2012

Space Shooter

Software Requirement Specifications

COP 4331: Processes for Object Oriented Software Development
Fall – 2012

Team 2
9/3/2012

Version	Date	Who	Comment
v0.5	09/03/2012	Alex Banke	Initially unfinished compilation
v0.6	9/5/12	Alex Banke	Modification to Contact information
v1.0	9/6/2012	Alex Banke	First Completed Version
v1.1	9/11/2012	Alex Banke	Formatting
v1.11	9/13/2012	Alex Banke	Formatting
v2.0	09/17/2012	Alex Banke	Completed second version
v2.1	9/18/2012	Joshua Thames	Formatting
v2.2	9/18/2012	Joshua Thames	Security Requirements

Team Name:

Team 2

Team Members:

Name	Email address
Andre Meireles	andre.meireles@knights.ucf.edu
Alex Banke	banke@knights.ucf.edu
Christopher Margol	margol_chris@knights.ucf.edu
Chris Lin	christophercklin@gmail.com
Joshua Thames	jthames88@knights.ucf.edu
Thaddeus Latsa	tlatsa@knights.ucf.edu

Table of Contents

SECTION 1: Introduction

Software to be Produced:

- The production to be achieved by this project is a 2-dimensional space shooter game. The game will offer the user to control an upgradable "space ship". The project will feature an interactive environment with asteroids moving across the screen at various times, each having their own gravitational pull. The project will also offer the user to "battle" other enemy NPC ships. The purpose of this software is to entertain the users of it, as well as offering a challenge for those seeking a good game.

Reference Documents:

- Concept of Operations
- Project Plan
- Project Management Plan

Definitions, Acronyms, and Abbreviations:

- NPC: Non-player character
- OS: Operating System

SECTION 2: Product Overview

Assumptions:

- We assume that the technology used is bug free and stable
- Ne able to easily integrate with these technologies by reading their docs
- User will have a computer, mouse, and keyboard to play the game
- User will have an internet connection to download the game
- User will have sufficient system spec to run the game

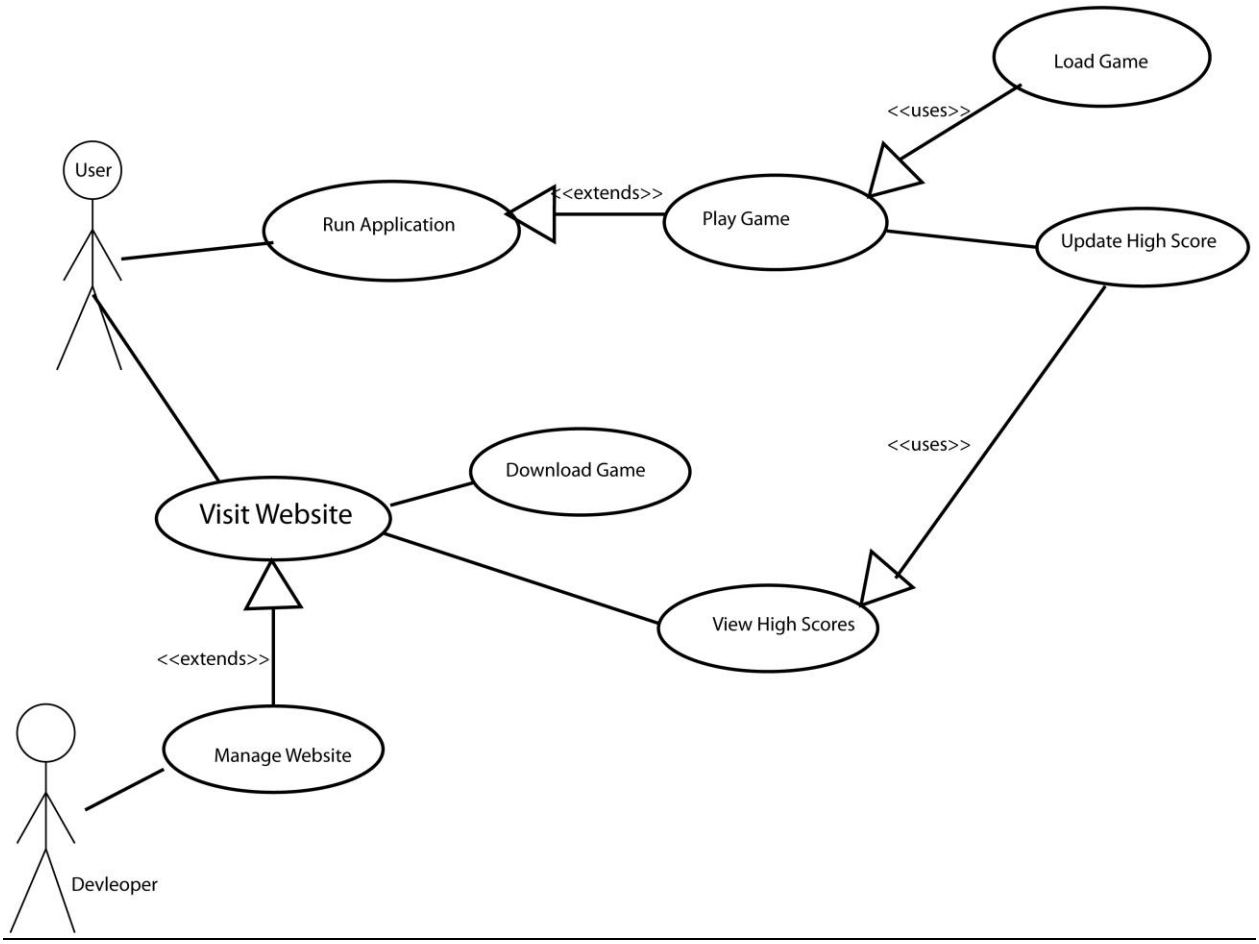
Stakeholders:

