Jonathan Gift Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My primary feature is the navigation and interaction of the monster and how it relates to the player. I need to make sure that the monster can properly track the player, move between locations without glitching through walls, and chase the player in a way that is balanced for gameplay. I’ll also be in charge of the sounds that the monster makes as well as how it interacts with the player and the level itself.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

A close up of a map

Description automatically generated

### Scenarios

**Name:** Monster Chasing/Searching

**Summary:** The monster follows the player if it can spot them.

**Actors:** Player.

**Preconditions:** Player has been spotted by the monster.

**Basic sequence:**

**Step 1:** Set the target as the location of the player.

**Step 2:** Move towards the target.

**Step 3:** Once the monster reaches the target, scan for a new target.

**Step 4:** Repeat until the monster gets the player or the player runs away.

**Exceptions:**

**Step 1:** Player escapes vision range: remove from target in queue.

**Post conditions:** Target is removed from queue

**Priority:** 1

**ID:** M01

**Name:** Control Object

**Summary:** The monster opens a door or disables a light.

**Actors:** Player.

**Preconditions:** Door/light is in the list of targets for the monster.

**Basic sequence:**

**Step 1:** Move towards interactable object.

**Step 2:** If object is a light, disable it. If object is a door, open it.

**Step 3:** Proceed to the next point in the monsters’ target queue.

**Exceptions:**

**Step 1:** Light is already disabled, or door is already open.

**Post conditions:** Light is disabled, or door is opened.

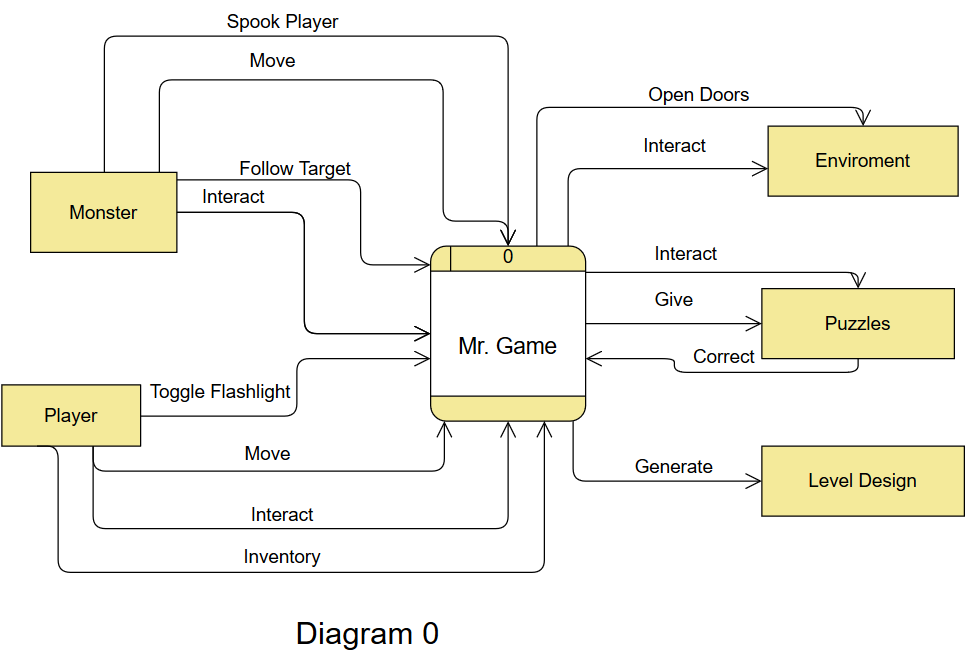
**Priority:** 3

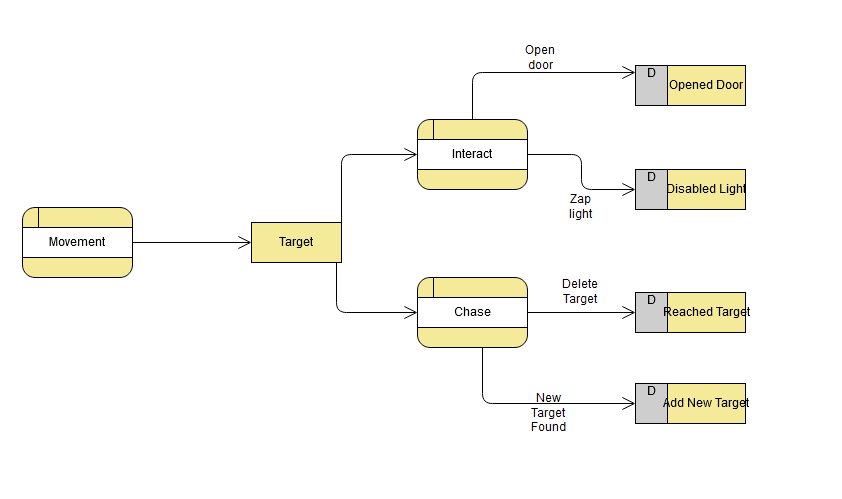
**ID:** M02

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

My feature is the monster and its interactions with both the player and the environment itself.

