WHERE'S MY SPACESHIP?

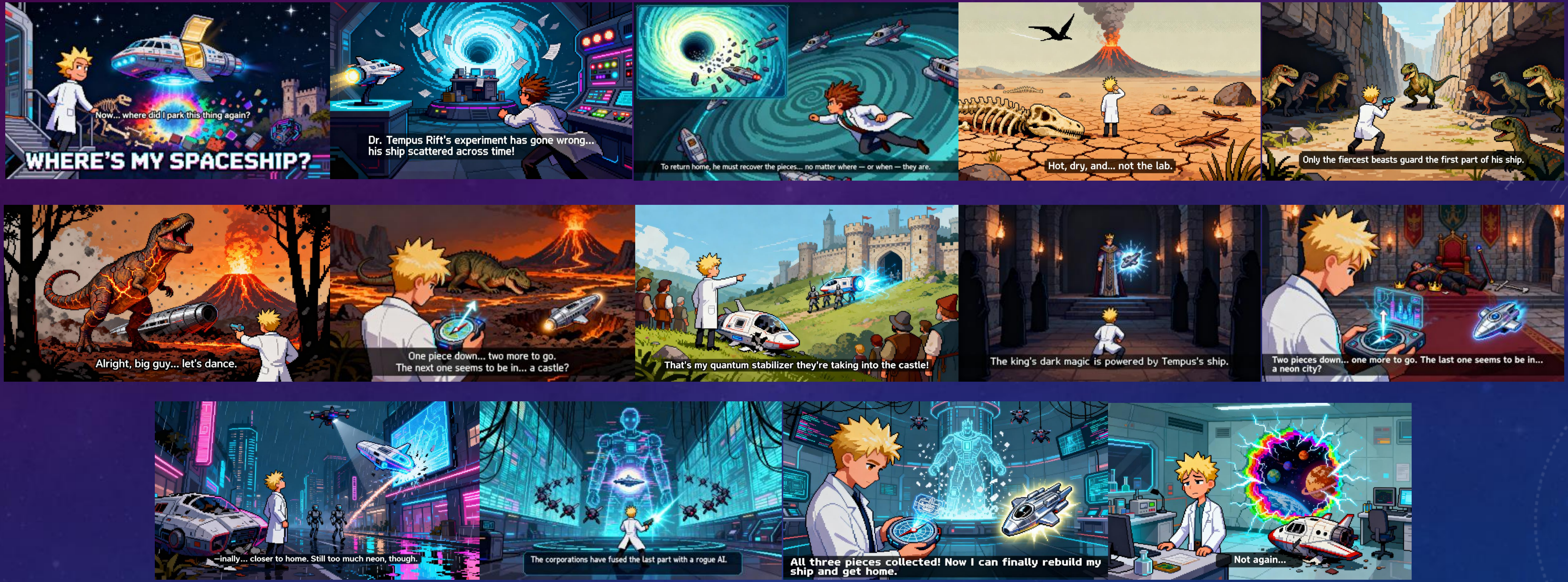
YOU'RE SUCKED INTO A TIME RIFT, AND YOUR SHIP IS TORN TO PIECES, SCATTERING ACROSS THE EONS. CAN YOU RECOVER THEM ALL?

LOST MAP STUDIO

GROUP MEMBERS: RESPONSIBILITIES:

1. Adam: IT Manager	Inventory, Menus, UI
2. Gabe: Software Architect	Player Control / Character
3. Alex: QA Manager	Level Design
4. Urvashi: Project Manager	Sound
5. AJ: AI Specialist	Enemies
6. Qiwei: Version Control Manager	Boss fights

STORYBOARD


Now... where did I park this thing again?
WHERE'S MY SPACESHIP?
Dr. Tempus Rift's experiment has gone wrong... his ship scattered across time!
To return home, he must recover the pieces... no matter where — or when — they are.
Hot, dry, and... not the lab.
Only the fiercest beasts guard the first part of his ship.
Alright, big guy... let's dance.
One piece down... two more to go. The next one seems to be in... a castle?
That's my quantum stabilizer they're taking into the castle!
The king's dark magic is powered by Tempus's ship.
Two pieces down... one more to go. The last one seems to be in... a neon city?
—inally... closer to home. Still too much neon, though.
The corporations have fused the last part with a rogue AI.
All three pieces collected! Now I can finally rebuild my ship and get home.
Not again...

SCENE 1:

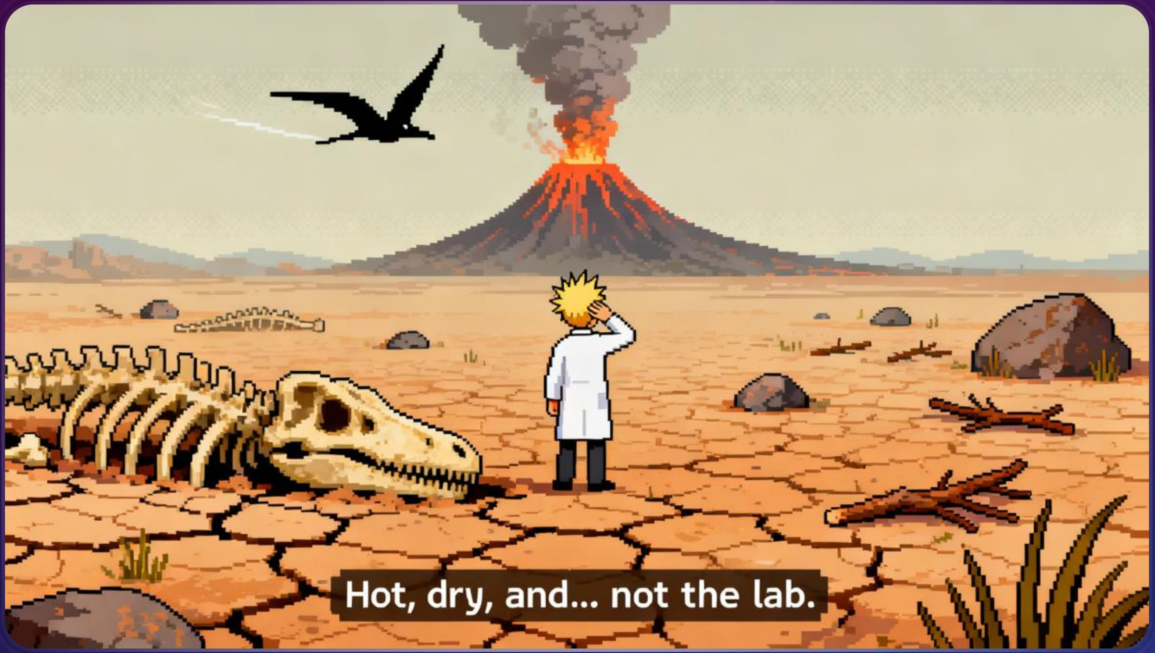
Dr. Tempus Rift's experiment has gone wrong...
his ship scattered across time!

- Dialogue:
  Dr Tempus Rift: "I got to go after mt Spaceship!!!"

- Action:
- Load player into main lobby
- Time Rift on the right to start the game (by walking into it)

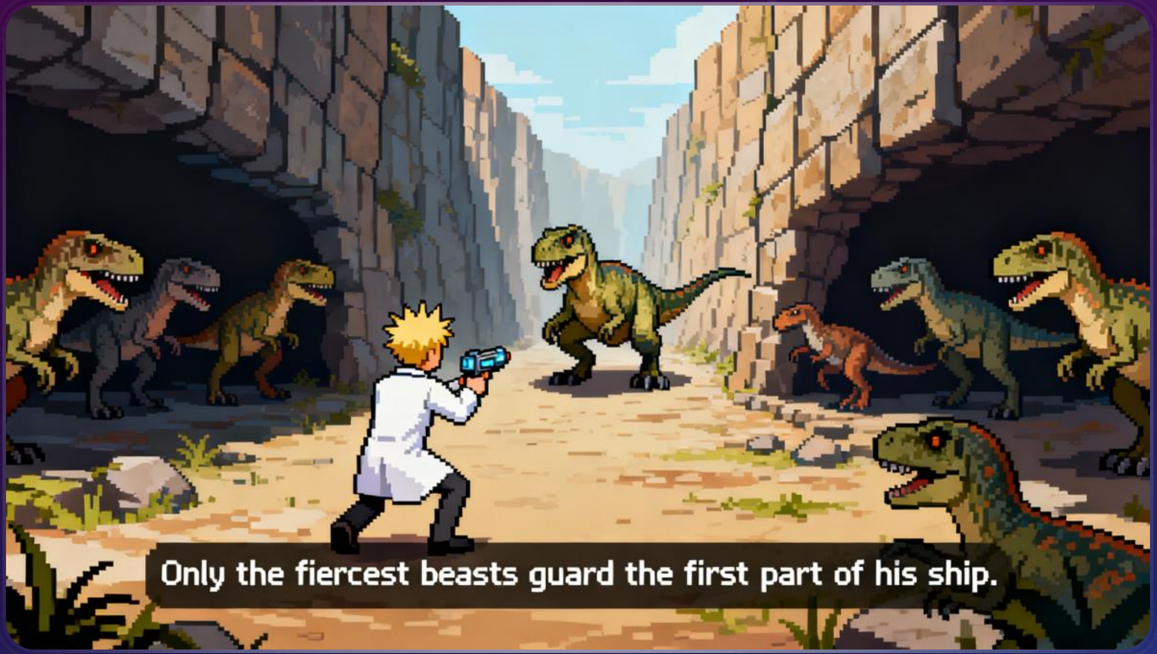
- Notes:
- Laboratory: Panels, computers, 4 walls, Time Rift to the right
- Music playing

SCENE 2:



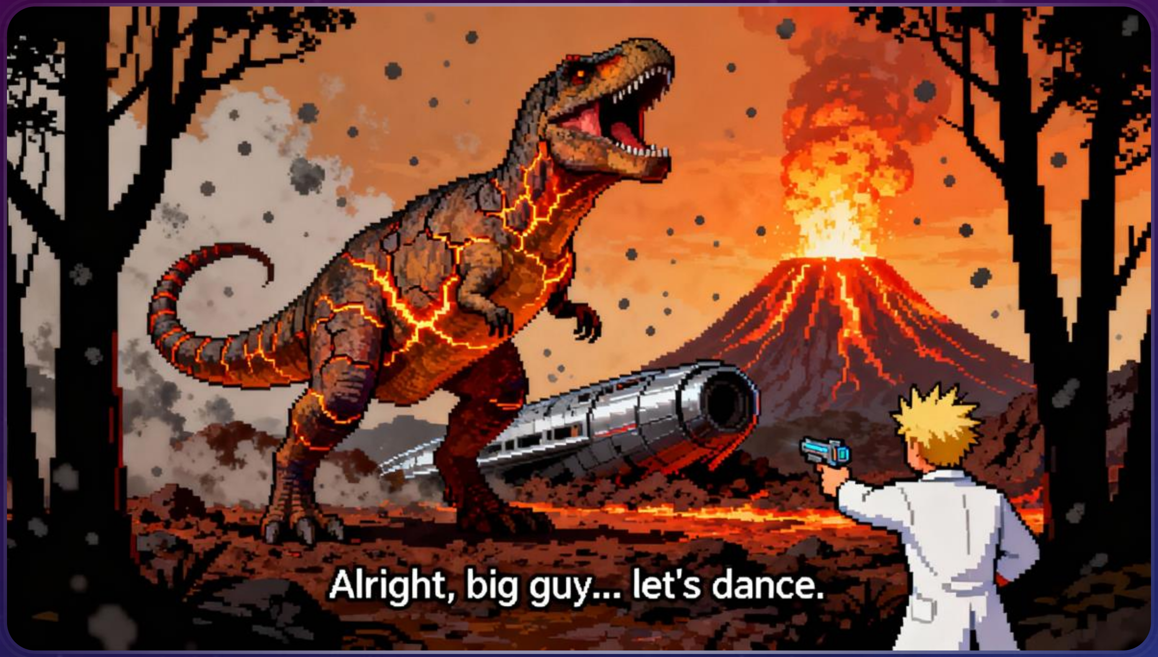
- Dialogue:
  Dr Tempus Rift: "Okay, where is my spaceship?"
- Action:
  Load Player into the first scene (Desert Wasteland)
- Load monsters
- Notes:
- Big volcano in the background
- Desert style: rocks, bones, trigs
- Audio/music playing

SCENE 3:



- Dialogue:
  None
- Action:
- Player Navigates through the environment
  Enemy interactions
- Weapon usage
- Notes:
- Desert style: rocks, bones, trigs
- Audio/Music playing

