Name: Arjun Rai Mark _____/50

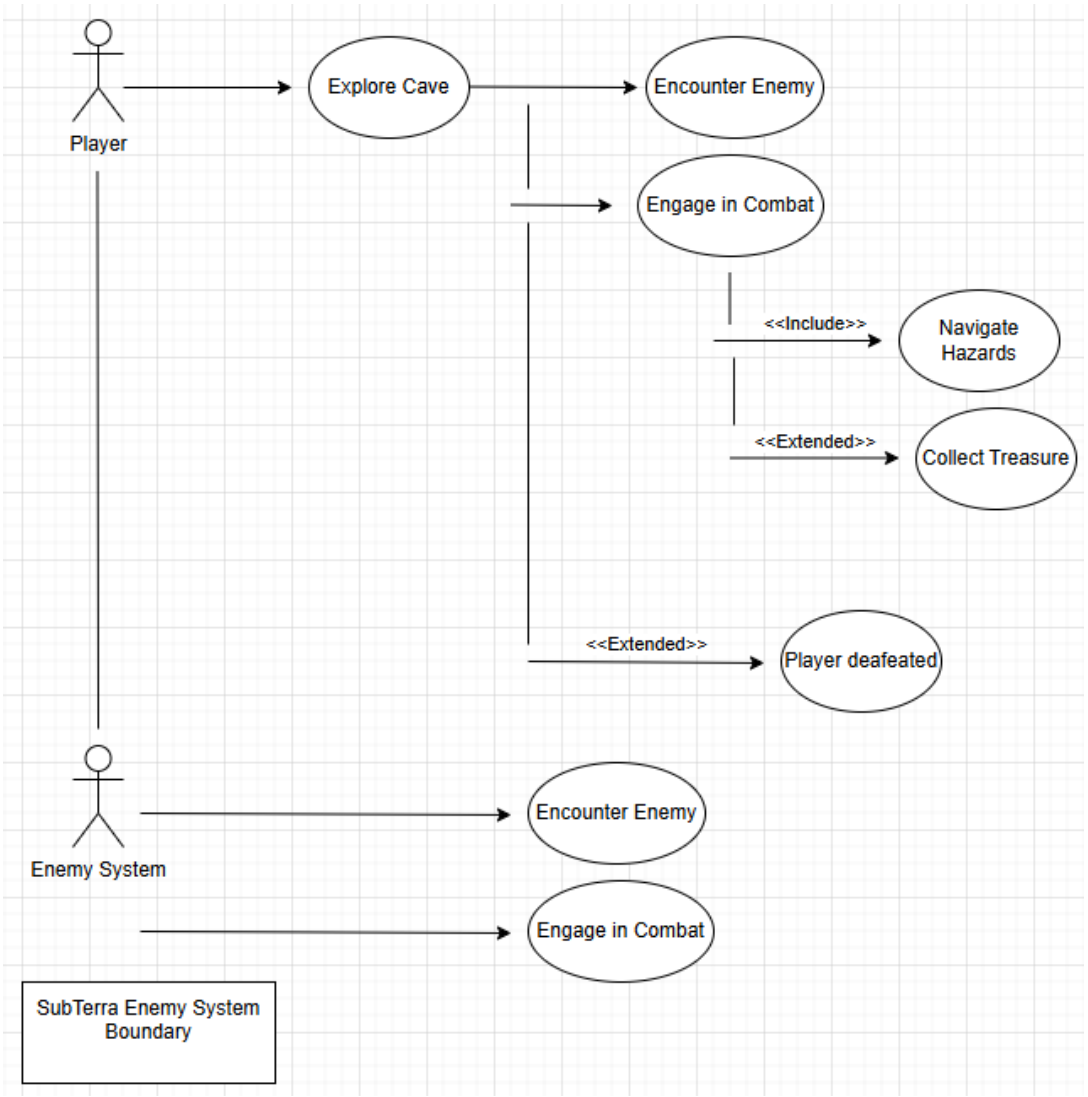
[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. Brief introduction __/3

For the *SubTerra* project, I'm designing a unique enemy. They might take a swing at the player with some damage or throw up obstacles, like blocking the path or setting off little traps, to keep things interesting. The player will have the opportunity to engage in combat with the enemy and, upon victory, collect valuable treasure as a reward.

