



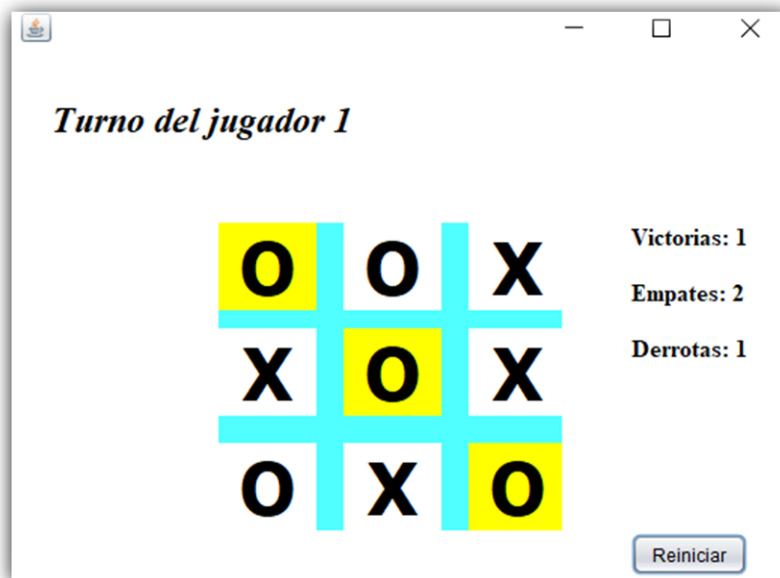
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ciencias Físico-Matemáticas



Lic. en Ciencias computacionales

Laboratorio de Programación Orientada a Objetos

Tema: Producto Integrador de Aprendizaje



Nombre del maestro: José Luis Candelario Tovar

Nombre Del alumno: Raymundo Garza Garza

Matrícula: 1960730

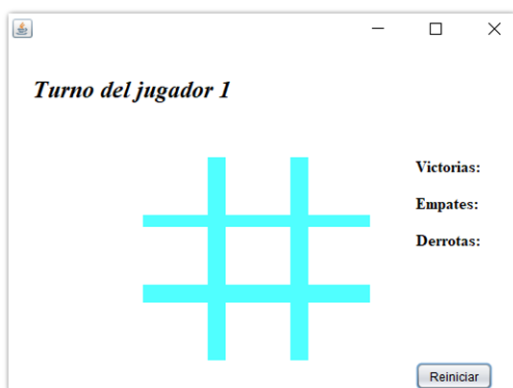
02/11/2021

Contenido

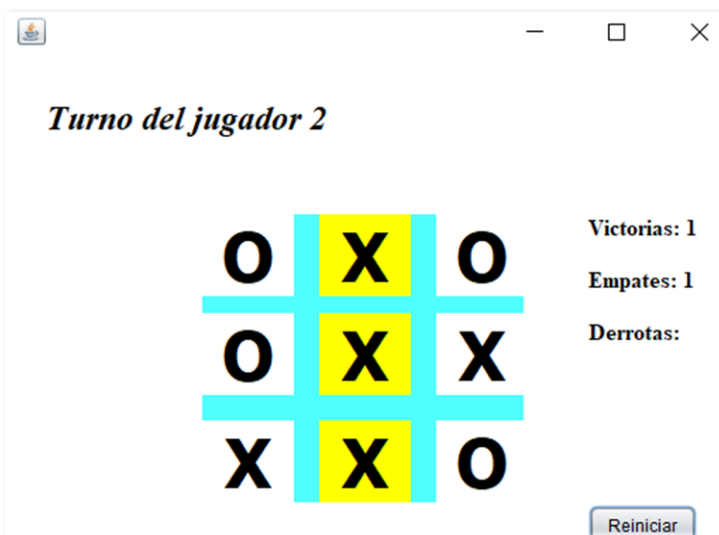
Introducción	3
Detalles del programa	4
Modo un jugador:	5
Respecto a la programación del programa.....	6
Creación del Bot.....	7

Introducción

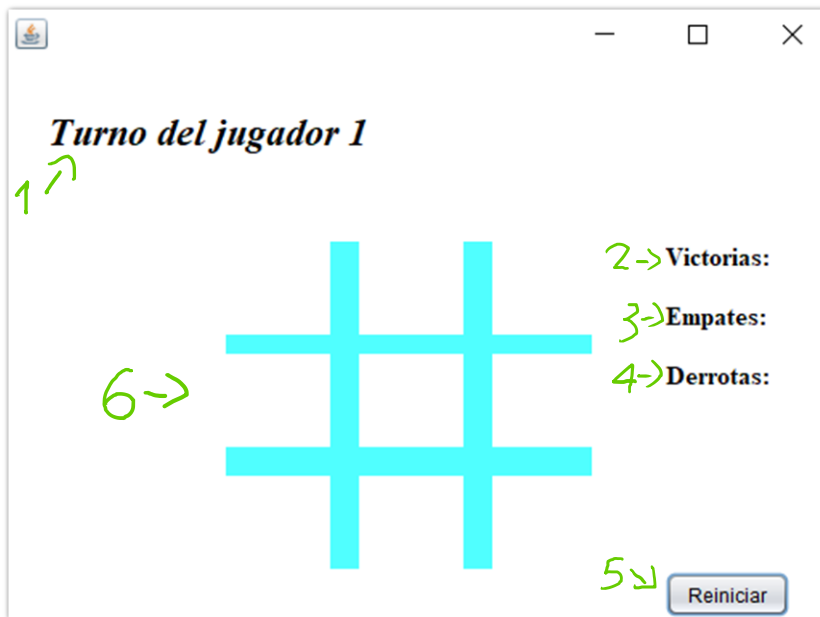
Este programa es una representación del clásico videojuego del 3 en raya (también conocido como tic-tac-toe). En él, es posible jugar solo o con 2 jugadores. La manera de jugar es muy sencilla:



