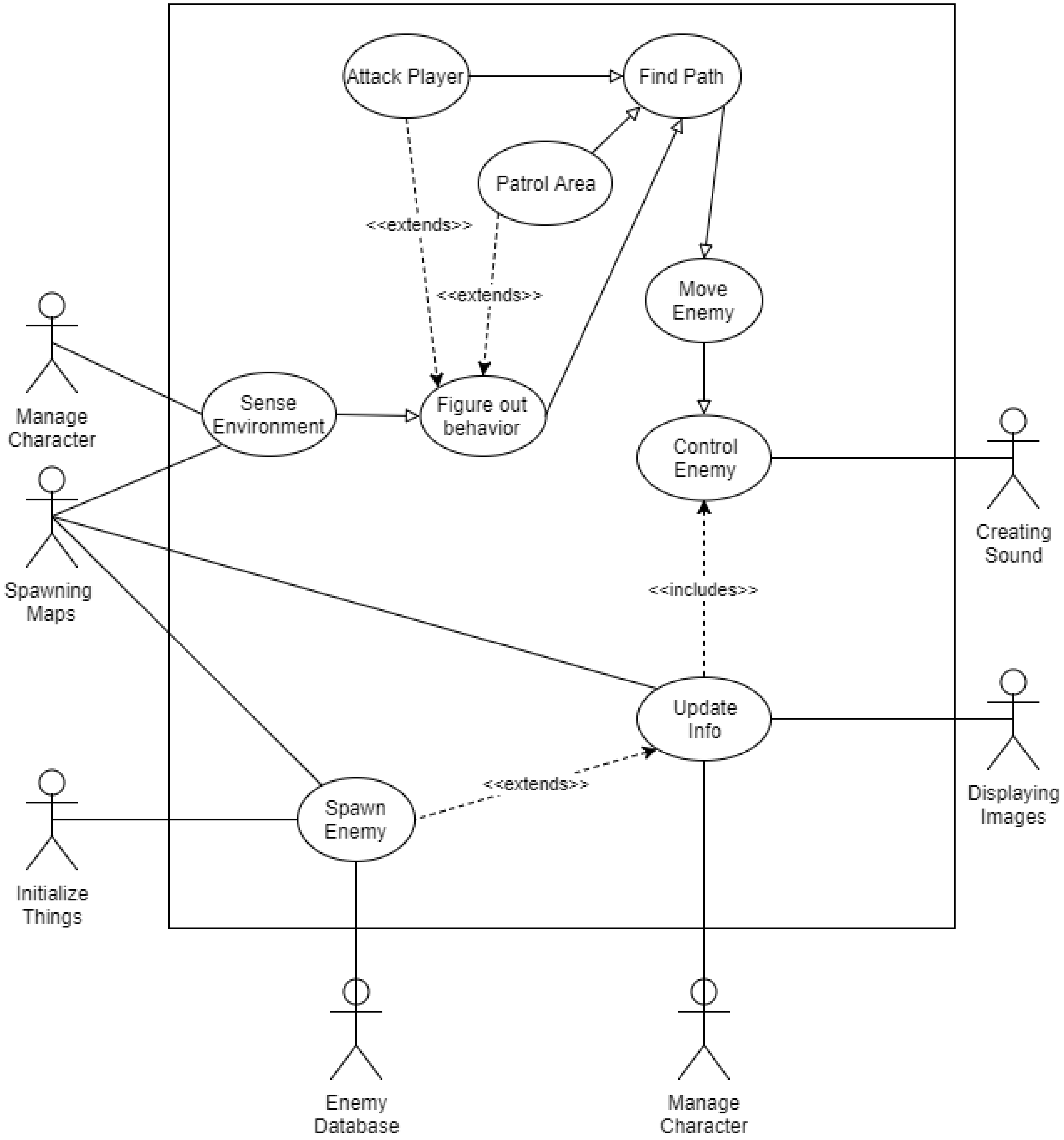
Name_________Nick Avery__________ Mark _______________________/50

1. Brief introduction __/3

My feature will handle the management of enemy agents. This will include of all the algorithms that the agents will use to make decisions and manage the agents' health, damage values and other elements needed for an effective enemy agent.

