Pokemon Attack Simulator

Anna Edgar

June 10, 2013

1 Introduction

Pokemon is a turn based battle game where the user raises and battles creatures name Pokemon. Each Pokemon has various attributes that determine how well they will fare in battle and may learn up to 4 attacks. These attacks also have attributes that will help determine how much damage the Pokemon will deal. Even though the total amount of damage each attack will do is easy to determine, it does require a lot of information. The attacking Pokemon's attack and level attributes, the defending Pokemon's defense attribute, and the attack's power attribute all help determine the total amount of damage done. I have created this attack simulator to allow the user to determine the total damage an attack will use. This will allow the user to make more educated moves and better strategize.

1.1 Using XML

When creating a Pokemon quite a few details are needed. In order to speed up the process for the user the information on Pokemon and attacks will be saved to an XML document. This also allows multiple users to build off of each other's work. Ideally, many people would use this simulator and eventually a comprehensive list of all Pokemon would be built. Then a new user would never have to take time to input all of the details for a Pokemon. Reading and writing to the XML does take some additional computation time. However, the benefit of decreased user input outweighs the additional computation time. Ideally, a database would be used to store this information, but for the purpose of this assignment an XML document will suffice.

2 The Attack Class

An Attack object represents an attack that a Pokemon may use in battle. Each object must have a name, element, power, and accuracy.

2.1 Attack Methods

- **__init__(self, name)**
  This method will create a new instance of an Attack that has already been added to the XML. The name parameter must be a string.

- **create(name, element, power, accuracy, special = False)**
  This method add a new attack to the XML document. Some attacks are considered special attacks and will use the Pokemon's special attack and special defense attributes instead of the standard attack and defense attribute.

3 Pokemon Class

A Pokemon object represents the Pokemon who will be using the attacks. Each Pokemon must have a name, attributes, element, and attacks. The attributes must include level, hp, attack, defense, sp_atk, sp_def, speed. When the Pokemon's element matches the element of an attack used additional damage will be dealt, so be sure the correct element is assigned.

3.1 Pokemon Methods

- **__init__(self, name, attributes = None, attacks = [Attack("tackle")])**
  This method will create a new instance of a Pokemon that has already been added to the XML. If attributes are passed in those attributes will be used in place of the XML attributes. Users may have trained their Pokemon differently, so the Pokemon will have non-standard attributes. If no attacks are passed in the default tackle attack will be added. This attack can be overwritten using the add_attack_at_index method.

- **add_attacks(self, attacks)** Up to 4 attacks will be added to the Pokemon. "attack" must be a list of strings. The method will use the XML to find the attacks. Please always create attacks before adding them to a Pokemon.

- **add_attack_at_index(self, position, attack)** This method overrides the attack at the given index.

- **attack(self, index, defending)** This method is arguably the most important method in the Pokemon class. Self is the attacking Pokemon, position is the index of the attack being used and defending is the Pokemon being attacked. This method will use the same formula to calculate damage as the original game uses. An integer representing the total damage will be returned.

- **create(name, attributes, element)** This method will add the Pokemon information to the XML file.

4 Example

This is a simple walkthrough where an attack and 2 Pokemon will be created. One Pokemon will then attack the other and the damage will be noted.

```
#Create the spark attack in the XML
Attack.create("spark", "electric", 65, 100);
#set the attributes for Pikachu
attributes = {"level": 50,"hp": 150,"attack":50
    ,"defense":50,"sp_atk":50, "sp_def":50,"speed":50}
#Create Pikachu in the XML file
Pokemon.create("pikachu", attributes, "electric")
#Pikachu is now avaliable to use, lets make one
pika = Pokemon("pikachu",None, ["tackle", "spark"])
#Create a squirtle
attributes2 = {"level": 50,"hp": 104,"attack":47
    ,"defense":65,"sp_atk":49, "sp_def":62,"speed":43}
Pokemon.create("squirtle", attributes2, "water")
squirt = Pokemon("squirtle",None, ["tackle"])
#Have Pikachu attack squirtle with shock and note the results
print pika.attack(1, squirt)
#When I ran it 69.17 damage was done. There is a bit of
    randomness in the formula so your result may vary
```