GPS Game
Software Requirements Specification
COP 4331C Processes OO Software Fall 2011

Team 14 - Team (Cauc)asians

Team Members:

- Bernard Feeser
- Jon Leonard
- Danh Nguyen
- Jolene Wan
- Juan Chen

Modification history:

Version	Date	Who	Comment
v0.0	08/15/2000	G. H. Walton	Template
v1.0	09/22/2011	Bernard Feeser	Updated template with names, sent to others
v2.0	10/14/2011	Jon Leonard	For first set of deliverables

Contents of this Document

Section 1: Introduction

Software to be Produced:

- The software to be developed is a mobile device based game. It will utilize the technology commonly implemented on modern phones such as the GPS and the gyroscope for game play. Both GPS location tracking and gyroscope-based phone orientation will be used to create an augmented reality that will form the basis of the game play elements.

Reference Documents:

- Concept of Operations
- Project Plan

Applicable Standards

- This project does not need to conform to any standards to operate.

Definitions, Acronyms, and Abbreviations

- GPS: Global Positioning System

Section 2: Product Overview

Assumptions:

- We assume that the Android devices capable of running the application will have access to the device's GPS location information and the device's orientation.

Stakeholders:

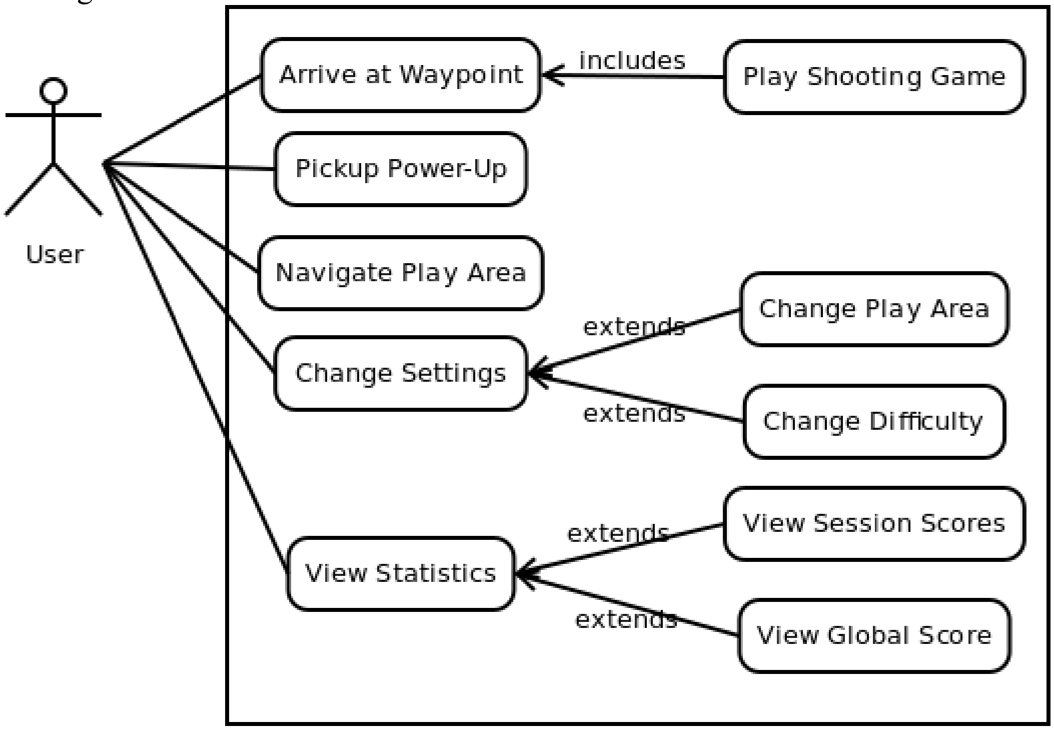
- The only stakeholder we have in our application are people that enjoy video games, to meet their needs we need to make our game easy to use and most importantly, enjoyable.

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Player arrives at waypoint	The user has physically arrived at the waypoint's location	The user begins the shooting game.	The waypoint is removed from the map and the user is presented with the option to start the shooting game.
Player finishes Shooting game	The user has either lost all life or has defeated all the enemies	The user is presented with their score and then the map is displayed with a new waypoint generated.	The player's game score is added to the player's session score and a waypoint is added within the play area.
Player arrives at power up location.	The user has physically arrived at a power up's location.	The player has a power up added to their inventory.	The power up is added to a list of the user's current power ups, for use in the shooting game. The power up is removed from the game area.
Player starts the game	The user launches the application	The game starts and displays the map, a waypoint is then generated that the player must go to.	A waypoint is generated in the user's play area.

Player finishes a game session	The user closes the application or the application is shut down	The application disappears from view.	The player's session score is saved to a file.