2. Use case diagram with scenarios __14

Use Case Diagrams



Actors: Spawning Maps, Manage Character

Preconditions: Process has been given data from Spawning Maps and Manage Characters processes.

Basic sequence:

Step 1: Parse data for needed info for other processes

Step 2: Send needed info to processes

Exceptions:

Step 1: There is no info to parse: Send an error to the processes

Step 2: The info is not parsed correctly: Send an error to the processes

Post conditions: Data is sent to the processes

Priority: 1

ID: ME1

Name: Figure out behavior

Summary: Info given is used to figure out what actions an enemy will take.

Actors: none

Preconditions: Process has been given data from Sense Environment process.

Basic sequence:

Step 1: Utilize the data given with an algorithm

Step 2: After algorithm has finished, use the result to see which action is best

Step 3: Send the action decision to Control Enemy process

Exceptions:

Step 1: An error was sent instead of info: Take no Action

Step 2: All actions are equally viable: Take an action that interacts/will interact with the player

Step 3: Attack action was sent: Attack the player

Step 4: No player is present: patrol the area

Post conditions: Action is sent to the Control Enemy Process

Priority: 1

ID: ME2

Name: Attack Player

Summary: A special behavior that is expressed when the enemy is attackable.

Actors: none

Preconditions: Figure out behavior has given the attack action

Basic sequence:

Step 1: If the character is in range, give a damage report

Step 2: If the character is not in range, find a path to it

Step 3: Send info to Find Path

Exceptions:

Step 1: The Character has moved after the attack action was finished: continue attack until another action has been chosen

Step 2: The enemy has died: Do not send a damage report

Post conditions: Info has been sent to Find Path

Priority: 2

ID: ME3

Name: Patrol Area

Summary: Patrol an area until another action is taken

Actors: none

Preconditions: Figure out behavior has given the patrol action.

Basic sequence:

Step 1: Figure out location to the nearest point in the patrol loop

Step 2: send data to Find Path

Exceptions:

Step 1: There is no location/no patrol loop: Stand in place and wait for another action

Step 2: Two or more locations are equidistant: chose one at random

Post conditions: Info is sent to find path

Priority: 3

ID: ME4

Name: Find Path

Summary: Find a path to traverse to a location

Actors: none

Preconditions: A location has been given to this process

Basic sequence:

Step 1: Use location to inform an algorithm of where a goal state is

Step 2: Once path is found, send path to move enemy

Exceptions:

Step 1: There is no location: send a path of length 0

Step 2: A location is unreachable: Send a path of length -1

Post conditions: Info is sent to Move Enemy

Priority: 1

ID: ME5

Name: Move Enemy

Summary: Move an enemy according to a path.

Actors: none

Preconditions: A path has been given by Find Path

Basic sequence:

Step 1: Move enemy until the goal has been met

Step 2: if goal met, stand still

Step 3: if a time limit is reached, do not move further

Step 4: send the new enemy location to the control enemy process

Exceptions:

Step 1: The time limit has been reached: do not move

Step 2: there is a path of length 0: do not move

Step 3: there is a path of length -1: do not move

Post conditions: enemy location is sent to the control enemy process

Priority: 1

ID: ME6

Name: Control Enemy

Summary: Info from multiple processes is gathered and applied to the enemy

Actors: Creating Sound

Preconditions: Info is given to this process

Basic sequence:

Step 1: Find info that has been changed and get a difference

Step 2: send the difference and other reports to Update Info

Exceptions:

Step 1: There is a damage report to the character: send it to Update info

Step 2: There is no difference: send Update info no difference

Step 3: An action was taken that makes sound: send a signal to Creating Sound

Post conditions: Info is sent to Update Info

Priority: 1

ID: ME7

Name: Update Info

Summary: Data is given to update info, and the enemy info is updated accordingly

Actors: Spawning Maps, Displaying Images, Manage Character

Preconditions: Data from multiple sources has been given to Update Info

Basic sequence:

Step 1: Update the necessary info for the enemy

Step 2: Send outgoing info to processes that need it

Exceptions:

Step 1: There is a damage report: Send it to Manage character

Step 2: The enemy has changed location on the map: Send necessary info to Spawning maps and Displaying Images

Step 3: There is no info given: do not update any info

Step 4: there is no info to update: do not update any info

Step 5: there is a new enemy: Initialize the enemy with info

Step 6: there is an error, or no data has been given: Do not update info

Post conditions: Info is sent to the processes that need it.

