Alt(Alternative) Pokémon

By: Sayed Kibria

(sayed.kibria45@login.cuny.edu)

Supervisor: Gulshan Meem

(Gulshanmeem60@gmail.com)

Goal

This project is a 2D top-down RPG inspired by the classic Pokémon franchise but with a twist: instead of turn-based battles, players engage in real-time combat, actively controlling their Pokémon during fights. The game maintains core features like exploration, wild Pokémon encounters, and trainer battles but changes the experience through action-based mechanics. This project explores real-time systems in monster-taming game, focusing on responsive controls, and AI enemy behaviors.

Tool

Unity

C# (for scripting)

Visual studio

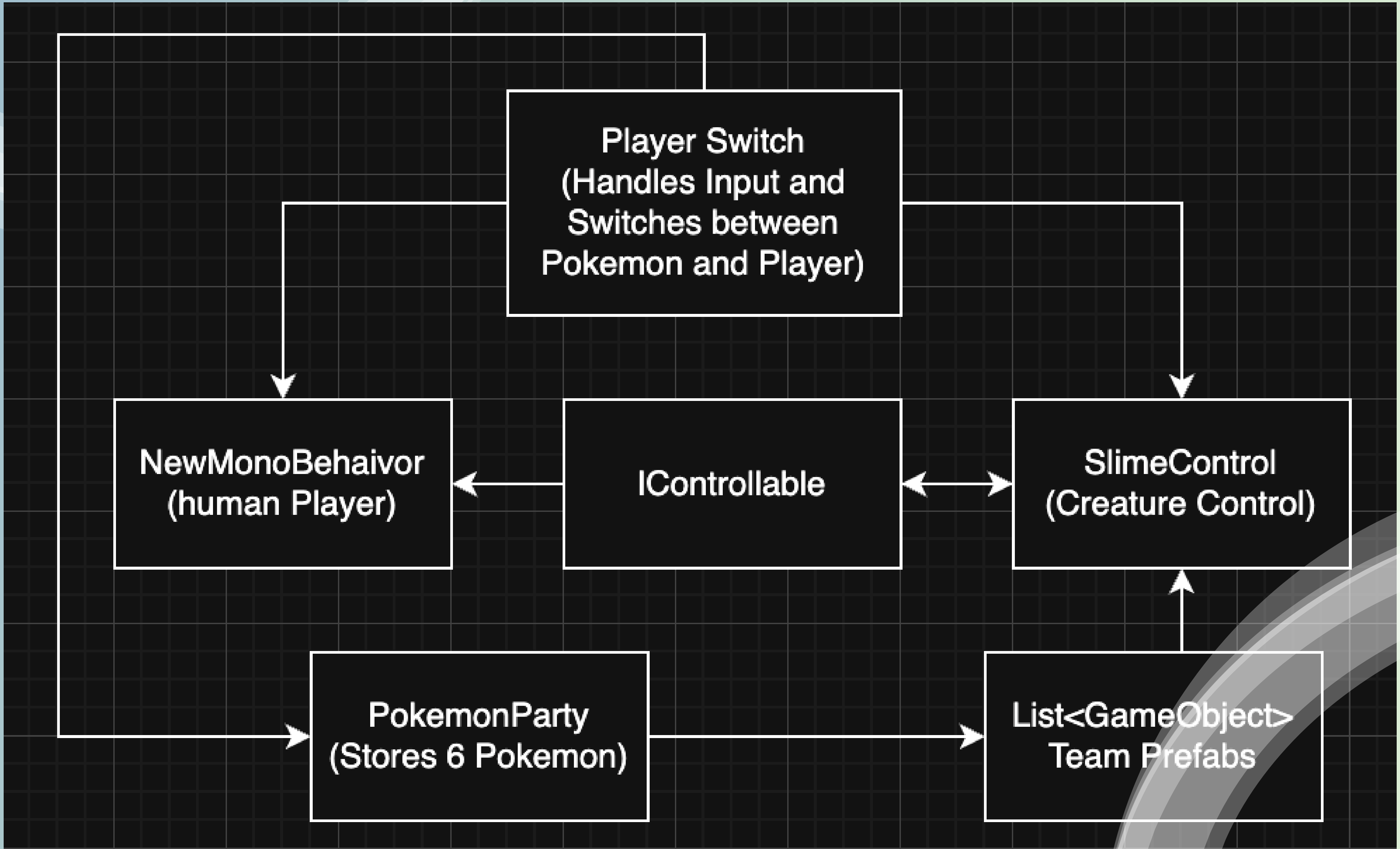
Pixilart (to make pixel art)

Data Source

Pokemondb Pokedex (for stats and what attack Pokémon can use)

Spite and Assets (from online open-sources)

System Overview



Player Switch (Handles Input and Switches between Pokemon and Player)

NewMonoBehaivor (human Player)

