<u>***Daniel Elias Becerra A01208905***</u>

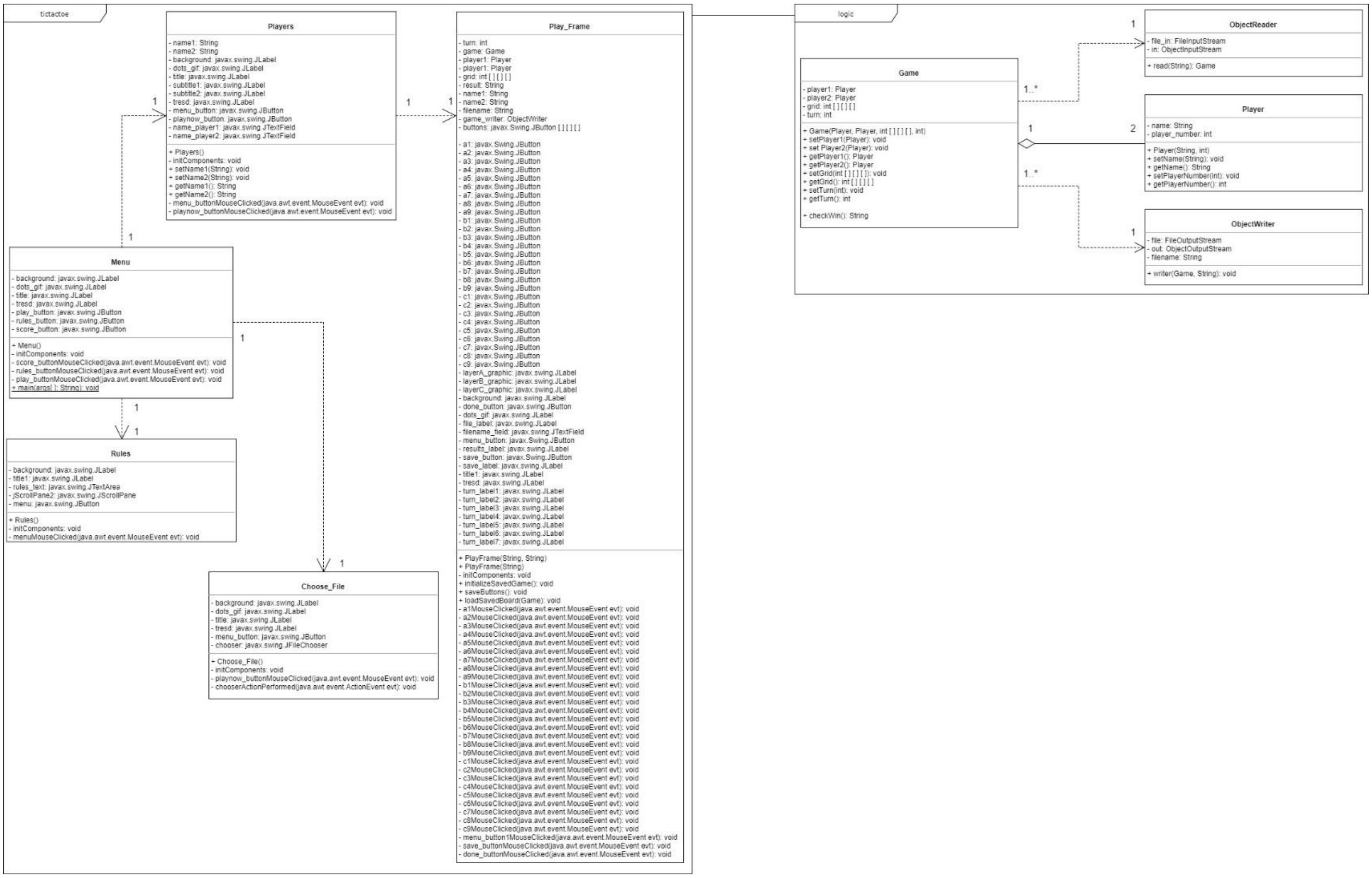
Individual Project

Description of the program's functionality:

The program simulates a 3D Tic Tac Toe separated into 3 layers. The main Menu lets the user choose between Rules, Play and Open. The Rules button moves the user to a Rules Interface that tells them the rules of the 3D Tic Tac Toe. The Play button moves the user to a Players interface where the users can introduce the names of the player 1 and player 2 to be displayed while playing and saved in a file. After introducing the names, the user proceeds to the Play_Frame interface where the real game is played. The users can select where do they want to put their specific symbol (X or O). After, the user has selected where to put their symbol, the game checks if their symbol completes a 3 in a row. In case it is true the game stops and displays the winner, if it is false then the turn passes to the next player and the game continues until one person completes 3 in a row or no one wins. *the ways of winning can be found in the project proposal.

