Name: Michael Atkinson

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

1. Brief introduction __/3

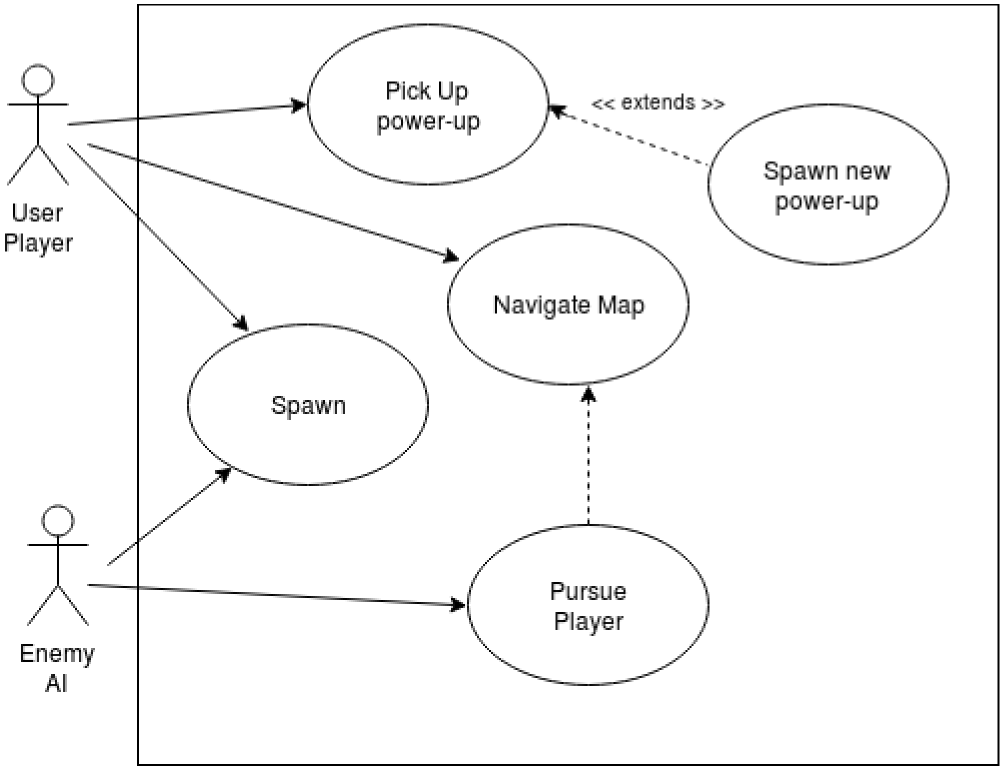
For our game I will be writing code to procedurally generate the map and environment for each level. It will use a randomized seed to generate a 2D grid representing the level, then that grid will run through an algorithm to rearrange the values, generating a desired pattern. After the layout is generated, another random number generator will place obstacles, power-ups, and spawn locations throughout the level.

2. Use case diagram with scenario __14

[Use the lecture notes in class. Ensure you have at least one exception case]

Example:

Use Case Diagrams



Scenarios

[You will need a scenario for each use case]

Name: Pick Up Power-up

Summary: The user will find the randomly-located items throughout the level

Actors: User
Preconditions: power-up and user have been spawned
Basic sequence:
- **Step 1:** generate level, including item locations
- **Step 2:** check for item collision with user

Post conditions: item disappears from level, new item can be spawned at a new location and set time
Priority: 3

Name: Navigate Level
Summary: the map will dictate regions of the level which are traversable by the player and by the enemy, and which regions are not.
Actors: User, Enemy
Preconditions: Level has been generated
Basic sequence:
- **Step 1:** generate level
- **Step 2:** use generated data to create navigational mesh
- **Step 3:** pass navigation data to player character and enemy AI

Priority: 1

Name: Spawn
Summary: The user and enemy will spawn at a random location in the level
Actors: User, Enemy
Preconditions: Level has been generated
Basic sequence:
- **Step 1:** generate level
- **Step 2:** using data generated from the map, determine valid spawn locations within the level
- **Step 3:** generate an instance of player character and enemy AI at randomly chosen valid locations

