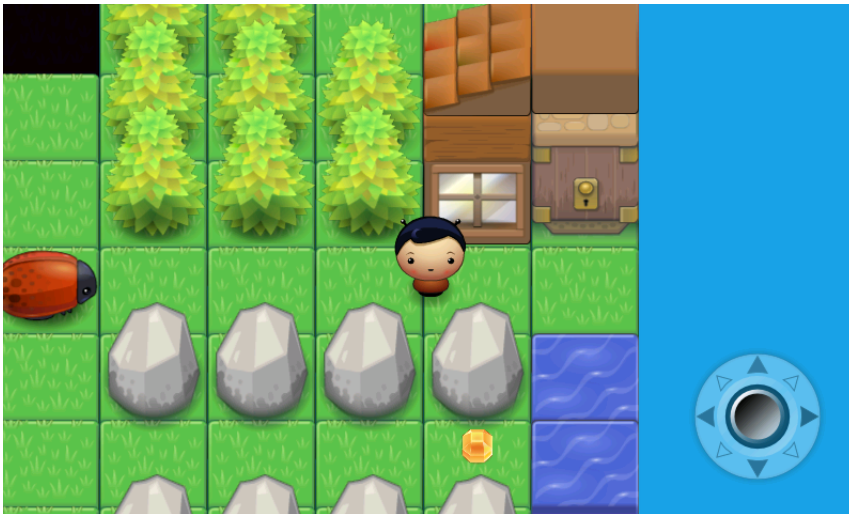




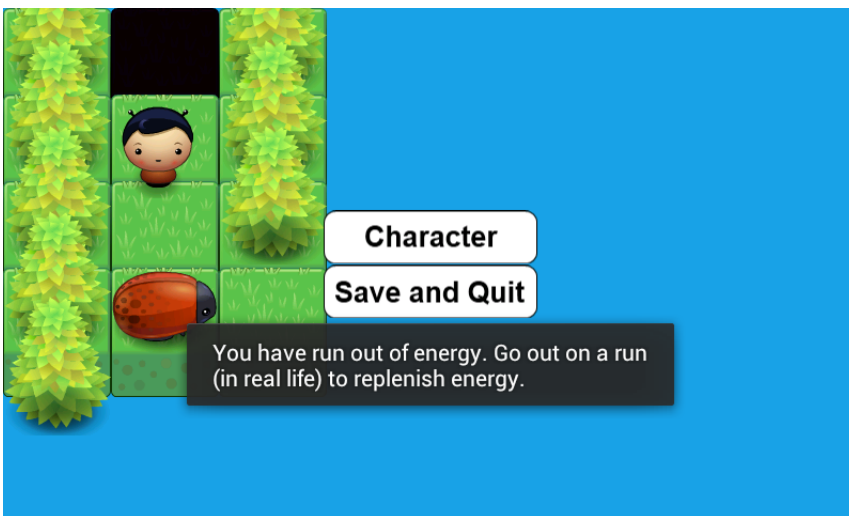
This is what the player sees upon playing the game for the first time. The darkened tile behind the player allows access to the player's village.



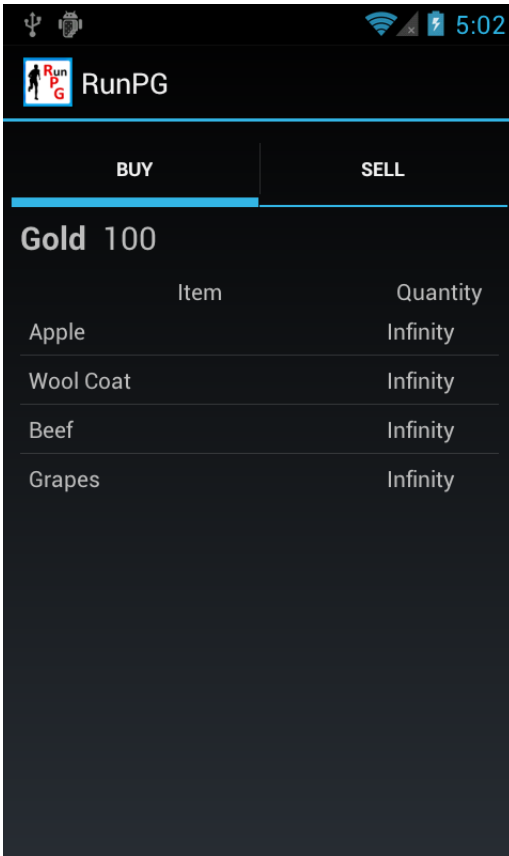
The resource requirements will increase each day proportional to population. If the player manages to reduce all the requirements to 0, the population will increase.



