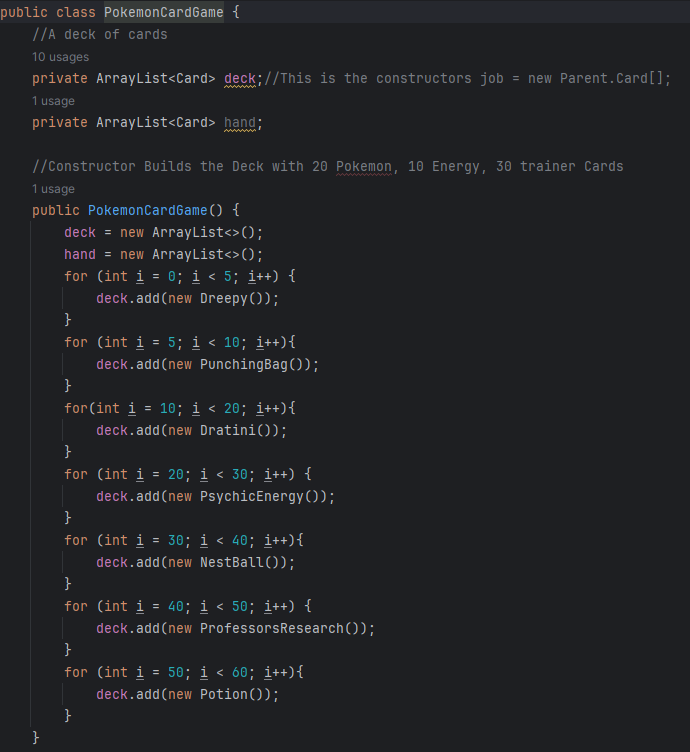
Dustin McDonnell

**Documentation for Pokémon Project and Stats Library**

**Pokémon Card Game Class**



Initializes the deck with 20 Pokémon, 10 Energy, and 30 trainer Cards.

The three Pokémon used for the required three for this project are Dreepy, Punching Bag, and Dratini. Energy did not matter outside of it existing, and the three trainer cards were Next Ball, Professor’s Research and Potion.

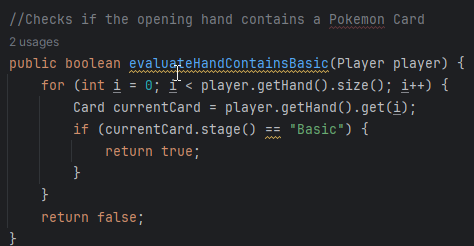
A screenshot of a computer program

Description automatically generated

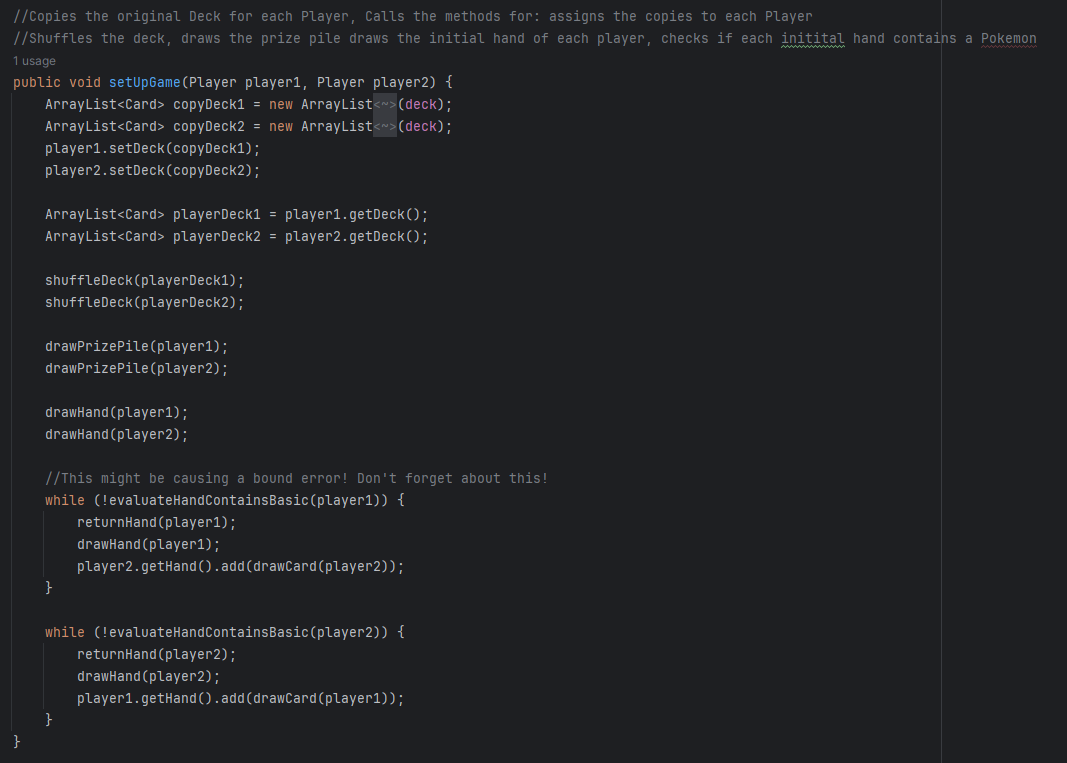
Four methods for drawing cards, drawing the complete hand, the prize pile for the Player, and returning a hand to the deck.

