Arcade Game Box

Name: Krushan Bauva

Class: X

Division: C

Roll Number: 10

Year: 2014-2015

Lady Ratanbai & Sir Mathurdas Vissanji Academy
Address: Dr. S. Radhakrishnan Marg, Off Old Nagardas Road, Andheri (E), Mumbai- 400 069

This page is intentionally left blank

Acknowledgement

I would like to express my special thanks of gratitude to my computer teacher, Mr. Mahendra Kanojia, and my school, Lady Ratanbai & Sir Mathurdas Vissanji Academy, which gave me the golden opportunity to do this wonderful project on the topic,Arcade Game Box, which also helped me in doing a lot of research and where I also came to know about many new things. I am really thankful to them.

I also take this opportunity to extend my deep appreciation to my family and my friends for all that they meant to me during the crucial times of this wonderful project.

Preface

Stop wasting paper. This Arcade Game Box is specially designed to stop the wastage of paper as well as to test your skills and teaching strategy to kids. You can play either of the two classic professional games (Tic tac toe or four in a row).

Tic tac toe is also known as ‘noughts’ and ‘crosses’. Here the player has to occupy three positions in a row in order to claim victory.

Four in a row is also a very famous game which you must have definitely played in your childhood. Here also the player has to occupy four positions in a row in order to claim victory but here he cannot choose the row number. The character falls down to the lowest position on its own.

This software allows you to compete against another human player on the same computer.

This software does not include artificial intelligence neither does it allow you to compete against a human player on another computer.

Index

Software and Hardware Requirements

Software

OS: Windows XP / 7 / 8 / Vista

Language: Java

JDK version 1.7.0_71 build 14

JRE version 1.7.0_71 build 14

IDE: Bluej version 3.1.4

Back End: -

Hardware

RAM: 256 MB

Processor: 1GHz

Storage: -

Mouse: No

Keyboard: Yes

Monitor: Yes

Description

- You have to choose any of the two games either Tic Tac Toe or Four in a Row whichever you wish to play.
- Both the games are excellent for testing your skills or teaching strategy to kids.
- You have to then enter the name of two players.
- Both the games are two-player game where Player 1 plays against Player 2.

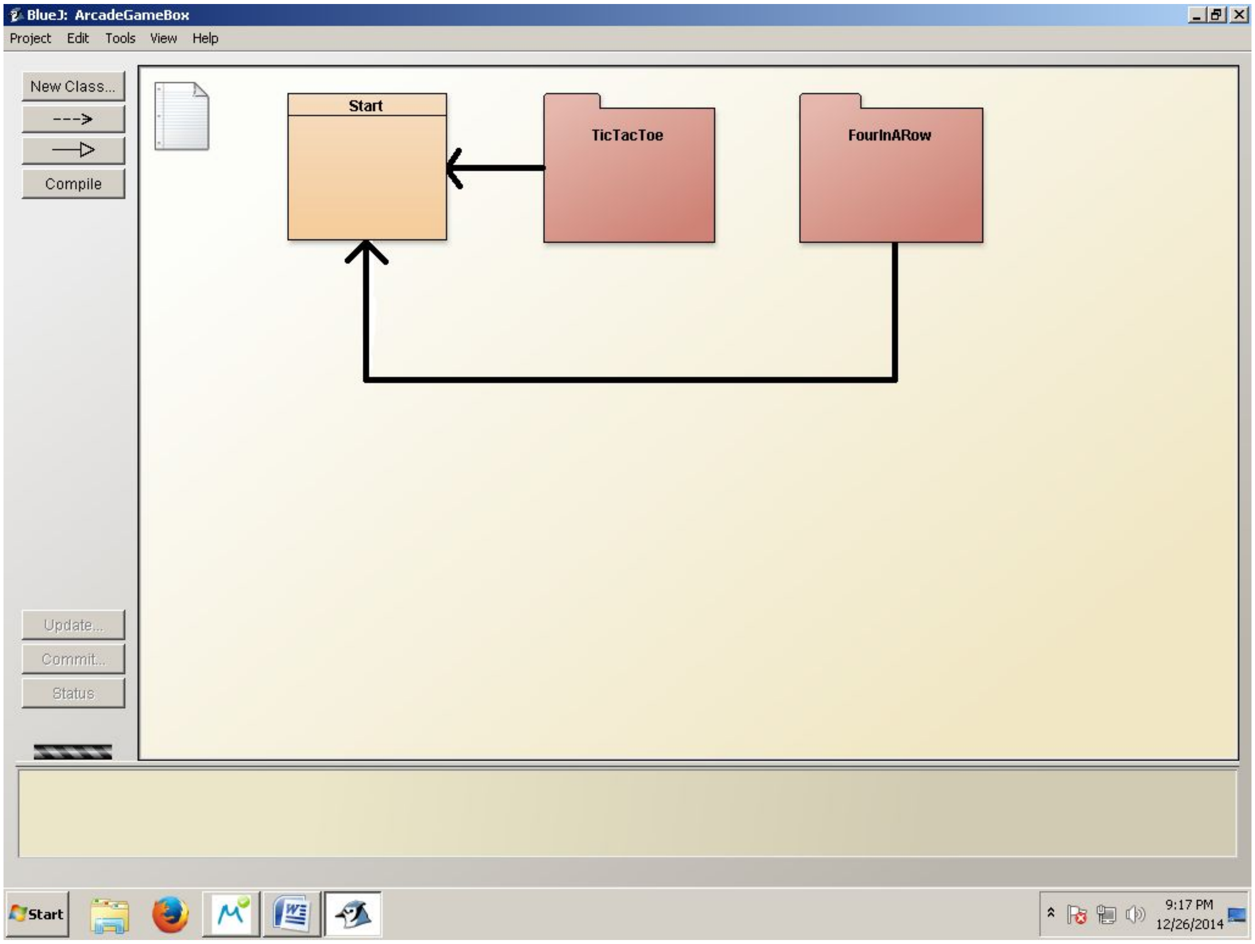
Tic Tac Toe

- Tic Tac Toe is a classic professional game also known as noughts or crosses.
- Here Player 1 is ‘X’ and Player 2 is ‘O’.
- A board will be shown with numbers from 1 to 9.
- Player 1 chooses the position where he/she wants to put his/her ‘X’ by entering the number of the position.
- Then Player 2 chooses his/her position in the same way.
- Then again Player 1 chooses his/her position and this continues till a player wins.
- Winner: If a player occupies 3 positions in a row (i.e. either vertically or horizontally or slanting).
- Stalemate (Draw): If all the positions are filled but no player occupies 3 positions in a row.

Four in a Row

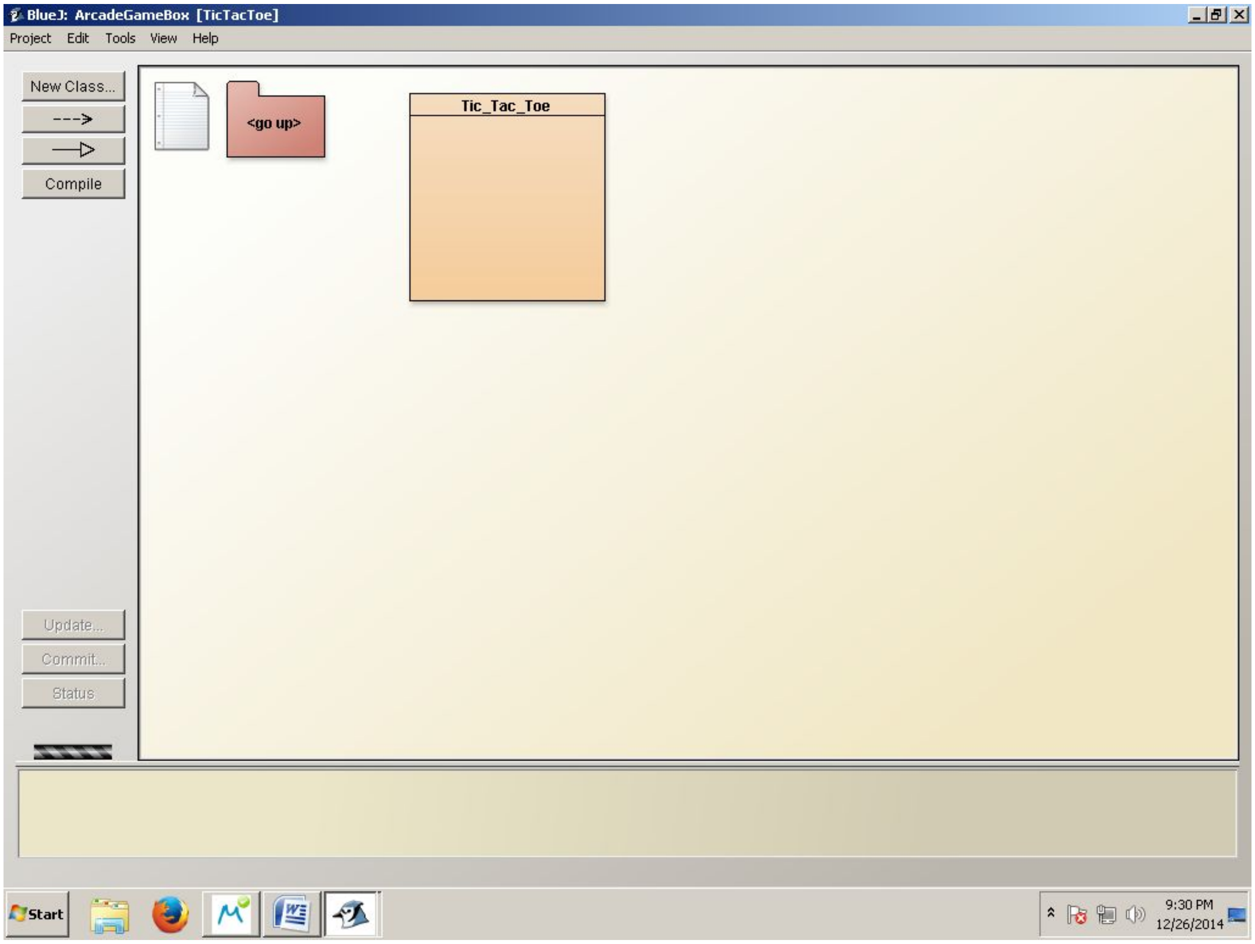
- Four in a Row is also a classic professional game.
- Here Player 1 is ‘$’ and Player 2 is ‘#’.
- A board will be shown with column numbers from 1 to 8 at their top.
- Player 1 chooses the column in which he wants to drop his/her character by entering the column number.
- The character will fall down to the lowest position.
- Then Player 2 chooses his/her column in the same way.
- Then again Player 1 chooses his/her column and this continues till a player wins.
- Winner: If a player occupies 4 positions in a row (i.e. either vertically or horizontally or slanting).
- Stalemate (Draw): If all the positions are filled but no player occupies 4 positions in a row.

Package Diagram



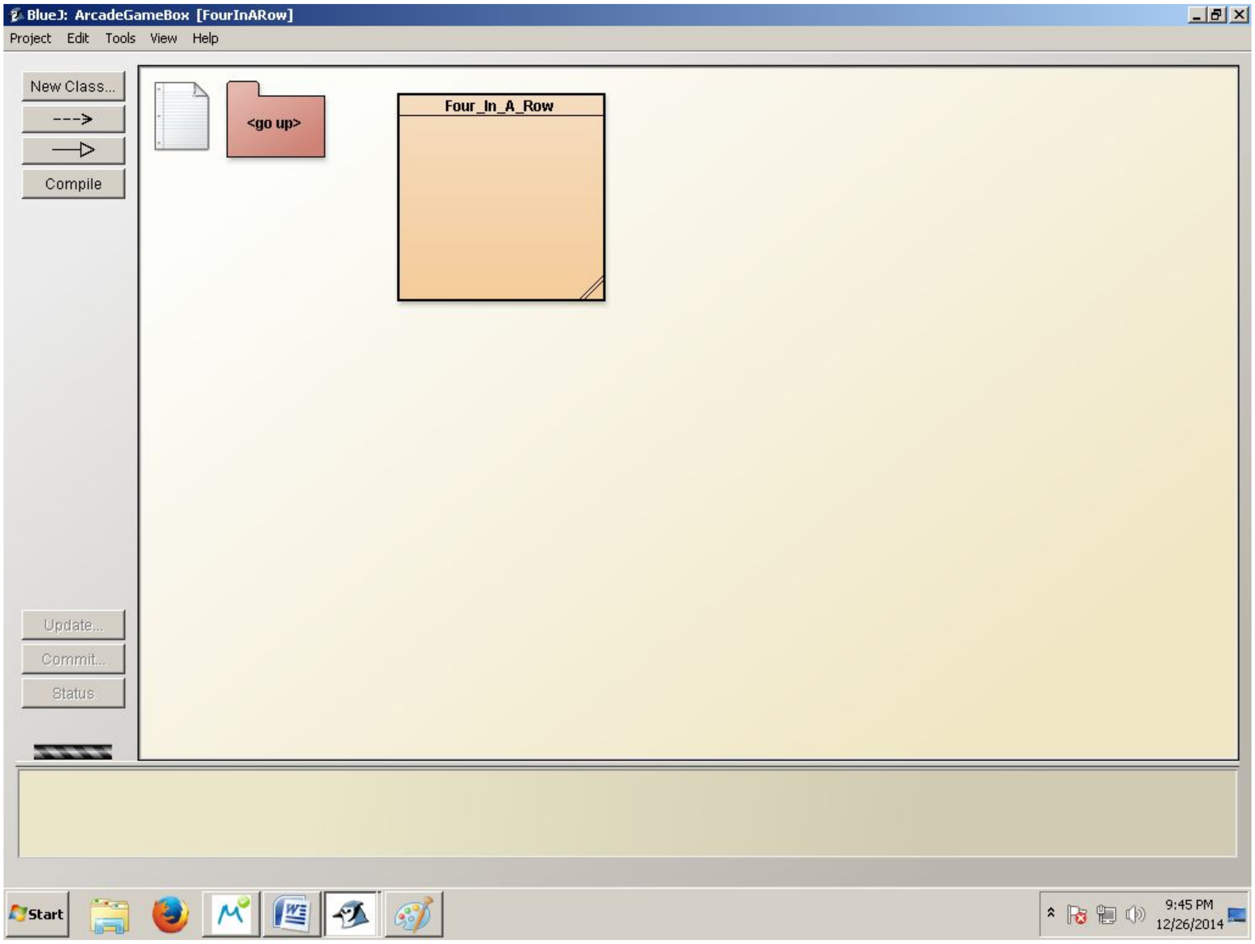
Class Diagram

Package Name: TicTacToe



Class Diagram

Package Name: TicTacToe



Method Diagram

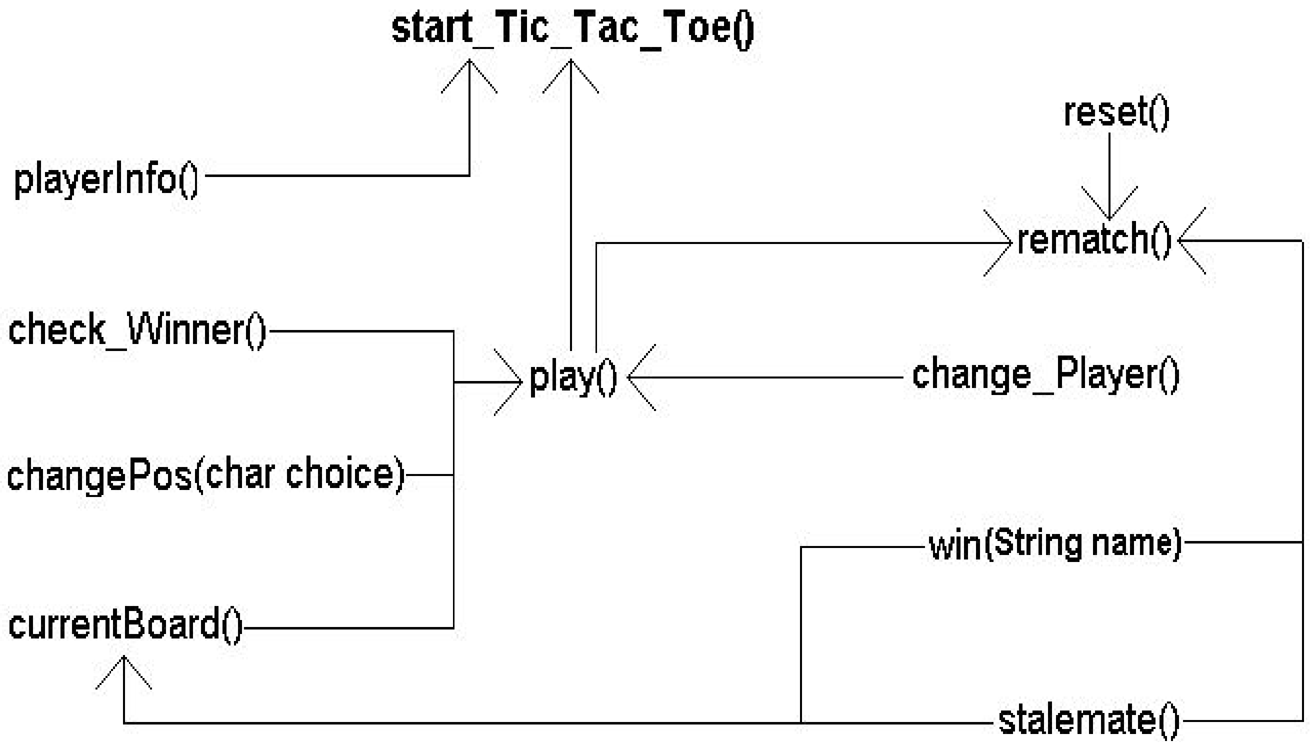
Class Name: Start

main(String[] args)

Method Diagram

Package Name: TicTacToe

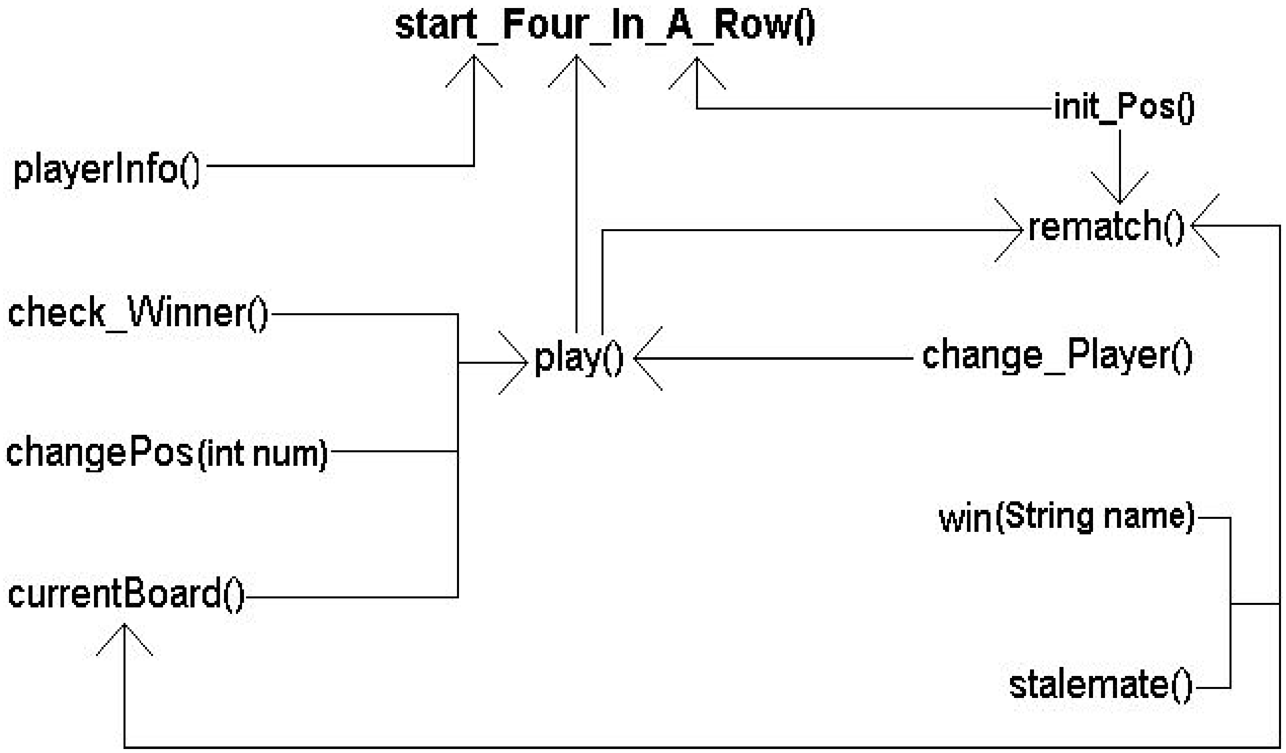
Class Name: Tic_Tac_Toe



Method Diagram

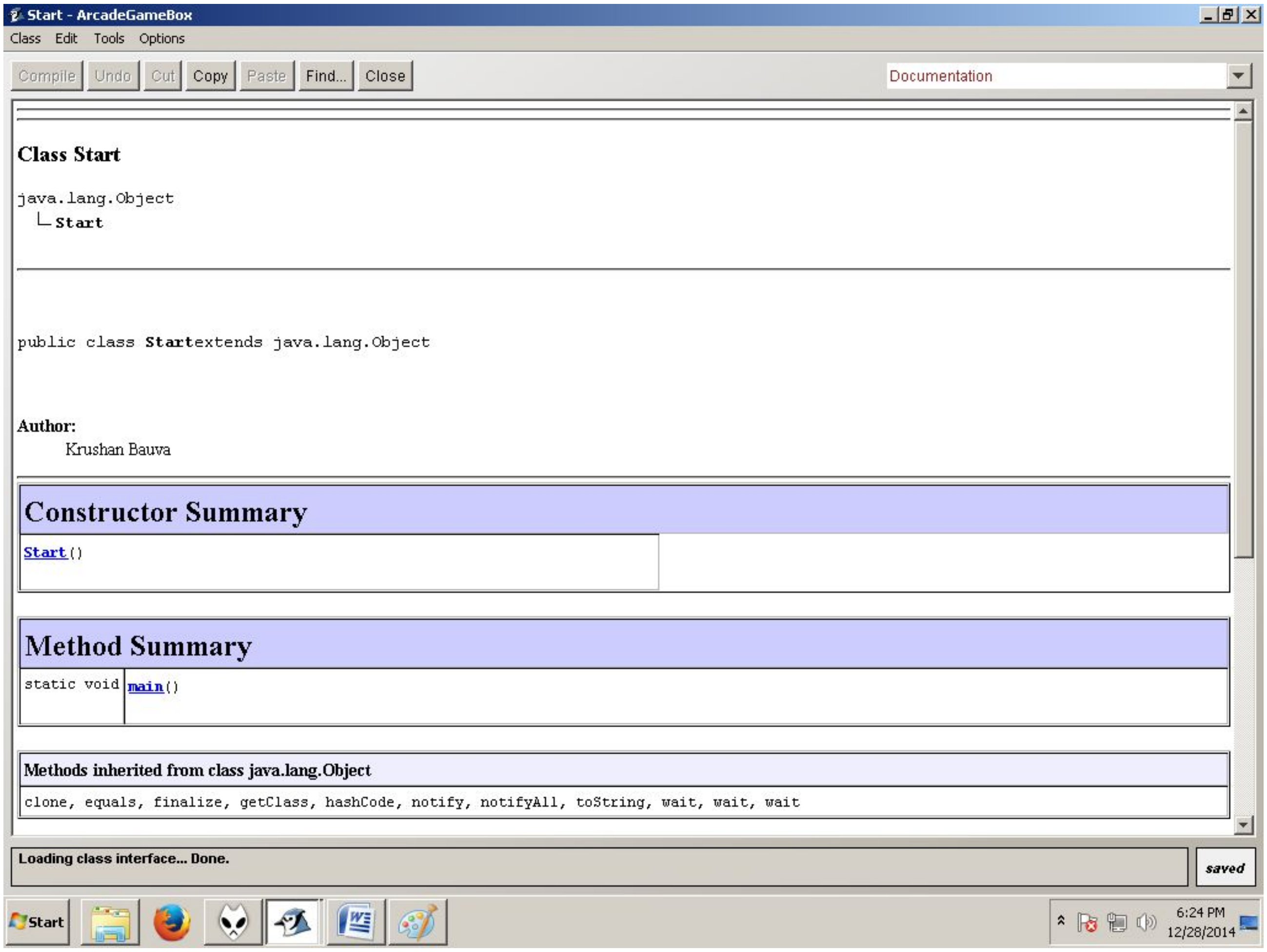
Package Name: FourInARow

Class Name: Four_In_A_Row



Project Documentation

Class Name: Start



Class Start

```
java.lang.Object
  └ Start
```

```
public class Startextends java.lang.Object
```

Author:
Krushan Bauva

Constructor Summary

Start()

