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 a 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.

## 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



### Scenarios

**[You will need a scenario for each use case]**

**Name:** Add Numbers

**Summary:** The accountant uses the machine to calculate the sum of two numbers.

**Actors:** Accountant.

**Preconditions:** Calculator has been initialized.

**Basic sequence:**

**Step 1:** Accept input of first number.

**Step 2:** Continue to accept numbers until [calculate] is entered.

**Step 3:** Accept calculate command.

**Step 4:** Calculate and show result.

**Exceptions:**

**Step 1:** [calculate] is pressed before any input: Display 0.

**Step 2:** A button other than [calculate] or a number input is pressed: ignore input.

**Post conditions:** Calculated value is displayed.

**Priority:** 2\*

**ID:** C01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## 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 50 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 divide 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