### Data Flow Diagrams





### Process Descriptions

Search station:

WHILE player on station:

If no target:

Generate random target()

else:

pathToTarget(target)

if hear noise OR spot player:

target = player’s location at time of if statement.

END WHILE

pathToTarget(vector3 target):

if door block way:

damage door

else if area is bright:

damage lights

else

pathfind to target

END FUNC

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

Test 1: Monster chases player through station.

Success: Monster does not warp through walls, doors, correctly reaches destination.

Test 2: Monster interacts with doors and lights.

Success: If a door is blocking the monster, pathfinding will pause until the door is opened. Lights will disable pathfinding until they are deactivated.

Test 3: Monster correctly listens for and sees the player.

Success: Target is updated in the event that the player makes a loud noise or is in the view angle of the monster.

Test 4: Monster patrols station.

Success: Monster is never idle, moves between seemingly random locations.

Test 5: Monster can attack player.

Success: Player is attacked by monster and must restart.

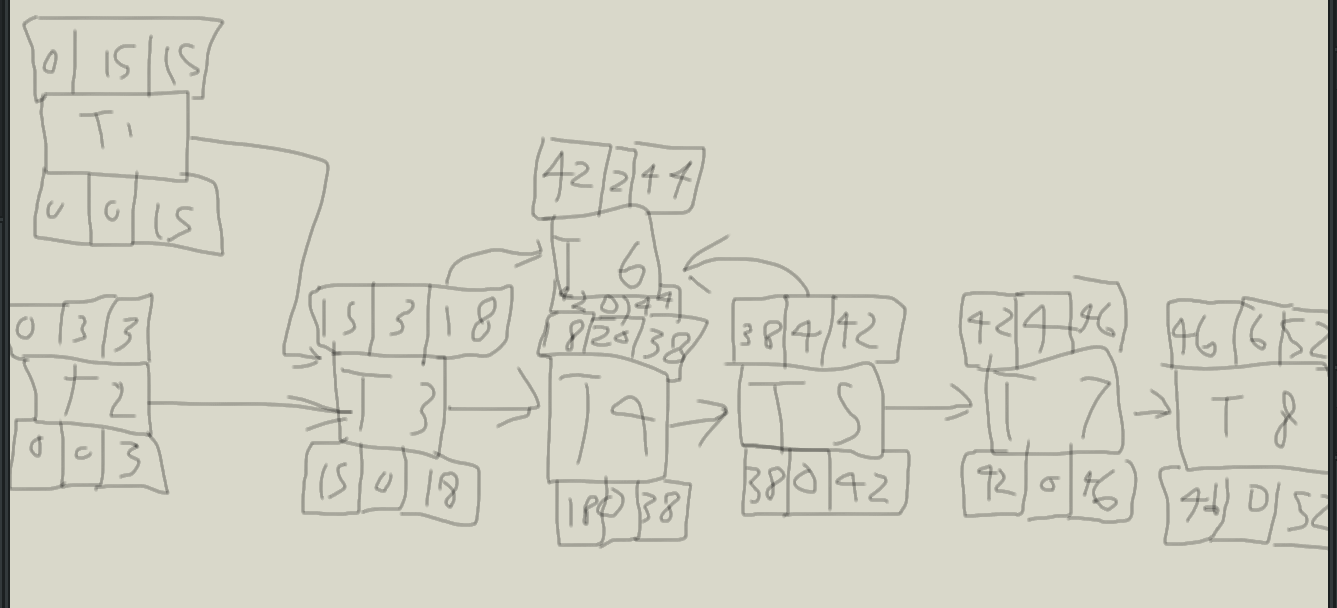
## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Team planning | 4 | - |
| 2. Asset research | 1 | - |
| 3. Asset dispersal | 1 | 1,2 |
| 4. Program monster, coordinate with level designer and player dev | 4 | 3 |
| 5. Test monster cases | 2 | 4 |
| 6. Repeat 4, 5 | 1 | 3, 5 |
| 7. Document code | 1 | 5 |
| 8. Installation | 1 | 7 |

### Pert diagram

Note: Work items, Pert, and Gannt are mostly identical between team members. I’m unsure as to how they should be different, need more information?



Gantt Timeline

