Proyecto CandyCrush Consola Orientado a Objetos

Prof. Luis Ernesto Garreta U.

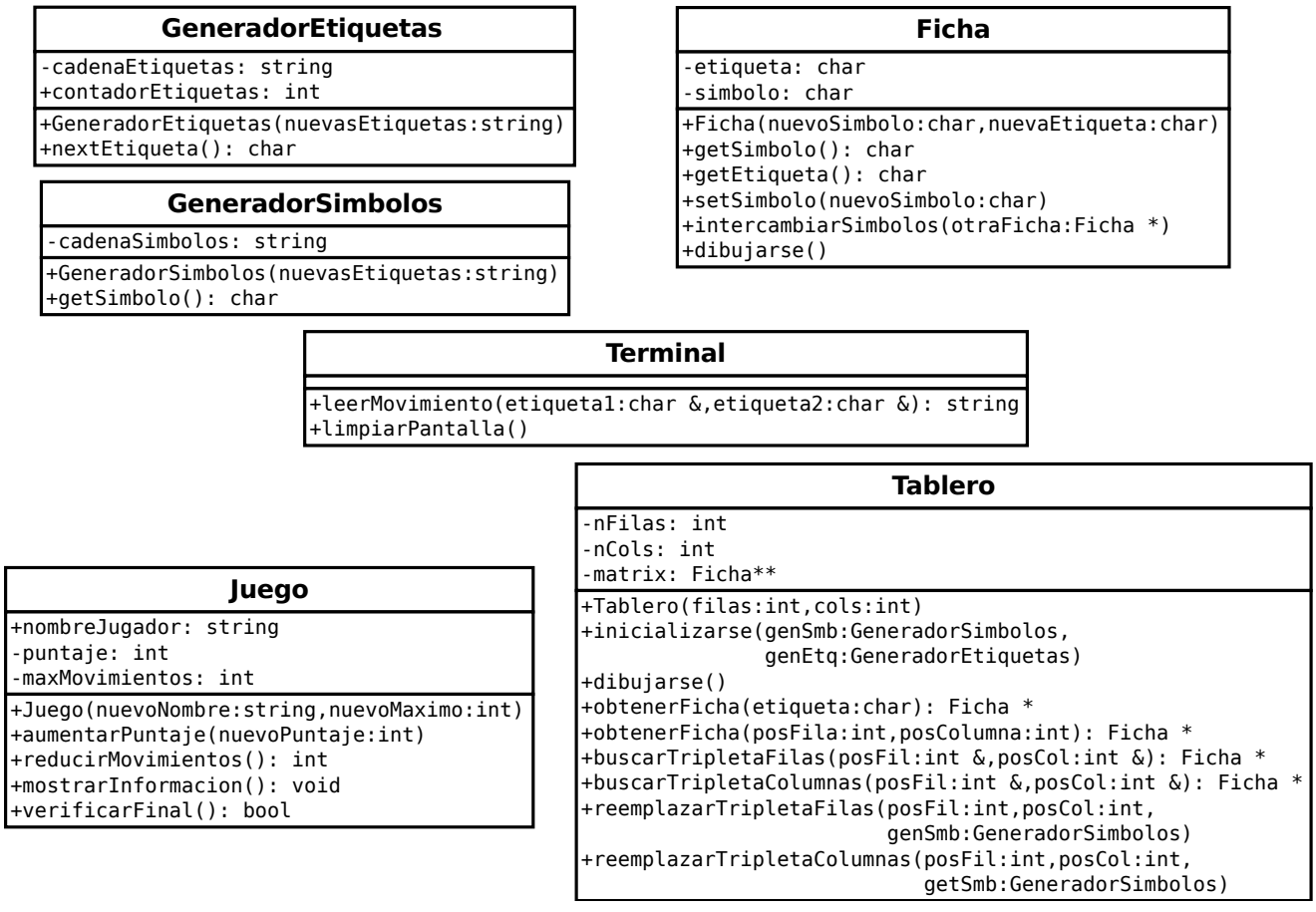
September 29, 2017

1 Introducción

- La siguiente versión de nuestro proyecto de CandyCrush es programarlo bajo un enfoque orientado a objetos.
- Vamos a seguir un enfoque de programación colaborativo donde cada estudiante va a implementar uno o más métodos de diferentes clases de objetos que conforman el juego.
- Para cada método que se le asigne, usted deberá derivar su propia clase a partir de las ya existentes y sobreescribir el método original con el nuevo método desarrollado por usted.

2 Diagrama de Clases de CandyCrush Consola

El siguiente es el diagrama de clases con la especificación de los atributos y métodos del juego CandyCrush Consola:



3 Recursos

Para lograr lo anterior, estarán disponibles los siguientes tipos de archivos:

- Archivo principal del juego con extension .cpp: con la función main que implementa la dinámica del juego:
  - maincandyoo.cpp
- Archivos de cabecera (headers) de extensión .h: con la definición de la clase (Atributos y Métodos):
  - Ficha.h, GeneradorEtiquetas.h, GeneradorSimbolos.h, Juego.h, MiJuego.h, MiTerminal.h, Tablero.h, Terminal.h
- Archivos objetos de extensión .o u .obj: con las implementaciones de los métodos de las clases:
  - Ficha.o, GeneradorEtiquetas.o, GeneradorSimbolos.o, Juego.o, maincandyoo.o, MiJuego.o, MiTerminal.o, Tablero.o Terminal.o
- Archivo de compilación Makefile para compilar desde la línea de comandos con la herramienta *make.* (Para los que compilan en linux o windows desde la línea de comandos)
- Archivo proyecto devcpp, para los que trabajan este entorno bajo windows

4 Instrucciones

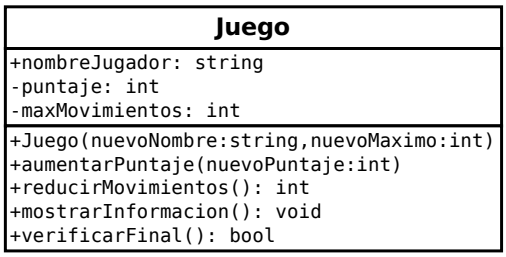
1. Usando la herramienta *git* clone los archivos del proyecto. El link de github es:

   https://github.com/lgarreta/puj-candycrush.git

2. Para cada método que le corresponda implementar, derive una nueva clase y sobreescriba el método. La implementación se debe realizar en dos archivos:
   (a) El .h con la definición de la nueva clase
   (b) El .cpp con la implementación del nuevo código.
3. Incluya los nombres de los archivos tanto del .h como del .cpp en el archivo *Makefile* (o si trabaja en windows, adicione estos archivos a su proyecto en el IDE que utilece)
4. Incluya el encabezado (.h) de su nueva clase en el archivo principal de candy *maincandyoo.cpp*
5. Reemplaze la antigua clase con el nombre de su nueva clase en la línea donde se declara el objeto de esa clase.
6. Compile usando la herramienta *make (o desde su entorno IDE)*.
7. Corrija los errores
8. Ejecute y comprueba que su función trabaja correctamente de acuerdo a los requierimientos de nuestro juego Candy.

5 Ejemplo

Vamos a mostrar un ejemplo donde se sobreescribe un método de la clase "Juego", especificamente el método "*mostrarInformacion*" donde se muestra la información del juego (nombre, movimientos, puntaje). La siguiente es la clase Juego:


Juego
+nombreJugador: string
-puntaje: int
-maxMovimientos: int
+Juego(nuevoNombre:string,nuevoMaximo:int)
+aumentarPuntaje(nuevoPuntaje:int)
+reducirMovimientos(): int
+mostrarInformacion(): void
+verificarFinal(): bool