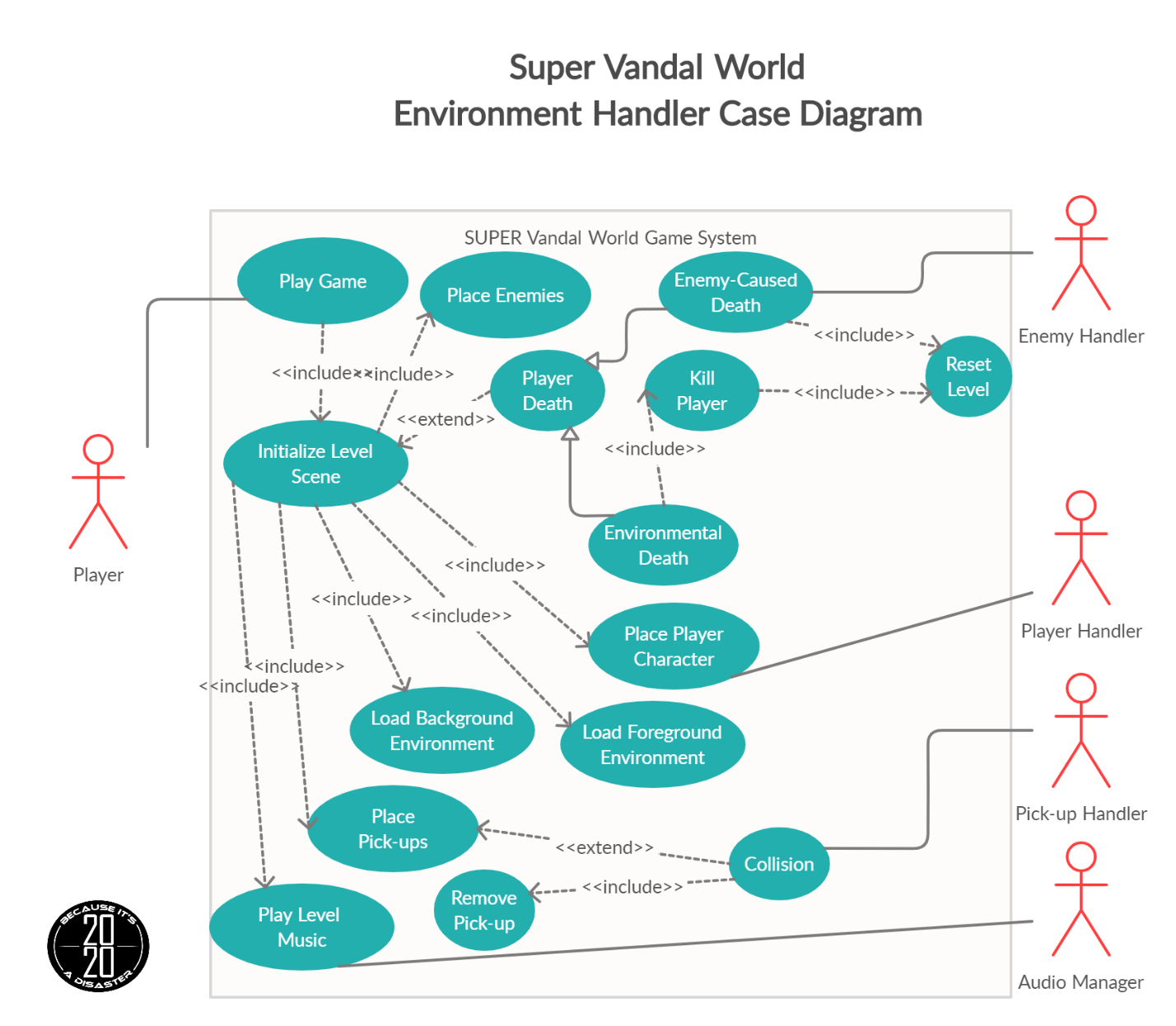
Name: Justin Harris Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

The feature that I will be handling is Level Design and Implementation. This will include determining the layout of each level, creating and placing the background and foreground assets necessary for each level, placing the player character, placing enemies in a pre-determined location, placing pick-ups at a pre-determined location, and calling and playing music for the level. I will also be implementing environmental mechanics, such as traps or hazards, and resetting the level on the player’s death, and destroying pick-up assets when they collide with the player character.