Al inicio del programa, se muestra una especie de #, dentro de cada espacio del signo, es posible poner una X, después de hacer el movimiento, se puede volver a seleccionar un espacio diferente donde se pondrá una O y así sucesivamente intercalándose entre X y O. Se dice que el jugador 1 es el jugador que controla los lugares donde estará el símbolo X, mientras que el jugador 2 es el que controla el símbolo O. Gana el jugador que consiga poner 3 de sus símbolos de manera seguida, ya sea en fila, columna, o diagonal:



Detalles del programa

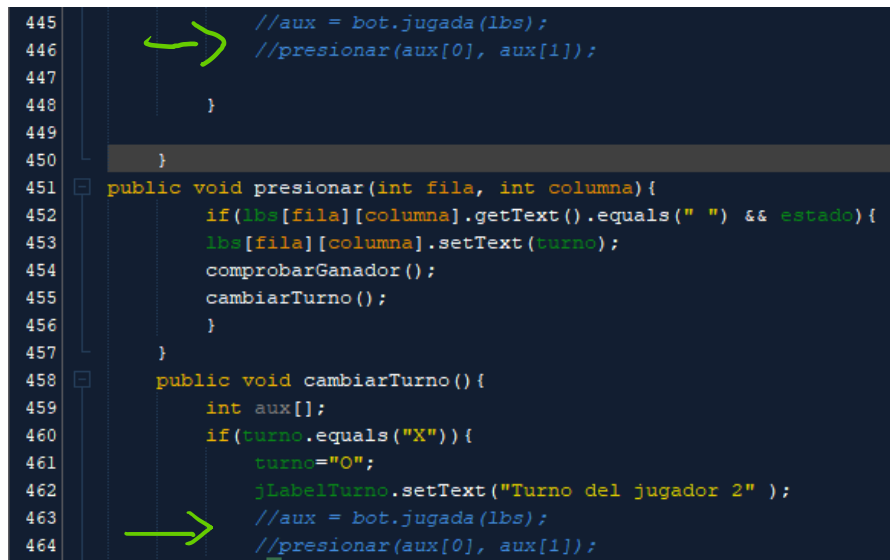


- 1- Muestra el turno del jugador
- 2- Muestra el número de victorias a lo largo del programa
- 3- Muestra el número de empates a lo largo del programa
- 4- Muestra el número de derrotas a lo largo del programa
- 5- Es un botón que al presionarlo reinicia el estado del juego
- 6- Es la parte principal del programa, ahí se hacen los movimientos del juego

Modo un jugador:

Este programa tiene implementado un Bot que puede entender el juego y participar en él, la habilidad del Bot en este juego es bastante avanzada y definitivamente te tomará más de un intento poder vencerlo.

Para activar el modo un jugador solamente es necesario remover las barras de las líneas de código 445, 446, 463 y 464 de la clase interfaz:



```
445 //aux = bot.jugada(lbs);
446 //presionar(aux[0], aux[1]);
447
448 }
449
450 }
451 public void presionar(int fila, int columna){
452     if(lbs[fila][columna].getText().equals(" ") && estado){
453         lbs[fila][columna].setText(turno);
454         comprobarGanador();
455         cambiarTurno();
456     }
457 }
458 public void cambiarTurno(){
459     int aux[];
460     if(turno.equals("X")){
461         turno="O";
462         jLabelTurno.setText("Turno del jugador 2");
463         //aux = bot.jugada(lbs);
464         //presionar(aux[0], aux[1]);
```

Después de realizar eso, ahora el jugador 2 será controlado por un Bot y podrás jugar contra él.

Respecto a la programación del programa

Lo más fundamental para entender el funcionamiento del programa, es saber que cada cuadro dentro de la almohadilla tiene una etiqueta, y cuando presionamos esa etiqueta, el texto dentro de ella se cambia. La forma para navegar entre cada etiqueta es guardar todas estas en un arreglo y asignarles una coordenada:

```
//Fila Columna
lbs[0][0] = jLabel1;
lbs[0][1] = jLabel2;
lbs[0][2] = jLabel4;
lbs[1][0] = jLabel5;
lbs[1][1] = jLabel6;
lbs[1][2] = jLabel7;
lbs[2][0] = jLabel11;
lbs[2][1] = jLabel12;
lbs[2][2] = jLabel13;
```

Diagrama de una almohadilla 3x3 con coordenadas de fila y columna:

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

También existe una función para presionar la etiqueta, para llamarla, es necesario ingresarle la coordenada de la etiqueta que se quiere presionar:

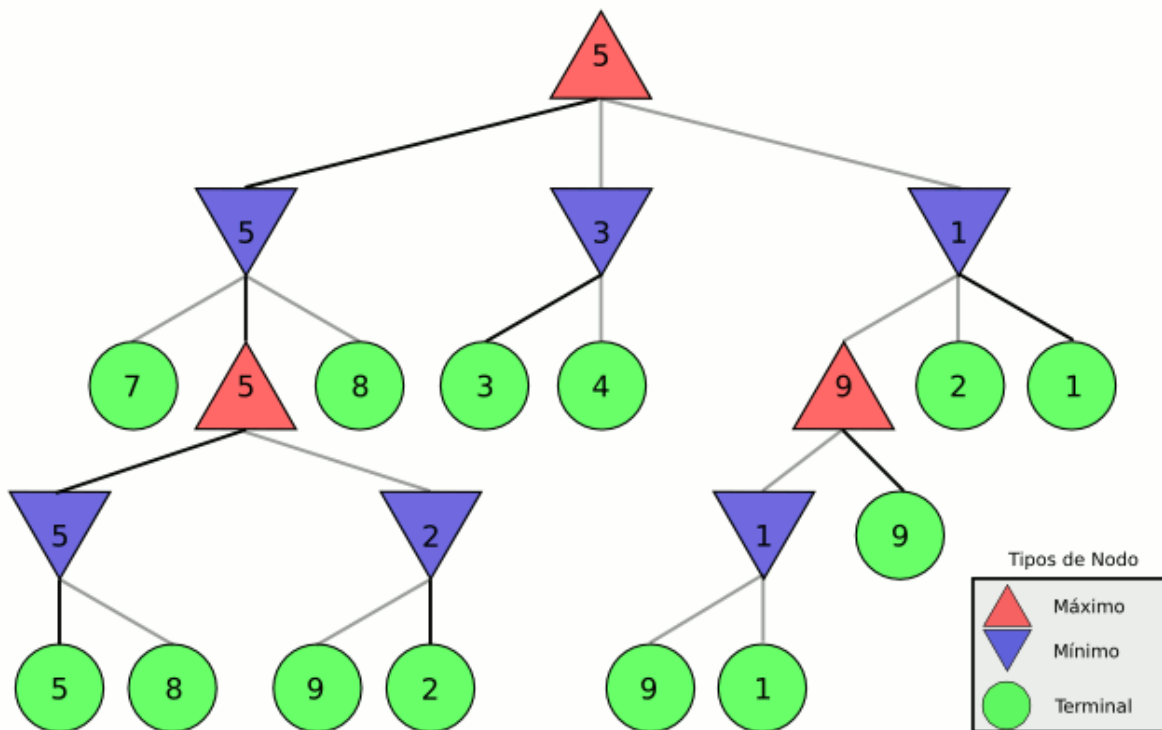
```
public void presionar(int fila, int columna){
    if(lbs[fila][columna].getText().equals(" ") && estado){
        lbs[fila][columna].setText(turno);
        comprobarGanador();
        cambiarTurno();
    }
}
```

Creación del Bot

Para crear el Bot, es necesario saber sobre teoría de juegos, pues la base del algoritmo del Bot es un método conocido como MiniMax.

En teoría de juegos, MiniMax es un método de decisión para **minimizar la pérdida máxima** esperada en juegos con adversario y con información perfecta. MiniMax es un **algoritmo recursivo**.

El funcionamiento de MiniMax puede resumirse en cómo elegir el mejor movimiento para ti mismo suponiendo que tu contrincante escogerá el peor para ti.



Función Mínima:

- El valor inicial es 0
- Al valor se le resta 1 por cada carril sin movimientos míos
- Al valor se le resta 1 por cada carril con un movimiento del oponente (Solo si no hay un movimiento mío en el carril)
- Si un carril tiene 2 movimientos del oponente, entonces al valor se le resta 10

Ejemplo con turno del jugador O:

X	-4	-4
-4	X	-2
-4	-2	O

Función Máxima:

- El valor inicial es 0
- Al valor se le suma 1 por cada carril sin movimientos del oponente
- Al valor se le suma 1 por cada carril con un movimiento mío (Solo si no hay un movimiento del oponente en el carril)
- Si un carril tiene 2 movimientos míos, entonces al valor se le suma 10

Ejemplo con turno del jugador O:

X	O	2
0	X	2
2	2	O

Función de utilidad

La función de utilidad se obtiene restandole la función mínima a la función máxima.

Ejemplo con turno del jugador O:

X	4	6
4	X	4
6	4	O

Finalmente, la mejor jugada es aquella que tenga un mayor valor, en este caso, tanto (0,2) como (2,0) son las mejores jugadas, y ambas son igual de buenas, así que se puede elegir la que sea.