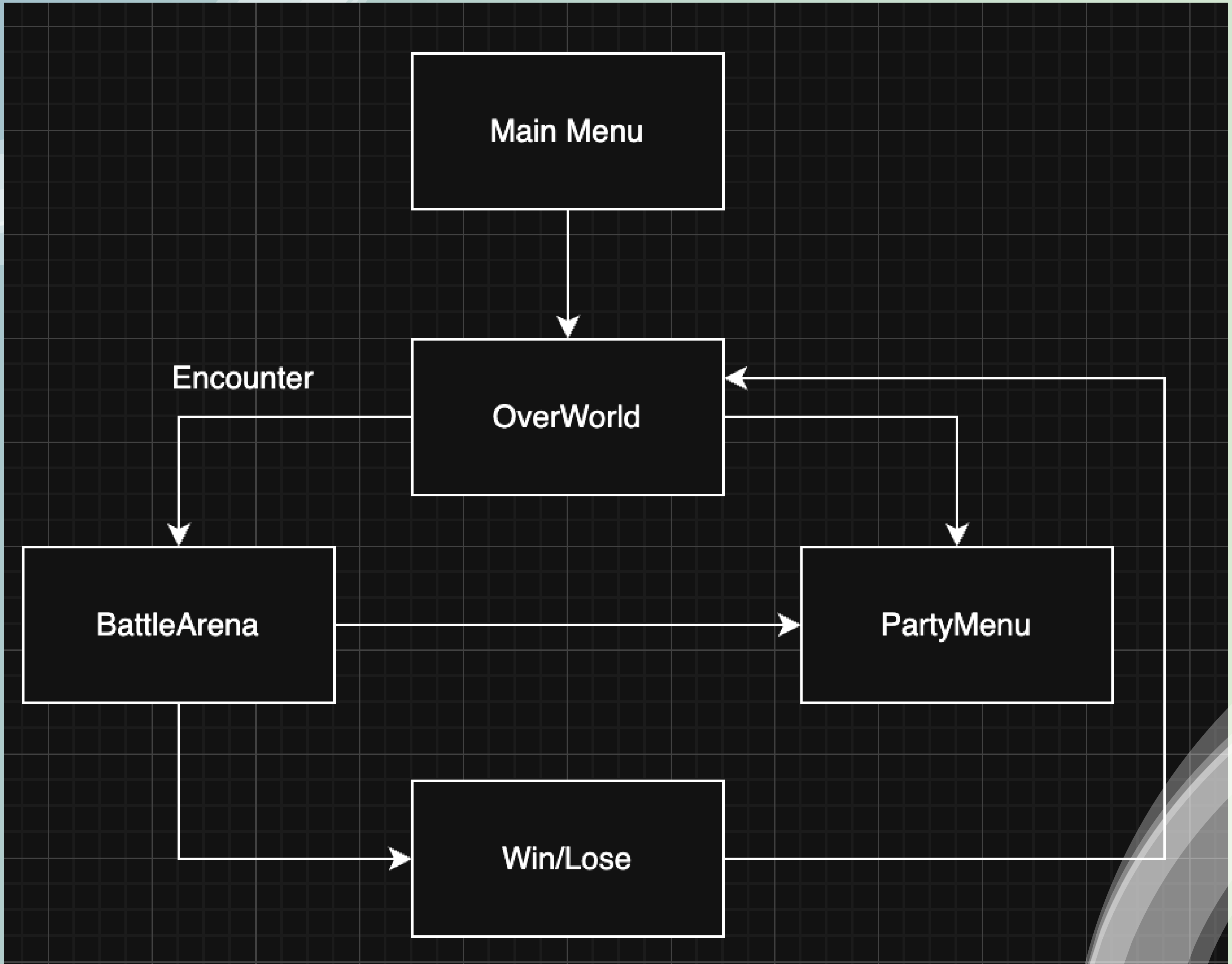
IControllable

SlimeControl (Creature Control)