Priority: 1

ID: ME8

Name: Spawn Enemy

Summary: Get a location and template for a new enemy to be added into the game

Actors: Spawning Maps, Initialize Things, Enemy Database

Preconditions: Spawning Maps and Enemy Database has given info to this process

Basic sequence:

> **Step 1:** Set the location of the new enemy
>
> **Step 2:** Utilize the new enemy template and make an enemy slightly different than it
>
> **Step 3:** Send the new enemy info to Update Info

Exceptions:

> **Step 1:** There is missing data: do not make the enemy and send an error
>
> **Step 2:** An initialize signal has been received: Make enemy

Post conditions: Info is sent to Update info

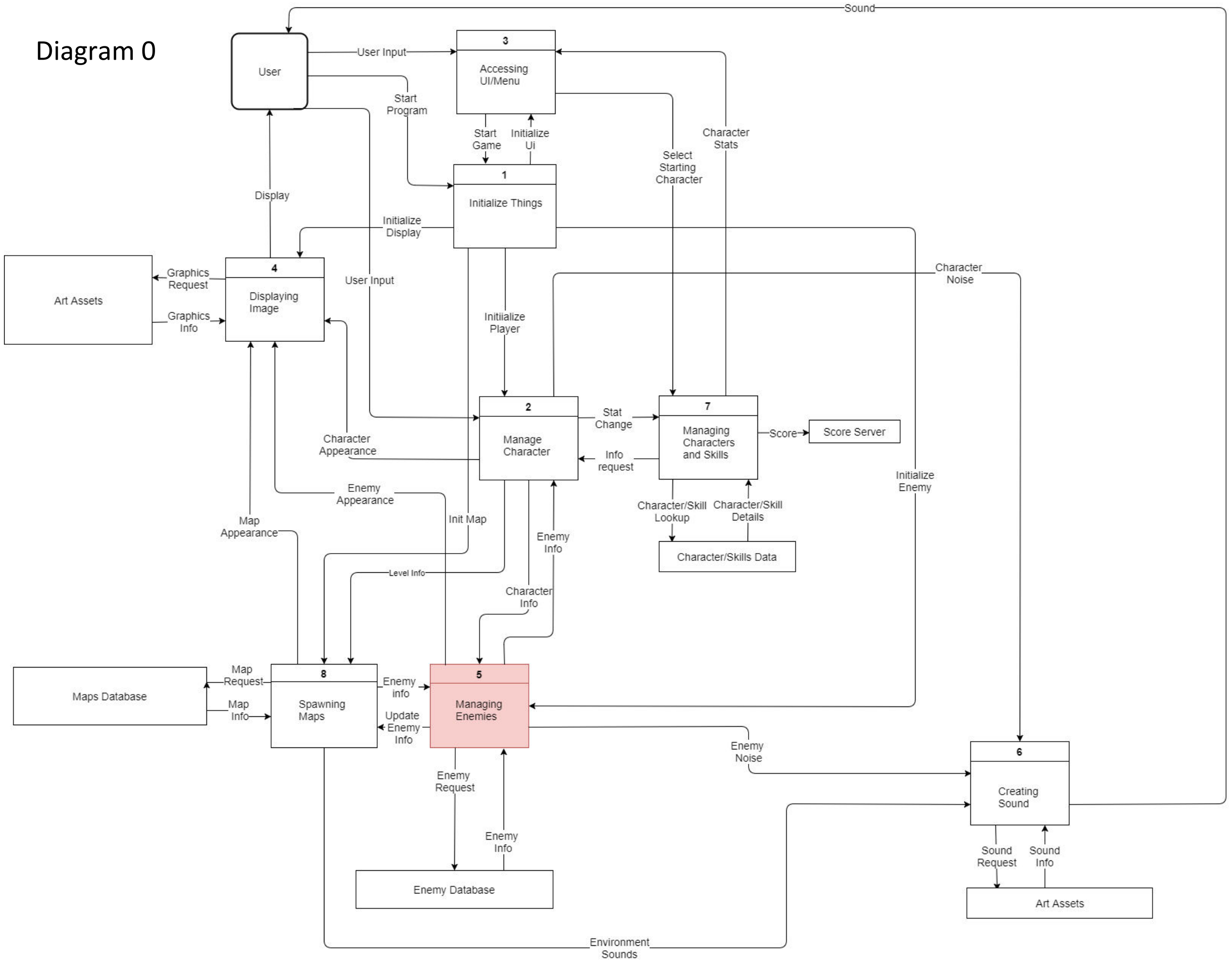
Priority: 1

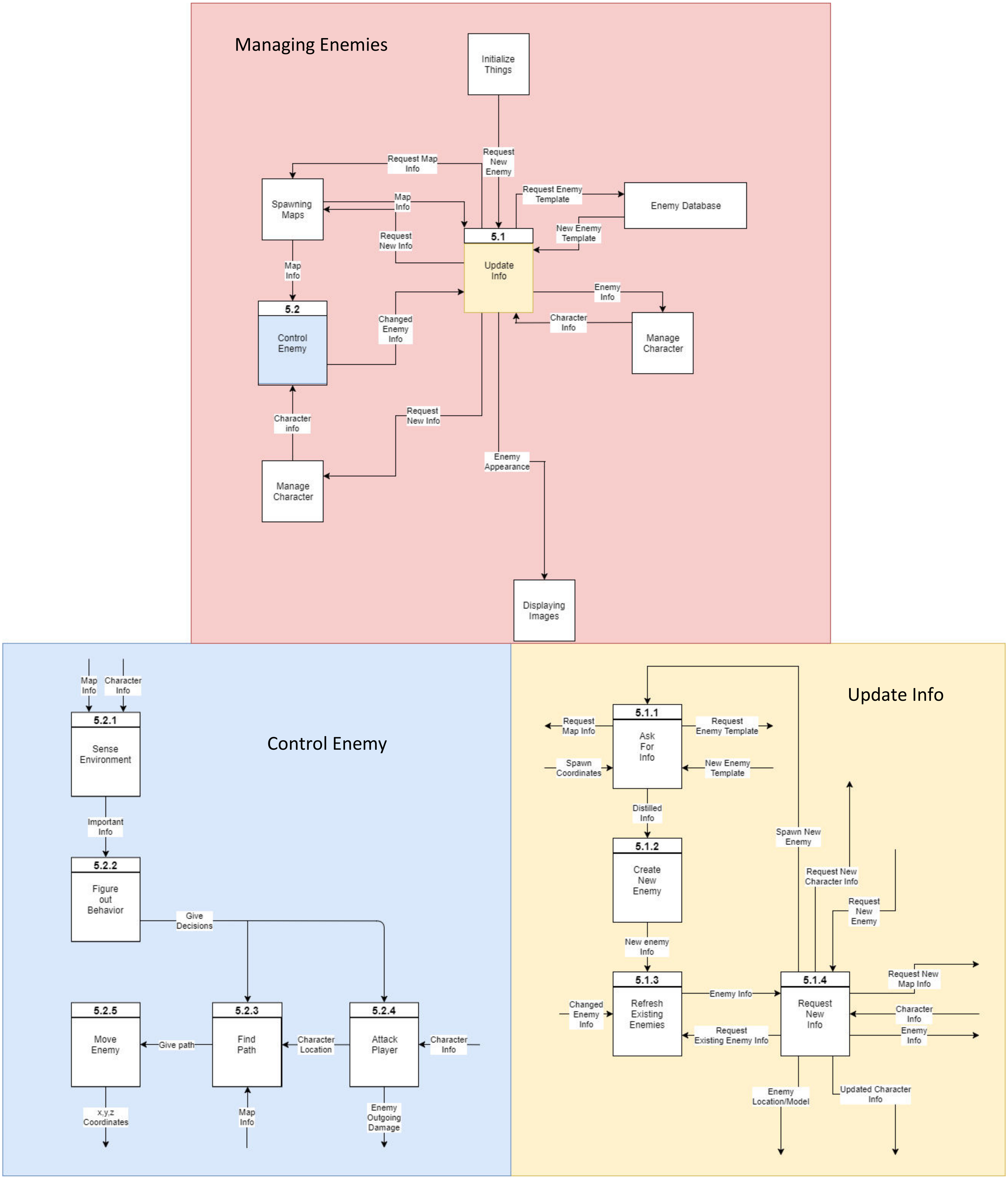
ID: ME9

3. Data Flow diagram(s) from Level 0 to process description for your feature ______14

Data Flow Diagrams

Diagram 0




Managing Enemies
Initialize Things
Request New Enemy
Request Map Info
Spawning Maps
Map Info
Request New Info
Request Enemy Template
Enemy Database
New Enemy Template
5.1
Update Info
Map Info
5.2
Control Enemy
Changed Enemy Info
Enemy Info
Character Info
Manage Character
Character info
Request New Info
Manage Character
Enemy Appearance
Displaying Images
Control Enemy
Map Info
Character Info
5.2.1
Sense Environment
Important Info
5.2.2
Figure out Behavior
Give Decisions
5.2.5
Move Enemy
Give path
5.2.3
Find Path
Character Location
5.2.4
Attack Player
Character Info
x,y,z Coordinates
Map Info
Enemy Outgoing Damage
Update Info
5.1.1