Use Case Diagram



Use Case Descriptions:

- Arrive at Way-point - when the user physically arrives at a way-point, the user is presented with a choice to play the shooting game or move onto another way-point if the surrounding area is too dangerous to play a game.
- Play Shooting Game - The user engages the enemy with the phone to shoot them, scoring points as they defeat enemies. The game ends when a specified number of enemies have been destroyed or the user runs out of life points.
- Pick up Power-up - The user physically arrives at the location of a power up, which is added to the user's inventory and will then be available for use during the shooting game.
- Navigate Play Area - The user physically navigates the play area, searching for power ups and navigating to the next way-point
- Change Settings - The user wants to change parameters of the game
- Change Play Area - The user can change the area the game is played in, by setting the center of the play area and the radius of the play area from that center point.
- Change Difficulty - The user can change how difficult the game is with enumerated values ranging from "Easy" to "Insane" which effects how far apart way-points are separated, the number of enemies required to be defeated before winning, and how difficult the enemies are to defeat.
- View Statistics - The user wants to view statistics of previous game sessions
- View Session Score - The user views the high scores of previously played game sessions
- View Global Score - The user views the cumulative score over all games

Section 3: Specific Requirements

Requirement 1

Statement: The application will take in GPS data.
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: If the system can take in the GPS location at regular intervals and display that information in any format.
Revision History: N/A

Requirement 2

Statement: The application will display a map.
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: If the application starts up and displays a map of any region of the world.
Revision History: N/A

Requirement 3

Statement: The application will take in GPS data and display the user’s location on the map.
Source: None
Dependency: Requirement 1, Requirement 2
Conflicts: None
Supporting Materials: None
Evaluation Method: If the application starts up and displays a map of the area local to the user with a marker denoting their position.
Revision History: N/A

Requirement 4

Statement: The application will define the game’s play area.
Source: None
Dependency: Requirement 3
Conflicts: None
Supporting Materials: None
Evaluation Method: If the application starts up, gets the user’s location and then generates a circular feature around the player to denote the play area.
Revision History: N/A

Requirement 5

Statement: The application will run on the Android Operating System 2.2
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The application launches and runs successfully on a device running Android OS 2.2 with GPS and gyroscope
Revision History: N/A

Requirement 6

Statement: The application will use the gyroscope to determine the device’s orientation
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The application will be able to accurately obtain and display the phone’s orientation angles.
Revision History: N/A

Requirement 7

Statement: The player will be able to use the device’s orientation to view a complete 3D space

Source: None
Dependency: Requirement 6
Conflicts: None
Supporting Materials: None
Evaluation Method: Changing the phone’s orientation will correctly display objects in a 3D space
Revision History: N/A

Requirement 8

Statement: The player will be able to use the phone’s orientation to generate weapon fire in the center of the device’s screen
Source: None
Dependency: Requirement 7
Conflicts: None
Supporting Materials: None
Evaluation Method: The application will be able to display a moving projectile in the 3d space that travels along an axis that corresponds to the center of the device’s screen.
Revision History: N/A

Requirement 9

Statement: The application will generate and display power ups on the map, within the play area.
Source: None
Dependency: Requirement 4
Conflicts: None
Supporting Materials: None
Evaluation Method: When the game session starts, power ups will be displayed on the map corresponding to their physical location.
Revision History: N/A

Requirement 10

Statement: The application will store the player’s score for a game session when the program exits

Source: None
Dependency: Requirement 21
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to view their score for the domain session
Revision History: N/A

Requirement 11

Statement: The player will be able to obtain power ups in the map view.
Source: None
Dependency: Requirement 10
Conflicts: None
Supporting Materials: None
Evaluation Method: When the game session starts, the player will be able to go to the physical location of a power up and have that power up added to their inventory.
Revision History: N/A

Requirement 12

Statement: The application will generate and display a waypoint on the map at the start of the game, within the play area.
Source: None
Dependency: Requirement 4
Conflicts: None
Supporting Materials: None
Evaluation Method: When the game session starts, a waypoint will be generated and displayed on the map, corresponding to its physical location.
Revision History: N/A

Requirement 13

Statement: The player will be able to activate a waypoint by going to it’s physical location

Source: None
Dependency: Requirement 12
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to go to the waypoint's physical location that then generates some sort of event.
Revision History: N/A

Requirement 14

Statement: Zombie Chickens will charge towards the player in the FPS mini game.
Source: None
Dependency: Requirement 7
Conflicts: None
Supporting Materials: None
Evaluation Method: When the FPS mini game mode begins, Zombie Chickens will spawn, and attack the player by moving towards them.
Revision History: N/A

Requirement 15

Statement: Player health with change depending on how many zombie chickens hit you, and how many power ups you have collected.
Source: None
Dependency: Requirement 14
Conflicts: None
Supporting Materials: None
Evaluation Method: The players health is a variable that changes depending on how many chickens have attacked the player or how many power ups the player has collected
Revision History: N/A

Requirement 16

Statement: The player will be able to play the shooting game.

