

**GPS Game
Project Management Plan
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	08/15/00	G. H. Walton	Template
v1.0	9/22/2011	Bernard Feeser	Updated template with names, sent to others
v1.1	9/23/2011	Jolene Wan & Danh Nguyen	Project Overview
v1.2	10/07/2011	Jolene Wan & Danh Nguyen	Project Team Organization, Risk Management, Deliverables, Software Life Cycle Process, Tools and Computing Environment, Quality Insurance
v2.0	10/14/2011	Jon Leonard	For first set of deliverables

Contents of this Document

Project Overview
Project Team Organization
Deliverables
Software Life Cycle Process
Tools and Computing Environment
Configuration Management
Quality Assurance
Risk Management
Table of Work Packages, Time Estimates, and Assignments
Technical Progress Metrics
Plan for tracking, control, and reporting of progress

Project Overview

This project encompasses the development of an Android game that utilizes the phone's GPS device and gyroscope. This game will place the user in an augmented reality based on his location where he will have to move to pre-defined checkpoints on the map within a certain time limit to achieve their goal. Enemies, as well as power-ups, will be spawned randomly around the map at the beginning of every level. The user must either avoid or eliminate these enemies in order to proceed to the next checkpoint. Once the user reaches a checkpoint, the user will be required to play a mini-game to progress.

Project Team Organization

Our group consists of Bernard Feeser, Jon Leonard, Danh Nguyen, Jolene Wan, Juan Chen.

Our group leaders are Jon Leonard and Bernard Feeser. Everyone in the group will be coding, and each will be responsible for their part. The parts that each person will be responsible for are the following:

Bernie Feeser

- Coding
- Testing
- Maintenance
- Team leader 2

Jolene Wan

- Coding
- Graphical Design
- Maintenance

Juan Chen

- Coding
- Testing
- Maintenance

Danh Nguyen

- Coding
- Maintenance

Jon Leonard

- Training
- Coding
- Testing
- Maintenance
- Team leader 1

We will have several methods of communication, including face-to-face meetings, Google Docs meetings, texting, and e-mail. Our main form of communication would be weekly face-to-face meetings on Friday, where we discuss progress made, adjust and assign tasks, and work together on the week's objective. Also, each artifact will be managed by several people simultaneously or by a single person so any artifacts due will be worked on together or by that single manager.

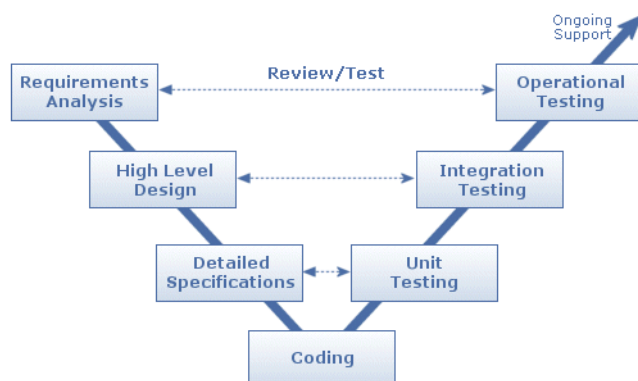
Deliverables

Artifact	Due Dates
Meeting Minutes	Every Monday
Individual Logs	Continually

Group Project Management Reports	Every Monday
Concept of Operations	10/14/2011
Project Plan	10/14/2011
Software Requirements Specifications	10/14/2011
High-Level Design	11/04/2011
Test Plan	11/04/2011
Detailed Design	11/04/2011
User's Manual	12/02/2011
Final Test Results	12/02/2011
Source, Executable, Build Instructions	12/02/2011
Project Legacy	12/02/2011

Software Life Cycle Process

The Software Life Cycle Process that we are following is the V Model. We chose the V-Model because at every step, we can test it and have a chance to change the design if needed and continue on with more designing and testing.



Tools and Computing Environment

We'll be using Java to program an Android application. The application will be built for the Android OS version 2.2. We will also be using the Eclipse IDE with the Android SDK to develop and compile, as well as emulate our program during the coding and testing phases.

Configuration Management

For version control, we will be using Subversion (SVN) provided by Google Projects. SVN will allow us to create, modify, and delete source code changes simultaneously while handling code merges and handling source conflicts that could arise. It will also be able to revert or refer back to previous revisions if needed.

Quality Assurance

Our group will be constantly testing the program throughout the development cycle of the project. Our project is a game that is solely for entertainment purposes; therefore, the QA activities will consist of a game that is enjoyable and keeps the players attention. We will obtain this satisfaction by testing the project throughout its development. We will accomplish this by playing it and having others view and play the game.

Risk Management

The main risks for this project would be related to player injury and safety. Since the GPS does not account for people, trees, cars, and various other possible obstacles on the map, it is possible for the player to accidentally collide with these things during game-play and possibly injure themselves or others. The player could also unknowingly walk into the middle of a road during game-play.

