# Trabajo Practico 1: BATALLA CAMPAL V1.0

# 75.41-Algoritmos y programación II

Cuestionario:

1- ¿Que es un debug?
2- ¿Qué es un “breakpoint”?
3- ¿Qué es “Step into”,”Step over” y “Step out”?

1. El termino ***debug***,***debugging*** *o* ***depuracion*** viene de la palabra *bug* en ingles que se traduce como un error en el código que este haciendo que este funcione de forma adecuada. De ahí , el termino ***debugar*** es la acción de eliminar estos mismos errores en el código para que este funcione como corresponde.

Las paginas de desarrollo web , cuentan con sus propios **debuggers**, son fundamentales para códigos que cuenten con miles y miles de líneas de código y ahorran una enorme cantidad de tiempo a que un programador tenga que buscar manualmente cada error en el código.

Otra herramienta usada para ***debugar*** son los ***logs*** de errores, que son listas que dicen que fue lo que paso al momento de la ejecución de un código, línea por línea , lo que ayuda mucho a saber donde empezar para ***debugar*** un programa.

En conclusión los ***debuggers*** son herramientas fundamentales para los programadores que trabajan en grandes proyectos, estas herramientas ahorran el tiempo de búsqueda de errores y agilizan el tiempo de codeo.

2.Los ***breakpoints*** o puntos de interrupción son puntos de un código o programa , donde se decide detener la ejecucion por algún motivo investigar el valor de las variables o ver que ocurre en ese momento en la ejecución del código.

3.Step into, step over y step out son funciones normales de un debug, cuya función es la ejecución paso a paso de un código o programa.

-***Step into****: esta función ejecuta la siguiente línea de código de donde se encuentre el programa. Si esta línea es una llamada a función, ingresa en esta.*

-***Step over****:esta función ejecuta la siguiente línea de código, pero, si esta es una llamada a función, la omite y no ingresa en ella.*

-***Step out****: esta función , si me encuentro dentro de una función, sale de ella.*

Padron: 108205. Burgos Lautaro.

# MANUAL DE USUARIO

Batalla campal es un juego del estilo juego de mesa, se juega de  a 2 jugadores, cada uno introduce 3 soldados a un tablero(No pueden ponerse en la misma posición 2 soldados de distintos jugadores, inicialmente). Se ingresa las coordenadas de cada uno de los soldados(en formato,*fila* y *columna)*, acorde a lo que indica la pantalla, 3 por cada jugador. Los soldados del jugador 1 se representan en el tablero con un "1" y los soldados del jugador 2 respectivamente con un "2".

El tablero consta de 10 filas y 10 columnas de casilleros. En estos casilleros va a ser donde los soldados están parados.

El juego es por turnos , en cada turno , el jugador realiza un disparo y un movimiento de un soldado(de a 1 casillero).En cada turno el tablero se actualiza y se crea un archivo "tablero.txt" donde se guarda el estado actual del tablero.

**Disparo:**

Cuando un jugador dispara hacia alguno de los casilleros del  tablero ,si en este casillero, hay un soldado del jugador contrario, este es eliminado, y el casillero donde se encontraba queda, "inhabilitado".

**Movimiento:**

Luego de disparar , el usuario puede eligir uno de sus soldados en el tablero, y moverlo 1 casillero en 8 direcciones:

*Arriba,abajo,izquierda,derecha, diagonal arriba derecha, diagonal abajo derecha, diagonal abajo izquierda y diagonal arriba izquierda.*

Si al moverse , en el casillero hacia donde se movio el soldado, hay un soldado del jugador contrario, ambos soldados son eliminados, y el casillero queda "inhabilitado".

**¿Que es un casillero inhabilitado?**

Un casillero inhabilitado queda inaccesible para que un soldado se mueva hacia el, en caso de que un soldado intente moverse a un casillero inhabilitado , este jugador perdera su turno de moverse.  Este se representa en el tablero con una "X".

