

Informe TP N° 2: “N EN LÍNEA”

Grupo “Los Panas”

Joaquin Gandolfo Mariet - 102592

Nicolas Ezequiel Gareca - 101215

Franco Lighterman Reismann - 106714

Guillermina Hoffmann - 104406

Facultad de Ingeniería | Universidad de Buenos Aires

Algoritmos y Programación II (75.11 / 95.15) - Curso P. Calvo

- Índice -

Manual del Usuario - 3

Elementos - 3

Cartas especiales - 3

Menú Principal - 4

Reglas del Juego - 5

Fin del Juego - 6

Visualización del Tablero - 7

Informe del proyecto - 8

¿Cómo se intentó resolver? - 8

¿Cómo trabajamos? - 8

Ideas - 9

Conflictos Resueltos - 9

Conclusión - 9

Manual del Programador - 10

Librerías - 10

Mantenimiento - 12

Manual del Usuario

N EN LÍNEA es una adaptación del famoso juego de Hasbro “4 in a row” a un tablero tridimensional donde M jugadores compiten por lograr alinear N fichas. El primero en lograrlo, ganará.

Elementos

- M jugadores (como mínimo dos).
- Un tablero de tres dimensiones, tendrá un ancho (X), un alto (Y) y un largo (Z).
- Un mazo de cartas especiales.
- N será el número de fichas a alinear para ganar la partida. (mayor o igual a 2 y menor que la mayor de las dimensiones del tablero)
- Una cantidad de fichas por jugador para colocar en el tablero(mayor o igual que N)
- Cada jugador contará con un mazo propio a lo largo del juego de hasta 3 cartas para jugadas especiales.

Cartas especiales

- JuegaDoble: el jugador que la utilice, podrá colocar dos fichas seguidas.
- BloquearTurno: se bloqueará al siguiente jugador (por un solo turno).
- AgarrarCincoFichas: se suman cinco al total de fichas a jugar por el jugador que la utiliza.
- EliminarMazoJugadorSiguiente: al utilizarla, se elimina el mazo del siguiente jugador.

Menú Principal

Al ejecutarse el juego se visualizará el siguiente menú:

```
- Menu Principal -  
  
1. Jugar  
2. Opciones  
3. Salir  
  
Seleccione una opción (1-3):
```

Jugar(1): instancia en la que se da por inicializado el juego.

Opciones(2): instancia donde el usuario ingresa los parámetros del juego. Si no se ingresan, se iniciará un juego con los siguientes valores por default.

1. Ancho del tablero(x): 3
2. Alto del tablero(y): 4
3. Largo del tablero(z): 5
4. Fichas Necesarias para ganar: 2
5. Cantidad de Jugadores: 3
6. Cantidad de fichas por jugador: 30
7. Cantidad de Cartas Totales del Mazo:
8. Cantidad de Cartas 'Juega Doble': 10
9. Cantidad de Cartas 'Bloquear Turno': 8
10. Cantidad de Cartas 'Agarrar Cinco(5) Fichas': 15
11. Cantidad de Cartas 'Eliminar Mazo del Jugador Siguiente': 3

Aclaración: todos los valores ingresados deben ser un número entero y válidos para jugar.

Una vez ingresado a 'Opciones', para modificar alguna de ellas, simplemente ingrese su número correspondiente y continuación indique la cantidad deseada. Al finalizar, los cambios, introduzca 0 y volverá al menú principal. Ejemplo:

```
- Opciones -

1. Ancho del tablero (x): 3
2. Alto del tablero (y): 4
3. Largo del tablero (z): 5

4. Fichas Necesarias para ganar: 3

5. Cantidad de Jugadores: 3
6. Cantidad de fichas por jugador: 20

Cantidad de Cartas Totales del Mazo: 20

7. Cantidad de Cartas 'Juega Doble': 5
8. Cantidad de Cartas 'Bloquear Turno': 5
9. Cantidad de Cartas 'Agarrar Cinco (5) Fichas': 5
10. Cantidad de Cartas 'Eliminar Mazo del Jugador Siguiete': 5

Seleccione una opción (1-10) (o presione '0' para volver): 4

Ingrese la nueva cantidad de fichas para ganar (o presione '0' para cancelar):
```

Salir(3): finaliza la ejecución del programa.

Reglas del Juego

- A cada jugador se le asignará un tipo de ficha aleatoria que será una letra de la A a la Z con la cierta cantidad de fichas establecidas en el menú.
- Al iniciar cada turno, el jugador se le dará una carta del mazo si este posee menos de 3 y el mazo de juego aún no está vacío. Si agarro carta se le informará con un mensaje:

```
¡Agarraste una carta del Mazo!
```

Caso contrario no se informará y el juego continuará.

- Antes de colocar la ficha, el jugador puede optar por si usar una de las cartas especiales que tiene y hacer efecto de ella.

```
¿Desea jugar una de sus cartas Especiales? (s/n): s
-= Cartas de 'I' =-
[ 1. Suma cinco fichas ]
[ 2. Suma cinco fichas ]
[ 3. Bloquea turno al jugador siguiente ]
¿Desea utilizar alguna carta? (1-3) (o presione '0' para negarse):
```

- Una vez arrojada la carta o no, el jugador procederá a dejar caer su ficha en el tablero si este tiene fichas, eligiendo la columna y profundidad (los valores de estas van de 0 al número de su dimensión). Si dicha columna está completa se le volverá a pedir el ingreso nuevamente hasta que el lugar a colocar sea válido.

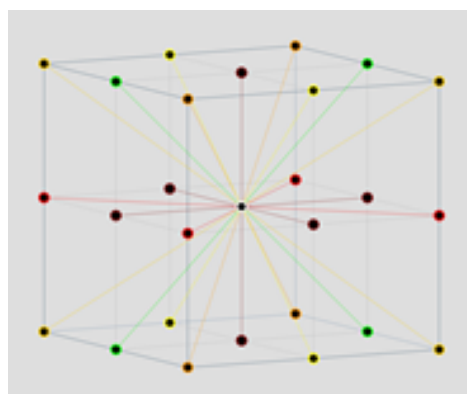
```
Seleccione una columna en la que tirar ficha (0-2): 1
Seleccione la profundidad en la que tirar ficha (0-4): 1
```

- Una vez colocada la ficha, el turno pasa al siguiente jugador.

Aclaración: Si el jugador no tiene fichas ni cartas y el mazo se quedó vacío el turno pasará automáticamente al primer jugador que si tenga alguna de ellas para poder jugar.

Fin del Juego

El juego finaliza en una victoria cuando uno de los jugadores logre alinear N fichas en cualquiera de las 13 direcciones mostradas a continuación:

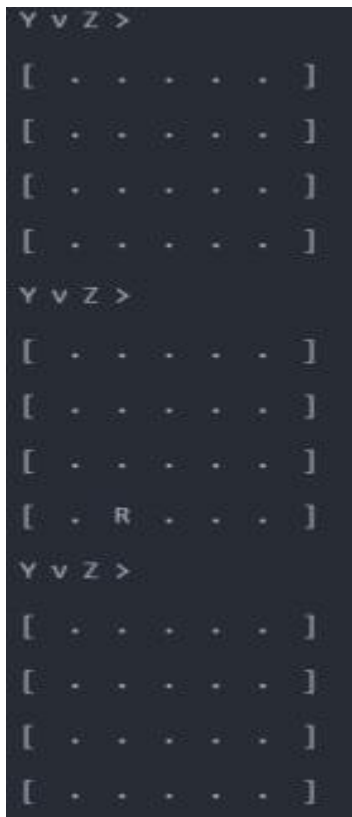


Caso contrario, el juego finalizara en un empate si:

- El tablero está lleno y ningún jugador logró alinear N fichas.
- Todos los jugadores no disponen de cartas ni fichas y el mazo del juego quedó vacío.

Visualización del Tablero

A continuación mostramos el ejemplo de un tablero con los valores por default, donde X es el lado más corto visualizado en “slices” y será el que se mostrará por pantalla a lo largo del juego como se puede ver en la imagen.



Como se puede apreciar, hay una ficha ‘R’ colocada en la columna 1 y profundidad 1.

Si hay 2 o 3 dimensiones con el mismo valor, se informará como en la imagen, los ejes que se están visualizando, y el que no aparece, es el que está dividido en “slices”.

Informe

¿Cómo se intentó resolver?

Para resolver el proyecto del juego “N en línea” en 3 dimensiones con cartas y multijugador, intentamos dividirlo en distintos TDA que tengan la suficiente fuerza para ser creados.

El TP está conformado por el *“TDA Menú”*, *“TDA Juego”* y otro para el Exportador (bitmap) *“TDA Exportador”*. A su vez es necesario la creación de los *“TDA Tablero”*, *“TDA Casillero”*, *“TDA Jugador”*, *“TDA Carta”* y *“TDA Mazo”* para facilitar el manejo de cada elemento para la creación del juego.

Cabe destacar que fue vital el uso de Lista, nodos y cola para la realización del tp, la cual ayudó mucho a entender el manejo de ellas y notar la vitalidad a la hora de crear código con memoria dinámica.

¿Cómo trabajamos?

