**Software Requirements Specification**

For

Project Community Game

# Eddie O’Hagan, Justin Dale, Edgardo Linero, Candy Wong

CSE 308

Prof. McKenna

2/22/12

Table of Contents

1. Introduction ................................................................................................................. 3

1.1 Purpose........................................................................................................... 3

1.2 Scope.............................................................................................................. 3

1.3 Definitions....................................................................................................... 3

1.4 References...................................................................................................... 4

2. Overall Description...................................................................................................... 4

2.1 Product perspective........................................................................................ 4

2.2 Product functions............................................................................................ 4

2.3 User characteristic........................................................................................ 12

2.4 Constraints................................................................................................... 12

2.5 Assumptions and dependencies.................................................................. 13

3. Specific requirements................................................................................................ 13

3.1 External interfaces........................................................................................ 13

3.2 Functions...................................................................................................... 13

3.3 Performance requirements........................................................................... 16

3.4 Logical database requirements.................................................................... 16

3.6 Software system attributes........................................................................... 16

Supporting information.................................................................................................. 17

**1. Introduction**

1.1 Purpose

a) To define and effectively communicate a specific design of Project Community. This is a game that will be created for Windows.

b) This document is initially intended for but not limited to use by a software developer to provide guidelines for implementation or by a user of the software created who wishes to become more knowledgeable about the software’s specific implementation.

1.2 Scope

a) This is a game that will follow the UN 2012 Millennium Development Goals and will be entered into the XNA Imagine Cup.

1.3 Definitions, Acronyms, and Abbreviations

* + - CSV File (Comma Separated File): The file containing numbers and positions corresponding to where tiles need to be placed to create the game world.
    - Tile: A tile is a piece of artwork that when put next to a bunch of other tiles, creates a seamless picture.
    - Viewport: Contains the constraint view that the user can see at any given time. The user can however manipulate the position of the viewport to see parts of a map that may or may not be showing at any given time.
    - Reputation (Rep): A numeric value given to each player to depict the current standing in the games society. Rep will be used to determine whether or not the user can accomplish certain tasks.
    - Money: Will be used to purchase items in the game. The player will be able to buy items with money at a General Store and at a Pharmacy.
    - SRS: Software Requirements Specification
    - GUI: A GUI or Graphical User Interface is an Acronym which will be used to describe all the visual elements that the user will be able to interact with.
    - Player: The Player is the same as the User of the program. This is the person interacting with the end result of the described software.
    - Box2D: The physics library we will be using to help us resolve collisions between game objects.
    - XNA: This is the Xbox development library which allows us to make Windows and Xbox games.
    - XACT: Xbox Audio Creation Tool allows us to import .mp3 files and convert them to an XNA compatible format.

1.4 References

a) <http://create.msdn.com/en-US/>

* XNA Homepage

**2. Overall Description**

2.1 Product Perspective

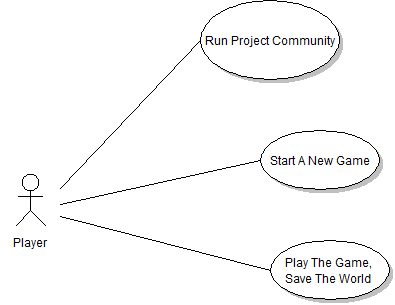
We are dependent on the Windows operating system, C#, XNA framework, and the CLR virtual machine to run C#.

2.2 Product Functions

The gamer will be able to perform various actions through the Main GUI including but not limited to playing game. The player will also be able to perfom actions in game such as viewing your progress.

2.2.1 Function Use Cases

A1. Use Case: New game.

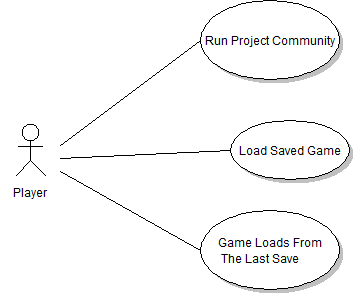


- Purpose: Creates a new game by removing all the entities and rebuild the level

- Description:

* On game start, the level will remove any previous entities left over from a previous game.
* The game world will now be created by reading in a .csv file. All of the data in the .csv file corresponds to a particular tile to use in a particular location in the game world.
* Once the game world is constructed, we position the game entities in the world. Some of these entities are quest items, quest givers, and town’s people.

A2. Use Case: Load Game

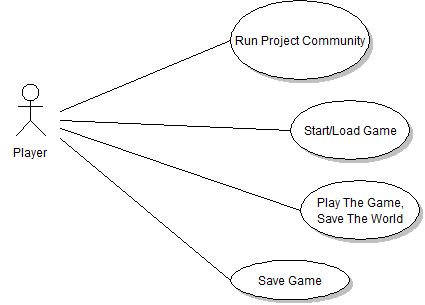


- Purpose: Loads a previously saved game from an external file(s).

- Description:

* The information from the save file will be read in by the program and that data will be stored in its respective locations.
* The game will then load the game world and allow the player to continue from where they left off.