## Use case diagram with scenario \_\_14

### Use Case Diagram



### Scenarios

**Name:** Initialize Level Scene

**Summary:** Initializes the level including: The foreground, background, character, pick-up and enemy assets.

**Actors:** Player

**Preconditions:** Game has been initialized

**Basic sequence:**

**Step 1:** Loads the Background Environment

**Step 2:** Loads the Foreground Environment

**Step 3:** Load Main Character

**Step 4:** Initializes Pick-ups and Enemies.

**Step 5:** Initializes camera to main character

**Step 6:** Calls for level specific audio song to be played from the Audio Manager.

**Step 7:** Continue to monitor for player collision with the environment objects.

**Step 8:** Continue to monitor for collision states with pick-ups and enemies from the Pick-up Handler and the Enemy Handler.

**Exceptions:**

**Step 1:** Player has collided with an enemy and has died, based on the signal from the Enemy Handler: Reset Level

**Step 2:** Player has collided with an environmental object that causes the player to register a “hit” (i.e. Falling off the scene, falling into a trap): Reset Level

**Step 3:** Player has collided with a pick-up object, as determined by the Pick-up Handler: Remove pick-up / power-up asset from the scene.

**Post conditions:** The level is started, the player has control of the player character, level music is player, background/foreground animations are playing, and enemies and pick-ups are enabled to move.

**Priority:** 1

**ID:** JH5.1

**Name:** Load Background Environment

**Summary:** Initializes background scenery.

**Actors:** Player

**Preconditions:** Level has been called to initialize

**Basic sequence:**

**Step 1:** Load Background Scenery

**Step 2:** Play background animations

**Exceptions:**

**Step 1:** No background animations are in the level: Do nothing.

**Post conditions:** The background scenery is loaded and playing any animations.

**Priority:** 2

**ID:** JH5.2

**Name:** Load Foreground Environment

**Summary:** Initializes foreground environment assets

**Actors:** Player

**Preconditions:** Level has been called to initiate

**Basic sequence:**

**Step 1:** Initialize foreground assets that the player character will play the game on.

**Step 2:** Play animations happening in the foreground in the scene

**Exceptions:**

**Step 1:** No foreground animations are in the level: Do nothing.

**Post conditions:** Foreground assets are initialized and playing animations.

**Priority:** 1

**ID:** JH5.3

**Name:** Place enemies

**Summary:** Initializes enemy assets in the level.

**Actors:** Player

**Preconditions:** Game has been initialized

**Basic sequence:**

**Step 1:** Initialize enemies that have been placed in the level

**Exceptions:**

None. There are enemies present in every level.

**Post conditions:** The level is started, and enemy movements have been enabled.

**Priority:** 2

**ID:** JH5.4

**Name:** Place Pick-ups

**Summary:** Initializes pick-up assets in the level, including places where power-up type pick-ups will be hidden in the scene.

**Actors:** Player

**Preconditions:** The level has been called to be initialized.

**Basic sequence:**

**Step 1:** Initialize pick-ups and power-ups in the level.

**Step 2:** Enable animations for pick-ups and power-ups to be played.

**Step 3:** Monitor for collision signals from the Pick-up Handler.

**Exceptions:**

**Step 1:** Pick-ups in the level do not have an animation: do nothing.

**Step 2:** A signal for a collision has been received from the Pick-up Handler, remove the pick-up asset that was collided with from the scene.

**Post conditions:** The level is started, pick-ups and power-ups are able to be interacted with by the player character.

**Priority:** 2

**ID:** JH5.5

**Name:** Play Level Music

**Summary:** Requests music from the Audio Manager to play the audio corresponding to the level number.

**Actors:** Player

**Preconditions:** The level has been initialized

**Basic sequence:**

**Step 1:** Send a request to the Audio Manager, containing the level number, to play the level music.

**Exceptions:**

None. Every level should have music audio that should play for the duration of the level.

