Trabajo Práctico 1: Tesoro Binario V1.0

Datos personales

Alumno: Ontiveros Emilio
DNI: 36089085
Legajo: 109048
Mail: eontiveros@fi.uba.ar
Nombre del archivo zip entregado: "109048-TP1.zip"

Cuestionario

1) ¿Que es un Debug?:
Es la herramienta utilizada y procedimiento realizado para purgar los errores en el código. Un "bug" es un error en el código, de ahí la palabra "Debug"
2) ¿Qué es un "Breakpoint"?:
Es un punto intermedio que se setea para para detener la ejecución durante el debuging con la finalidad de poder analizar el estado actual del programa en esa precisa instrucción.
3) ¿Qué es "Step Into", "Step Over" y "Step Out"?
Son acciones que el desarrollador puede utilizar para recorrer el código durante el debuging.

- **Step into** permite acceder a función para ver el correcto procesamiento de las instrucciones dentro de ella.
- **Step over** permite "saltearse" la revisión de la instrucción actual para continuar revisando la siguiente.
- **Step out** le permite "volver" de la instrucción que se está revisando hacia aquella que se encuentra por fuera. Así como el step into se utiliza para entrar a una función, el step out se utiliza para salir de ella.

Manual de usuario

Indicaciones previas: Se asume que el usuario leyó y entendió el siguiente documento, que sus acciones son bienintencionadas y que está intentando jugar el juego de la manera que corresponde.

Ingreso de casillas: El juego se realiza sobre un tablero de 20x20 cuyos casilleros están numerados del 0 al 19. Siempre que a los jugadores se les solicite ingresar un casillero, se hará de la siguiente manera:

- En cuanto el programa lo solicite, el usuario indicara por consola las coordenadas de la celda pedida.
- Para hacerlo ingresara un valor de 4 cifras que determinara 2 pares de valores.
    - Las primeras dos cifras corresponden al par de la columna
    - Las últimas dos cifras corresponden al par de las filas.
    - Algunos ejemplos:
        - **1203**, corresponderá la casilla de la **columna 12** y **fila 3**.
        - **0214**, corresponderá la casilla de la **columna 02** y **fila 14**.
    - Es importante tener en cuenta que, en ambos casos, los pares irán del 00 hasta el 19, ***evitando colocar valores fuera de esos rangos para cada par***.
- Esta será la forma en que los usuarios colocaran tanto espías como tesoros.

Inicio de juego:

- Al inicio del juego, el tablero estará vacío.
- El jugador 1 colocara sus 4 tesoros en el tablero según lo indique el programa. Ingresando sucesivamente las celdas en las que desea esconderlos.
- Luego el jugador 2 repetirá los pasos del jugador 1 hasta que ambos hayan colocado ambos tesoros.
    - Es importante comprender que debido a que los tesoros son “Secretos”, ningún jugador conocerá la posición de los tesoros de su oponente.
    - Es posible que ambos decidan enterrar sus tesoros en la misma casilla sin saberlo. Esto está permitido.

Sucesión de turnos: Una vez comenzado el juego, los turnos irán ocurriendo sucesivamente hasta que alguno de los 2 jugadores haya cumplido con la condición de victoria. Durante cada turno ocurrirán los siguientes eventos.

- **Colocación de un espía en un casillero:** El jugador del turno actual colocara un espía en un casillero donde no aun no haya un espía propio.
    - Si un jugador decidió colocar su espía sobre uno de sus cofres, deberá mover ese cofre hacia otro casillero adyacente. Este movimiento podrá ser vertical, horizontal o diagonal. Es importante tener en cuenta que, al mover el cofre, no se debe elegir una casilla que ya posea cofres o espías
    - Si el jugador coloco su espía en un casillero que no contiene un cofre propio, se la dará la opción de mover cualquiera de sus cofres como si hubiera colocado un espía sobre él.
    - Si el jugador coloco su espía en un casillero que contiene un cofre enemigo. Se le avisara al jugador que el espía ha comenzado a cavar para desenterrar el cofre, inhabilitando la casilla para ambos jugadores por 5 turnos.

    - Si el jugador coloco a su espía sobre un casillero en el cual ya había un espía enemigo, ambos espías morirán liberando la casilla.
- **Recolectando el tesoro:** Al final del turno, todos los espías que estén cavando avanzar un turno hacia el tesoro. Aquellos que recolecten un tesoro, se irán a su casa dejando la casilla vacía.
- **Condición de victoria:** Si un jugador desenterró todos los cofres de su oponente, habrá ganado y la partida habrá terminado. En caso de que aun queden cofres por desenterrar, comenzara el turno del jugador contrario.

