Name: Robert Goes Mark _____/50

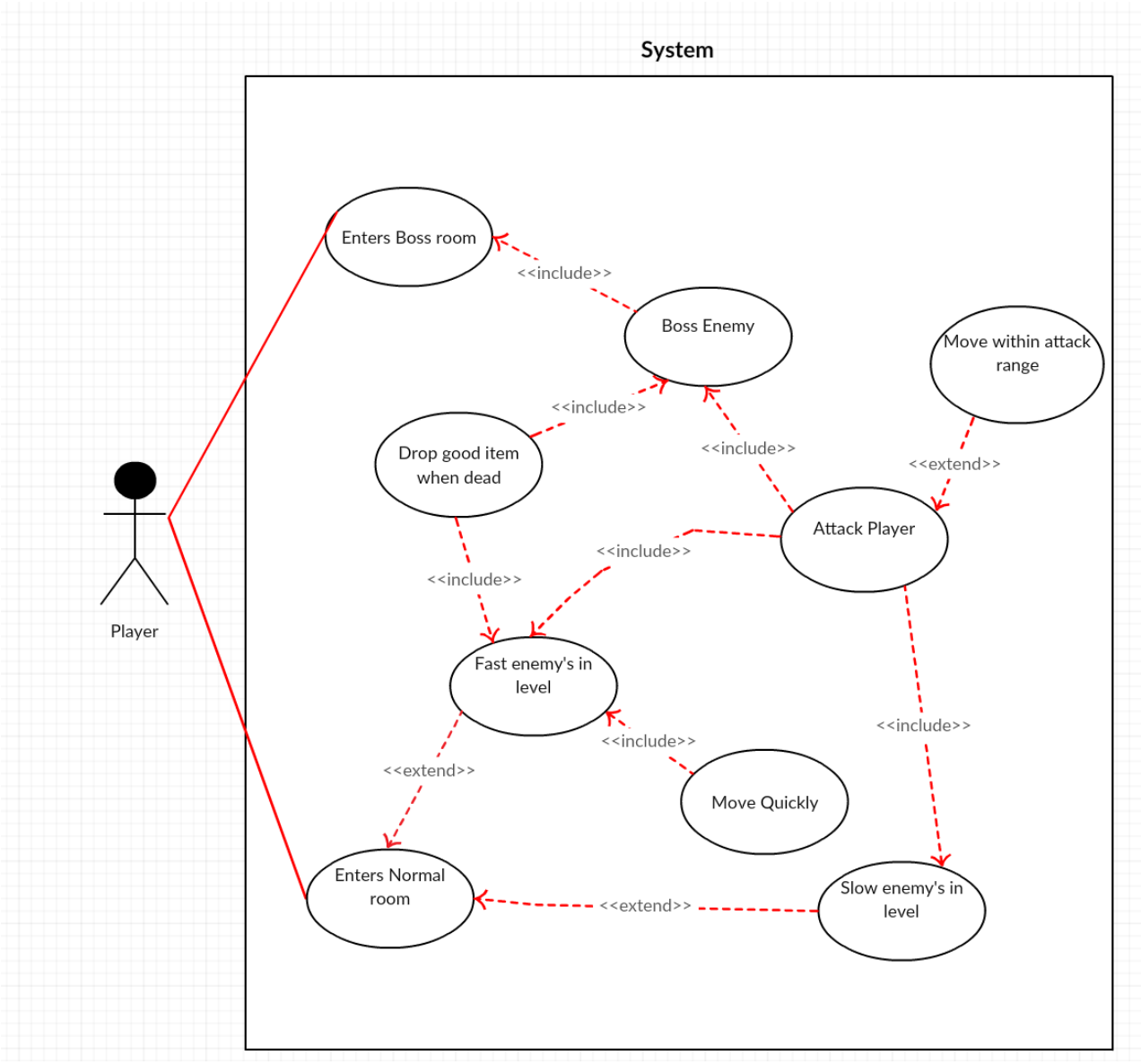
[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. Brief introduction __/3

My feature is the enemy behavior. This includes attacking the player, finding a path to the player, and dropping items on death.