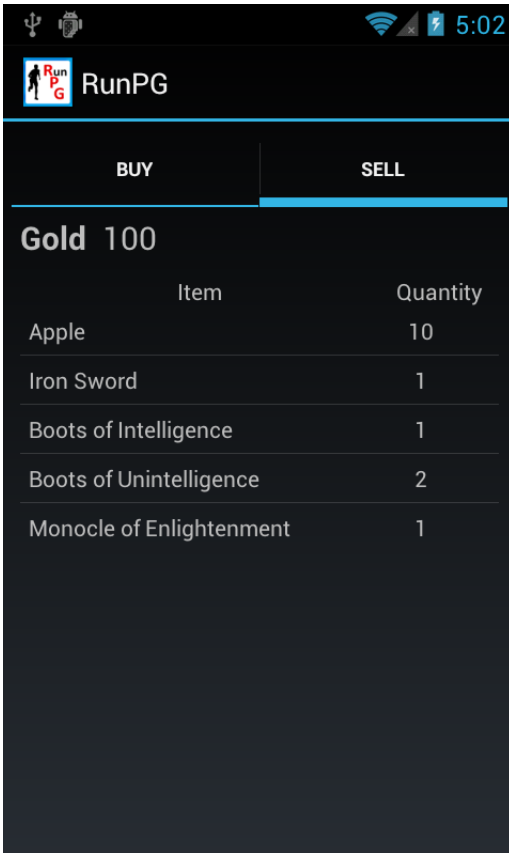
As the player moves around the map, they reveal tiles hidden by the fog of war.



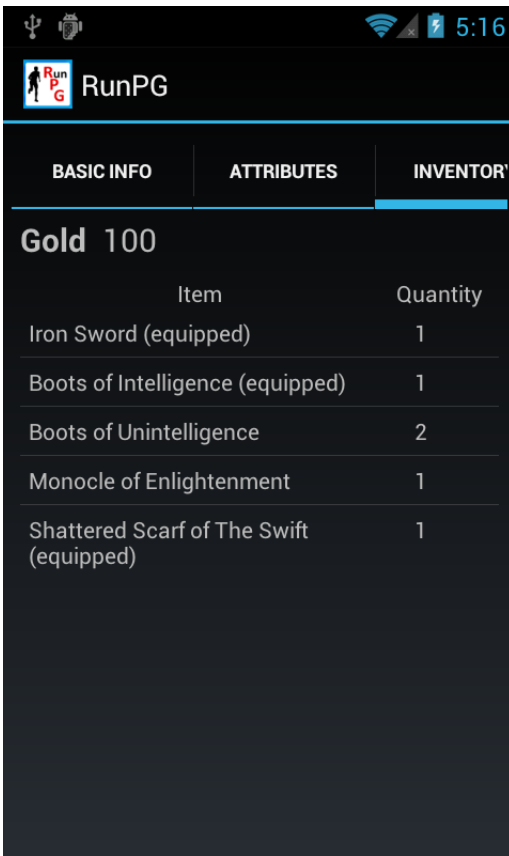
However, exploring the map costs energy and when the player has expended their energy, the game prompts them to save and go out for a run. The menu screen can also be accessed by pressing the phone's menu button. Selecting "Character" lets the player view and manager their character, and selecting "Save and Quit" will save the current status of the map and return to the main menu.



