### **Pokemon Project Document**

#### **Overview-** This document describes the implementation of the Pokemon Trading Card Game in java. The game follows the Pokémon Trading Card Game (TCG) rules, allowing players to take turns performing various actions such as playing Pokémon, attaching Energy, using Trainer cards, and attacking. The game state is updated accordingly based on the user’s inputs.

#### **Key Features**

#### **Player Turn Management**

#### Players draw a card at the start of the turn.

#### Energy and supporter card resets ensure compliance with game rules.

#### Players can choose from multiple actions: playing Pokémon, attaching Energy, playing Trainer cards, attacking, or ending their turn.

#### **Playing Pokémon**

#### Players can place a Pokémon in the active spot or on the bench.

#### The playPokemon method handles this, displaying available Pokémon from the player's hand and updating the game state accordingly.

#### **Attaching Energy**

#### Players may attach one Energy card per turn.

#### The attachEnergy method ensures that only one Energy is attached per turn and allows Energy to be attached to either the active Pokémon or a benched Pokémon.

#### **Playing Trainer Cards**

#### The playTrainerCard method allows the player to play Trainer cards with special effects.

#### Specific Trainer cards, such as "Bill," "Giant Cape," and "Potion," have unique effects implemented in the code.

#### Supporter cards like "Professor Oak" and "Professor's Research" enforce the one-per-turn rule.

#### **Attacking**

#### The attack method enables players to choose from two possible attacks, provided they have enough Energy attached.

#### Damage is calculated and subtracted from the opponent's Pokémon HP.

#### If an opponent's Pokémon is knocked out, the game state is updated accordingly, including prize card drawing and replacing defeated Pokémon.

#### **Game State Display**

#### The displayGameState method provides an overview of the current game state, including active Pokémon and the bench pokemon of the opponent AI and player.

#### **Code Logic & Flow**

#### The game begins with each player drawing a hand of cards.

#### The player and Opponent AI take turns

#### Actions are processed using switch-case logic in playerTurn.

#### The game state is continuously updated after each action.

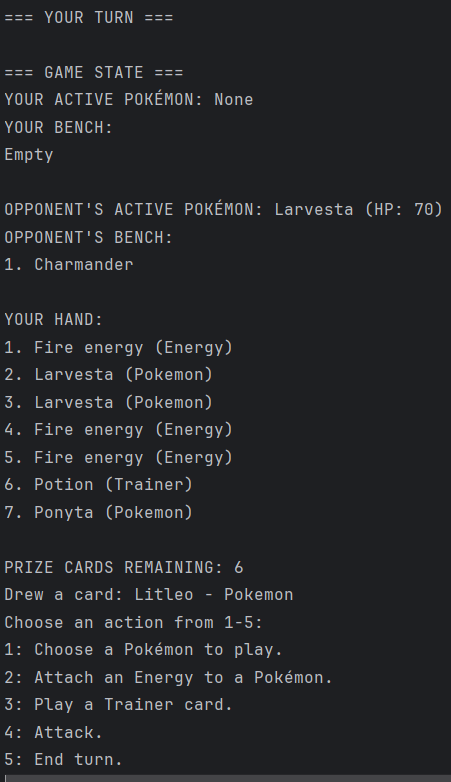
#### Attacking checks for Energy requirements and calculates damage.