Another feature is that the user can save the game they are playing in whatever moment they want, after somebody has won or in the middle of playing a game. By clicking on the Open button on the main menu, the user can choose a .txt file to open, they can find the game they saved and resume it or see who won. This is all possible to object serialization and deserialization.

Class Diagram:


tictactoe
Players
- name1: String
- name2: String
- background: javax.swing.JLabel
- dots_gif: javax.swing.JLabel
- title: javax.swing.JLabel
- subtitle1: javax.swing.JLabel
- subtitle2: javax.swing.JLabel
- tresd: javax.swing.JLabel
- menu_button: javax.swing.JButton
- playnow_button: javax.swing.JButton
- name_player1: javax.swing.JTextField
- name_player2: javax.swing.JTextField
+ Players()
- initComponents: void
+ setName1(String): void
+ setName2(String): void
+ getName1(): String
+ getName2(): String
- menu_buttonMouseClicked(java.awt.event.MouseEvent evt): void
- playnow_buttonMouseClicked(java.awt.event.MouseEvent evt): void
1
1
1
1
Menu
- background: javax.swing.JLabel
- dots_gif: javax.swing.JLabel
- title: javax.swing.JLabel
- tresd: javax.swing.JLabel
- play_button: javax.swing.JButton
- rules_button: javax.swing.JButton
- score_button: javax.swing.JButton
+ Menu()
- initComponents: void
- score_buttonMouseClicked(java.awt.event.MouseEvent evt): void
- rules_buttonMouseClicked(java.awt.event.MouseEvent evt): void
- play_buttonMouseClicked(java.awt.event.MouseEvent evt): void
+ main(args[ ]: String): void
1
1
1
Rules
- background: javax.swing.JLabel
- title1: javax.swing.JLabel
- rules_text: javax.swing.JTextArea
- jScrollPane2: javax.swing.JScrollPane
- menu: javax.swing.JButton
+ Rules()
- initComponents: void
- menuMouseClicked(java.awt.event.MouseEvent evt): void
1
Choose_File
- background: javax.swing.JLabel
- dots_gif: javax.swing.JLabel
- title: javax.swing.JLabel
- tresd: javax.swing.JLabel
- menu_button: javax.swing.JButton
- chooser: javax.swing.JFileChooser
+ Choose_File()
- initComponents: void
- playnow_buttonMouseClicked(java.awt.event.MouseEvent evt): void
- chooserActionPerformed(java.awt.event.ActionEvent evt): void
Play_Frame
- turn: int
- game: Game
- player1: Player
- player1: Player
- grid: int [ ] [ ] [ ]
- result: String
- name1: String
- name2: String
- filename: String
- game_writer: ObjectWriter
- buttons: javax.Swing.JButton [ ] [ ] [ ]
- a1: javax.Swing.JButton
- a2: javax.Swing.JButton
- a3: javax.Swing.JButton
- a4: javax.Swing.JButton
- a5: javax.Swing.JButton
- a6: javax.Swing.JButton
- a7: javax.Swing.JButton
- a8: javax.Swing.JButton
- a9: javax.Swing.JButton
- b1: javax.Swing.JButton
- b2: javax.Swing.JButton
- b3: javax.Swing.JButton
- b4: javax.Swing.JButton
- b5: javax.Swing.JButton
- b6: javax.Swing.JButton
- b7: javax.Swing.JButton
- b8: javax.Swing.JButton
- b9: javax.Swing.JButton
- c1: javax.Swing.JButton
- c2: javax.Swing.JButton
- c3: javax.Swing.JButton
- c4: javax.Swing.JButton
- c5: javax.Swing.JButton
- c6: javax.Swing.JButton
- c7: javax.Swing.JButton
- c8: javax.Swing.JButton
- c9: javax.Swing.JButton
- layerA_graphic: javax.swing.JLabel
- layerB_graphic: javax.swing.JLabel
