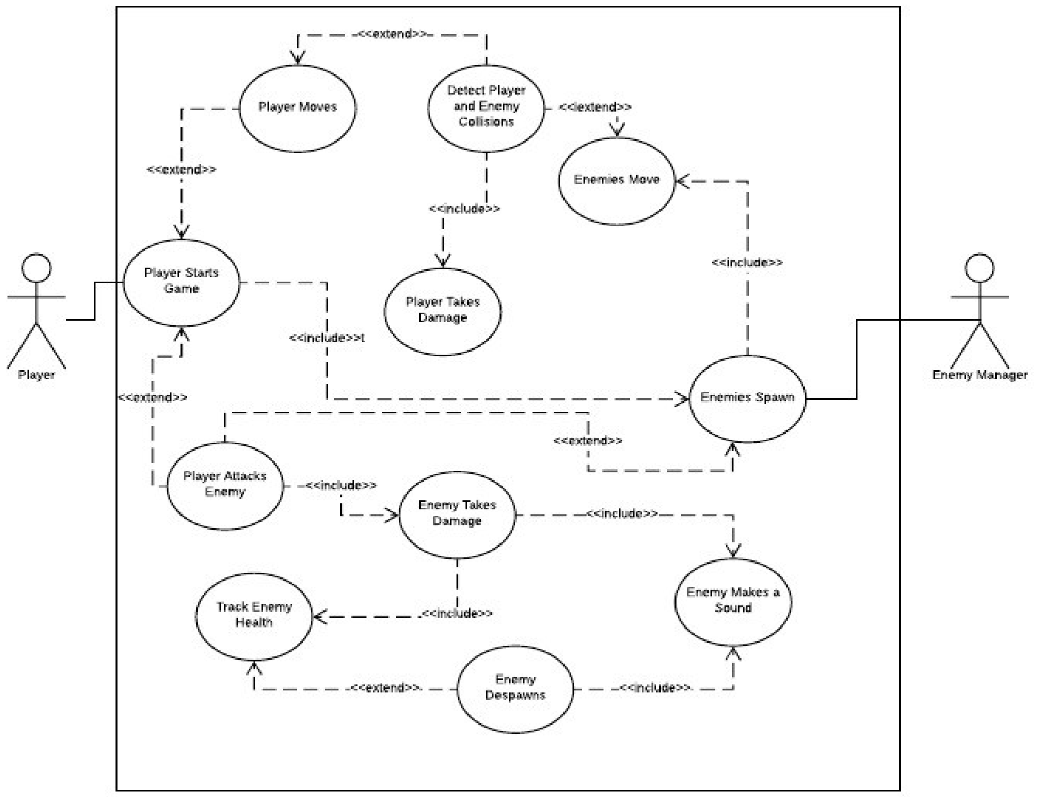
Corbin Schueller   Mark ____________________/50

1. Brief introduction __/3

The feature that I will be developing is the manager for the enemies in our game. This will include their spawning, death, how they move, how they will interact with the player, how they will interact with items on the map, and when they trigger sounds in game. The created enemies in will be 3 dragons and a bat.

2. Use case diagram with scenario   __14

Use Case Diagrams



Scenarios

Name: Enemies Spawn
Summary: Enemies Spawn in the game
Actors: Player, Enemy Manager
Preconditions: Player has game installed.
Basic sequence:
 Step 1: Player presses start in menu
Exceptions:
 None
Post conditions: Enemies spawn in the game
Priority: 1
ID: C07.1

Name: Enemies Move
Summary: Enemies start moving
Actors: Player, Enemy Manager
Preconditions: Player has game installed.
Basic sequence:
 Step 1: Player presses start in menu
 Step 2: Enemies spawn in the game
Exceptions:
 None
Post conditions: Enemies start moving around in game
Priority: 2
ID: C07.2