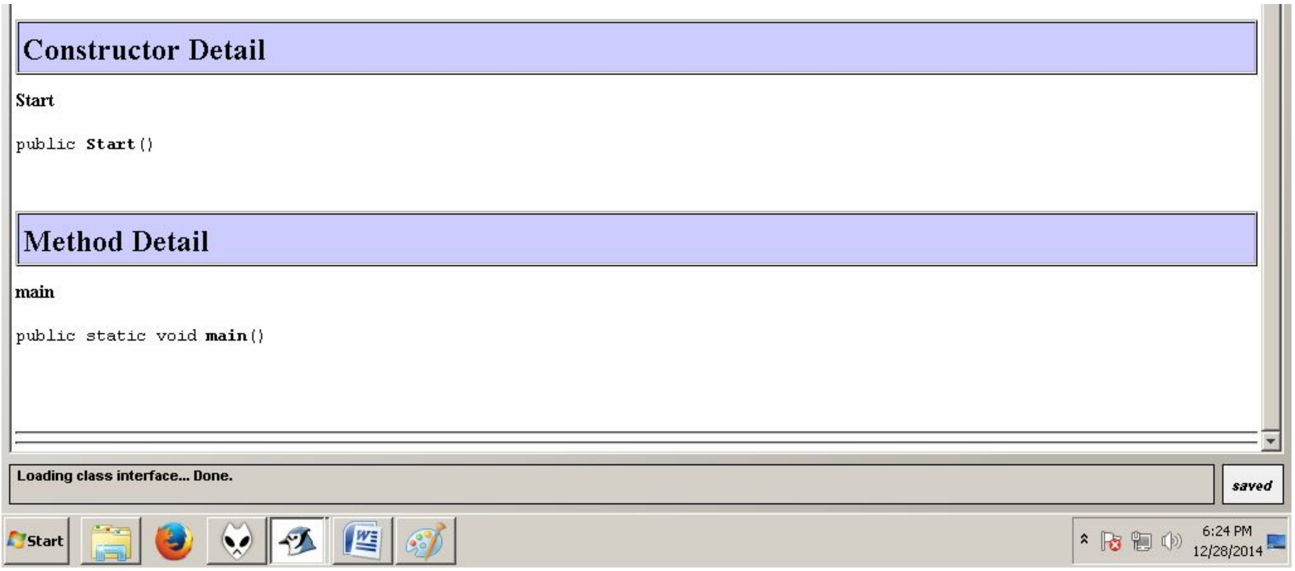
Method Summary

static void	main()

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Loading class interface... Done.

saved



Constructor Detail

Start

```
public Start()
```

Method Detail

main

```
public static void main()
```

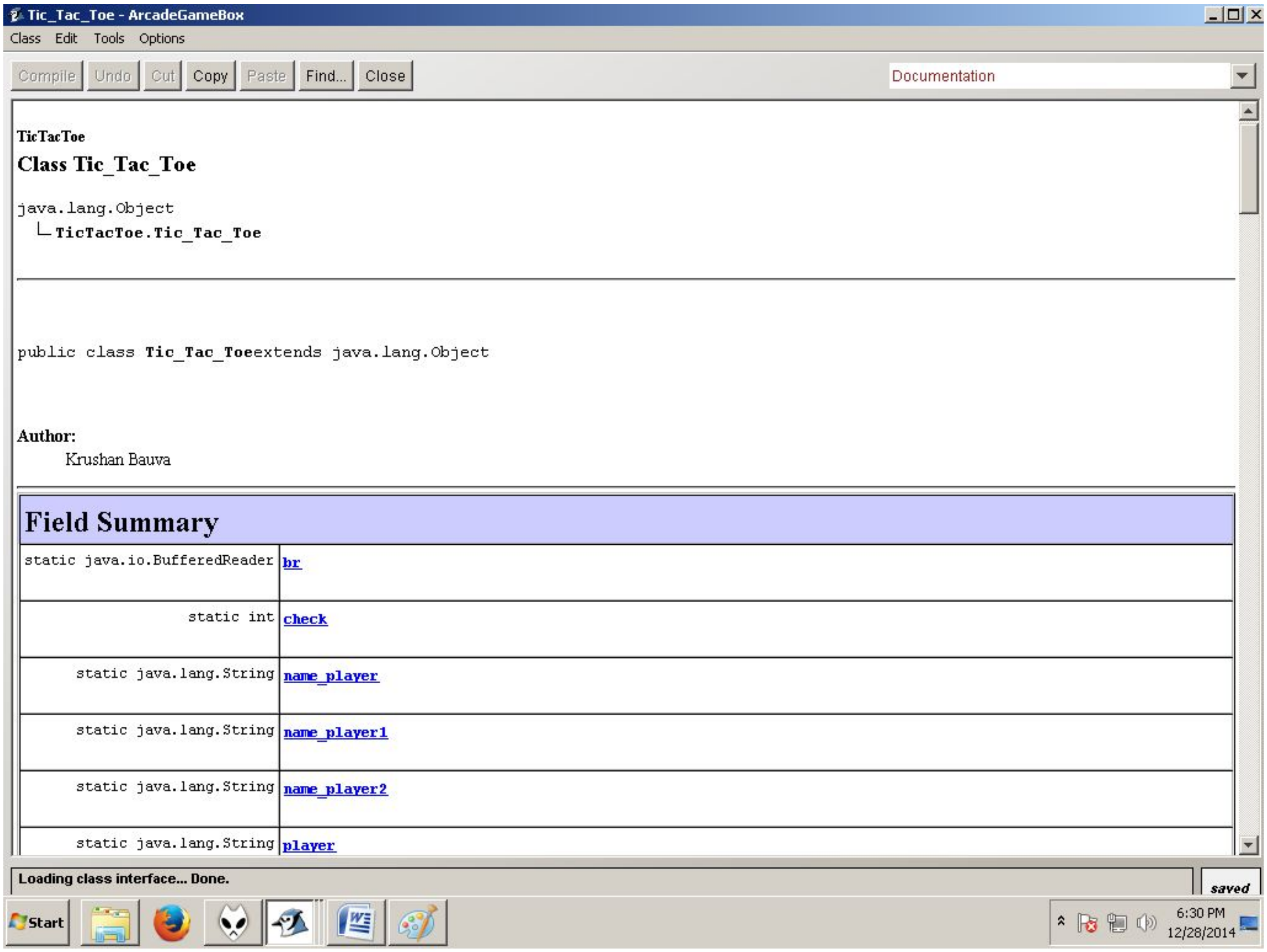
Loading class interface... Done.

saved

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe



TicTacToe

Class Tic_Tac_Toe

```
java.lang.Object
  └ TicTacToe.Tic_Tac_Toe
```

```
public class Tic_Tac_Toeextends java.lang.Object
```

Author:
Krushan Bauva

Field Summary

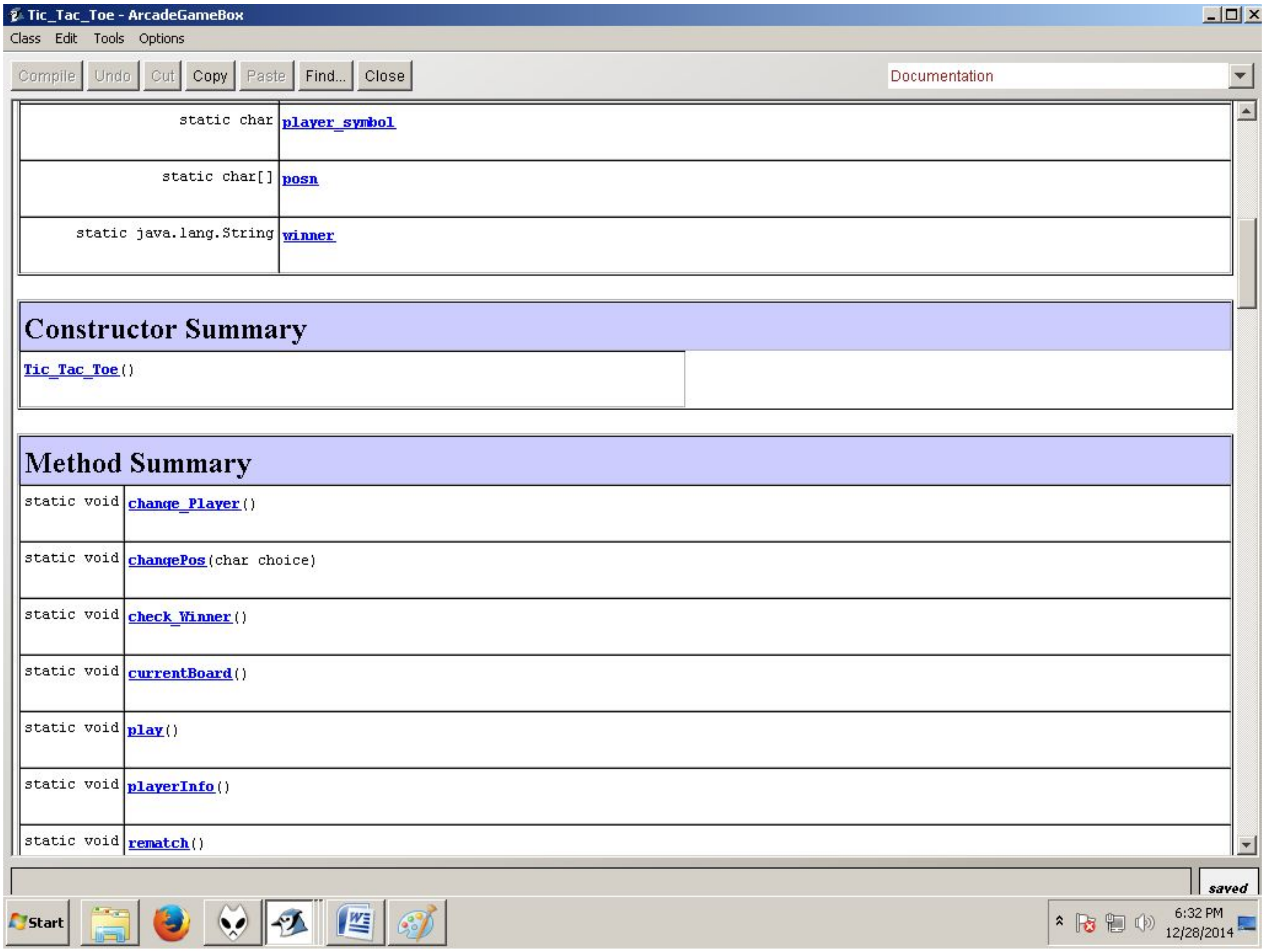
static java.io.BufferedReader	br
static int	check
static java.lang.String	name_player
static java.lang.String	name_player1
static java.lang.String	name_player2
static java.lang.String	player

Loading class interface... Done.

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe


Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:32 PM
12/28/2014

static char	player_symbol
static char[]	posn
static java.lang.String	winner

Constructor Summary

Tic_Tac_Toe()

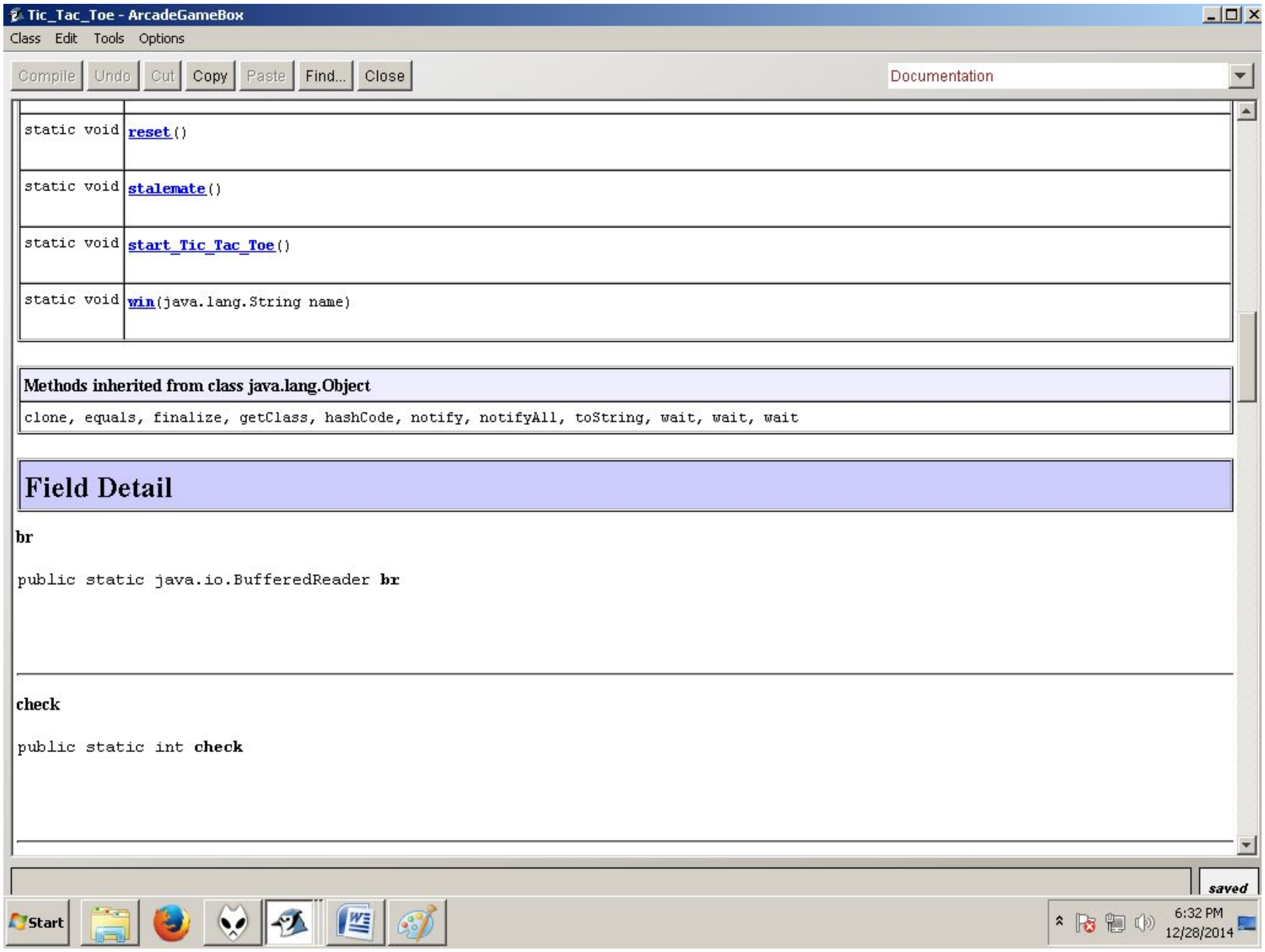
Method Summary

static void	change_Player()
static void	changePos(char choice)
static void	check_Winner()
static void	currentBoard()
static void	play()
static void	playerInfo()
static void	rematch()

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe


Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:32 PM
12/28/2014

static void	reset()
static void	stalemate()
static void	start_Tic_Tac_Toe()
static void	win(java.lang.String name)

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Field Detail

br

```
public static java.io.BufferedReader br
```

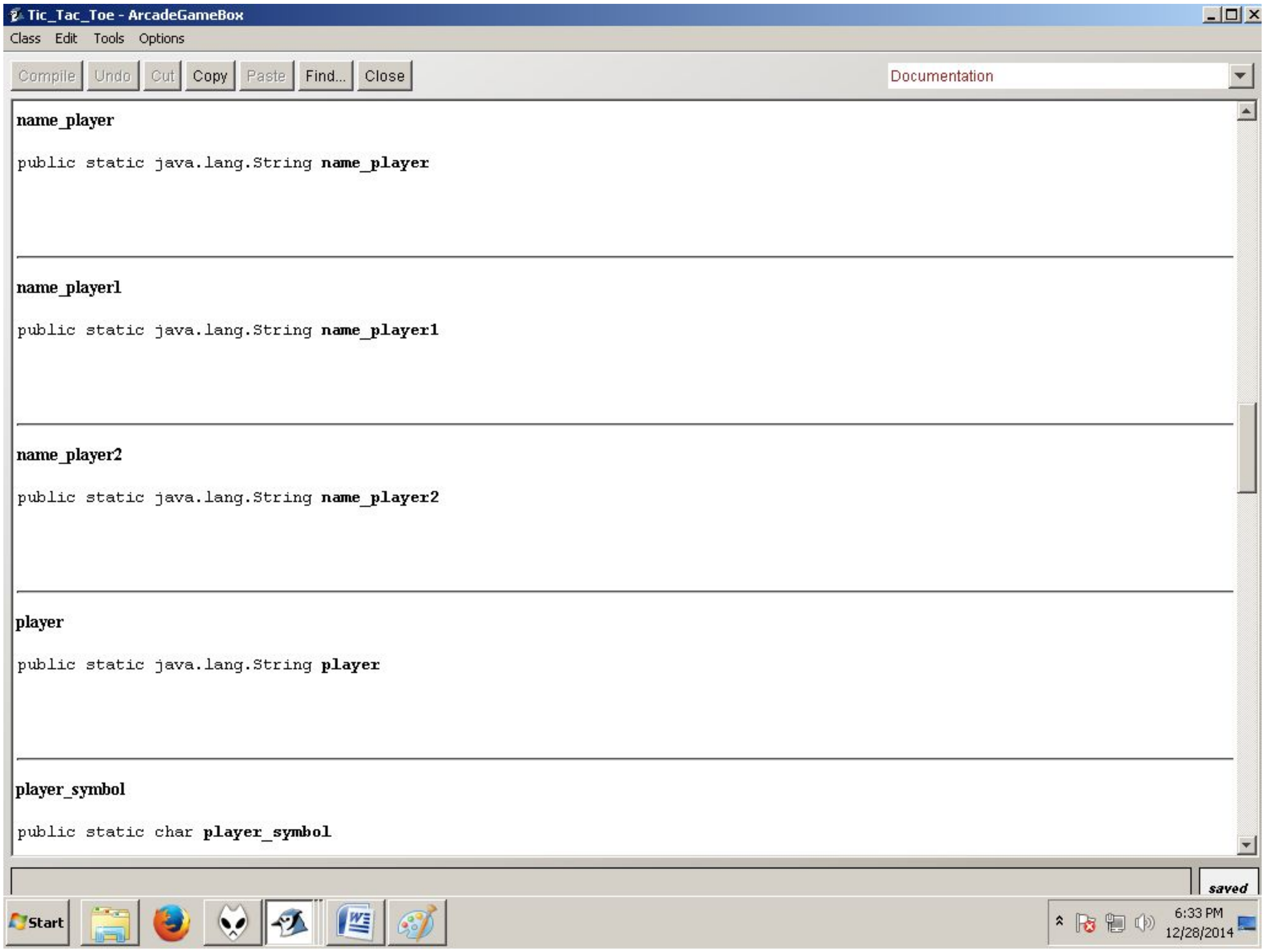
check

```
public static int check
```

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe



Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:33 PM
12/28/2014

name_player

`public static java.lang.String name_player`

name_player1

`public static java.lang.String name_player1`

name_player2

`public static java.lang.String name_player2`

player

`public static java.lang.String player`

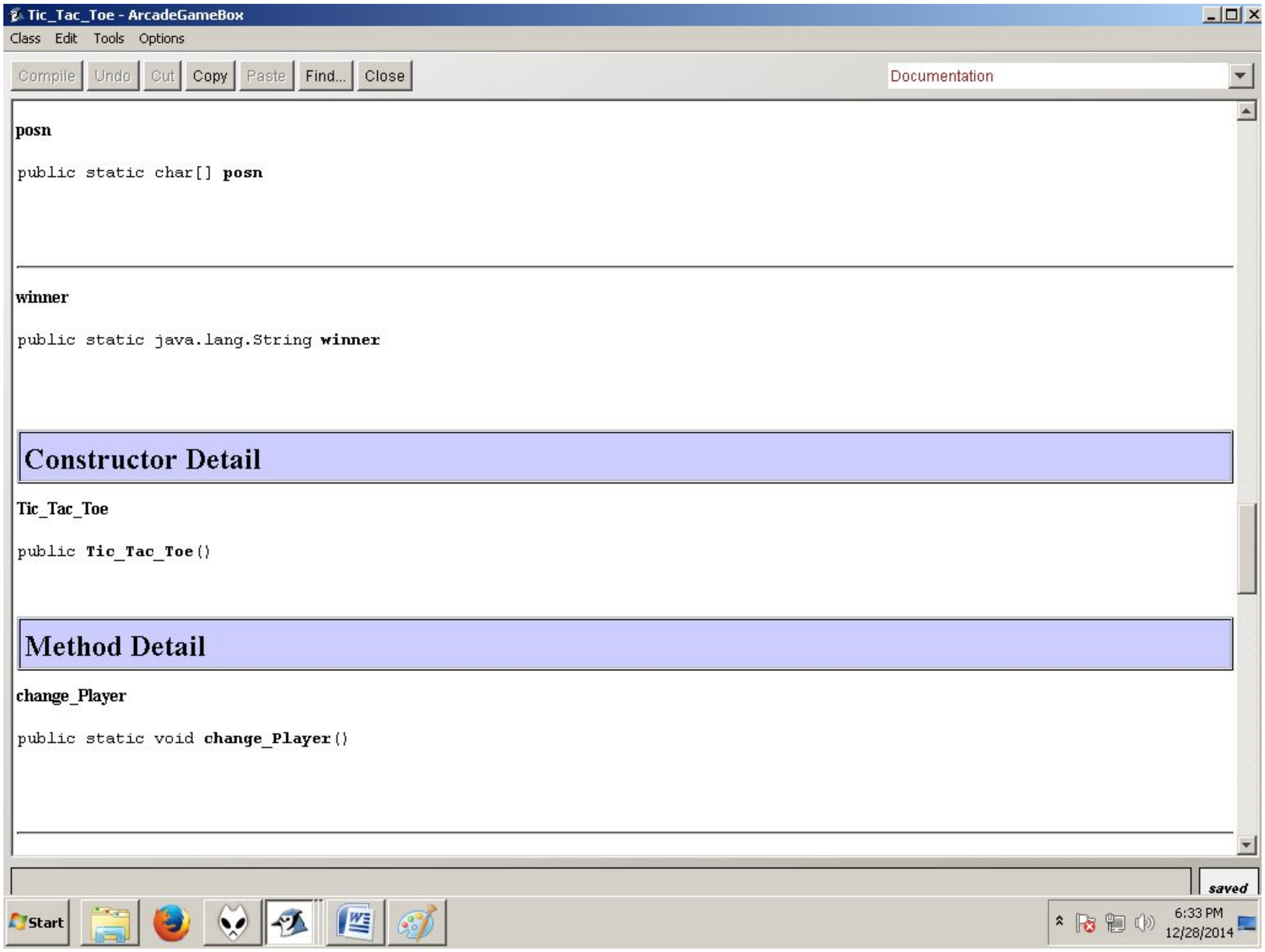
player_symbol

`public static char player_symbol`

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe


Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:33 PM
12/28/2014

posn

```
public static char[] posn
```

winner

```
public static java.lang.String winner
```

Constructor Detail

Tic_Tac_Toe

```
public Tic_Tac_Toe()
```

Method Detail

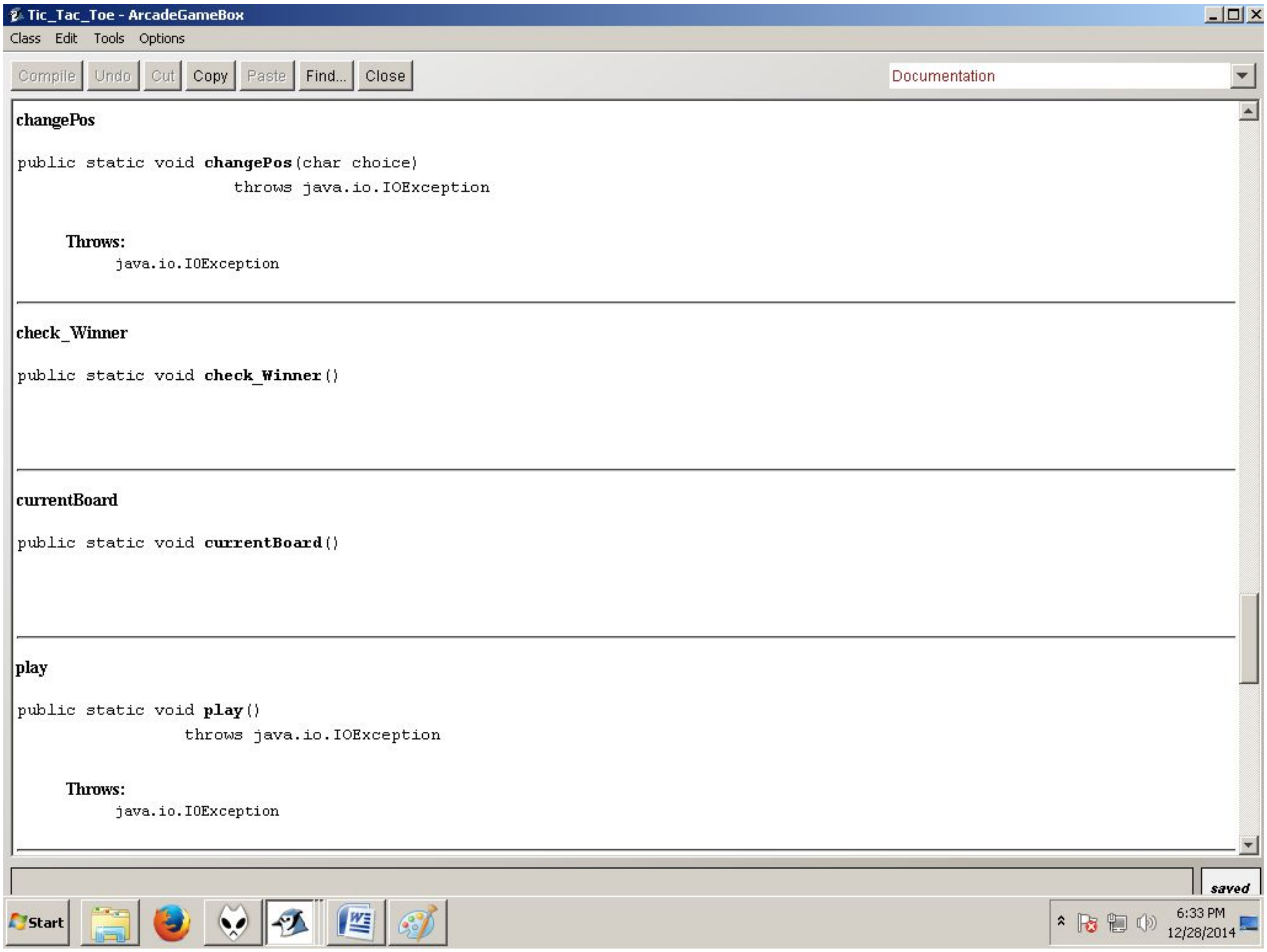
change_Player

```
public static void change_Player()
```

