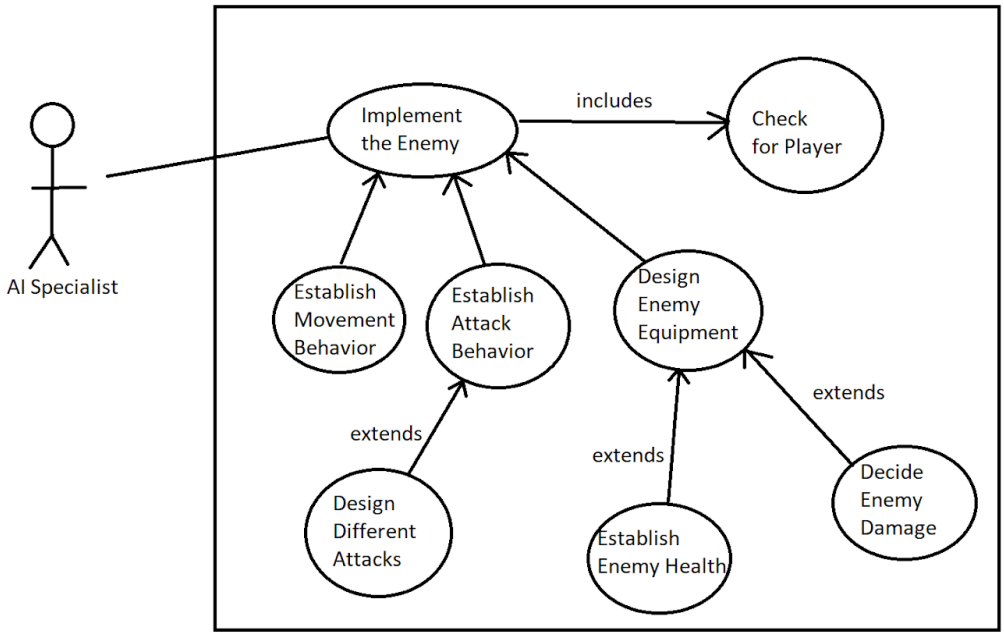
Name: __Conner Mullins_____ Mark _____/50

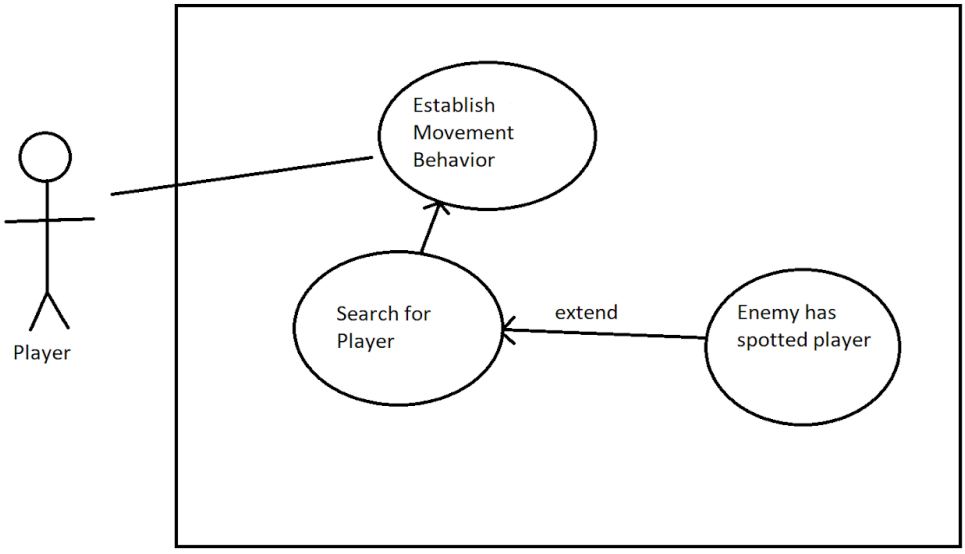
1. Brief introduction __/3

My feature for the Unity game Septicemia is the enemy design and behavior when interacting with the level and player.

