Name: David Trail Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature is the enemy interactions for Hot Dog Jones: THE VIDEO GAME. Players will face enemies spawning at different speeds, creating a challenging obstacle between them and victory. The enemy interactions revolve around the player and enemy health and deaths. This feature promises a unique and engaging experience, where overcoming obstacles becomes a test of skill and adaptability.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

A diagram of a game

Description automatically generated

### Scenarios

**Scenario 1**

**Name:** Enemy advantage

**Summary:** The Player Interacts with an enemy and gets attacked.

**Actors:** Player.

**Preconditions:** The game is running, and the player is in the main menu.

**Basic sequence:**

**Step 1:** The level begins and the enemies spawn.

**Step 2:** The player spots an enemy on screen.

**Step 3:** The enemy touches the player, and the players health is decreased. The players health bar will be updated accordingly.

**Exceptions:**

**Step 1:** If all of the players health is depleted, they will die.

**Step 2:** The current game ends.

**Post conditions:** The player returns to the main menu.

**Priority:** 1

**ID:** C05.1

**Scenario 2**

**Name:** Player advantage

**Summary:** The Player Interacts with an enemy, and attacks the enemy

**Actors:** Player.

**Preconditions:** The game is running, and the player is in the main menu.

**Basic sequence:**

**Step 1:** The level begins and the enemies spawn.

**Step 2:** The player spots an enemy on screen.

**Step 3:** The players projectile (hotdog) hits the enemy and the enemy’s health decreases.

**Exceptions:**

**Step 1:** If all the enemy’s health is depleted, they will die.

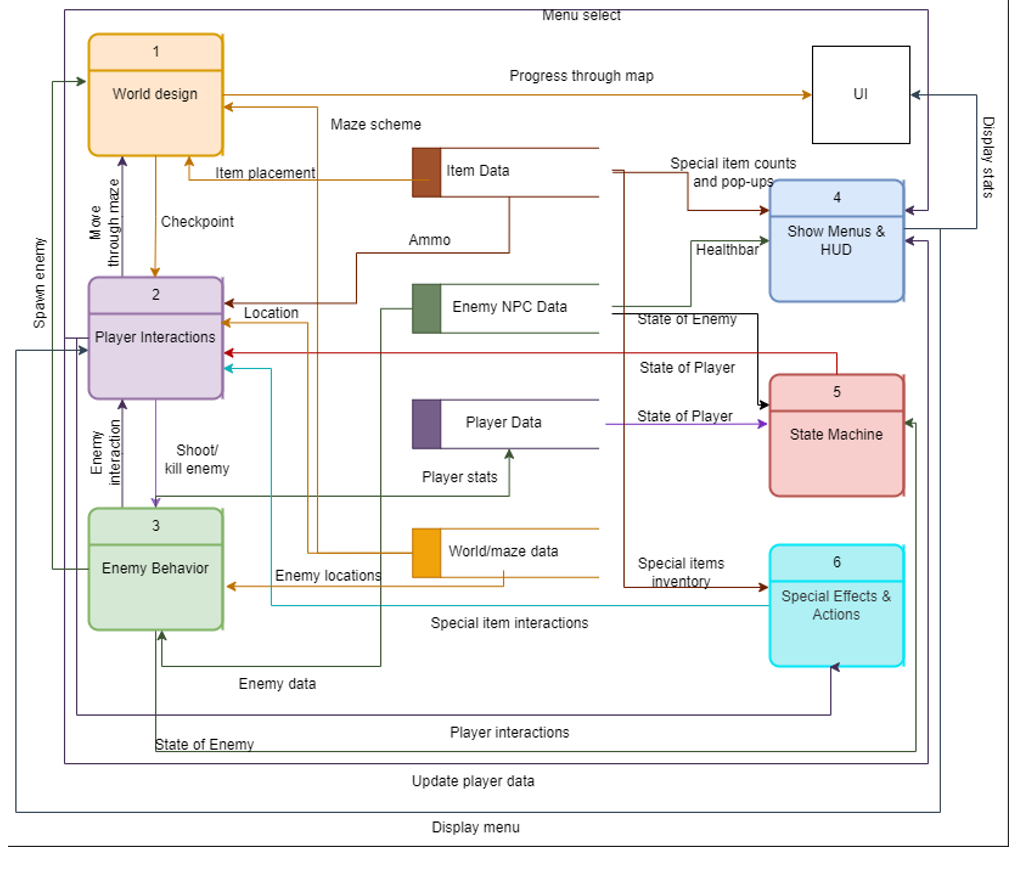
**Post conditions:** The enemy count will be updated.