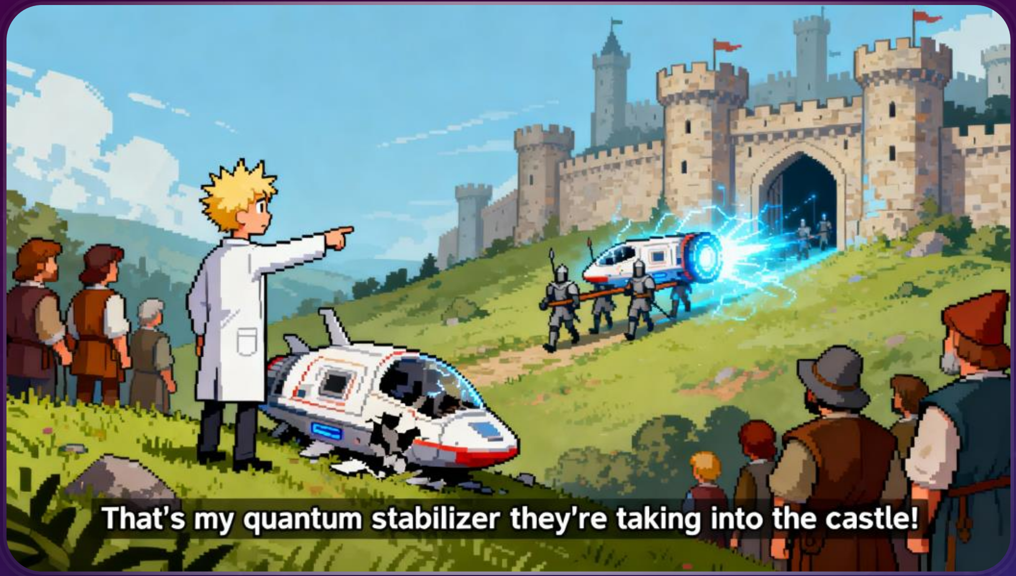
SCENE 4

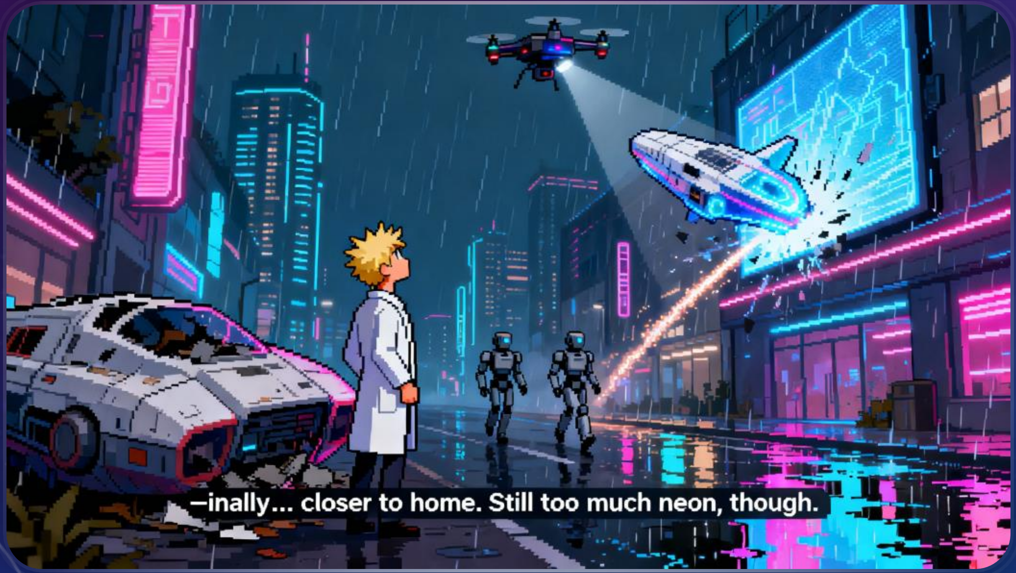


- Dialogue:
  Dr Tempus Rift: "Why are you right where I need to be..."
- T-REX: "Big load Roar"

- Action:
- Boss Fight
- Interactables

- Notes:
- Desert style: rocks, bones, trigs/trees
- Boss music playing


That's my quantum stabilizer they're taking into the castle!

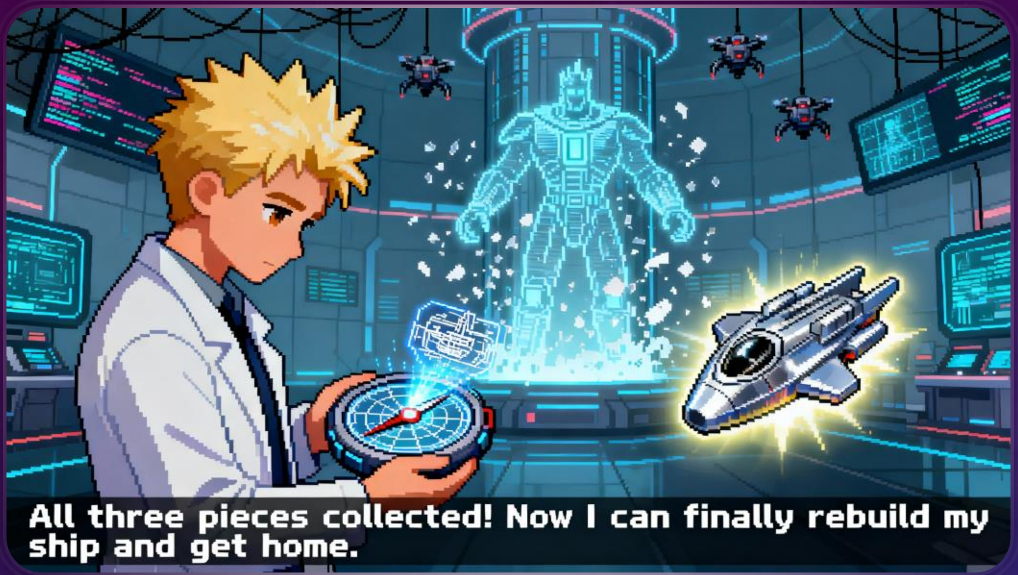


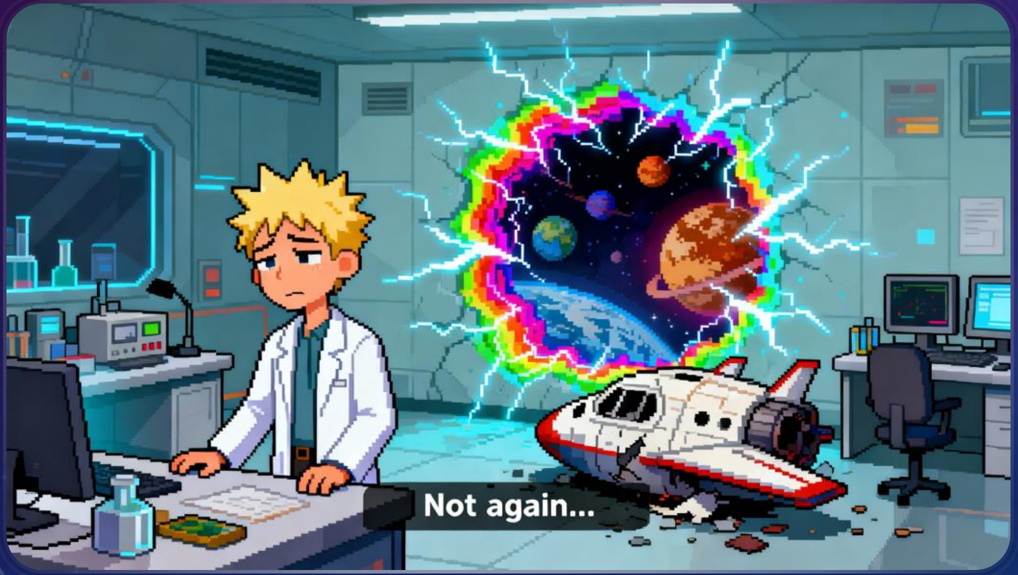
SCENE 5 & 6

- Dialogue:
  None

- Action:
- Loading next stages after boss fights
- Player continues the different era's

- Notes:
- Different backgrounds for respective eras
- Similar elements from previous scenes
- Audio/Music playing though out the Stages/Eras


All three pieces collected! Now I can finally rebuild my ship and get home.



SCENE 7 & 8

- Dialogue:
  None
- Action:
- Player gets all pieces for the spaceship therefore winning the game.
- Option to restart
- Notes:
- Small cutscene
- Menu opening up
- No sounds/music

