Pokémon Arena – Gotta beat 'em all!

Marks: 50   **Due:**

Your company, **The Pulse,** tries hard to be on "the pulse" of what is hot and new with kids, unfortunately all the cool stuff that is new costs too much money for the rights. After extensive research (involving a dowsing rod and tea leaves) the big boss has decided that Pokemon is ripe for a revival. Your job is to mock-up the newest in addictive computer gaming. The final version will have mind-blowing graphics with ear-splitting sounds. Being only a mock-up, your version will have neither graphics nor sounds (and to add either might get you in trouble with the union.)

Pokemon Arena

According to the boss kids these days are not interested in any sort of story. All they want is senseless violence and flashy lights. Your job is to provide the violence. Pokemon arena is a very simple game (after all simple = addictive, right?) The goal is to defeat all of the enemy Pokemon. The game starts with the player choosing which four Pokemon he/she wishes to use. The rest of the game is a series of battles where the player uses his/her four Pokemon to try to defeat each other Pokemon one at a time. Although there may be some changes before the game is released a data file of all the Pokemon in the arena is being provided.

The Pokemon

Each Pokemon has the following stats:

HP: the amount of damage the poor fella can take before being KO'ed

ENERGY: energy is used to pay for attacks. All Pokemon have a maximum and starting value of 50.

TYPE: there are 6 types EARTH, FIRE, GRASS, WATER, FIGHTING and ELECTRIC. The type of the Pokemon determines the type of damage it does, and thus affects weakness and resistance.

RESISTANCE: This can be one of the 6 types from above. If a Pokemon is hit by a Pokemon that he is resistant to, then the damage from that attack is cut in half.

WEAKNESS: This can be one of the 6 types from above. If a Pokemon is hit by a Pokemon that he is resistant to, then the damage from that attack is doubled.

ATTACKS: Each Pokemon has one or more attacks. Each Attack Look like:

ATTACK NAME, ENERGY COST, DAMAGE, SPECIAL

> **ENERGY COST:** Each attack costs an amount of energy, this limits the number of times a particular Pokemon can use its special attack.

> **DAMAGE:** A simple number that will be taken off the enemies HP, but may be modified by a variety of things.

> **SPECIAL:** There are five special type of attacks.

>> **STUN:** on top of normal damage there is a 50% change that the opponent will be stunned for one turn. If a Pokemon is stunned it my not attack or retreat.

>> **WILD CARD:** The attack only has a 50% chance of success. If it does not succeed no damage is done.

>> **WILD STORM:** Base attack has a 50% chance of success, again no damage on a miss, but if it succeeds then the Pokemon does a free wild storm attack (yes this can go on forever.)

DISABLE: The target Pokemon becomes disabled, and its attacks will do 10 less damage for the rest of the battle (to a minimum of zero). A Pokemon can only be disabled once.

RECHARGE: Adds 20 energy to the attacking Pokemon.

Setup

At the start of the game load all Pokemon from file and allow the user to pick four of them to use. All the remaining Pokemon will be used as enemies.

Game play

The game progresses in a series of battles. In each battle are the four Pokemon the user picked (the good guys) and one randomly picked Pokemon from the group that is left (the bad guys.) To win the game and be declared "Trainer Supreme" the user must battle and defeat all of the bad guys one at a time. At the end of each battle all of the good guys whom are still awake get healed 20 HP.

The Battle

At the start of each battle all Pokemon start with 50 energy. This energy is used to activate its attacks, and it can never go above 50. Although each battle is between the player's four Pokemon and one computer Pokemon, only one Pokemon from each side fights at a time (Pokemon have a code of honor.) At the start of the battle the player chooses which Pokemon will start the battle. The player MUST say "<name>, I choose you" (don't make the user type this in, just display it) Randomly determine which enemy Pokemon they will be fighting from the group of bad guys that they haven't fought yet. The battle is played out in a series of rounds. In each round the user can perform one action and the computer can perform one **action**. Randomly determine whether the computer or the user goes first. An **action** is one of the following:

- Attack
  - Subtract the power cost from the current amount of energy the Pokemon has then resolve the attack. Do not allow the user or computer to pick an attack that they can not afford to perform.
- Retreat
  - The Pokemon is replaced with one of your remaining Pokemon. The new Pokemon can not perform an action this turn. The Computer can not retreat because only one of the bad guys is in the Battle at any given time.
- Pass
  - If a Pokemon does not have enough energy to perform an attack then it may have to sit there and do nothing for a turn while its energy recharges.

To resolve the computer's action, simply randomly choose one of the attacks that it can afford to do, if it can't afford any then Pass. After each round each Pokemon recovers 10 energy, to a maximum of 50. (Yes, even those not fighting.)

The Data File

The data file begins with a number that represents the number of Pokemon in the file. Each Pokemon is recorded on one line and looks like:

```
<name>,<hp>,<type>,<resistance>,<weakness>,<num attacks>,[<attack name>, <energy
cost>,<damage>,<special>]
```
e.g.
```
voltorb,40,electric,earth,water,2,shock,10,20, ,explode,40,40,wild storm
```

The square brackets [ ] mean that there can be any number of repeats of this pattern. The number of repeats will be equal to the number of attacks. If an attack has no special then there will be a space in that place.

Submission

Create a Folder in your hand-in folder with the name **"Pokemon"** place only the files from your final working version in that folder.

Pokemon Evaluation

6   Pokemon Class Basics

- This program will require two classes, a PokemonArena class that has your main method in it and a Pokemon class that is used to create a bunch of Pokemon. The Pokemon class should have the access specifiers set correctly, should have accessors methods and should have other useful methods that make making your main class easier. For example if you have a setHP() in Pokemon then, odds are, you are doing all of the logic for taking damage in your main class. If you instead have a method called attack() that takes the enemy Pokemon as a parameter then your main class becomes much cleaner and your Pokemon is useful.

3   Battle all enemies

- These should be easy marks. Is your Arena class setup in a way that allows you to battle each enemy in turn until you beat them all and are crowned "Trainer Supreme"

6   Load From File

- The marks are given here for your file IO and string manipulation. Can you take the data from the file and store it in an ArrayList of Pokemon?

3   Pick User's Pokemon

- Easy marks. Let the user pick the four Pokemon they want to use in the game. Show them a menu and let them pick by number, don't let them pick the same one twice.

6   Basic Battle Setup

- Pick the enemy, randomly determine whom goes first, set the energy at start, alternate who's turn it, give energy between rounds and heal them after the battle is over.

5   Attack Basics

- Pick valid attack, subtract energy, have one Pokemon damage another taking into consideration weakness and resistance.

3   Interface

- It's all going to be text based, but you can do some things to make it look a little better or worse. Try to make life easy on the user, show them their options and let them type numbers rather than typing in "retreat" when they want to retreat, for example.

10   Special Attacks

- stun, wild card, wild storm, disable, recharge.

8   Programming Style

- Comments, choice of variable names, logical division of methods, logical choice of data types.

/50

Creative Option

Some students get all giddy and exited when they just think of Pokemon and some students eyes start to twitch at the thought of being subjected to 4 weeks of "Cute overkill." If you are in this second group feel free to talk to me and together we can design an assignment for you that is large enough, contains enough logical challenges and explores the basics of designing and using objects in a real program.