Source: None
Dependency: Requirement 15, Requirement 17
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to shoot the enemies and ultimately defeat them or lose all health, either of which ending the game.
Revision History: N/A

Requirement 17

Statement: Players bullets will remove or weaken zombie chickens
Source: None
Dependency: Requirement 8
Conflicts: None
Supporting Materials: None
Evaluation Method: Bullets that have made contact with the zombie chicken will either remove them, or cause them to become weaker. Upon becoming weak enough, they will be removed from game.
Revision History: N/A

Requirement 18

Statement: The player will be able to view their previous session statistics
Source: None
Dependency: Requirement 10
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to view a screen that displays information about their previous sessions.
Revision History: N/A

Requirement 19

Statement: Activating a waypoint will start the shooting game.
Source: None

Dependency: Requirement 18
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to go to a waypoint’s physical location which will then start the shooting game, which they will be able to complete and then return to the map screen.
Revision History: N/A

Requirement 20

Statement: The player will be able to change parameters about the game
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The user will be presented with a menu that displays different configuration options.
Revision History: N/A

Requirement 21

Statement: The player will be able to exit the application.
Source: None
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The player will be able to exit the application, notifying the system that the application is closing.
Revision History: N/A

3.1 Functional Requirements

- The user needs to be able to view their own GPS position on a map, the location of power ups on a map, and the location of way-points on the map.
- The user needs to be able to utilize the gyroscope to aim at enemies displayed on the screen and shoot them to defeat them.
- The users needs to be able to control game parameters:
    - The play area size
    - The difficulty of the enemies

- The user needs to be able to view their statistics
    - The session high scores
    - The global high score

3.2 Interface Requirements

- The input to our application will be GPS location, phone orientation, user defined settings, and the current time
- The output of our application will be the player's location, the player's score, and the game's objects
- The GPS data will be formatted as latitude, longitude, and altitude. The phone's orientation will be formatted as three Euler angles. The user's settings will be defined as play area and game difficulty. The player's location will be formatted as a position on a map. The player's score is formatted as an integer. The game's objects will be formatted as enemies, way-points, and player power ups.
- GPS location data and phone orientation will collected asynchronously while the rest of the system will operate imperatively.
- GPS location data and phone orientation will be received on the order of several per second, the player's location will also be sent on the order of several per second while the others will happen as they are required.

3.3 Physical Environment Requirements

- The application will be designed to run on Android Operating System devices
- We do not impose any serious physical requirements to run the software, just that the mobile device has a strong GPS signal and is being operated in a safe area.
- However, we must consider if:
    - The user is in a GPS dead zone
    - The user is in a safe operating area
    - The device's environmental requirements

3.4 User and Human Factors Requirements

- Our system does not need to support any specific users.
- We are assuming the user has low technical knowledge of the system and therefore we will embed all the information the user needs to operate the application within the application.
- We defer the risk of the user tampering with the application to the operating system as the inputs to the system are being collected from resources outside of our control.

3.5 Documentation Requirements

- The documentation will be included in the application itself because we want our application to be as self contained as possible to facilitate ease of use.
- Because we want our application to be simple to use, we will design the documentation to explain use to the lowest sensible skill level.

3.6 Data Requirements

- The primary equations that we need to utilize in our system is the distance formula, to compare distances between the user and the objectives and map objects, as well as transformation from 3D coordinates to 2D Screen coordinates to display the enemies in the local area minigame
- We need to retain the player's score and settings in our system to persist over time.

3.7 Resource Requirements

- We require programmers to develop the system as well as mathematicians to generate the functions of our data requirements. We may also require graphic artists to improve the interface and display of the application.

- We do not require any special resources operate this application.
- The goal is to have this project completed by the end of the semester.
- We have not and will not seek funding for this project at this time.
- To develop for this application we will require the Java Eclipse Application with the Android SDK installed as well as some Android based device to do device testing on.

3.8 Security Requirements

- The application itself will have no security features. Any and all security features will be derived from the features of the operating system.

3.9 Quality Assurance Requirements

- We define our requirements of quality to be as follows:
    - If the application operates for an entire session without error then we deem the application reliable, however we will not assure reliability all the time and instead will handle errors gracefully without ending the application session.
        - If the application can run again after completing a session without error, it will deem reliable. As it must be able to store data, such as high scores, from previous runs.
    - Availability will be outside of our control as the OS will ultimately be the limiting factor.
    - Maintainability is not guaranteed unless the stakeholders provide the support to continue.
    - Security is not guaranteed and will be deferred to the OS to handle.
    - Portability is guaranteed as it is the goal of the application to be accessible everywhere.
- The application will be able to detect and handle faults that are created in the application and then handle them gracefully without exiting the application.
- There is no guaranteed minimum turnaround time for errors.
- There is no guarantee the system will always be available.
- There is no guaranteed maximum time to reset the system.
- Response times are guaranteed to be nearly instant, within what we can control in our code.

Section 4: Supporting Material

- Java Platform API Specification (Javadoc)
- Android Software Developer's Kit API
- Google Android Software Developer's Kit API

Template created by G. Walton (GWalton@mail.ucf.edu) on Aug 30, 1999.
Last updated by Sarah Applegate September 9, 2011.