My job is to create the enemy movement, attacks, behavior, and equipment that will be used by the enemies. The enemies will be placed throughout the level, passively searching until the player reaches their sight range. Their behavior will change depending on whether or not the player is within this range, and their main objective will be to path-find a route to the player and attack them once they are in attacking range. Different enemies will have different attacks, attacking distances, and movement behaviors.

2. Use case diagram with scenario __/14





Scenario 1 (1st Use Case Diagram)

Name: Implement the Enemy

Summary: The AI Specialist designs enemies that will have unique attacks and health.

Actors: AI Specialist

Preconditions: Game level has been initialized.

Basic sequence:

Step 1: The spawn locations and enemy numbers are specified.

Step 2: Enemy equipment and health is decided.

Step 3: Enemy movement or search patterns are established.

Step 4: Enemy attacks are specified.

Exceptions:

Step 1: The level is a boss area where normal enemies will not spawn.

Step 2: The only enemy on the level is a boss which has its own special equipment.

Post conditions: The enemy will pursue and attack the player.

Priority: 1*

ID: CM1

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

Scenario 2 (2nd Use Case Diagram)

Name: Establish Movement Behavior

Summary: The enemy will change its movement behavior based on its interaction with the player.

Actors: Player

Preconditions: The enemy location, equipment, and attacks have been specified.

Basic sequence:

> **Step 1:** The Enemy will check if the player is within its sight range.
>
> **Step 2:** The Enemy will stop movement when the player is within sight range and within attacking range.

Exceptions:

> **Step 1:** The player has not entered the same location as the enemy.
>
> **Step 2:** The player may kill the enemy before it enters the attacking range of the enemy.

Post conditions: The enemy has approached the player and is within attack range.

