

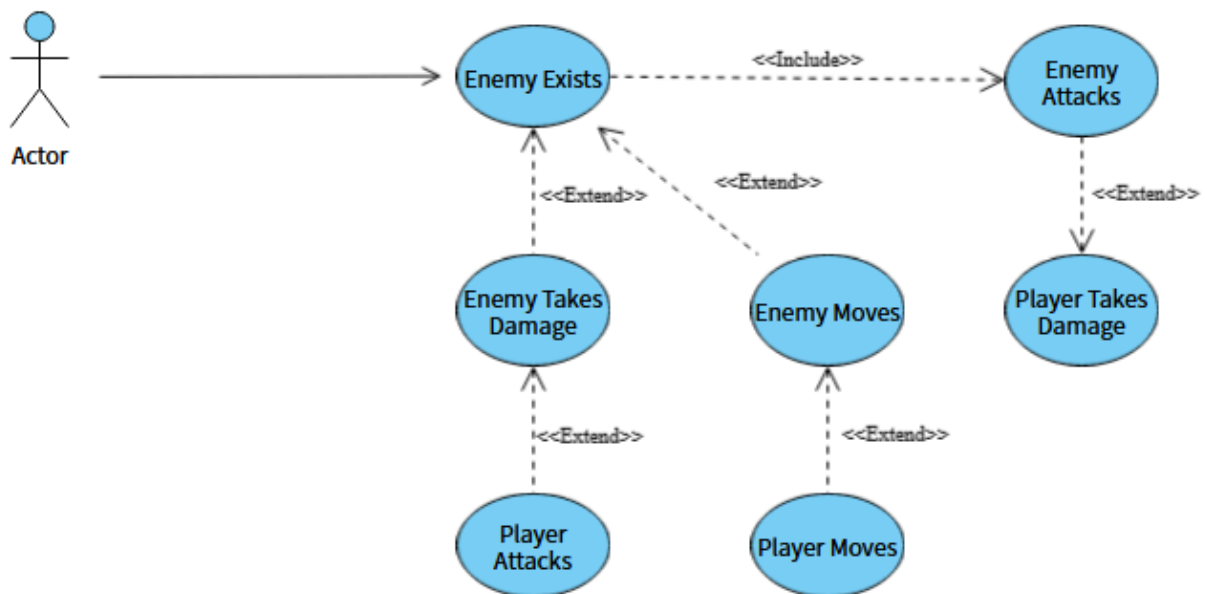
1. Brief introduction __/3

My feature for the game is the enemy AI and enemy variety. This includes designing and implementing different enemy movement and attack patterns to create engaging and challenging gameplay. I am responsible for ensuring that enemies behave uniquely based on their type, reacting to the player and the environment in dynamic ways.

Additionally, I am in charge of creating the art for the game and level design, making sure that the visuals and layouts complement the enemy behaviors.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Enemy AI

Summary: The enemy AI is active

Actors: Enemy, Player

Preconditions: The enemy is spawned in and on screen.

Basic sequence:

Step 1: The enemy moves tracking player movement.

Step 2: The enemy will always try to attack the player.

Step 3: If the enemy's attack hits the player, the player takes damage.

Exceptions:

Step 1: The player attacks the enemy.

Step 2: If the player's attack hits the enemy, the enemy takes damage.

Step 3: If the enemy takes too much damage, it stops existing.