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe


Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:33 PM
12/28/2014

changePos

```
public static void changePos(char choice)
                      throws java.io.IOException
```

Throws:
java.io.IOException

check_Winner

```
public static void check_Winner()
```

currentBoard

```
public static void currentBoard()
```

play

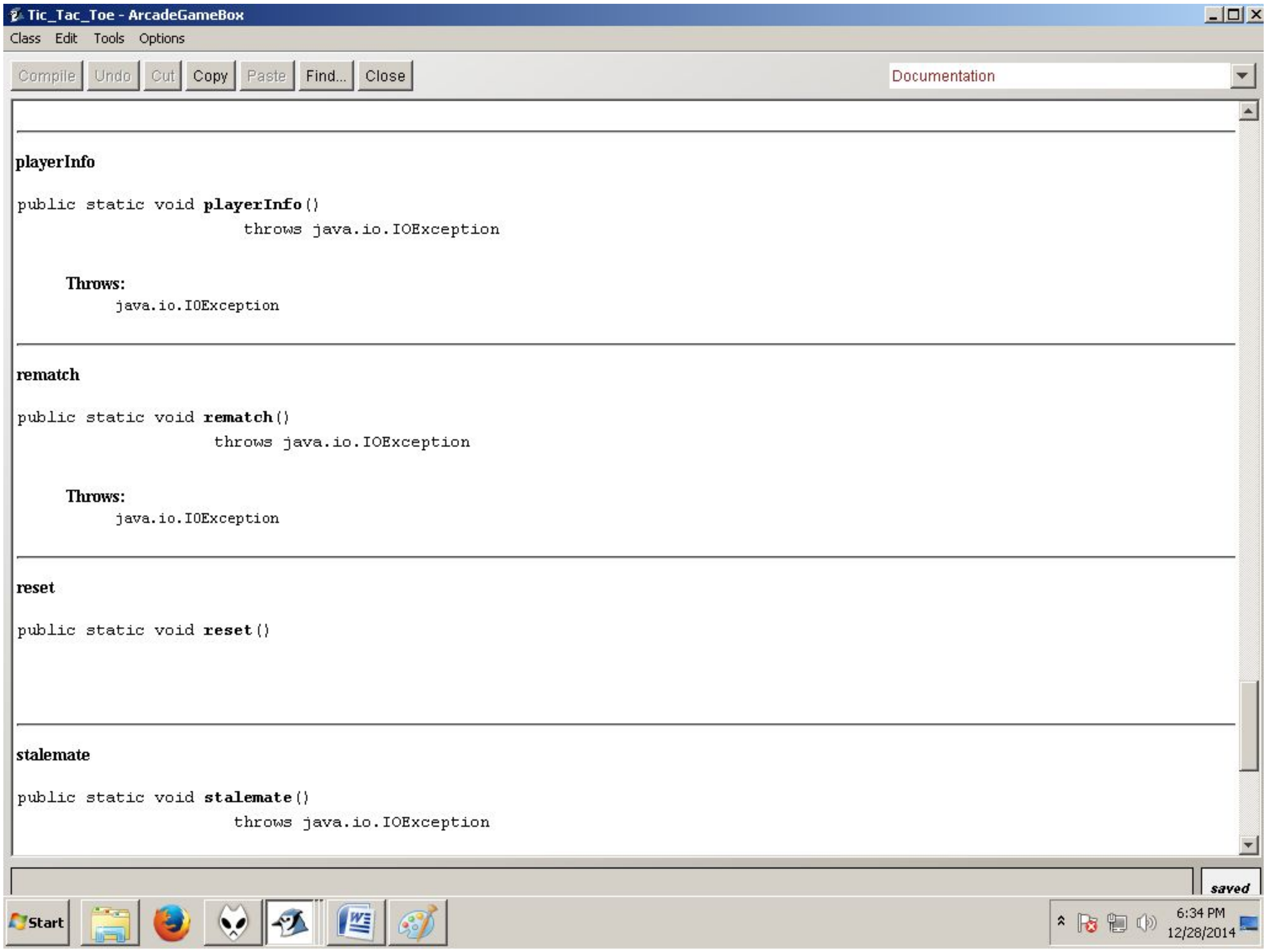
```
public static void play()
                throws java.io.IOException
```

Throws:
java.io.IOException

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe


Tic_Tac_Toe - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:34 PM
12/28/2014

playerInfo

```
public static void playerInfo()
                       throws java.io.IOException
```

Throws:
`java.io.IOException`

rematch

```
public static void rematch()
                    throws java.io.IOException
```

Throws:
`java.io.IOException`

reset

```
public static void reset()
```

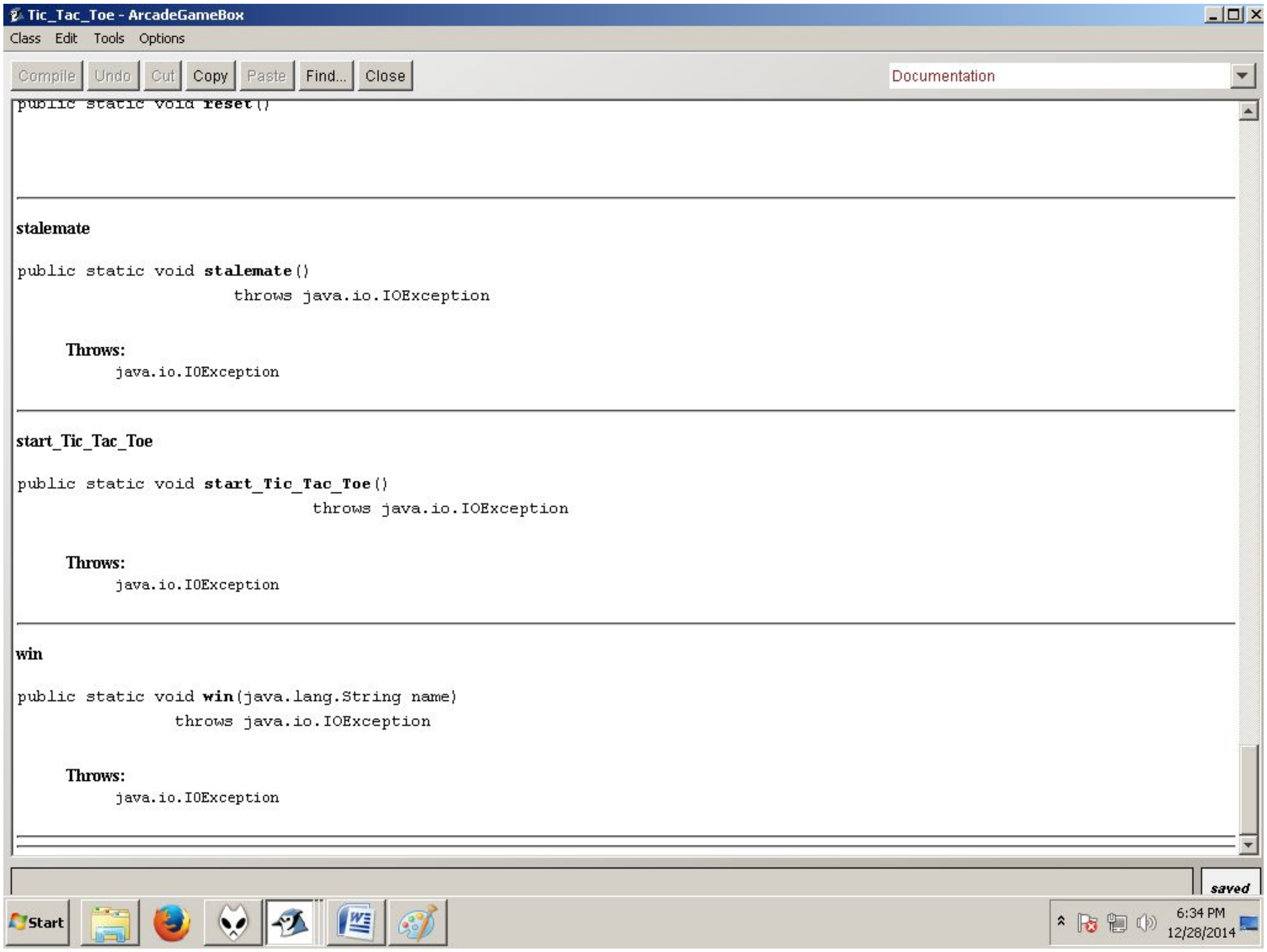
stalemate

```
public static void stalemate()
                      throws java.io.IOException
```

Project Documentation

Package Name: TicTacToe

Class Name: Tic_Tac_Toe



```
public static void reset()
```

stalemate

```
public static void stalemate()
                      throws java.io.IOException
```

Throws:
`java.io.IOException`

start_Tic_Tac_Toe

```
public static void start_Tic_Tac_Toe()
                              throws java.io.IOException
```

Throws:
`java.io.IOException`

win

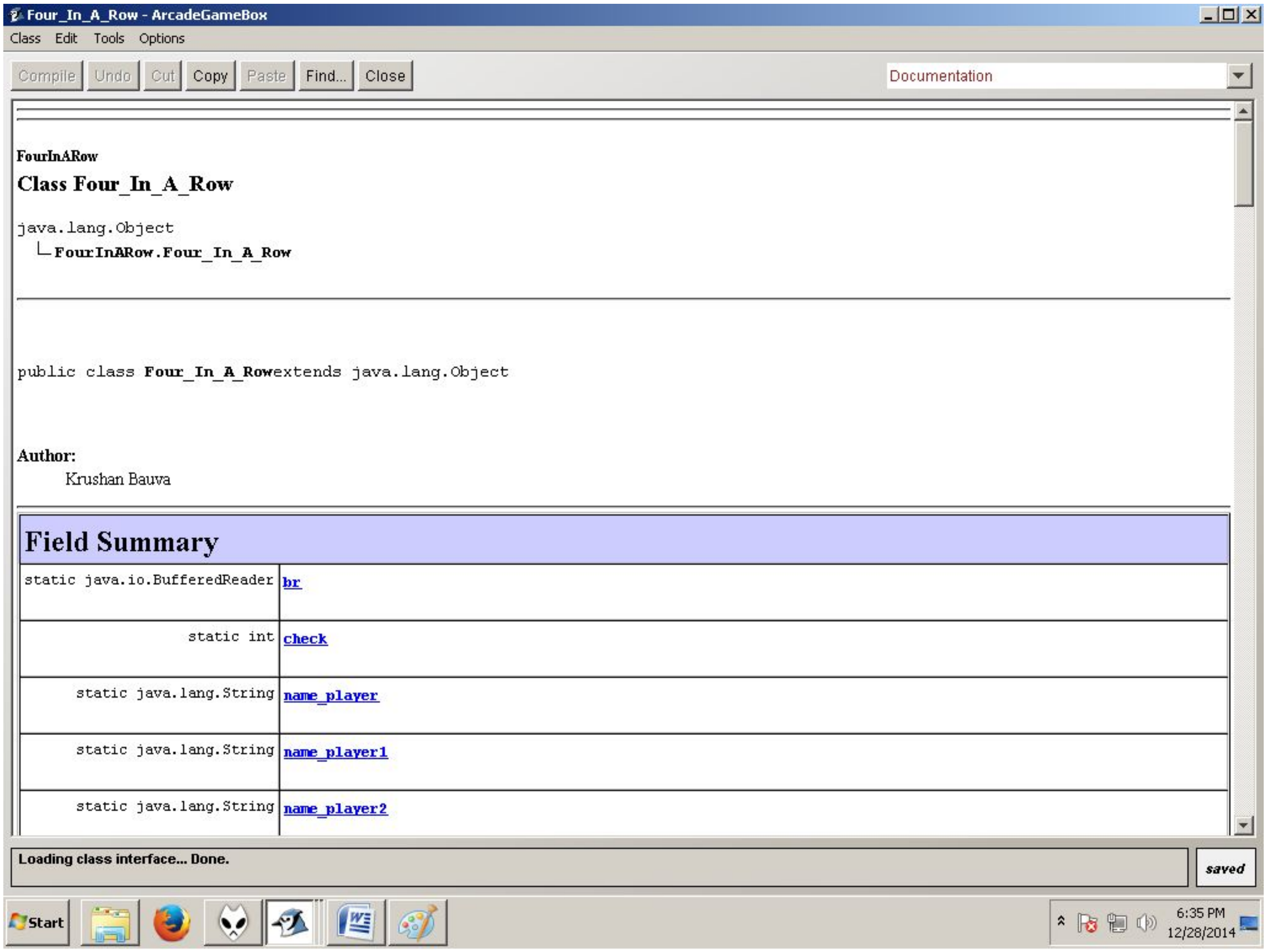
```
public static void win(java.lang.String name)
                throws java.io.IOException
```

Throws:
`java.io.IOException`

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row



FourInARow

Class Four_In_A_Row

```
java.lang.Object
  └─FourInARow.Four_In_A_Row
```

```
public class Four_In_A_Rowextends java.lang.Object
```

Author:
Krushan Bauva

Field Summary

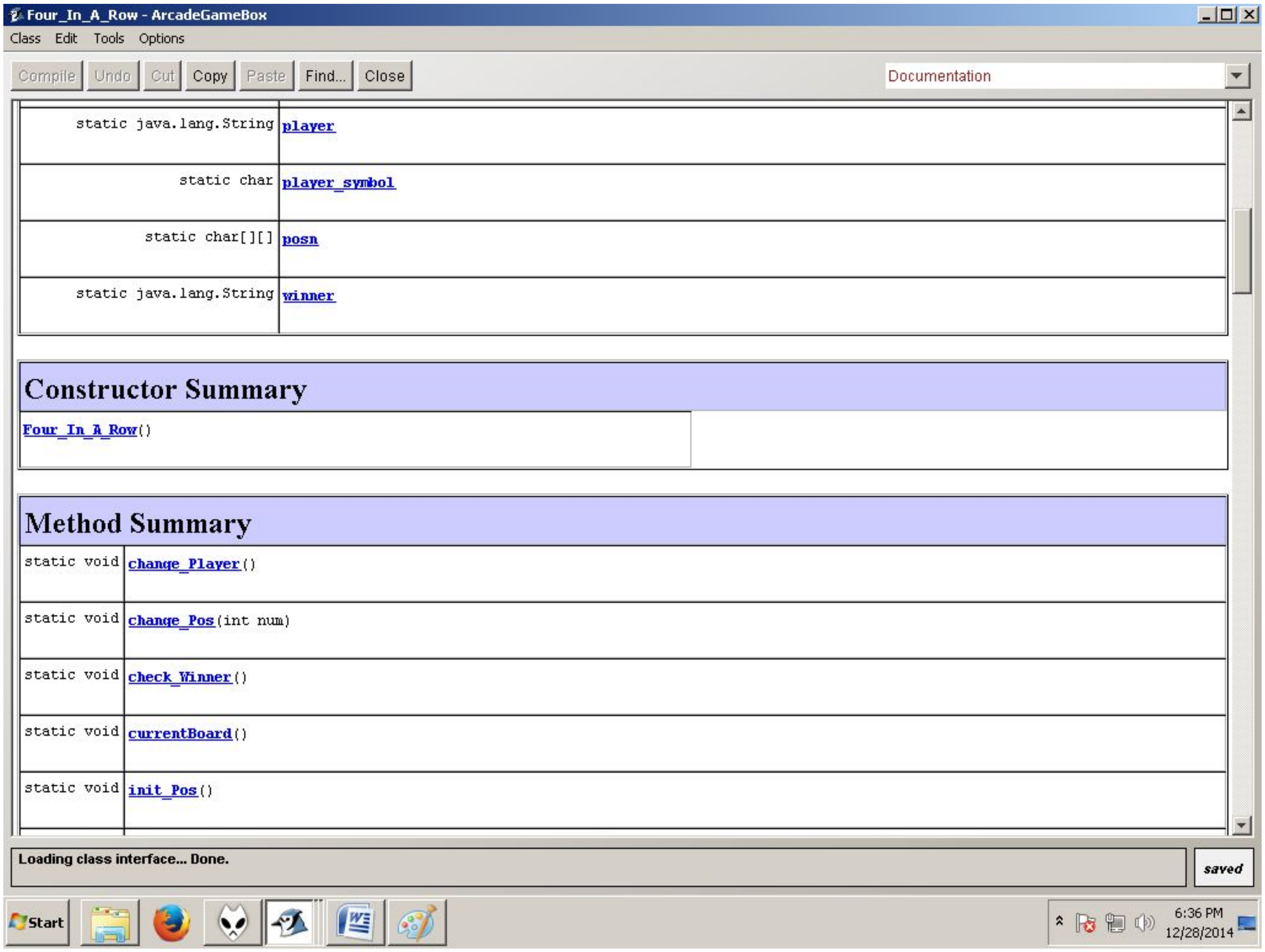
static java.io.BufferedReader	br
static int	check
static java.lang.String	name_player
static java.lang.String	name_player1
static java.lang.String	name_player2

Loading class interface... Done.

saved

Package Name: FourInARow

Class Name: Four_In_A_Row


Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
Start
6:36 PM
12/28/2014

static java.lang.String	player
static char	player_symbol
static char[][]	posn
static java.lang.String	winner

Constructor Summary

Four_In_A_Row()	

Method Summary

static void	change_Player()
static void	change_Pos(int num)
static void	check_Winner()
static void	currentBoard()
static void	init_Pos()

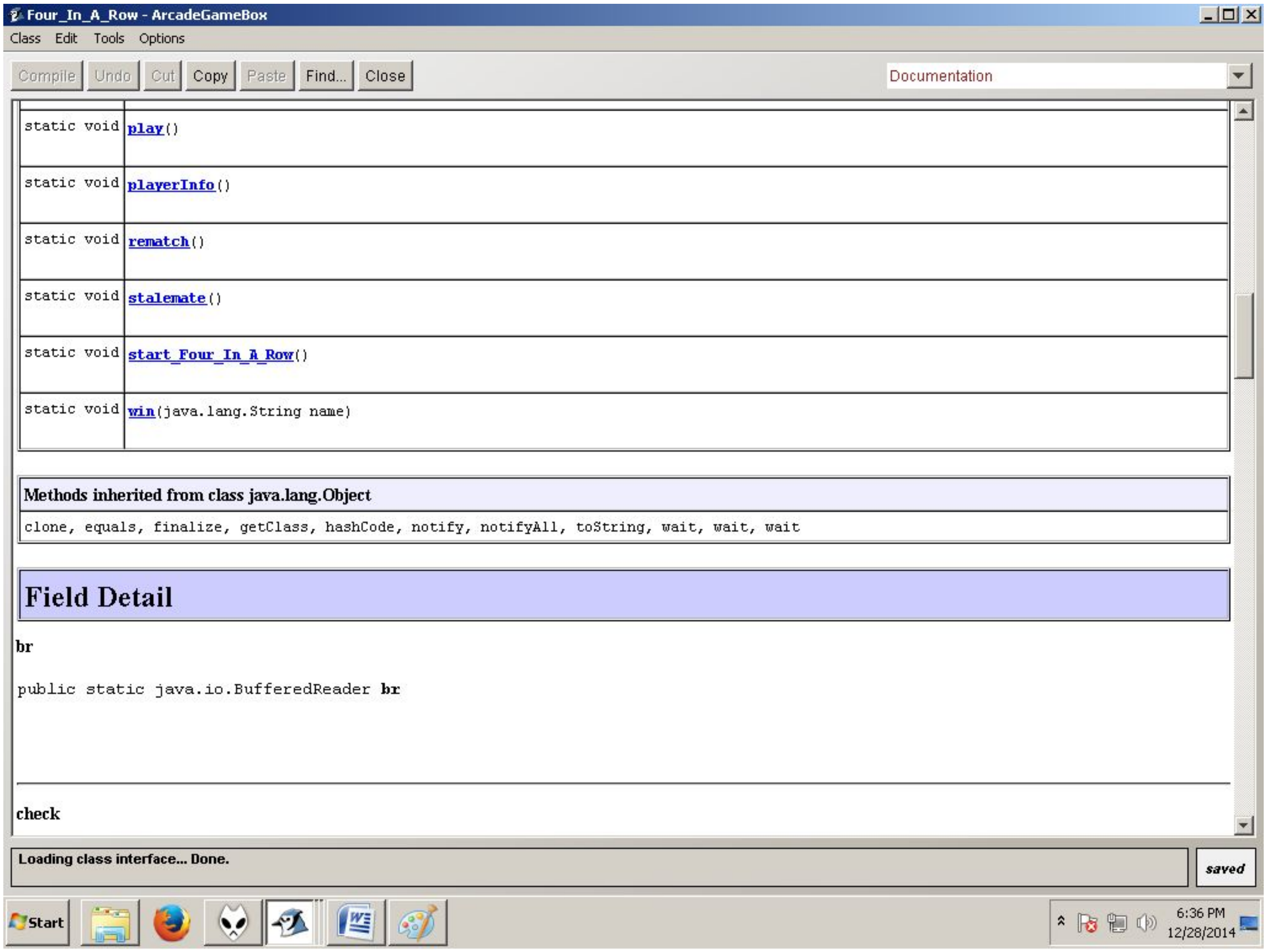
Loading class interface... Done.

saved

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row


Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
Start
6:36 PM
12/28/2014

static void	play()
static void	playerInfo()
static void	rematch()
static void	stalemate()
static void	start_Four_In_A_Row()
static void	win(java.lang.String name)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

br

```
public static java.io.BufferedReader br
```

check

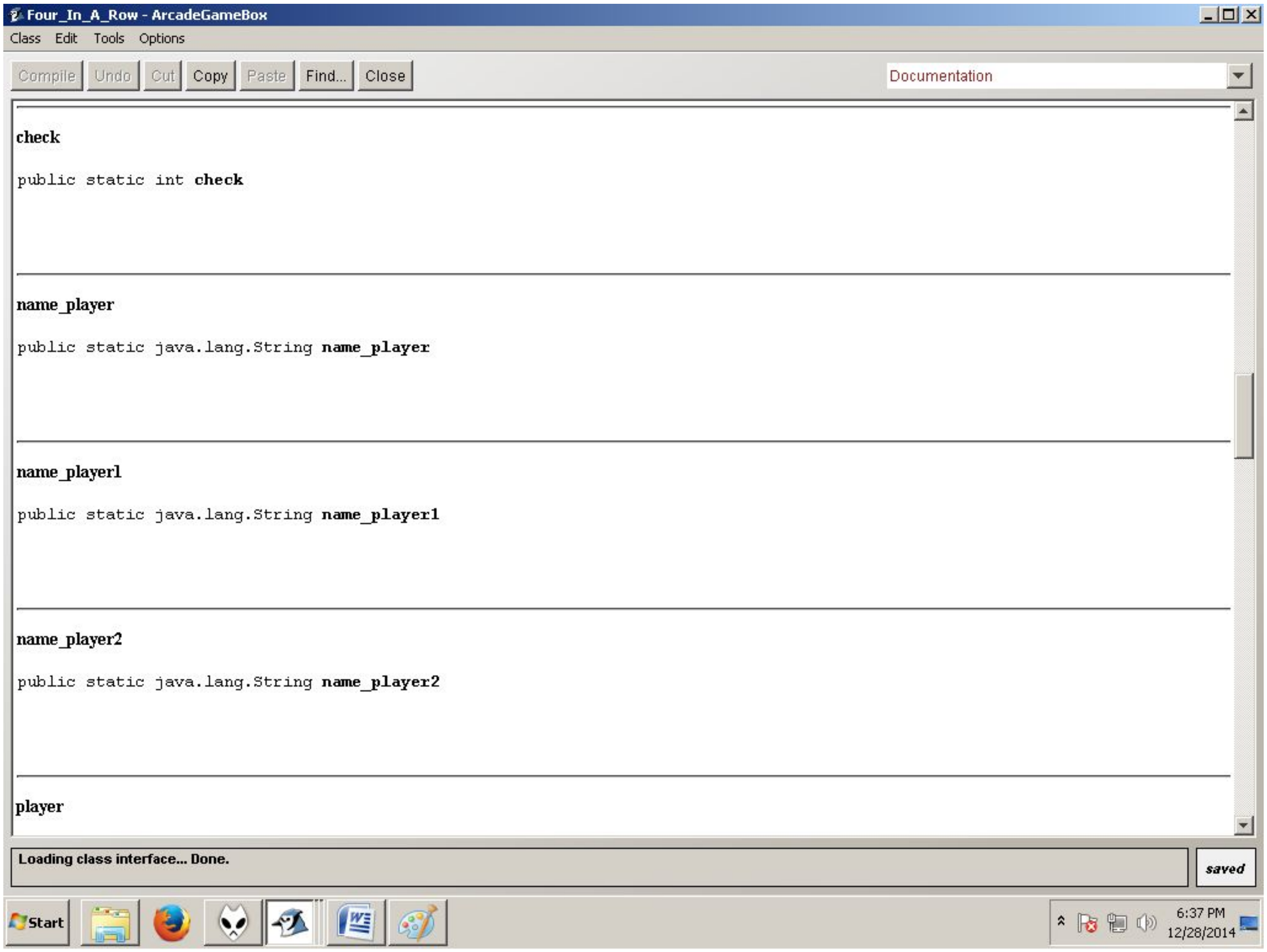
Loading class interface... Done.