A3. Use Case: Save Game

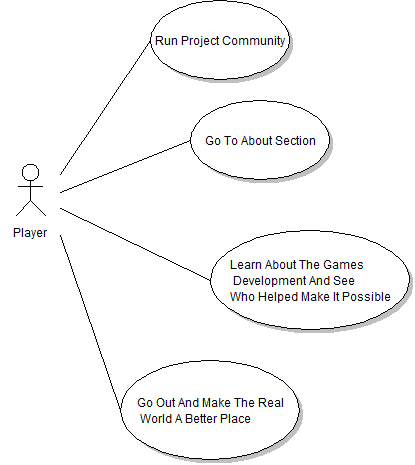


- Purpose: Saves character/environment data to an external file(s).

- Description:

* The information from the game world and the player will be taken and written to a file. This way the user can exit the game after saving and play it again another time.

A4. Use Case: About/Credits

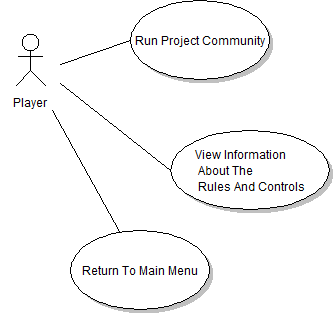


- Purpose: Displays credit screen.

- Description:

* The credit screen shows in plain text who worked on the game and their respective jobs on the project

A5. Use Case: Help/Controls

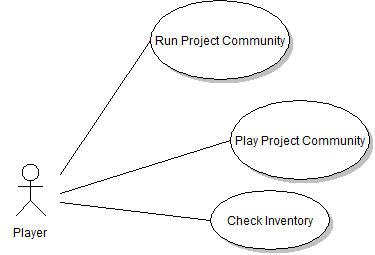


- Purpose: Displays controls screen.

- Description:

* This screen shows the user the keyboard and mouse controls that are used in the game.

A6. Use Case: Inventory

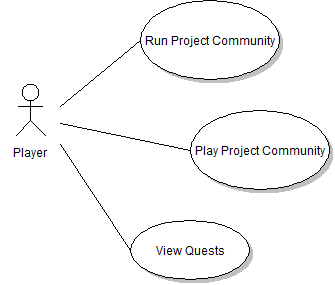


- Purpose: Holds items for a player.

- Description:

* Holds items that the player will obtain throughout the game. The player’s inventory will be displayed when they click on the inventory button or hit the “i” key on the keyboard.

A7. Use Case: Quest Log

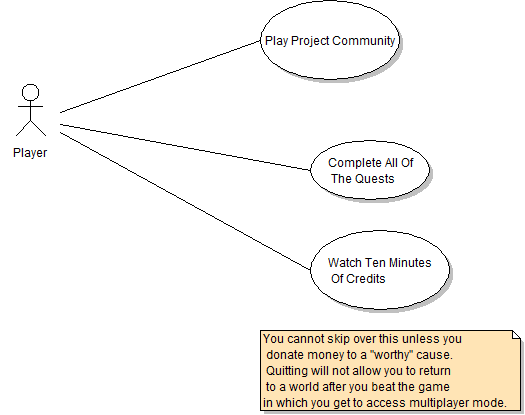


- Purpose: Holds quest information for a player.

- Description:

* The Quest log will hold information about current quests the player has and quests that the player has completed. The Quest log will come up when the user click on the quest log button or hits the “L” key.

A8. Use Case: Beat the Game

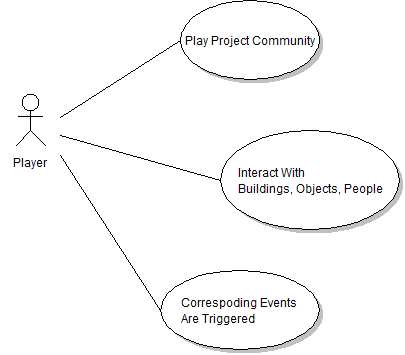


- Purpose: The overall goal for the player.

- Description:

* When the player is able to complete all the quests in the game, then they have beaten the game.

A9. Use Case: Interaction with buildings and people

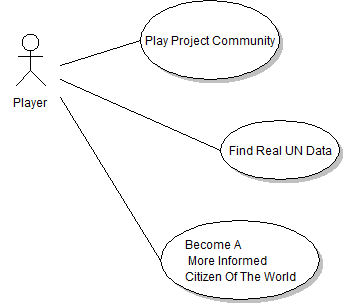


- Purpose: Player goes into a building

- Description:

* If the player collides with the entrance of a building, we will change the current game world to that particular building’s game world.

\*A10. Use Case: View UN Data



- Purpose: Obtain UN database data for the game.

- Description:

* We will incorporate UN statistical data into the game.

(\* indicates if time permits)

2.3 User characteristics

2.3.1 End User

* School Teacher who’s forcing their students to play the game for educational purposes.
* A Gamer that has nothing better to play

2.4 Constraints

2.4.1 The viewport cannot have negative values otherwise the game will crash.

2.4.2 Only so many textures can be used in a single sprite batch and we can only have so many sprite batches per frame.

2.4.3 The game must contain minimal violence and be rated E for everyone.

2.5 Assumptions and Dependencies

2.5.1 System Constraints

- 2.10 GHz Processor, 2GB System Ram, 2GB of HDD space.