2. Use case diagram with scenario __14

Use Case Diagrams



1. Scenarios

Name: Encounter Enemy

Summary: The player encounters an enemy in the environment. The enemy can attack directly, or trigger environmental hazards such as falling rocks, collapsing ground, or hidden traps.

Actors: Player, Enemy System

Preconditions: The player is moving through the cave environment.

Basic sequence:

Step 1: Player comes into range of an enemy.

Step 2: The enemy attacks directly, if applicable.

Step 3: Environmental hazards may occur independently, requiring the player to react to falling rocks, collapsing ground, or triggered traps.

Step 4: The player must navigate both the enemy's attacks and the environmental hazards.

Exceptions:

Step 1: The player avoids the attack or bypasses the obstacle.

Post conditions: The player must choose to fight, avoid, or retreat.

Priority: 2*

ID: EE1

2. Scenarios

Name: Engage in Combat

Summary: The player fights with the enemy, and the result determines the next state.

Actors: Player, Enemy System

Preconditions: Player has encountered an enemy.

Basic sequence:

Step 1: Player initiates combat with enemy.

Step 2: Enemy responds with combat actions.

Step 3: Combat continues until one side is defeated.

Step 4: If the enemy is defeated, treasure is awarded.

Exceptions:

Step 1: Player is defeated, ending the encounter unsuccessfully.

Post conditions:

If enemy defeated → Player gains treasure.

If player defeated → Enemy remains active or restart the level.