**Post conditions:** The level is started and the corresponding music clip for the level is playing.

**Priority:** 3

**ID:** JH5.6

**Name:** Player Death

**Summary:** The player has died due to an encounter with an enemy or an environment hazard.

**Actors:** Player

**Preconditions:** Player has collided with an enemy or an environment hazard.

**Basic sequence:**

**Step 1:** Signal has been received from the Enemy Handler indicating that the player has died. Or, the player has had a collision with an environment hazard.

**Step 2:** In the case where a signal has been received from the Enemy Handler: Reset the level.

**Step 3:** In the case where a player has died due to the environment: Kill the player.

**Step 4:** Reset the level.

**Exceptions:**

None.

**Post conditions:** The level has been reset and the player is back at the start of the level.

**Priority:** 2

**ID:** JH5.7

**Name:** Reset Level

**Summary:** Resets the current level from the beginning. Re-initialize the level scene.

**Actors:** Player

**Preconditions:** Player death has occurred.

**Basic sequence:**

**Step 1:** Loads the Background Environment

**Step 2:** Loads the Foreground Environment

**Step 3:** Replace and Initialize Pick-ups and Enemies.

**Step 4:** Call for level specific audio song to be re-started from the Audio Manager.

**Step 5:** Continue to monitor for player collision with the environment objects.

**Step 6:** Continue to monitor for collision states with pick-ups and enemies from the Pick-up Handler and the Enemy Handler.

**Exceptions:**

None.

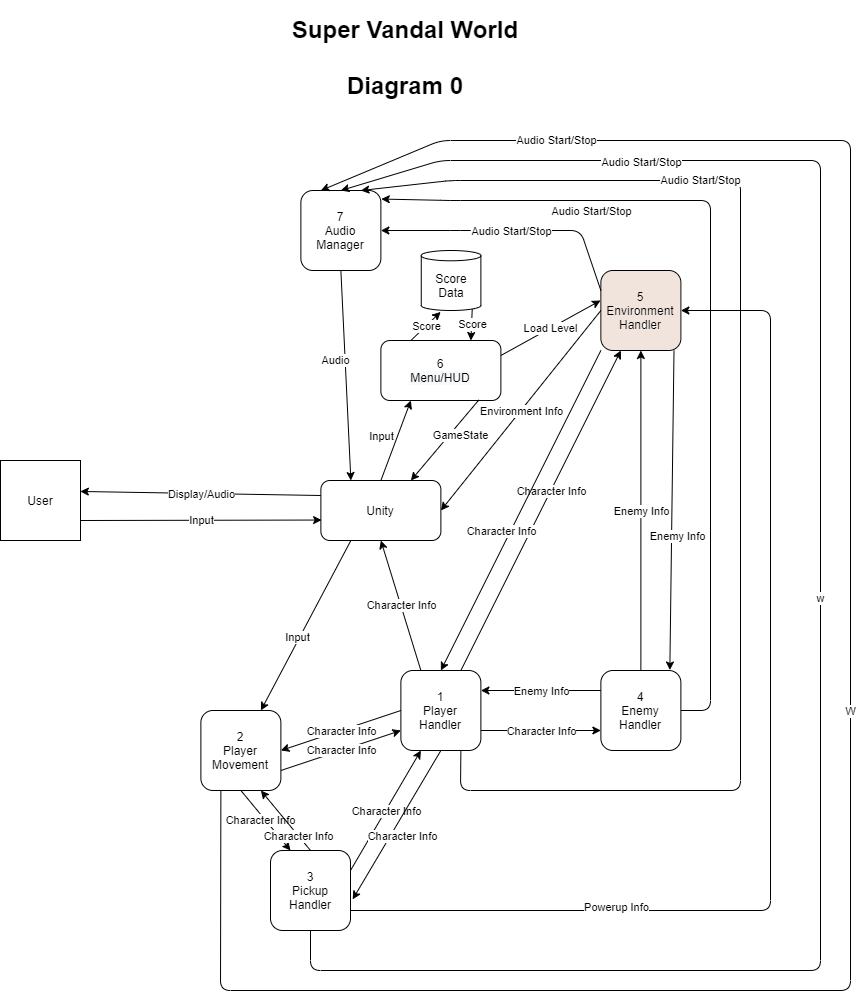
**Post conditions:** The level has been restarted, the player has control of the player character, level music is player, background/foreground animations are playing, and enemies and pick-ups are enabled to move.

