Name: Michael J Habermann Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

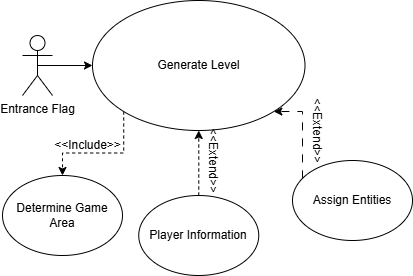
My feature for Ducks game Duck Quest is the level generation.

When the player starts the game from the main menu, they will start out in a house, which will consist of furnishings and when the player exits, they enter the overworld. The overworld will consist of at least one area which will be a town consisting of a church (generic no religious themes), a dungeon where the player can fight enemies, and townsfolk that will walk around.

All of the areas will be pre-made meaning there will not be any random generation for the rooms or the town. The dungeon will be where the player primarily explores as it will consist of several enemies, items, and where most of the core gameplay will happen.

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

**Name:** Generate the level

**Summary:** The game moving from one area to the next

**Actors:** Entrance Flag

**Preconditions:** None

**Basic sequence:**

**Step 1:** The player gives the game an entrance flag.

**Step 2:** The game spawns in an area (this could be the overworld, town, dungeon, etc.) based on conditions from which flag got triggered.

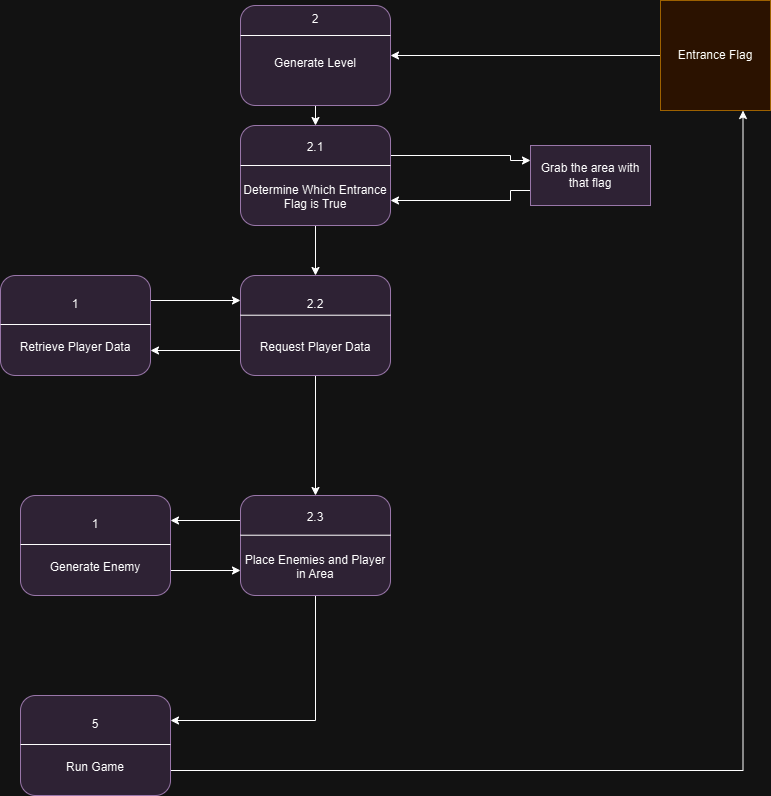
**Exceptions:**

**Step 1.1:** The game requests what entities to spawn (player, items, enemies, bosses) .

**Step 1.2:** The game spawns the requested entities.

**Step 2.1:** The game receives information on what the player has (if the player has a lantern equipped for instance it would light up the room).

**Step 2.2**: The game gives equip the player with those specific items.



**Post conditions:** The game world appears

**Priority:** 1

**ID:** C01

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

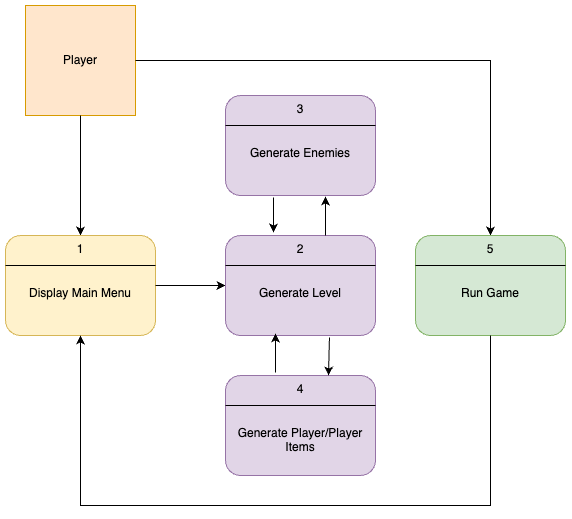
## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

**Level 0 Data Flow Diagram (Context Diagram)**

A close up of a card

Description automatically generated

**Level 1 Data Flow Diagram (Diagram 0)**





**Level 2 Data Flow Diagram (Generate Area)**

A screenshot of a computer

Description automatically generated

### Process Descriptions

Generate Level\*:

IF the game recognizes an entrance flag being true

Determine which flag became true

Based on flag location know where to spawn player in

Request player data

Retrieve the player data based on items and everything the player should have

Place Player in area

ENDIF

IF the area is supposed to have enemies

Request to generate enemies

ENDIF

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

Run feature at different areas of the map in a few different orders:

-Intended order in which you visit the area

-The order reversed

-The order randomized taking the previous flag from the list if the area has already been visited

The output file will have the following characteristics:

-The number of enemies that spawned if there is a discrepancy, we know the feature is not working properly

-The player’s spawning coordinates

-What flag was pulled when you spawned in the area

-The player data when entering that specific area

-The characteristics of the enemies that spawned in that area

-Whether the level successfully loaded or not

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Designing the home and overworld | 4 | - |
| 2. Implementing flags for entering rooms | 3 | 1 |
| 3. Designing the town, shops, area | 6 | 2 |
| 4. Designing the dungeon | 6 | 2 |
| 5. Programming the town side events (church, townsfolk, spawning everything in) | 4 | 3 |
| 6. Programming the dungeon (Enemies, Items, rooms) | 6 | 4 |
| 7. Documentation | 4 | 5,6 |
| 8. Testing | 3 | 5,6 |
| 9. Installation | 1 | 8 |

### Pert diagram

A black and white grid with white numbers

Description automatically generated

### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5,6 |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5,6 |  |  |  |  |
| 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 |