Name: Player Takes Damage
Summary: The enemy does damage to player
Actors: Player. Enemy Manager
Preconditions: Player has game installed.
Basic sequence:
 Step 1: Player presses start in menu
 Step 2: Enemies spawn in the game
 Step 3: Enemies start moving
 Step 4: Player starts moving around by pressing inputs
 Step 5: A collision between an enemy and the player is detected
Exceptions:
 Step 3: The enemies could move in such a way such that a collision with the player is never made.
 Step 4: The player could move in such a way such that a collision with an enemy is never made.
Post conditions: The enemy that made a collision with the player does damage equal to it’s damage stat

Priority: 2
ID: C07.3

Name: Enemy Takes Damage
Summary: The player does damage to an enemy
Actors: Player. Enemy Manager
Preconditions: Player has game installed.
Basic sequence:

- **Step 1:** Player presses start in menu
- **Step 2:** Enemies spawn in the game
- **Step 3:** The player attacks an enemy

Exceptions:

- **Step 3:** The player could never choose to attack an enemy while playing the game.

Post conditions: The enemy that was attacked by the player would take damage equal to the damage stat on the sword
Priority: 1
ID: C07.4

Name: Enemy Despawns
Summary: An enemy despawns after its health drops to zero
Actors: Player. Enemy Manager
Preconditions: Player has game installed.
Basic sequence:

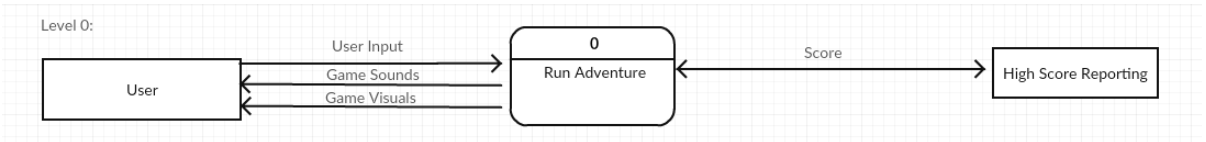
- **Step 1:** Player presses start in menu
- **Step 2:** Enemies spawn in the game
- **Step 3:** The player attacks an enemy
- **Step 4:** That enemy takes damage equal to the damage stat on the sword
- **Step 5:** The health on that enemy is tracked

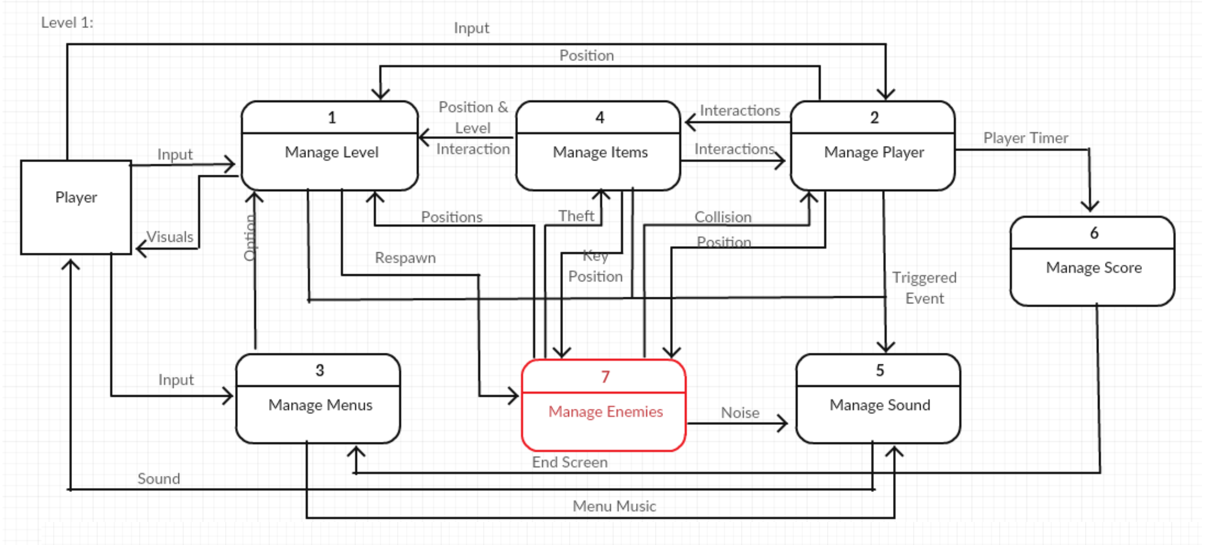
Exceptions:

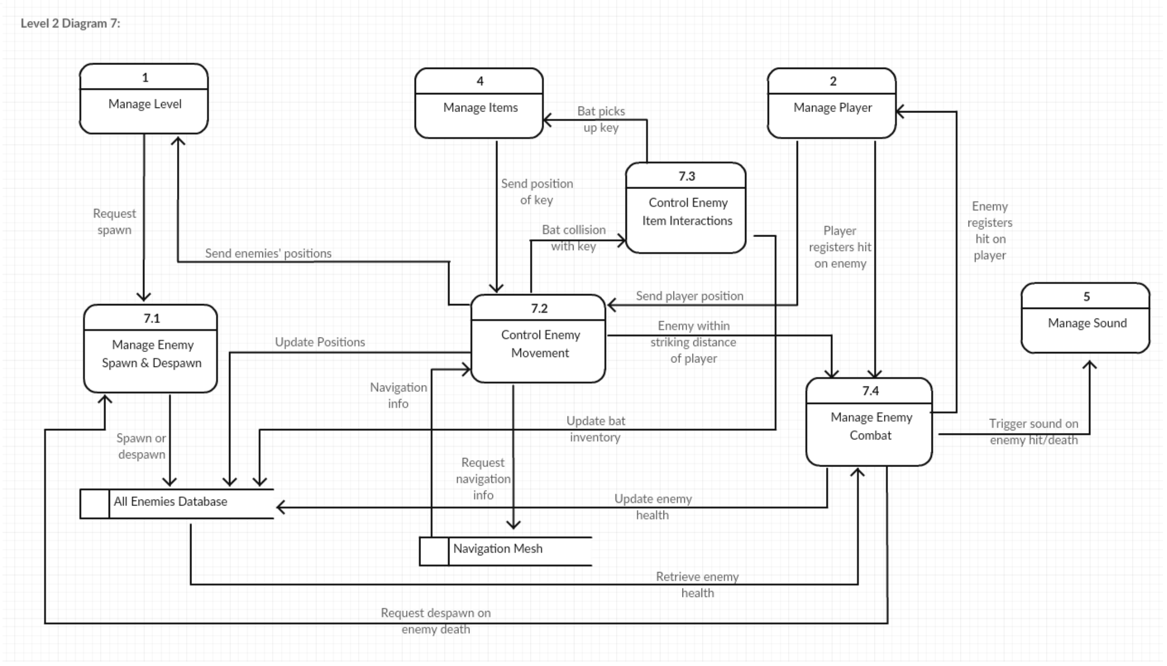
- **Step 3:** The player could never choose to attack an enemy while playing the game.
- **Step 5:** After being attacked by the player, the enemy could still have health above zero

Post conditions: The enemy that has health below zero is despawned in the game
Priority: 1
ID: C07.5

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







Process Descriptions:

Manage Enemy Spawn & Respawn:

```
IF Manage Level requests spawn all THEN:
        spawn all enemies in database
        initialize health to 10 and damage to 5
ENDIF
IF Manage Level requests despawn all THEN:
        despawn all enemies in database
ENDIF
IF  Enemy Combat Manager requests despawn THEN:
        despawn specific enemy enemies in database
ENDIF
```

Control Enemy Movement:

```
IF all enemies are initially spawned or all are respawned
    Request navigation mesh
END IF
FOR each spawned enemy that isn't a bat
    Use player position and navigation mesh to move towards
    player
    IF any non bat enemy is in striking distance of player
        Send enemy info to Enemy Combat Manager
    END IF
END FOR
IF Bat does not have the key
    IF Bat collides with key
        Send collision signal to Control Enemy Item
        Interactions
    ELSE IF key is still on the map
```

```
            Use key position and navigation mesh to move bat
            towards key
        ELSE
            Use player position and navigation mesh to move
            bat away from player
        END IF
        END IF
    ELSE
        Use player position and navigation mesh to move way
        from player
    END IF
    Send positions of enemies to Manage Level
    Send positions of enemies to All Enemies Database
```

