Laboratorio 4

Ing. Msc. Víctor Orozco

22 de febrero de 2023

1. DESCRIPCIÓN

El objetivo del laboratorio es explorar la implementación del algoritmo Minimax para el problema TicTacToe.

2. CONTENIDO PREVIO

Para iniciar la resolución por favor utilice la siguiente tarea de GitHub Classroom:

`https://classroom.github.com/a/WK0MaeeC`

3. LABORATORIO

3.1. INSTRUCCIONES GENERALES

Para el laboratorio y tareas de esta semana usted debe crear las clases *com.vorozco.totito.Minimax* (elaborada en el material previo), *com.vorozco.totito.Expectimax* (objetivo en clase) y *com.vorozco.totito.Automax* (tarea), las cuales deben implementar los siguientes métodos para al menos compilar:

Minimax:

```
public static<Move> double minimax(Board<Move> board, boolean maximizing,
Piece originalPlayer, int maxDepth)

public static <Move> Move findBestMove(Board<Move> board, int maxDepth)
```

Expectimax:

```
public static<Move> double minimax(Board<Move> board, boolean maximizing,
Piece originalPlayer, int maxDepth)

public static <Move> Move findBestMove(Board<Move> board, int maxDepth)
```

Automax:

```
public String doGame(int startposition, TotitoPiece piece){
```

Recuerde que como primer paso debe implementar AL MENOS, las clases con estos metodos vacios -e.g. retornando valores demostrativos- para que puedan ejecutarse las pruebas en GitHub classroom.

3.2. Evaluación en clase (40 % del total de este laboratorio)

Como primer paso compruebe que todos los tests de la clase MinimaxTest funcionan correctamente con su implementación elaborada previamente (40 puntos).

En la clase *com.vorozco.totito.ExpectimaxTest* observara un test que simula 100 veces sobre el mismo tablero con una profundidad de búsqueda 2. El problema se ha configurado de tal forma que el tablero asuma como último movimiento el jugador *O*, y siendo así, si el siguiente jugador *X* jugara de forma perfecta/adversaria (es decir minimizando), obtendría siempre como resultado perfecto 6 (como referencia observe el test *piezaInversa* en *MinimaxTest*).

Modifique su implementación de la clase *Expectimax*, para que los tres movimientos posibles en el tablero (4, 6, 7) tengan la misma probabilidad de ser seleccionados, es decir que el adversario falle en seleccionar 6 un 66 % de veces. Como sugerencia copie de forma integra su implementación de Minimax y analice el comportamiento del minimizador (60 puntos).

Fecha límite de entrega: Viernes 24 de febrero, 23:55 SIN EXCEPCIONES

3.3. Tarea (60 % del total de este laboratorio)

Para la resolución de su tarea por favor utilice la siguiente tarea de GitHub Classroom:

`https://classroom.github.com/a/0Gaf1gYh`

El problema Minimax para TicTacToe se considera resuelto por lo que al establecer el primer movimiento en un juego determinado, se conoce el resultado si dos adversarios perfectos se enfrentan.

El objetivo de esta tarea es que usted implemente una clase denominada *Automax* cuyo único método *doGame* sea capaz de ejecutar Minimax para ambos jugadores de tal forma que

pueda demostrar los juegos perfectos para *O* y *X*.

Fecha límite de entrega: Domingo 26 de febrero, 23:55 SIN EXCEPCIONES