Ask For Info
Request Map Info
Request Enemy Template
Spawn Coordinates
New Enemy Template
Distilled Info
5.1.2
Create New Enemy
Spawn New Enemy
Request New Character Info
Request New Enemy
New enemy Info
5.1.3
Refresh Existing Enemies
Changed Enemy Info
Enemy Info
5.1.4
Request New Info
Request New Map Info
Character Info
Enemy Info
Request Existing Enemy Info
Enemy Location/Model
Updated Character Info

Process Descriptions

5.1.1 – Ask For Info

```
Receive new enemy request
Request map info from Spawning Maps
Receive spawn coordinates for new Enemy and store in struct
Request enemy template from Enemy Database
Receive template for new Enemy and store in struct
Combine info in one struct and give that to Create Enemy
```

5.1.2 – Create New Enemy

```
Receive from Ask For Info
Get Random number
    If Random % 10 < 3
        Reduce enemy size by an amount
    Else if Random % 10 is between 3 and 8
        Keep enemy size
    Else
        enlarge enemy size by an amount
base damage values on enemy size
base health values on enemy size
base speed on enemy size
give changed enemy values to Refresh Existing Enemies
```

5.1.3 – Refresh Existing Enemies

```
If new info is given
    Add new enemy to list of enemies
Take in changed enemy info
For every enemy in the list of enemies
    Apply the changes to the enemy that corresponds to
    the new info
send enemy info to Request New Info
```

5.1.4 – Request New Info

```
If request for new enemy
    Send request for new enemy to Ask for Info
Request Existing enemy info
Receive existing enemy info
Request new Character info for Manage character
    Also send enemy info and damage to character to Manage
    character
Receive character info from Manage character
Request map info from Spawning maps
Send Enemy locations and models to Displaying Images
```

```
5.2.1 – Sense Environment
    Receive Map and character info
    Extract relevant info
        Character position and health
        Map information
    Put relevant info into struct
    Send struct to Figure out Behavior
5.2.2 – Figure out Behavior
    Receive info from Sense Environment
    If character is present
        If enemy is not below 30% health
            Attack player
        If character is low health
            Attack player
        If enemy is below 30% health
            Flee
    If character is not present
        Patrol area
    Send decision to find path and Attack Player
5.2.3 – Find Path
    Get Decision from Figure out Behavior
    If Attack Player
        Get character location from Attack Player
        Find path to player
    If flee
        Find path away from player
    If patrol area
        Find patrol path
    Send path to Move enemy
5.2.4 – Attack Player
    Get character info
    If in range to attack
        Calculate character damage
        Send Damage to Update Info
    If not in range
        Send character location to Find path
5.2.5 – Move enemy
    Receive path from Find path
    While time to move is not exceeded
        Move enemy on path
    Return x, y, z coordinates to Update Info
```