saved

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row



Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation

check

```
public static int check
```

name_player

```
public static java.lang.String name_player
```

name_player1

```
public static java.lang.String name_player1
```

name_player2

```
public static java.lang.String name_player2
```

player

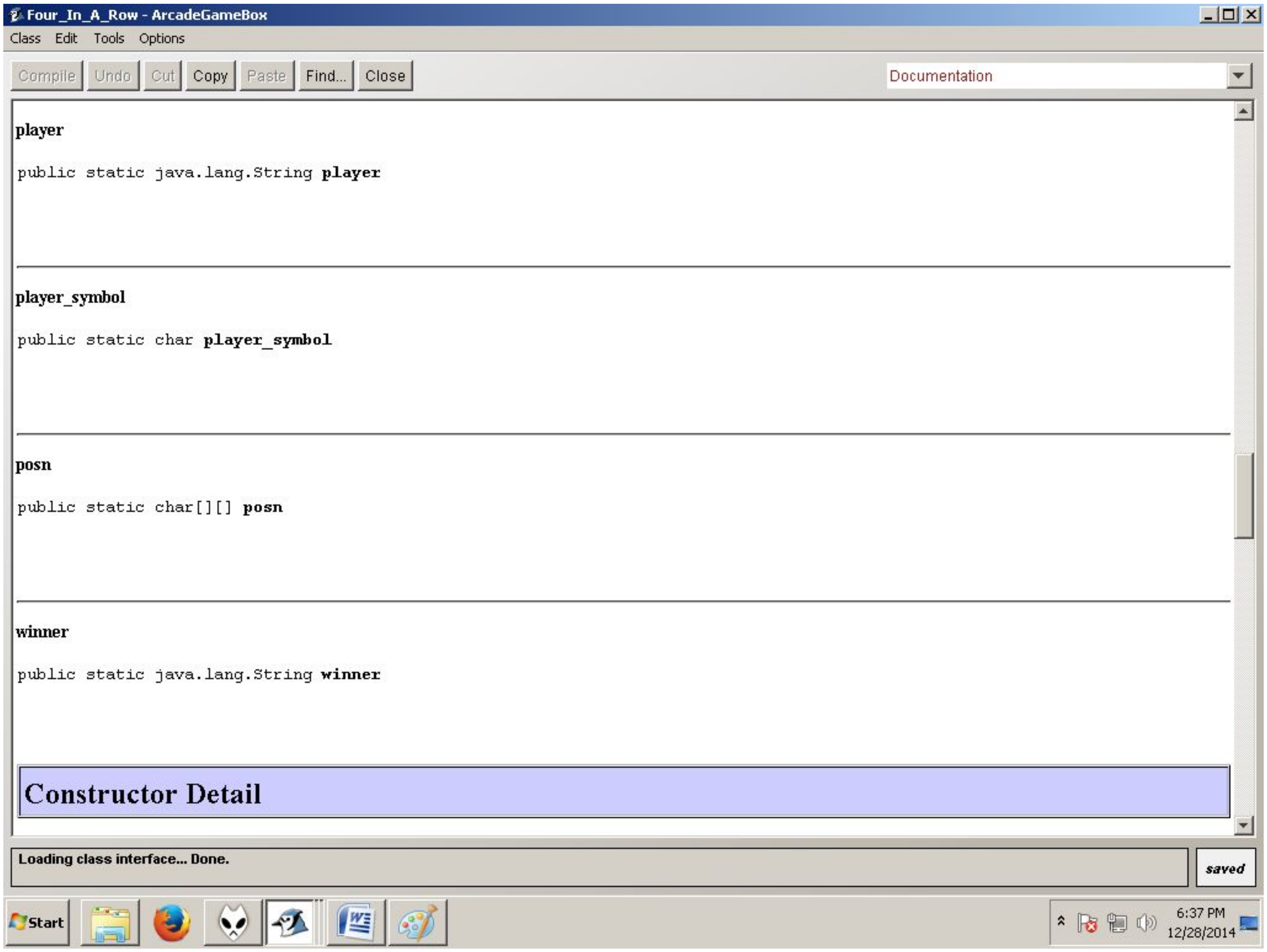
Loading class interface... Done.

saved
Start
6:37 PM
12/28/2014

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row


Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation

player

```
public static java.lang.String player
```

player_symbol

```
public static char player_symbol
```

posn

```
public static char[][] posn
```

winner

```
public static java.lang.String winner
```

Constructor Detail

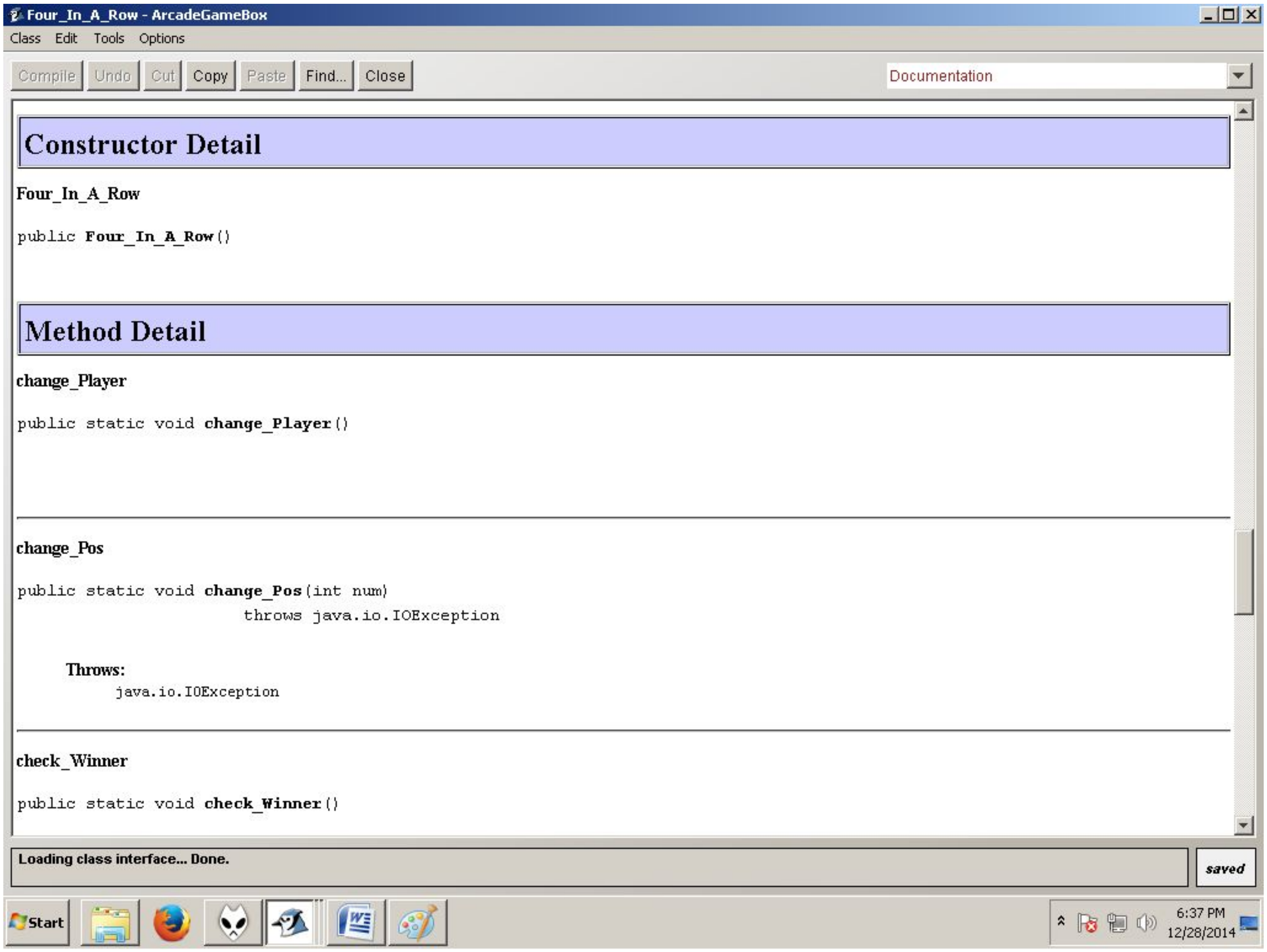
Loading class interface... Done.

saved
Start
6:37 PM
12/28/2014

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row



Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation

Constructor Detail

Four_In_A_Row

```
public Four_In_A_Row()
```

Method Detail

change_Player

```
public static void change_Player()
```

change_Pos

```
public static void change_Pos(int num)
                       throws java.io.IOException
```

Throws:
`java.io.IOException`

check_Winner

```
public static void check_Winner()
```

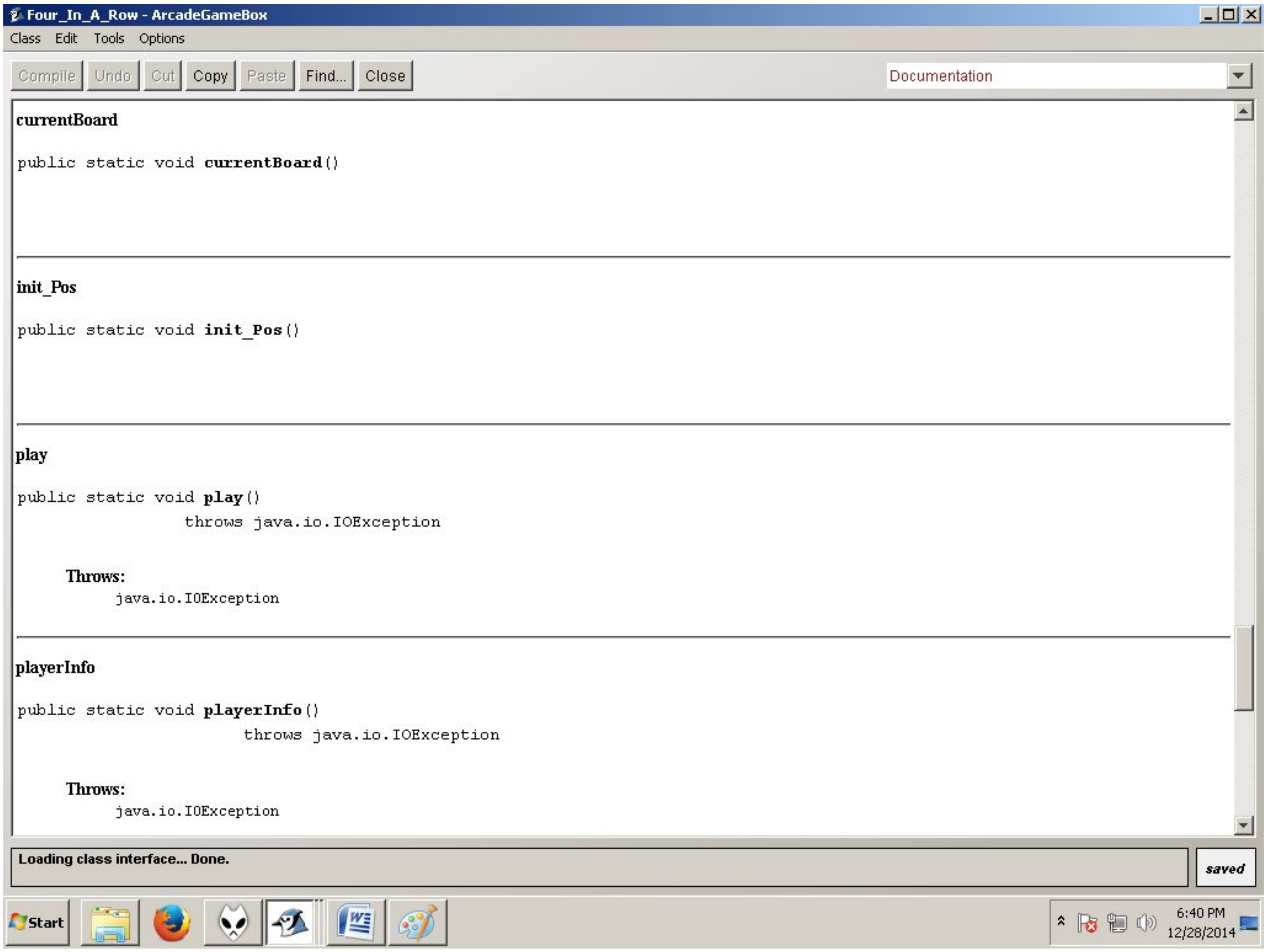
Loading class interface... Done.

saved

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row



Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation

currentBoard

```
public static void currentBoard()
```

init_Pos

```
public static void init_Pos()
```

play

```
public static void play()
                 throws java.io.IOException
```

Throws:
`java.io.IOException`

playerInfo

```
public static void playerInfo()
                       throws java.io.IOException
```

Throws:
`java.io.IOException`

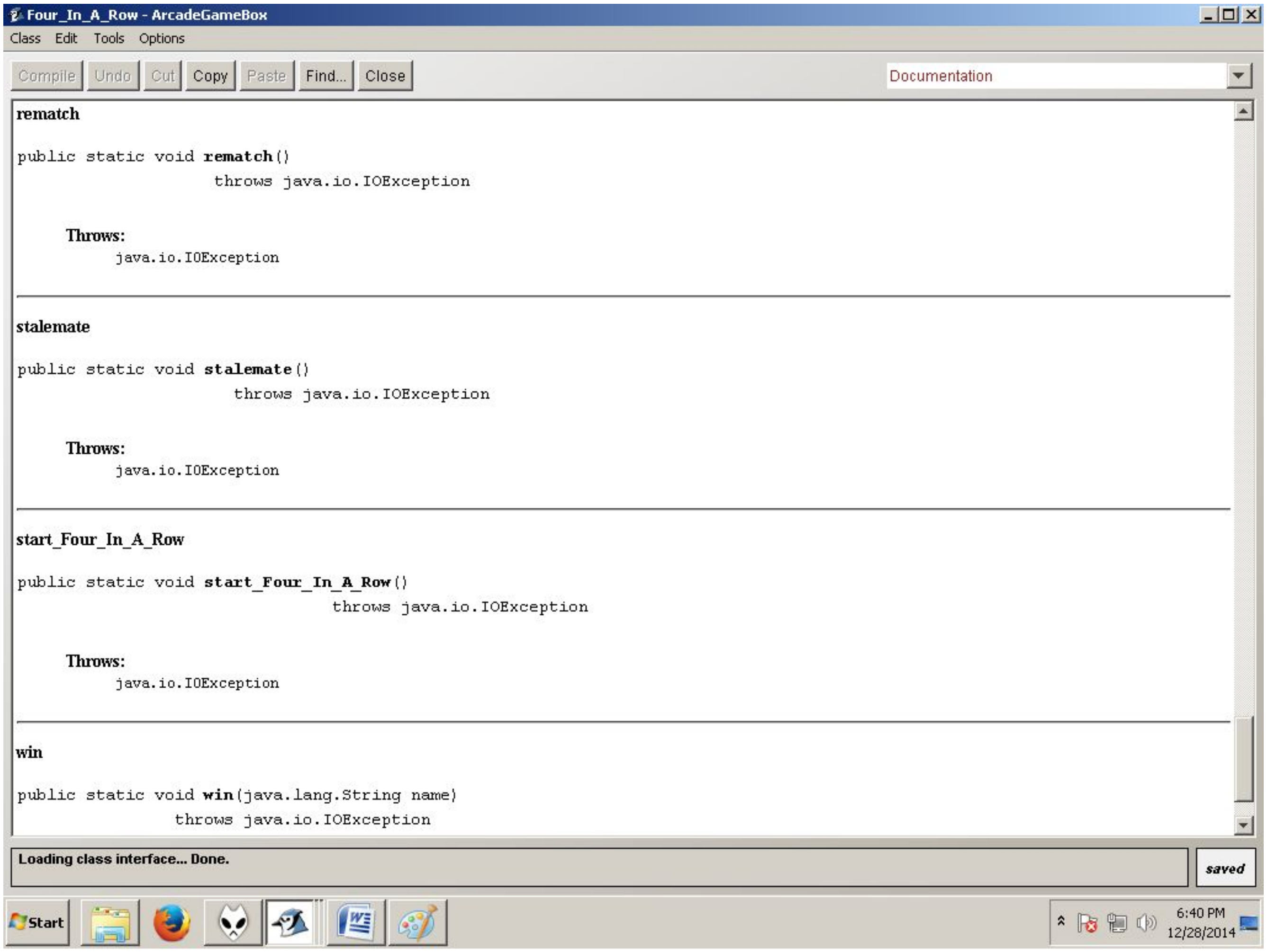
Loading class interface... Done.

saved
Start
6:40 PM
12/28/2014

Project Documentation

Package Name: FourInARow

Class Name: Four_In_A_Row


Four_In_A_Row - ArcadeGameBox
Class Edit Tools Options
Compile Undo Cut Copy Paste Find... Close
Documentation
saved
Start
6:40 PM
12/28/2014

rematch

```
public static void rematch()
                    throws java.io.IOException
```

Throws:
`java.io.IOException`

stalemate

```
public static void stalemate()
                      throws java.io.IOException
```

Throws:
`java.io.IOException`

start_Four_In_A_Row

```
public static void start_Four_In_A_Row()
                                throws java.io.IOException
```

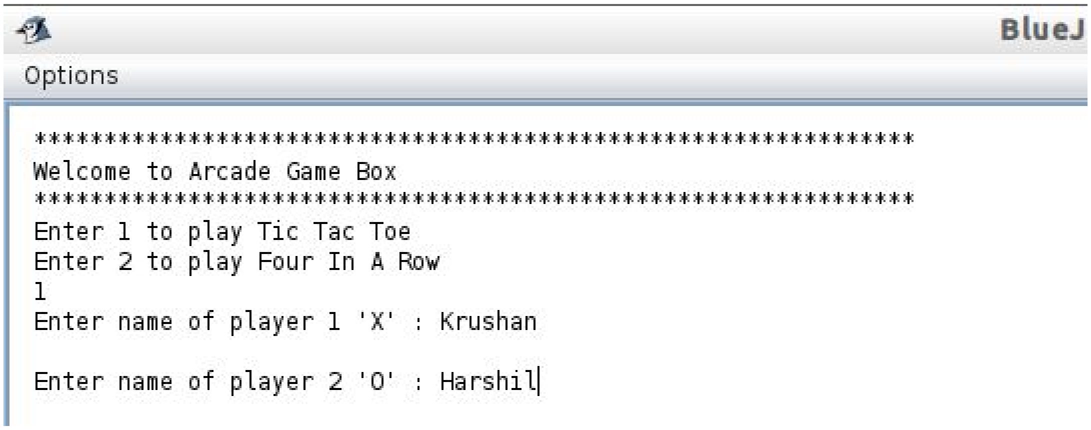
Throws:
`java.io.IOException`

win

```
public static void win(java.lang.String name)
                throws java.io.IOException
```

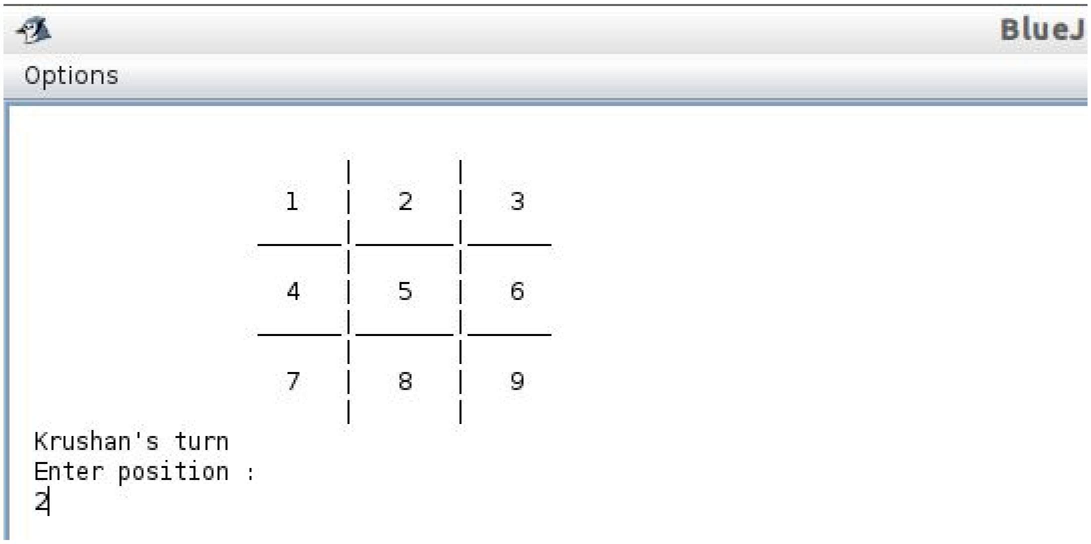
Loading class interface... Done.

Screen shots



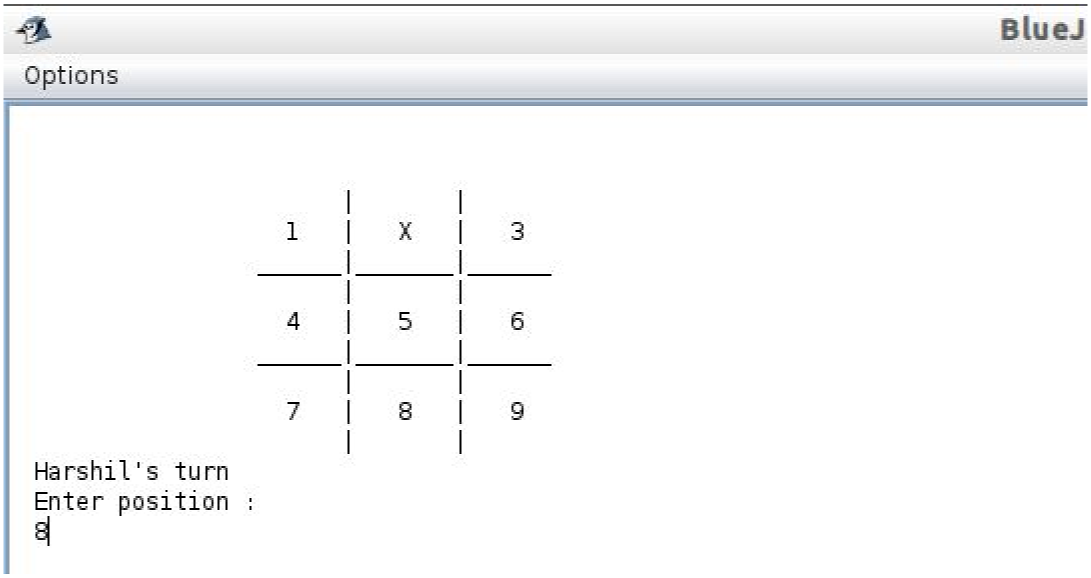
```
***************************************************************
Welcome to Arcade Game Box
***************************************************************
Enter 1 to play Tic Tac Toe
Enter 2 to play Four In A Row
1
Enter name of player 1 'X' : Krushan

Enter name of player 2 'O' : Harshil
```

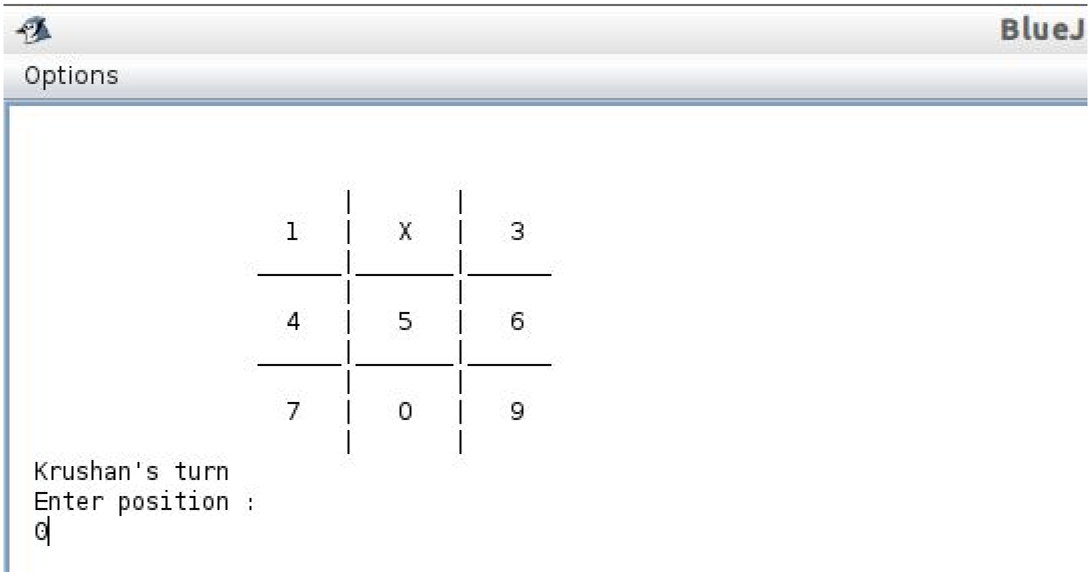


```
                      |       |
                  1   |   2   |   3
                ______|_______|______
                      |       |
                  4   |   5   |   6
                ______|_______|______
                      |       |
                  7   |   8   |   9
                      |       |
Krushan's turn
Enter position :
2
```

Screen shots

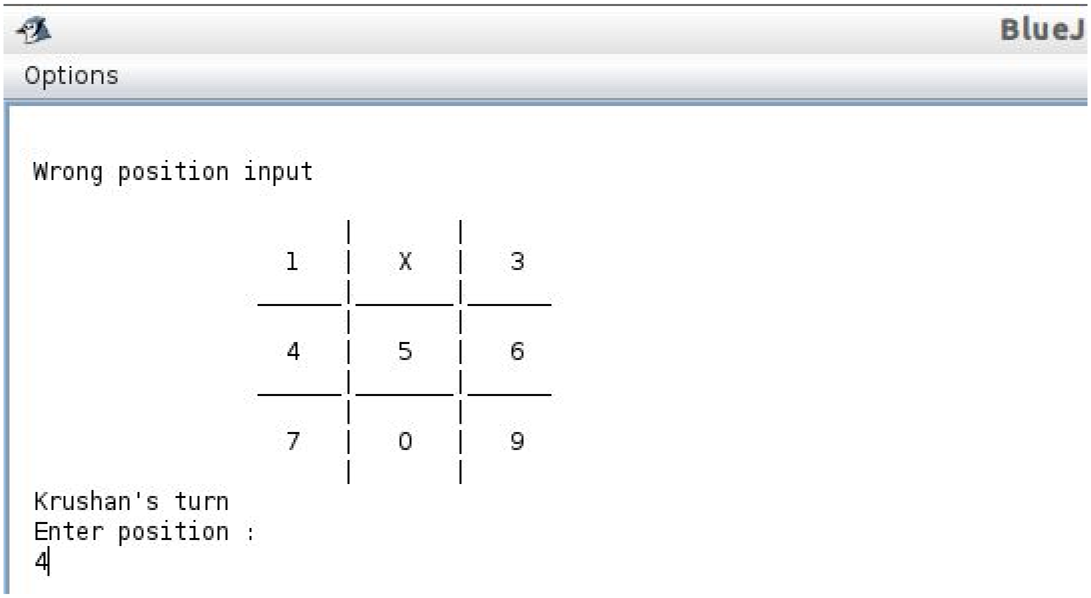


```
                      |         |
               1      |    X    |    3
             _________|_________|_________
                      |         |
               4      |    5    |    6
             _________|_________|_________
                      |         |
               7      |    8    |    9
                      |         |
Harshil's turn
Enter position :
8
```



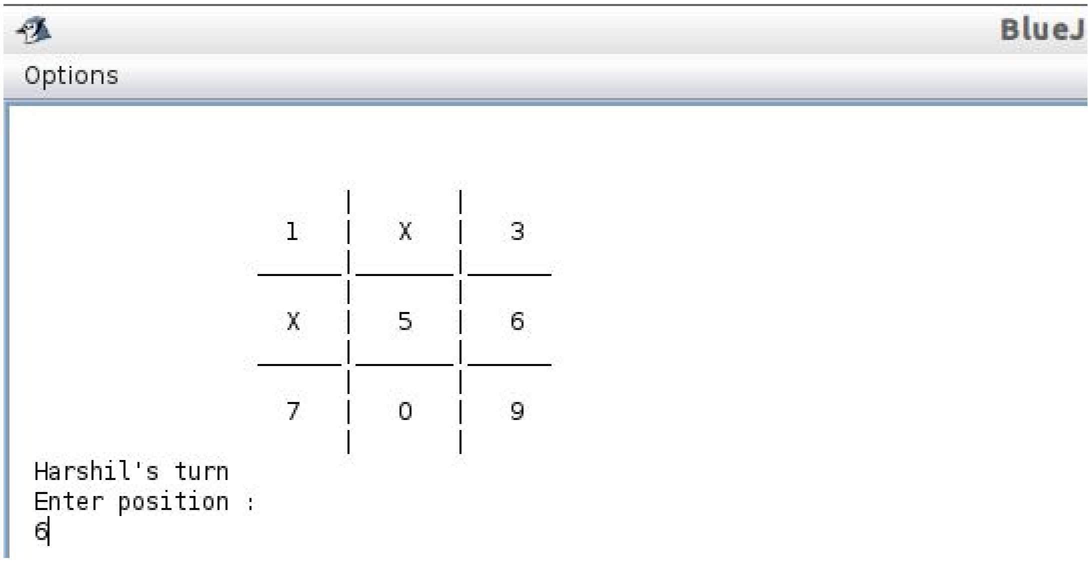
```
                      |         |
               1      |    X    |    3
             _________|_________|_________
                      |         |
               4      |    5    |    6
             _________|_________|_________
                      |         |
               7      |    0    |    9
                      |         |
Krushan's turn
Enter position :
0
```

Screen shots



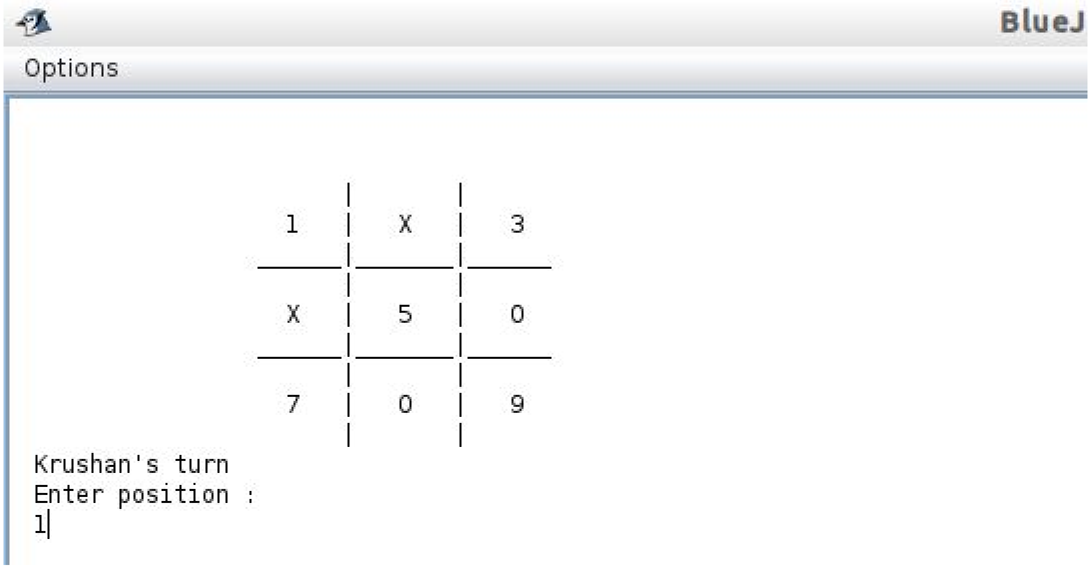
```
Wrong position input

                 |       |
           1     |   X   |   3
         ________|_______|________
                 |       |
           4     |   5   |   6
         ________|_______|________
                 |       |
           7     |   0   |   9
                 |       |
Krushan's turn
Enter position :
4
```

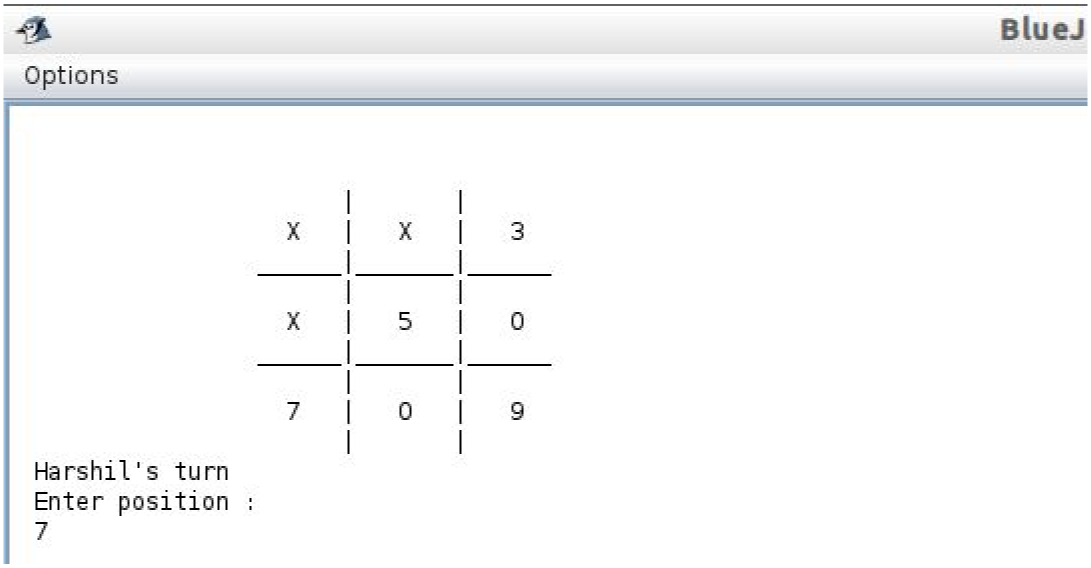


```
                 |       |
           1     |   X   |   3
         ________|_______|________
                 |       |
           X     |   5   |   6
         ________|_______|________
                 |       |
           7     |   0   |   9
                 |       |
Harshil's turn
Enter position :
6
```

Screen shots

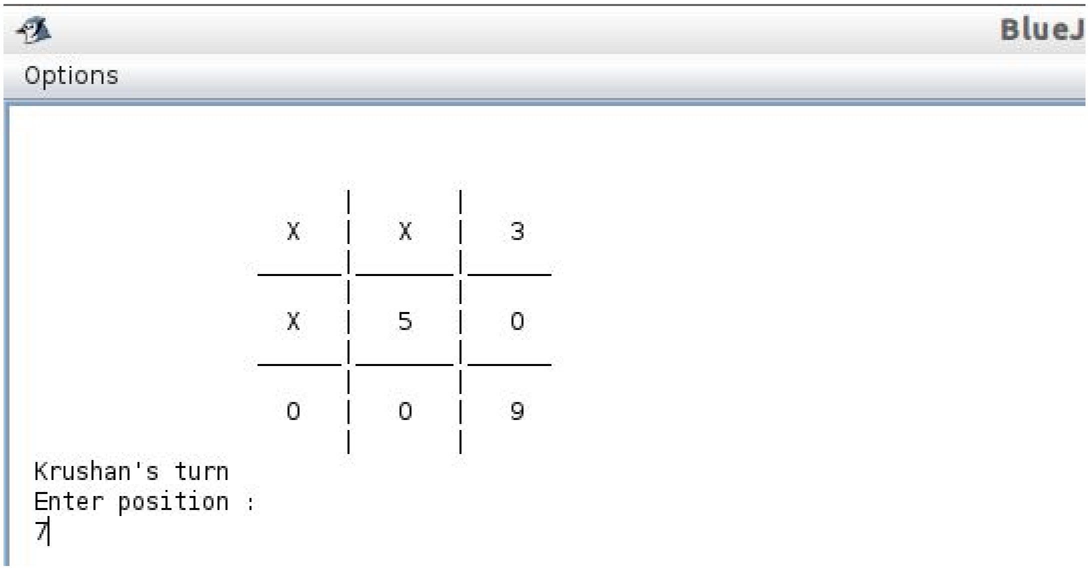


```
                     |         |
              1      |    X    |    3
           __________|_________|__________
                     |         |
              X      |    5    |    O
           __________|_________|__________
                     |         |
              7      |    O    |    9
                     |         |
Krushan's turn
Enter position :
1
```

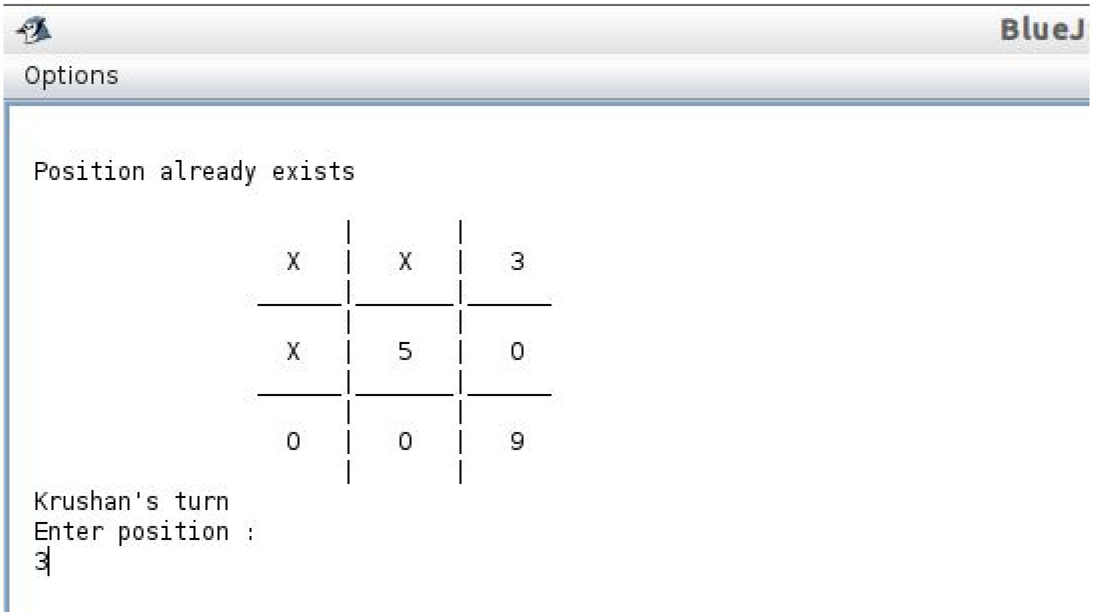


```
                     |         |
              X      |    X    |    3
           __________|_________|__________
                     |         |
              X      |    5    |    O
           __________|_________|__________
                     |         |
              7      |    O    |    9
                     |         |
Harshil's turn
Enter position :
7
```

Screen shots



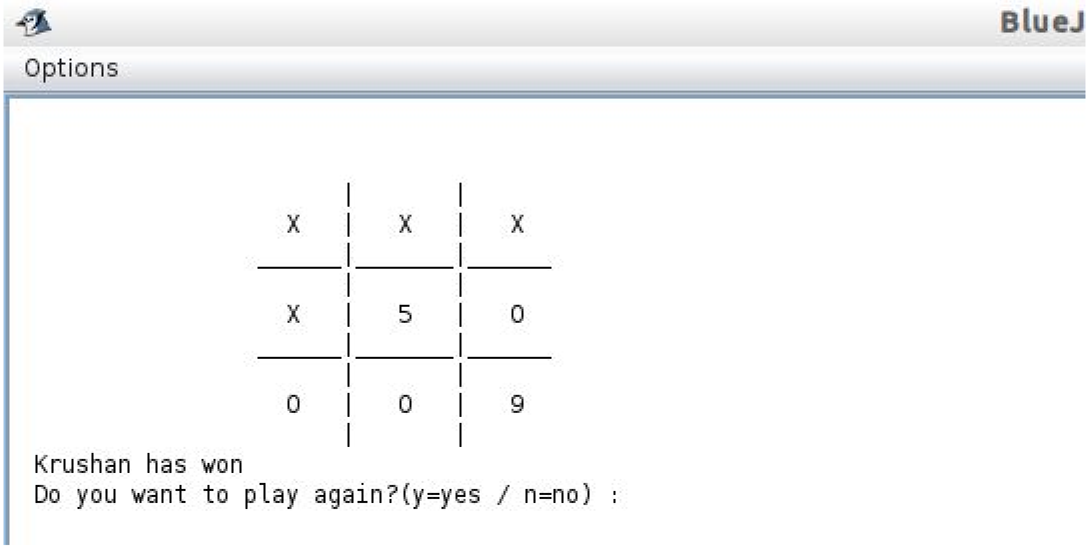
```
                        |       |
                    X   |   X   |   3
                 _______|_______|_______
                        |       |
                    X   |   5   |   O
                 _______|_______|_______
                        |       |
                    O   |   O   |   9
                        |       |
Krushan's turn
Enter position :
7
```



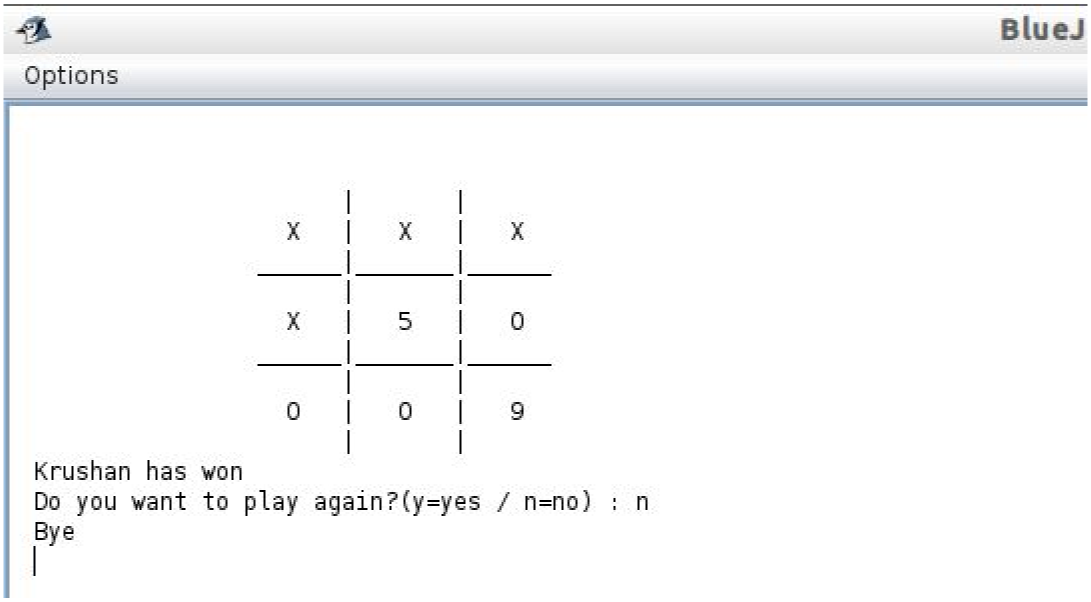
```
Position already exists

                        |       |
                    X   |   X   |   3
                 _______|_______|_______
                        |       |
                    X   |   5   |   O
                 _______|_______|_______
                        |       |
                    O   |   O   |   9
                        |       |
Krushan's turn
Enter position :
3
```

Screen shots

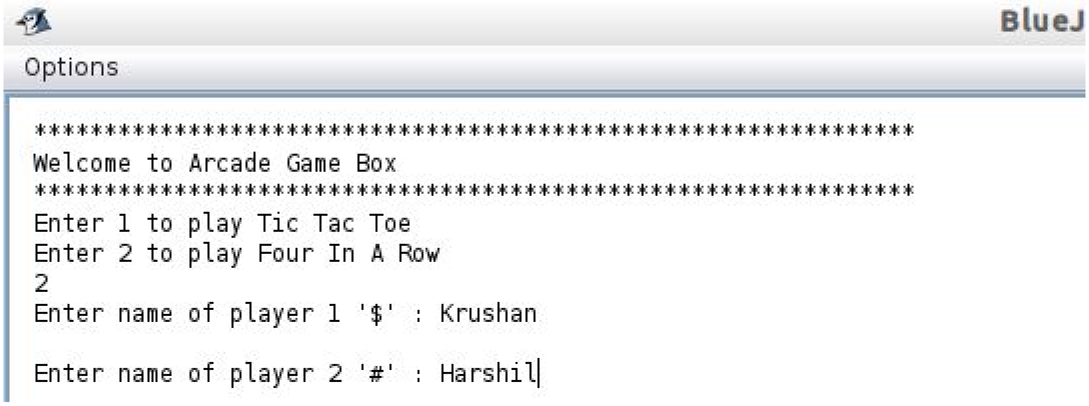


```
               |       |
          X    |   X   |   X
        _______|_______|_______
               |       |
          X    |   5   |   O
        _______|_______|_______
               |       |
          O    |   O   |   9
               |       |
Krushan has won
Do you want to play again?(y=yes / n=no) :
```



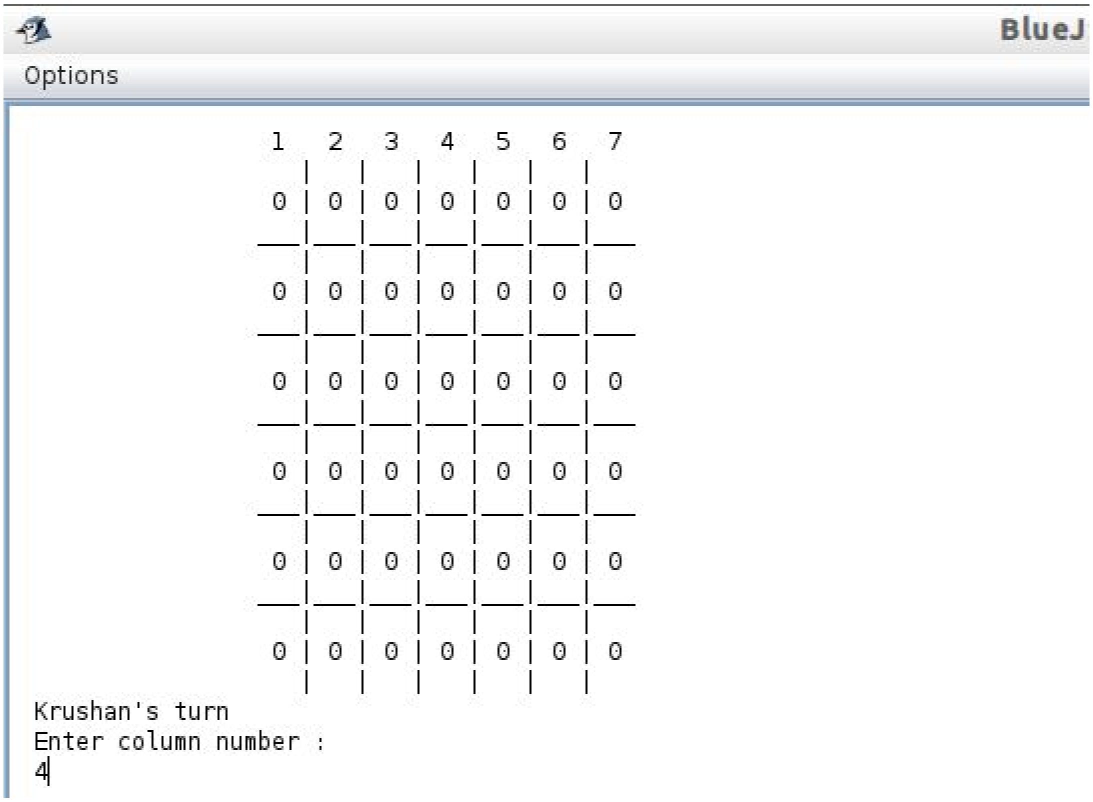
```
               |       |
          X    |   X   |   X
        _______|_______|_______
               |       |
          X    |   5   |   O
        _______|_______|_______
               |       |
          O    |   O   |   9
               |       |
Krushan has won
Do you want to play again?(y=yes / n=no) : n
Bye
```

Screen shots



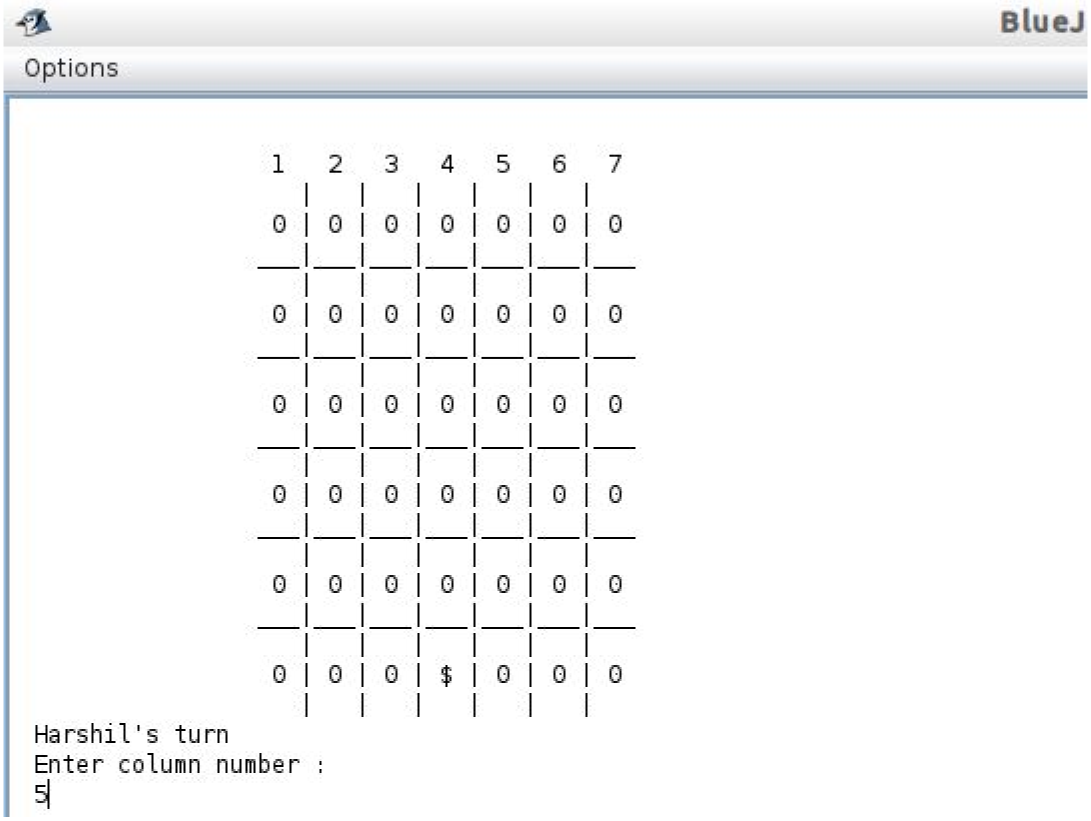
```
***************************************************************
Welcome to Arcade Game Box
***************************************************************
Enter 1 to play Tic Tac Toe
Enter 2 to play Four In A Row
2
Enter name of player 1 '$' : Krushan

Enter name of player 2 '#' : Harshil
```