Priority: 2*

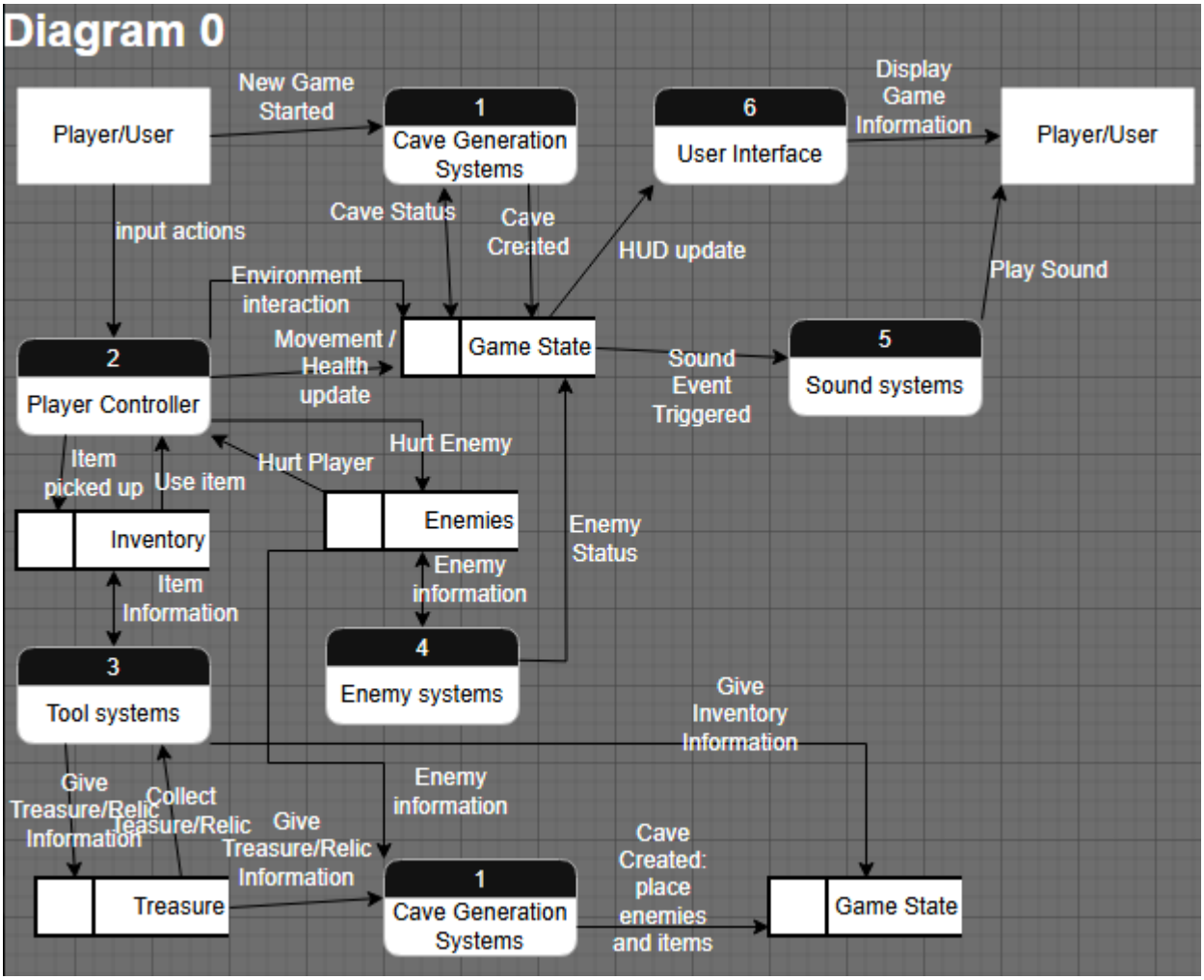
ID: EC1

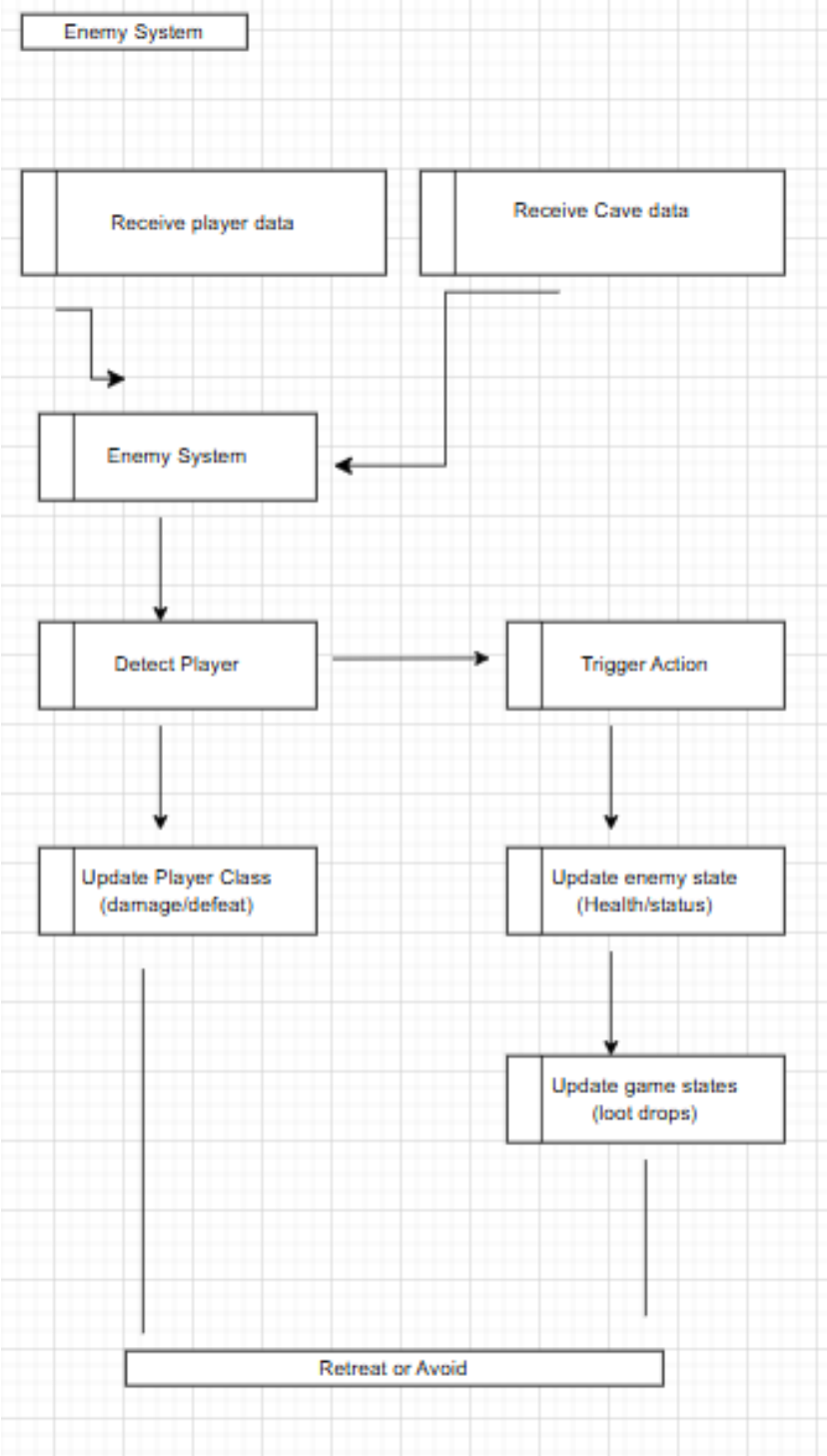
3. Data Flow diagram(s) from Level 0 to process description for your feature _____14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