Para mantener el código de manera grupal, se usó la herramienta GitHub, en la cual cada uno de los integrantes tenía sus TDA asignados a realizar y una vez terminado por cada uno, éste hacía un “Pull Request” en el cual todos los integrantes comentan el código para ayudar mejorarlo y pulirlo cada vez más. Sentimos que fue una herramienta muy importante ya que los “Pulls” si no están del todo bien hasta que no tengan el “ok” de todos los participantes no es mergeado y así no genera conflictos. Además, es imprescindible a la hora de un proyecto que requiere muchas partes de mantenimiento y está constantemente testeando.

Ideas

La decisión del grupo fue crear una especie de menú donde el usuario puede elegir si quiere jugar valores por default o cambiar las características de juego mencionadas en el “Manual del Usuario”. Habíamos pensado en un juego donde se elija la cantidad de jugadores inicialmente y luego se elija un nivel de los 3 niveles diseñados por nosotros, pero nos pareció más dinámico que los usuarios elijan de manera personalizada como quieren jugar.

Conflictos resueltos

Lo que más nos costó fue entender cómo lograr saber si el juego es ganado o no, por lo cual acudimos a la ayuda del profesor Gustavo dada en clases y pulimos aún más su idea la cual nos sirvió mucho.

Conclusión tp

Sentimos por parte de todos los integrantes que el tp fue de mucha ayuda para aprender a manejar tipo de datos abstractos(TDA) y memoria dinámica, para así encarar nuestros futuros proyectos como profesionales con un enfoque distinto a lo que veníamos viendo.

Manual del Programador

Para el desarrollo del proyecto se utilizaron las siguientes librerías para un buen manejo y mantenimiento de el:

Librerías:

1. Librerías propias del juego:

tablero.h: Su función es crear un tablero con dos atributos: uno que consiste en una lista ligada de punteros que apuntan a listas ligadas de punteros a listas ligadas, estas últimas son listas ligadas que apuntan a punteros de la clase Casilleros. Y el otro atributo es un puntero al último casillero donde se colocó una ficha, llamado ultimoCasillero para facilitar las búsqueda de algún ganador en el juego.

casillero.h: Aca se encuentra la clase Casillero la cual almacena la ficha del jugador de no ser vacío y sus coordenadas (x, y ,z). También tiene como atributo una matriz con punteros a los vecinos del casillero para así hacer más fácil la búsqueda de fichas del mismo jugador.

Jugador.h: Crea a cada jugador con sus respectivos atributos; fichaJugador será un carácter creado de manera aleatoria, cantidadDeFichas será indicada en el menú principal. Creará para cada instancia Jugador un mazo de cartas con un capacidad máxima de tres, que será una lista de memoria dinámica.

mazo.h: Es la encargada de crear nuestro mazo del juego, donde este será una cola de puntero a Carta. El mazo se construye de manera aleatoria donde vamos acolando los punteros a cartas especiales con la cantidad de cada tipo de cartas elegidas por el usuario para así construirlo y quede listo para jugar. Con esto se podrá retirar las cartas para el uso dentro del juego.

carta.h: Es donde se encuentra declarada la clase Carta, donde el tipo de carta puede ser: *BloquearTurno*, *JuegaDoble*, *AgarrarCincoFichas* y *EliminarMazoSiguiente*, y cada una con su respectiva descripción para mostrar al usuario que hacen sus poderes.

menu.h: Van a ser los encargados de la interacción con los usuarios a jugar pidiendo las características del juego para así poder ir cumpliendo con las reglas de nuestro “N En Línea” e ir jugando.

Juego.h: Su función principal es crear nuestro juego con dichas características elegidas por los usuarios en el menú e ir conectando cada una de ellas. Su función principal es manejar el tablero, el mazo de cartas, cada jugador con sus propiedades y el estado del juego para saber si se llegó a ganar o empatar.

constantes.h: se explica por sí mismo. Contiene definiciones a valores que se mantienen constantes en todo el programa, pero que pueden ser cambiados antes de la compilación de éste hasta cierto punto si así se desea.

exportador.h: contiene la documentación para usar el exportadorBMP.

2. Librerías de la cátedra:

- a. **listaLigada.h**
- b. **nodo.h**
- c. **cola.h**

Aclaración: todas las librerías listadas son clases pertenecientes al trabajo práctico número 2.

Mantenimiento

Entre las cosas a tener en cuenta al momento de realizar mantenimiento sobre el código fuente del programa, es importante saber que:

- Este proyecto fue desarrollado en el entorno de desarrollo integrado Visual Studio Code, con la extensión de Windows Subsystem for Linux (WSL). La consola de distribución instalada en WSL (Debian Buster), es donde se compilaban los archivos haciendo uso del compilador g++ en el estándar C++98.

- Sin embargo, la versión final fue probada compilando como un proyecto en el IDE Code::Blocks, para el cual se obtuvieron resultados exitosos.