# **Pokémon Card Game - Code Documentation**

## **1. Main.java (Entry Point)**

* The starting point of the game. It initializes and launches the game menu.
* **Methods:**
  + public static void main(String[] args): Starts the game by opening MenuGUI.

## **2. MenuGUI.java (Main Menu)**

* Displays the main menu where players can start the game, view rules, or quit.
* **Methods:**
  + MenuGUI(): Initializes the menu interface, loads a background image, and creates buttons.
  + **Button:**
    - Start Game: Launches GameGUI.
    - Rules: Displays game rules in a pop-up.
    - Quit: Closes the application.

## **3. GameGUI.java (Main Game Interface)**

* The main game screen where players interact and play their turns.
* **Methods:**
  + startGame(): Initializes the game, sets up the board, and manages turn-based actions.
  + initiateCoinFlip(): Manages the heads/tails coin flip.
  + processCoinFlip(): Determines which player goes first.
  + drawCard(): Allows a player to draw a card from their deck.
  + playPokemon(): Lets a player move a Pokémon to the bench or battlefield.
  + attachEnergyToPokemon(): Assigns an energy card to a Pokémon.
  + attack(): Handles Pokémon attack actions.
  + endTurn(): Switches the turn to the next player.
  + updateHandAndBenchDisplay(): Updates the UI for a player's hand and bench Pokémon.

## **4. Player.java (Manages Player Actions)**

* Handles the actions and properties of each player.
* **Attributes:**
  + name: The player's name.
  + hand: Cards in hand.
  + bench: Pokémon on the bench.
  + active: The active Pokémon.
  + deck: The player's deck.
  + pool: Prize cards.
* **Methods:**
  + drawCard(Deck deck): Draws a card from the deck.
  + initializePool(): Sets up the prize pool with three cards.
  + attackOpponent(Player opponent): Handles attacking an opponent's active Pokémon.
  + playPokemonToBenchUI(): Moves a Pokémon from hand to bench.
  + addPokemonToField(): Moves a Pokémon from the bench to the battlefield.
  + claimPrizeCard(): Claims a prize card when an opponent's Pokémon faints.

## **5. Deck.java (Manages the Player's Deck)**

* Contains the player's shuffled cards and manages drawing.
* **Methods:**
  + initializeDeck(): Fills the deck with Pokémon, Trainer, and Energy cards.
  + shuffle(): Randomly shuffles the deck.
  + drawCard(): Draws a card from the top of the deck.
  + initializePool(): Assigns a prize pool card.

## **6. CoinFlip.java (Handles Coin Toss for First Turn)**

* Simulates a coin flip to decide the first player.
* **Method:**
  + flip(): Returns true for heads and false for tails.

## **7. PokemonCard.java (Abstract Class for Pokémon)**

* Base class for all Pokémon cards.
* **Attributes:**
  + hp: Pokémon's health points.
  + energy: Energy required for attacks.
  + attack1, attack2: Two possible attacks.
  + statusEffect: Conditions like Paralyze, Burn, Sleep.
* **Methods:**
  + takeDamage(int damage): Reduces HP when attacked.
  + applyStatusEffect(): Applies effects like paralysis or sleep.
  + attack(PokemonCard opponent): Inflicts damage.
  + addEnergy(EnergyCard energy): Attaches energy.
  + abstract useAttack(PokemonCard opponent): Must be implemented by subclasses.

## **8. Attack.java (Defines Pokémon Attacks)**

* Represents an attack move.
* **Methods:**
  + getName(): Returns attack name.
  + getDamage(): Returns attack damage.

## **9. EnergyCard.java (Assigns Energy to Pokémon)**

* Represents energy cards.
* **Methods:**
  + getEnergyBoost(): Returns the energy amount provided.
  + getHpBoost(): If applicable, provides HP boost.

## **10. TrainerCard.java (Special Effect Cards)**

* Provides effects like healing or drawing cards.
* **Methods:**
  + applyEffect(Player player): Applies effects based on the card.
  + Potion: Heals 20 HP.
  + Full Heal: Removes status effects.
  + Professor Oak: Draws 3 new cards.

**11. PrizeCards.java (Handles Prize Pool Mechanics)**

* **Purpose:** Manages prize cards.
* **Methods:**
  + claimPrizeCard(): Allows the player to claim a card after a knockout.

## **12. MusicPlayer.java (Handles Background Music)**

* **Purpose:** Plays and stops background music.
* **Methods:**
  + playMusic(String filePath): Plays a background music file.
  + stopMusic(): Stop the currently playing music.

## **13. Pokémon Subclasses**

These classes inherit from PokemonCard and define specific Pokémon:

* **Pikachu**
* **Charmander**
* **MewTwo**
* **Golem**

Each Pokémon has:

* Custom HP
* Unique attacks
* Special attributes

public class Pikachu extends PokemonCard {

public Pikachu() {

super("Pikachu", 60, "Electric", 20, "pikachu.png", new Attack("Thunder Shock", 20), new Attack("Volt Tackle", 40));

}

}

* Thunder Shock deals 20 damage.
* Volt Tackle deals 40 damage.

## **14. Battle.java (Handles Fights)**

* **Purpose:** Manages the logic of Pokémon battles.
* **Methods:**
  + executeAttack(PokemonCard attacker, PokemonCard defender): Applies attack damage.
  + applyStatusEffect(): Manages paralyze, burn, sleep effects.

## **15. Potion50.java (Trainer Card for Healing)**

* **Purpose:** Specialized healing card.
* **Method:**
  + applyEffect(Player player): Heals 50 HP when used.

**Summary:**

* **Game Flow:** Main, MenuGUI, GameGUI
* **Game Mechanics:** Player, Deck, CoinFlip
* **Card Types:** PokemonCard, EnergyCard, TrainerCard, PrizeCards
* **Battle System:** Attack, Battle
* **Pokémon:** Pikachu, Charmander, MewTwo, Golem
* **Utility:** MusicPlayer

NOTE: Asset directory has the image and sound files that has been used in this game.