2. Use case diagram with scenario _14

Use Case Diagrams



Scenarios

Name: Enter Normal Room
Summary: The player enters a normal room through a doorway.
Actors: Player.
Preconditions: The room has been initialized.
Basic sequence:
- **Step 1:** Detect if user has entered through the door
- **Step 2:** Check if enemies in room(level)
- **Step 3:** Begin enemy behavior

Exceptions:
- **Step 1:** Player leaves room: stop enemy behavior
- **Step 2:** Player Dies: Show game over screen

Post conditions: Player is in normal room
Priority: 1
ID: EB11

Name: Enter Boss Room
Summary: The player enters a boss room through a doorway.
Actors: Player.
Preconditions: The room has been initialized.
Basic sequence:
- **Step 1:** Detect if user has entered through the door
- **Step 2:** Begin boss enemy behavior

Exceptions:
- **Step 1:** Player leaves room: Reset boss statistics(health, armour, etc)
- **Step 2:** Player dies: Show game over screen

Post conditions: Player is in a boss room.
Priority: 2
ID: EB11

Name: Boss Enemy
Summary: More powerful then normal enemy
Preconditions: Player enters boss room
Basic sequence:
- **Step 1:** Find player location
- **Step 2:** Begin Attack Player

Exceptions:
- **Step 1:** Player leaves room: Reset enemy armour/health
- **Step 2:** Player kills boss: Handle moving to next level

Post conditions: Boss enemy
Priority: 2*
ID: EB21

Name: Fast enemy in level
Summary: Begin behavior for the fast enemy type .
Preconditions: Player is in a normal room, fast enemy in level
Basic sequence:
Step 1: Find player location
Step 2: Begin Move Quickly
Step 3: Begin Attack Player
Exceptions:
Step 1: Player leaves room: stop enemy behavior
Step 2: Player Dies: Show game over screen
Step 3: Enemy Dies: Begin drop good item
Post conditions: Fast enemy in level
Priority: 2*
ID: EB21

Name: Slow enemy in level
Summary: Begin behavior for the slow enemy type.
Preconditions: Player is in a normal room, fast enemy in level
Basic sequence:
Step 1: Find player location
Step 2: Begin Attack Player
Exceptions:
Step 1: Player leaves room: stop enemy behavior
Step 2: Player Dies: Show game over screen
Step 3: Enemy Dies: Drop nothing
Post conditions: Slow enemy in level
Priority: 1*
ID: EB21

Name: Attack player
Summary: Begin attacking player
Preconditions: fast enemy in level or slow enemy in level or boss enemy
Basic sequence:
Step 1: Find player location
Step 2: Begin damaging player with weapon
Exceptions:
Step 1: Weapon not in range: Move within attack range
Post conditions: Attacking player
Priority: 1*
ID: EB31

Name: Move within attack range
Summary: Find a path to become within attack range of player
Preconditions: Not within range for attack player
Basic sequence:
Step 1: Find player location
Step 2: Find a path to the player
Step 3: Move to player
Exceptions:
Step 1: No path found: Raise exception
Post conditions: Moving within attack range
Priority: 1*
ID: EB21

Name: Move Quickly
Summary: Find a path to as close as possible to player
Preconditions: Fast enemy in level
Basic sequence:
Step 1: Find player location
Step 2: Find a path to the player
Step 3: Move quickly to player
Exceptions:
Step 1: No path found: Raise exception
Post conditions: Moving quickly
Priority: 3*
ID: EB21

Name: Drop good item when dead
Summary: When an enemy of the included type (Boss/fast) dies, drop a good item
Preconditions: Fast/boss enemy dies
Basic sequence:
Step 1: Get item from the Item System
Step 2: Drop it on the ground in the location the enemy died
Exceptions:
Step 1: Player leaves room: remove item from ground
Post conditions: Good item dropped
Priority: 2*
ID: EB31

