

# 75.41/95.15 - Algoritmos y Programación II

## Cátedra Calvo - 2do cuatrimestre 2020

### Trabajo Práctico 1: Juego de la Vida V1.0

Autor: Alfonso Campodonico

Legajo: 103598

## Objetivo

Realizar una aplicación que lleve adelante la ejecución del juego de la vida en un tablero limitado, de 20 filas por 80 columnas, a partir de una configuración inicial de células vivas indicadas por el usuario.

## Enunciado

El juego de la vida es en realidad un juego de cero jugadores, lo que quiere decir que su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posterior.

El "tablero de juego" es una malla formada por cuadrados ("células"), que en nuestro caso estará limitado a 20x80. Cada célula tiene hasta 8 células vecinas, que son las que están próximas a ella, incluso en las diagonales. Las células tienen dos estados: están "vivas" o "muertas". El estado de la malla evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por turnos). El estado de todas las células se tiene en cuenta para calcular el estado de estas al turno siguiente. Todas las células se actualizan simultáneamente.

Las transiciones dependen del número de células vecinas vivas:

- Una célula muerta con exactamente 3 células vecinas vivas "nace" (al turno siguiente estará viva).
- Una célula viva con 2 o 3 células vecinas vivas sigue viva, en otro caso muere o permanece muerta (por "soledad" o "superpoblación").

## Requerimientos

1. La aplicación debe solicitar al usuario que indique las posiciones en las que se colocarán células vivas al inicio del juego; todas las otras están muertas.
2. El usuario debe poder indicar tantas células vivas como desee, desde ninguna hasta 1.600.
3. Las células vivas se indicarán por fila y columna, siendo (1, 1) la celda superior izquierda y la (20, 80) la celda inferior derecha.
4. Una vez finalizado el proceso de carga de células vivas debe comenzar el juego, mostrando el estado inicial del tablero por consola utilizando caracteres e indicando la cantidad de células vivas.
5. A partir de este momento el usuario debe decidir entre:
  - ejecutar un turno,
  - reiniciar el juego o
  - terminar.

6. Con cada ejecución de un turno se debe mostrar:
  - el estado del tablero (usando caracteres para representarlo)
  - la cantidad de células vivas
  - la cantidad de células que nacieron en el último turno
  - la cantidad de células que murieron en el último turno
  - el promedio de nacimientos a lo largo del juego
  - el promedio de muertes a lo largo del juego
  - si el juego se congeló, es decir si no sufrió modificaciones en dos turnos consecutivos.

## Cuestionario

¿Qué es Debug?

Es el proceso de encontrar errores de ejecución en el código. En este, el programador coloca un punto de interrupción (breakpoint) de donde comenzara a verificar paso a paso, si así lo desea, para confirmar la correcta ejecución del programa.

¿Qué es un “Breakpoint”?

Es un punto de interrupción en el programa, esta indica la línea específica donde va a ser interrumpido el programa para poder debuggearlo.

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

Step Into: Significa que se ejecutará instrucción por instrucción, pero si el debugger encuentra una subrutina. Al pulsar Step Into se irá a la primera instrucción de dicha función, entrará en ella.

Step Over: Significa que se ejecutará instrucción por instrucción, pero si el debugger encuentra una subrutina. Al pulsar Step Over se irá a la siguiente instrucción del código sin entrar en la subrutina.

Step Out: Ejecuta todas las instrucciones de la subrutina en la que se encuentre y te saca de ella a la rutina superior.

## Manual del Programador

El programa comienza generando un tablero, el cual es un arreglo bidimensional de longitud fija de 20 de largo y 80 de ancho. En este tablero, se le asigna un valor de “Muerta” en este caso representada con un punto “.”. Posteriormente por consola, se le pide al usuario iniciar el juego de la vida, con la opción de si o no, en caso de que decida que no, se termina el programa, de lo contrario el juego le pide que introduzca por fila y por columna las posición de las células a las cuales el usuario quiere darle vida, una vez termine de cargar células, le muestra por consola el estado inicial del tablero, y cuantas células existen vivas. Luego, se tienen 3 opciones de enteras del 1 al 3, en el cual puede decidir continuar al siguiente turno, reiniciar el juego y volver a introducir las vidas, o terminar el programa. A medida que el usuario elige la primera opción, el tablero se le van asignando valores de vida o muerte dependiendo el estado del arreglo del turno anterior. En caso de que 3 turnos seguidos no se mueva el tablero, se mostrara por pantalla que se encuentra freezado el juego.

## Manual del Usuario

El juego de la vida es en realidad un juego en el cual se avanza a través de turnos, a partir de un estado inicial en el tablero.

El "tablero de juego" es una malla formada por células, que en nuestro caso estará limitado a 20x80. Cada célula tiene hasta 8 células vecinas, que son las que están próximas a ella, incluso en las diagonales. Las células tienen dos estados: están "vivas" o "muertas". Las células van aumentando o bajando en cantidad dependiendo cuantas células vecinas tienen.

Al comenzar, le aparecerá por pantalla el título del juego y si quiere empezar a jugar o no, en caso de que no se cerrara el programa. Después, se le pedirá introducir las posiciones de las células que quiere que tengan vida al inicio del tablero, para esto se le pedirá por fila y por columna un número que representa la posición en el tablero, del 1 al 20 para el caso de las filas y del 1 al 80 para el de las columnas. Puede cargar hasta 1600 células en el tablero ya que este es el máximo que tiene el tablero. Una vez que decida no agregar más, se le mostrará por pantalla el tablero que origino, y la cantidad de células vivas. Posteriormente tendrá tres opciones, la de continuar con el juego, la de reiniciarlo y volver a cargar las posiciones de las células vivas, o la de terminar el juego. Si decide la primera, el juego continuara y dependiendo la cantidad de células vivas alrededor de las células del turno anterior, empezara a tomar vida el juego.

## Informe

A la hora de iniciar con el trabajo practico, debido a la simpleza del programa, se comenzó haciendo en un solo archivo .cpp para probar la funcionalidad del Código en primera instancia. Una vez el programa cumplía con los Requerimientos, se procedió a modularizarlo en distintos archivos de source y de headers. Se realizó un archivo que maneja la lógica de turnos y de la creación de la vida (JuegodelaVida.cpp), un archivo que maneje el estado inicial del tablero (Tablero.cpp) y otro que sirva como validación de input para este último (InputTablero.cpp).

Se comenzó el trabajo practico, haciendo funcionar el arreglo bidimensional fijo, y la actualización de datos en este con su respectiva validación, luego se diseñó la lógica de turnos del juego, y finalmente la lógica de vida/muerte de las células.

La única validación reiterativa, es al introducir un numero para las filas y columnas del tablero, tiene que ser un numero que este entre 1 y 80, en caso de que el usuario intente cualquier otra forma, le pedirá ingresar un numero entre esos valores.

El usuario siempre tendrá 3 opciones después de la carga inicial, continuar, reiniciar y terminar, estas serán tomadas por el programa como los números 1, 2 y 3. En caso de que elija cualquier otro input, se asumirá que quiere continuar.

Al comienzo del juego se le pregunta al usuario si decide comenzar el juego o no, si quiere comenzar tendrá que introducir el valor 's' en mayúscula o minúscula, en caso contrario, se asumirá que no quiere comenzar el juego.

Se genero un struct para almacenar los datos referentes a lo que se tiene que mostrar por pantalla después de cada turno, ya sea promedio de nacimiento o muertes, o cuantas células nacieron o murieron, se decidió de esta forma para probar el lenguaje, y solo pasar una variable a la función que hará el conteo.