- layerC_graphic: javax.swing.JLabel
- background: javax.swing.JLabel
- done_button: javax.swing.JButton
- dots_gif: javax.swing.JLabel
- file_label: javax.swing.JLabel
- filename_field: javax.swing.JTextField
- menu_button: javax.Swing.JButton
- results_label: javax.swing.JLabel
- save_button: javax.Swing.JButton
- save_label: javax.swing.JLabel
- title1: javax.swing.JLabel
- tresd: javax.swing.JLabel
- turn_label1: javax.swing.JLabel
- turn_label2: javax.swing.JLabel
- turn_label3: javax.swing.JLabel
- turn_label4: javax.swing.JLabel
- turn_label5: javax.swing.JLabel
- turn_label6: javax.swing.JLabel
- turn_label7: javax.swing.JLabel
+ PlayFrame(String, String)
+ PlayFrame(String)
- initComponents: void
+ initializeSavedGame(): void
+ saveButtons(): void
+ loadSavedBoard(Game): void
- a1MouseClicked(java.awt.event.MouseEvent evt): void
- a2MouseClicked(java.awt.event.MouseEvent evt): void
- a3MouseClicked(java.awt.event.MouseEvent evt): void
- a4MouseClicked(java.awt.event.MouseEvent evt): void
- a5MouseClicked(java.awt.event.MouseEvent evt): void
- a6MouseClicked(java.awt.event.MouseEvent evt): void
- a7MouseClicked(java.awt.event.MouseEvent evt): void
- a8MouseClicked(java.awt.event.MouseEvent evt): void
- a9MouseClicked(java.awt.event.MouseEvent evt): void
- b1MouseClicked(java.awt.event.MouseEvent evt): void
- b2MouseClicked(java.awt.event.MouseEvent evt): void
- b3MouseClicked(java.awt.event.MouseEvent evt): void
- b4MouseClicked(java.awt.event.MouseEvent evt): void
- b5MouseClicked(java.awt.event.MouseEvent evt): void
- b6MouseClicked(java.awt.event.MouseEvent evt): void
- b7MouseClicked(java.awt.event.MouseEvent evt): void
- b8MouseClicked(java.awt.event.MouseEvent evt): void
- b9MouseClicked(java.awt.event.MouseEvent evt): void
- c1MouseClicked(java.awt.event.MouseEvent evt): void
- c2MouseClicked(java.awt.event.MouseEvent evt): void
- c3MouseClicked(java.awt.event.MouseEvent evt): void
- c4MouseClicked(java.awt.event.MouseEvent evt): void
- c5MouseClicked(java.awt.event.MouseEvent evt): void
- c6MouseClicked(java.awt.event.MouseEvent evt): void
- c7MouseClicked(java.awt.event.MouseEvent evt): void
- c8MouseClicked(java.awt.event.MouseEvent evt): void
- c9MouseClicked(java.awt.event.MouseEvent evt): void
- menu_button1MouseClicked(java.awt.event.MouseEvent evt): void
- save_buttonMouseClicked(java.awt.event.MouseEvent evt): void
- done_buttonMouseClicked(java.awt.event.MouseEvent evt): void
logic
Game
- player1: Player
- player2: Player
- grid: int [ ] [ ] [ ]
- turn: int
+ Game(Player, Player, int [ ] [ ] [ ], int)
+ setPlayer1(Player): void
+ set Player2(Player): void
+ getPlayer1(): Player
+ getPlayer2(): Player
+ setGrid(int [ ] [ ] [ ]): void
+ getGrid(): int [ ] [ ] [ ]
+ setTurn(int): void
+ getTurn(): int
+ checkWin(): String
1..*
1
1
2
1..*
1
ObjectReader
- file_in: FileInputStream
- in: ObjectInputStream
+ read(String): Game
Player
- name: String
- player_number: int
+ Player(String, int)
+ setName(String): void
+ getName(): String
+ setPlayerNumber(int): void
+ getPlayerNumber(): int
ObjectWriter
- file: FileOutputStream
- out: ObjectOutputStream
- filename: String
+ writer(Game, String): void