**evaluateHandContainsBasic(Player player)**

The following method checks for a Basic, stage 1, Pokémon card instead of just any Pokémon card but I did not add Pokémon other than stage 1 so it was a little bit of wasted effort. Possibly useful for the future.



**setUpGame(Player player1, Player player2)**



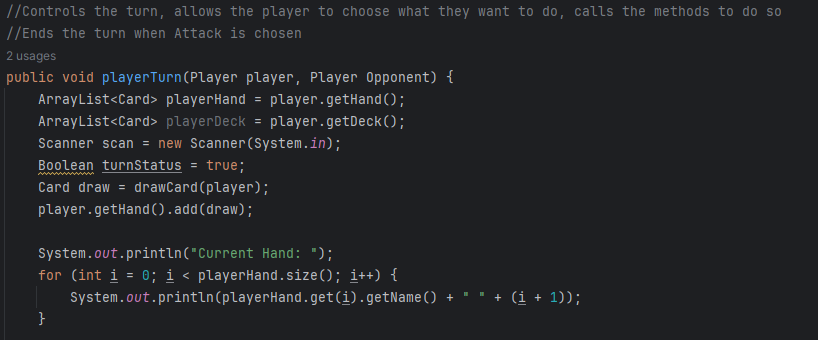
Uses helper methods to copy the initialized starting deck for both players. Shuffles the deck then draws the prize pile and starting hand and evaluates if the hand contains a basic Pokémon. If not, the hand is shuffled and redrawn until it contains one with the opposing player getting an additional card in their starting hand. Shown in the picture below with a starting hand of eight cards.

. A screenshot of a computer program

Description automatically generated

**playerTurn(Player player, Player Opponent)**

Controls what the player can do during their turn. Calls various helper methods depending on what the player wishes to do, input controlled with a Scanner class. The help methods will be shown in screenshots below.



**help()**

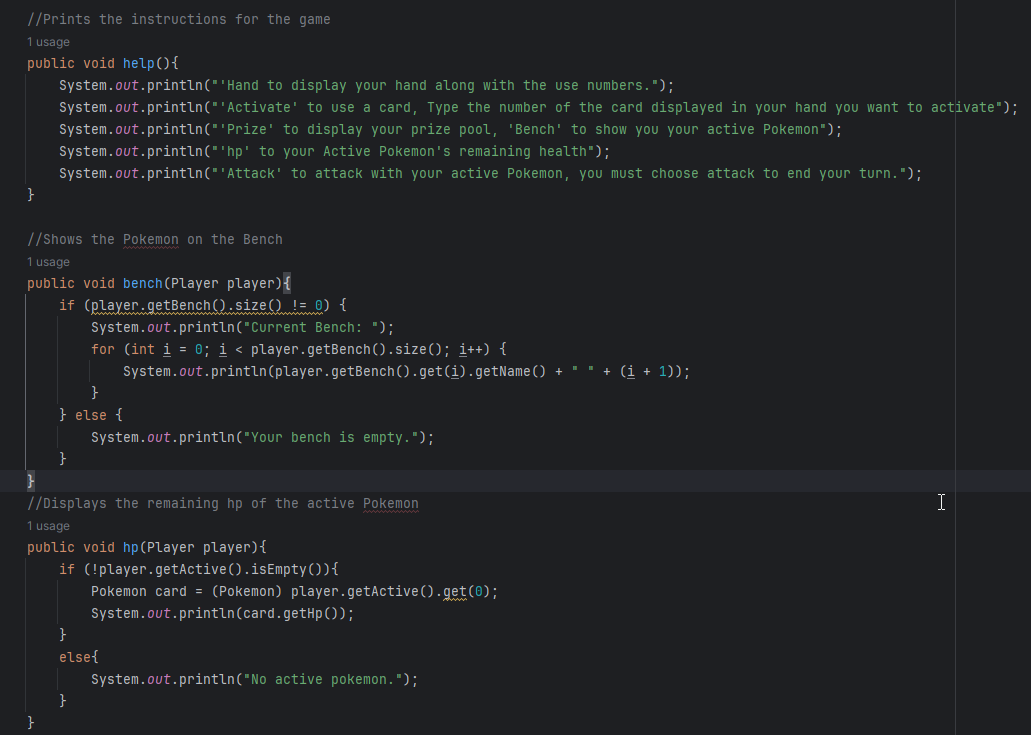
Displays the options available to the player.