Manual de programador

Introducción

- A fines prácticos el programa se comporta sobrescribiendo una matriz dinámica denominada “Tablero” o “Board”, en esencia es un array de arrays de enteros.
- El programa reescribe los valores en cada posición con un formato de similar a “Tablero[columna][fila]=nuevoValor”.
- A cada posición se la llama celda. A cada celda se le asignan diferentes valores que representaran el tipo de celda es, por ejemplo “Espía” o “Tesoro”.

Partes del programa

Debido a que no me fue posible separar el programa en diferentes clases, decide dividir el main de la forma mas ordenada posible. Se encuentran listadas en orden a continuación con su principal función.

Variables globales

En esta sección se crean las constantes y variables globales del programa. Se utilizan para:

- Definir los tipos de valores que puede haber en las celdas

```
//-----------------------------------------------------------------
// VARIABLES GLOBALES | Esta seccion gestiona las constantes globales.
//-----------------------------------------------------------------
// Constantes que determinan los valores de las celdas
const int EMPTY = 0;           // No hay tesoro
const int CHEST= 6;            // Hay tesoro
const int DIGGINGTIME = 5;     // Cavando
const int SPY = 7;             // Espia
const int CHESTS = 8;          // Hay 2 tesoros
```

- Activar el menú de debuging (sesteándolo en True)
- Definir las configuraciones de juego tales como el tamaño del tablero (20x20) o los cofres a encontrar (4).

Coordenadas

En esta sección se utiliza principalmente para definir la función **askForCoordinate()** la cual solicita al usuario ingresar un valor de 4 cifras que serviría para identificar filas y columnas. A lo largo del programa se la utiliza siempre que se requiere que el usuario ingrese información nueva por lo que es una de las funciones principales a entender dentro del código.

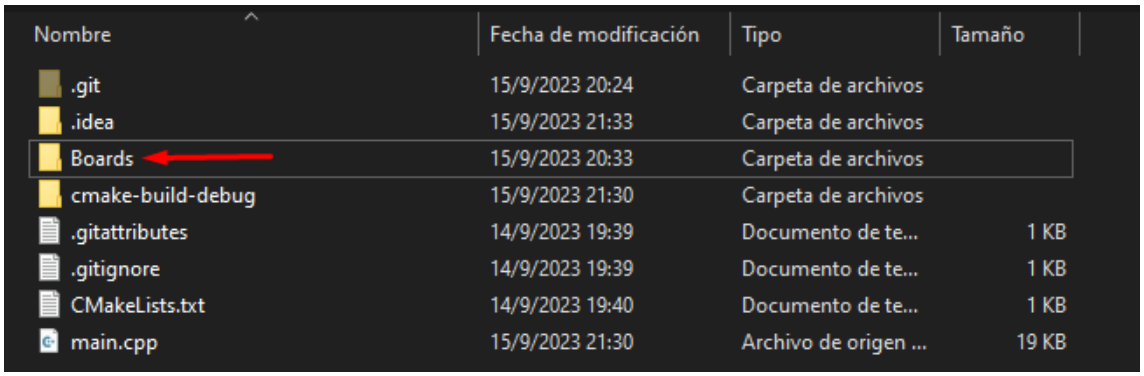
Salida

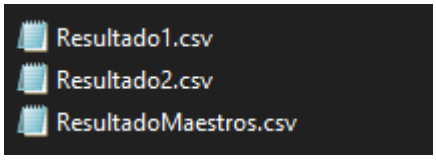
En esta sección tiene como finalidad encargarse de la extracción de información del juego para que llegue al usuario.

La función “exportBoard()” exportar 3 archivos.csv. Uno para cada jugador con la información que pueden ver y un archivo maestro con toda la información de las celdas.

Los tableros se encuentran en la carpeta boards.

IMPORTANTE: Es necesario aclarar que el programa no obtiene información de ellos, simplemente sirven para para mostrar la información al usuario.





Tablero
En esta sección se encarga principalmente de la creación y actualización del tablero turno a turno. La función “printBoard()” permite al desarrollador ver por consola el tablero turno a turno, pero se encuentra deshabilitada por defecto. Se encuentra incluida en el código, pero solo será posible verla activando el modo debug.

Cofres
En esta sección se encarga de la colocación e interacción entre cofres y espías.

