Name: Michael Atkinson Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

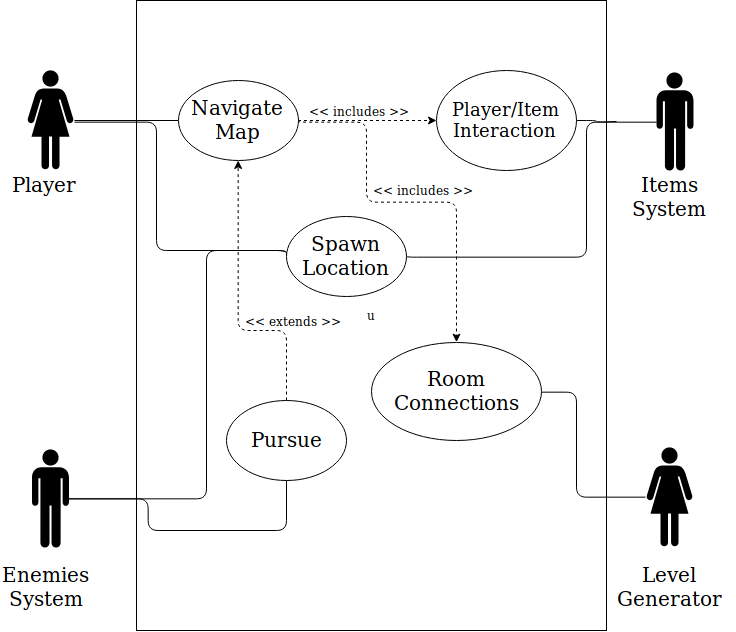
[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

I am building the Level Design/Layout. My system will generate random levels consisting of collectible items, obstacles to avoid and puzzles to get through, and enemies to battle­.

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

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

**Name:** Spawn Location

**Summary:** The level layout will include randomized locations for different actors in the game to begin.

**Actors:** Player, Enemy System, Items System.

**Preconditions:** Level map has been initialized.

**Basic sequence:**

**Step 1:** Choose location for player character.

**Step 2:** Choose location for Enemy characters.

**Step 3:** Choose locations for other items and objects.

**Post conditions:** Level is created.

**Priority:** 1

**Name:** Navigate map

**Summary:** The player is able to move around the map.

**Actors:** Player.

**Preconditions:** Level map has been initialized.

**Basic sequence:**

**Step 1:** The user provides input via keyboard of where the character should move.

**Post conditions:** The character moves.

**Priority:** 1

**Name:** Pursue

**Summary:** Depending on the level and the type of enemy, the enemy character will attempt to pursue the player.

**Actors:** Player, Enemy.

**Preconditions:** Level map has been initialized.

**Basic sequence:**

**Step 1:** The enemy character moves in the direction of the player.

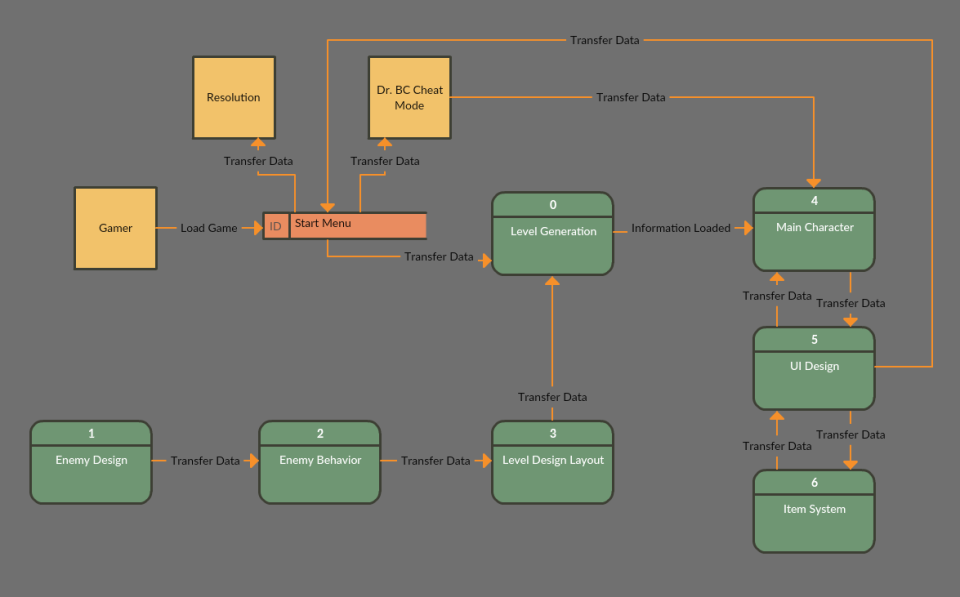
**Post conditions:** Battle ensues.

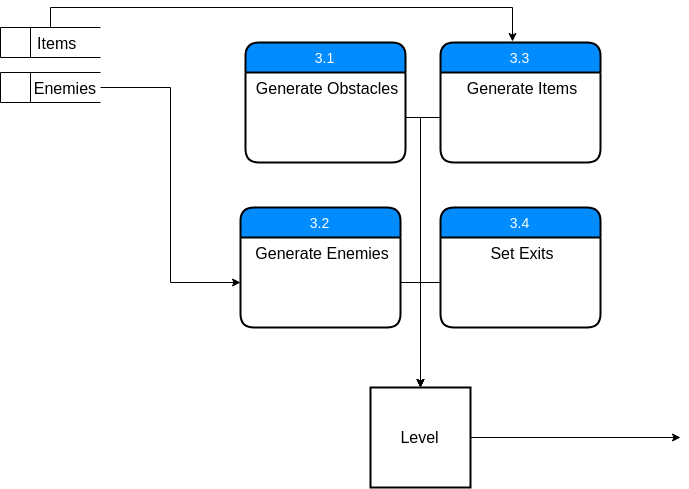
**Priority:** 2

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

Example:

### Data Flow Diagrams





### Process Descriptions

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

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

* Create 1000 rooms, calling for specific number of items, enemies, and exits.  
  Expected behavior:
  + All navigable areas of the room, including exits and items, shall be accessible, and both player and all enemies can navigate to all areas
  + spawn locations for player, enemies, and items shall not overlap.

Accessibility of room can be determined through a flood-fill algorithm.

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

[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Requirements Collection | 6 | - |
| 2. Layout Design | 4 | 1 |
| 3. Generation Algorithm | 4 | 2 |
| 4. Implementation/  Programming | 6 | 2, 3 |
| 5. Documentation | 4 | 3 |
| 6. Testing | 3 | 3,4 |

### Gantt timeline

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