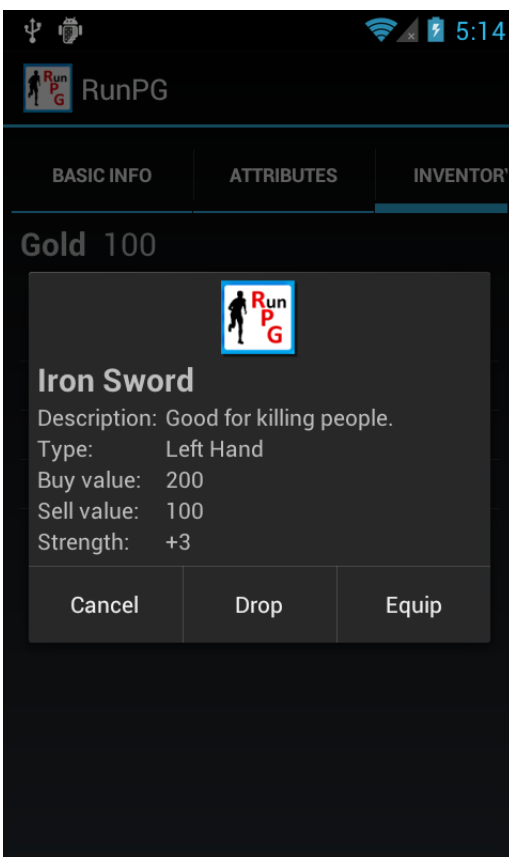
The building the character discovered is a shop where the player can purchase items for sale.



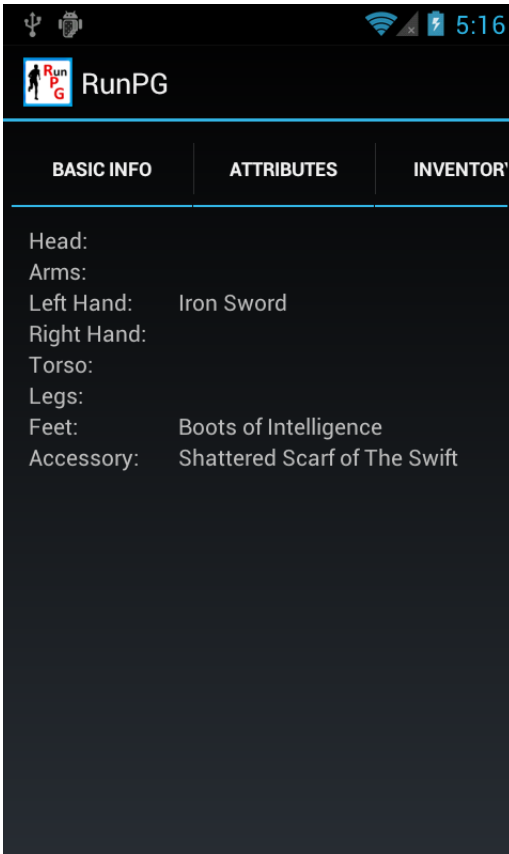
The player can also sell their items in inventory.



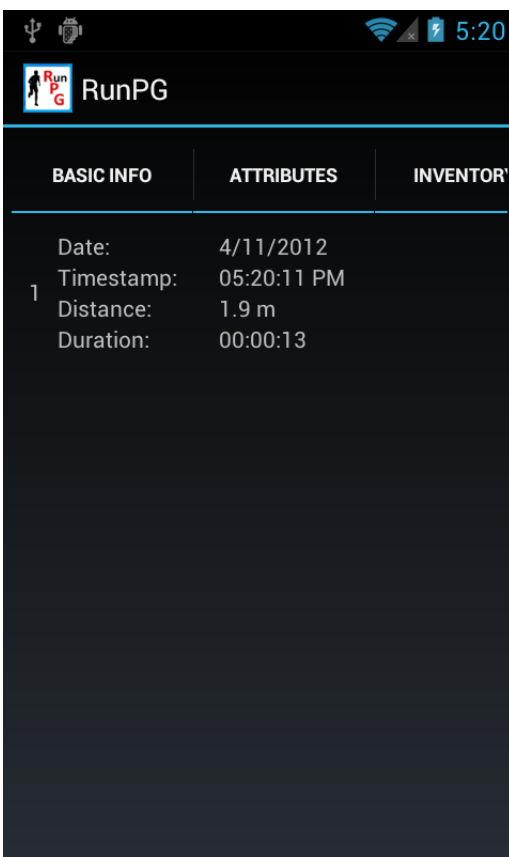
Here the player can equip certain items in their inventory to enhance their attributes.



