Name: Cory Clairmont Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

I will oversee audio design when it comes to the game. This will include all enemy and player sound effects, noise notifications, alerts, and background music. This is an important part of the game since it can set the tone of levels or help show what is happening in the game. Poor sound design will lead to a boring and unamusing game. I will also be helping with enemy design in the game.

Some more specific scenarios of what work I will be doing includes sourcing audio samples. There are many free audio samples online but not many will work thematically or add to the game in a meaningful way. Each action should have multiple sound effects as well. This is so a single sound effect never gets played out or annoying. For example, if there is a lot of attacking in the game, having a single attack sound effect may get annoying to hear repeatedly. There will also be multiple power ups which will include sound effect variants. For those scenarios, I will have to edit our current sound effects so they are both similar but apparent that they are different.

Background or theme music will also be very important. Proper theme music will tie a level together and ensure that there are never empty sections in the game. Transitions will also be an important indication of game flow and emphasis. For example when a boss appears having the music change will help show it’s significance.

## Use case diagram with scenario \_\_14

[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

Also include an <<include>> that is a suitable candidate for dynamic binding]

Example:

### Use Case Diagrams

Use Case 1:A diagram of a load level

Description automatically generated

Use Case 2:

A diagram of a game

Description automatically generated

### Scenarios

**Name:** Use Case 1

**Summary:** The level loader generates the level and plays the appropriate theme.

**Actors:** Level Loader.

**Preconditions:** Level Context needs to be known. (i.e.: what level is to be generated)

**Basic sequence:**

**Step 1:** Level Generator puts a request in to load a scene.

**Step 2:** This request extends to the music loader.

**Step 3:** Both the level generator and the music loader require the context of the request to know what level and theme to load.

**Exceptions:**

**Step 1:** Level failed to load: cancel music and level load request, put to main menu.

**Step 2:** Music fails to load: retry music request.

**Post conditions:** Level is generated and ready to play and theme music is playing.

**Priority:** 1

**ID:** UC1

**Name:** Use Case 2

**Summary:** When to player does an action, the proper sound effect plays.

**Actors:** player controller.

**Preconditions:** Game needs to be running and player is performing actions.

**Basic sequence:**

**Step 1:** Player controller filters input into specific action.

**Step 2:** When action is performed, a request for a sound effect is sent.

**Step 3:** The sound effect is located and played during the action.

**Exceptions:**

**Step 1:** Failed to load sound effect: Do not play sound effect.

**Step 2:** Action is cancelled: Cancel sound effect.

**Post conditions:** action is played with sound effect.

**Priority:** 3

**ID:** UC2

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

### Data Flow Diagrams



### Process Descriptions

Assign rooms\*:

WHILE teacher in two places at once OR two classes in the same room

Randomly redistribute classes

END WHILE

**\*Notes**: Yours should be much longer. You could use a decision tree or decision table instead if it is more appropriate.

## Acceptance Tests \_\_\_\_\_\_\_\_9

**Example for audio trigger features**

Run audio triggering features 50 times each and verify each auditorily.

The expected behavior of each feature is:

* Played on time:
  + Does not trigger randomly
  + Does not trigger late
  + Does not trigger early
* Apparent what the sound belongs to:
  + Ie: a jump sound effect helps indicate that the player jumped
* Executed at minimum 49 times
* Executed at most 51 times
* Executed sound matches with expected object:
  + a sound effect doesn’t play for the wrong trigger
  + ie: Jump sound effect playing during attack animation

**Example for sound feature**

|  |  |  |
| --- | --- | --- |
| **Output** | **Input Command** | **Notes** |
| Jump sfx | Player Jump | Jump noises show when the player jumps. |
| Attack sfx | Player Attack | Attack noises play when the player attacks. |
| Death Sounds and theme change | Player Dies | When the player dies, the music adds to the scenario. |
| Change background music between levels | Scene Transition | The music changes between levels for a coherent theme change. |
| Change themes for different level features | Environmental Transition | The theme changes in a level to help indicate important events. |

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items





A diagram of a computer generated diagram

Description automatically generated with medium confidence