

# 75.41 - Algoritmos y Programación II

*Cátedra Ing. Patricia Calvo - 1er cuatrimestre 2019*

## Trabajo Práctico 3: CUATRO EN LÍNEA 3D V2.0

### Objetivo

Generar una pieza de software que simule el funcionamiento del juego 4 en línea en su versión Jugador contra Jugadores en 3D.

### Enunciado

Conecta 4 (también conocido como 4 en Línea en algunas versiones) es un juego de mesa para dos jugadores distribuido por Hasbro, en el que se introducen fichas en un tablero vertical con el objetivo de alinear cuatro consecutivas de un mismo color. Fue creado en 1974 por Ned Strongin y Howard Wexler para Milton Bradley Company.

### Reglas del juego

El objetivo de Conecta 4 es alinear cuatro fichas sobre un tablero formado por  $n$  filas,  $m$  columnas y  $z$  profundidad ( $n$ ,  $m$  y  $z$  definidas al iniciar el juego) definiendo un cubo. Cada jugador dispone de  $((n \times m \times z) / 2)$  fichas de un color distintivo, pudiendo haber  $L$  jugadores. Por turnos, los jugadores deben introducir una ficha en la columna que prefieran (siempre que no esté completa) y ésta caerá a la posición más baja. Gana la partida el primero que consiga alinear cuatro fichas consecutivas de un mismo color en horizontal, vertical o diagonal en cualquiera de las dimensiones. Si todas las columnas están llenas pero nadie ha hecho una fila válida, hay empate. Como adicional cada jugador tiene 4 posiciones para colocar cartas de

poderes. Al final de cada turno de cada jugador el juego puede darle o no una carta de poder (con un algoritmo definido por el grupo) y se coloca en la posición vacía del jugador. Si está lleno, no recibe el poder. Las cartas de poder se pueden utilizar antes de realizar la jugada.

Cartas de poder disponibles:

- 1) Cambiar el color de una ficha. El color resultante de la ficha va con la cantidad de veces que se jugó la carta. La primera vez que se juega, toma el color del jugador 1, la segunda vez del jugador 2 y así siguiendo.
- 2) Eliminar una ficha del tablero. Al jugar esta carta se elimina una ficha del tablero.
- 3) Intercambio: al jugar esta carta se cambia una ficha con su vecina inmediata, ambas seleccionadas por el jugador.
- 4) Portal: hace la unión de 2 casillero laterales, continuando la secuencia de fichas para formar 4 en línea.
- 5) Ignorar: elimina el movimiento del jugador anterior.
- 6) Carta del grupo: el grupo debe definir el poder y su lógica.

Conecta 4 es un juego de estrategia abstracta donde los contrincantes disponen de información perfecta. Por norma general, el primer jugador tiene más posibilidades de ganar si introduce la primera ficha en la columna central. Si lo hace en las contiguas se puede forzar un empate, mientras que si la mete en las más alejadas del centro su rival puede vencerle con mayor facilidad.

Existe una versión de mesa y otra de viaje. Conecta 4 es una variante de Captain's Mistress, un juego de mecánica similar pero con bolas de madera en vez de fichas de plástico.

Al iniciar el juego se deben presentar 3 niveles de dificultad. El grupo decide que pasa en cada nivel y lo explica en el informe (por ejemplo: limitar la cantidad de fichas, limitar los poderes, la probabilidad de sacar una carta de poder, etc). En algún nivel deben crear una regla que

modifique el juego, por ejemplo: el ganador se evalúa al terminar todos los turnos de la ronda (en vez de que sea al instante de formar 4 en línea), por alguna condición se puede jugar 2 fichas, etc.

## Interfaz de usuario

Toda la interfaz de usuario debe estar basada en texto. El estado del tablero tiene que mostrarse usando caracteres dispuestos en filas y columnas. Por ejemplo, usando ‘ ’ para representar un casillero vacío, ‘X’ para representar un casillero ocupado por un jugador y ‘O’ para el otro.

No es necesario que se limpie la pantalla, simplemente escribir el estado del tablero luego de cada jugada.

Luego de cada turno se deberá dibujar en un bitmap el estado del tablero según la librería Easybmp del campus.

## Cuestionario

Responder el siguiente Cuestionario:

- 1) ¿Que es un SVN?
- 2) ¿Que es “GitHub”?
- 3) ¿Que es “collabnet” y “Tigris”?

## Normas de entrega

Trabajo práctico individual: 5 persona. Cada grupo deberá ponerse un nombre.

Reglas generales: respetar el Apéndice A.

Se deberá subir un archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

Nombre del Grupo-TP2.zip

Deberá contener los archivos fuentes (no los binarios), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF).

La fecha de entrega vence el día lunes 23/05/19 a las 23.59hs.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

## Apéndice A

- 1) Usar las siguientes convenciones para nombrar identificadores.
  - a) Clases y structs: Los nombres de clases y structs siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.
  - b) Métodos y funciones: Deben comenzar con letra minúscula, y si está compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().
  - c) Variables y objetos: las variables siguen la misma convención que los métodos. Por Ejemplo: alumno, padronElectoral.
  - d) Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "\_". Ejemplo: ANCHO, VACIO, COLOR\_BASE.
- 2) El lenguaje utilizado es C++, esto quiere decir que se debe utilizar siempre C++ y no C, por lo tanto una forma de darse cuenta de esto es no incluir nada que tenga .h, por ejemplo `#include <iostream>` .
- 3) No usar sentencias 'using namespace' en los .h, solo en los .cpp. Por ejemplo, para referenciar el tipo string en el .h se pone `std::string`.
- 4) No usar 'and' y 'or', utilizar los operadores '&&' y '||' respectivamente.
- 5) Compilar en forma ANSI. Debe estar desarrollado en linux con eclipse y g++. Utilizamos el estándar C++98.

- 6) Chequear memoria antes de entregar. No tener accesos fuera de rango ni memoria colgada.
- 7) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.
- 8) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.
- 9) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios en el .h que los declara.
- 10) Modularizar el código. No entregar 1 o 2 archivos, separar cada clase o struct con sus funcionalidades en un .h y .cpp
- 11) No inicializar valores dentro del struct o .h.
- 12) En el TP 2 y TP 3 no se pueden utilizar Arreglos, la estructura dinámica debe ser una vista en Clases.
- 13) En el TP 2 y TP 3 no se pueden utilizar objetos estáticos, todas las instancias deben ser dinámicas.
- 14) Si cualquier estructura de control tiene 1 línea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    std::cout << i;  
}
```