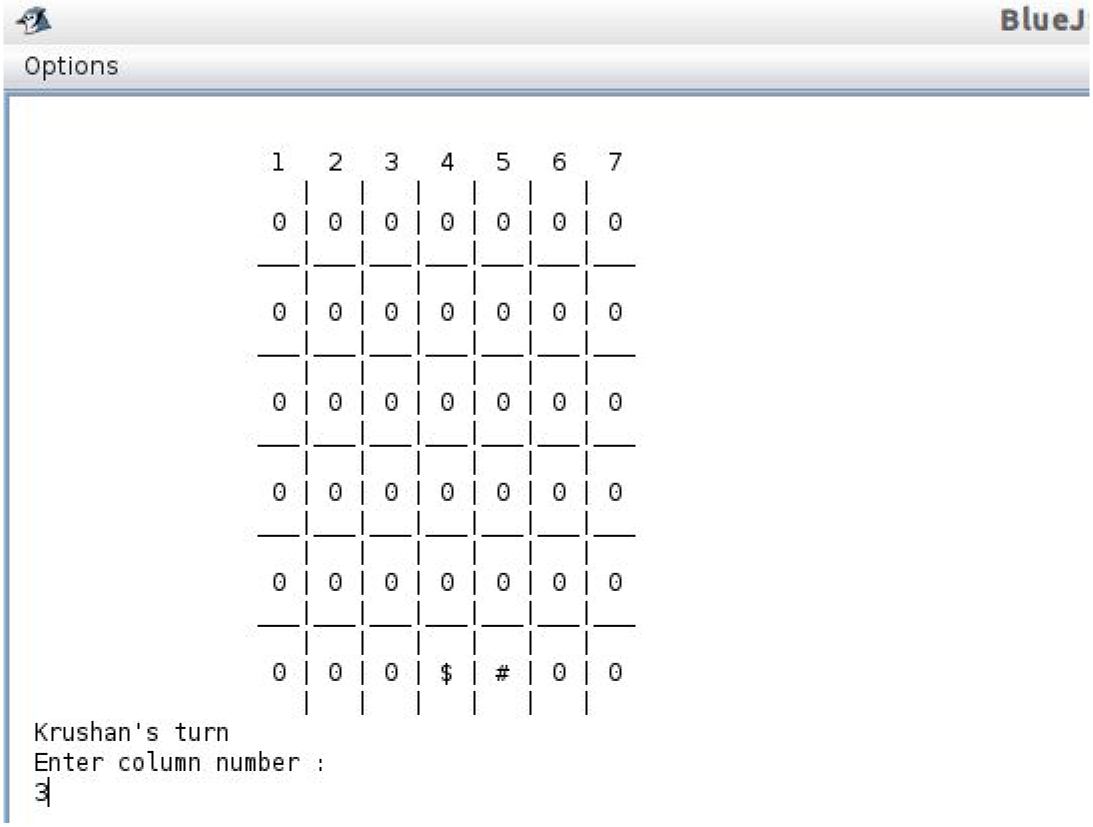


```
                 1   2   3   4   5   6   7
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                ___|___|___|___|___|___|___
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                ___|___|___|___|___|___|___
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                ___|___|___|___|___|___|___
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                ___|___|___|___|___|___|___
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                ___|___|___|___|___|___|___
                   |   |   |   |   |   |
                 0 | 0 | 0 | 0 | 0 | 0 | 0
                   |   |   |   |   |   |
Krushan's turn
Enter column number :
4
```

Screen shots

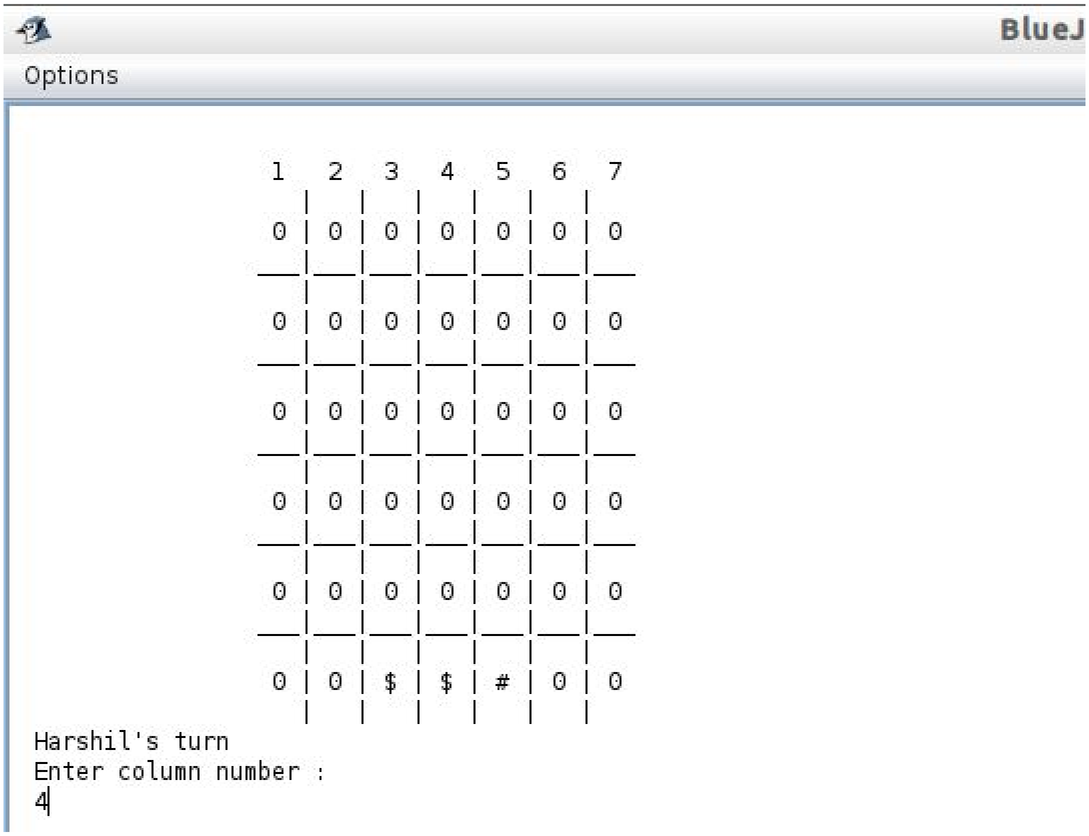


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | $ | 0 | 0 | 0
   |   |   |   |   |   |
Harshil's turn
Enter column number :
5
```

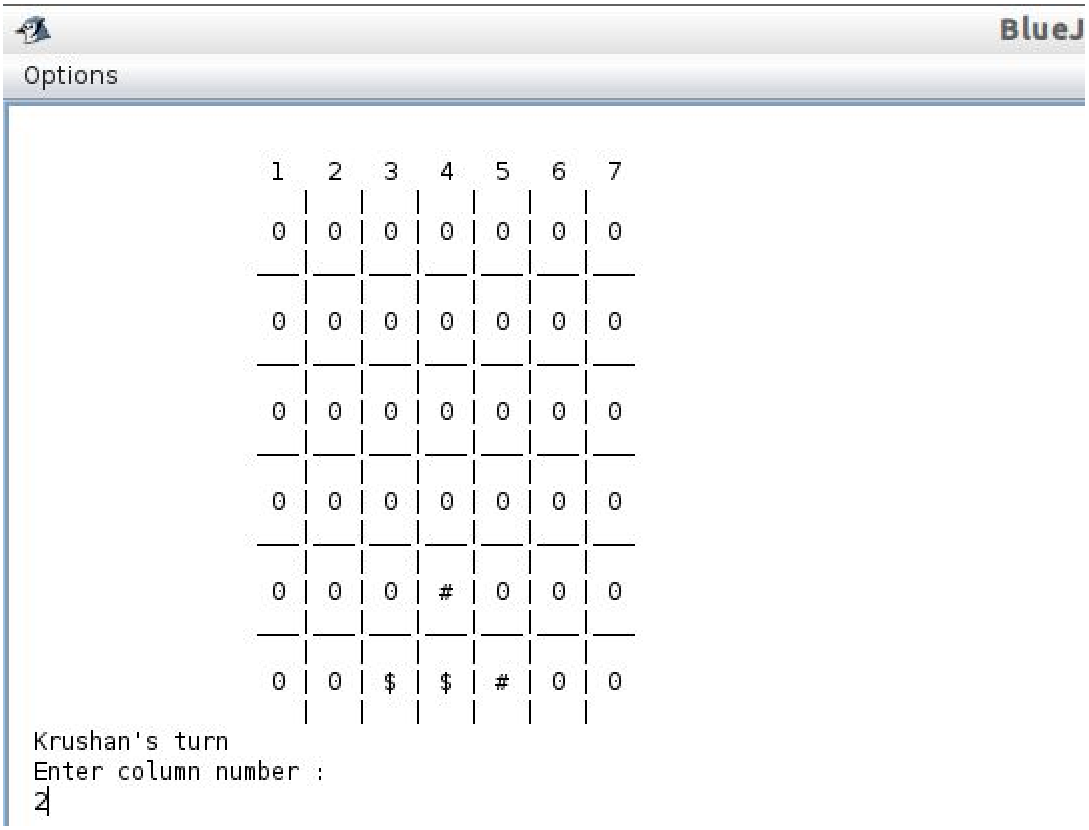


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | $ | # | 0 | 0
   |   |   |   |   |   |
Krushan's turn
Enter column number :
3
```

Screen shots

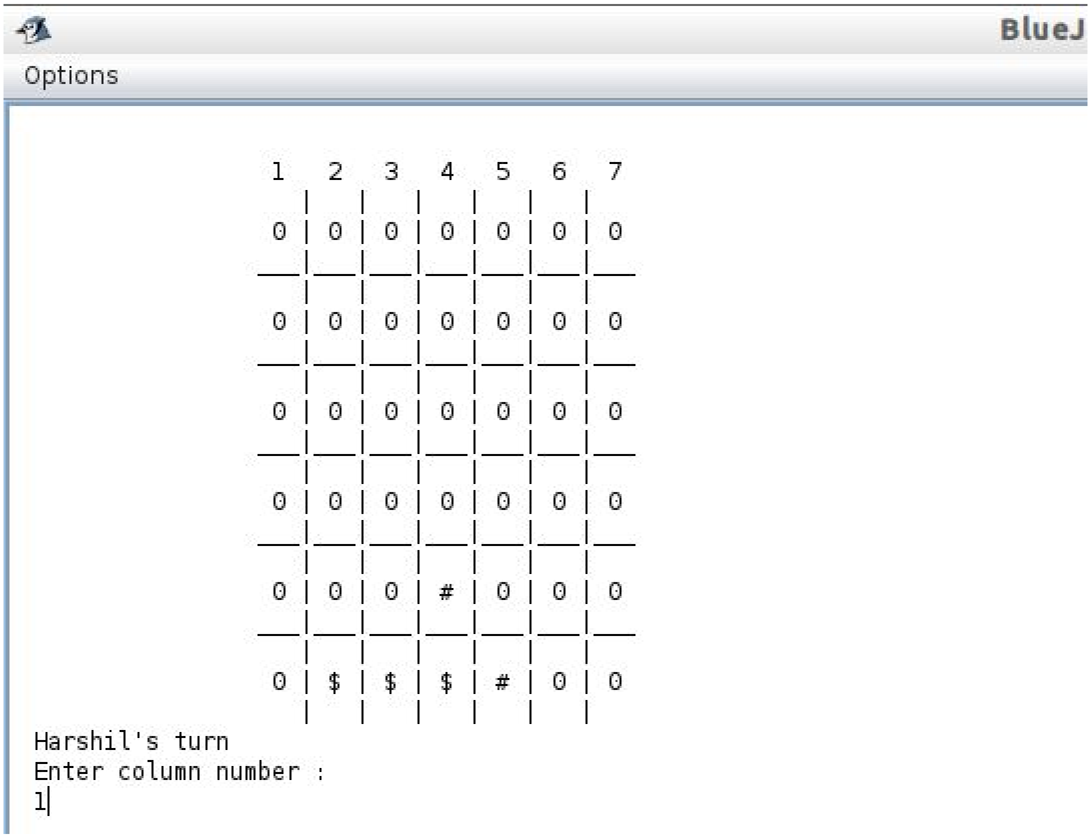


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | $ | $ | # | 0 | 0
   |   |   |   |   |   |
Harshil's turn
Enter column number :
4
```

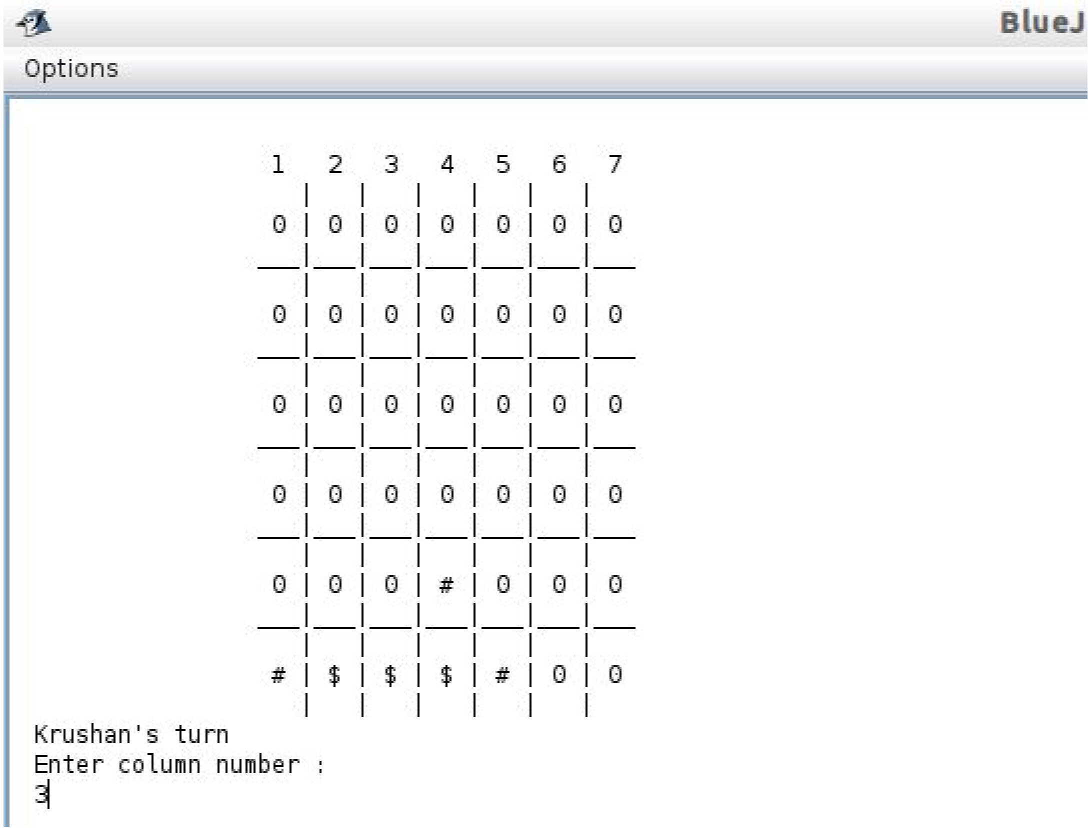


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | # | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | $ | $ | # | 0 | 0
   |   |   |   |   |   |
Krushan's turn
Enter column number :
2
```

Screen shots

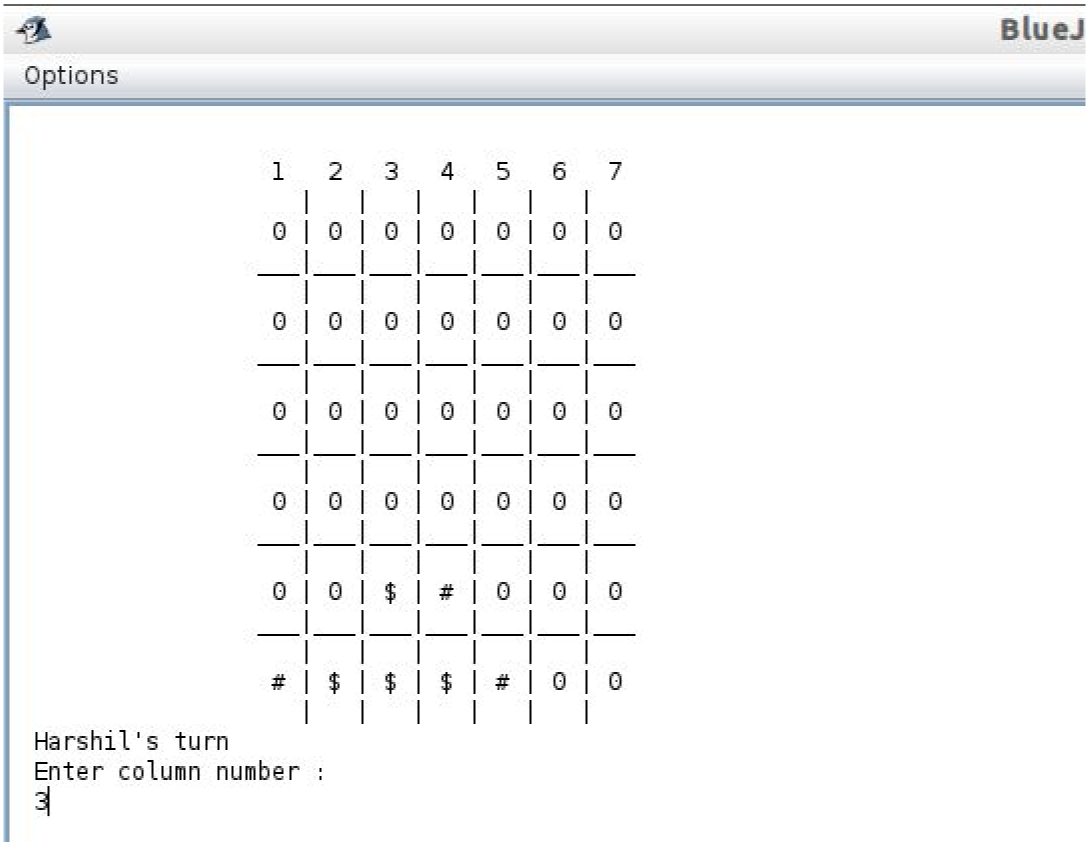


```
  1   2   3   4   5   6   7
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | # | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | $ | $ | $ | # | 0 | 0
    |   |   |   |   |   |
Harshil's turn
Enter column number :
1
```

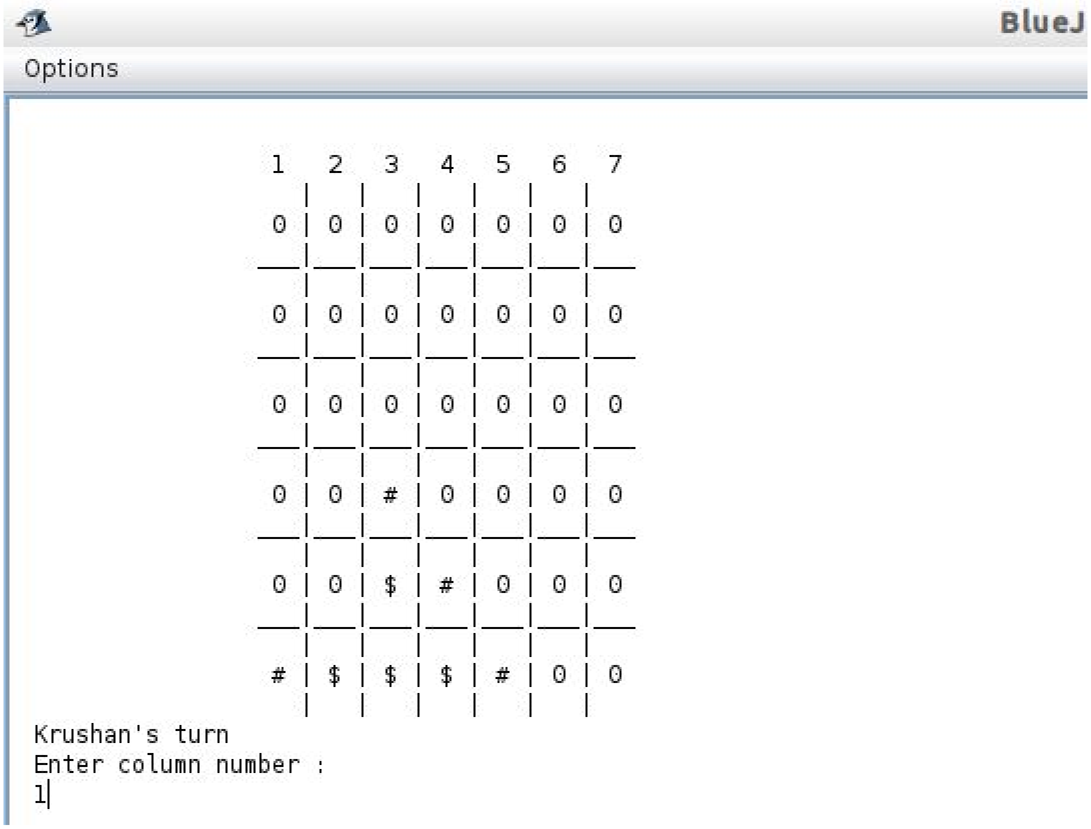


```
  1   2   3   4   5   6   7
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | 0 | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  0 | 0 | 0 | # | 0 | 0 | 0
 ___|___|___|___|___|___|___
    |   |   |   |   |   |
  # | $ | $ | $ | # | 0 | 0
    |   |   |   |   |   |
Krushan's turn
Enter column number :
3
```

Screen shots

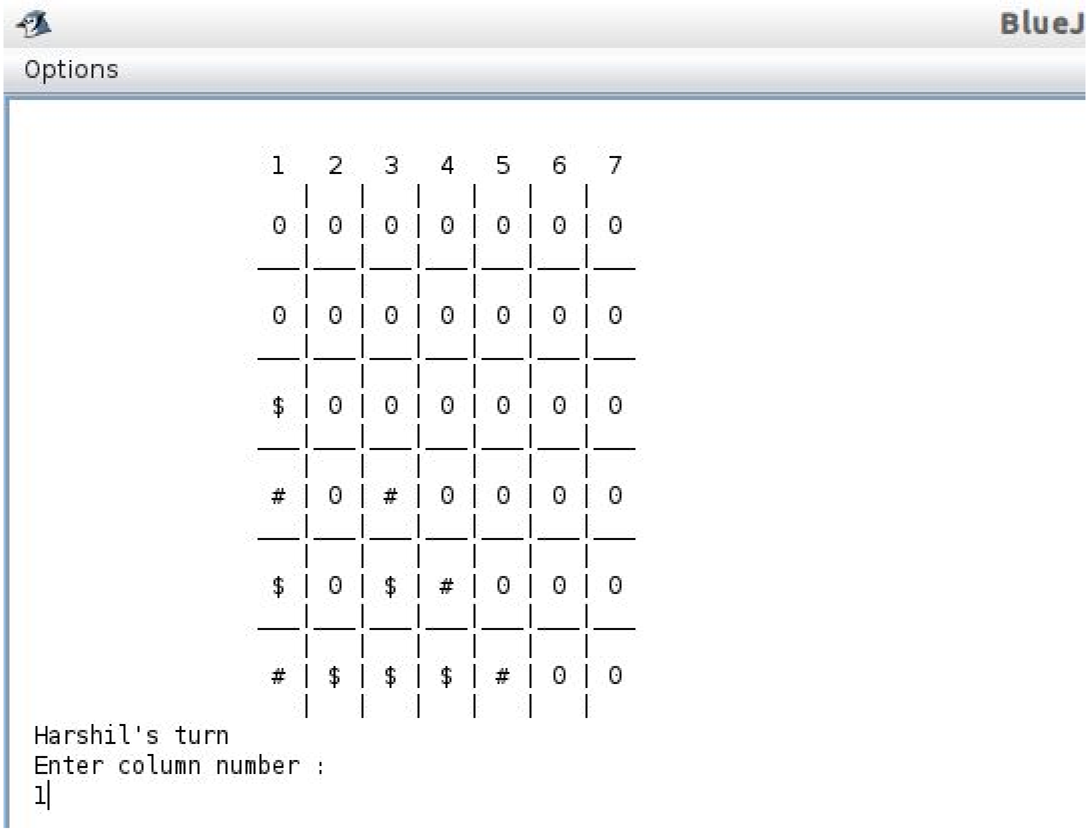


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | $ | # | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 # | $ | $ | $ | # | 0 | 0
   |   |   |   |   |   |
Harshil's turn
Enter column number :
3
```

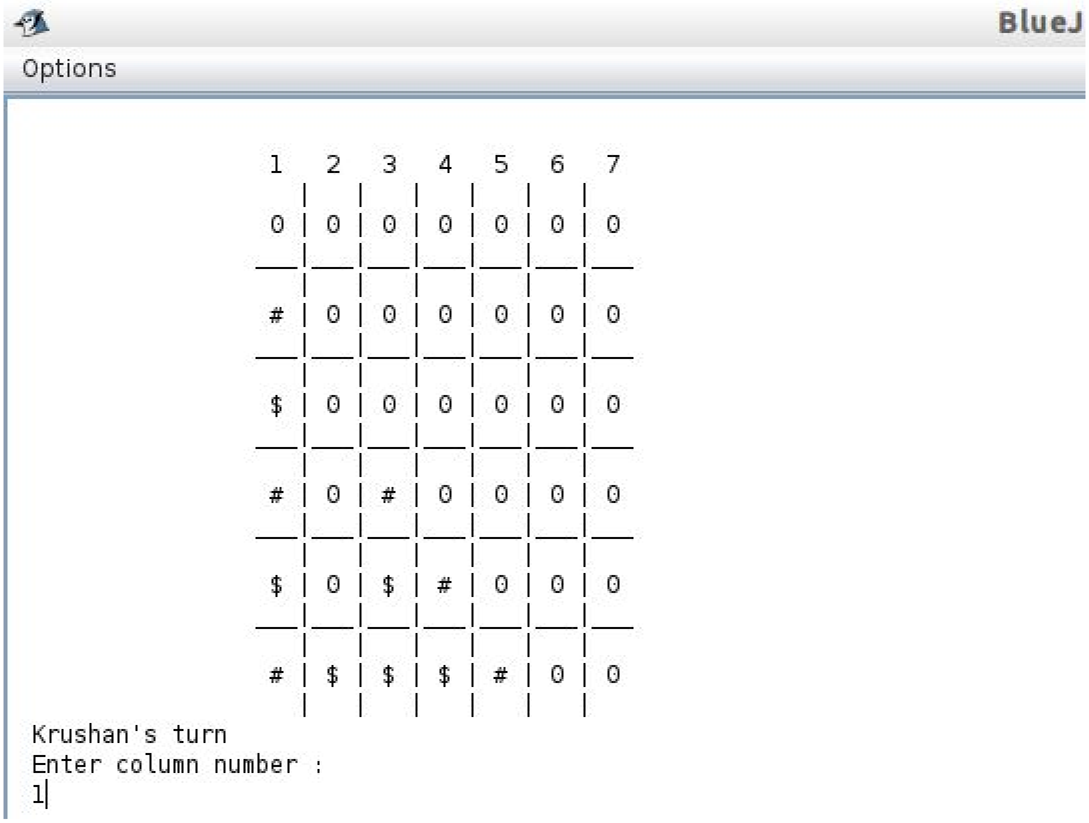


