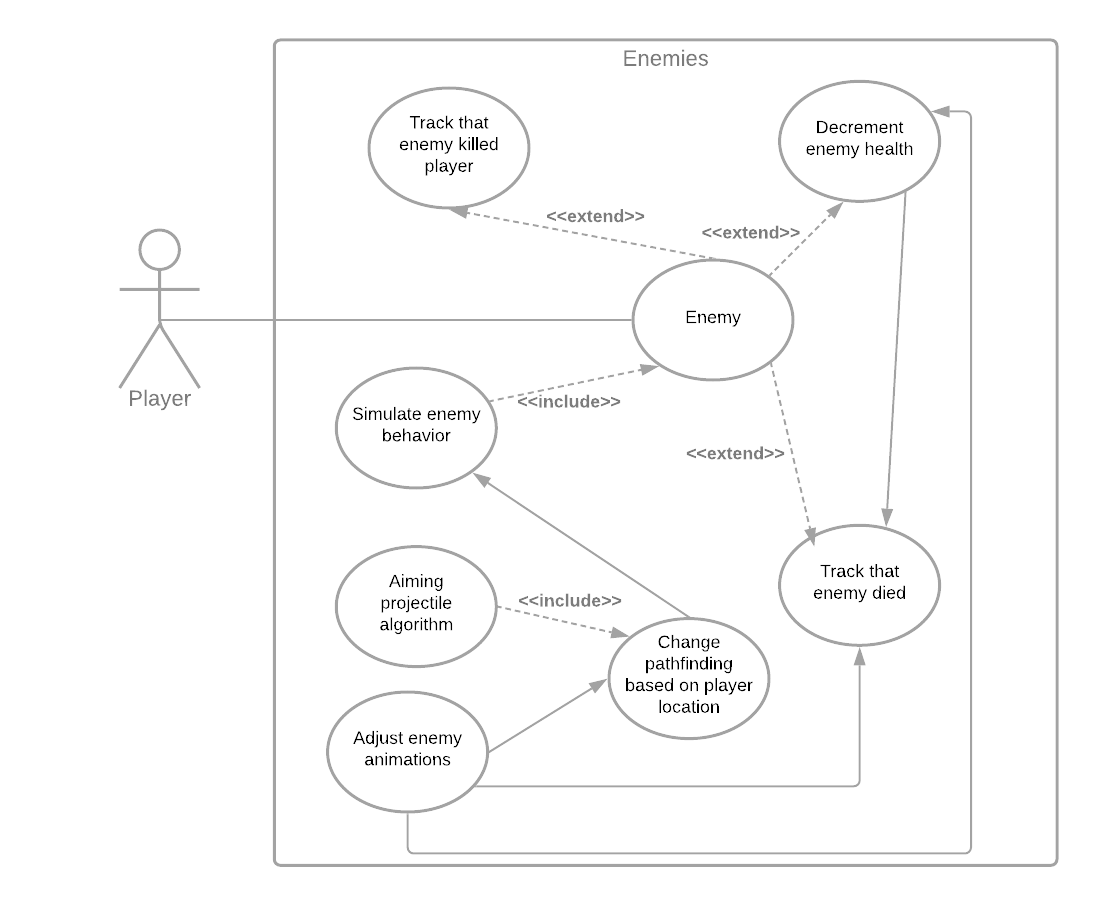
yName Zachary Hale\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

I am doing the character and enemy design for the game. I will be doing the animations and some of the control of the enemies

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

**Name:** Enemy

**Summary:** Telling what a certain enemy should do and when to do it. Where to go during an animation and which direction to face.