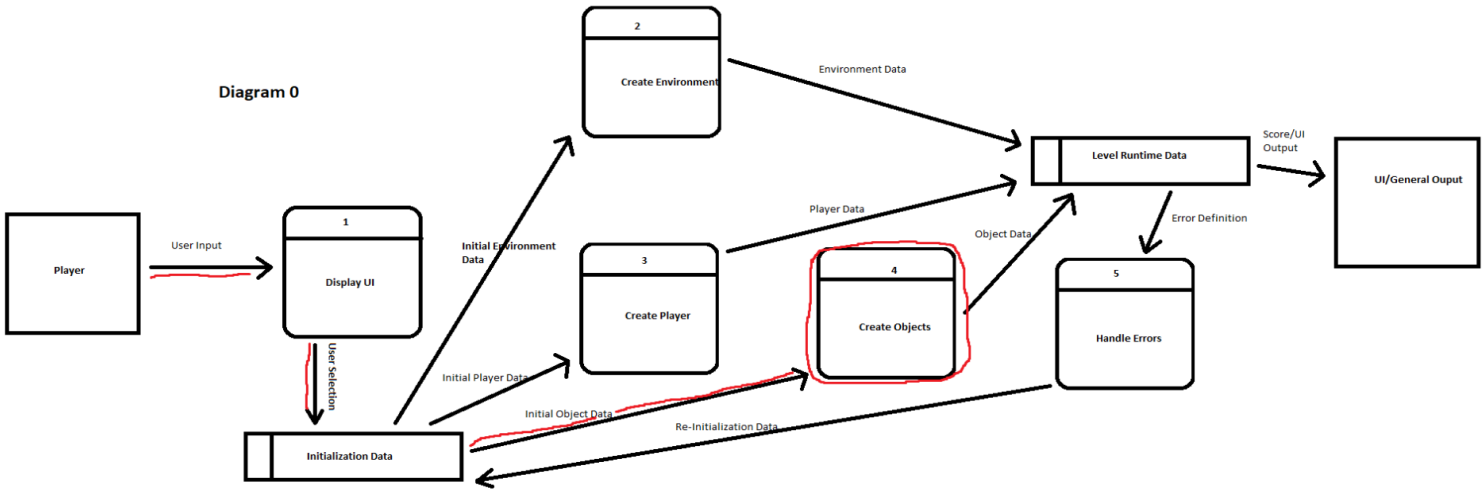
Priority: 2*

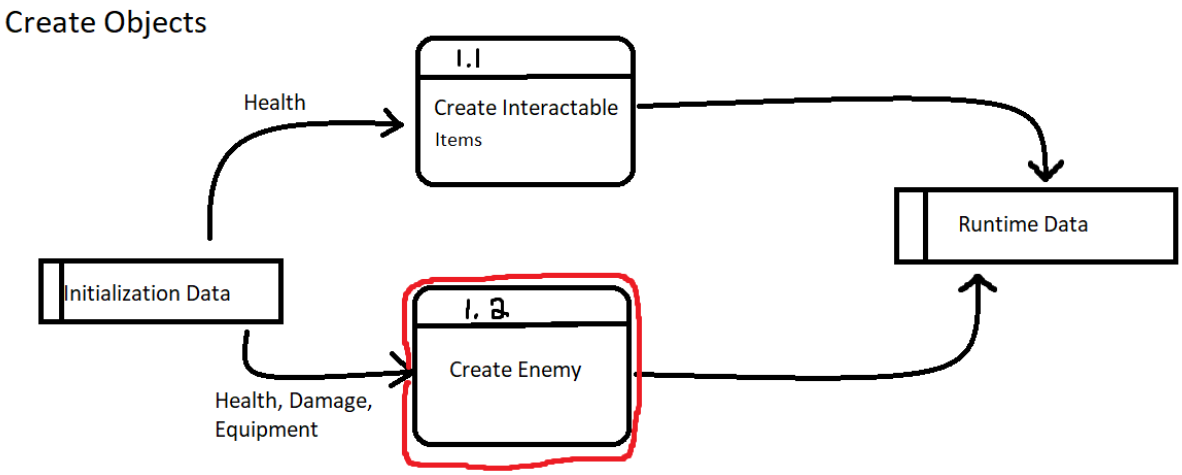
ID: CM2

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

3. Data Flow diagram(s) from Level 0 to process description for your feature ___/14

In the data flow diagrams below, I will cover part of the Create Objects feature. I specifically will be dealing with the “Create Enemy” sub-process.





Process Descriptions

Create Enemy:

IF player is on the first level

```
        Initialize enemies with base health and equipment
END IF
IF player is on the second level
        Increase enemy health, improve equipment, increase enemy
        damage, increase enemy numbers
END IF
WHILE player health is above 0
        WHILE (player is within sight range AND not within attack range)
        AND (enemy health is above 0 AND player health is above 0)
                Move towards player
        END WHILE
        WHILE (player is within sight range AND within attack range) AND
        (enemy health is above 0 AND player health is above 0)
                Attack player
        END WHILE
END WHILE
```

4. Acceptance Tests ___/9

This feature mostly has predetermined elements that will not change with each runtime (Enemy health, weapons, equipment), however the pathfinding of the enemy will be dependent on the location of the player. This acceptance test will primarily test the consistency of the enemy pathfinding as it approaches the player.

This test will run 64 times on each level with each type of enemy (4 types of enemies on 2 different levels for a total of 8 tests). It will place the player randomly within the sight range of the enemy, which will cause the enemy to approach the player. The level, equipment, sight range, and attack range of the enemy will be sent to the output file. The distance from enemy to player once it has begun attacking will also be sent to an output file since the enemy should stop moving once the player is within the attack range. Once the player's health is at or below 0, the test will end.

Once the data has been sent to the output file, it will be used to determine if the radius of the attack range is equal to the distance from the player to the enemy. Since the player will not be moving, the attack range and distance to the player should be the same at the end of the test. The output file will display said distance and the attack radius for the specific enemy. If the two values are the same, then the test will be successful.