- User
  - Casual Gamers: The casual gamer will be looking to play a game in which a short amount of time investment is used to have fun and complete the game.

  - Serious Gamers: The serious gamer will be looking to increase their score to the max possible scoring range.

- Developer
  - As the developers we will be responsible for fixing any bugs post-release and maintaining the accompanies website.

Use Case Diagram:



Use Case Descriptions

- Run Application (User): When the user starts up the application on their computer
  - Play Game: When the user actually plays through the game from start.
    - Load Game: The User can load a previously saved game.
- Visit Website (User): The user can visit the site to do a variety of things.
  - Download Game: The user downloads the game so they can play it.
  - View High Scores: Where the user can view the high scores of completed level
- Manage Site(Dev): The developer manages the site

SECTION 3: Specific Requirements

3.1 Functional Requirements:

No: F-1 GUI
Statement: The system shall have a GUI
Source: To view the game
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: The game will display to the screen

No: F-2 Menu System
Statement: The system shall have a menu system
Source: To easily navigate the different parts of the game
Dependency: F-1
Conflicts: None
Supporting Material: None
Evaluation Method: Unit Testing will ensure the menu system works as it should
Revision History: Alex Banke, September 17, 2012

No: F-3 Gameplay
Statement: The system shall have an environment to play the game in
Source: So the user can play the game
Dependency: F-1, F-2
Conflicts: None
Supporting Materials: None
Evaluation Method: Testing the gameplay
Revision History: Alex Banke, September 17, 2012

No: F-4 Website
Statement: The system shall have a Website
Source: to track high scores and store created levels
Dependency: None
Conflicts: None
Supporting Materials: UCD
Evaluation Method: Unit Testing, and Security Testing, Stress Testing
Revision History: Alex Banke, September 17, 2012

No: F-5 High Score Database
Statement: The system will have a method for keeping track of high scores
Source: Leaderboards
Dependency: F-3, F-4
Conflicts: None
Supporting Materials: UCD
Evaluation Method: Unit Testing, Stress Testing, Security Testing
Revision History: Alex Banke, September 17, 2012

3.2 Interface Requirements:

No: I-1 Menus
Statement: The game will have menus for navigation
Source: Required for game functionality
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Menus usable
Revision History: Alex Banke, September 17, 2012

No: I-1.1 Main menu
Statement: Main menu of the game
Source: Required for game functionality
Dependency: I-1
Conflicts: None
Supporting Materials: Menu Diagram
Evaluation Method: Ensuring the menu is use able
Revision History: Alex Banke, September 17, 2012

No: I-1.2 Options
Statement: This menu shall contain all game options
Source: Required for game functionality
Dependency: I-1
Conflicts: None
Supporting Materials: None
Evaluation Method: Ensuring the menu is usable
Revision History: Alex Banke, September 17, 2012

No: I-1.3 High Score Menu
Statement: This menu shall contain user high scores
Source: Game functionality requirement
Dependency: I-1
Conflicts: None
Supporting Materials: None
Evaluation Method: The usability of menu will be tested
Revision History: Alex Banke, September 17, 2012

No: I-2 Website
Statement: The website shall contain a download and high score list
Source: Functionality requirement
Dependency: None
Conflicts: None
Supporting Materials: None
Evaluation Method: Testing high score viewing\submission
Revision History: Alex Banke, September 17, 2012

No: I-2.1 High Scores
Statement: A high score list will exist on the site
Source: Game functionality requirement
Dependency: I-2
Conflicts: None
Supporting Materials: Database
Evaluation Method: Ensuring high score viewing functions
Revision History: Alex Banke, September 17, 2012