#### Trainer cards apply specific effects to the game state.

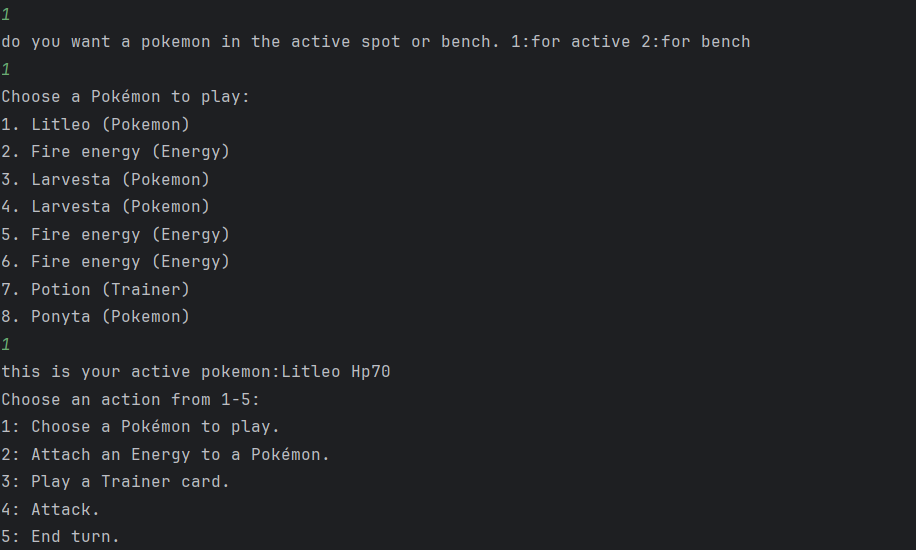
#### The game progresses until a victory condition is met (No bench pokemon to replace the knocked out active pokemon, or all prize cards have been drawn from the prize pile).

**Results**

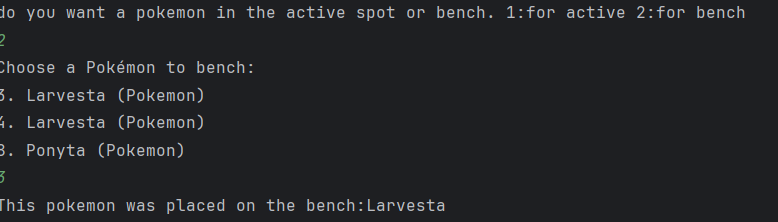
* **Starting state:**

****

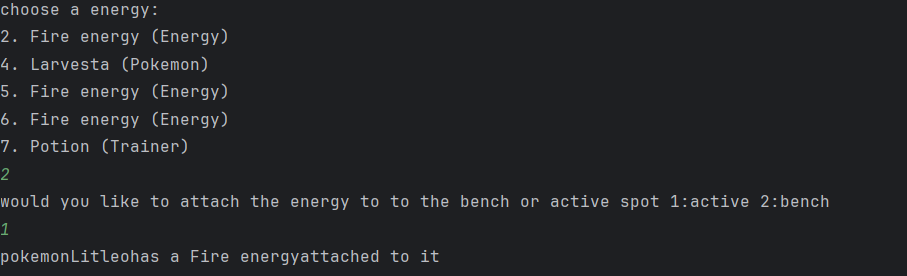
* **Playing a Pokemon:**

****

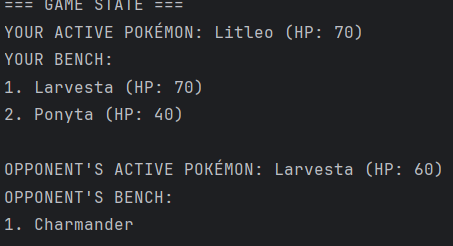
* **Benching a Pokemon:**

****

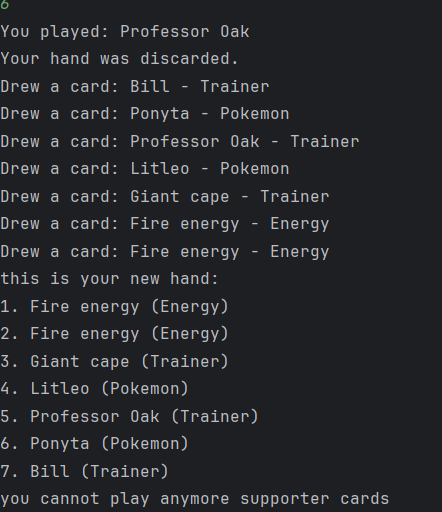
* **Attaching an Energy:**

****

* **Displaying Game State:**

****

* **Playing a Trainer Card:**

****

#### **Potential Enhancements**

#### Adding graphical UI for a more interactive experience.

#### Introducing more Trainer cards with complex effects.

#### Enhancing the turn logic to include retreating Pokémon and switching strategies.

#### 

#### 