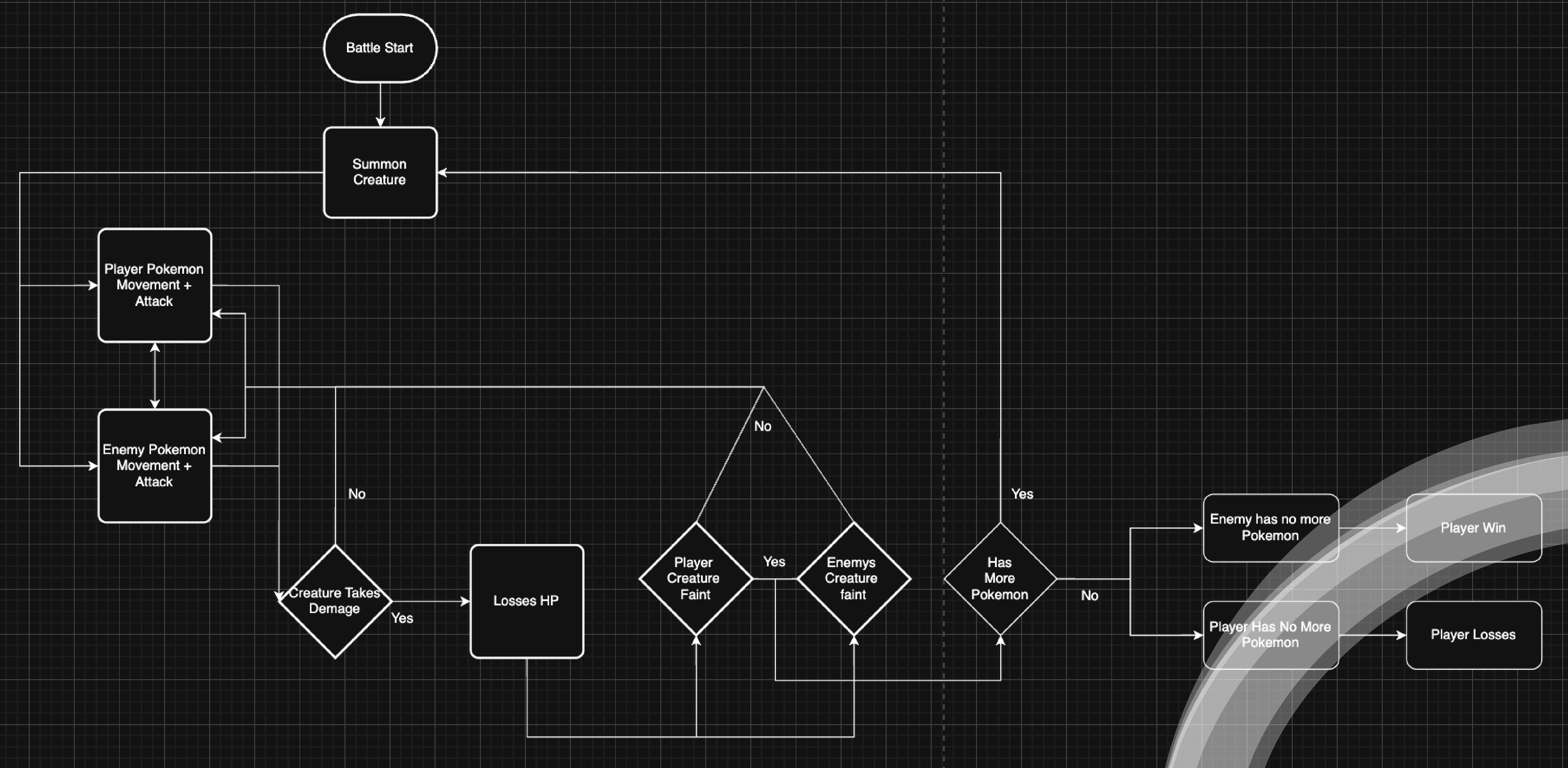
PokemonParty (Stores 6 Pokemon)

List<GameObject> Team Prefabs

Scene Transition



Battle Flow


Battle Start
Summon Creature
Player Pokemon Movement + Attack
Enemy Pokemon Movement + Attack
No
Creature Takes Demage
Yes
Losses HP
No
Player Creature Faint
Yes
Enemys Creature faint
Yes
Has More Pokemon
No
Enemy has no more Pokemon
Player Win
Player Has No More Pokemon
Player Losses

Use Case 1: Trainer Battle

Input: Player approaches an NPC Trainer on the map.

Process: Dialogue starts → Scene transitions to battle → Enemy Trainer sends out Pokémon → AI uses logic.

Output: Player must defeat all of the trainer's Pokémon; receives rewards or progression if successful.

Use Case 2: Summon Pokémon from Party

Input: Player presses number key (1–6) during battle.

Process: PlayerSwitcher gets Pokémon from the party → Player character is disabled → Pokémon is spawned → Control is given to Pokémon.

Output: Player controls the summoned Pokémon in real-time.

Use Case 3: Real-Time Combat Attack

Input: Player presses an attack key while controlling a Pokémon.

Process: Attack animation plays → System checks if enemy is in range → Enemy takes damage if hit.

Output: Enemy loses HP; possibly defeated if HP reaches 0.

Tentative Schedule

Basic Battle Logic: 15 hours

attack. Damage. Health. Hit feedback.

Battle Arena: 15 hours

Arena map. Transition to battle arena when encountering npc. Arena boundaries

Enemy AI: 15 hours

Chasing. Attacking. Avoiding. Hit feedback

Win/Loss conditioning + fainting: 15 hours

Show win/lose states. Handle KO events properly

Tentative Schedule

Ranged vs Melee Attacks + Stats System: 15 hours

Implement ranged and melee attacks. Add stat value(attack, defense, special attack, special defense).

Type System + Damage Calculation: 15 hours

Add Pokémon types. Damage formula

Dodge/Dash Mechanics + Combat Fluidity: 15 hours

Add dash move with cooldown. Temporary invincibility. Balance combat movement speed.

Creature Switching & Party Expansion: 15 hours

Mid-combat swap and cooldown. Party UI. Store current state of pokemon

Tentative Schedule

Final Implementation: 15 hours

Add basic main menu. Add few more Pokémon. Final demo preparation.