**Priority:** 1

**ID:** JH5.8

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

### Data Flow Diagrams



### 

### Process Descriptions

Initialize Level:

WHILE GameState is true

Get Current Level

IF Current Level is level 0

Set Current Level to level 1

END IF

END WHILE

Set Player Death to false

Get Background Animations for Current Level

Get Foreground Animations for Current Level

Get Enemy Positions

Send Level Music Request

IF Pick-up collision is true

Remove pick-up from scene

END IF

IF Player Death is set to true

Reset Level

END IF

Load Background Environment:

IF Background has animations

Enable animations

END IF

Load Foreground Environment:

IF Foreground has animations

Enable animations

END IF

Place Enemies

IF GameState is true

Get Current Level

END IF

Return Enemy Locations

Place Pick-ups

Poll Pick-up Status

IF Collision with pick-up happens

Get coordinates for pick up

Remove pick-up from level

END IF

Player Death

IF Player death is true

IF Caused from Environment collision

Kill Player

Set Reset Level to true

END IF

ELSE

Set Reset Level to true

END ELSE

Reset Level

IF Player death is true

Disable Player Controls

Clear current level data

Initialize Level

END IF

## Acceptance Tests

Test Starting Background Animations a minimum of 25 times and output to a file:

* List background animation name
  + Report True if successfully started
  + Report Fales if not successfully started

Test Starting Foreground Animations a minimum of 25 times and output to a file:

* List foreground animation name
  + Report True if successfully started
  + Report Fales if not successfully started

Test getting pick-up starting locations a minimum of 25 times and output to a file:

* If successful: report pick-up ID and location
* If unsuccessful: report pick-up ID and “error” or false

Test getting enemy starting locations a minimum of 25 times and output to a file:

* If successful report enemy id and location
* If unsuccessful: report enemy id and “error” or false

Test collision of player with standard level terrain a minimum of 25 times and output to a file:

* Player successfully collided with each standard level terrain: report True
* If the player did not successfully collide with standard terrain: report False
  + Additionally, report coordinates(x,y) of where the player did not successfully collide, indicating the player fell through the terrain.

Test player collision with the hazard terrain in the level, that would result in player death a minimum of 25 times and output to a file:

* Player successfully collided with each hazard terrain in the level: report True
* If the player did not successfully collide with hazard terrain: report Falst
  + Additionally, report coordinates (x,y) of where the player did not successfully collide, indicating the player fell through the terrain.
* Test if player was killed on successful contact with each hazard terrain in level: report True
* If the player was not successfully killed on collision with hazard terrain: report False
  + Additionally, report coordinates (x,y) of where the player was not killed on collision.

Test Playing level music on each level a minimum of 25 times and output to a file:

* Level Number / Name that requested music
* Name of music audio file that was played during the level
* Compare requested audio with returned audio
  + If both match report True
  + If either don’t match, report false

Test removal of Pick-up items from level scene a minimum of 25 times/pick-up and output to a file:

* Report pick up coordinates (x,y)
* If asset removal was successful report True
* If asset removal was unsuccessful report False

Test Resetting current level a minimum of 25 times and output to a file. The file will contain the following information:

* Current level number/name
* Clear score data: report true or false
  + Report score value before level reset and after level reset
* Reset level timer (if implemented): report true or false
  + Report timer value before level reset and after level reset
* Enemies re-positioned/re-created in level: report true or false
  + Report enemy ID’s followed by coordinates (x,y) enemy was placed at
    - Compare coordinates against enemy starting location defaults
* Pick-ups re-created in level: report true or false
  + Report pick-up ID followed by coordinates (x,y) the pick up was placed at
    - Compare coordinates against pick-up starting location defaults
* Level Music name
  + Report that level music was successfully stopped: report True or False
  + Report that level music was successfully requested: report True or False
  + Compare requested audio with returned audio
    - If both match report True
    - If either don’t match, report false
* Player position reset
  + Report player’s coordinates (x,y) before reset was called
  + Set player’s coordinates to default starting value for the level
    - Report True if successful
    - Report False if unsuccessful
* Disable and re-enable player controls
  + Player controls disabled at the beginning of Reset call
    - Report True if successful; False if unsuccessful
  + Player controls re-enabled at the end of the Reset call
    - Report True if successful; False if unsuccessful