CONTEXT DIAGRAM

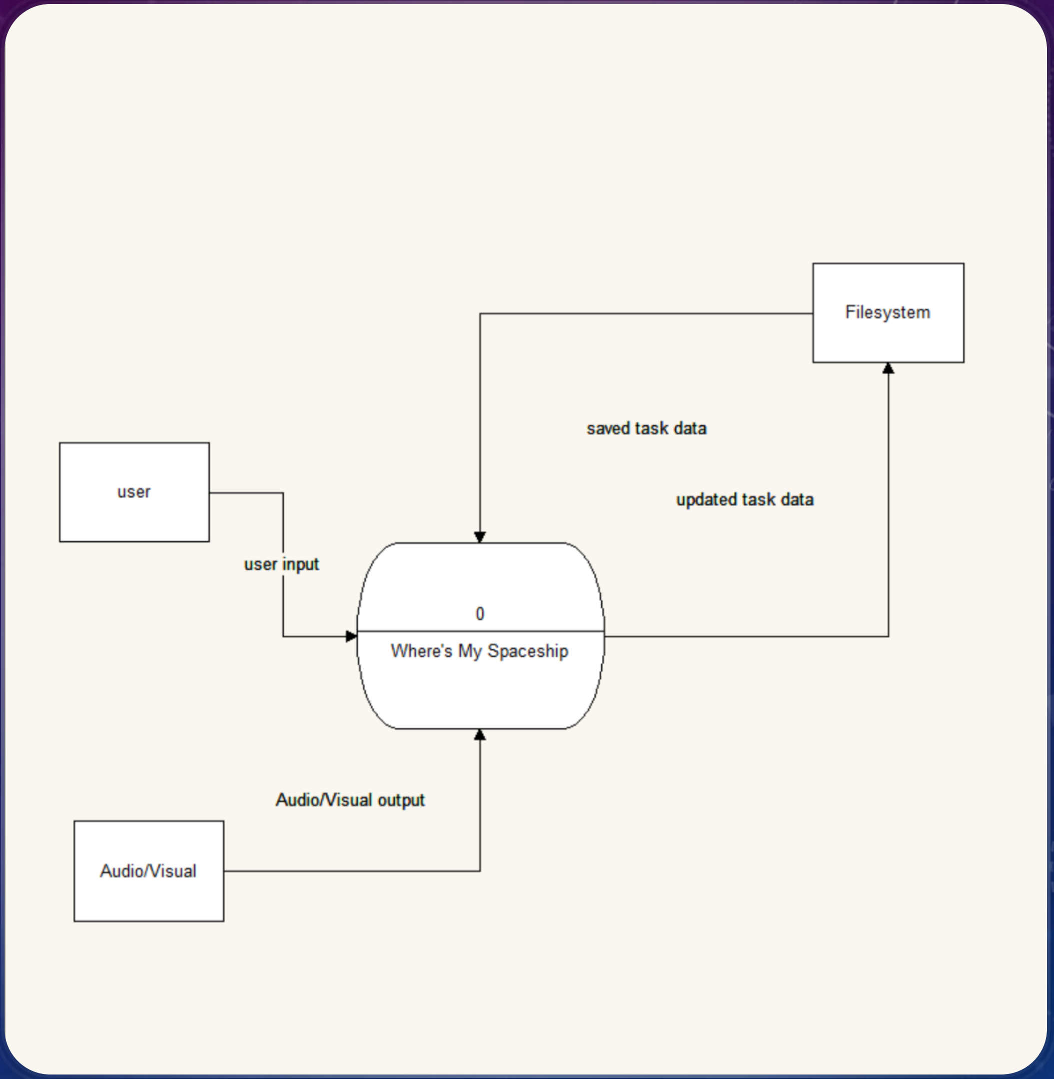
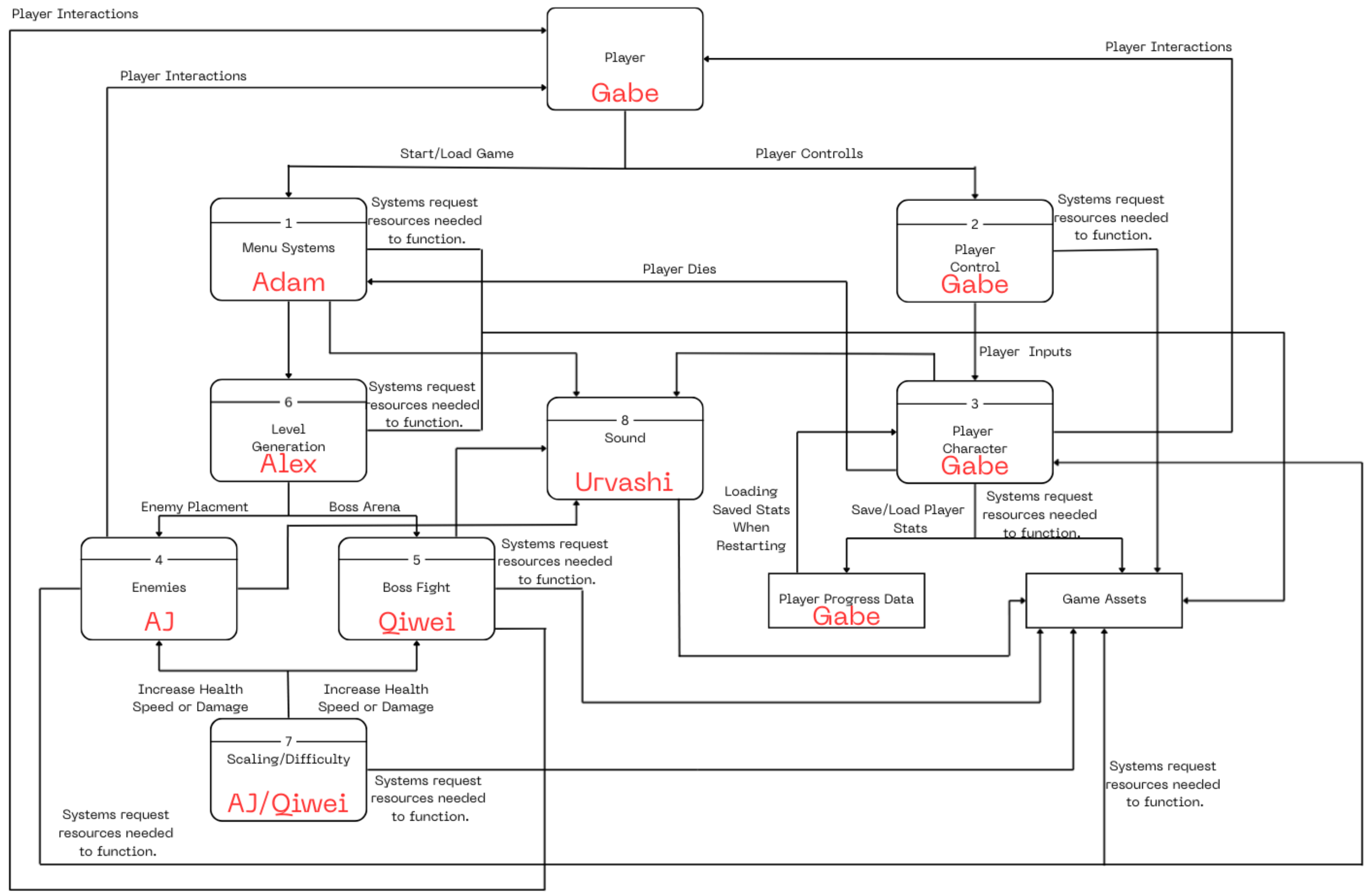
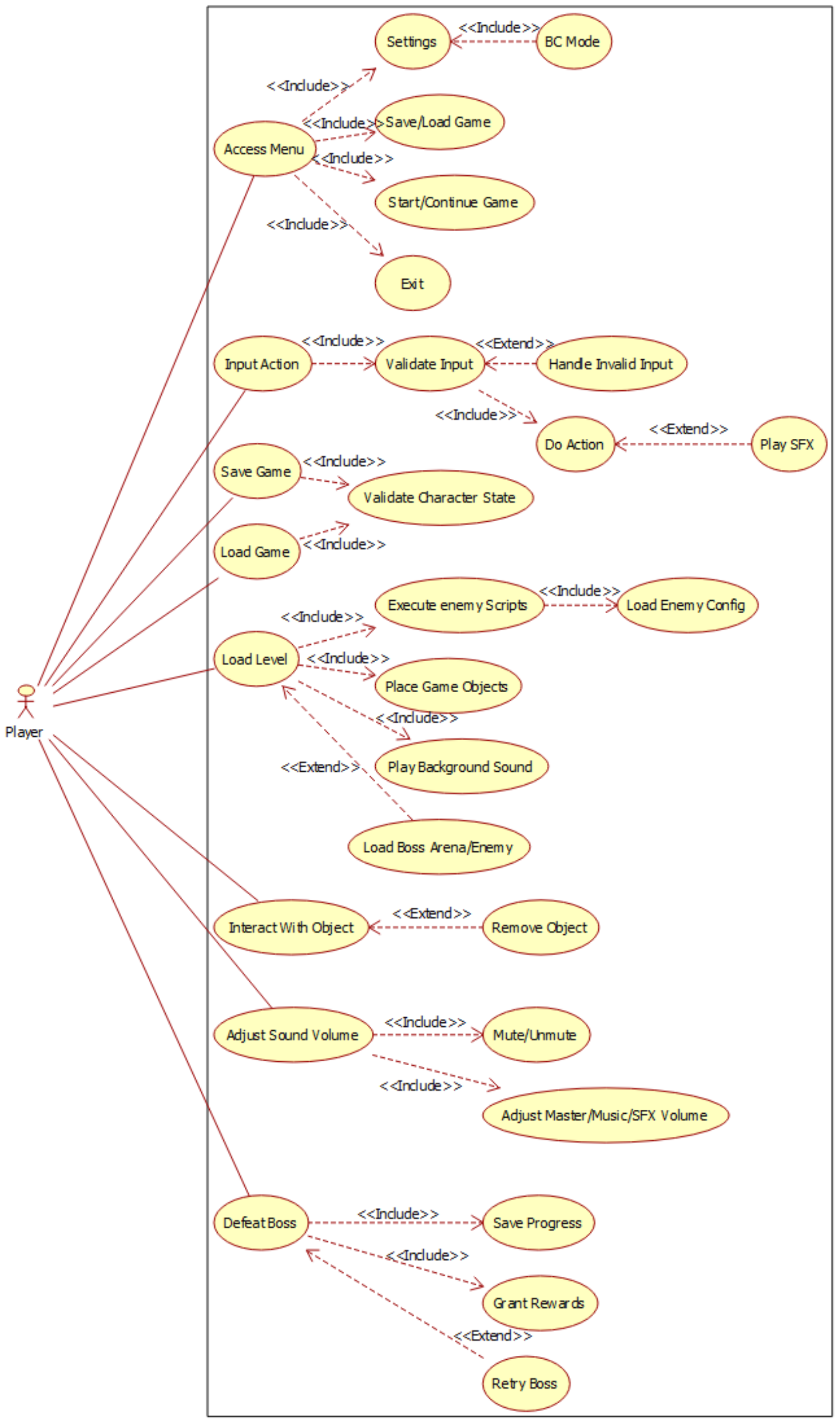
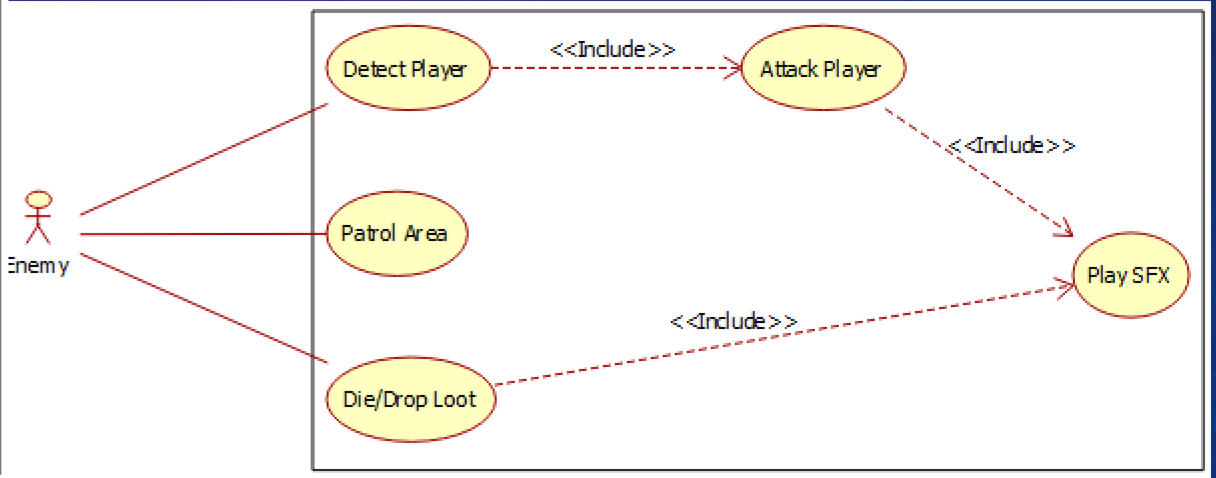


DIAGRAM 0


Player Interactions
Player Interactions
Player
Gabe
Player Interactions
Start/Load Game
Player Controlls
1
Menu Systems
Adam
Systems request resources needed to function.
2
Player Control
Gabe
Systems request resources needed to function.
Player Dies
Player Inputs
6
Level Generation
Alex
Systems request resources needed to function.
8
Sound
Urvashi
3
Player Character
Gabe
Loading Saved Stats When Restarting
Save/Load Player Stats
Systems request resources needed to function.
Enemy Placment
Boss Arena
4
Enemies
AJ
5
Boss Fight
Qiwei
Systems request resources needed to function.
Player Progress Data
Gabe
Game Assets
Increase Health Speed or Damage
Increase Health Speed or Damage
7
Scaling/Difficulty
AJ/Qiwei
Systems request resources needed to function.
Systems request resources needed to function.
Systems request resources needed to function.

GLOBAL USE CASE





GLOBAL USE CASE SCENARIOS

Refer to the individual use case scenarios on the below slides:

- 17 for Access Menu
- 21, 22 for input Action, Save Game, Load Game
- 26, 27 for Load Level, Interact with Object
- 29 for Adjust Sound Volume, Play SFX
- 40 for Defeat Boss
- 33, 34, 35 for Detect Player, Patrol Area, Die/Drop Loot

*The above includes any includes/extends relationships except Play SFX

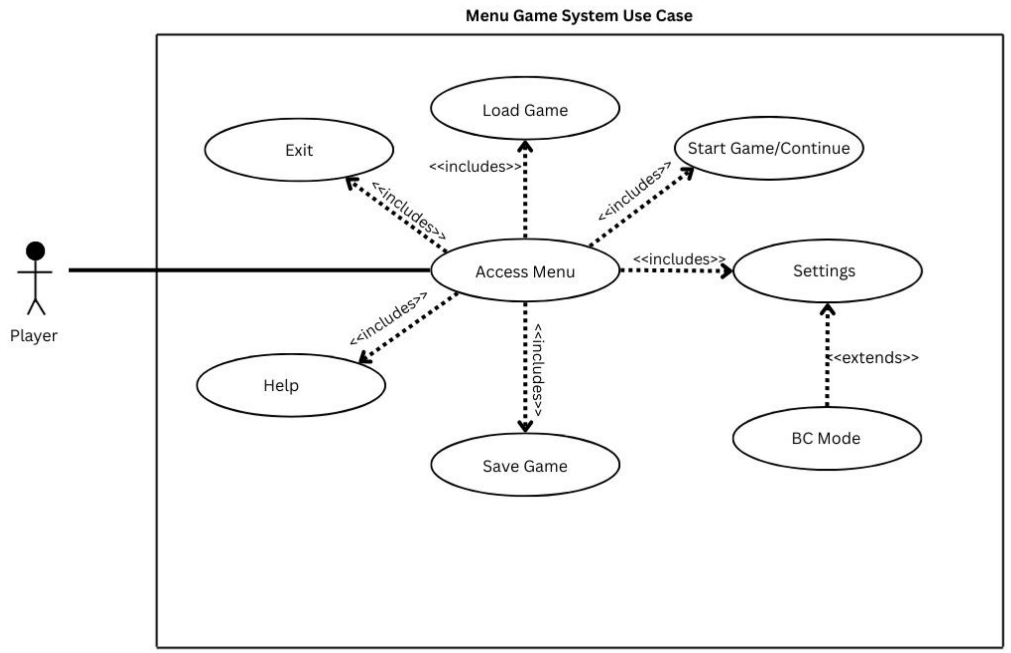
INDIVIDUAL PARTS

ADAM: MENU MANAGEMENT

- **Providing Game access:**
  - Start Game
  - Continue
  - Load
  - Save
- **Allowing Exiting/Pausing:**
  - Exiting to main Menu safely.
  - Pause Game Freezing the Gameplay while player navigates the menu.
- **Adjusting Sounds:**
  - Adjust volume for music and sound effects.
- **Providing Help and Guidance:**
  - Instructions: explaining the controls and objectives.