**bench()**

Displays the Pokémon set on the bench.

**hp()**

Displays the remaining hp of the active Pokémon.



**active()**

Displays the Active Pokémon

A screen shot of a computer code

Description automatically generated

**hand()**

Displays the cards in the active player’s hand.

**prize()**

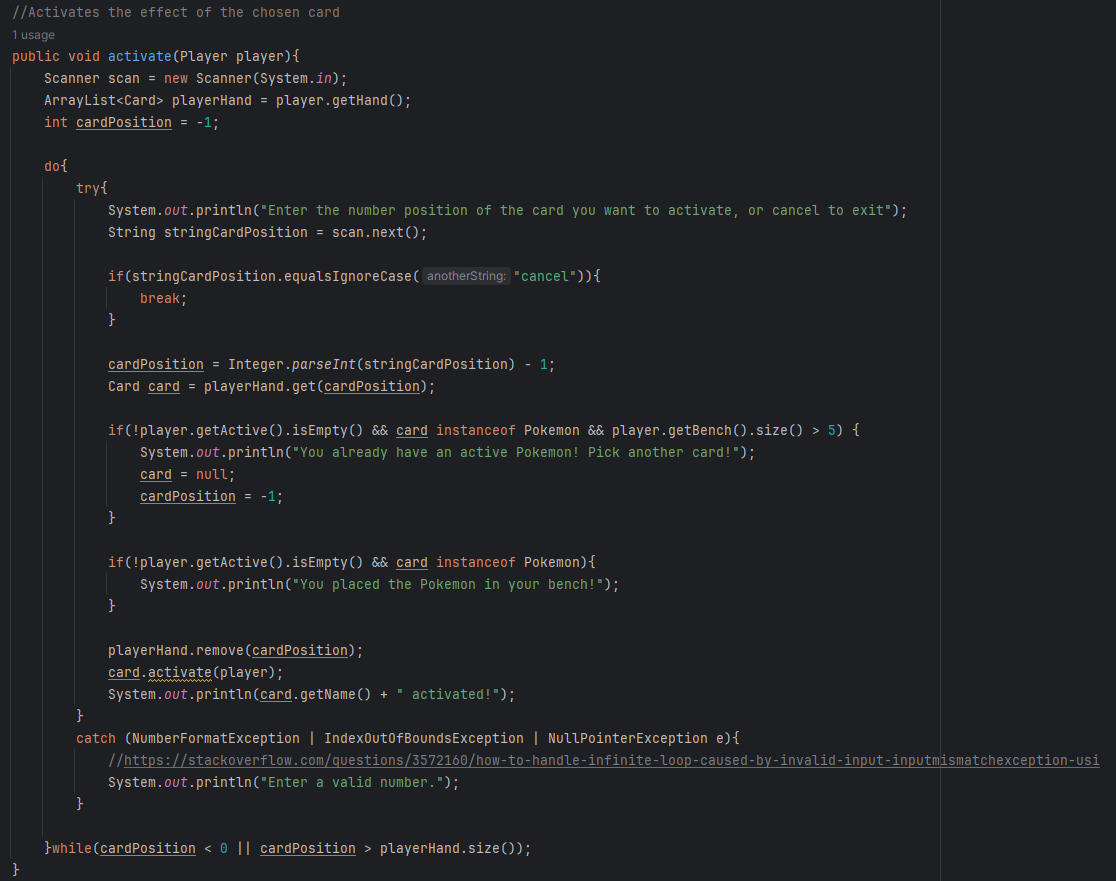
Displays the cards in the active player's prize pool.