Post conditions: The enemy is dead

Priority: 2*

ID: C02

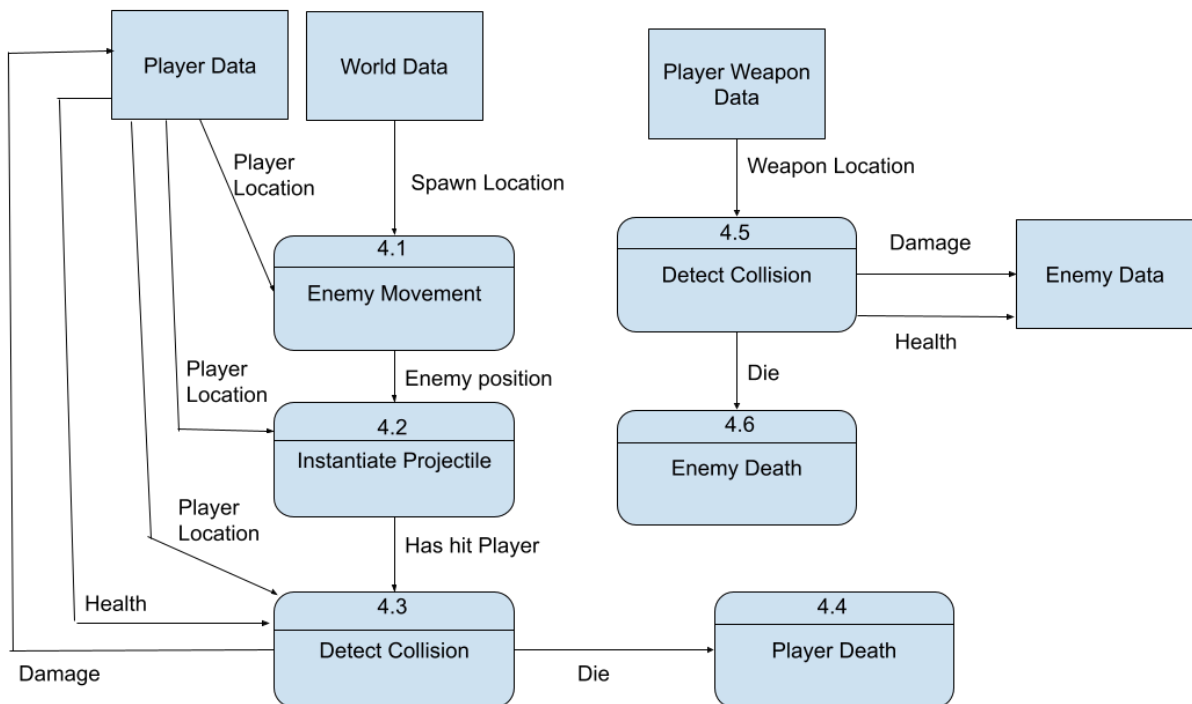
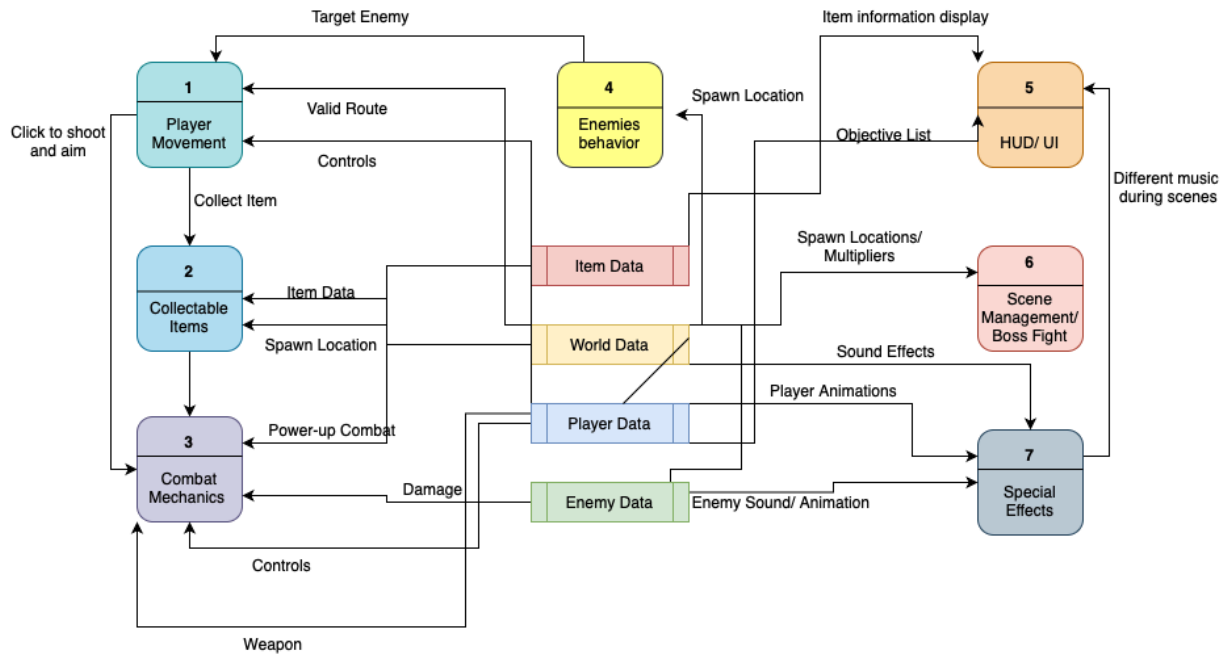
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

Data Flow Diagrams



My process is process 4, highlighted in yellow.



Process Descriptions

Move Towards Player:

```
WHILE enemy is active in the scene
    Set enemy's target position to the player's position
    Move enemy towards the player's position
END WHILE
```

Evade Player:

```
IF the enemy is within a defined distance of the player
    Determine the opposite direction from the player
    Move enemy away from the player's position
END IF
```

Attack Player (Melee):

```
IF enemy is colliding with the player
    Trigger damage event on the player
END IF
```

Attack Player (Ranged):

```
IF the enemy has a ranged attack available
    Instantiate a projectile at enemy's position
    Set projectile's trajectory towards the player
END IF
WHILE projectile is active
    IF projectile collides with the player
        Trigger damage event on the player
        Destroy projectile
    END IF
END WHILE
```

Enemy Takes Damage:

```
WHEN player's attack collides with the enemy
    Reduce enemy's health by 1
END WHEN
```

1. Enemy Moves Towards Player

Test Input:

- Place the enemy and player at random positions on the map.
- Ensure the player is within enemy detection range.

Expected Output:

- The enemy moves toward the player.
- The enemy's position updates each frame in the direction of the player.
- If the enemy reaches the player, movement stops.