Priority: **Medium-High** (Settings/audio not needed, but Needed for game to start and pause)
Complexity: **Low-Medium** (Easy enough to make a menu screen for pause and resume but not as complex as the other systems.

ADAM: USE CASE DIAGRAM



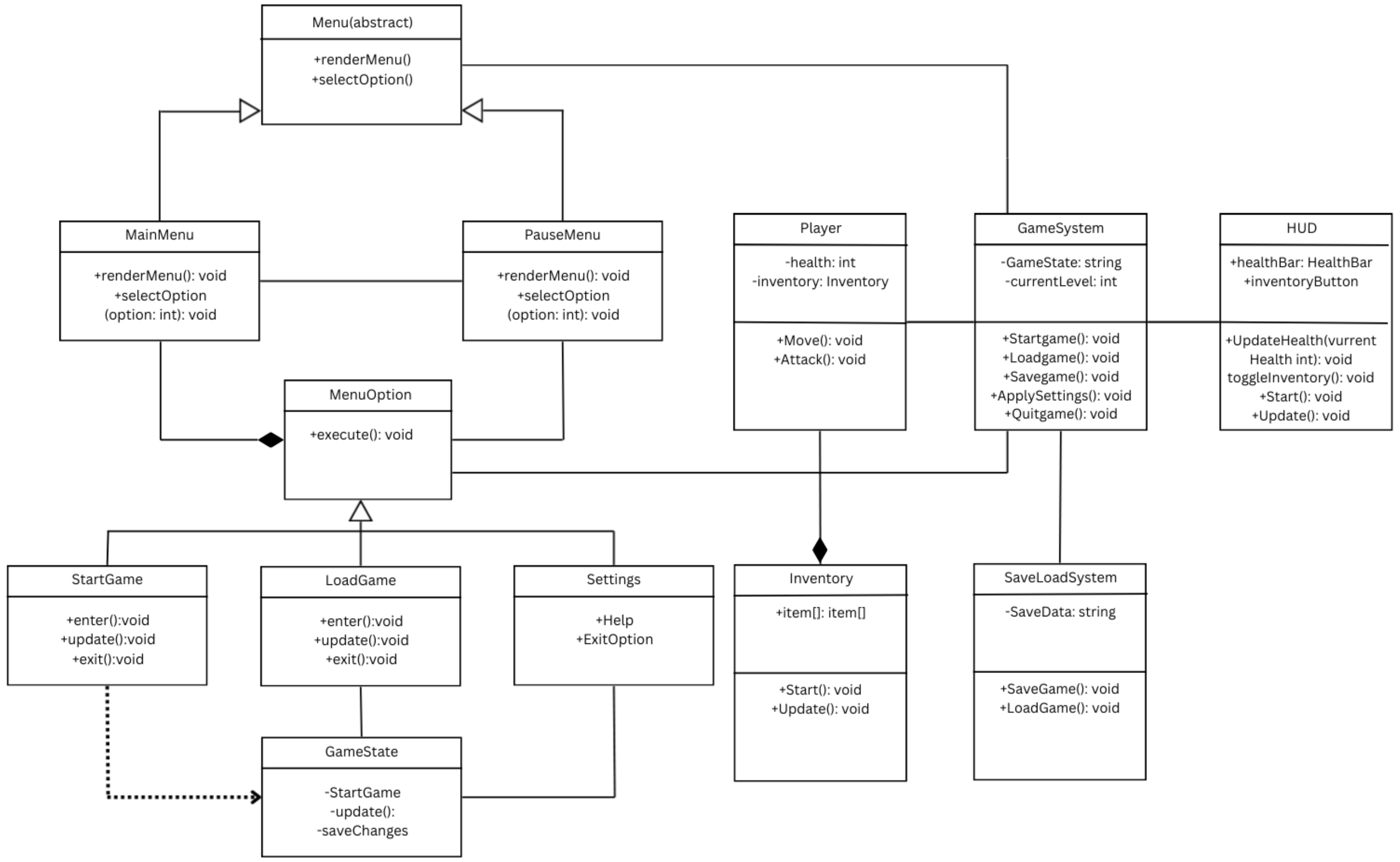
ADAM: USE CASE SENARIO

- **Step 1:** The player accesses the main menu.
- **Step 2:** The system displays available menu options.
- **Step 3:** The player selects **Start Game/Continue**.
- **Step 4:** The system initializes or resumes the game state and begins gameplay.

Exceptions:

- **Step 3.1 (Load Game):** The player selects **Load Game**. The system retrieves previously saved data and resumes the game.
- **Step 3.2 (Save Game):** The player selects **Save Game**. The system stores the current progress. If saving fails, the player is prompted to try again.
- **Post conditions:** The player's choice is executed successfully (game started, loaded, saved, configured, help shown, or exited).
- **Priority:** 1*
- **ID:** M01
- *The priorities are 1 = must have, 2 = essential, 3 = nice to have.*

ADAM: MENU/INVENTORY (CLASS DIAGRAM)



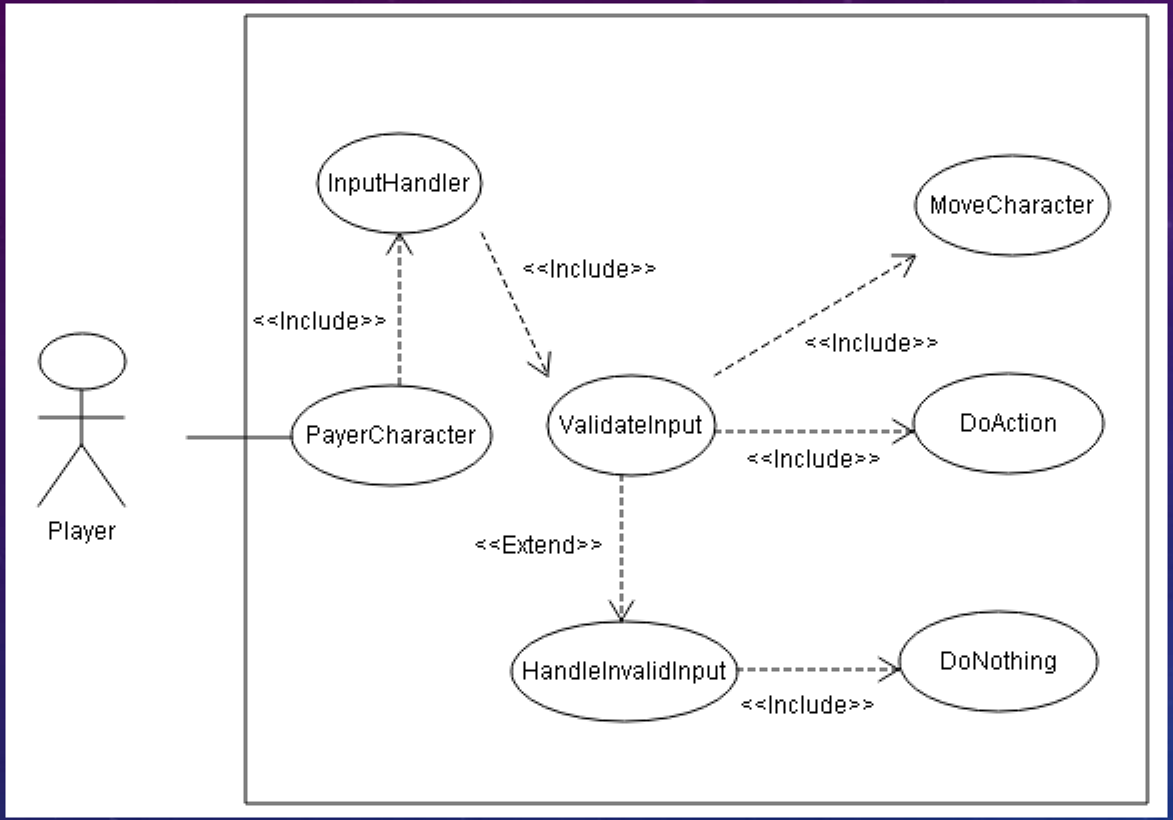
TL2

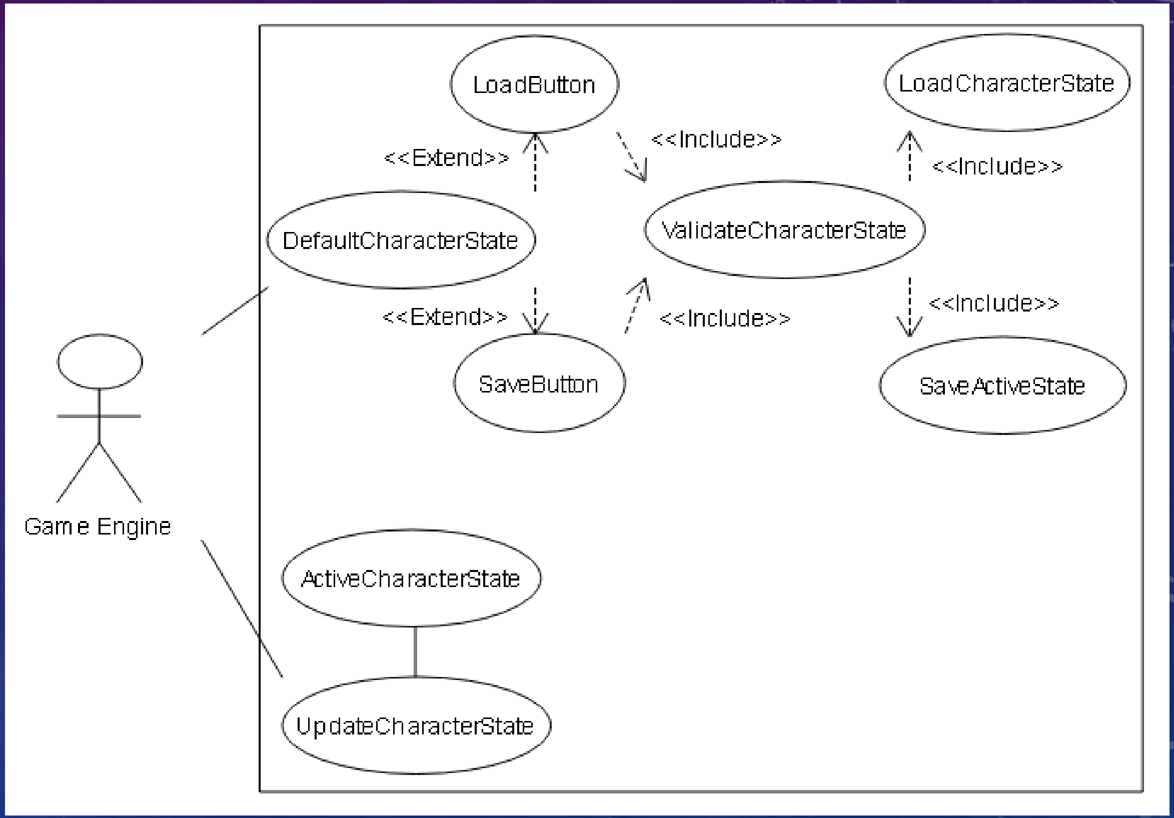
GABE (PLAYER CONTROL / CHARACTER)

- Control Feature:
    - Allows the player to use W,A,S,D keys to control the in-game character's movements.
    - Ability to attack enemies with left mouse-click, Melee || Ranged.
    - Renders and uses Animation based on character's movement.

- Character Feature:
    - Manages all character related items/ progress.
        - Progress Includes Weapon(s) Collected, World(s) Visited, Puzzles Solved, Obstacles Cleared, Health Amount, and Bosses Defeated.
    - Implements a save-state within game to save progress made within the game.

GABE (PLAYER CONTROL/CHARACTER) USE CASE DIAGRAM





TL2

GABE (PLAYER CONTROL) USE CASE SCENARIO

1. Player Launches Game.

2. Player Starts The Game.

3. Player Uses W,A,S, or D to move the character.
   3.1 Animations used corresponding to actions / movement.

4. Player Uses Left Click to perform equipped attack(s).
   4.1 Melee Weapon Attack
   4.2 Ranged Weapon Attack

5. As Player Explores:
   5.1 Weapons Are Collected
   5.2 Worlds Are Visited
   5.3 Puzzles Are Solved
   5.4 Bosses Are Defeated
   5.5 Obstacles Are Cleared

6. Character's Health Is Updated Based On Damage And Healing.

7. All Progress Is Tracked By Character System.

Preconditions:

Game is installed and running successfully.
Player has access to keyboard + mouse.

Exceptions:

No input detected.
Invalid key pressed(Ignored).
Game freezes during movement.

TL2

GABE (PLAYER CHARACTER) USE CASE SCENARIOS

1. Player Arrives At A Checkpoint.
   1.1 Player Given A Button To Save Game.

2. System Captures Active Character State:
   2.1 Character Health
   2.2 Weapons Collected
   2.3 Puzzles Solved
   2.4 Obstacles Cleared
   2.5 Bosses Defeated
   2.6 Worlds Entered

3. System Writes Save Data To File.

4. Player Is Alerted A Save Has Successfully Taken Place.

5. On Menu Screen, Player Has Option To Load Game.

6. The Load Engine:
   6.1 Accesses Save File Location
   6.2 Loads The Saved Active Information Into Current Active Information

Preconditions:

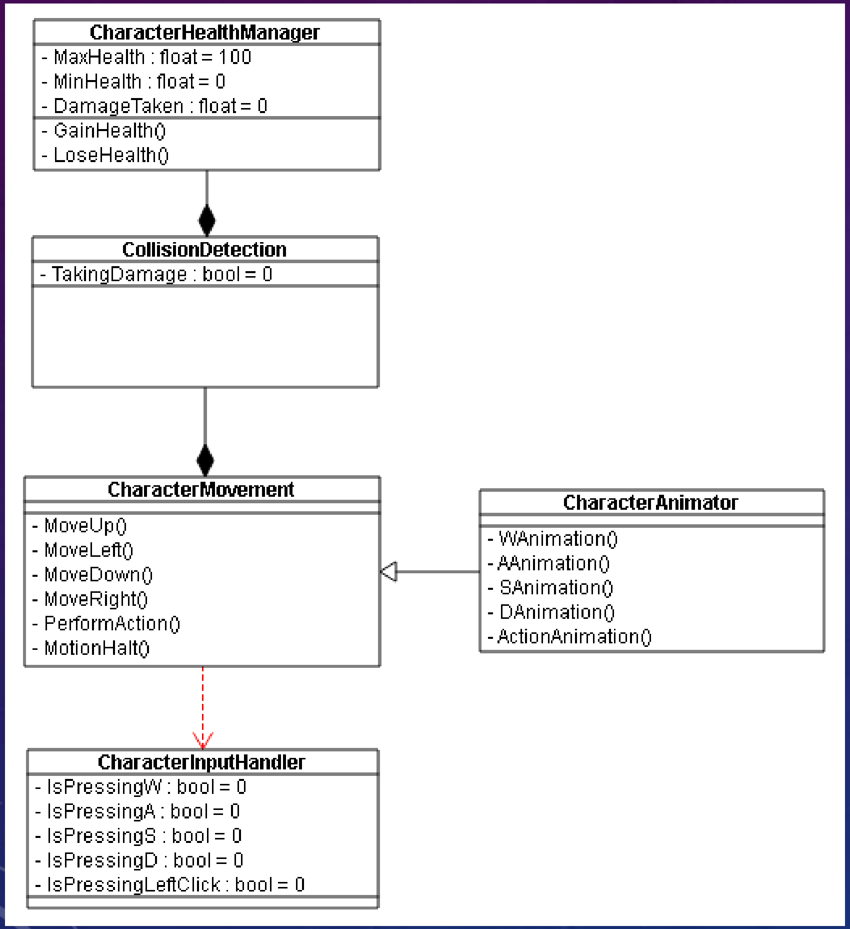
Game is installed and running successfully.
Menu is accessible
Save system is enabled
Load system is enabled
Storage access is available to system

Exceptions:

If save fails, alerts player and retries save
Game crash during save, system attempts rollback

TL2

GABE (PLAYER CONTROL) CLASS DIAGRAM



PRIORITY: **(High)**

Without a character for the player to control, the game will be unplayable.

Without a movement mechanic, the game will not progress.

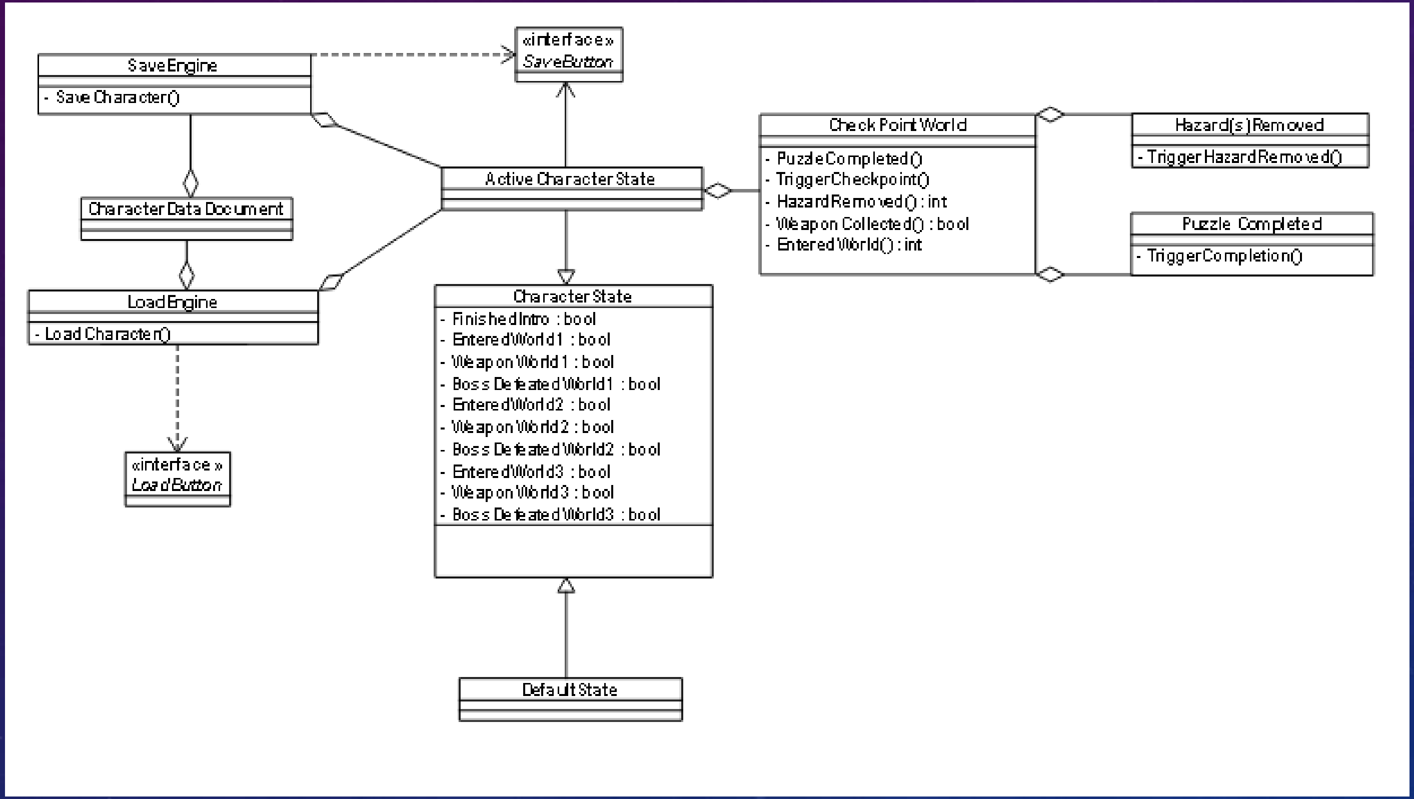
Without Health or Weapons there is no combat system.

COMPLEXITY: (**Low-Medium**)

Simple Unity commands exist for tracking input, and translating to player movement is relatively easy. Compared to other systems within the game, this will be one of the easier systems. May provide a challenge when presented with multiple weapons + animations to go with those weapons.

TL2

GABE (PLAYER CHARACTER) CLASS DIAGRAM



PRIORITY: (**Low**)
  This Feature is unnecessary but will provide a quality-of-life feature that will make the game more "beginner friendly" by allowing them to not start completely over, especially on a loss. This feature was more thought of as a challenge to myself and something I would love to implement.

COMPLEXITY: (**High**)
  Will require extensive research on other game's save features, how to write to a disk or file, and reading from the specified storage spot. At the time of starting this project, it feels as though this may be one of the more complicated systems to implement in our game.

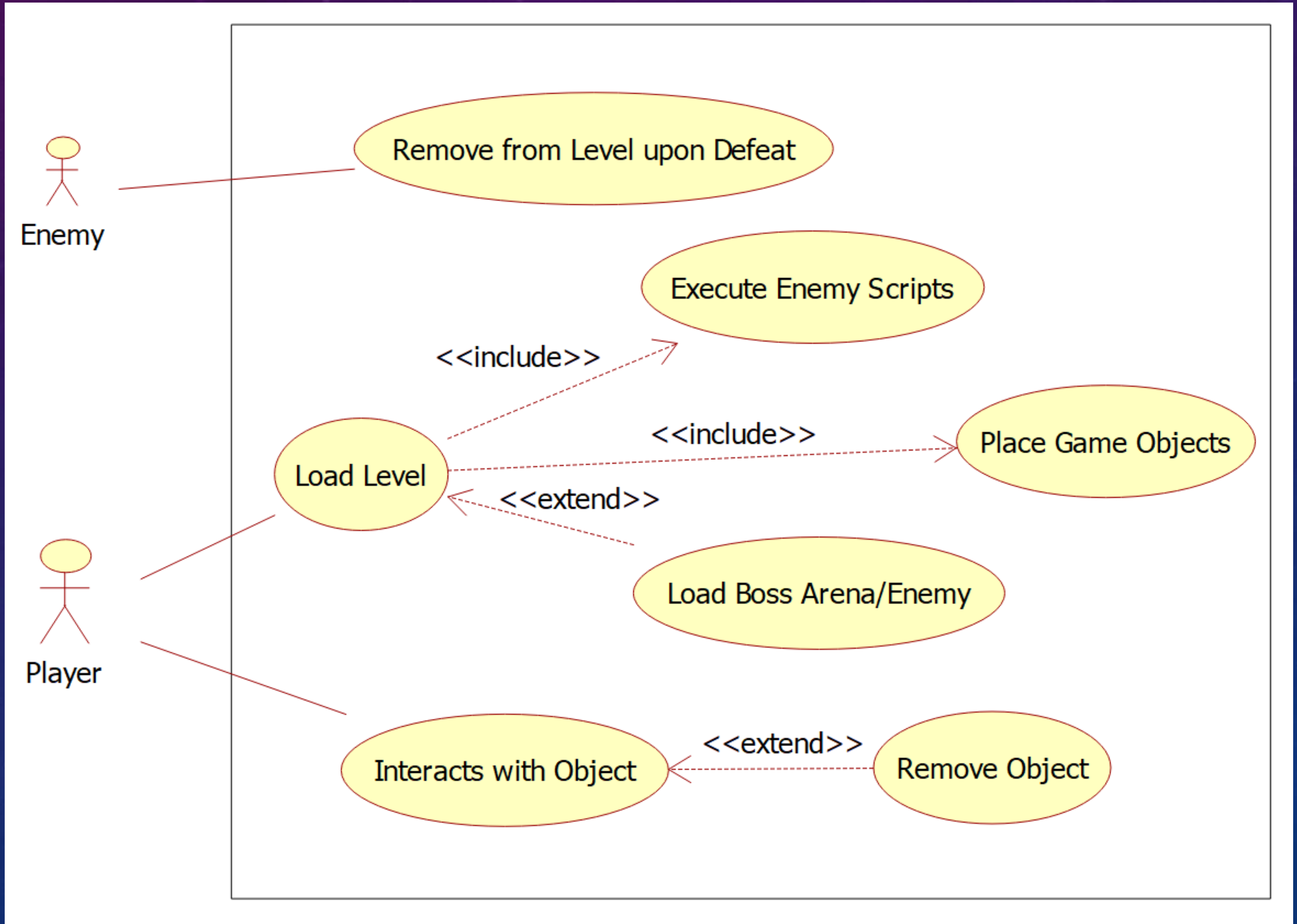
ALEX (LEVEL DESIGN) OVERVIEW

- Loading new scenes
- Creating terrain
- Creating interactable objects/puzzles
- Placing enemies