**Actors:** Player

**Preconditions:** Player is within the view of the enemy

**Basic sequence:**

**Step 1:** Enemy sees the player

**Step 2:** Animate turning and facing the player and begin pathing

**Step 3:** Fire at or run after the player

**Step 4:** Change pathing based on player position

**Step 5:** Deal damage to the player.

**Exceptions:**

**Step 1:** Player kills the enemy, removing it.

**Step 2:** Enemy kills the player, replacing it back in it’s original position in the level

**Step 3:** Enemy is hit and displays a getting hit animation and has knockback

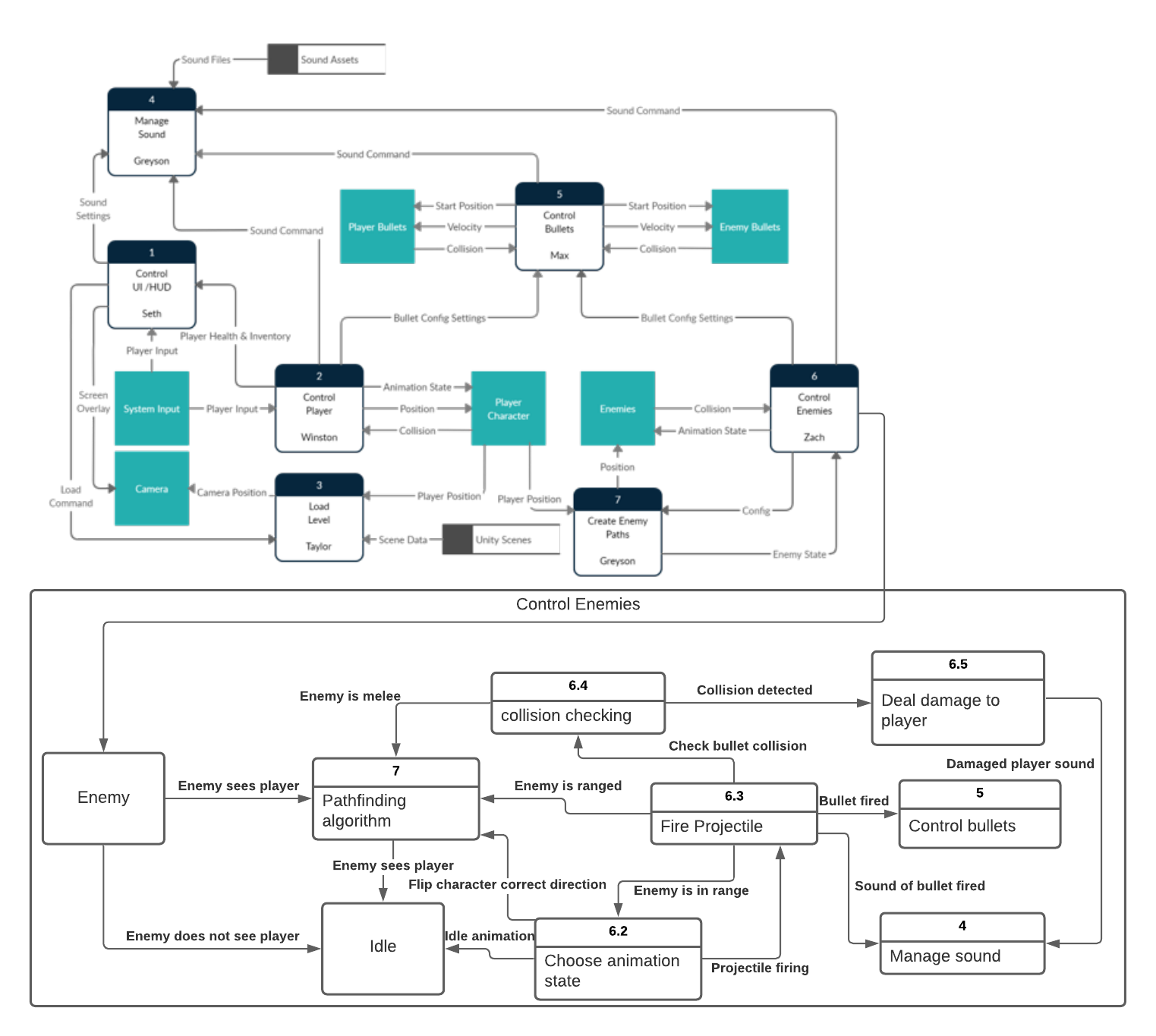
**Post conditions:** Dies or kills player

**Priority:** 1

**ID:** EB1

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

### Data Flow Diagrams



### Process Descriptions

Collision checking:

WHILE bullet OR enemy is not colliding with player

do not cause damage

END WHILE

Fire Projectile:

WHILE enemy is in range of player

Fire projectile

END WHILE

Choose animation state:

WHILE enemy is alive

Choose animation state, based on the input from other functions

END WHILE

Deal damage to player:

IF player is colliding with bullet or enemy

deal damage proportional to said enemy difficulty

END IF

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

**Check to see if animations work properly**

Run each feature in different states 5 times each sending output to a file.

The output file will have the following characteristics:

* Current animation state
* Last input
* Check if last input and current animation state match
* Check distance from player
* No animation should be played at the incorrect time

**Collision checking**

Check if objects are within each other using functions given by unity, check for any possible problems in alignment through play testing. Play through at least twice.

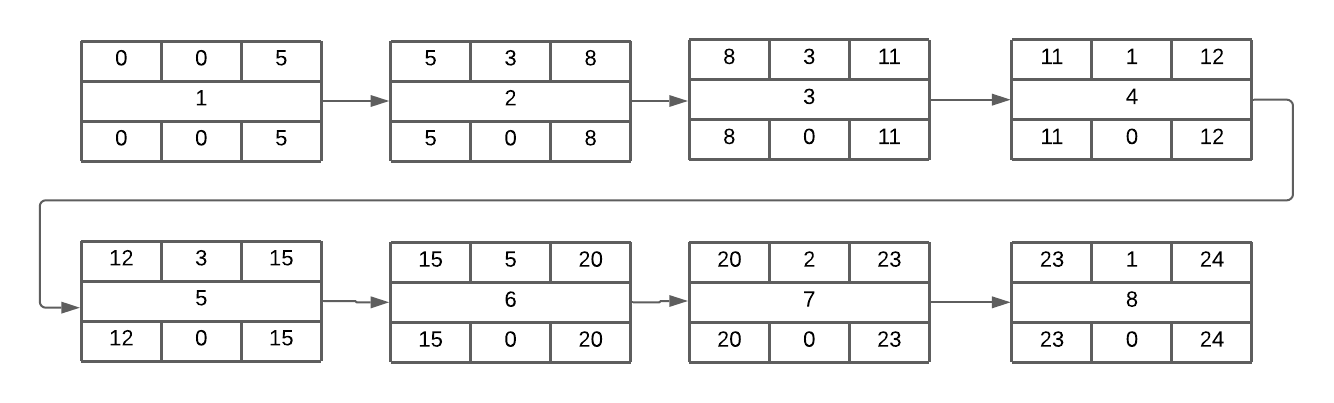
## Timeline \_\_\_\_\_\_\_\_\_/10

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Design Animations | 5 | - |
| 2. Design animation state setter | 3 | 1 |
| 3. Collision checking | 3 | 2 |
| 4. Implement damage communication | 1 | 3 |
| 5. Enemy class hierarchy | 3 | 4 |
| 6. Programming | 5 | 5 |
| 7. Testing | 2 | 6 |
| 8. Installation | 1 | 7 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |