Laboratorio 04 - 30/08/2019

Archivos, Tipos por valor y por referencia, Enums

Introducción

En este laboratorio leerás archivos para instanciar objetos, además se verá una introducción a tipo por referencia. Finalmente, deberas agregar Enum para mejorar la calidad de código.

Parte 1: Archivos

- Cree una carpeta en el proyecto llamada Files, luego agregue los archivos Decks.txt y Captains.txt
- Genere un método en la clase *Game* que lea el archivo Decks.txt y agregue los mazos de cartas. Para esto, el archivo tiene la siguiente estructura: El inicio de un mazo se define por medio de la palabra START, el término con la palabra END. Luego cada linea representa una carta con la siguiente estructura *clase,nombre,efecto*, y en caso de ser *CombatCard* tiene dos atributos más *ataque y hero*

  Note que la clase *Game* tiene un atributo llamado *decks* que es una lista de mazos. Debe guardar los mazos en dicho atributo. A la vez, los mazos tienen la propiedad Cards, esta recibe una lista de cartas.
- Cree una lista en la clase *Game* que guarde solo cartas tipo capitán y no agregue estas cartas al mazo (ya que el usuario puede elegir cualquiera de las dos).
- Cree un método en la clase *Game* que lea el archivo Captains.txt y agregue los capitanes a lista anteriormente creada.
- Por último, asegurate que se agreguen de manera correcta las listas de cartas leídas desde el archivo Decks.txt a los objetos clase *Deck*. Debe recordar que las listas son por referencia, y por lo tanto la propiedad Cards del mazo debe recibir la lista de la siguiente forma:

```
decks[i].Cards = new List<Card>(cards);
```

  donde, decks[i] representa un mazo Deck y cards representa la lista de cartas leída desde el archivo. Esto asegura que se cree una nueva lista en un espacio de memoria aparte.

Hints:
Las listas debe inicializarlas antes de llamarlas. Por ejemplo:

```
List<int> sample = new List<int>();
```

Esto lo debe realizar para la lista de mazos como para la lista de capitanes en el constructor de la clase Game.

Por otra parte, para encontrar los archivos puede usar el siguiente comando

```
string path = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.
    Parent.FullName + @"\Files\Decks.txt";
```

1. El comando anterior obtiene el directorio donde se ejecuta el programa con el comando Directory.GetCurrentDirectory(), el cual debería ser:

   C:\user\....\Laboratorio$_X$*bin**Debug**netcoreapp*2,1

2. Luego, obtiene el padre de dicho directorio con Directory.GetParent(Directory.GetCurrentDirectory()), el cual es una carpeta mas arriba, es decir:

   C:\user\....\Laboratorio$_X$*bin**Debug*

3. Luego obtiene el Padre del anterior con Directory.GetParent(Directory.GetCurrentDirectory()).Parent, lo cual retrocede a:

   C:\user\....\Laboratorio$_X$*bin*

4. Finalmente obtiene el padre con su path completo con Directory.GetParent(Directory.GetCurrentDirectory()) que es donde debe estar la carpeta File:

   C:\user\....\Laboratorio$_X$

5. A esto se le agrega que ingrese a la carpeta Files y lea el archivo que corresponde, con + @"\Files\Decks.txt";

Dependiendo del versión de .net que estén utilizando puede que la carpeta netcoreapp2.1 no exista, es decir que Directory.GetCurrentDirectory() retorna C:\user\....\Laboratorio$_X$*bin**Debug*
En dicho caso deben usar el comando con una llamada a Parent menos, es decir:

```
string path = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.
    FullName + @"\Files\Decks.txt";
```

Parte 2: Enums

- Crea una carpeta en el proyecto llamada Enums.
- Agrega una nueva clase en la carpeta llamada EnumType.cs
- La clase debe tener la siguiente estructura

```
public enum EnumType
{
    None,
    melee,
    range,
    longRange,
    buff,
    buffmelee,
    buffrange,
    bufflongRange,
    captain,
    weather
}
```

- Implementa en el código el Enum. Para esto, debes cambiar el tipo del atributo *type* de la clase *Cards* a *EnumType*, luego corrige los errores. Notar que debes cambiar todos los string escritos en el código que comparaban contra el tipo de la carta, por ejemplo, cambiar "melee", por *EnumType.melee*, *null* por *EnumType.None*, etc. Además en la lectura del archivo deberás forzar que el tipo de la carta pase de string a EnumType, para esto un Enum se parsea de la siguiente forma:

```
(yourEnumClass) Enum.Parse(typeof(yourEnumClass),yourString)
```

Entrega

Forma de entrega

Debes crear tu aplicación dentro del repositorio previamente clonado de Github Classroom en `https://classroom.github.com/a/owsPuL7J`.

Al usar los computadores de la universidad debes limpiar las credenciales de git antes de nada. Para esto desde una terminal de git ejecuta los siguientes comandos:

```
$ git config --global user.name "tu nombre"
$ git config --global user.email "tu mail @miuandes.cl"
```

El proceso para entregar es el mismo que explicamos en el video disponible en SAF, esto es: debes realizar un commit, que es una foto al estado actual de tu proyecto y luego subirlo a github, con los siguientes comandos desde una consola:

```
$ git add .
$ git commit -m"tu mensaje"
$ git push origin master
```