2.5.2 Programming Language

- C#, XNA, and XML.

2.5.3 Operating System

- Windows XP (SP 3) or Windows 7. Windows Vista will explicitly not be supported.

**3. Specific Requirements**

-See Game Design Doc

3.2 Functions

3.2.1 New Game

|  |  |
| --- | --- |
| Use Case  Name: | New Game |
| Priority: | High |
| Trigger: | Clicking “New Game” button |
| Precondition: | User is at the main menu or in-game menu and selects new game. |
| Basic Path: | 1. User starts the application and is brought to the main menu. 2. User clicks new game button 3. A new game is started. |
| Alternate Path: | 1. User could access in-game menu via Esc key during game play. 2. User could then click the new game button. |
| Post-condition: | The user will now start at the beginning of the game. |
| Exception Path: | 1. Game could fail to start due to improper map loading. |
| Other: | - |
| Reference: | SRS Section 2.2.1.A1 |

3.2.2 Load Game

|  |  |
| --- | --- |
| Use Case  Name: | Load Game |
| Priority: | High |
| Trigger: | Clicking “Load Game” button |
| Precondition: | User is at the main menu or in-game menu and selects load game. |
| Basic Path: | 1. User starts the application and is brought to the main menu. 2. User clicks load game button 3. The game checks to see if there is a previously saved file and opens it. |
| Alternate Path: | 1. User could access in-game menu via Esc key during game play. 2. User could then click the load game button. |
| Post-condition: | The user will now start where they left off previously. |
| Exception Path: | 1. The save file might not exist or the save file might be corrupted. If so then we will let the user know that we could not load the save file. |
| Other: | - |
| Reference: | SRS Section 2.2.2.A1 |

3.2.2 Save Game

|  |  |
| --- | --- |
| Use Case  Name: | Save Game |
| Priority: | High |
| Trigger: | Clicking “Save Game” button |
| Precondition: | User is at the main menu or in-game menu and selects load game. |
| Basic Path: | 1. User is already playing the game and presses Esc key. 2. In-game menu is opened. 3. When the user clicks the save button we create a file that holds game and player data. |
| Alternate Path: | 1. None |
| Post-condition: | The user will be informed if the save was successful or not. |
| Exception Path: | 1. The save file might not exist or the save file might be corrupted. If so then we will let the user know that we could not load the save file. |
| Other: | - |
| Reference: | SRS Section 2.2.2.A1 |

* + 1. About / Credits

|  |  |
| --- | --- |
| Use Case  Name: | About / Credits |
| Priority: | High |
| Trigger: | Clicking “About/Credits Game” button |
| Precondition: | User is at the main menu or in-game menu and selects About/Credits. |
| Basic Path: | 1. User starts the application and is brought to the main menu. 2. User clicks load game button 3. The game checks to see if there is a previously saved file and opens it data. |
| Alternate Path: | 1. User can access the About/Credits screen via the in-game menu. |
| Post-condition: | The user will be returned to their previous screen once they chose to exit the About/Credits screen. |
| Exception Path: | 1. None |
| Other: | - |
| Reference: | SRS Section 2.2.2.A1 |

* + 1. Help / Controls

|  |  |
| --- | --- |
| Use Case  Name: | Help/Controls |
| Priority: | High |
| Trigger: | Clicking “Help/Controls Game” button |
| Precondition: | User is at the main menu or in-game menu and selects Help/Controls. |
| Basic Path: | 1. User starts the application and is brought to the main menu. 2. User clicks load game button 3. The game checks to see if there is a previously saved file and opens it data. |
| Alternate Path: | 1. User can access the About/Credits screen via the in-game menu. |
| Post-condition: | The user will be returned to their previous screen once they chose to exit the About/Credits screen. |
| Exception Path: | 1. None |
| Other: | - |
| Reference: | SRS Section 2.2.2.A1 |

**3.3** **Performance Requirements**

3.3.1 The game will check at install if basic requirements are met and will inform the user if their system specifications do not meet the basic requirements.

**3.4** **Logical Database Requirements**

3.4.1 UN Database

- If time permits we will implement a database to hold UN statistical information. This information will be used in the game and possibly for multiplayer

**3.6** **Software System Attributes**

3.6.1 Reliability – Project Community will have to be reliable in saving and loading game-play. If this is not done correctly the user could lose their work and become discouraged from playing the game. AI pathways could be malformed and cause an AI to walk into walls. This could cause some quests to become unavailable and impossible to complete.

3.6.2 Availability – Project Community will be available via the United Nations home page. We will also attempt to make it available to download on the Xbox Live Marketplace and on my website http://www.katianie.com.

3.6.3 Security – The save files will be serialized therefore being unreadable to a user.

3.6.4 Extensibility– Project Community can be extended to the windows phone or any mobile device with the XNA framework.

3.6.5 Maintainability – Once the game is complete, it will be play tested indefinitely and updated according to player feedback.

4. **GUI Diagrams**