Turnos
En esta sección se encarga de organizar la progresión del turno y dar paso al siguiente jugador una vez que el anterior turno ha finalizado.

Principal
En esta sección se encuentra el main, el cual se utilizará para correr las demás funciones. Una vez ejecutados los primeros pasos, el programa se mantendrá en un loop de turnos hasta que se haya completado la condición de victoria.

“Partes a mejorar”
En el programa algunas secciones tienen la indicación “TODO”, utilizada para las secciones que aun requieren revisión.

Condición de victoria

```
// Condicion de victoria
int playerOneRemainingChests = CHESTQUANTITY;       // Cofres restantes del jugador 1
int playerTwoRemainingChests = CHESTQUANTITY;       // Cofres restantes del jugador 1
bool victory = false;                               // Condicion de victoria.
```

Cada turno, luego de que el jugador haya completado todas sus acciones, se ejecuta una especie de sort que recorrerá todos los valores del tablero.

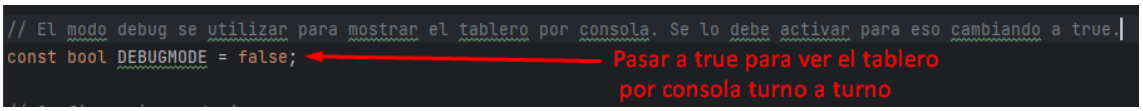
Cuando se ejecute el “UpdateBoard()” de cada turno, se volverá a verificar la condición de victoria.

Inicialmente el programa regresara la cantidad de cofres de cada equipo a 0. Y luego ira sumando los que haya encontrado en el tablero. Al terminar la revisión, si se da el caso de que un jugador no posee tesoros enterrados, el jugo le dará la victoria al oponente.

```
void updateBoard(int** board){
    // Vacia el contador de cofres, para ir sumandolos a medida que aparescan
    playerOneRemainingChests = 0;
    playerTwoRemainingChests = 0;
```

Modo debug

El código cuenta con un modo debug para ver el tablero por consola a medida que se juega, para eso se debe pasar el booleano a "true". Actualmente esta deshabilitado para que los jugadores solo puedan ver la consola y sus tableros.



Consola

Ingreso de valores: Se asume que el usuario es bienintencionado y siempre ingresa enteros positivos de 4 cifras. Estos valores indican 2 cifras para la columna y 2 para la fila tal como se aclara en el manual de usuario. En caso de que se ingresen otros valores el programa tiende a interpretarlos dentro de lo posible pero no es posible garantizar que funcione correctamente.

Casillas

Casillas compartidas: Las casillas (o celdas) pueden poseer diferentes valores dependiendo del tipo de elemento que halla en ellas. Los valores numéricos de la celda determinan el estado en la que se encuentra, (Vacía, con tesoro, con espía o inhabilitadas).

- **0 vacío:** No hay espías ni cofres en el casillero y la casilla esta habilitad para ser elegida.
- **1 a 5 Inhabilitada:** La casilla se encuentra bloqueada por tantos turnos como indica su número. Representan el tiempo que se requiere para desenterrar el cofre de esa celda. Estos valores son siempre positivos. Al final de cada turno su número baja en uno, quedando nuevamente libre al llegar a cero.
- **|6| Cofre:** Representa un cofre enterrado por un jugador. ***(Ver casillas con signo)***
- **|7| Espía:** Representa un espía oculto. ***(Ver casillas con signo)***
- **8 múltiples cofres:** Si ambos jugadores colocaron sus cofres en el mismo lugar. Se indica la celda con el numero 8.

Casillas con signo (Cofres y espías): En el caso de las celdas de cofre simple y espías, el signo de las celdas determina a que jugador pertenecen, siendo el positivo el jugador 1 y el negativo para el 2.

- **6 y -6 Cofre:** Representa un cofre enterrado por un jugador. El "6" para el jugador 1 y "-6" para el jugador 2.

- **7 y -7 Espía:** Representa un espía oculto. El “7” para el jugador 1 y “-7” para el jugador 2.
- Es importante tener en cuenta, que cuando hay más de un cofre en una celda el signo no es relevante, ya que cuando un espía caiga en ella, podrá mover su cofre a otra posición y comenzar a cavar.
- Es importante tener en cuenta que cuando el programa exporta los tableros de los jugadores. Los valores de -6 y -7 del jugador 2, serán pasados a positivos para mostrarlos de forma mas amena a su tablero. Esto se realizará automáticamente en el proceso de exportación