Priority: High (player needs a space to interact with)

Complexity: Medium (includes making objects with unique behaviors)

ALEX (LEVEL DESIGN) - USE CASE DIAGRAM



TL3

ALEX (LEVEL DESIGN) - USE CASE SCENARIOS 1

- **Name:** Remove from Level upon Defeat
  - **Summary:** An enemy is removed from the level when it is defeated.
  - **Actors:** Enemy
  - **Preconditions:** Enemy health <= 0
  - **Basic sequence:**
    - **Step 1:** Player damages an enemy until its health reaches zero.
    - **Step 2:** Save enemy ID so it isn't spawned when going back to the level.
  - **Exceptions:**
    - None
  - **Post conditions:** The instance of the enemy will no longer be spawned in the current level.
  - **Priority:** 3*
  - **ID:** AE01

- **Name:** Interacts with object
  - **Summary:** A player interacts with an object in the level and it changes something about the level.
  - **Actors:** Player
  - **Preconditions:** Level is loaded and running, player does something to the object (specific to each object)
  - **Basic sequence:**
    - **Step 1:** Player is in a level and performs some action on an interactable object.
    - **Step 2:** Run some script based on the object to perform some specific action.
  - **Exceptions:**
    - **Step 1:** If object is destroyed, remove it from the level.
  - **Post conditions:** The action of the specific object is executed.
  - **Priority:** 2*
  - **ID:** AE01

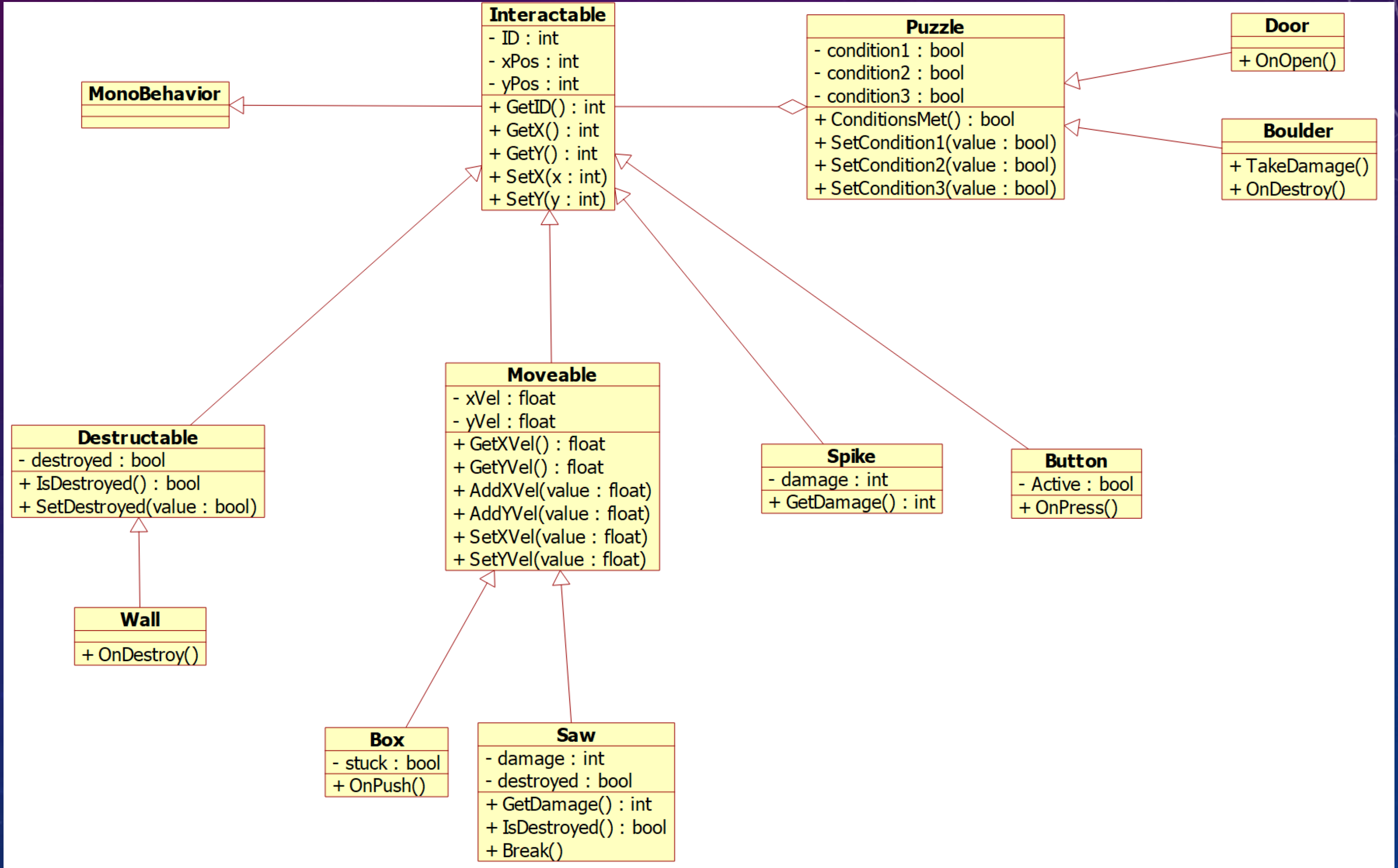
TL3

ALEX (LEVEL DESIGN) - USE CASE SCENARIOS 2

- **Name:** Load Level
    - **Summary:** A player selects a level or loads a save, and the game loads the appropriate level.
    - **Actors:** Player
    - **Preconditions:** Player has selected a level or loaded a save
    - **Basic sequence:**
        - **Step 1:** The player starts the game and selects a level or loads a save.
        - **Step 2:** Load the scene containing the appropriate level, which includes the terrain, any intractable objects, enemies, and sounds. It also include the placement of and assets for those.
        - **Step 3:** Run the scripts for any enemies and objects in the area.
    - **Exceptions:**
        - **Step 2:** The area being loaded is a boss area, in which case the player will be transported to the next area upon victory.
    - **Post conditions:** The appropriate level is loaded and the player can interact appropriately with it.
    - **Priority:** 1*
    - **ID:** AE02

*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

ALEX (LEVEL DESIGN) - CLASS DIAGRAM



URVASHI (SOUND DESIGN OVERVIEW)

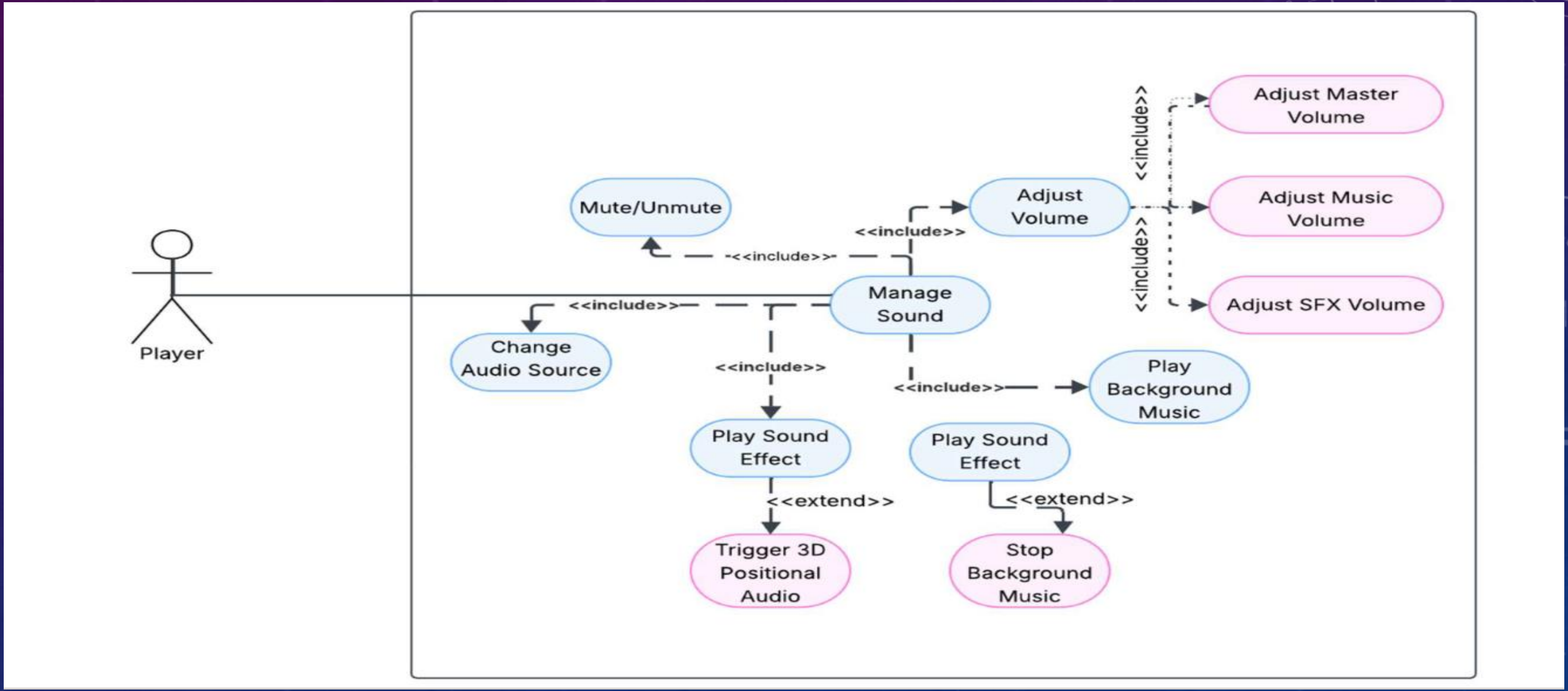
Manage Sound

- Adjusting volume (Master, Music, SFX)
- Mute / Unmute
- Change audio source
- Play sound effects (with optional 3D positional audio)
- Play / Stop background music

Priority: Medium (nice-to-have feature)
Complexity: Medium (multiple audio controls and dependencies)

TL4

URVASHI(USE CASE DIAGRAM)



USE CASE SCENARIO (Urvashi)

Name: Manage Sound
Summary:
The player controls audio within the game, including adjusting volume, muting/unmuting, changing audio sources, playing sound effects, and playing/stopping background music.
Actors:
Player
Preconditions:
- The game is running.
- The sound system is initialized.

Basic sequence:
- The player initiates a sound-related action (e.g., adjust volume, mute/unmute, change audio source).
- The system processes the request through the Sound Manager.
- If the player adjusts volume, the system updates the master, music, or SFX volume accordingly.
- If the player chooses to play sound effects, the system plays the appropriate effect (optionally triggering 3D positional audio).
- If the player chooses to play background music, the system starts playback (which can later be stopped).

