Name: Bryce Hendrickson Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

My feature for the Zombie Dog Outbreak 2 game is the character control, movement, design, animation, and weapon control and interactions.

My job is to create a character sprite equipped with responsive movement and attack(melee) functions that will interact with the enemies in the game. I will also be responsible for controlling player health when attacked. As well as the BC character mode (character subclass) that cannot die and will have different animations based off when they should be dead.

## Use case diagram with scenario \_\_14

### Use Case Diagram

Diagram

Description automatically generated

Scenario:

Name: Melee

Summary: The character activates the melee attack.

Actors: Character

Preconditions: player has been initialized.

Basic sequence:

Step 1: Input melee button

Step 2: Check for player motion and enemy detection

Exceptions:

(1)

Step 1: Character is moving toward enemy in range

Step 2: Character lunges with hit animation

(2)

Step 1: character is not moving toward an enemy in range

Step 2: Checks for collision with enemy

Step 3: Plays associated melee animation

Post conditions: Character returns to previous animation / state.

Priority: 2\*

ID: MC1

\*The priorities are1 = 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

Diagram

Description automatically generated

Process Descriptions

Reload weapon:

IF (the equipped guns mag != full and there are reserve bullets)

THEN Play reload animation() & update gun inventory()

Else ignore input

Shoot:

IF (the equipped guns mag > 0)

THEN spawn bullet in direction of gun

Else play empty gun click sound and call Reload weapon()

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

Input movement (WASD)

Check for collision up when W

Check for collision left when A

Check for collision down when S

Check for collision right when D

Call melee() when enemy is close and directional input matching enemy direction is pushed.

Check for lunge function to be called

Check for collision with enemy

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

[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | Duration (Hours) | Predecessor Task(s) |
| 1 | Sprite creation | 1 | - |
| 2 | Character artwork & animations | 20 | 1 |
| 3 | Movement | 3 | 1 |
| 4 | Melee | 5 | 1, 3 |
| 5 | Animation controller | 3 | 1,2 |
| 6 | Weapon sprite holder creation | 2 | 1 |
| 7 | bullet creation | 4 | 1,6 |
| 8 | Bullet interaction | 6 | 1,6,7 |
| 9 | Weapon inventory system | 5 | 6,7 |
| 10 | Reload animation and inventory implementation | 3 | 1,2,5,6,7,9 |

### Pert diagram

Diagram, engineering drawing

Description automatically generated

### Gantt timeline

