



Escuela de Computación

IC-4700 Lenguajes de Programación

Profesora: MSc. Samanta Ramijan Carmiol

Proyecto Programado
Programación Imperativa en C

Estudiantes

Emmanuel López Ramírez 2018077125

Luis Esteban Jiménez 2015098300

Rodolfo Cruz Vega 2013235955

Alajuela, I Semestre, 2021

Diseño de la solución

A-) Datos

Para la implementación del programa es necesario el uso constantes, variables y estructuras, las cuales se mencionan a continuación:

Tipo	Nombre	Definición
Int	conexionesMinimax	conexión de cuatro filas para verificar si se ganó
Char	espacioVacio	campo vacío en la impresión del tablero el cual es representado como " "
Char	Tablero[7][7]	tamaño definido para el tablero de juego
Char	Jugador_1	Es la representación del jugador usará las "X" en el tablero de juego
Char	Jugador_2	Es la representación del jugador usará las "O" en el tablero de juego
Int	Filas	Número que representa las filas del tablero
Int	Columnas	Número que representa las columnas del tablero

Al ser un programa en el cual los datos son reutilizables en varias funciones se repite el mismo dato pero con un nombre más significativo acorde a lo que se está realizando en la función, un ejemplo de ello lo encontramos con Filas y Columnas los cuales se les alterno el nombre por dimensión.

Para el juego el dato más relevante es la matriz de caracteres que puede entenderse de la siguiente forma:

Tablero

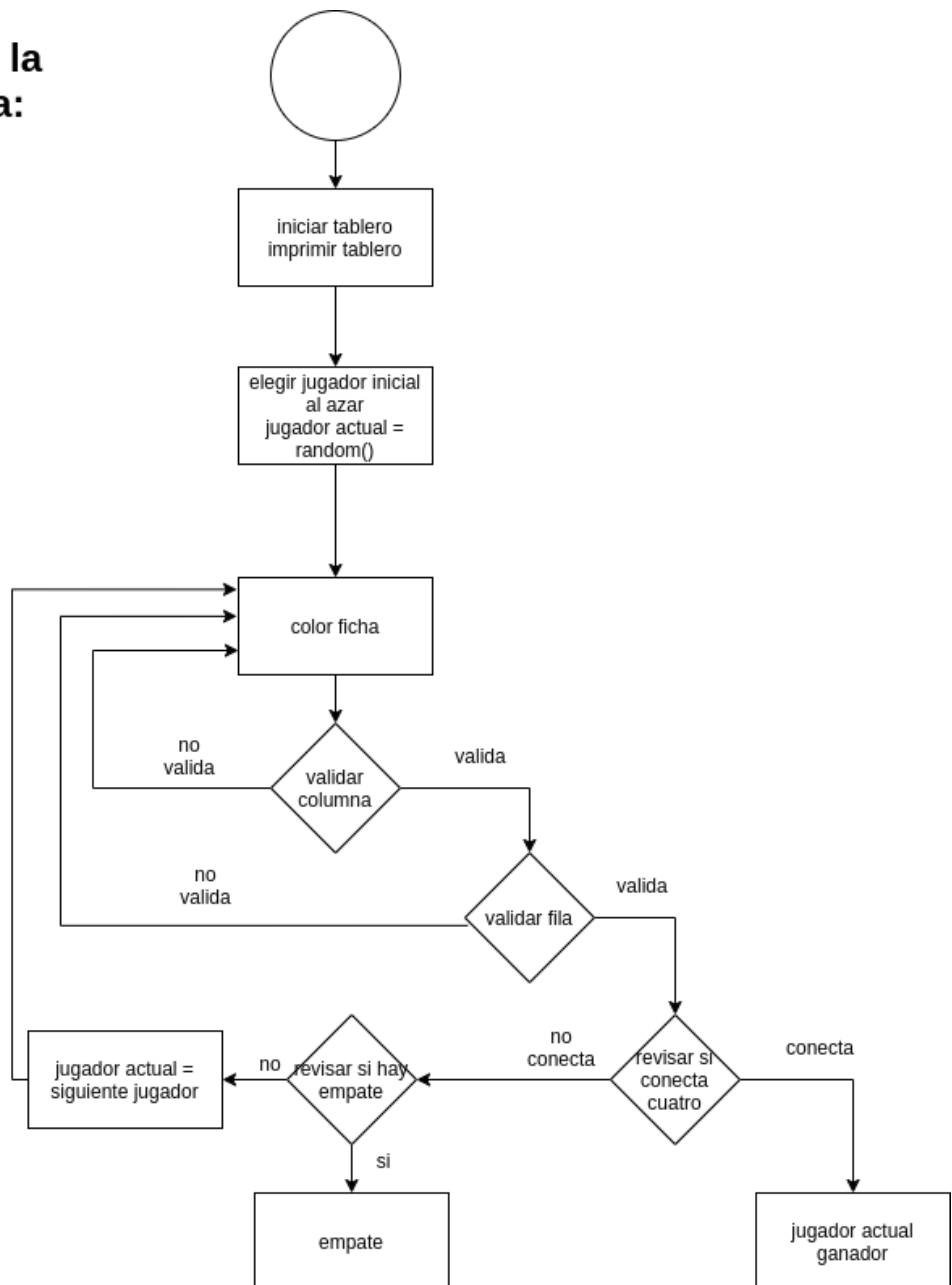
	vacio "	jugador1 O	jugador2 X			

B-) Interacción con la persona usuaria:

La interacción de nuestro programa con el usuario, es por medio de consola, esta pide al usuario un dato de entrada numérico (tipo int), su representación sería de la siguiente forma.

Entradas	Salidas
El usuario selecciona un número el cual representa la columna con la que desea interactuar	<p>Caso 1: Se presenta que la fila esté llena, el programa vuelve a repetirle el turno al usuario.</p> <p>Caso 2: Presenta que la opción escogida es una opción inválida, el programa vuelve a repetirle el turno al usuario.</p> <p>Caso 3: Imprime el tablero con la ficha en la casilla seleccionada.</p>

Interacción con la persona usuaria:



C-) Descripción de las estructuras de datos y algoritmos

La estructura utilizada para generar el tablero es una matriz de caracteres de tamaño 7x7 el cual es llenado con un espacio entre unas comillas.

El algoritmo para la inteligencia artificial que jugará como computadora es minimax, el cual evaluará jugadas alternas del jugador y de la computadora con en el fin de obtener la jugada definitiva que le permita ganar, este verificará con respecto a jugadas anteriores cual le puede beneficiar, en caso de que le beneficie colocar una pieza para ganar o generar el comienzo de una fila éste será una jugada max, en caso que ocupe defenderse de tres o dos fichas del jugador este bloqueara colocando la ficha donde bloqueara la jugada, este será una opción mini.

El algoritmo principal que busca las jugadas del computador y las del jugador es llamado jugada minimax, este busca fichas seguidas hacia arriba, horizontal o en diagonal, de estos

resultados se utilizara el de mayor puntaje el cual es quien tenga más conexión de fichas, esta información provee la posible jugada del jugador y de la computadora, en caso de que haya posibilidad de que el jugador gane, este dará la posición de la columna donde se completa las cuatro conexiones y hace el bloqueo, en caso de que la jugada de la computadora sea más eficaz que la defensa o sea el gane del juego, este escoge seguir la jugada planeada o ganar.

Las estructuras que usamos en la parte del tablero se basan en:

```
int realizarJugada(char jugador, int columna, char tablero[dimension][dimension]) {
```

Es el encargado de verificar si el usuario selecciono una columna inválida o una columna llena, si es el caso se repite el turno del jugador, caso contrario el programa se encarga de colocar la ficha de dicho jugador en el primer espacio en blanco que encuentre de la columna.

```
int verificarFilaVacía(int columna, char tablero[dimension][dimension]) {
```

Esta función es una anexo de la anterior , ya que es usada por la función de arriba, esta se encarga de validar que haya un espacio en blanco en la fila, si no es así la función devuelve un -1 que tomaremos para validar que la columna está llena.

```
int verificarTableroLleno(char tablero[dimension][dimension]) {
```

Esta función está asociada con la función de arriba ya que esta es un recorrido de columnas usando verificarFilaVacía. Si una columna está llena esta función pasa a la siguiente hasta verificar el último espacio del tablero.

```
void borrarTablero(char tablero[dimension][dimension]) {
```

```
int imprimirTablero(char tablero[dimension][dimension]) {
```

Estas dos últimas son como su nombre lo dice imprimir el estado actual del tablero y el otro se encarga de crear un tablero nuevo.

2. Lecciones aprendidas:

Con el desarrollo de este juego se logró comprender cómo una inteligencia artificial se puede desarrollar para tareas en específico como lo son crear un oponente para juegos de mesa como lo son tres en raya, ajedrez, damas o tablero.

El uso de una matriz de caracteres limita crear condiciones en las que es necesario validar que no se jueguen campos fuera de la matriz ya que esos índices superiores no crean en compilación pero sí en tiempo de ejecución.

La ejecución recursiva requiere de un manejo de memoria importante especialmente en recorridos en profundidad donde la cantidad de nodos se incrementa de forma rápida.