4. Acceptance Tests _______9

- Make 500 enemies to test Spawning Enemies
    - Expected behavior
        - Enemies of Various sizes
        - Enemy characteristics based on those sizes
        - If there is an error
            - No enemy will be made
    - Unexpected behavior
        - Enemies are all the same size
        - Enemy characteristics don't match size
        - No enemies are made
        - Enemies are made from errors/null info
- Give 500 character positions to test Control Enemy
    - Expected behavior
        - A path for each position
        - If the path is large
            - Movement on each path
        - If the path is small or zero
            - Enemy trying to attack character
            - Or staying still
        - If character is not in the map
            - No path is made/path of zero is given
    - Unexpected behaviors
        - A path to a wrong location
        - A path that is blatantly false or not ideal
        - A path to a character outside of the map
        - A path to a character that does not exist
- Give 200 character statuses to test Figure out Behavior
    - Expected behavior
        - Enemy makes a logical decision for the status
        - Enemy follows the decision to its end or finds a better on
    - Unexpected behavior
        - Enemy makes an illogical decision
        - An enemy never makes a decision
        - An enemy will follow a bad decision without finding a better one
- Attack a character 500 times to test attacking
    - Expected behavior
        - Attacks are appropriate for the given circumstance
        - Attacking effects the character
    - Unexpected behavior
        - Attacks never effect the character

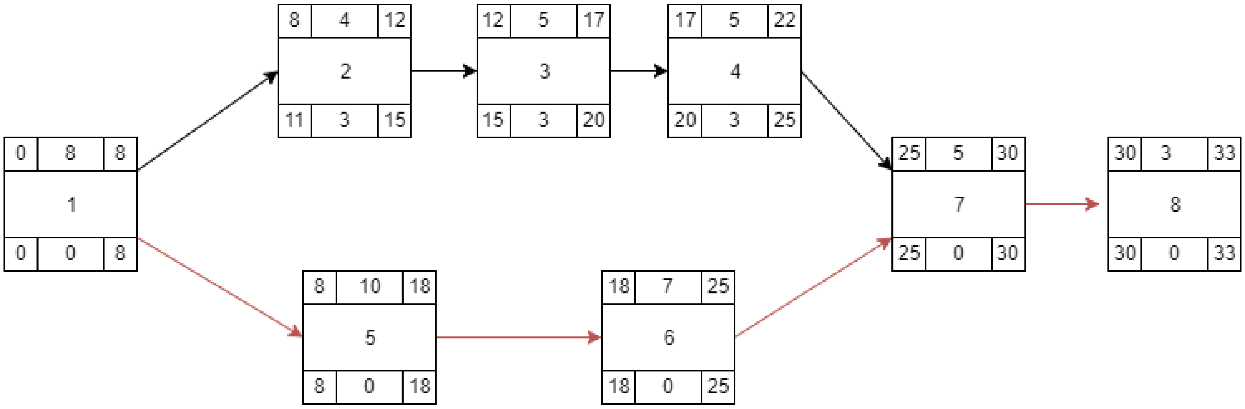
- Attacks will always kill the player
- Attacks never trigger in a circumstance where they should not

5. Timeline _______/10

Work items

Task	Duration (Hours)	Predecessor Task(s)
1. Research	8	-
2. Enemy Database	4	1
3. Enemy Templates	5	2
4. Spawning Enemies	5	3
5. Modeling Decision Making	10	1
6. Movement and Combat	7	5
7. Documentation	5	4,6
8. Testing	3	7

Pert diagram



Gantt timeline

1	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥																									
2									🟥	🟥	🟥	🟥	🟦	🟦	🟦																		
3													🟥	🟥	🟥	🟥	🟥	🟦	🟦	🟦													
4																		🟥	🟥	🟥	🟥	🟥	🟦	🟦	🟦								
5									🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥															
6																			🟥	🟥	🟥	🟥	🟥	🟥	🟥								
7																										🟥	🟥	🟥	🟥	🟥			
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	25	26	27	28	29	30	31	32	33