A computer screen shot of a program code

Description automatically generated

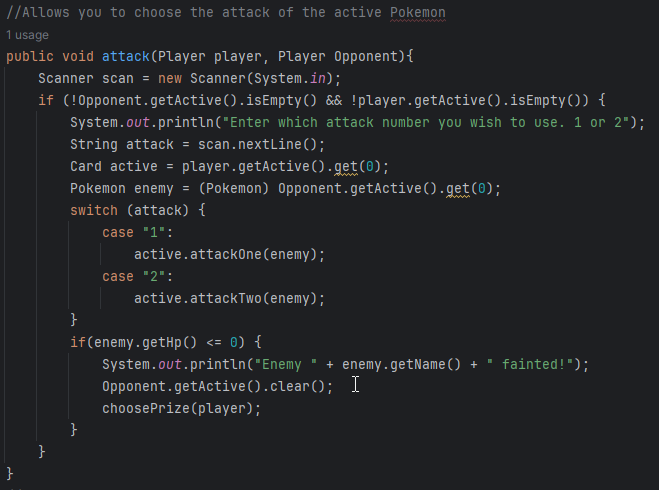
**activate()**

Calls the activate method that all Cards possess, has different outputs depending on number of Pokémon in bench, Active Pokémon, and multiple catch exceptions for typos in scanner.



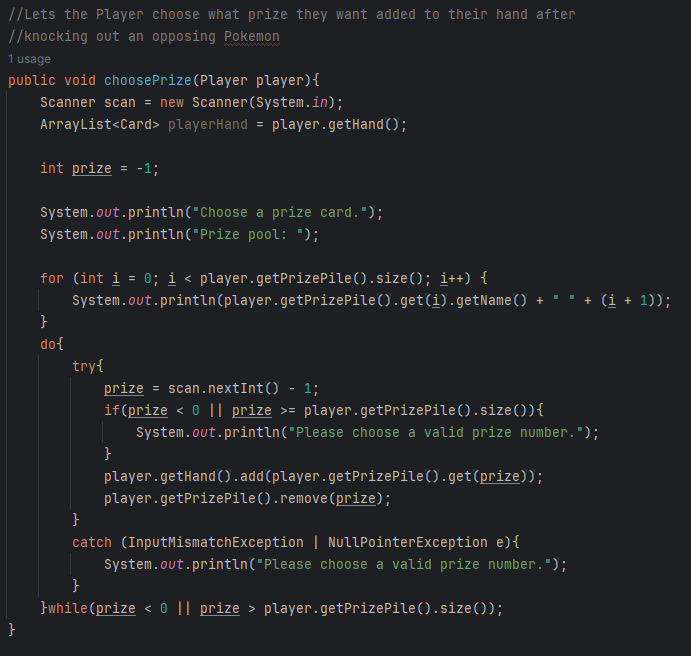
**attack()**

Allows the active player to choose what attack the Active Pokémon will use. Also ends the turn.



**choosePrize()**

If the attacking Pokémon knocks out the opponent’s active Pokémon this method is called and allows the active player to choose a card from their prize pool to add to their hand.

