ARTIFICIAL INTELLIGENCE PROJECT

CONNECT FOUR

Kamal Atluri
Yasasvi Yeleswarapu
Leeladhar Reddy Munnangi

Game Rules

- Consists of Eight-column, Six-row vertically suspended grid.
- Players starts the game by dropping colored discs from the top.
- The pieces fall straight down, occupying the next available space within the column.

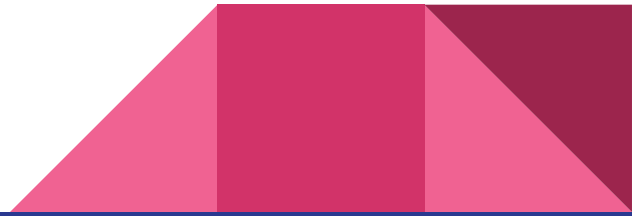
Objective

- Connect four of one's own discs of the same color next to each other

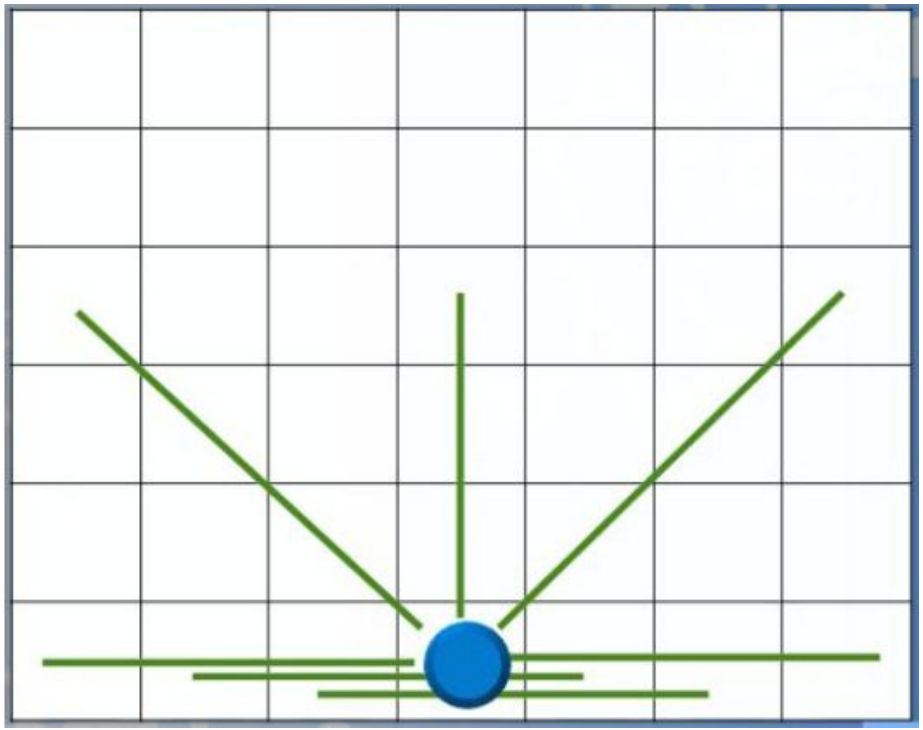
- Vertically
- Horizontally
- Diagonally

Four Different Heuristics

- Number of winning lines
- How many pieces in a row will i get if I move here
- Do I block a good path for the enemy if I move here ?
- Does the space above give the opponent an advantage ?

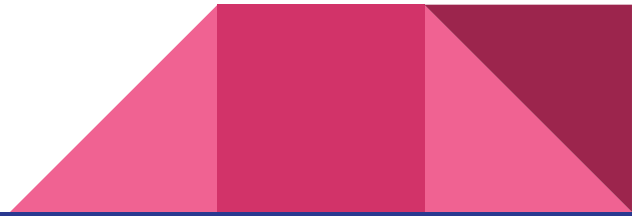


Position for max possibilities



Implementation

- Game board with size 8x6 which gives 48 slots for the players.
- Implementing in Java.
- Algorithms : Min-Max , Alpha-Beta Pruning.
- Mode : Human VS computer.
- Number of moves.
- Time taken by computer to make a move(in Milliseconds).



Game Board

```
/** Creates new C4Board */
public C4Board(Player firstPlayer, Player secondPlayer)
{
    initSlots();

    //create the moves arrays, these are all the possible moves
    firstPlayerMoves = new Move[NUMBER_OF_COLUMNS];
    secondPlayerMoves = new Move[NUMBER_OF_COLUMNS];

    for(int i = 0; i < NUMBER_OF_COLUMNS; i++)
    {
        firstPlayerMoves[i] = new C4Move(firstPlayer, i);
        secondPlayerMoves[i] = new C4Move(secondPlayer, i);
    }



}

/**
 * Create the slots and organize them into rows
 */
private void initSlots()
```

Tracking the mouse

```
public boolean mouseMove(Event evt,int x, int y)
{
     if((BOARD_TOP_X < x) & (x < (BOARD_TOP_X + BOARD_WIDTH)))
     {
          if((TIP_TOP_Y < y) & (y < (BOARD_TOP_Y + BOARD_HEIGHT)))
          {
               x = x - BOARD_TOP_X - 10;
               int column = x /COLUMN_WIDTH;
               if(column >= C4Board.NUMBER_OF_COLUMNS)
               {
                    column = -1;
               }

               setBlackColumn(column);
          }
     }
     else
     {
          setBlackColumn(-1);
     }

     return true;
}
```

Applet Code

```
//-------------------------------------------------
    //instance variables
    private Button restart;
    private Choice levels;
    private ImagePanel imagePanel;

    private MinimaxPlayer computer;
    private AsynchronousPlayer human;
    private C4Board board;
    private GameMaster gameMaster;

//-------------------------------------------------
    //instance methods

    public void init ()
    {
        System.out.println("C4Applet initializing");
        System.out.println(ABOUT);

        Image blackPieceImage = null;
        Image redPieceImage = null;
        Image boardImage = null;

        //load the images
```

Game Event Listener

```
public class DefaultGameEventListener implements GameEventListener {

    /**
     * The game has started.
     */
    public void gameStarted() {}

    /**
     * The game has stopped
     */
    public void gameStoped() {}

    /**
     * The game has been restarted.
     */
    public void gameRestarted() {}

    /**
     * A player has moved.
     */
    public void moveMade(Move aMove) {}
```

MiniMax Code

```
public class MinimaxPlayer extends DefaultPlayer
{


    //the number of levels minimax will look ahead
    private int depth = 1;
    private Player minPlayer;


  /** Creates new MinimaxPlayer */
    public MinimaxPlayer(String name, int number, Player minPlayer)
    {
        super(name, number);

        this.minPlayer = minPlayer;
    }

    /**
     * Get the number of levels that the Minimax Player is currently looking
     * ahead.
     */
    public int getDepth()
    {
        return depth;
```

Game over

```
public void gameStoped()
{
    if(board.isGameOver())
    {
        offscreenGraphics.setFont(TEXT_FONT);
        offscreenGraphics.setColor(Color.black);
        if(board.getBoardStats().getScore(human) != 0)
        {
            offscreenGraphics.drawString("You win", TEXT_TOP_X, TEXT_TOP_Y);
        }
        else if(board.getBoardStats().getScore(computer) != 0)
        {
            offscreenGraphics.drawString("You lose", TEXT_TOP_X, TEXT_TOP_Y);
        }
        else
        {
            offscreenGraphics.drawString("Tie game", TEXT_TOP_X, TEXT_TOP_Y);
        }
        update();
    }

}
```


THANK YOU