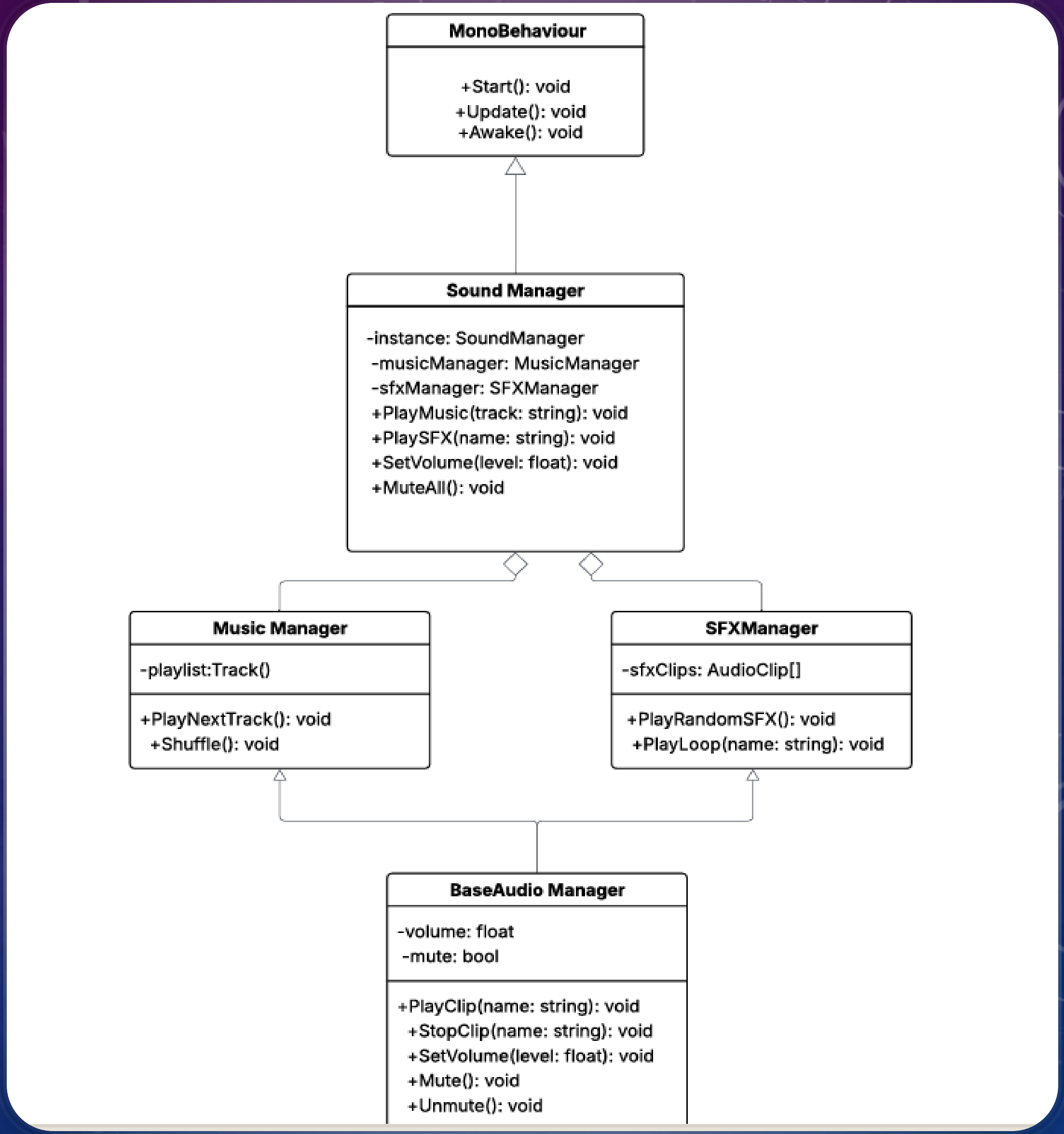
Exceptions:
- Volume adjustments exceed system limits → system clamps to minimum or maximum allowed values.
- Attempting to change audio source while one is unavailable → system ignores or provides default audio source.
- Attempting to play background music when no track is loaded → system does nothing or plays default track.

Post conditions:
- The requested audio action is applied (volume change, playback, mute, etc.).
- The player experiences updated sound settings or effects in-game.

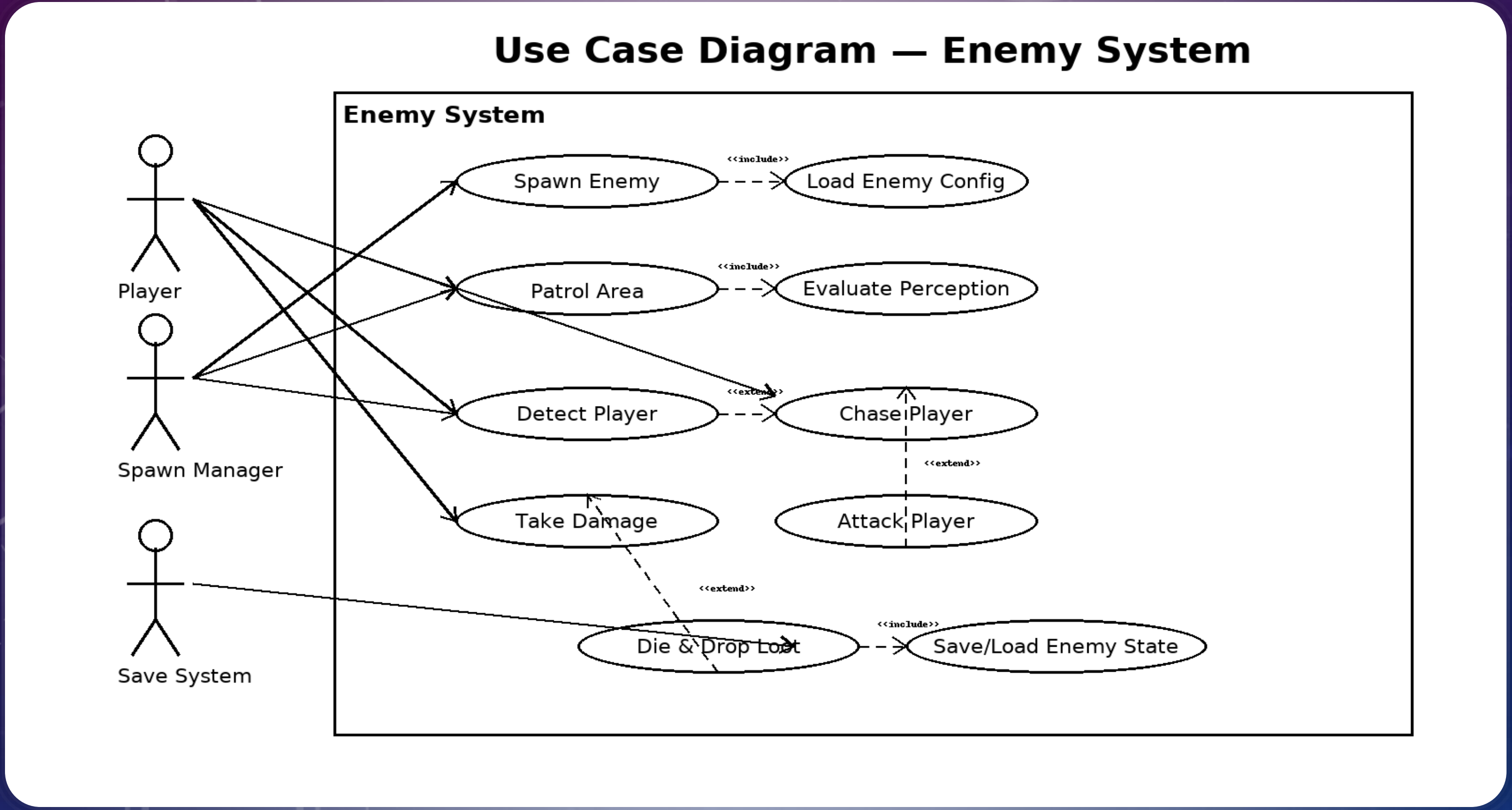
Priority: 3*
ID: AE03
*Priorities are: 1 = must have, 2 = essential, 3 = nice to have.

URVASHI CLASS DIAGRAM (SOUND)

TL4



TL –5 AJ (USECASE DIAGRAM OF ENEMIES)



TL5

AJ (ENEMIES OVERVIEW)

- **Show up & start:** Enemies **appear** and begin **walking a set route** *(patrol = loop between fixed points while watching for the player)*.
- **Look for you:** They keep **checking** with sight/hearing *(perception & line-of-sight)* to find the player.
- **Change mode:** If they spot you → **Patrol → Chase → Attack → (Flee or Die)**.
- **Chase smartly:** They **run toward you** using paths; if they **lose you** for a few seconds, they **return to patrol**.
- **Fair attacks:** Clear **wind-up** and short **hit window**; send **damage** to Combat and update the **HUD**.
- **When hit:** They can **stagger**; at **0 HP** they **play death**, **drop loot**, and we **mark them defeated** *(save flag)*.
- **Gets harder: Scaling** makes enemies tougher later *(more HP / speed / aggression)*.
- **Remember progress:** The game **saves** defeated enemies and **loot rolls**.
- **Run smoothly: Throttle checks**, **ignore far enemies**, and use simpler off-screen behavior for **performance**.
- **Priority: High** — drives challenge/tension and connects **Level, Combat, UI, Scaling, Save**.
  Complexity: Medium–High — live **AI state machine**, **pathfinding + perception**, lots of enemies, multi-system integration, plus **save/load**.

AJ (USECASE SCENARIO 1)

Name: Detect & Chase

Summary: Enemy notices the player and starts chasing until in attack range or the player is lost.
Actors: Player, Enemy System
Preconditions: Enemy is active and patrolling; game running.
Basic sequence:

- Enemy patrols its route.
- Perception check runs (vision/hearing/LOS). *(<<include>> Evaluate Perception)*
- If the player is detected, switch state to **Chase Player**. *(<<extend>> Detect Player)*
- Enemy moves toward player using pathfinding.
- If within attack distance, hand off to **Attack Player**.

Exceptions:

- Step 3: Player breaks line-of-sight for N seconds → **Handle Lost Sight** and return to **Patrol Area**.
- Step 4: Path blocked → choose alternate path; if none, return to **Patrol Area**.

Post conditions: Enemy is either attacking or back on patrol.
Priority: 1*
ID: UC-EN-01

AJ (USECASE SCENARIO 2)

- Name: Attack Player
- **Summary:** Enemy performs a readable attack when close enough to the player.
  Actors: Player, Enemy System
  Preconditions: Enemy is in **Chase Player**; player within attack range.
  Basic sequence:
- Switch to **Attack Player**. *(<<extend>> from Chase Player)*
- Telegraph the attack (wind-up).
- Execute attack; hitbox active briefly.
- If the hit connects, apply damage and show HUD feedback. *(calls Combat)*
- Enter short recovery, then decide to **repeat** attack or return to **Chase Player**.
  Exceptions:
- Step 3: Player dodges/blocks/parries → enemy **stagger/recover** and return to **Chase Player**.
- Step 2–3: Player moves out of range mid wind-up → cancel and return to **Chase Player**.
  Post conditions: Attack finishes; enemy either chases again or resets.
  Priority: 1*
  ID: UC-EN-02

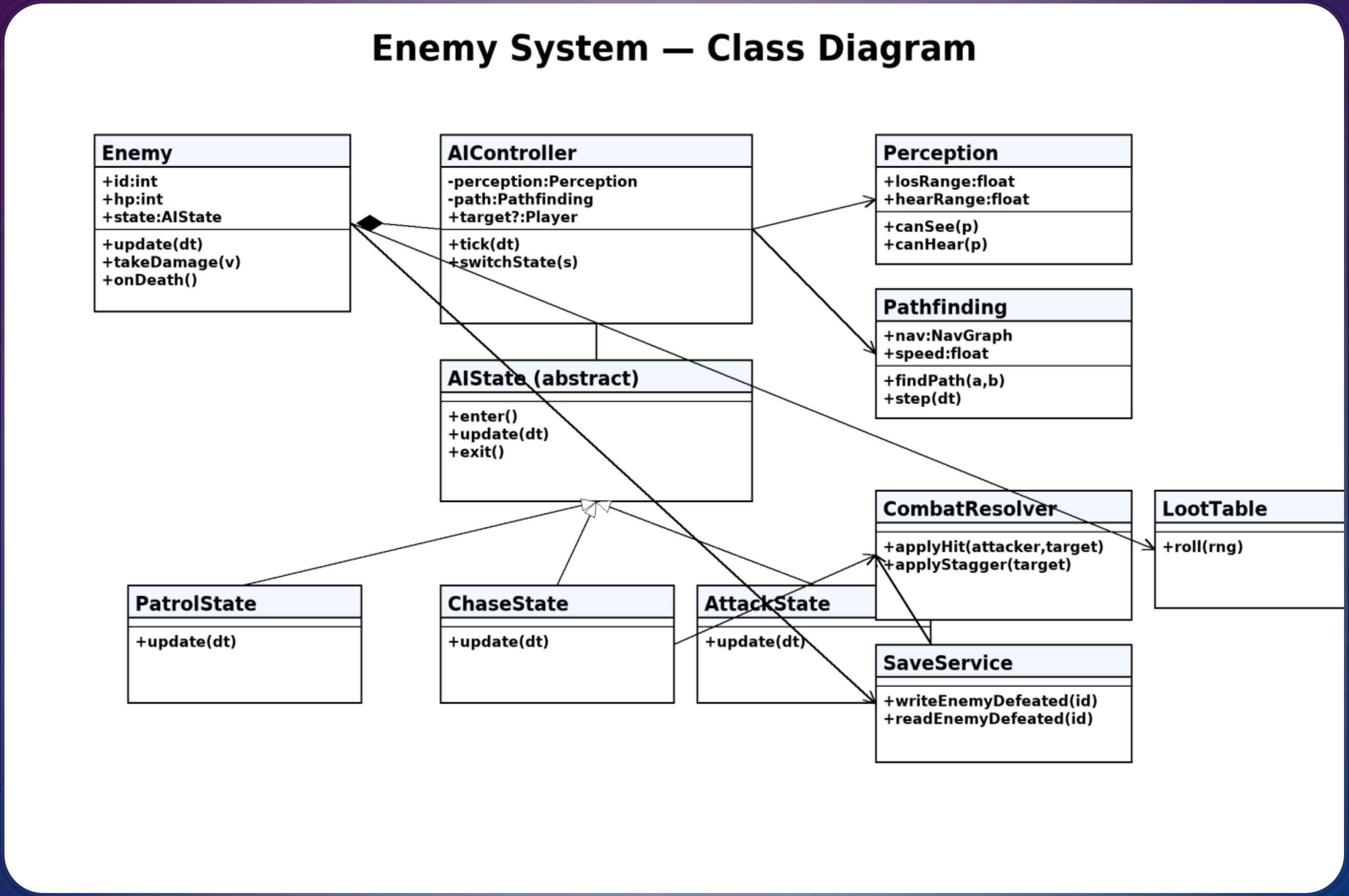
TL5

AJ (USECASE SCENARIO 3)