3.3 Physical Environment Requirements:

No: P-1 Hardware
Statement: The system will run on any personal computer with specs meeting or beyond the following: 1.5ghz CPU, 512MB RAM, 500MB Free Hard Drive Space, and a 64MB GPU.
Source: Required to let users know if they can use our system.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: The system will be tested on various PC setups to determine if we've met our benchmarks.
Revision History: Alex Banke, September 17, 2012

No: P-2 Physical Location
Statement: The system should run in any conditions in which a PC would normally run.
Source: Users will expect the system to work wherever they're PC is working.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None, So long as the PC is running, this requirement should be automatically met.
Revision History: Alex Banke, September 17, 2012

3.4 Users and Human Factors Requirements:

No: UH-1 Users must be able to operate a keyboard.
Statement: The system should be fully functional using just a keyboard (Although a mouse will be compatible too).
Source: Computer users often expect to be able to fully operate an application with at least a keyboard.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: We will play the game with just a keyboard in order to ensure it is completely functional.
Revision History: Alex Banke, September 17, 2012

No: UH-2 Users should have enough skill to be able to play most casual games.
Statement: The system should be playable by someone with average gaming skill levels.
Source: Users will expect the difficulty to be around average.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: We will play the game with just a keyboard in order to ensure it is completely functional.
Revision History Alex Banke, September 17, 2012

3.5 Documentation Requirements:

No: D-1 Game Manual
Statement: A game manual will be provided allowing an average gamer to quickly learn how to play our game.
Source: Users will expect some sort of manual or instructions so they know how to play our game.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: We will have test subjects read our manual and play our game providing feedback along the way.
Revision History: Alex Banke, September 17, 2012

No: D-2 Code Documentation
Statement: Our code will be well documented.
Source: Anyone developing in our application will expect well documented code.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: We will have fellow developers look at and evaluate our code documentation.
Revision History: Alex Banke, September 17, 2012

3.6 Data Requirements:

No: D-1 Basic Physics
Statement: The game should have basic physics.
Source: Users will expect the game to feel relatively real and basic physics are needed for this.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: Play testers will give us feedback on our realism.
Revision History: Alex Banke, September 17, 2012

No: D-1.1 Gravity
Statement: The game will implement gravity so the character moves as expected.
Source: Users will expect realistic movement.
Dependency: D-1
Conflicts: None
Supporting Material: None
Evaluation Method: We will have test subjects read our manual and play our game providing feedback along the way.
Revision History: Alex Banke, September 17, 2012

No: D-2 Scoring
Statement: The game will keep track of players scores and store them in a database.
Source: Users will expect a way to see how well they performed.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: We will have test subjects read our manual and play our game providing feedback along the way.
Revision History: Alex Banke, September 17, 2012

3.7 Resource Requirements:

No: R-1.1 Resource Requirements (Java)
Statement: The game will be written on the Java framework
Source: N/A.
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: R-1.2 Resource Requirements (Network)
Statement: The game will require networking for level downloading and submissions
Source: None
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: R-1.3 Resource Requirements (Maintenance)
Statement: The website and database will require maintenance
Source: None
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

3.8 Security Requirements:

Access: Do to the entertainment nature of our program, no restricted access will be necessary
Data: Each user's data will be isolated from another's via storage to a txt document
Faults: The system will not attempt to isolate or detect faults. We will rely on heavy testing to find any defects before release and any in the release will have to patched as they arise.
Mean time: No prescribed mean time between failures, hopefully there will be none.
Back up: Github will back up our source code as we upload it to their network. Each programmer will also have a copy of the project stored on his hard drive
Precautions: none will be taken in case of fire, water damage, etc because our code will be stored on an external server. The team will use login passwords for their operating systems and login passwords to upload to the server but our code will be public to view.
Recovery: no requirements for this project

3.9 Quality Assurance Requirements:

No: QA-1 Levels are Beatable
Statement: Levels must be beatable
Source: Levels must be beatable to ensure game enjoyment
Dependency: None
Conflicts: None
Supporting Material: None

Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: QA-1.1 Regarding Framerate
Statement: Game must run at a minimum 24 FPS
Source: Game must be smooth to ensure game enjoyment
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: QA-1.2 Game Mechanics and Framerate
Statement: Game mechanics must be independent of FPS
Source: Game must be smooth to ensure game enjoyment
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: QA-1.3 Consistent Themes
Statement: Themes must be clean and polished
Source: Themes must be good looking to ensure user enjoyment
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

No: QA-1.4 Bug Free
Statement: Game must be bug free
Source: Game must be smooth to ensure game enjoyment
Dependency: None
Conflicts: None
Supporting Material: None
Evaluation Method: None
Revision History: Alex Banke, September 17, 2012

SECTION 4: Supporting Material

See “Reference Documents” above.