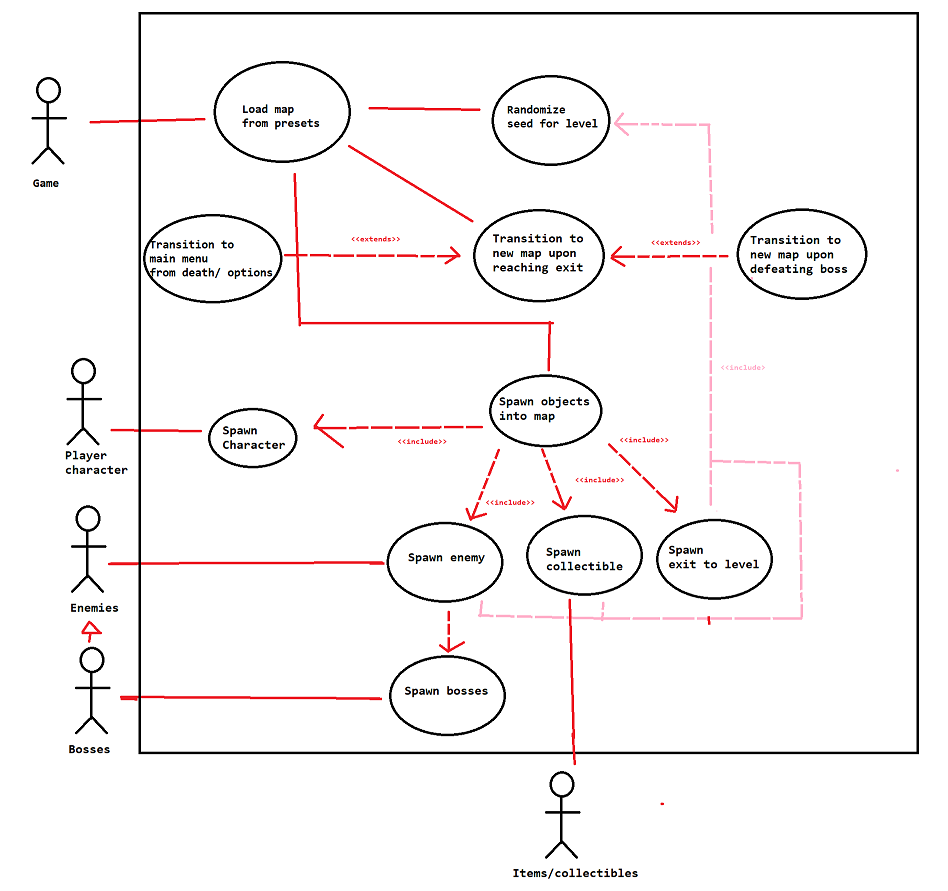
Name: Erik Peavey Mark: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## 1. Brief Introduction

My feature for the group game is to do work on the game’s level design. Because the game we are attempting to make falls under the “roguelike” category, there isn’t an explicit set order to the levels a player might encounter. This contrasts with a lot of traditional “linear” game designs because after every time the player starts up the game or dies in that the levels are completely different upon replays.

A lot of roguelike games handle this by generating level layouts on the fly through a lot of very pinpointed curation of level generation algorithms, but for the scope of this class, creating a bunch of set arenas with randomizable “chunks” which get stitched together with each having scaling randomized enemy and powerup spawns will suffice. My feature will involve the creation and implementation of these “stages” in the game, and all the components that would be required to spawn in everything and transition from one map to the next.

## 2. Use Case with potential scenario

**Name:** Load map from presets

**Summary:** The game starts loading in a new map. It is randomly selected depending on how many levels the player has already completed and the randomized seed.

**Actors:** Game/Player , Player characters, Items, Enemies, Bosses

**Preconditions:** Player is loading in new level from previous or from title screen.

**Basic Sequence:**

**Step 1:** Random seed is first generated and main “archetype” or “preset” of level is selected.

**Step 2.** Objects are spawned and initialized into map, with specific enemy and item spawns placed at dedicated points of the “preset” levels and then randomized based on the seed

**Step 3:** Gameplay begins and the Player must make their way to the exit / defeat the boss if it is a boss arena.

**Step 4:** When the exit has been reached, transition into the next map (which on its own starts to load a new map with a new seed)

**Exceptions:**

**Step 1:** If the map has a boss / is a boss arena, the boss must be defeated in order to be able to progress to further stages.

**Step 2:** If player dies / completes the game / exits to main title screen, the game is sent back to main title screen without any further levels being loaded

**Post conditions:** Player is brought to another level / title screen.

**Priority: 1\***

**ID:** C01

## 3. Data flow diagrams

I don’t have one of these on here right now...

## 4. Acceptance Tests

**Example of test of seed/map randomizer**

Run the overall map initialization feature ≥ 1000 number of times (with different random seeds for a stage from all previous tests) and print debug output / error logs to a file. These differing random seeds could be determined either by incrementing by 1 each time from the previous seed or by running a random number generator to get a seed and checking it between all the previous seeds which have already been run. This will obviously be more computationally expensive, but it would eliminate a lot of the bias in the results that might occur from doing it sequentially.

We can’t be immediately sure that every possible random map game seed will work perfectly fine, so we will have to test a set of seeds for stability and if they happen to result in any warnings or errors (game crashes, terrain intersecting itself somehow, severe performance issues from enemy spawning, or any other funky errors) then the details of the result will be placed onto a text file with the random seed and other pertinent details for further analysis.

The following file will have the following characteristics:

* Line of whitespace in-between separate results which resulted in an error for visual clarity
* # of randomized seed and the subsequent errors / warnings other details with a linebreak in between.

## 5. Timeline

Work Items

|  |  |  |
| --- | --- | --- |
| Tasks | Duration (PWks) | Predecessor Task(s) |
| 1. Re-learning Blender |  | - |
| 1. Modeling Basic Stage Layouts |  | 1 |
| 1. Establishing Basic Level Gameplay Loop |  | 1,4 |
| 1. Object Spawner Superclass Design |  | - |
| 1. Enemy Spawner Subclass Design |  | 4 |
| 1. Boss Spawner Subclass Design |  | 5 |
| 1. Item Spawner Subclass Design |  | 4 |
| 1. Curating Level Randomization |  | 1,4 |
| 1. Stage Texture Work |  | 1 |
| 1. Documentation |  | 8 |
| 1. Testing |  | 8 |
| 1. Installation |  | 8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Job | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |