Lab 01 virtual: Introducción a C

Nivel 1

1. Objetivo

Este laboratorio busca introducir a los estudiantes en el manejo del lenguaje C, para así conocer la sintaxis y capacidades iniciales del mismo.

2. TicTacToe SEP GAME

2.1. Introducción

TicTacToe, Gato o Tres en línea, es un conocido juego de estrategia en la que se busca completar tres elementos similares en linea, ya sea horizontal, vertical o diagonal. Para ello se trabaja con cruces y circulos, sobre un tablero de 3×3.

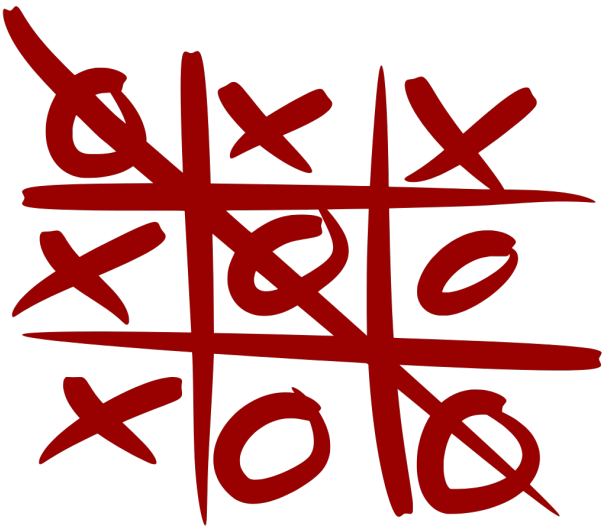

Figura 1: Ejemplo de juego finalizado

Este es un juego cuya cantidad de posibles combinaciones es conocida; además existe una estrategia estudiada que puede ser aplicada tanto por un jugador como por un computador que desee siempre ganar o empatar.

2.2. Actividad a realizar

La actividad de este laboratorio consistirá en implementar una versión de este juego en consola, en la que sea posible una serie de rondas jugar entre dos personas o versus el computador. Un output posible se muestra en la siguiente imagen:

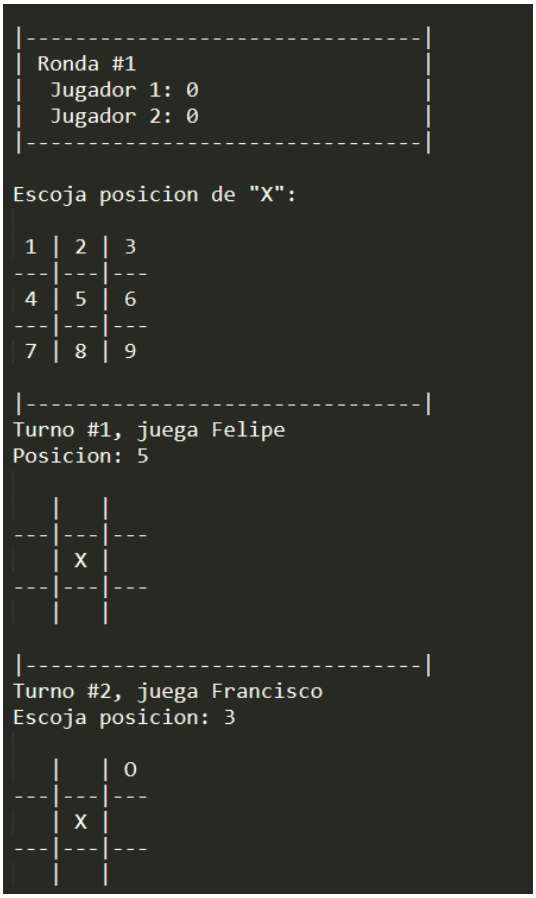

Figura 2: Output posible en consola

2.3. Requisitos

Para lograr la realización de este juego, se deben cumplir con las siguientes acciones.

2.3.1. Task 1: Jugabilidad inicial (50 %)

El juego en su forma inicial, en que juegan 2 jugadores humanos, deberá satisfacer las siguientes especificaciones:

- Contar con una interfaz sencilla, que contenga las instrucciones de cómo jugar, comenzando siempre con cruces.
- Permitir el ingreso de los nombres de los jugadores.
- Poder jugar una ronda de 3 juegos, indicando al final el nombre del jugador ganador o si hubo un empate.
- Se debe ir alternando qué jugador comienza en cada ronda.
- En cada ronda, se debe indicar el jugador que gana.
- **Chequeo de errores**. El programa debe ser robusto y alertar acerca de los siguientes posibles errores.
  - No permitir que los jugadores ingresen el mismo nombre.
  - No permitir sobreescribir una posición ya ocupada.
  - No permitir el ingreso de una posición no existente, pensando en un número o letra que no corresponda a una posición válida.
  - En esencia, con un programa robusto se refiere a que ante entradas que no corresponden, el programa pueda seguir funcionando sin cerrarse abruptamente.
- Se debe agregar una opción de finalizar el juego sin haber terminado todas las rondas. (Rendirse).

2.3.2. Task 2: Incorporar computador (30 %)

En esta actividad, se debe incorporar un computador que sea capaz de jugar el juego de forma aleatoria (nivel fácil) y con un leve nivel de inteligencia (nivel medio). Este debe considerar las siguientes características:

Nivel Fácil

- Mismas condiciones de Task 1.
- El computador debe hacer solo movimientos válidos.
- Se debe ir alternando quién comienza cada ronda.

Nivel Medio

1. Si el computador tiene un movimiento en el que puede ganar, debe ganar.
2. Si el jugador está por ganar, y es turno del computador, el computador debe bloquear su acción.
3. En otras situaciones, se comportará de forma aleatoria.

Esta actividad puede ser realizada en un código aparte, sin embargo, se espera que el programa final sea **realizado solo en 1 código.**

2.3.3. Task 3: Incorporar registro de última partida. (20 %)

Esta actividad corresponde a registrar los movimientos de la última partida realizada. Para esto, al finalizar cada ronda, deberá desplegar la opción:

`¿Desea mostrar la última ronda?`

En la que al responder afirmativamente, deberá replicar los movimientos realizados, independiente si el modo de juego es persona vs. persona o persona vs. computador. Es su deber idear una forma de almacenar esta información a medida que el juego avanza y de eliminarla al comenzar cada ronda, no se permite el uso guardar la información en un archivo externo de cualquier extensión. HINT: Utilizar buffers.

2.3.4. Características comunes a todos los Task

Independiente de lo realizado en los task, se debe lograr una formalidad en su entrega, abarcando los siguientes puntos. El no cumplimiento de esto podría incurrir en un descuento.

- Código ordenado y debidamente comentado. No se exigirá ningún estándar en particular, pero se debe mantener una estructura que permita una lectura fácil. [1] [2]

[1] En https://developer.gnome.org/programming-guidelines/stable/c-coding-style.html.en pueden encontrar ejemplos de cómo mantener un buen formato de código.

[2] En https://codebeautify.org/c-formatter-beautifier pueden reformatear un código.

- Recursos utilizados se deben citar en el README.md de su repositorio.
- Su código debe tener un encabezado como el mostrado en el ejemplo de "Hello World!".
- Luego de cada ronda se debe desplegar las opciones:

    1. Continuar.
    2. Rendirse.
    3. Reiniciar.
    4. Visualizar la última partida. (Solo si realizó el Task 3).

3. Lectura Complementaria

Páginas que hablen de C son útiles.

- C Tutorial https://www.tutorialspoint.com/cprogramming/index.htm
- Learn C Programming https://www.programiz.com/c-programming. En este se sugiere llegar hasta " C Arrays". https://www.programiz.com/c-programming/c-arrays sin llegar a "Multidimensional Arrays".
- GCC: información y explicaciones sobre las opciones de la línea de comando de GCC, el compilador recomendado para utilizar en este laboratorio

4. Requerimientos mínimos para ejecutar

Para poder realizar la programación de la interfaz y de su código deben realizar los pasos indicados dependiendo de su sistema operativo.

4.1. Windows

Los pasos de instalación son los siguientes:

1. Instalar compilador de gcc MinGW http://www.mingw.org/. Link directo de descarga https://osdn.net/projects/mingw/downloads/68260/mingw-get-setup.exe/

2. Es importante agregar gcc al PATH de windows, el siguiente tutorial puede ser útil en caso de que tenga problemas. https://www.youtube.com/watch?v=sXW2VLrQ3Bs

Para ejecutar y compilar se debe abrir una consola de comandos, moverse a la carpeta utilizando el comando `cd`. Por ejemplo:

```
cd /c/users/USUARIO/Documents/SEP/LAB_01
```

Posterior ejecutar el comando

```
gcc -o Hello hello.c
```

Luego, ejecutar:

```
Hello.exe
```

Pueden seguir el tutorial en https://www.tutorialspoint.com/cprogramming/c_program_structure.htm.

4.2. MAC OS

Dado que macOS es un sistema operativo basado en UNIX, el compilador GCC viene por defecto en el terminal por lo cual no requiere software adicional.

4.3. Linux

Abrir el terminal y ejecutar: sudo apt install gcc.

4.4. Compiladores online

Dado que solo se requiere de un compilador, existen opciones online que pueden ser utilizadas, algunas de ellas son:

- **Repl.it** https://repl.it/languages/c
- **Online C Compiler** https://www.onlinegdb.com/online_c_compiler

5. Fecha y forma de de entrega

Se tiene hasta el día lunes 13 de septiembre a las 23:59 hrs.

Deberá subir su **código completo y ordenado** a su repositorio de GitHub junto con un archivo Readme.md que indique las fuentes utilizadas y una lista de lo que no fue implementado en su código. Para ello se le proveerá de una plantilla que deberá rellenar. De forma que el ayudante corrector lo tenga claro. Además debe indicar sistema operativo usado y/o si utilizó un compilador online.

Conectarse en el horario de laboratorio para la realización del control de salida. Lunes 13 de septiembre a las 10:00 hrs.