Example Output Test Check Analysis for Success/Failure

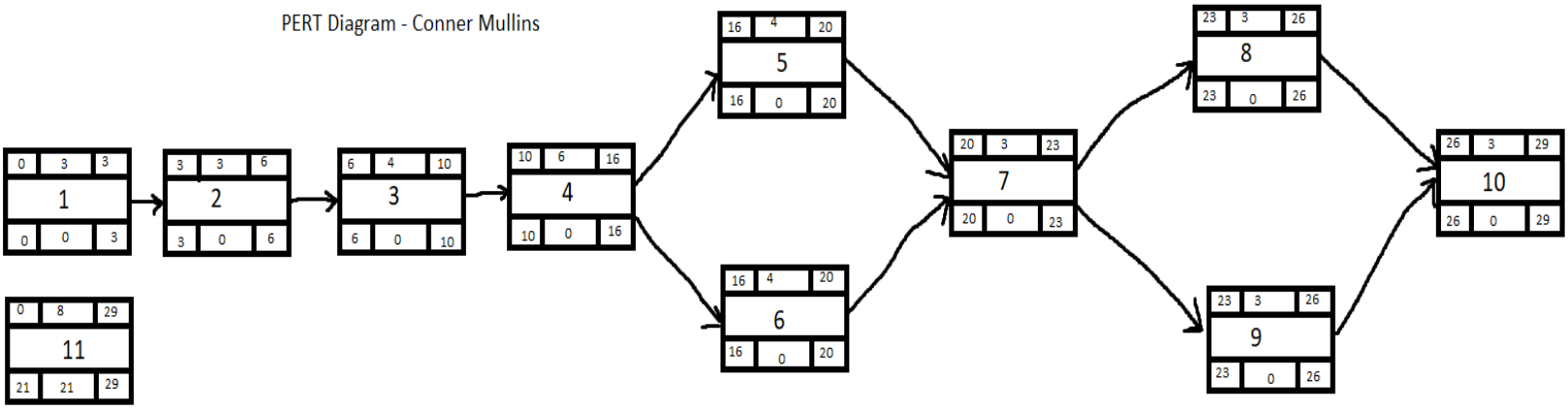
Object	Level	Equipment	Sight Range	Attack Range	Distance	Conclusion
Enemy 1	1	Crowbar	5 meters	1 meters	1 meters	Success
Enemy 1	1	Crowbar	5 meters	1 meters	1.48 meters	Fail
Enemy 1	1	Crowbar	5 meters	1 meters	1 meters	Success
Enemy 1	1	Crowbar	5 meters	1 meters	1.23 meters	Fail
Enemy 1	1	Crowbar	5 meters	1 meters	1 meters	Success
Enemy 1	1	Crowbar	5 meters	1 meters	1 meters	Success
Enemy 1	1	Crowbar	5 meters	1 meters	1 meters	Success
Enemy 1	1	Crowbar	5 meters	1 meters	0.96 meters	Fail

5. Timeline ___/10

Work Items

Task	Duration (Hours)	Predecessor Task(s)
1. Enemy Object Creation	3	-
2. Enemy Health	3	1
3. Enemy Attack	4	1
4. Enemy Equipment	6	3
5. Enemy Sight Range	4	3,4
6. Enemy Movement	4	4
7. Enemy Attacks Range	3	5,6
8. Documentation	3	7
9. Testing	3	7
10. Installation	3	8
11. Enemy Animations/Artwork	8	-

Pert Diagram



Gantt Timeline

Gantt Timeline	Conner Mullins																													
Key:																														
	Work Hours																													
	Slack																													
	Hour (right)																													
	Job (below)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	1																													
	2																													
	3																													
	4																													
	5																													
	6																													
	7																													
	8																													
	9																													
	10																													
	11																													