**¿Cuándo termina el juego?**

El ganador del juego será el que elimine los 3 soldados del jugador contrario, mediante disparos o moverse encima. En caso de que ambos se queden sin soldados en el mismo turno , se producirá un empate.

Padron: 108205. Burgos Lautaro.

# MANUAL DEL PROGRAMADOR

El archivo principal Tp1.cpp, consta de una estructura simple, basándose casi completamente en las funciones, estructuras y constantes de la biblioteca funciones.h.

**Estructuras:**

Las 2 estructuras que se utilizan son Soldado y Casillero.

La estructura de tipo Soldado cuenta con una variable de tipo caracter que identifique a que jugador pertenece, y dos variables enteras para representar su coordenada, fila y columna.

La estructura de tipo Casillero , aparte de también tener 2 variables enteras para su coordenada, cuenta con una variable booleana para representar si esta activo (true) o si no lo esta(false), y también cuenta con una variable tipo caracter estado, que es utilizada para representar su estado actual a la hora de imprimir el tablero.

**Main:**

El tablero es una matriz de tipo Casillero de 10x10, y los soldados se guardan en 2 vectores de máximo 3 soldados cada uno(1 vector para cada jugador), con sus respectivos topes.

El juego se inicializa mediante la función **inicializarJuego** , que se encarga de solicitar las coordenadas de los 3 soldados de cada jugador, guardándolos en sus respectivos vectores y aumentando el tope según corresponda, tambien deja el tablero con todos sus casilleros en estado “VACIO”.

Acto seguido , la función **actualizarTablero** , cambia los estados de cada casillero en donde corresponda un soldado , de “VACIO” , a “1” o “2” según corresponda. Luego muestra el tablero por pantalla mediante la función **imprimirTablero** y crea un archivo .txt del estado actual del tablero a través de la función **guardarArchivoTablero.**

**Bucle del juego:**

El juego en si, se encuentra dentro de un bucle while, cuya condición de corte es una variable booleana de nombre **juegoTerminado**, que se inicializa fuera del bucle, en estado **false.**

**Funcion ejecutarTurnos:**

Esta función contiene todos los pasos correspondientes de ambos turnos.

Se vale de la función **turnoJugador:**

Esta a su vez se vale de la función **solicitarDisparo** y la función **disparar**, la primera pide por pantalla la coordenada hacia donde disparar y la otra realiza la acción en si de

disparo y pone el campo estado del Casillero hacia donde se disparo , en inhabilitado "X" mediante la función **desactivarCasillero**, y elimina si corresponde a un soldado mediante la función **eliminarSoldado**.

Luego de disparar ,se realiza el movimiento de un soldado,se elige cual se va a mover mediante la función **seleccionarSoldadoMover,**función que le muestra un menú al usuario por pantalla con la posición de cada uno de los soldados que tenga disponible y le solicita que elija.
Luego se ejecuta la función **moverSoldado** que guarda en una variable de tipo carácter lo que devuelva la función **solicitarMovimiento** que será un carácterquerepresenta uno de los movimientosposibles(*Arriba,abajo,izquierda,derecha, diagonal arriba derecha, diagonal abajo derecha, diagonal abajo izquierda y diagonal arriba izquierda*), cada uno fue definido como constante en funciones.h. Luego mediante un **switch** se realiza el movimiento elegido, siempre y cuando , el destino elegido sea valido (Que se encuentre dentro del tablero y el casillero este activo) . Dejando el casillero donde antes se encontraba, en estado "VACIO". En caso de haberse movido a un casillero donde desde antes se encontraba un soldado del jugador contrario, ambos son eliminados, esto se comprueba mediante la función **soldadosIguales**. Este proceso se repite para el jugador 2. Osea que en la función **ejecutarTurnos** realiza un turno de cada jugador en una sola ejecución de la función.

Todo este proceso se repite dentro del bucle while hasta que uno de los 2 jugadores se quede sin soldados, dejando como ganador al jugador que aun tenga algún soldado. O un empate en caso especial que ambos se queden sin soldados en el mismo movimiento. Mostrando un mensaje dependiendo de cual sea el caso.[1]