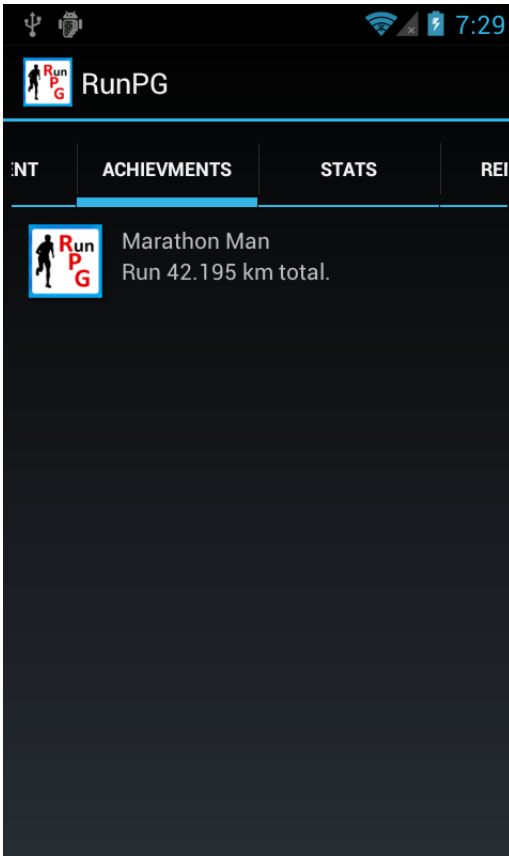
When the player selects an item in their inventory a dialog displays that item's information.



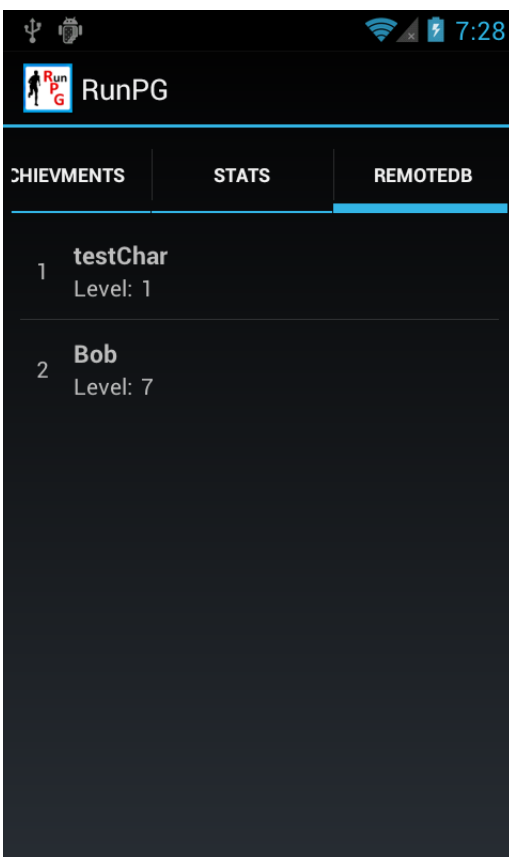
The player can view the items they currently have equipped.



The player can view statistics about previous runs.



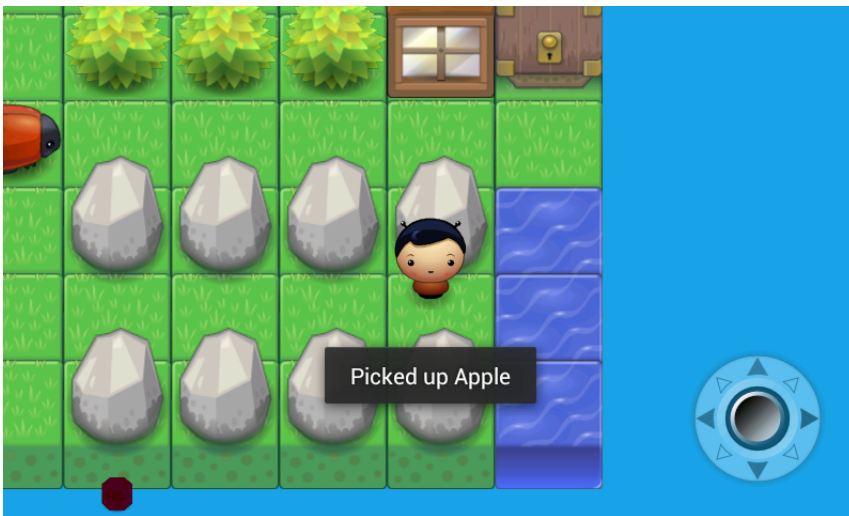
The player can view the achievements they have obtained.