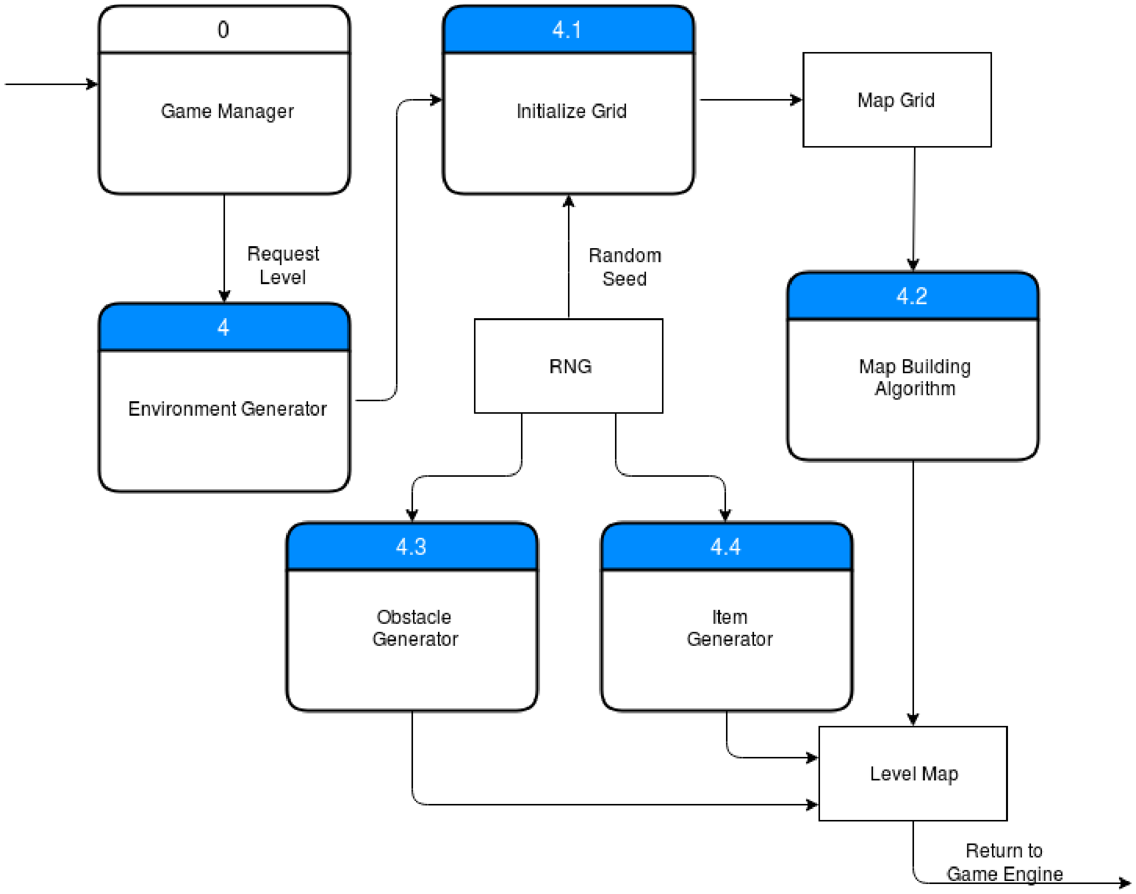
Priority: 1

3. 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



Process Descriptions

Initialize Grid:

```
Declare 2D data structure
    FOR [x,y] IN grid:
        request random integer value in range (0,1)
        assign value to [x,y]
```

Map Building Algorithm:

```
FOR i in range (0,8):
    FOR [x,y] IN grid:
        if [x,y] == 1 AND [x,y] has fewer than 4
        neighbors with value == 1:
            [x,y] == 1
        IF [x,y] == 0 AND [x,y] has greater than 4
        neighbors with value == 1:
            [x,y] == 0
        (Note:  while this algorithm generates quality maps,
         further parameters will likely be instated to increase
         level of detail)
```

Obstacle Generator:

```
FOR [x,y] IN grid:
    if [x,y] != 1:
        create instance of obstacle,
        assigned to cell [x, y]
```

Item Generator:

```
FOR [x,y] IN grid:
    if [x,y] != 1:
        create instance of item,
        assigned to cell [x,y]
```

4. Acceptance Tests ________9

G**enerated map has no disconnected areas.**

This test will involve generating a series of map data sets from the Generator, and using a pathfinding algorithm to confirm that all areas of the map are reachable by the characters.

Generated items and obstacles are only placed in open areas.

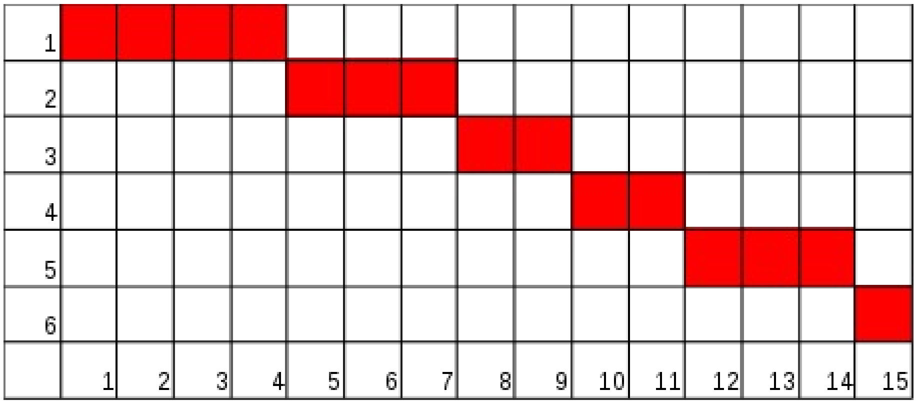
Taking in datasets from the map generation, output random spawn points for generated items. Confirm the correct number of items are generated, and they are all placed in accessible regions.

5. Timeline ________/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	4	-
2. Algorithm Design	3	1
3. Item Generation	2	1,2
4. User Documentation	2	1,2,3
5. Testing	3	1,2,3,4
6. Installation	1	1,2,3,4,5

Gantt timeline



1	🟥	🟥	🟥	🟥											
2					🟥	🟥	🟥								
3								🟥	🟥						
4										🟥	🟥				
5												🟥	🟥	🟥	
6															🟥
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15