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

Diagram level zero:

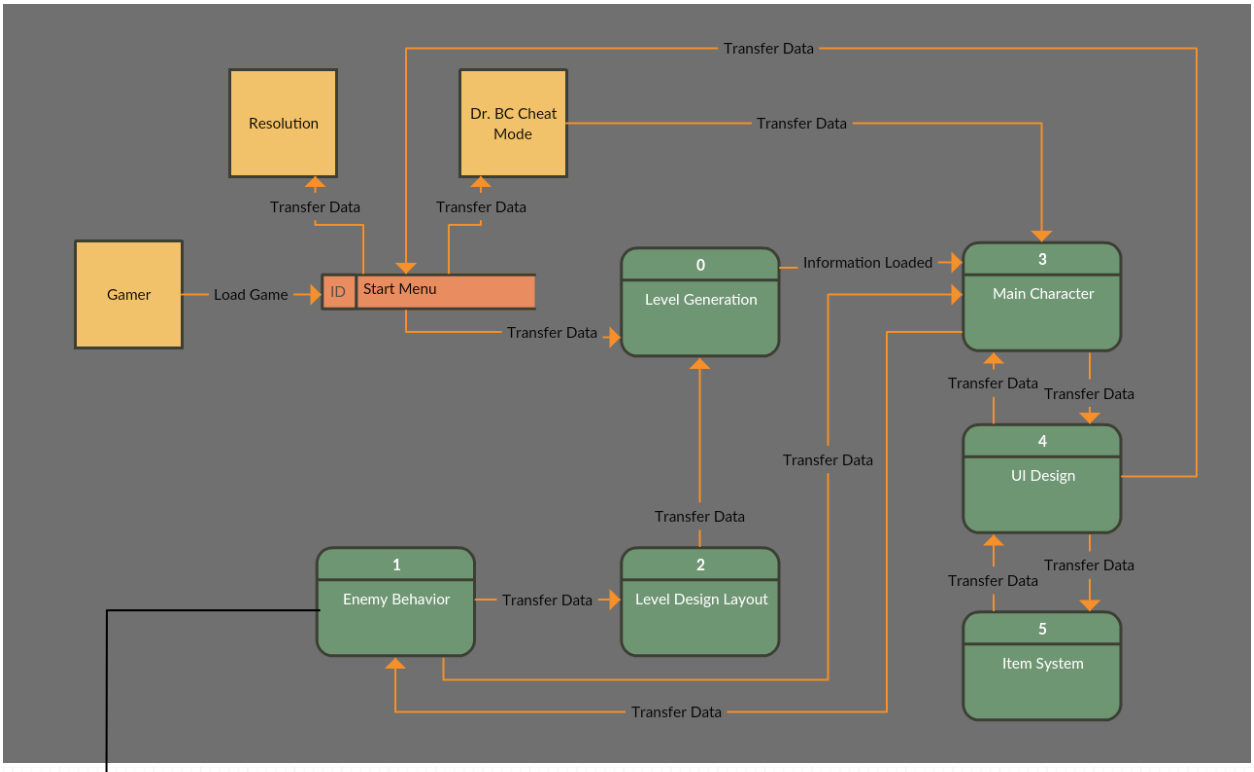
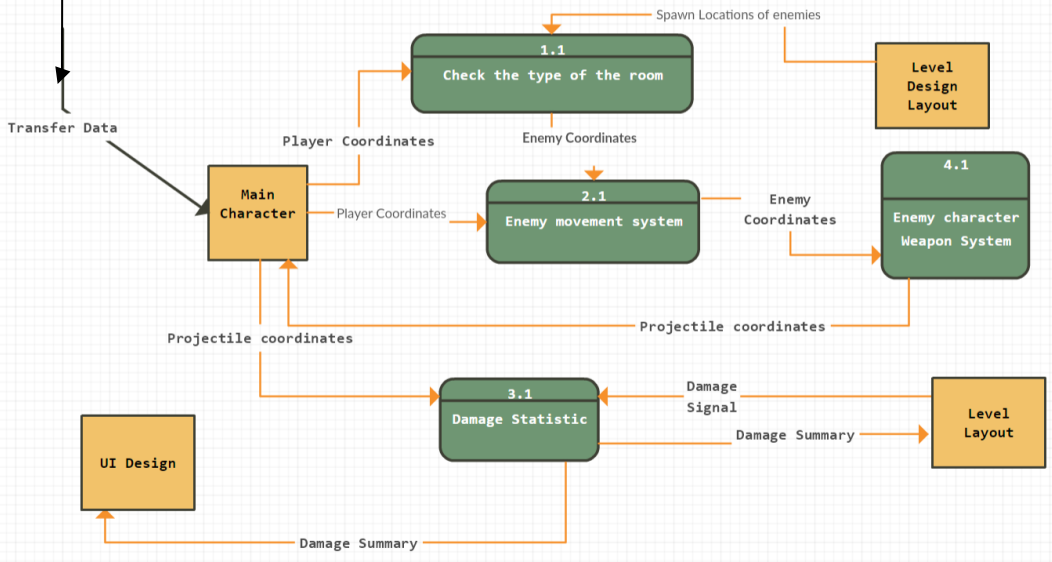