Boundary Cases:

- If the player is already touching the enemy, movement does not occur.
- If the player moves, the enemy continuously updates its path.

2. Enemy Evades Player

Test Input:

- Place the enemy near the player within the defined evasion distance.

Expected Output:

- The enemy moves in the opposite direction from the player.
- If the enemy moves outside the evasion distance, it stops fleeing.

Boundary Cases:

- If the enemy is exactly at the edge of the evasion distance, no movement occurs.
- If there is an obstacle behind the enemy, it attempts to find another escape route.

3. Melee Attack

Test Input:

- Place the enemy adjacent to the player.

Expected Output:

- If the enemy collides with the player, the player's health decreases.
- If the enemy is not colliding with the player, no damage occurs.

Boundary Cases:

- If the enemy reaches the player but is not coded to attack, no damage occurs.
- If the player moves out of range before the collision check, no damage occurs.

4. Ranged Attack

Test Input:

- Have an enemy with a ranged attack target the player.

- Fire 1000 projectiles towards the player.

Expected Output:

- Each projectile travels toward the player's last known position.
- If a projectile collides with the player, the player takes damage.
- If the projectile misses, it disappears after a set distance or duration.

Boundary Cases:

- If the player moves after the projectile is fired, the projectile does not adjust its trajectory.
- If a projectile hits an obstacle before reaching the player, it does not deal damage.
- If a projectile collides with another projectile, they do not interfere with each other.

5. Enemy Takes Damage

Test Input:

- Have the player attack an enemy with a weapon or ability.
- Perform 10 consecutive hits on an enemy with 10 health.

Expected Output:

- The enemy's health decreases by 1 per attack.
- If the enemy's health reaches 0, it is removed from the scene.

Boundary Cases:

- If the player attacks but does not connect, no health is lost.
- If an enemy is already dead, additional attacks do nothing.
- If multiple players attack simultaneously, damage is registered per hit.

5. Timeline _____/10

[Figure out the tasks required to complete your feature]

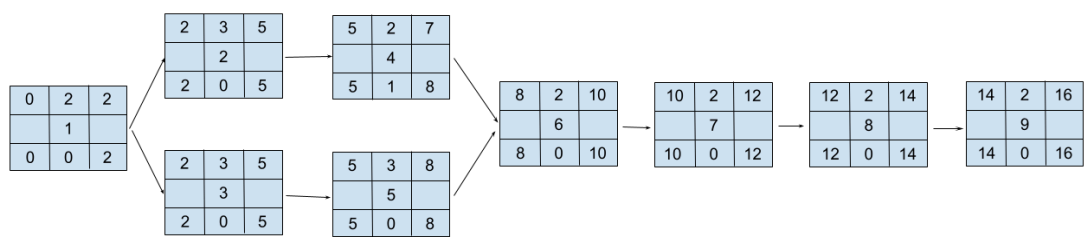
Example:

Work items

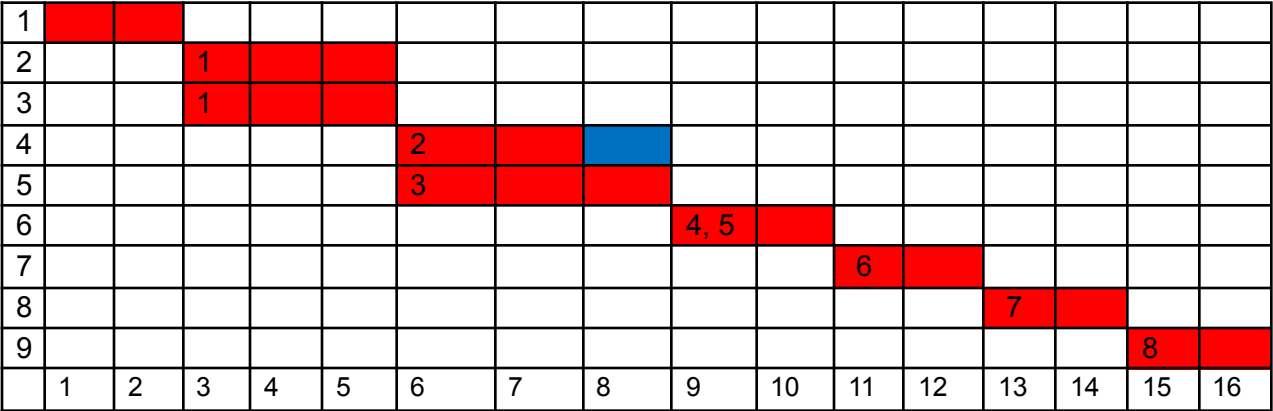
Task	Duration (PWKs)	Predecessor Task(s)
1. Requirements Collection	2	-
2. Enemy Movement Logic	3	1

3. Enemy Evasion Logic	3	1
4. Melee Attack Implementation	2	2
5. Ranged Attack Implementation	3	3
6. Enemy Health & Damage System	2	4,5
7. Testing	2	6
8. Bug Fixing & Adjustments	2	7
9. Integration	2	8

Pert diagram



Gantt timeline



Wrk Hrs	Slack
█	█