```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | # | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | $ | # | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 # | $ | $ | $ | # | 0 | 0
   |   |   |   |   |   |
Krushan's turn
Enter column number :
1
```

Screen shots

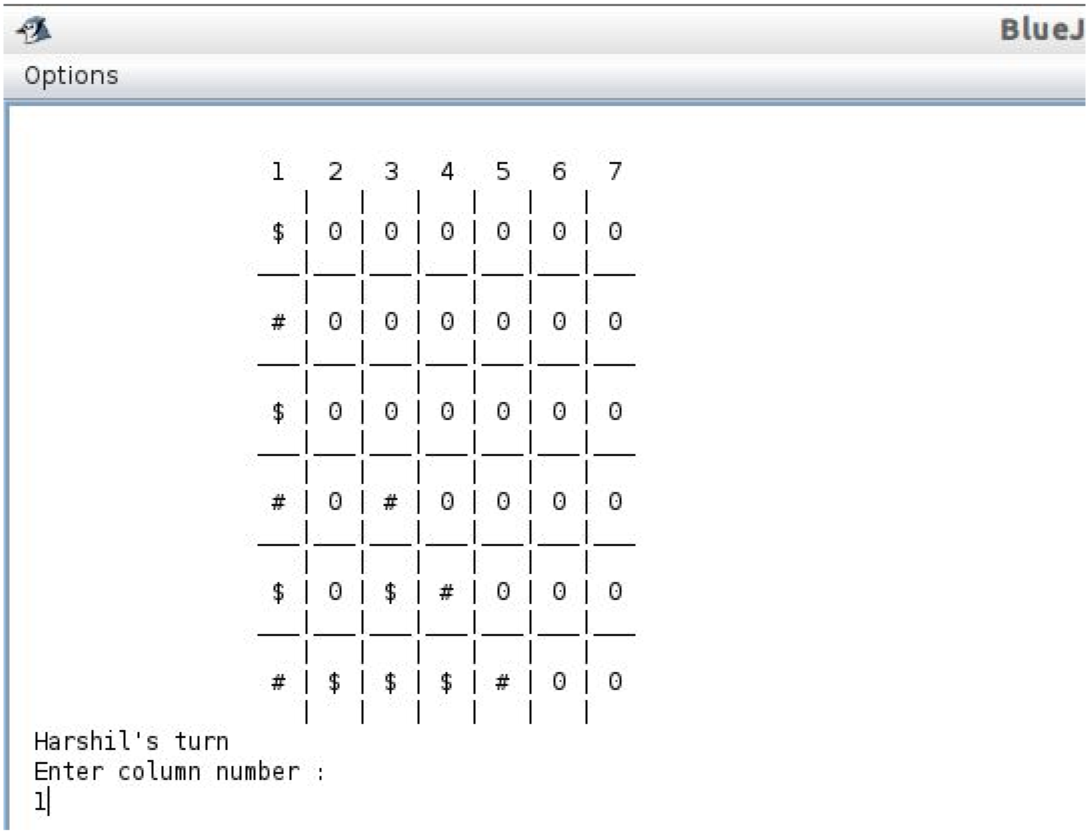


```
 1   2   3   4   5   6   7
    |   |   |   |   |   |
 0  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 0  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 $  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 #  | 0 | # | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 $  | 0 | $ | # | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 #  | $ | $ | $ | # | 0 | 0
    |   |   |   |   |   |
Harshil's turn
Enter column number :
1
```

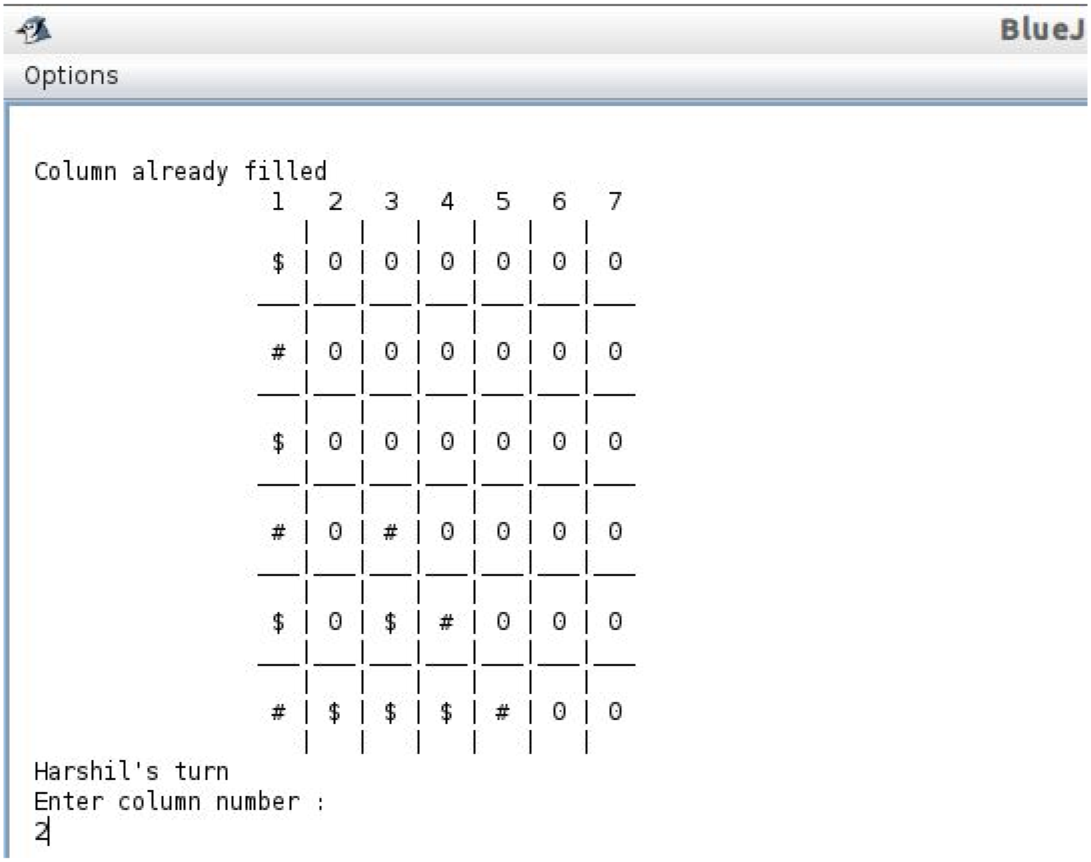


```
 1   2   3   4   5   6   7
    |   |   |   |   |   |
 0  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 #  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 $  | 0 | 0 | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 #  | 0 | # | 0 | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 $  | 0 | $ | # | 0 | 0 | 0
____|___|___|___|___|___|___
    |   |   |   |   |   |
 #  | $ | $ | $ | # | 0 | 0
    |   |   |   |   |   |
Krushan's turn
Enter column number :
1
```

Screen shots

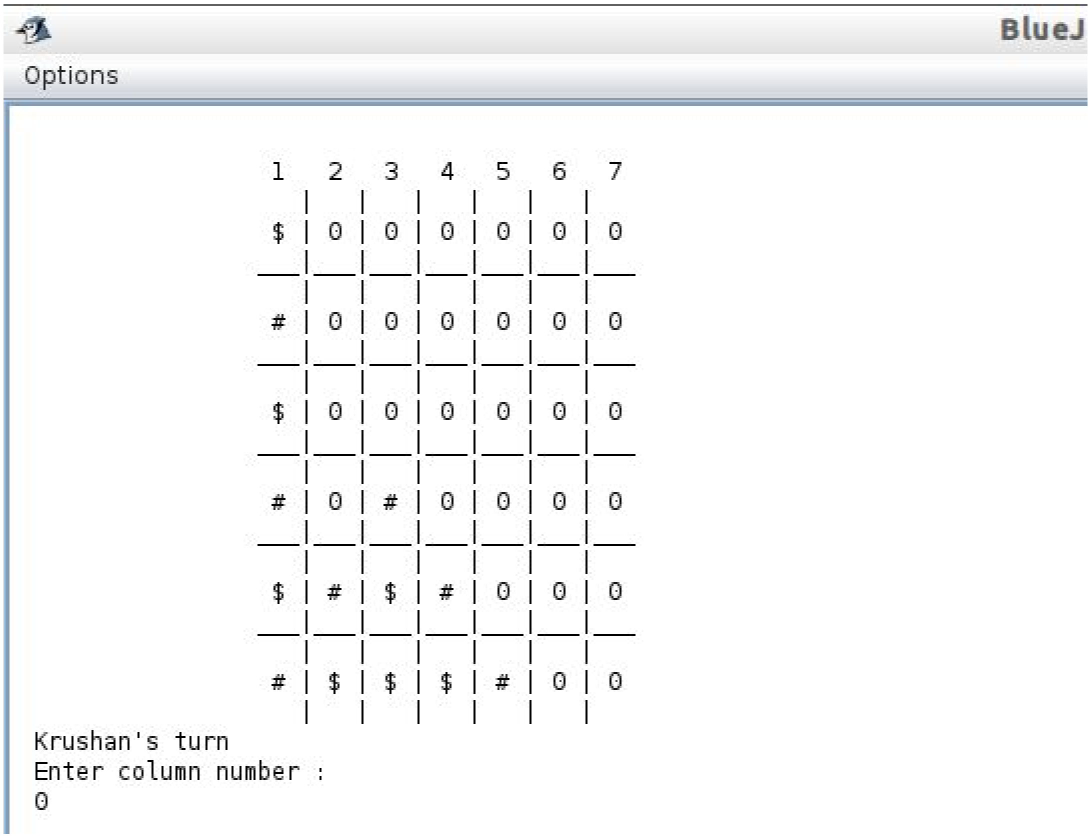


```
                    1   2   3   4   5   6   7
                      |   |   |   |   |   |
                    $ | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    $ | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | 0 | # | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    $ | 0 | $ | # | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | $ | $ | $ | # | 0 | 0
                      |   |   |   |   |   |
Harshil's turn
Enter column number :
1
```

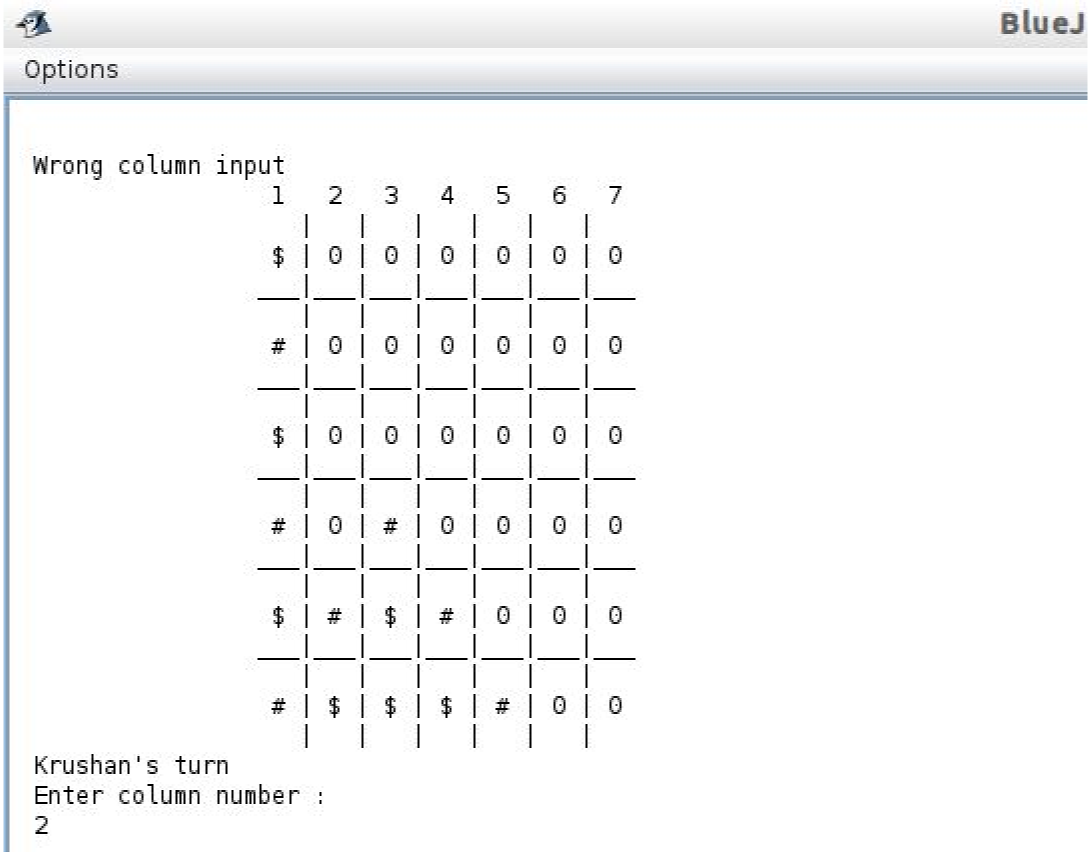


```
Column already filled
                    1   2   3   4   5   6   7
                      |   |   |   |   |   |
                    $ | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    $ | 0 | 0 | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | 0 | # | 0 | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    $ | 0 | $ | # | 0 | 0 | 0
                   ___|___|___|___|___|___|___
                      |   |   |   |   |   |
                    # | $ | $ | $ | # | 0 | 0
                      |   |   |   |   |   |
Harshil's turn
Enter column number :
2
```

Screen shots

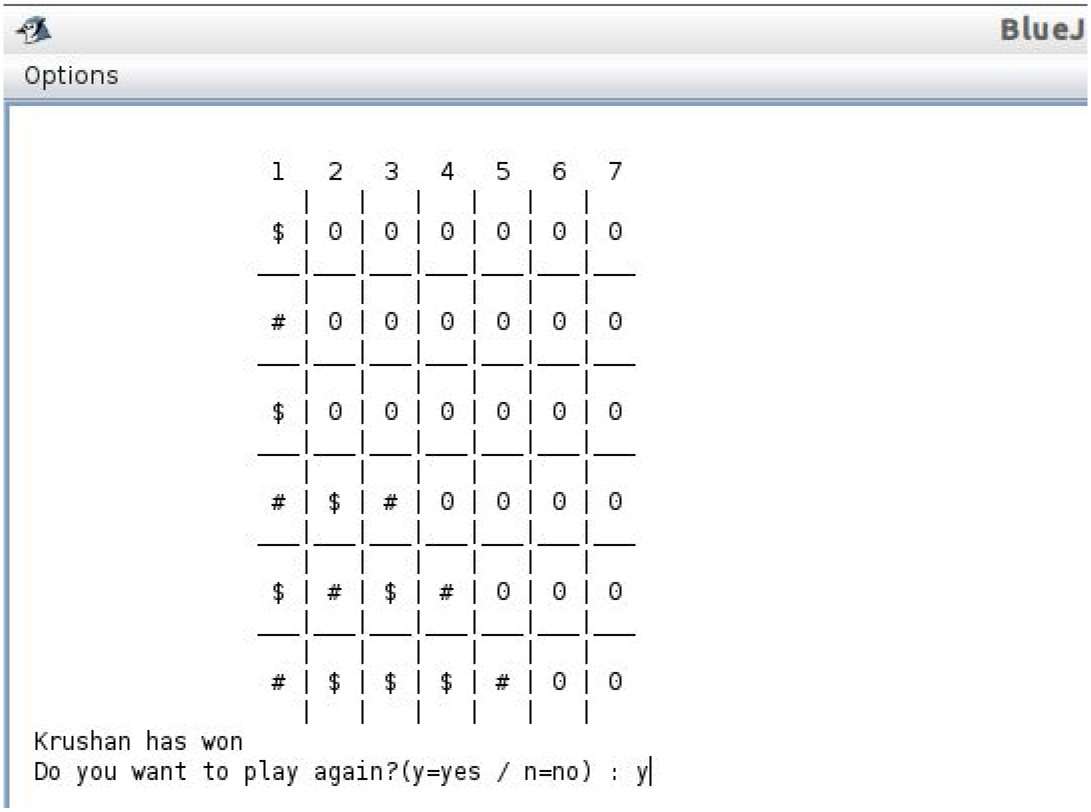


```
                     1   2   3   4   5   6   7
                       |   |   |   |   |   |
                     $ | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     $ | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | 0 | # | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     $ | # | $ | # | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | $ | $ | $ | # | 0 | 0
                       |   |   |   |   |   |
Krushan's turn
Enter column number :
0
```

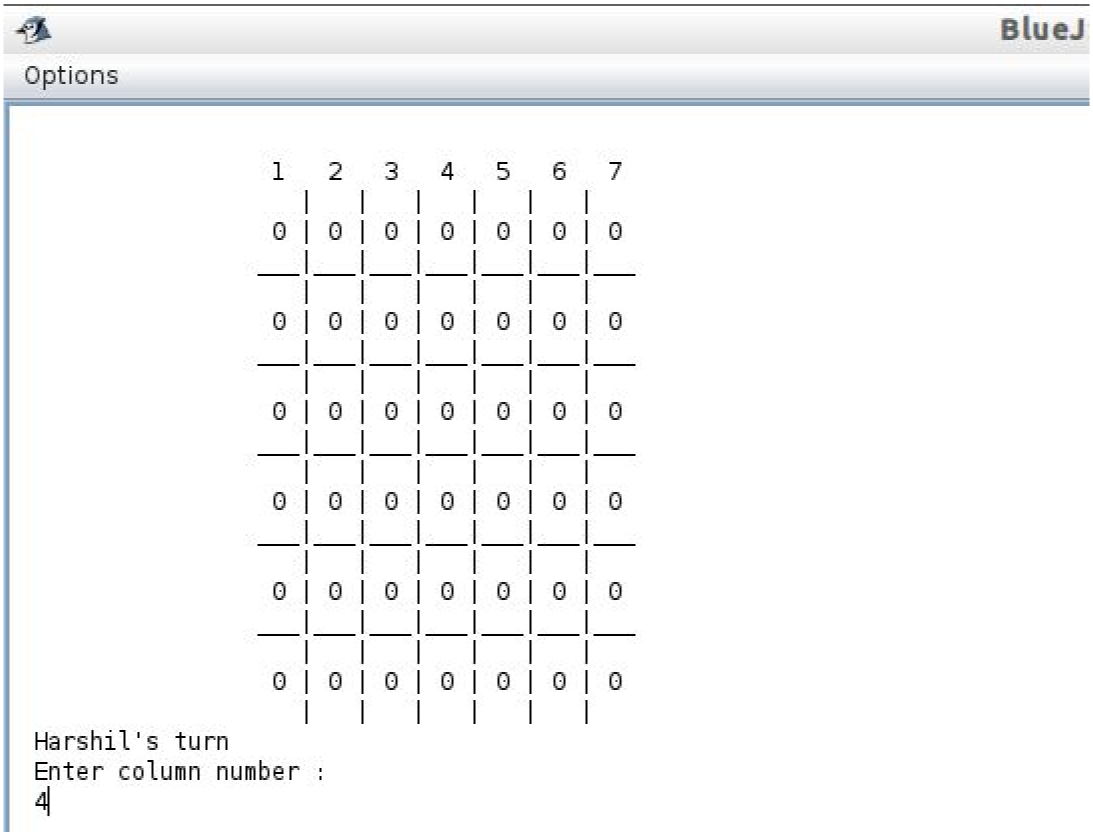


```
Wrong column input
                     1   2   3   4   5   6   7
                       |   |   |   |   |   |
                     $ | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     $ | 0 | 0 | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | 0 | # | 0 | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     $ | # | $ | # | 0 | 0 | 0
                    ___|___|___|___|___|___|___
                       |   |   |   |   |   |
                     # | $ | $ | $ | # | 0 | 0
                       |   |   |   |   |   |
Krushan's turn
Enter column number :
2
```

Screen shots



```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 $ | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 # | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 $ | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 # | $ | # | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 $ | # | $ | # | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 # | $ | $ | $ | # | 0 | 0
   |   |   |   |   |   |
Krushan has won
Do you want to play again?(y=yes / n=no) : y
```



```
 1   2   3   4   5   6   7
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
___|___|___|___|___|___|___
   |   |   |   |   |   |
 0 | 0 | 0 | 0 | 0 | 0 | 0
   |   |   |   |   |   |
Harshil's turn
Enter column number :
4
```

Source Code

Class Name: Start

```
import java.io.*;
import TicTacToe.Tic_Tac_Toe;
import FourInARow.Four_In_A_Row;
/**
* @author Krushan Bauva
*/
public class Start
{
public static void main(String[] args)
{
BufferedReader br;
try
{
System.out.print("\f");
System.out.println("********************************************************************");
System.out.println("Welcome to Arcade Game Box");
System.out.println("********************************************************************");
System.out.println("Enter 1 to play Tic Tac Toe");
System.out.println("Enter 2 to play Four In A Row");
br = new BufferedReader(new InputStreamReader(System.in));
int choice = Integer.parseInt(br.readLine());
if(choice==1)
{
Tic_Tac_Toe.start_Tic_Tac_Toe();
}
else if(choice==2)
{
Four_In_A_Row.start_Four_In_A_Row();
}
else
{
System.out.println("Wrong input");
}
}
catch(Exception e)
{
System.out.println("Sorry an error occured!");
}
}
}
```

Source Code

Package Name: TicTacToe

Class Name: Tic_Tac_Toe

```
package TicTacToe;

import java.io.*;

/**

* @author Krushan Bauva

*/

public class Tic_Tac_Toe

{

        public static int check;

        public static String winner;

        public static String player;

        public static char player_symbol;

        public static String name_player;

        public static String name_player1;

        public static String name_player2;

        public static char posn[]={'0','1', '2', '3', '4', '5', '6', '7', '8', '9'};

        public static BufferedReader br = new BufferedReader (new InputStreamReader
(System.in));

        public static void start_Tic_Tac_Toe()throws IOException

        {

                playerInfo();
```

```
            System.out.print("\f");
            play();
        }
        public static void playerInfo()throws IOException
        {
            System.out.print("Enter name of player 1 'X' : ");
            name_player1 = br.readLine();
            System.out.print("\nEnter name of player 2 'O' : ");
            name_player2 = br.readLine();
            player_symbol = 'X';
            name_player = name_player1;
        }
        public static void currentBoard()
        {
            System.out.println("\n\t\t      |       |");
            System.out.println("\t\t  "+posn[1]+"   |   "+posn[2]+"   |   "+posn[3]);
            System.out.println("\t\t_______|_________|_______");
            System.out.println("\t\t      |       |");
            System.out.println("\t\t  "+posn[4]+"   |   "+posn[5]+"   |   "+posn[6]);
            System.out.println("\t\t_______|_________|_______");
            System.out.println("\t\t      |       |");
            System.out.println("\t\t  "+posn[7]+"   |   "+posn[8]+"   |   "+posn[9]);
            System.out.println("\t\t      |       |");
        }
        public static void play()throws IOException
        {
            check = 0;
        for( ; check == 0 ; )
        {
            currentBoard();
            System.out.println(name_player+"'s turn");
            System.out.println("Enter position : ");
            char choice = (br.readLine()).charAt(0);
            System.out.println("\f");

            if((choice=='1')||(choice=='2')||(choice=='3')||(choice=='4')||(choice=='5')||(choice=='6')||(choice=='7')||(choice=='8')||(choice=='9'))
            {
                changePos(choice);
                check_Winner();
                change_Player();
            }
            else if((choice!='1')||(choice!='2')||(choice!='3')||(choice!='4')||(choice!='5')||(choice!='6')||(choice!='7')||(choice!='8')||(choice!='9'))
            {
                System.out.println("Wrong position input");
            }
        }
        if(winner == name_player1)
        win(name_player1);
        else if (winner == name_player2)
        win(name_player2);
        else if (winner == "none")
        stalemate();
    }
```

