LaSalle College
Vancouver

VGP 232 Game Tools and Pipeline

Assignment 2a

Console Application (improvement)

Objective:

Refactoring

Using your existing assignment 1 **C# Console Application** improve it by refactoring:

- Extending the Pokemon class (i.e. Pokemon.cs)
  - Implement this method to Pokemon
    static bool TryParse(string rawData, out Pokemon pokemon)
    should similar to float.TryParse(string, out float)
  - Also add an **enum** member **PokemonType** containing the following: Grass, Fire, Water, Bug, Normal, Poison, Electric, Ground, Fairy, Fighting, Psychic, Rock, Ghost, Ice, Dragon
  - Override the ToString() method so it can print the pokemon back in the **CSV (comma separated value)** format (should be done from Assignment 1)
- Add an interface IPersistence
  - bool Save(string filename)
  - bool Load(string filename)
- Add a PokeDex class (i.e. PokeDex.cs) which inherits from List<Pokemon>
  i.e. public class PokeDex : List<Pokemon>
  implements the following methods:
  - Pokemon GetHighestDefense()
  - Pokemon GetHighestAttack()
  - Pokemon GetHighestHp()
  - Pokemon GetHighestSpeed()
  - Pokemon GetHighestTotal()
  - Pokemon GetHighestSpecialAttack()
  - Pokemon GetHighestSpecialDefense()
  - List<Pokemon> GetAllPokemonOfType(PokemonType type)
  - void SortBy(string columnName)
  - Inherit from IPersistance above

Refactoring:

- Replace the **List<Pokemon>** results with the class **PokeDex**
- Move the static **Parse** function out of Program.cs main method and invoke **Load** function of from PokeDex class

VGP 232 Game Tools and Pipeline

- The Pokemon class will have a TryParse(...) mentioned above which Load will call and handle any exceptions that may occur when the values that are being parsed is invalid i.e. invalid number of rows, wrong types to convert etc.

Unit Test:

Add nUnit tests to confirm that your class can do the following:

- PokeDex
    - Find highest defense
    - Find highest hp
    - Find highest attack
    - Find highest speed
    - Find highest special attack
    - Find highest special defense
    - Find highest total
    - Load a pokemon.csv file success, and fail
    - Save the csv file output
        - Note for this, you may need to use the SetUp and TearDown step so it runs it cleanly
- Pokemon
    - TryParse valid values
    - TryParse invalid values

Error Handling:
- invalid path (path does not exist)
- invalid arguments
- invalid data from the csv

C# Helpful links

https://support.microsoft.com/en-us/help/816149/how-to-read-from-and-write-to-a-text-file-by-using-visual-c

https://www.dotnetperls.com/sort-list

Due date

Next class, week 3 (at the start of class 6:30pm)

LaSalle College
Vancouver

VGP 232 Game Tools and Pipeline

Submission

Implement the answers in a new project with the name “Assignment2”. This project should be submitted through committing and pushing through GIT to your VGP232 repository which you shared with the instructor.

Grading

Marks: Out of 100 (10% of final grade)

(60) Functional: Does it compile? Does it meet the requirements and work? Does it give the correct results?

(25) Error Handling: Does your application handle bad input? If so, does it handle failures and exceptions gracefully or does it crash?

(15) Naming convention & Comments: Does it follow the coding standards? Are your variables and method names descriptive? Did you leave descriptive comments on methods that does not have obvious functionality?

Coding Standard

C#
https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions

Late Assignments

After the due date, the student will no longer be able to submit their assignment and will receive a 0 as I will go over the solution in the next class.