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

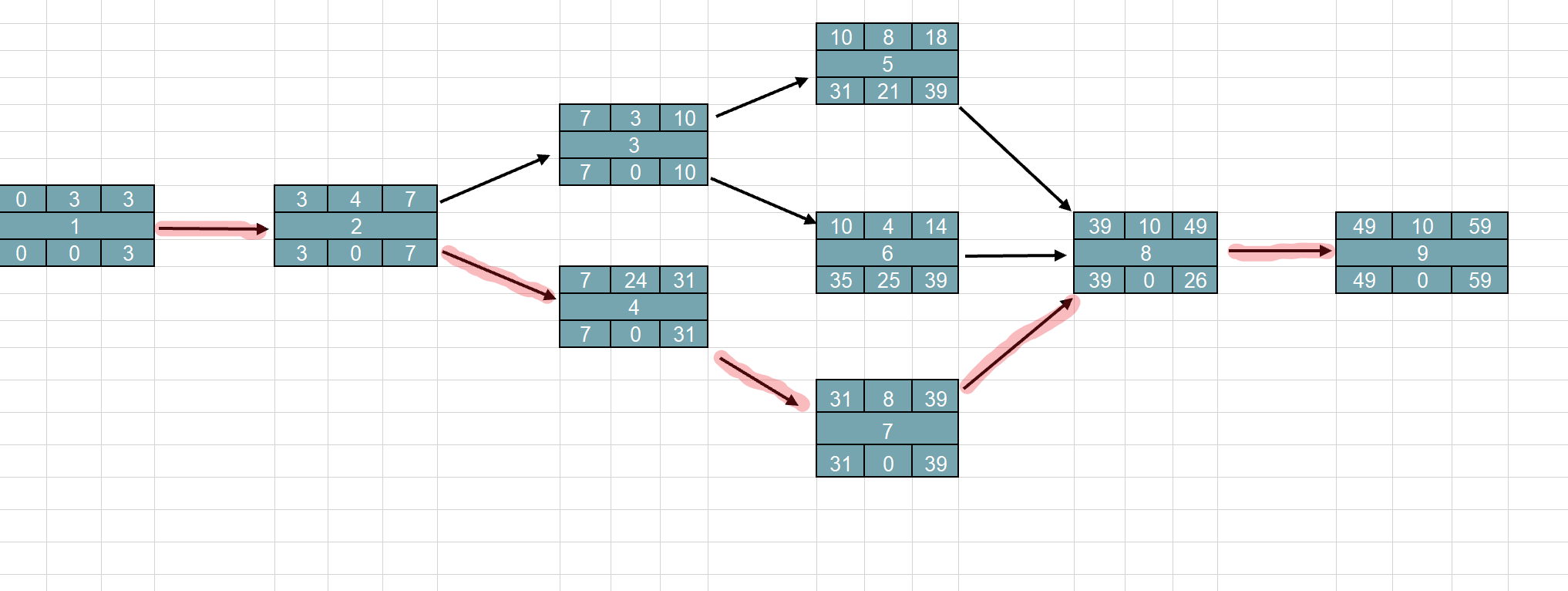
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Requirements Collection | 3 | - |
| 2. Design Levels | 4 | 1 |
| 3. Generate Camera | 3 | 2 |
| 4. Create Scenery/Terrain | 24 | 2 |
| 5. Track Player/Enemies/Pick-ups | 8 | 3 |
| 6. Request and Play Music | 4 | 3 |
| 7. Create Animations for Scenery | 8 | 4 |
| 8. Reset Level | 10 | 5,6,8 |
| 9. Testing | 10 | 10 |
| Total | 78 | - |

### Pert diagram



### Gantt timeline