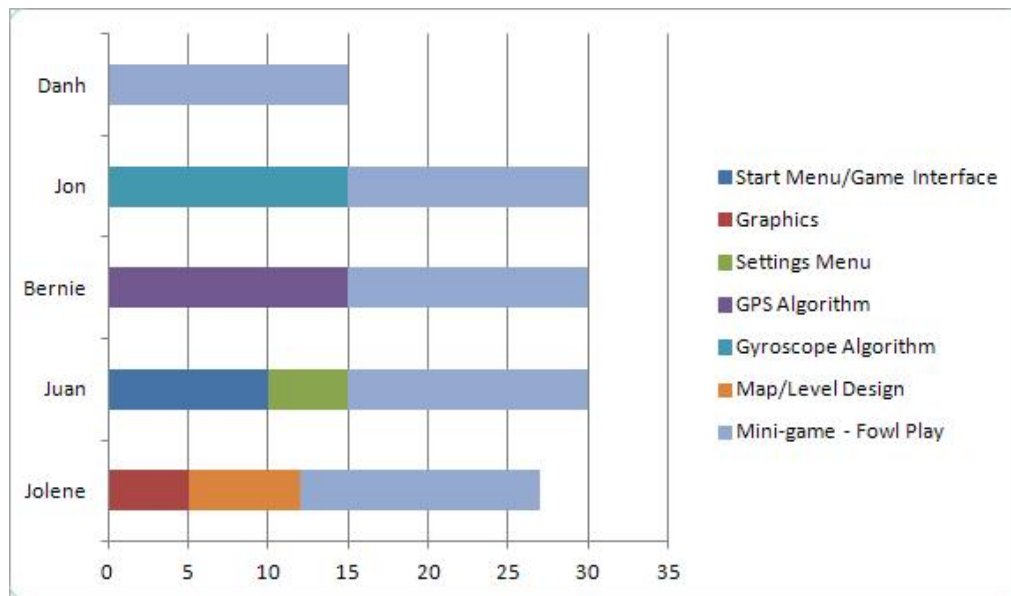
We will be transferring these types of risks by displaying a strong warning message at the beginning of the game. The message will warn players to watch out for objects around them and advise them to play in an wide open area for the best gaming experience.

Another risk is that the player may enter a dead zone where there is limited or no GPS signal. In this case, the player would not be able to play the game at all.

We will be assuming this risk. In the case of a lost or nonexistent signal, the program will display a message to inform the player of the situation and remain there until a valid GPS signal is regained.

Table of Work Packages, Time Estimates, and Assignments

Work Package	Estimated Completion Time (includes debugging!)	Responsible Team Member
Start Menu/Game Interface	10 hours	Juan Chen
Graphics	5 hours	Jolene Wan
Settings Menu	5 hours	Juan Chen
GPS algorithm	15 hours	Bernie Feeser
Gyroscope algorithm	15 hours	Jon Leonard
Map/Level Design	7 hours	Jolene Wan
Mini-game - Fowl Play	15 hours	All



Technical Progress Metrics

Requirements Phase:

Specific Requirements (22 as listed in Software Requirements Specification Section 3.0)

The game is not coded yet so the requirements have not been met yet

Functional Requirements (4 listed in Software Requirements Specification Section 3.1)

The game is not coded yet so the requirements have not been met yet

Interface Requirements (5 listed in Software Requirements Specification Section 3.2)

The game is not coded yet so the requirements have not been met yet

Physical Environment Requirements (3 listed in Software Specification Section 3.3)

The game is not coded yet so the requirements have not been met yet

User and Human Factors Requirements (2 listed in Software Specification Section 3.4)

The game is not coded yet so the requirements have not been met yet

Documentation Requirements (2 listed in Software Specification Section 3.5)

The game is not coded yet so the requirements have not been met yet

Data Requirements (2 listed in Software Specification Section 3.6)

We have obtained the distance formula for use but haven't programmed the application yet so one requirement is not met

Resource Requirements (5 listed in Software Specification Section 3.7)

Most requirements are met we have 5 programmers with math skills up to linear algebra, all have

eclipse software with android SDK but the project is not completed as of yet.

Security Requirements

We don't have any security features so this requirement has been met

Quality Assurance Requirements (6 listed in Software Specification Section 3.9)

The game is not coded yet so the requirements have not been met yet

Design Phase:

We have a Use-Case UML diagram created already to help us with our design which is also within our Software Requirements Specification in Section 2. With this UML diagram, we have a general idea of how our design will be made by seeing how modules will interact with each other.

Specifications Phase:

We will have instructions that depict how to use the software. We will also create documentations that will specify how certain classes will work and how they communicate with each other. Such as how the GPS map will correlate with the enemies spawning.

Coding Phase:

The code will be implemented by all the team members, it will be tested as the modules of the program are completed.

Testing Phase:

For the testing phase, we will be testing our application on an Android Phone to how functional it is working as well as edit changes that we deem necessary to change. After the testing, we can also add things we might want to add to further enhance the game to improve playability, consumer happiness with the game and more as well. If a test does not have the desirable results that we want then that part of the project will be revisited and modified to meet them or those requirements will be altered.

Plan for tracking, control, and reporting of progress

We will have several ways of tracking how much progress we are making with the project.

One way is through Google Docs. Our group shares the deliverables on Google Docs and it tells us every time a document is edited, including the person who edited it and time that it was edited. This way, our project leaders can check to see that everyone is working on the deliverables diligently.

Another way that we will be tracking our progress is to have a spreadsheet in which each team member will log their progress every time they work on a certain part of the project. Every week, our team leaders John and Bernie will be checking this log to make sure everyone is working on their assigned part. At the completion of the project, we will add up the total hours that each team member worked on the project. If there are any problems, the team leader(s) will address the issue at the next scheduled team meeting.