Requirement and Analysis for Hills

Version 1

1. Introduction:

1.1. Purpose of Application

The aim of the application is to create a 3D application which contains a small island world, populated by a player, NPC:s and various item that the player can collect. Furthermore, the island and the island's terrain will be procedurally generated by the computer.

1.2. Characteristics of Application:

The application will be an Linux / Windows / Mac application.

1.3. Scope:

The project will not be networked.
Saving a game is not supported.
The project will use both keyboard for movement and actions, and mouse for character directional changes.
Sound effects will not be implemented.
The GUI will be limited to the menu, and a very simple HUD for ingame.

1.4. Definitions, acronyms and abbreviations

- NPC: Non player character: a character controlled by the computer.
- Procedurally generated: Data (in this case the island) created by the computer in a seemingly random pattern, following a set of parameters.
- GUI: Graphical user interface: the interface the user uses to communicate with the application.
- HUD: Heads up display: A GUI that is set atop of the game itself.

2. Requirements

2.1. User interface Sketches, drawings and explanations of the application user interface (possible navigation).

The user will upon application start be greeted with a menu. From this menu the user can either continue into game by pressing a start button. If the user wants to exit the application while in the menu he/she can press the exit button.

After pressing start a loading splash screen will appear while loading the game world is taking place.

When in-game the user will see a crosshair in the middle of the screen which will be used to interact with the world.

2.2. Functional requirements What will the user be able to do ? Write a list of use case names (id's) in the language of the customer. The specific flows for each use case is recorded below. Specify a use cases in priority order.

Use cases:

- Start playing
- Move
- Jump
- Collect item
- Fire gun
- Hit by enemy
- Stop playing

2.3. Non-functional requirements Any special considerations besides functionality? Usability, reliability, performance, supportability, legal, implementation, …
NOTE: Testability mandatory (must have tests)

The program should be runnable on less computational powerful computers, however it is recommended to be used with a more higher end computer. A computer that runs this application must support OpenGL 4.0. The application must include tests.

3.

	Meny		
Start playing		Actor	System
	1	Start button clicked	
	2		A loading bar is initiated, that indicates how far the world has been generated

Stop playing		Actor	System
	1	Close button clicked	
	2		The program shuts down

	In Game		
Move		Actor	System
	1	W,A,S or D is clicked	
	2		the player moves to the corresponding direction

Jump		Actor	System
	1	Space Bar is clicked	
	2		The player jumps

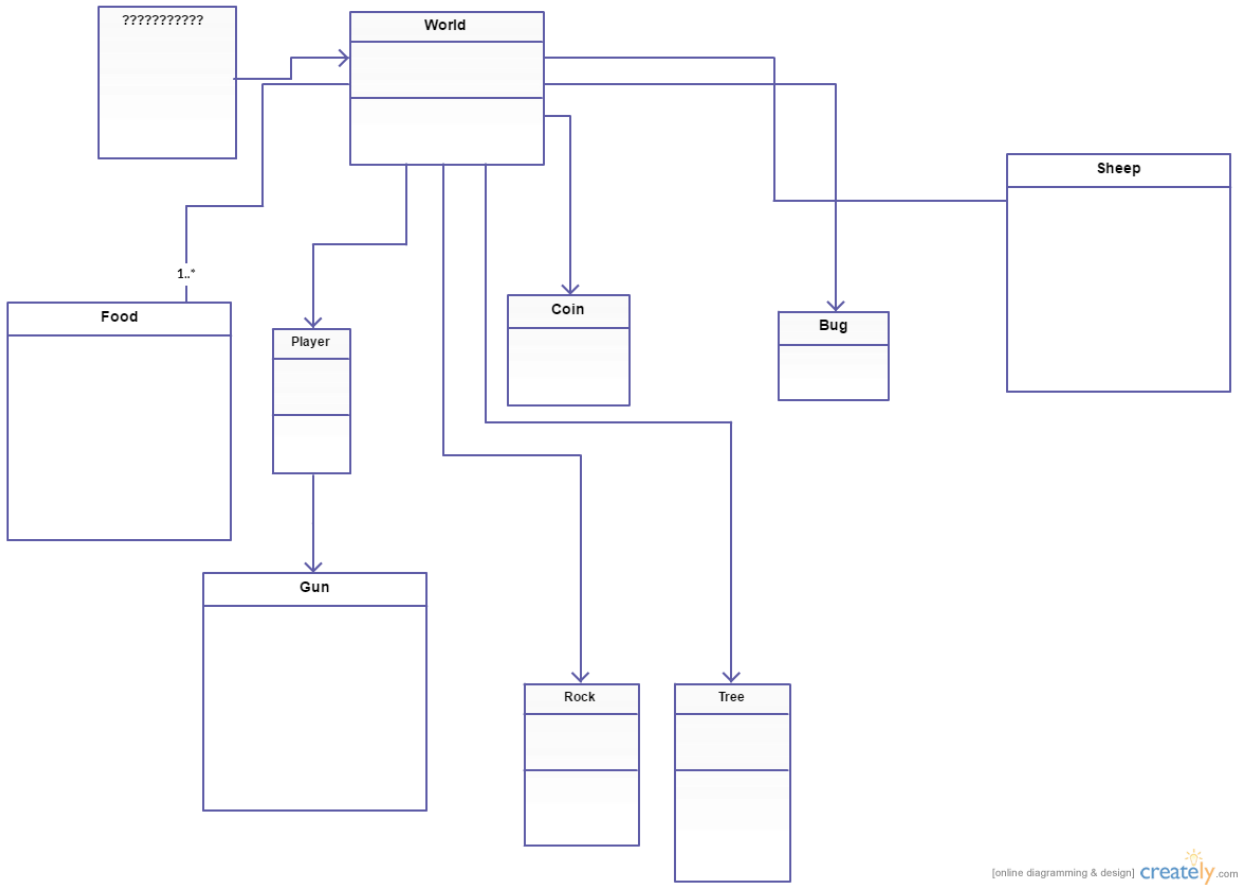
Collect item		Actor	System
	1	Player passes through a coin or Bug	
	2		The coin or bug is removed from the world
	3		A Message is displayed on the screen. "Bug/Coin was collected"
	4		A counter indicating amount of collectibles collected is shown

Fire gun		Actor	System
	1	Player uses gun to shoot	
	2.1		Shot hits enemy. Enemy takes damage. Hitmarker is displayed
	2.2		Shot misses.

Take damage		Actor	System
	1		Enemy hits player
	2		Player recieves indication noise that he's been hit HP Bar decreases

3.1. Use case listing Use case texts (using the use case template)

4.



4.1. Class responsibilities Explanation of responsibilities of classes in diagram

5. References