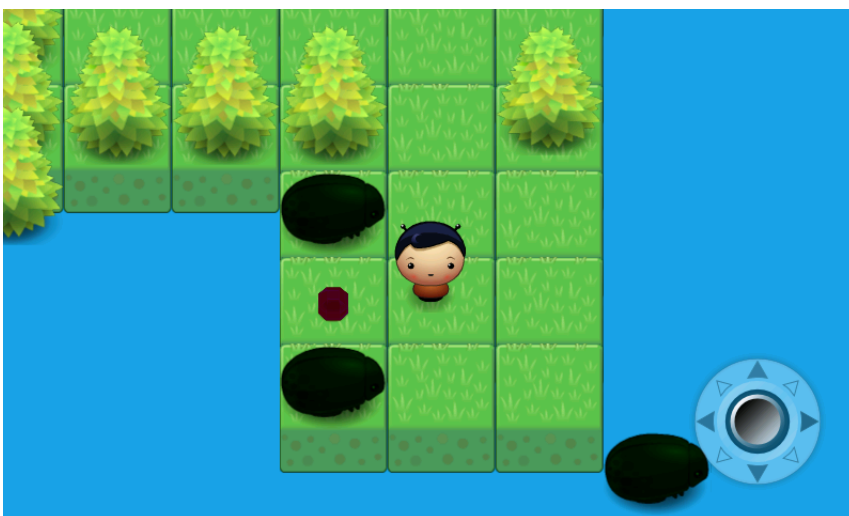
The player can view where they rank on the global player list.



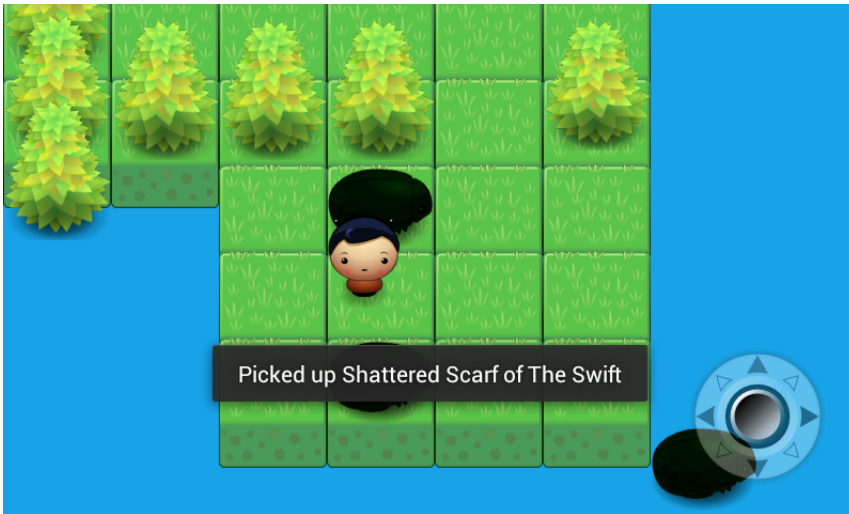
The player can attack on screen enemies if they are within range, by tapping on the enemy. The game will display the enemy's remaining hitpoints (HP) and the player's remaining energy. Each attack costs a certain amount of energy.