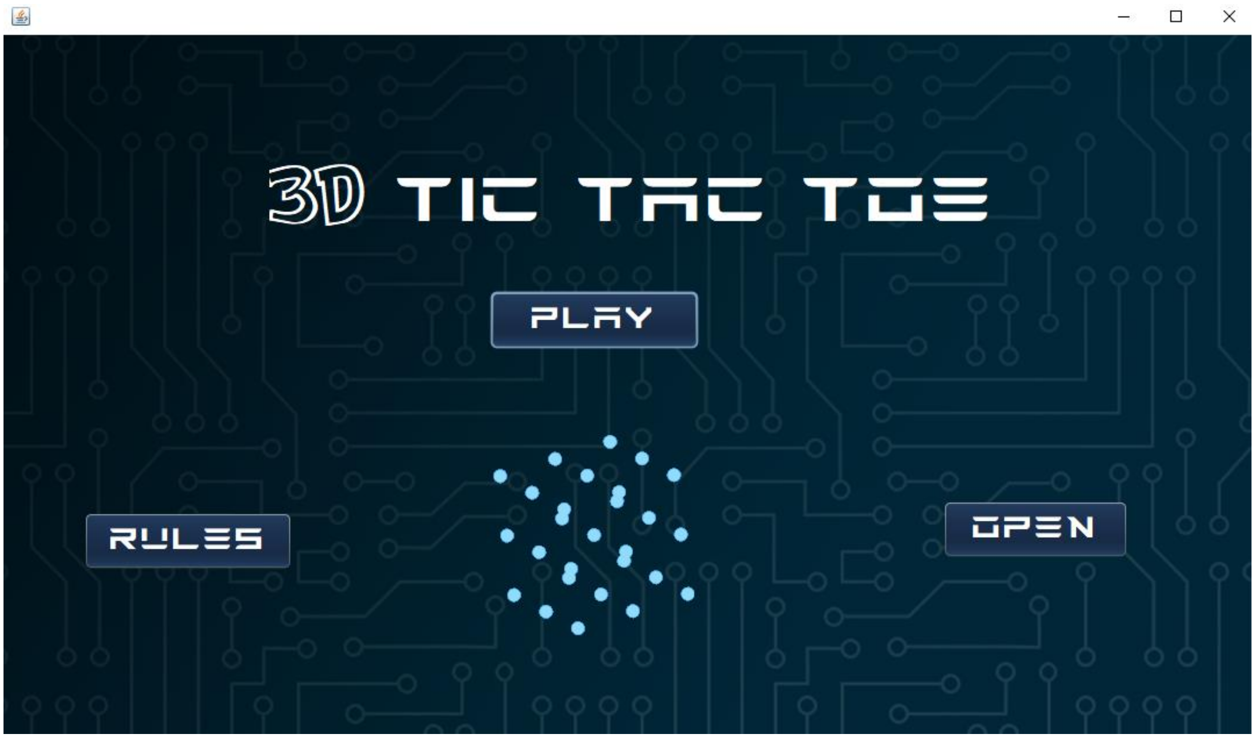
Instructions to run the program:

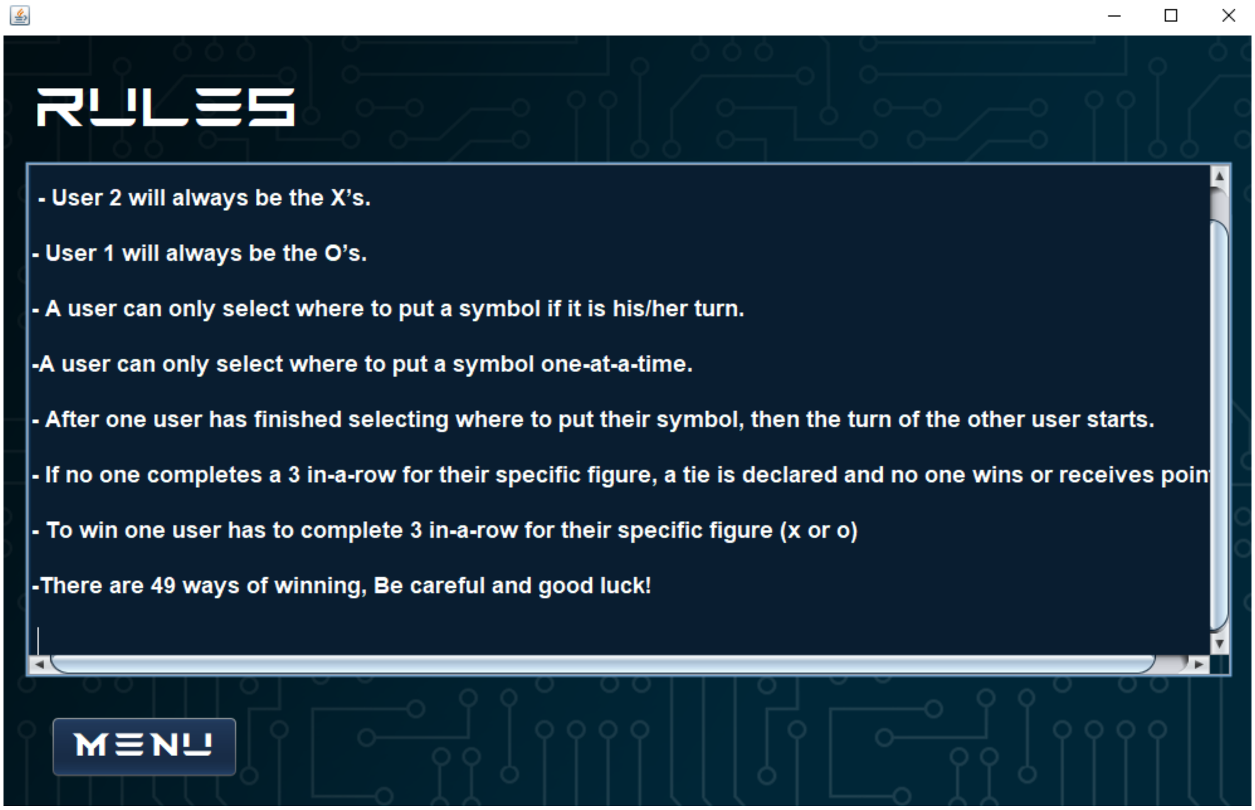
1. Open TicTacToe Project in NetBeans.
2. Run src/tictactoe/Menu