Control Enemy Item Interactions:

```
    IF receives message from Control Enemy Movement that the bat
    has collided with the key
        Signal to Item Manager that the bat has picked up the
        key
        Update the bat's inventory in the All Enemies Database
    END IF
```

Manage Enemy Combat:

```
IF receives signal that the player registers hit on an enemy from
Manage Player
    Retrieve that enemy's health from the All Enemies Database
    Reduce that enemy's health by 5
    IF Enemy's Health is less than or equal to zero
        Send request to despawn that enemy to Manage Enemy
        Spawn and Respawn
        Send enemy death sound trigger to Manage Sound
    ELSE
        Send enemy hit sound trigger to Manage Sound
```

```
            Update enemy health in All Enemies Database
        END IF
    END IF
    IF receives signal that enemy in within striking distance of player from
    Control Enemy Movement
        Send signal to Manage Player that an enemy has registered a hit
        for 5 damage
    END IF
```

4. Acceptance Tests _______9

Note: Each test will be run at least 5 times or until expected outputs occur

Note: Each bullet point will be a separate sub-test

Test for intended enemy movement

Create a Unity scene and using a created navigation mesh for our level:

- Instantiate 3 Dragons randomly on the map and have them move towards a test player
- Instantiate 3 Bats and have them move to the location of the key

Input: Enemies (Dragons and Bats)

Output: Observed enemy movement

Test for enemy and player/item collision:

Create a Unity scene and using a created navigation mesh for our level:

- Instantiate a Dragon on the map and have them move towards a test player and see if a collision is detected when their bodies collide.
- Instantiate a bat and have it move to the location of the key and see if a collision is detected when their bodies collide.

Input: Enemies (Dragons and Bats)

Output: Observed collisions or lack of collisions.

Test for enemy and player combat:

Create a Unity scene and using a created navigation mesh for our level:

- Instantiate a Dragon on the map and have them move towards a test player and see if when a collision is detected, the Dragon is able to deal damage to a test player's health equal to the Dragon damage stat.
- Instantiate a Dragon next to a real player and test if the player is able to register an attack with the sword on the Dragon and have the Dragon's health be deducted by the damage stat of the sword.

Input: Dragon and a player

Output: Observed hits to a dragon and hits to a player.

Test for enemy despawning upon death:

Create a Unity scene and place a player with a sword next to a dragon and test to see if after a dragon has been hit enough times to reduce it's health to zero, it will despawn.

Input: Dragon and a player

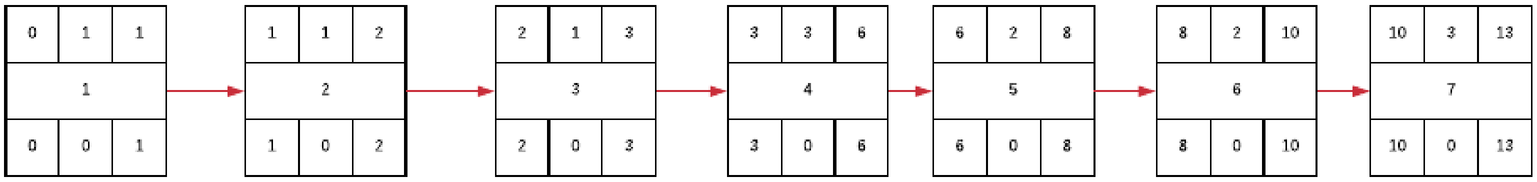
Output: Observed despawning of a dragon.

5. Timeline _________/10

Work items

Task	Duration (Wks)	Predecessor Task(s)
1. Requirements Collection	1	-
2. Prototype Design	1	1
3. Enemy Spawning	1	2
4. Enemy Movement	3	3
5. Enemy Item Interactions	2	4
6. Enemy Combat System	2	4
7. Testing	3	5,6

Pert diagram



Gantt timeline

Task	1	2	3	4	5	6	7	8	9	10	11	12	13
1	🟦												
2		🟦											
3			🟦										
4				🟦	🟦	🟦							
5							🟦	🟦					
6									🟦	🟦			
7											🟦	🟦	🟦