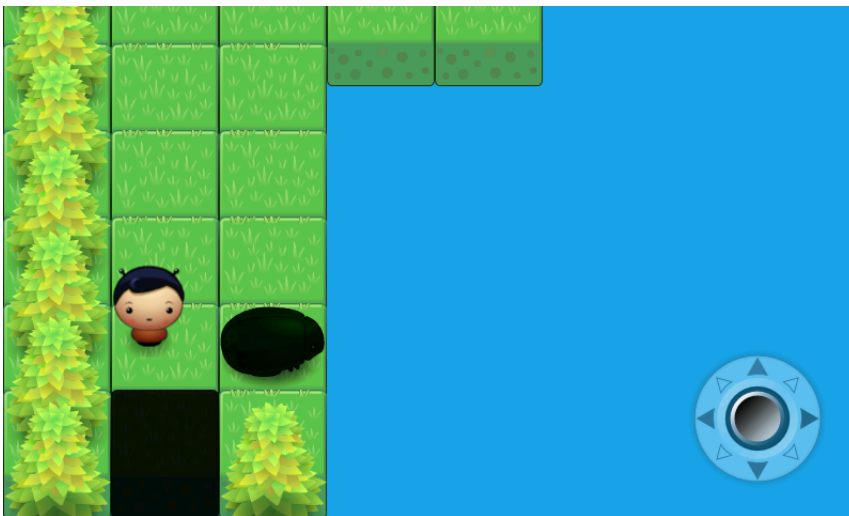
When a player walks over an item they can pick it up, provided that they have sufficient space in their inventory. If they do not have space, they can drop some of their currently held items.



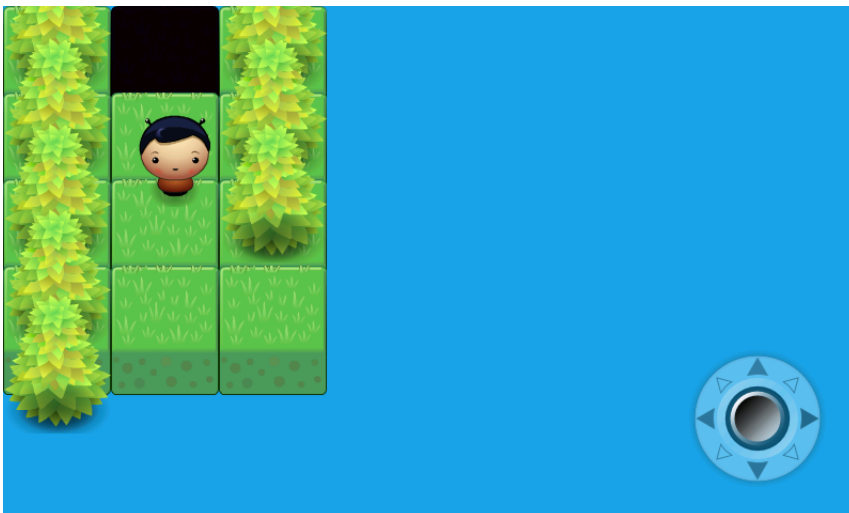
The player has encountered some random items and enemies.



The player has picked up a random item. The item's name is generated randomly.

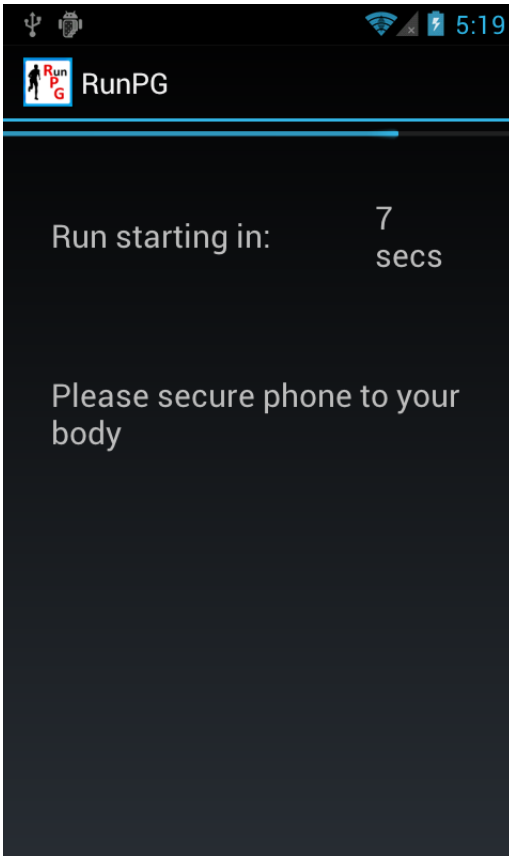


The player has found a pathway to the next map. During development it was found that as the map size increased, the game's lag would increase. Therefore, the game world was split into multiple smaller maps to increase the speed of the game.