Data Flow Diagrams





Process Descriptions

Enemy System:

The enemy system will be designed using an object-oriented approach, with a base Enemy class that defines shared attributes (e.g., health, detection range, attack power). Subclasses will represent

unique enemy types with distinct abilities, such as melee attackers or environmental hazard enemies.

Enemy instances use data from both the Player class (e.g., position, health, combat state) and the Environment class (e.g., cave layout, potential hazard zones). When a player encounters an enemy, the system triggers one of two paths:

Direct Combat Path – Enemy attacks the player directly, leading to health reduction or combat engagement.

Hazard Path – The enemy encounter triggers an environmental obstacle (e.g., falling rocks, blocked path, collapsing ground) that challenges the player's progress.

If combat occurs, the combat() function resolves until either the enemy or player is defeated. After resolution:

- Enemy defeated → Treasure is spawned for the player.
- Player defeated → Player health reduced or respawn triggered, enemy remains.
- Hazard encounter → Player must avoid, survive, or clear the obstacle to continue.

This design allows flexible enemy encounters while integrating natural cave hazards as part of the enemy experience.

4. Acceptance Tests _____9

Test Steps:

1. Generate mock player data (health, position, inventory).
2. Generate environmental data (cave layout, hazard zones).
3. Instantiate enemy objects (e.g., melee attacker, hazard-inducing enemy).
4. Move player into detection range.
5. Verify system triggers correct event: direct attack *or* environmental hazard.
6. If combat is triggered, simulate outcome (enemy defeated or player defeated).
7. Log results for verification.

Expected Output Characteristics:

- Enemy spawns correctly at designated locations.
- On encounter, system consistently triggers either combat or environmental hazard.
- Hazards (falling rocks, blocked paths, collapses) behave correctly and affect player progression.
- Combat results are accurate:
    - Enemy defeated → Treasure generated.
    - Player defeated → Health reduced/respawn.

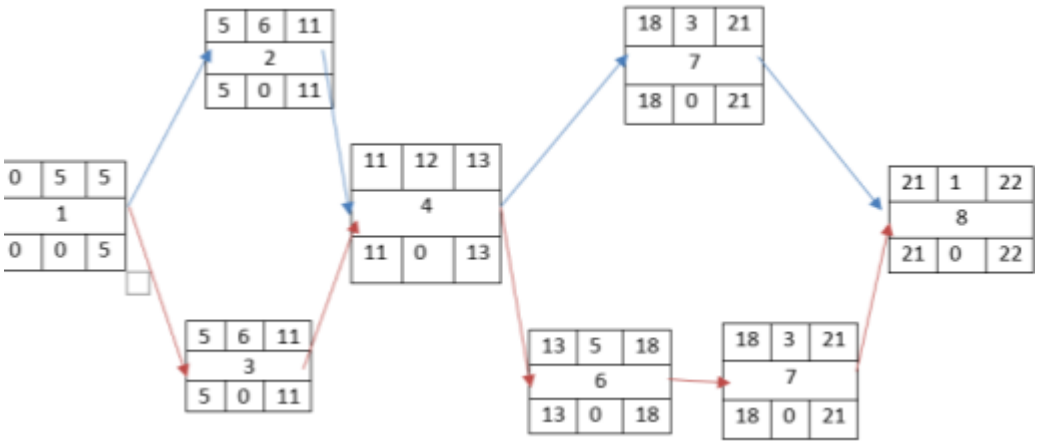
- Environment and player states update correctly (no ghost enemies, no duplicate treasures).
- System stable under repeated runs with multiple encounters.

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. **Define Enemy Concept**	5	-
2. **Design Enemy Mechanics**	6	1
3. **Integrate with Cave Generation**	6	1
4. **Implement Enemy AI**	2	2, 3
5. **Test Combat and Hazards**	6	4
6. **Add Loot and Feedback**	5	4
7. **Review and Debug**	3	6
8. **Documentation and Final Integration**	1	5, 7

Pert diagram



Gantt timeline