Name: Jacob McKenzie Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature is the level generator. It will take prefab levels then generate a map that connects random levels together along with 1 spawn room and 1 “stair” room to go to the next level.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

A diagram of a diagram

AI-generated content may be incorrect.

### Scenarios

**[You will need a scenario for each use case]**

**Name:** Generate Level

**Summary:** The level will generate every level.

**Preconditions:** Game has been started and PLAY has been clicked.

**Basic sequence:**

**Step 1: Get a set of random rooms.**

**Step 2:** Rooms are placed

**Step 3:** Rooms are joined

**Step 4:** Spawn the player in start room

**Step 5:** Spawn enemies in proper rooms

**Exceptions:**

**Step 2:** Rooms are not overlapping

**Step 3:** Connecting hallways are not overlapping

**Post conditions:** All rooms are generated

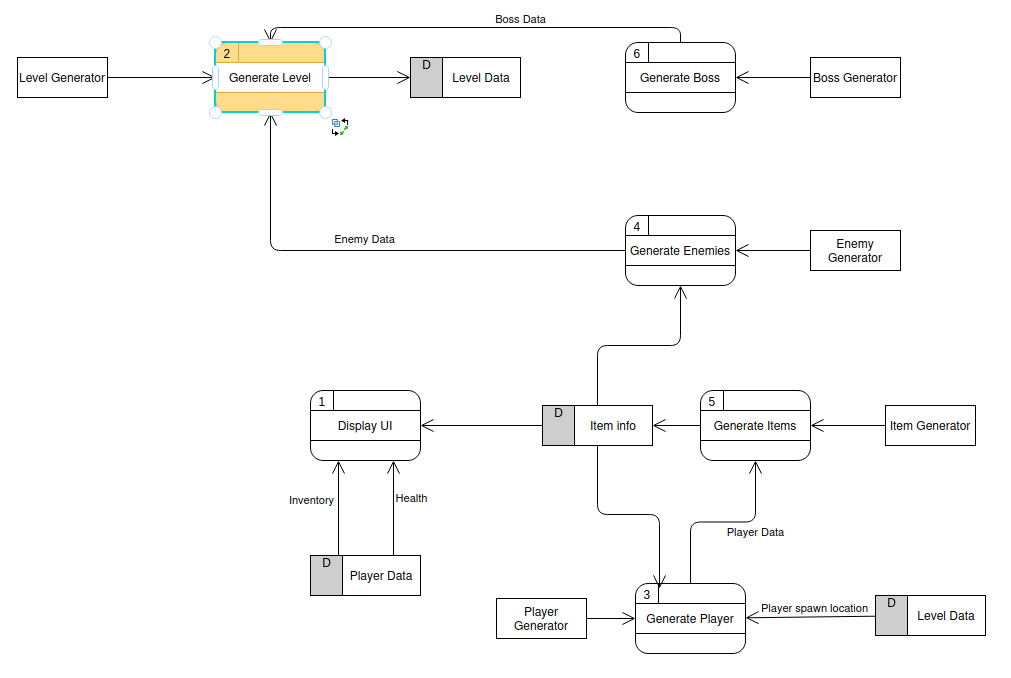
**Priority:** 1

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

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

### Data Flow Diagrams



A diagram of a diagram

AI-generated content may be incorrect.

### Process Descriptions

Connect Rooms\*:

**After building rooms get random locations for the start and end room with the minimum space between their y being the size of all rooms y size added together.**

**WHILE rooms still need to be placed**

**Get Random y location between start and end room**

**Place room**

**Check if overlapping with all other rooms**

**If overlapping get new random location**

**Endif**

**If not overlapping place room**

**Endif**

**Endwhile**

FOR i = room index = 0.

Get closest room to room[0] (start room)

Get rise and run from doorways of closest room

Use horizontal and vertical connection prefabs to connect doorways

Check if connection between start room and end room

IF no connection

Get closest room to end room.

IF connected to start room connect to end room

EndIF

IF not Connected find new room

Connect to end room

ENDIF

ENDIF

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

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Generate 1000 rooms sending output to a file.

The output file will have the following characteristics:

* Number of Rooms
* Path from start room to end room: True or false
* Number of rooms outside of connection
* All Rooms connected: True or false
* Start rooms and end rooms are connected directly: True or false
* Navmesh connected: True or false

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

[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Design All rooms | 24 | - |
| 2. Room Database Construction | 5 | 1 |
| 3. User Documentation | 3 | 2 |
| 4. Programming | 15 | 3 |
| 5. Testing | 5 | 4 |
| 6. Installation | 1 | 5 |

### Pert diagram

A group of black lines with white text

AI-generated content may be incorrect.

### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  | 5 |
|  | 1 | 2 | 3 | 4 | 5 |