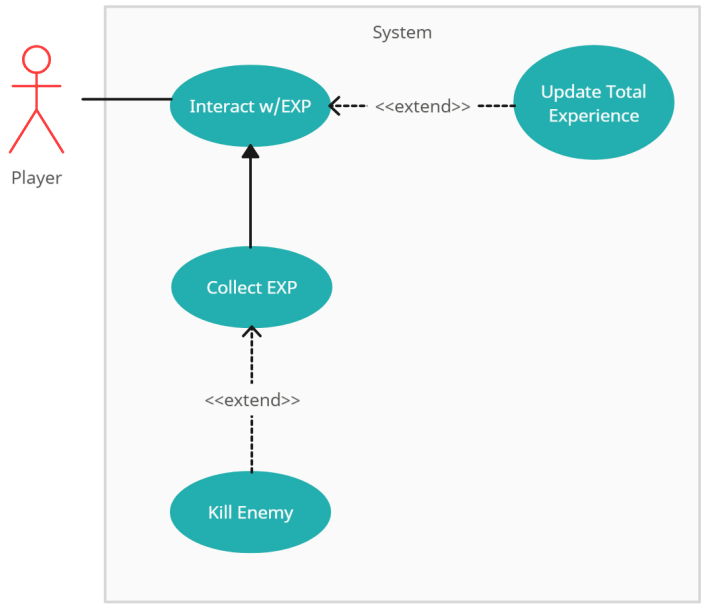
1. Brief introduction

My feature for CS Survivor is the experience (exp) collection for leveling up the player.

When the player kills an enemy, my job is to make sure that exp is dropped so that they can collect it. I need to make the exp collected interact with the level system, and the player can progress their level.

I'm also in charge of making sure the player can spend skill points gained by leveling up to unlock new skills or upgrade old skills.

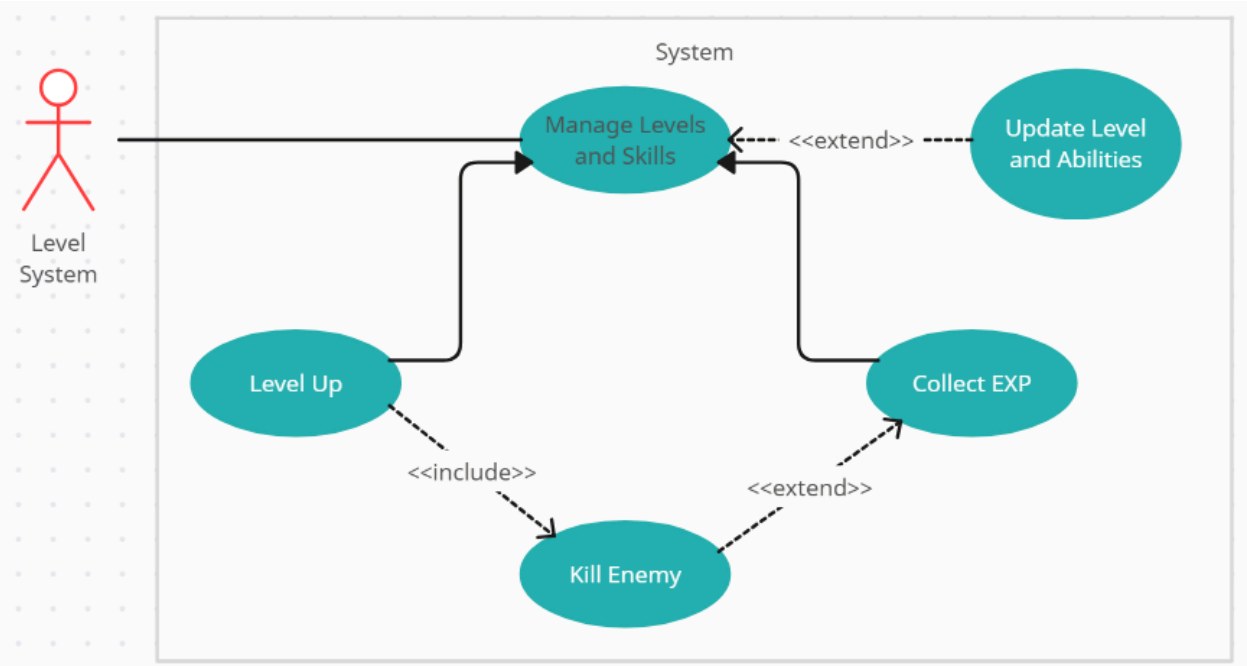
2. Use case diagram with scenario



Scenario 1:

Name: Interact with Exp
Summary: Player engages in combat with an enemy, defeats it, and collects the dropped experience points.
Actors: Player, Enemy
Preconditions: Player is in an area with enemies and is able to engage in combat
Basic Sequence:
Step 1: Player defeats enemy
Step 2: Enemy drops experience
Step 3: Player collects the experience
Step 4: Player's total experience is updated
Exceptions:
Step 1: No enemy is defeated
Step 2: No experience is collected

Post Conditions: Player's total experience is increased
Priority: 1
ID: EXP1

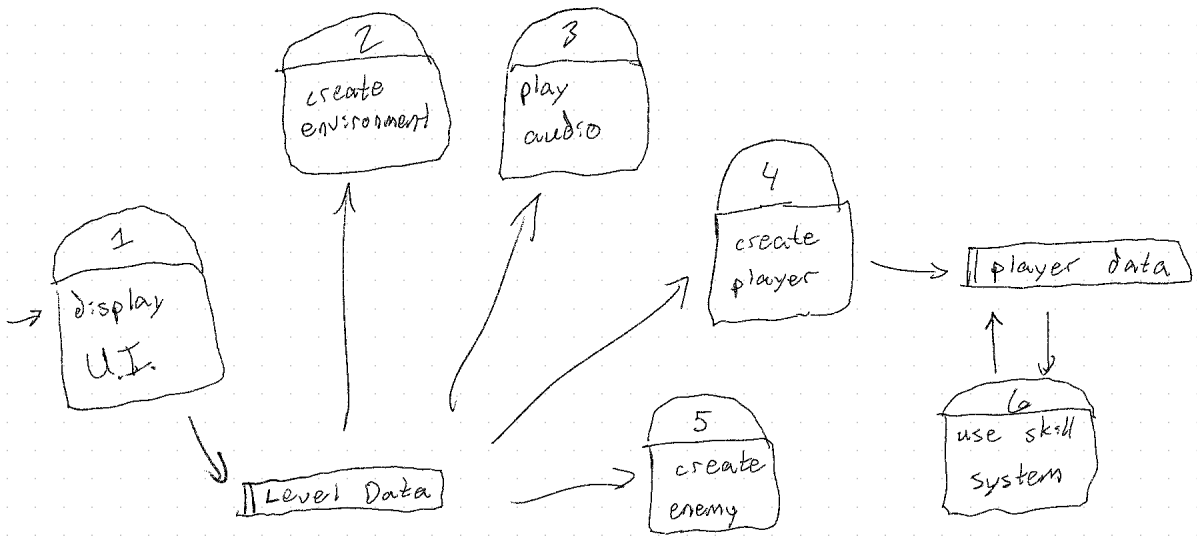


Scenario 2:

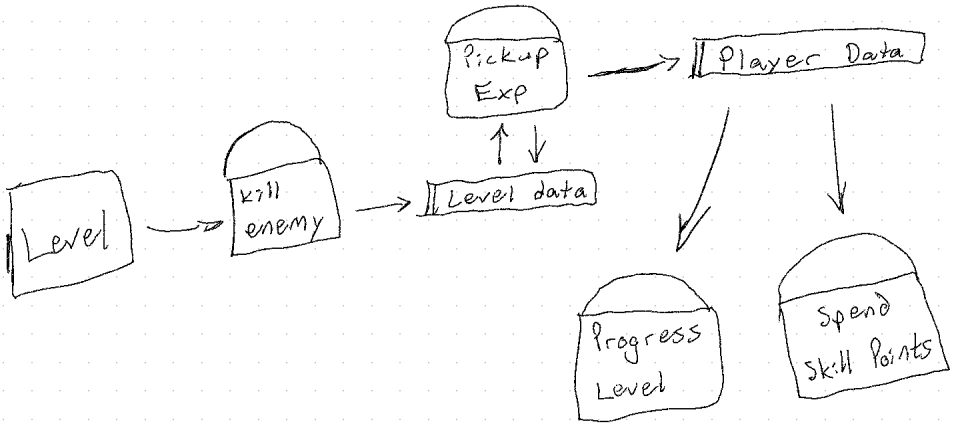
Name: Manage Levels and Skills
Summary: Player levels up after gaining exp, receives skill points, and spends them to improve abilities.
Actors: Level System, Actor
Preconditions: Player has collected exp
Basic Sequence:
Step 1: System detects player reached sufficient exp for level up
Step 2: Player gains skill point(s)
Step 3: Player chooses to spend skill point(s)
Step 4: Player's abilities are updated based on skills chosen
Exceptions:
Step 1: Player does not have sufficient skill point(s) to spend
Step 2: Player did not level up or choose a skill
Post Conditions: Player's level, skills, and skill points are updated
Priority: 1
ID: SKL1

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

Diagram 0:



Use Skill System:



Process Descriptions:

```
KillEnemy() {
  If (enemy not dead) {
    If (player hits enemy) {
      Enemy health decreases;
    }
  }
  Else {
    Drop exp;
    Destroy self;
  }
```

```
}

PickupExp() {
  If (player is within certain radius of exp) {
    Pick up exp;
    ProgressLevel();
    Destroy exp object;
  }
}

ProgressLevel() {
  Increase exp bar;
  If (exp == desired exp for level up) {
    Increase player level;
    Skill Points++;
    Reset exp bar;
    Increase desired exp for level up;
  }
}

SpendSkillPoints() {
  If (player clicks a skill in the skill tree ui) {
    If (player has sufficient skill points for that skill) {
      Upgrade the skill that was clicked;
      Change skill tree ui;
    }
    Else {
      Provide error message;
    }
  }
}
```

4. Acceptance Tests

For exp drop and collection, the player kills an enemy to make it drop exp so that they can collect it. This is done 100 times. Killing an enemy should drop exp. Collecting exp should progress the player's level.

Example Output Test Check:

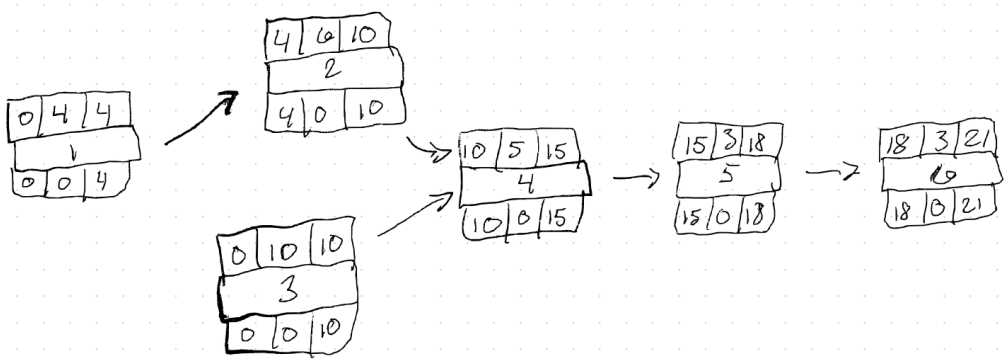
Test	**# of success**	**# of fail**	**% success**
Exp Drop	90	10	90%
Exp Collect	80	20	80%

5. Timeline

Work Items

Task	Duration (hours)	Predecessor Task(s)
1. Exp Drop & Collection	4	-
2. Exp/Level Progression	6	1
3. Skill Point & Skill Tree Implementation	10	-
4. UI development	5	2, 3
5. Testing & Balancing	3	4
6. Deployment	3	5

Pert Diagram



Gantt Chart