Diagram 1: Enemy Behavior



Process Descriptions

Check type of room

Take the users coordinates, and check if they have entered a room with enemy spawns by getting the spawn locations from level design layour

Logic:

```
If (Player coordinates) is in room with (enemy spawn locations)
    Spawn entities associated with the enemies and pass their coordinates to the enemy movement system
Else
    do nothing
```

Enemy movement system

Takes the players position and the enemies' position, and find's a path to move the enemy toward the player

Logic:

```
If enemy outside of weapon range
Then find path to player
Then move along the path
```

Enemy Character weapon system

Takes the enemies coordinates and the players coordinates and determines if it is range to shoot. If it is, it computes a projectile coordinate and sends that to the main character.

Logic:

```
If enemy and player are in range to shoot
Fire projectile toward the main character
and pass that data to the main character
```

Damage Statistics:

Takes "damage signal" as an input from the level system, when an enemy has been hit by projectile coordinates from the main character. It will then pass the data to the ui to update the displayed health for an enemy and to the level layout, to update the health for an enemy

Logic:

```
Damage signal is received form Level layout and
the projectile coordinates from the player
match that of an enemy.
Then send a damage summery to the level system
to update the enemy health and send a damage
summer to the ui manager to update the
displayed health
```

4. Acceptance Tests _____9

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

Damage tests:

- Try sending a damage signal from the level layout and projectile coordinates that do not match an enemy, and verify that damage summery is not sent
- Try sending a damage signal from the level layout and projectile coordinates that do match an enemy, and verify that damage summery is sent

Movement tests:

- Try placing player coordinates in every valid location in the level, and make sure a path can be found.
- Try placing player coordinates outside the level, and make sure a path is not found

Enemy Character weapon system:

- Spawn test players at various locations, and make sure that it can still damage them

Room type:

- For every room in a level, verify when the player coordinates enter it, “The check type of room” receives the proper number of enemy’s in that room.