```
public static void check_Winner()
{
        //Check if 'X' wins
        if ((posn[1] == 'X'&& posn[2] == 'X'&& posn[3] == 'X')||
                        (posn[4] == 'X'&& posn[5] == 'X'&& posn[6] == 'X')||
                        (posn[7] == 'X'&& posn[8] == 'X'&& posn[9] == 'X')||
                        (posn[1] == 'X'&& posn[4] == 'X'&& posn[7] == 'X')||
                        (posn[2] == 'X'&& posn[5] == 'X'&& posn[8] == 'X')||
                        (posn[3] == 'X'&& posn[6] == 'X'&& posn[9] == 'X')||
                        (posn[1] == 'X'&& posn[5] == 'X'&& posn[9] == 'X')||
                        (posn[3] == 'X'&& posn[5] == 'X'&& posn[7] == 'X'))
        {
                winner = name_player1;
                check = 1;


        }
        //Check if 'O' wins
        else if ((posn[1] == 'O'&& posn[2] == 'O'&& posn[3] == 'O')||
                        (posn[4] == 'O'&& posn[5] == 'O'&& posn[6] == 'O')||
                        (posn[7] == 'O'&& posn[8] == 'O'&& posn[9] == 'O')||
                        (posn[1] == 'O'&& posn[4] == 'O'&& posn[7] == 'O')||
                        (posn[2] == 'O'&& posn[5] == 'O'&& posn[8] == 'O')||
                        (posn[3] == 'O'&& posn[6] == 'O'&& posn[9] == 'O')||
                        (posn[1] == 'O'&& posn[5] == 'O'&& posn[9] == 'O')||
                        (posn[3] == 'O'&& posn[5] == 'O'&& posn[7] == 'O'))
        {
                winner = name_player2;
```

```
                        check = 1;

                }

                //Check if Game is stalemate

                for(int i=1;i<10;i++)

{

if(posn[i]=='X' || posn[i]=='O')

{

if(i==9)

{

winner = "none";

check = 1;

}

}

else

{

break;

}

}

        }

        public static void changePos(char choice)throws IOException

        {

                int flag = 1;

                for(int i=1 ; i<10 ; i++)

                {

                        if(posn[i] == choice)

                        {

                                posn[i]=player_symbol;
                flag = 1;
                break;
            }
            else
                flag = 0;
        }
        if(flag == 0)
        {
            System.out.println("Position already exists");
            play();
        }
    }
    public static void change_Player()
    {
        if(name_player == name_player1)
        {
            name_player = name_player2;
            player_symbol = 'O';
        }

        else
        {
            name_player = name_player1;
            player_symbol = 'X';
        }
    }
    public static void win(String name)throws IOException
```

```
          {
             currentBoard();
System.out.println(name+" has won");
rematch();
}
public static void stalemate()throws IOException
{
currentBoard();
System.out.println("Game is stalemate");
rematch();
}
public static void rematch()throws IOException
{
System.out.print("Do you want to play again?(y=yes / n=no) : ");
String input = br.readLine();
if((input.equalsIgnoreCase("y")) || (input.equalsIgnoreCase("yes")))
{
System.out.println("\f");
reset();
play();
}
else if((input.equalsIgnoreCase("n")) || (input.equalsIgnoreCase("no")))
{
System.out.println("Bye");
System.exit(1);
}
else
{
System.out.println("Sorry wrong input");
rematch();
}
}
public static void reset()
{
posn= new char[] {'0','1', '2', '3', '4', '5', '6', '7', '8', '9'};
}
}
```

Source Code

Package Name: FourInARow

Class Name: Four_In_A_Row

```
package FourInARow;
import java.io.*;
/**
* @author Krushan Bauva
*/
public class Four_In_A_Row
{
public static String winner;
public static int check;
public static String player;
public static char player_symbol;
public static String name_player;
public static String name_player1;
public static String name_player2;
public static char[][] posn = new char[7][8];
public static BufferedReader br = new BufferedReader (new InputStreamReader
(System.in));
public static void start_Four_In_A_Row()throws IOException
{
```

```
playerInfo();
init_Pos();
System.out.print("\f");
play();
}
public static void play()throws IOException
{
check = 0;
for( ; check == 0 ; )
{
currentBoard();
System.out.println(name_player+"'s turn");
System.out.println("Enter column number : ");
char choice = (br.readLine()).charAt(0);
int num = (int) choice;
num = num - 48;
System.out.println("\f");
if((num>0) && (num<8))
{
change_Pos(num);
check_Winner();
change_Player();
}
else
{
System.out.println("Wrong column input");
}
```

```
}
if(winner == name_player1)
win(name_player1);
else if (winner == name_player2)
win(name_player2);
else if (winner == "none")
stalemate();
}
public static void change_Pos(int num)throws IOException
{
int flag = 1;
for(int n=6 ; n>0 ; n--)
{
if(posn[n][num] == '0')
{
posn[n][num] = player_symbol;
flag = 1;
break;
}
else if(posn[1][num]!= '0')
{
flag = 0;
break;
}
else
{
flag = 1;
}
}
if(flag == 0)
{
System.out.println("Column already filled");
play();
}
}
public static void check_Winner()
{
if(((posn[1][1]=='$') && (posn[2][1]=='$') && (posn[3][1]=='$') && (posn[4][1]=='$'))||
((posn[2][1]=='$') && (posn[3][1]=='$') && (posn[4][1]=='$') && (posn[5][1]=='$'))||
((posn[3][1]=='$') && (posn[4][1]=='$') && (posn[5][1]=='$') && (posn[6][1]=='$'))||
((posn[1][2]=='$') && (posn[2][2]=='$') && (posn[3][2]=='$') && (posn[4][2]=='$'))||
((posn[2][2]=='$') && (posn[3][2]=='$') && (posn[4][2]=='$') && (posn[5][2]=='$'))||
((posn[3][2]=='$') && (posn[4][2]=='$') && (posn[5][2]=='$') && (posn[6][2]=='$'))||
((posn[1][3]=='$') && (posn[2][3]=='$') && (posn[3][3]=='$') && (posn[4][3]=='$'))||
((posn[2][3]=='$') && (posn[3][3]=='$') && (posn[4][3]=='$') && (posn[5][3]=='$'))||
((posn[3][3]=='$') && (posn[4][3]=='$') && (posn[5][3]=='$') && (posn[6][3]=='$'))||
((posn[1][4]=='$') && (posn[2][4]=='$') && (posn[3][4]=='$') && (posn[4][4]=='$'))||
((posn[2][4]=='$') && (posn[3][4]=='$') && (posn[4][4]=='$') && (posn[5][4]=='$'))||
((posn[3][4]=='$') && (posn[4][4]=='$') && (posn[5][4]=='$') && (posn[6][4]=='$'))||
((posn[1][5]=='$') && (posn[2][5]=='$') && (posn[3][5]=='$') && (posn[4][5]=='$'))||
((posn[2][5]=='$') && (posn[3][5]=='$') && (posn[4][5]=='$') && (posn[5][5]=='$'))||
((posn[3][5]=='$') && (posn[4][5]=='$') && (posn[5][5]=='$') && (posn[6][5]=='$'))||
((posn[1][6]=='$') && (posn[2][6]=='$') && (posn[3][6]=='$') && (posn[4][6]=='$'))||
((posn[2][6]=='$') && (posn[3][6]=='$') && (posn[4][6]=='$') && (posn[5][6]=='$'))||
((posn[3][6]=='$') && (posn[4][6]=='$') && (posn[5][6]=='$') && (posn[6][6]=='$'))||

((posn[1][7]=='$') && (posn[2][7]=='$') && (posn[3][7]=='$') && (posn[4][7]=='$'))||

((posn[2][7]=='$') && (posn[3][7]=='$') && (posn[4][7]=='$') && (posn[5][7]=='$'))||

((posn[3][7]=='$') && (posn[4][7]=='$') && (posn[5][7]=='$') && (posn[6][7]=='$'))||

((posn[1][1]=='$') && (posn[1][2]=='$') && (posn[1][3]=='$') && (posn[1][4]=='$'))||

((posn[1][2]=='$') && (posn[1][3]=='$') && (posn[1][4]=='$') && (posn[1][5]=='$'))||

((posn[1][3]=='$') && (posn[1][4]=='$') && (posn[1][5]=='$') && (posn[1][6]=='$'))||

((posn[1][4]=='$') && (posn[1][5]=='$') && (posn[1][6]=='$') && (posn[1][7]=='$'))||

((posn[2][1]=='$') && (posn[2][2]=='$') && (posn[2][3]=='$') && (posn[2][4]=='$'))||

((posn[2][2]=='$') && (posn[2][3]=='$') && (posn[2][4]=='$') && (posn[2][5]=='$'))||

((posn[2][3]=='$') && (posn[2][4]=='$') && (posn[2][5]=='$') && (posn[2][6]=='$'))||

((posn[2][4]=='$') && (posn[2][5]=='$') && (posn[2][6]=='$') && (posn[2][7]=='$'))||

((posn[3][1]=='$') && (posn[3][2]=='$') && (posn[3][3]=='$') && (posn[3][4]=='$'))||

((posn[3][2]=='$') && (posn[3][3]=='$') && (posn[3][4]=='$') && (posn[3][5]=='$'))||

((posn[3][3]=='$') && (posn[3][4]=='$') && (posn[3][5]=='$') && (posn[3][6]=='$'))||

((posn[3][4]=='$') && (posn[3][5]=='$') && (posn[3][6]=='$') && (posn[3][7]=='$'))||

((posn[4][1]=='$') && (posn[4][2]=='$') && (posn[4][3]=='$') && (posn[4][4]=='$'))||

((posn[4][2]=='$') && (posn[4][3]=='$') && (posn[4][4]=='$') && (posn[4][5]=='$'))||

((posn[4][3]=='$') && (posn[4][4]=='$') && (posn[4][5]=='$') && (posn[4][6]=='$'))||

((posn[4][4]=='$') && (posn[4][5]=='$') && (posn[4][6]=='$') && (posn[4][7]=='$'))||

((posn[5][1]=='$') && (posn[5][2]=='$') && (posn[5][3]=='$') && (posn[5][4]=='$'))||

((posn[5][2]=='$') && (posn[5][3]=='$') && (posn[5][4]=='$') && (posn[5][5]=='$'))||

((posn[5][3]=='$') && (posn[5][4]=='$') && (posn[5][5]=='$') && (posn[5][6]=='$'))||

((posn[5][4]=='$') && (posn[5][5]=='$') && (posn[5][6]=='$') && (posn[5][7]=='$'))||

((posn[6][1]=='$') && (posn[6][2]=='$') && (posn[6][3]=='$') && (posn[6][4]=='$'))||

((posn[6][2]=='$') && (posn[6][3]=='$') && (posn[6][4]=='$') && (posn[6][5]=='$'))||

((posn[6][3]=='$') && (posn[6][4]=='$') && (posn[6][5]=='$') && (posn[6][6]=='$'))||
((posn[6][4]=='$') && (posn[6][5]=='$') && (posn[6][6]=='$') && (posn[6][7]=='$'))||
((posn[1][1]=='$') && (posn[2][2]=='$') && (posn[3][3]=='$') && (posn[4][4]=='$'))||
((posn[1][2]=='$') && (posn[2][3]=='$') && (posn[3][4]=='$') && (posn[4][5]=='$'))||
((posn[1][3]=='$') && (posn[2][4]=='$') && (posn[3][5]=='$') && (posn[4][6]=='$'))||
((posn[2][1]=='$') && (posn[3][2]=='$') && (posn[4][3]=='$') && (posn[5][4]=='$'))||
((posn[3][1]=='$') && (posn[4][2]=='$') && (posn[5][3]=='$') && (posn[6][4]=='$'))||
((posn[2][2]=='$') && (posn[3][3]=='$') && (posn[4][4]=='$') && (posn[5][5]=='$'))||
((posn[3][3]=='$') && (posn[4][4]=='$') && (posn[5][5]=='$') && (posn[6][6]=='$'))||
((posn[2][3]=='$') && (posn[3][4]=='$') && (posn[4][5]=='$') && (posn[5][6]=='$'))||
((posn[3][2]=='$') && (posn[4][3]=='$') && (posn[5][4]=='$') && (posn[6][5]=='$'))||
((posn[1][6]=='$') && (posn[2][5]=='$') && (posn[3][4]=='$') && (posn[4][3]=='$'))||
((posn[2][5]=='$') && (posn[3][4]=='$') && (posn[4][3]=='$') && (posn[5][2]=='$'))||
((posn[3][4]=='$') && (posn[4][3]=='$') && (posn[5][2]=='$') && (posn[6][1]=='$'))||
((posn[2][6]=='$') && (posn[3][5]=='$') && (posn[4][4]=='$') && (posn[5][3]=='$'))||
((posn[3][5]=='$') && (posn[4][4]=='$') && (posn[5][3]=='$') && (posn[6][2]=='$'))||
((posn[3][6]=='$') && (posn[4][5]=='$') && (posn[5][4]=='$') && (posn[6][3]=='$'))||
((posn[1][5]=='$') && (posn[2][4]=='$') && (posn[3][3]=='$') && (posn[4][2]=='$'))||
((posn[2][4]=='$') && (posn[3][3]=='$') && (posn[4][2]=='$') && (posn[5][1]=='$'))||
((posn[1][4]=='$') && (posn[2][3]=='$') && (posn[3][2]=='$') && (posn[4][1]=='$')))
{
winner = name_player1;
check = 1;
}
else if(((posn[1][1]=='#') && (posn[2][1]=='#') && (posn[3][1]=='#') && (posn[4][1]=='#'))||
((posn[2][1]=='#') && (posn[3][1]=='#') && (posn[4][1]=='#') && (posn[5][1]=='#'))||
((posn[3][1]=='#') && (posn[4][1]=='#') && (posn[5][1]=='#') && (posn[6][1]=='#'))||
((posn[1][2]=='#') && (posn[2][2]=='#') && (posn[3][2]=='#') && (posn[4][2]=='#'))||
((posn[2][2]=='#') && (posn[3][2]=='#') && (posn[4][2]=='#') && (posn[5][2]=='#'))||
((posn[3][2]=='#') && (posn[4][2]=='#') && (posn[5][2]=='#') && (posn[6][2]=='#'))||
((posn[1][3]=='#') && (posn[2][3]=='#') && (posn[3][3]=='#') && (posn[4][3]=='#'))||
((posn[2][3]=='#') && (posn[3][3]=='#') && (posn[4][3]=='#') && (posn[5][3]=='#'))||
((posn[3][3]=='#') && (posn[4][3]=='#') && (posn[5][3]=='#') && (posn[6][3]=='#'))||
((posn[1][4]=='#') && (posn[2][4]=='#') && (posn[3][4]=='#') && (posn[4][4]=='#'))||
((posn[2][4]=='#') && (posn[3][4]=='#') && (posn[4][4]=='#') && (posn[5][4]=='#'))||
((posn[3][4]=='#') && (posn[4][4]=='#') && (posn[5][4]=='#') && (posn[6][4]=='#'))||
((posn[1][5]=='#') && (posn[2][5]=='#') && (posn[3][5]=='#') && (posn[4][5]=='#'))||
((posn[2][5]=='#') && (posn[3][5]=='#') && (posn[4][5]=='#') && (posn[5][5]=='#'))||
((posn[3][5]=='#') && (posn[4][5]=='#') && (posn[5][5]=='#') && (posn[6][5]=='#'))||
((posn[1][6]=='#') && (posn[2][6]=='#') && (posn[3][6]=='#') && (posn[4][6]=='#'))||
((posn[2][6]=='#') && (posn[3][6]=='#') && (posn[4][6]=='#') && (posn[5][6]=='#'))||
((posn[3][6]=='#') && (posn[4][6]=='#') && (posn[5][6]=='#') && (posn[6][6]=='#'))||
((posn[1][7]=='#') && (posn[2][7]=='#') && (posn[3][7]=='#') && (posn[4][7]=='#'))||
((posn[2][7]=='#') && (posn[3][7]=='#') && (posn[4][7]=='#') && (posn[5][7]=='#'))||
((posn[3][7]=='#') && (posn[4][7]=='#') && (posn[5][7]=='#') && (posn[6][7]=='#'))||
((posn[1][1]=='#') && (posn[1][2]=='#') && (posn[1][3]=='#') && (posn[1][4]=='#'))||
((posn[1][2]=='#') && (posn[1][3]=='#') && (posn[1][4]=='#') && (posn[1][5]=='#'))||
((posn[1][3]=='#') && (posn[1][4]=='#') && (posn[1][5]=='#') && (posn[1][6]=='#'))||
((posn[1][4]=='#') && (posn[1][5]=='#') && (posn[1][6]=='#') && (posn[1][7]=='#'))||
((posn[2][1]=='#') && (posn[2][2]=='#') && (posn[2][3]=='#') && (posn[2][4]=='#'))||
((posn[2][2]=='#') && (posn[2][3]=='#') && (posn[2][4]=='#') && (posn[2][5]=='#'))||
((posn[2][3]=='#') && (posn[2][4]=='#') && (posn[2][5]=='#') && (posn[2][6]=='#'))||
((posn[2][4]=='#') && (posn[2][5]=='#') && (posn[2][6]=='#') && (posn[2][7]=='#'))||
((posn[3][1]=='#') && (posn[3][2]=='#') && (posn[3][3]=='#') && (posn[3][4]=='#'))||
((posn[3][2]=='#') && (posn[3][3]=='#') && (posn[3][4]=='#') && (posn[3][5]=='#'))||
((posn[3][3]=='#') && (posn[3][4]=='#') && (posn[3][5]=='#') && (posn[3][6]=='#'))||
((posn[3][4]=='#') && (posn[3][5]=='#') && (posn[3][6]=='#') && (posn[3][7]=='#'))||
((posn[4][1]=='#') && (posn[4][2]=='#') && (posn[4][3]=='#') && (posn[4][4]=='#'))||
((posn[4][2]=='#') && (posn[4][3]=='#') && (posn[4][4]=='#') && (posn[4][5]=='#'))||
((posn[4][3]=='#') && (posn[4][4]=='#') && (posn[4][5]=='#') && (posn[4][6]=='#'))||
((posn[4][4]=='#') && (posn[4][5]=='#') && (posn[4][6]=='#') && (posn[4][7]=='#'))||
((posn[5][1]=='#') && (posn[5][2]=='#') && (posn[5][3]=='#') && (posn[5][4]=='#'))||
((posn[5][2]=='#') && (posn[5][3]=='#') && (posn[5][4]=='#') && (posn[5][5]=='#'))||
((posn[5][3]=='#') && (posn[5][4]=='#') && (posn[5][5]=='#') && (posn[5][6]=='#'))||
((posn[5][4]=='#') && (posn[5][5]=='#') && (posn[5][6]=='#') && (posn[5][7]=='#'))||
((posn[6][1]=='#') && (posn[6][2]=='#') && (posn[6][3]=='#') && (posn[6][4]=='#'))||
((posn[6][2]=='#') && (posn[6][3]=='#') && (posn[6][4]=='#') && (posn[6][5]=='#'))||
((posn[6][3]=='#') && (posn[6][4]=='#') && (posn[6][5]=='#') && (posn[6][6]=='#'))||
((posn[6][4]=='#') && (posn[6][5]=='#') && (posn[6][6]=='#') && (posn[6][7]=='#'))||
((posn[1][1]=='#') && (posn[2][2]=='#') && (posn[3][3]=='#') && (posn[4][4]=='#'))||
((posn[1][2]=='#') && (posn[2][3]=='#') && (posn[3][4]=='#') && (posn[4][5]=='#'))||
((posn[1][3]=='#') && (posn[2][4]=='#') && (posn[3][5]=='#') && (posn[4][6]=='#'))||
((posn[2][1]=='#') && (posn[3][2]=='#') && (posn[4][3]=='#') && (posn[5][4]=='#'))||
((posn[3][1]=='#') && (posn[4][2]=='#') && (posn[5][3]=='#') && (posn[6][4]=='#'))||
((posn[2][2]=='#') && (posn[3][3]=='#') && (posn[4][4]=='#') && (posn[5][5]=='#'))||
((posn[3][3]=='#') && (posn[4][4]=='#') && (posn[5][5]=='#') && (posn[6][6]=='#'))||
((posn[2][3]=='#') && (posn[3][4]=='#') && (posn[4][5]=='#') && (posn[5][6]=='#'))||
((posn[3][2]=='#') && (posn[4][3]=='#') && (posn[5][4]=='#') && (posn[6][5]=='#'))||
((posn[1][6]=='#') && (posn[2][5]=='#') && (posn[3][4]=='#') && (posn[4][3]=='#'))||
((posn[2][5]=='#') && (posn[3][4]=='#') && (posn[4][3]=='#') && (posn[5][2]=='#'))||
((posn[3][4]=='#') && (posn[4][3]=='#') && (posn[5][2]=='#') && (posn[6][1]=='#'))||
((posn[2][6]=='#') && (posn[3][5]=='#') && (posn[4][4]=='#') && (posn[5][3]=='#'))||
((posn[3][5]=='#') && (posn[4][4]=='#') && (posn[5][3]=='#') && (posn[6][2]=='#'))||
((posn[3][6]=='#') && (posn[4][5]=='#') && (posn[5][4]=='#') && (posn[6][3]=='#'))||
((posn[1][5]=='#') && (posn[2][4]=='#') && (posn[3][3]=='#') && (posn[4][2]=='#'))||
((posn[2][4]=='#') && (posn[3][3]=='#') && (posn[4][2]=='#') && (posn[5][1]=='#'))||
((posn[1][4]=='#') && (posn[2][3]=='#') && (posn[3][2]=='#') && (posn[4][1]=='#')))
{
winner = name_player2;
check = 1;
}
for(int i=6 ; i>0 ; i--)
{
for(int n=1 ; n<8 ; n++)
{
if((posn[i][n]=='$') || (posn[i][n]=='#'))
{
if(i==1 && n==7)
{
winner = "none";
check = 1;
}
}
else
{
break;
}
}
}
```

```
}

public static void currentBoard()

{

System.out.println("\t\t
"+posn[0][1]+""+posn[0][2]+""+posn[0][3]+""+posn[0][4]+""+posn[0][5]+""+posn[0][6]+""+
posn[0][7]);

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[1][1]+" | "+posn[1][2]+" | "+posn[1][3]+" | "+posn[1][4]+" |
"+posn[1][5]+" | "+posn[1][6]+" | "+posn[1][7]);

System.out.println("\t\t___|___|___|___|___|___|___");

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[2][1]+" | "+posn[2][2]+" | "+posn[2][3]+" | "+posn[2][4]+" |
"+posn[2][5]+" | "+posn[2][6]+" | "+posn[2][7]);

System.out.println("\t\t___|___|___|___|___|___|___");

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[3][1]+" | "+posn[3][2]+" | "+posn[3][3]+" | "+posn[3][4]+" |
"+posn[3][5]+" | "+posn[3][6]+" | "+posn[3][7]);

System.out.println("\t\t___|___|___|___|___|___|___");

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[4][1]+" | "+posn[4][2]+" | "+posn[4][3]+" | "+posn[4][4]+" |
"+posn[4][5]+" | "+posn[4][6]+" | "+posn[4][7]);

System.out.println("\t\t___|___|___|___|___|___|___");

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[5][1]+" | "+posn[5][2]+" | "+posn[5][3]+" | "+posn[5][4]+" |
"+posn[5][5]+" | "+posn[5][6]+" | "+posn[5][7]);

System.out.println("\t\t___|___|___|___|___|___|___");

System.out.println("\t\t   |   |   |   |   |   |");

System.out.println("\t\t "+posn[6][1]+" | "+posn[6][2]+" | "+posn[6][3]+" | "+posn[6][4]+" |
"+posn[6][5]+" | "+posn[6][6]+" | "+posn[6][7]);

System.out.println("\t\t   |   |   |   |   |   |");

}
```

```
public static void playerInfo()throws IOException

{

System.out.print("Enter name of player 1 '$' : ");

name_player1 = br.readLine();

System.out.print("\nEnter name of player 2 '#' : ");

name_player2 = br.readLine();

player_symbol = '$';

name_player = name_player1;

}

public static void init_Pos()

{

for(int i=1 ; i<7 ; i++)

{

for(int n=1 ; n<8 ; n++)

{

posn[i][n] = '0';

}

}

posn[0][1] = '1';

posn[0][2] = '2';

posn[0][3] = '3';

posn[0][4] = '4';

posn[0][5] = '5';

posn[0][6] = '6';

posn[0][7] = '7';

}

public static void change_Player()
```

```
{
if(name_player == name_player1)
{
name_player = name_player2;
player_symbol = '#';
}

else
{
name_player = name_player1;
player_symbol = '$';
}
}
public static void win(String name)throws IOException
{
currentBoard();
System.out.println(name+" has won");
rematch();
}
public static void stalemate()throws IOException
{
currentBoard();
System.out.println("Game is stalemate");
rematch();
}
public static void rematch()throws IOException
{
```

```
System.out.print("Do you want to play again?(y=yes / n=no) : ");
String input = br.readLine();
if((input.equalsIgnoreCase("y")) || (input.equalsIgnoreCase("yes")))
{
System.out.println("\f");
init_Pos();
play();
}
else if((input.equalsIgnoreCase("n")) || (input.equalsIgnoreCase("no")))
{
System.out.println("Bye");
System.exit(0);
}
else
{
System.out.println("Sorry wrong input");
rematch();
}
}
}
```

Bibliography

Computer Applications by Sumita Arora
Total Computers by Morning Star
Java : The Complete Reference by Herbert Schildt
Java Documentation at http://docs.oracle.com/javase/7/docs/api/