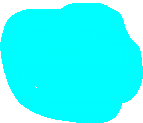
**Priority:** 1

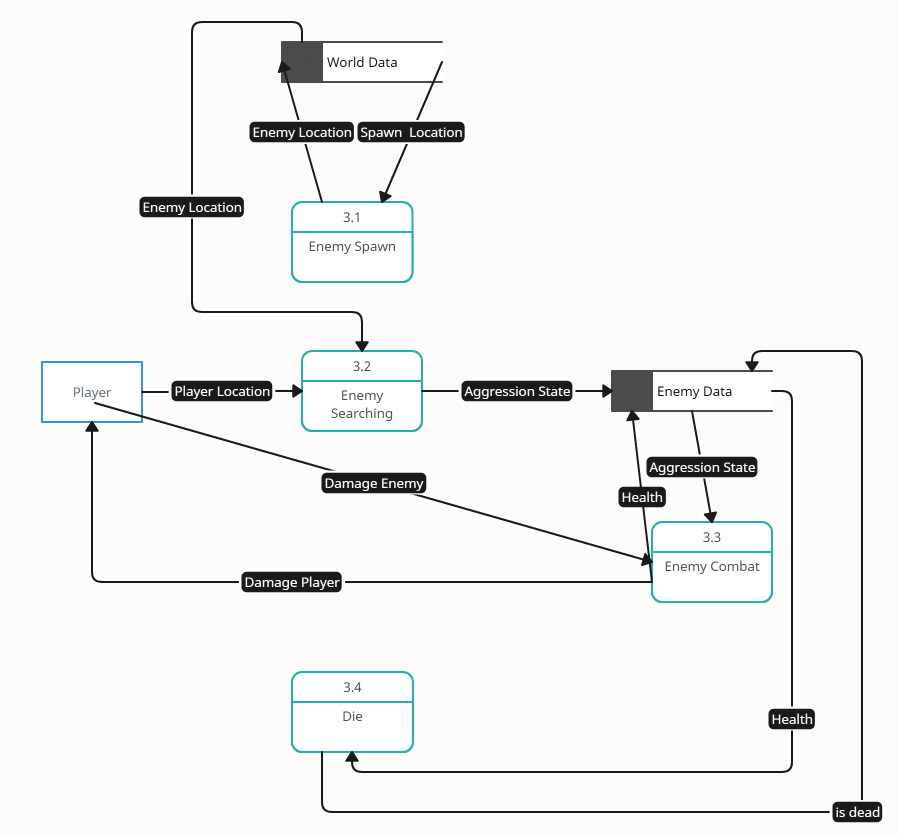
**ID:** C05.2

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

### Data Flow Diagrams





**Enemy Behavior**

### Process Descriptions

Enemy Spawn:

IF location is spawnable && location is loaded

Spawn Enemy

Store Enemy Location in world data

END IF

Enemy Searching :

IF Enemy Location & Player location are close enough to each other

Set Aggression State to true

Store Aggression state to enemy data

END IF

Enemy Combat:

Health = 3

WHILE aggression state = true

Move towards player location

IF at player location

Damage player

Send data to player

Wait 1 second

END IF

IF projectile hits enemy

Subtract 1 health point

Store enemy health in enemy data

END IF

END WHILE

Die:

IF enemy health == 0

Delete enemy

Send is dead to enemy data

END IF

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

**Enemy: Spawn and Despawn test**

Send Spawn location data to enemies and “isdead” state immediately afterwards 100 times and evaluate if behavior is correct.

The appropriate behavior will have the following characteristics:

* Enemies will spawn in location given by the send data.
* Enemies will despawn after “isdead” state is set.
* When enemies despawn, they no longer exist in the game in any given form after deletion.

**Enemy: Aggression and Detection test**

Spawn Enemies within player range and check aggression state 100 times and evaluate if behavior is correct.

The appropriate behavior will have the following characteristics:

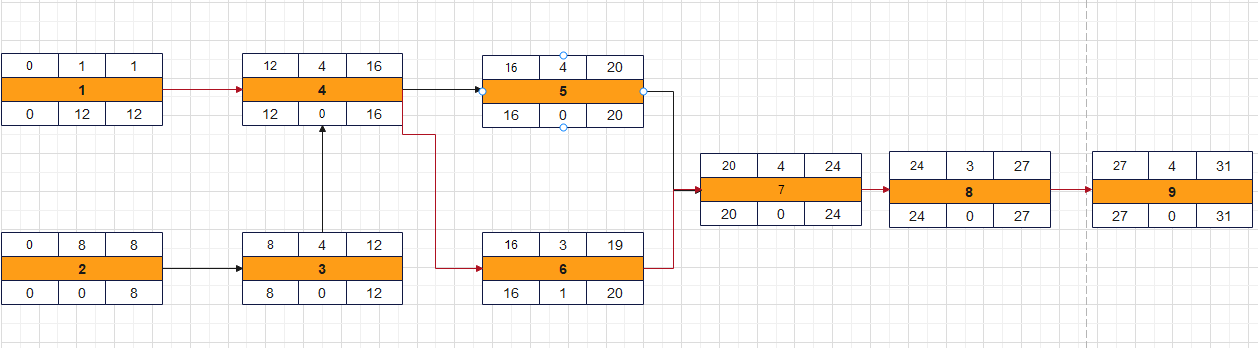
* Enemies react to players within a specified detection range.
* Aggression state is sent to enemy data.
* Enemies will move towards players once in aggression state.
* Enemies cause damage to player in aggression state.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Setting Up Unity | 1 | - |
| 2. Requirements Collection | 8 | - |
| 3. Sprite Aggregation | 4 | 2 |
| 4. Entity and States | 4 | 1,3 |
| 5. Enemy Centric Functions | 4 | 4 |
| 6. Enemy data storage | 3 | 4 |
| 7. Enemy Interaction Functions | 4 | 5, 6 |
| 8. Testing | 3 | 7 |
| 9. Integration | 4 | 8 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  | 1,3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5.6 |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |

