



Proyecto final: Un Hex Univalluno

Introducción a la programación orientada a objetos

Carlos Andres Delgado S, Ing *

Noviembre de 2016

1. Descripción del juego

El juego Hex es un juego entre dos jugadores, los cuales van colocando por turnos fichas en un tablero compuesto por casillas hexagonales. Cada jugador se distingue por un color y su objetivo es conectar dos laterales (lados) del tablero opuestos. Este juego inicialmente tiene un tablero vacío y cada jugador tiene un color de ficha. Cada jugador va colocando fichas hasta que logra conectar dos laterales opuestos.

Este juego tiene muchas variantes, en las cuales de acuerdo a ciertas configuraciones las fichas pueden cambiar de color, sin embargo, sólo nos enfocaremos a la conexión de los lados asignados a cada jugador.

En este proyecto usaremos una versión del juego con casillas cuadradas:

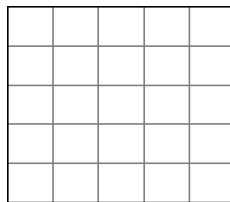


Figura 1: Estado inicial juego

Las jugadas de los jugadores podemos señalarlas con valores, por ejemplo 1, para el jugador 1 y 2 para el jugador 2, o colores. En este ejemplo utilizaremos valores numéricos. Por ejemplo para un estado donde jugador 1 ha jugado 3 veces y jugador 2 2 veces (estamos en su turno) veríamos:

1	2			
		2		
1				
1				

Figura 2: Estado cualquiera del juego

Y para el caso de que el jugador 1 ha ganado (conectando los lados superior e inferior)

* carlos.andres.delgado@correounivalle.edu.co

1	2			
2	1			
2	1	2		
1				
1				

Figura 3: Estado final del juego

1. **[15 puntos]** Diseñe las interfaces de tableros de tamaño 5x5, 10x10 y 15x15. Se puede usar una matriz de botones. En nuestro caso los jugadores:
 - **Jugador 1:** Siempre es un jugador humano, su objetivo es conectar los lados superior e inferior del tablero.
 - **Jugador 2:** Puede ser humano o máquina, su objetivo es conectar los lados derecho e izquierdo del tablero.
2. **[10 puntos]** Los tableros por defecto pueden ir en blanco, o pueden ser cargados por medio de la línea de comandos, ejemplo, si el tablero está contenido:

```
./aplicacion tablero.txt
```

En el caso de un tablero 5x5 el archivo es de la siguiente forma:

```
5
0 1 0 0 0
0 1 0 0 0
0 2 0 0 0
0 2 0 0 0
0 1 2 0 0
```

Donde 0 indica una posición sin jugar, 1 las que ha jugado el jugador 1 y la 2 que ha jugado el jugador 2. Haga el control de excepciones que corresponde para evitar errores inesperados.

3. **[10 puntos]** Diseñe la clase **tablero** la cual almacena el estado de un tablero. Explique cómo modela el estado del tablero. Recuerde que tenemos 3 clases de tableros.
4. **[15 puntos]** Diseñe la clase **jugador** la cual tiene dos tipos: un jugador humano cual recibe las jugadas del usuario y el jugador máquina el cual juega aleatoriamente una posible posición.
5. **[30 puntos]** Realice las validaciones para determinar si ha ganado uno de los jugadores o hay un empate.
6. **[20 puntos]** Interacción con el usuario:
 - a) La aplicación al iniciar debe preguntar al usuario que tablero desea trabajar, 5x5, 10x10 o 15x15
 - b) La aplicación debe preguntar si el rival del usuario es una máquina o otro humano
 - c) La aplicación genera un mensaje indicando si un jugador ha ganado o si hay empate.
 - d) Posibilidad de almacenar un juego, de tal forma se pueda cargar de nuevo.

El jugador máquina es bruto (juega aleatoriamente), sin embargo en el curso de Inteligencia Artificial, aprenderán a diseñar e implementar joponentes interesantes :)!

Aclaraciones

1. El proyecto es en grupos de máximo tres (3) estudiantes.
2. La solución del proyecto (archivo código fuente .cpp, .h y Makefile) debe ser subida al campus virtual en la fecha especificada por el docente. Envíela en un archivo comprimido. Por favor, sólo subir un estudiante del grupo la entrega al campus virtual.

3. Entregue un informe en formato PDF junto al código fuente con una explicación acerca de su implementación, capturas de pantalla sobre el funcionamiento de los puntos del taller y los diagramas de clase del diseño respectivo. La nota del informe va entre 0 y 1 la cual es multiplicada por la nota obtenida en el proyecto.
4. Debe existir un **Makefile** para compilar su proyecto.
5. Evite las super funciones, realice su implementación, reduciendo al mínimo posible el código necesario en la función main() y cada función debe realizar una tarea específica.
6. Tiene 4 horas adicionales para subir su problema, pero cada hora tendrá una penalización ponderada de 10 % por hora. Por ejemplo, si lo envía 30 minutos tarde tendrá una penalización del 5 %.
7. Las primeras líneas de cada archivo deben contener los nombres y códigos de los estudiantes.
8. Generar los archivos con su respectivo archivo de encabezado (.h)
9. Comente los archivos que cree, el no hacerlo traerá penalización en su nota del proyecto.
10. El proyecto debe ser sustentado, de acuerdo a su desempeño individual se asignará una nota entre 0 y 1, la cual se multiplicará por lo obtenido en el proyecto.

```
/*
 * Autores:
 *   Pepito perez código 123123
 *   Aguapanelito Rayado código 123122
 *   Cacique Nohi Cenada código 990023
 * Nombre: archivo.cpp
 * Descripción: Este programa imprime un mensaje y no hace más
 * Fecha de creación: 01 de Enero de 1900
 * Fecha de modificación: 02 de Enero de 1900
 */
#include <iostream>

using namespace std;

int main() {
    cout << "No lo hicimos profe \n Disculpe."<<endl;
}
```