Or

1. Go to the TicTacToe folder
2. Go to dist folder
3. Open the .jar file.

Screenshots of the code executed:





- User 2 will always be the X's.

- User 1 will always be the O's.

- A user can only select where to put a symbol if it is his/her turn.

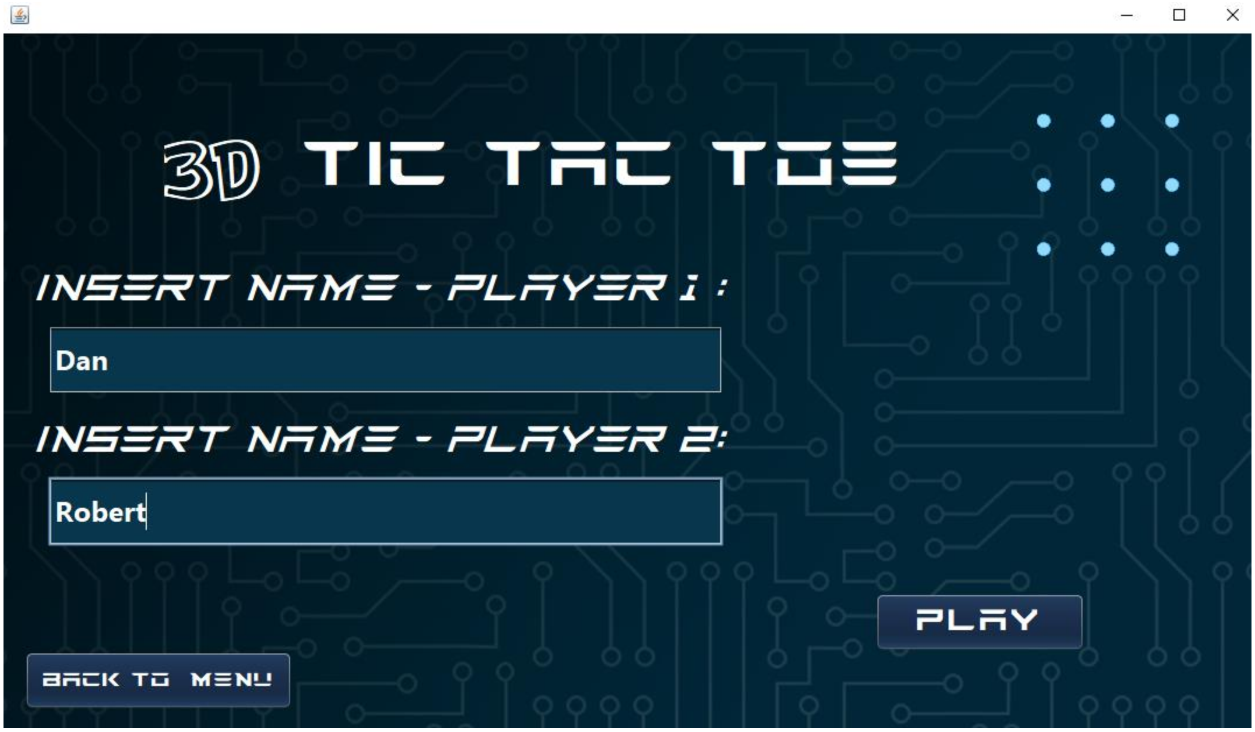
-A user can only select where to put a symbol one-at-a-time.