5. Timeline _____/10

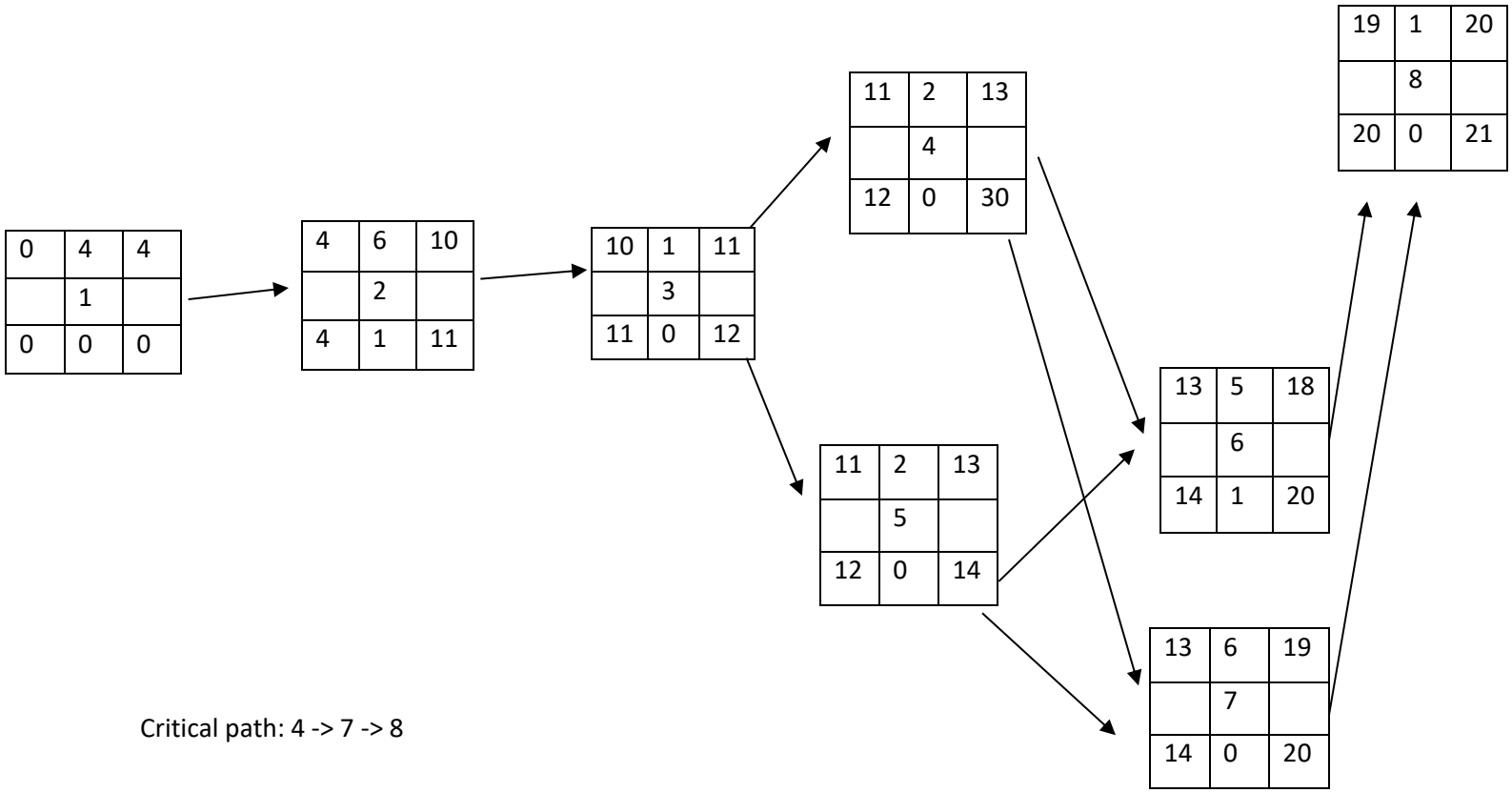
[Figure out the tasks required to complete your feature]

Example:

Work items

Task	Duration hours	Predecessor Task(s)
1. Requirements Collection	4	-
2. Enemy types design	6-7	1
3. Report Design	1	2
4. Enemy pathfinding	2	3
5. Enemy behavior	2	3
6. Programming	5-6	4,5
7. Testing	6	4,5
8. Installation	1	6, 7

Pert diagram



Critical path: 4 -> 7 -> 8

Gantt timeline