- Name: Die & Drop Loot
- **Summary:** When HP reaches 0, the enemy dies, drops loot, and the defeat is saved.
  Actors: Player, Enemy System, Save System
  Preconditions: Enemy HP reduces to 0 (via **Take Damage**).
  Basic sequence:
- Play death animation; disable AI/collisions.
- **Die & Drop Loot** spawns item(s) using loot table. *(notifies HUD/Inventory)*
- **Save/Load Enemy State** records `enemy_defeated = true`. *(<<include>> Save/Load Enemy State)*
- Despawn the body after a short delay.
  Exceptions:
- Step 2: Inventory full → keep item as ground pickup.
- Step 3: Save fails → mark retry flag for next checkpoint.
  Post conditions: Enemy removed; loot available; defeat state saved (won't respawn on reload).
  Priority: 2
  ID: UC-EN-03

TL5

AJ (CLASS DIAGRAM)



TL6

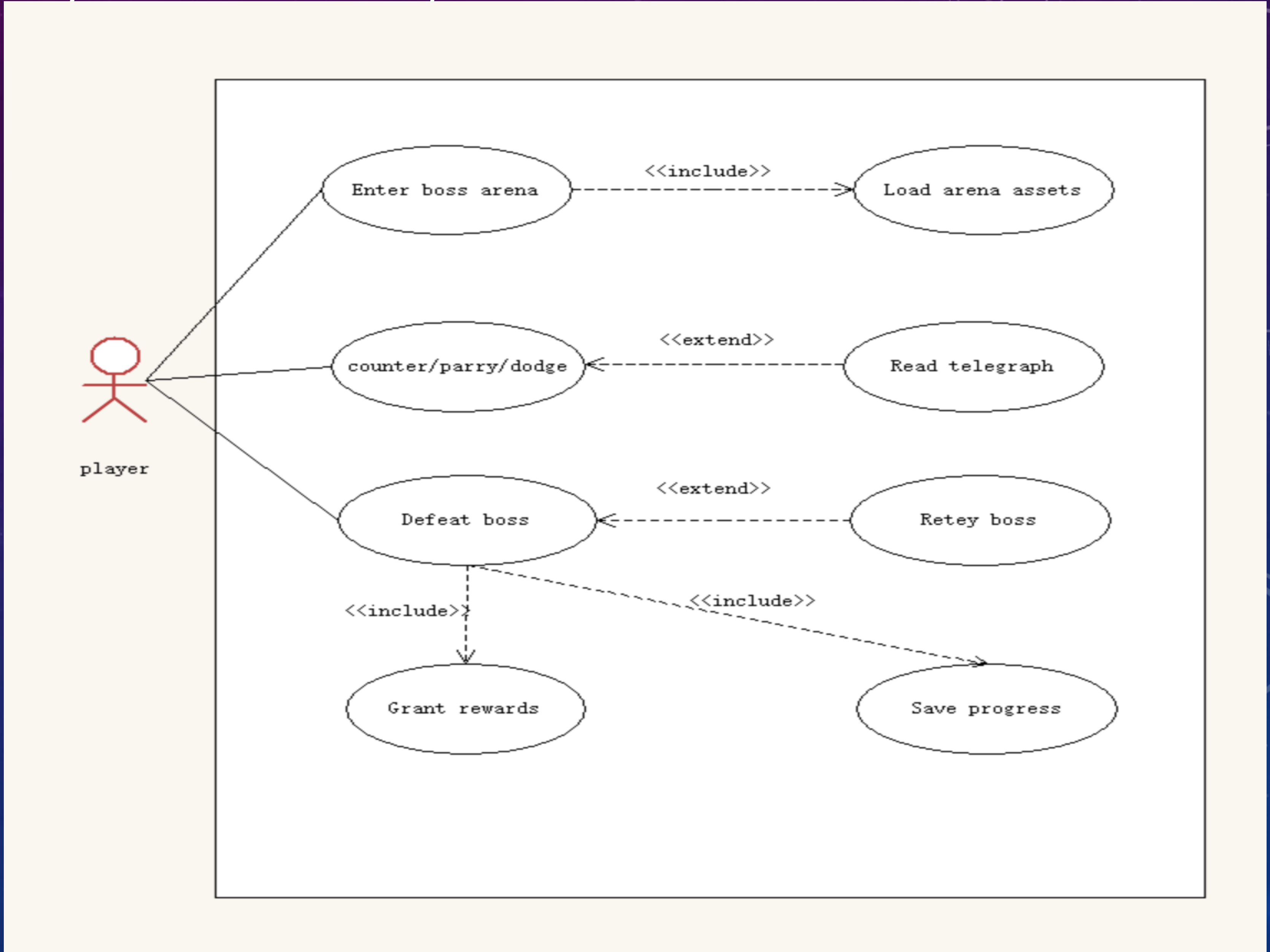
QIWEI (BOSS FIGHT/DIFFICULTY OVERVIEW)

- Enter boss arena and lock encounter space
- Load arena/boss assets and initialize phase 1
- Telegraph attacks; support counter/dodge/parry timing windows
- Handle phase transitions (P1→P2→P3)
- Integrate HUD elements (boss HP bar, objectives, hit/crit feedback)
- Manage defeat/victory flow: Retry Boss, Grant Rewards, Save Progress
- Ensure performance + readability (Visual Effects/*Sound Effects* cues, no hitching)

- **Priority:** High (core combat ; control player progression and rewards)
 Complexity: High (AI state machine, multi-phase tuning, checkpoint/retry integration, robust difficulty scaling and polish across VFX/SFX/HUD)

TL6

QIWEI(BOSS FIGHT/DIFFICULTY) - USE CASE DIAGRAM



QIWEI (USE CASE SENARIO)

- **Name:** Defeat Boss

o **Summary:** A player engages in combat with a boss and successfully defeats it to progress in the game.

o **Actors:** Player

o **Preconditions:** The player has entered the boss arena, and the fight with the boss has begun.

o **Basic sequence:**

- **Step 1:** The player attacks the boss and uses mechanics such as dodging, blocking, or parrying to avoid its attacks.
- **Step 2:** The player successfully reduces the boss's health to zero.
- **Step 3:** The boss is defeated, triggering its defeat animation and ending the combat phase.
- **Step 4:** The system grants rewards (e.g., experience, items) to the player and saves the game progress.

o **Exceptions:**

- **Step 1:** During combat, if the player's health is reduced to zero by the boss's attacks, the player is defeated. The system will present a "Retry" or "Load last save" option.

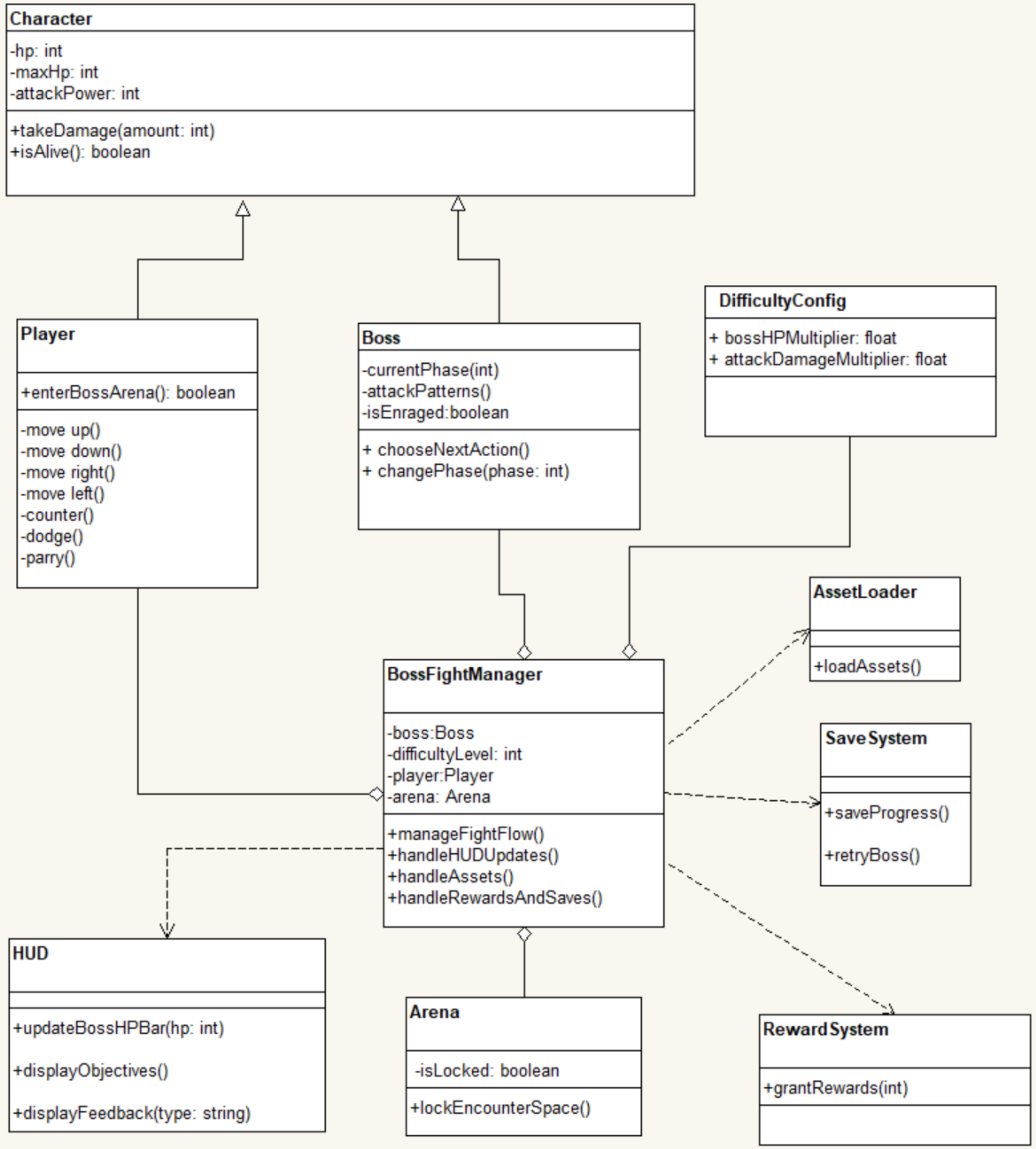
o **Post conditions:** The boss is defeated and is no longer a threat. The player has received rewards, progress is saved, and the path to the next area or story event is unlocked.

o **Priority:** 1*

- **The priorities are 1 = must have, 2 = essential, 3 = nice to have.*

TL6

CLASS DIAGRAM （BOSS FIGHT/DIFFICULTY）



Character
-hp: int
-maxHp: int
-attackPower: int
+takeDamage(amount: int)
+isAlive(): boolean
Player
+enterBossArena(): boolean
-move up()
-move down()
-move right()
-move left()
-counter()
-dodge()
-parry()
Boss
-currentPhase(int)
-attackPatterns()
-isEnraged:boolean
+ chooseNextAction()
+ changePhase(phase: int)
DifficultyConfig
+ bossHPMultiplier: float
+ attackDamageMultiplier: float
AssetLoader
+loadAssets()
BossFightManager
-boss:Boss
-difficultyLevel: int
-player:Player
-arena: Arena
+manageFightFlow()
+handleHUDUpdates()
+handleAssets()
+handleRewardsAndSaves()
Save System
+saveProgress()
+retryBoss()
HUD
+updateBossHPBar(hp: int)
+displayObjectives()
+displayFeedback(type: string)
Arena
-isLocked: boolean
+lockEncounterSpace()
Reward System
+grantRewards(int)