Name\_\_ Molly Meadows\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature includes making the enemy sprites for our game. In this case they will look like zombie dogs.

By the end of the project, they will be controlled by an AI. They should move towards the character, attack, and change in difficulty and speed as the levels progress. I will be responsible for their attack sequences as well as controlling their health when they are collided with the main character’s bullets.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

Diagram

Description automatically generated

### Scenarios

**Name:** Collision with Bullets

**Summary:** The dog takes damage and moves backward when it collides with a bullet.

**Actors:** Enemy Dog

**Preconditions:** Game is initialized, and Zombie Dogs are spawned.

**Basic sequence:**

**Step 1:** Dog sprite collides with the bullet.

**Step 2:** Dog moves backward away from the character.

**Step 3:** Dog takes damage and reduces health bar.

**Step 4:** The sprite calls the damage sound.

**Exceptions:**

**Step 3.1:** If the dog has zero health left, it dies.

**Step 3.2:** The sprite will call the item manager and has a randomized chance to drop an item that the user can pick up.

**Post conditions:** Dogs disappear offscreen when they die.

**Priority:** 2

**ID:** MM1

\*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

Diagram

Description automatically generated

Context Diagram

Diagram

Description automatically generated

### Data Flow Diagrams

Level 1 Diagram

Diagram

Description automatically generated

### Process Descriptions

Move toward character(5.1):

Get Position from player ();

IF character is below dog

Move Down;

ELSE IF character is above dog

Move Up;

IF character is to the left of the dog

Move to the left;

ELSE IF character is to the right

Move to the right;

IF Zombie dog collides with the player

Call AttackPlayer();

AttackPlayer(5.2):

Call attack sound();

Attack animation();

Random call bite or scratch animation();

Collide with Bullets or Character Knife(5.3):

Remove Health;

Move Character Backwards;

IF health is at zero

Call Death of Zombie Dogs();

Death of Zombie Dogs(5.4):

Death Animation;

Make dogs disappear;

Call Random drop item;

IF all dogs in the round are dead and no cure

Call Level up Difficulty()

Level Up Difficulty(5.5):

Up the difficulty and speed of the dogs;

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

The acceptance tests will primarily be collision tests based on the users output of their attacks or if they cure the dogs.

The acceptance tests for the Zombie Dog feature are described below:

Collision with Bullets:

There will be a collision check to see if the bullets will damage the dog’s health. I will initiate a test 1000 times to see if the dogs take the correct amount of damage upon the collision of a bullet object.

Leveling up the Difficulty:

There will be a check to see if the number of initialized dogs equals the number of dogs that have died. If the numbers are equal, then the dogs will get a higher speed number and a stronger attack damage to the main character. There will also be a check to see if the character has put the cure into the machine which will be a flag check. If the cure is in the machine, then it will go to the cut scene and the dogs will no longer move.

Death of Dogs:

I will initiate a test 1000 times to see if the dogs’ sprites correctly play the death animation and then disappear when their health reaches to zero. I could have a test function that fires bullets at the dog.

Attack Sequences:

I will be able to test this function by calling it 500 times and see if it randomizes the attack sequences correctly.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Design Sprite | 3 Hours | - |
| 2. Create Dog class | 1 Hour | 1 |
| 3. Create Zombie Dog subclass | 1 Hour | 2 |
| 4. Animation | 20 Hours | 1 |
| 5. Sprite Attack | 6 Hours | 3, 4 |
| 6. Sprite Damage | 5 Hours | 3, 4 |
| 7. Death processes | 10 Hours | 6 |
| 8. Leveling Difficulty | 3 Hours | 5, 7 |
| 9. Sprite AI | 10 Hours | 8 |
| 10. Documentation | 2 Hours | 9 |
| 11. Testing | 6 Hours | 10 |

### Pert diagram

Graphical user interface, Word

Description automatically generated

### Gantt timeline

Timeline

Description automatically generated