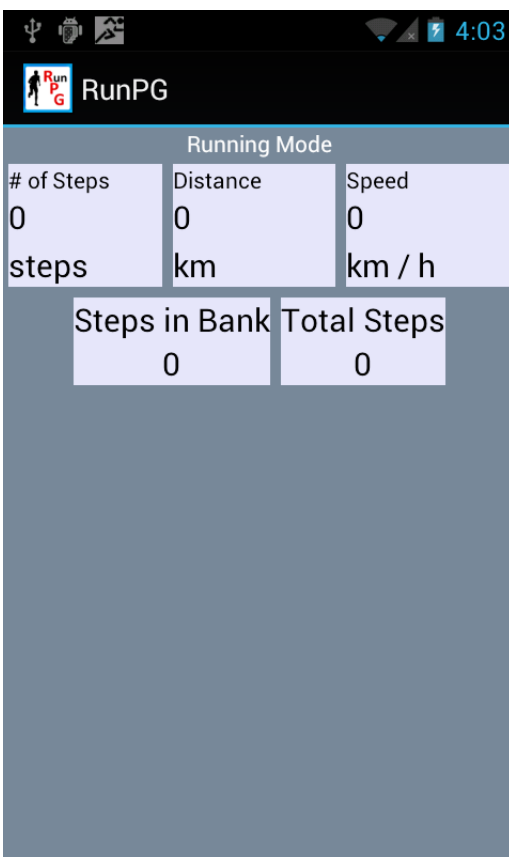


The player has moved from the first map to the second map. They can return to the first map through the darkened tile.





The player has decided to run. The application gives time for the user to secure the phone.



The pedometer shows the player banked steps and total steps taken since character creation.

## 3.2 Non-Fully Functioning Elements

The collision detection to prevent players traversing blocked tiles does not currently work properly. The player can sometimes traverse blocked tiles by moving in a certain fashion.

When the player attempts to move into a blocked tile, the game will detect that and will move the player back with a negative velocity. However, if the player is close enough to the blocked tile, and they try to move in the opposite direction away from the blocked tile, the game will think the player is actually within the blocked tile and move them into it, and this will allow them to traverse the blocked tile.

## 4 Key Learnings

Through the course of this project, 3rd party libraries should have been used earlier, as they can simplify complicated tasks. The graphics took longer than expected as we first attempted to develop the game graphics using OpenGL. Since no one had previous knowledge of OpenGL, it proved to be difficult.

In addition, the game mechanics could be improved. If game is not fun, players would not want to use our application. Therefore, it is important to have well designed game mechanics so that people playing the game can have fun and be engaged.

Developing a robust cheat detection mechanism is not easy. It requires complex analysis of movement patterns using the phones accelerometer. Noise filtering would also be needed to ensure that only useful data are processed. Extensive time was invested in testing algorithms for improving cheat detection to cover various cheating scenarios. Our cheat detection could be much simpler if the application was limited to be used outdoors and used GPS.

## 5 Contribution

### 5.1 Heyse

- Character manager, game manager, graphics, server, database, application design, UI design

### 5.2 Matthew

- Distance tracker, cheat detection, character statistics, code integration, logo design, application design, UI design

## 6 Future Work

- We could optimize our code to improve the response time of our application.

- The fog of war could be improved with a dark overlay with gradients that reveals the underlying map.
- Implement a player vs. player mode. This would add more social elements to the game.
- Instead of using open source sprites, we should develop our own sprites to enhance the overall look of the game.
- We could add more content to make the game more interesting.
- For cheat detection, we could use the light and camera to detect the player's pulse.

## **7 Business School**

The development team does not wish to have a business school class take up the application.