

TP1 – Algoritmos y programación II

Cátedra Calvo

Alumno: Nahuel Nicolás Gomez

N° Padrón: 106514

INFORME:

Este trabajo se trata sobre realizar una recreación de un juego llamado “El juego de la vida” de

John Horton Conway.

El trabajo lo dividí en 6 archivos:

1. **Principal.cpp** = En este archivo se incluirán las bibliotecas utilizadas y solo existirá la función MAIN con sus respectivas acciones.
2. **Celulas.h** = Aquí se crearan las estructuras a utilizar para el juego completo.
3. **Impresiones.cpp** = Será un archivo con las funciones que representen una impresión por pantalla, como por ejemplo, mostrar el mapa.
4. **Impresiones.h** = Son las declaraciones sobre el archivo “Impresiones.cpp”.
5. **Acciones.cpp** = Este archivo se basa en incluir todas las acciones que se realizan en el juego, como por ejemplo, pedir la cantidad de células vivas al comienzo de la partida y cómo posicionarlas en el terreno.
6. **Acciones.h** = Son las declaraciones sobre el archivo “Acciones.cpp”.

Mapa:

- Es un rectángulo 20x80.
- En los bordes se encuentran los números de las posiciones para que resulte mucho más sencillo ubicar a las células en un primer turno.
- Las células vivas están representadas con un “V” y las células muertas con un “-”.

Ejemplo:

	1	2	3
1	-	-	-
2	-	V	-
3	-	-	-

En este caso, la posición (2,2) posee una célula viva, mientras que en las demás posiciones solamente existen células muertas.

Juego:

- Se podrán jugar hasta 1000 turnos dado que se utiliza un vector de turnos con 1000 posiciones.

MANUAL DEL PROGRAMADOR:

El juego está dividido en estructuras, para que el trabajo se realice como mucha mayor facilidad, dado un tipo de dato "juego_t" se podrá manipular todo el contenido del juego. Esto permite un mayor entendimiento del código y que las funciones no requieran demasiados parámetros.

El juego utiliza muy pocos "Números mágicos", provocando que varias acciones se puedan modificar rápidamente desde la parte superior del código.

Es un código muy legible dado que cada acción está dividido en una función y poseen nombres intuitivos. Para un posible cambio es muy beneficioso.

MANUAL DEL USUARIO:

- Se trata de un juego donde no hay vencedores ni vencidos. Se lo conoce como un autómatas que va realizando acciones paso por paso dado una serie de instrucciones.

Pasos a seguir en la ejecución:

1) Luego de dar la bienvenida, las reglas y el mapa de juego, se deberá introducir la cantidad de células vivas al comienzo de juego.

2) Posteriormente, ubicará cada célula, primero se pedirá una fila y finalmente una columna. Si no es posible colocarla (Porque se extendió de los límites o si ya existe una célula ubicada en la misma posición) se deberá escribir nuevamente la fila y columna para la misma célula.

3) Se irán mostrando, por cada célula posicionada, el mapa actual colocando a las células que ya se han posicionado.

4) Luego de que se termina de ubicar a todas las células iniciales, se podrán realizar tres acciones ->

■ Jugar Turno (J).

■ Reiniciar (R).

■ Salir del juego (S).

■ Se jugará el turno próximo.

■ El juego volverá hasta el primer turno, exactamente como fueron posicionadas las células por primera vez.

■ El juego terminará.

CUESTIONARIO

- 1) ¿Qué es Debug?
- 2) ¿Qué es un “Breakpoint”?
- 3) ¿Qué es “Step Into”, “Step Over” y “Step Out”?

1. ¿Qué es Debug?

Debug, o depuración de un programa, es el proceso de corrección o modificación del código para su correcto funcionamiento. Está dirigido a desarrolladores para investigar fallos en programas.

Cuando se depura un programa se hace un seguimiento del funcionamiento de dicho programa y se van estudiando los valores de las distintas variables, así como los resultados obtenidos en las operaciones.

2. ¿Qué es un “Breakpoint”?

Un Breakpoint es un punto en nuestro código en el que al ejecutar con un depurador, la ejecución se detendrá y podremos ver los valores que tienen las variables, y diversos datos en ese momento sobre el programa para identificar errores y posteriormente solucionarlos. Estos pasos se realizan hasta que el programa no contenga ningún fallo.

3. ¿Qué es “Step Into”, “Step Over” y “Step Out”?

Dado que “debuggear” consiste en ejecutar el programa paso a paso, existen funciones que ayudan a ese proceso.

- Step Into = Se trata de ejecutar la siguiente línea de código. Si existe una función en esa misma línea, el depurador entrará y podrá ejecutar paso a paso cada línea de la función.
- Step Over = En este caso, aunque siga la misma idea de “Step Into” de ejecutar la siguiente línea de código, si esa línea es una función, la ejecuta sin entrar en ella.
- Step Out = Este comando permite que la aplicación se ejecute toda la función provocando la salida de la función donde se encuentra.