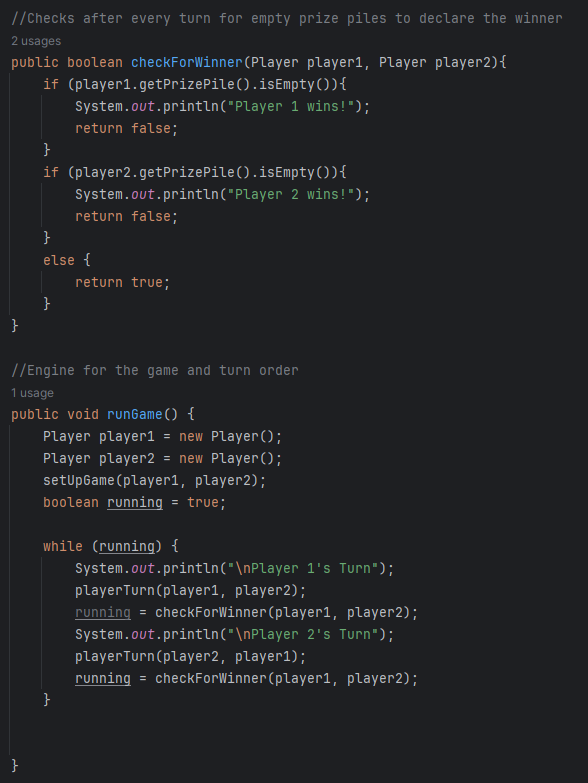


**checkForWinner()**

Checks both players prize pools after every turn. Ends the game if a prize pool is empty and declares the winner.

**runGame()**

Login for the turn order, controls which player is currently active and calls the checkForWinner method().



**Attack Sequence Class**

**damageCalculation()**

All attacks are run through the damage calculation function to determine any additional/reduced damage from weaknesses and resistances.

**moveBark()**

Attack names are used in conjunction with a randomly chosen phrase for flavor. Really makes you feel like a Pokémon trainer.

A screen shot of a computer program

Description automatically generated

**flip()**

Chooses a random number 0-1 to simulate a coin flip then prints out flavor. Used for Pokémon attacks requiring a coin flip.

A computer screen shot of a code

Description automatically generated

**Dreepy Class**

All the Pokémon have similar attributes to Dreepy except for Punching Bag. Type, Resistance, Weakness, Type, and hp are all declared. In addition, the same activation method that all the Pokémon have is here. Dreepy also has an attack that utilizes the flip() mentioned above.

A screen shot of a computer program

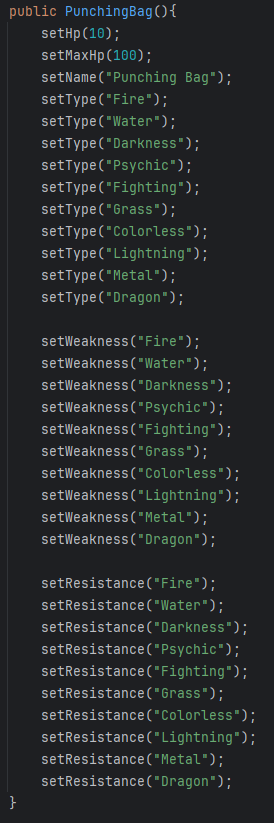
Description automatically generated

A computer screen shot of a program code

Description automatically generated

**Punching Bag Class**

Punching Bag is a Pokémon that was assigned every type, weakness, and resistance to allow for easy testing of any attack. There may have been an easier way to do this.



**nestBall()**

The nest ball searches the deck for basic Pokémon and allows you to choose one to add to your hand.

A computer screen shot of a program code

Description automatically generated

**potion()**

The potion heals a Pokémon up to 30 hp.

A computer screen shot of a game code

Description automatically generated

**Output**

**Help Output**

A screenshot of a computer

Description automatically generated

**Activating a Card**

A black background with white text

Description automatically generated

**Checking what Pokémon is Active**

A black background with white text

Description automatically generated

**Checking how much health is remaining on Active Pokémon**

A black background with white text

Description automatically generated

**Checking the Prize Pool**

A screen shot of a computer

Description automatically generated

**Checking the current player’s hand**

A screenshot of a computer program

Description automatically generated

**Showing the attack sequence and Prize Picking after a Pokémon is knocked out.**

A screenshot of a computer program

Description automatically generated

**Nest Ball Sequence**

A screenshot of a computer program

Description automatically generated

**Professor’s Research Sequence**

A screenshot of a computer program

Description automatically generated

**Using a potion**

A screenshot of a computer program

Description automatically generated

**Monte Carlo**

The first Monte Carlo checks for the probability of drawing a Pokémon in the initial hand depending on how many Pokémon are in the deck.

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Monte Carlo Charizard Brick**

The second Monte Carlo finds the probability of your deck “bricking” if all your Rare Candies are in your prize pool.

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer

Description automatically generated

**Std Library**

All the comments for these follow the formula of “Finds the X” and then whatever the function finds.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen with text on it

Description automatically generated

A screenshot of a computer code

Description automatically generated

They all contain very simple code. So, nothing worth discussing. The output for the tests is below.

A black background with white text

Description automatically generated

A computer screen shot of a number

Description automatically generated