- After one user has finished selecting where to put their symbol, then the turn of the other user starts.

- If no one completes a 3 in-a-row for their specific figure, a tie is declared and no one wins or receives poin

- To win one user has to complete 3 in-a-row for their specific figure (x or o)

-There are 49 ways of winning, Be careful and good luck!



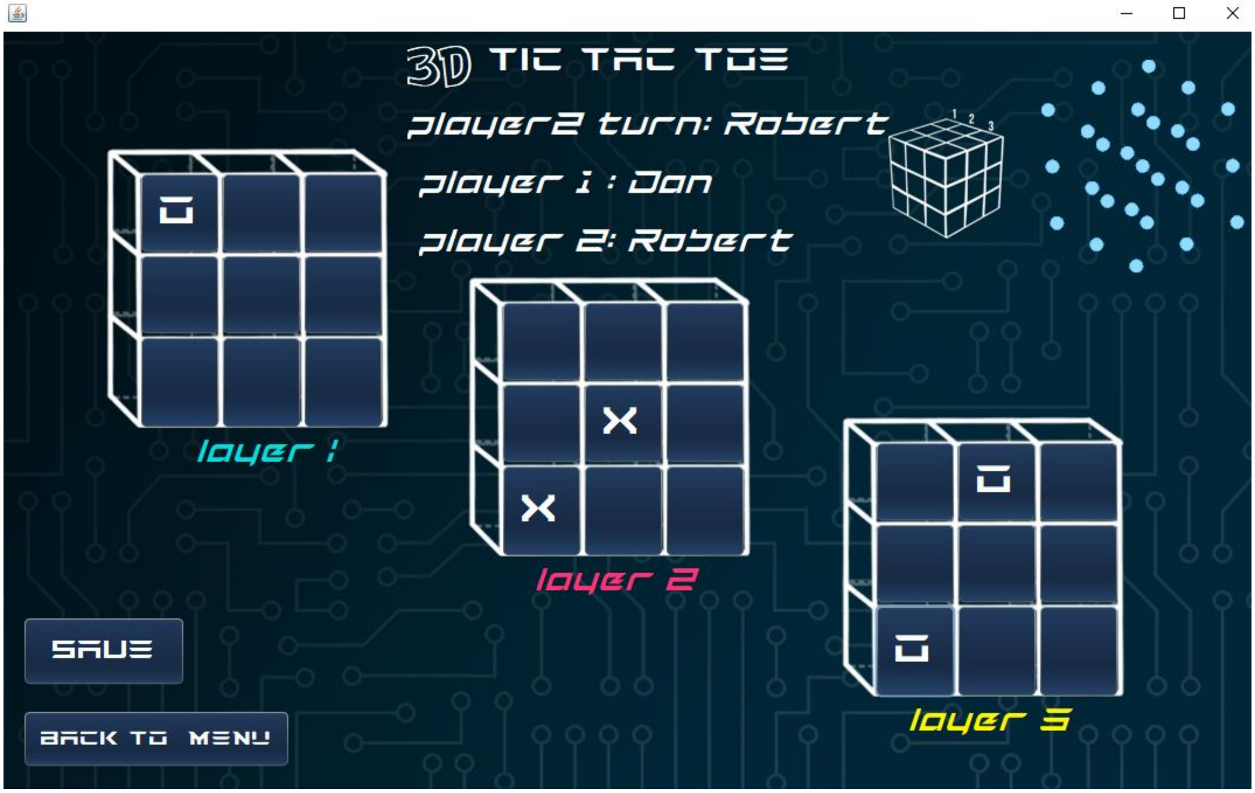
3D TIC TAC TOE

INSERT NAME - PLAYER 1 :

Dan

INSERT NAME - PLAYER 2:

Robert

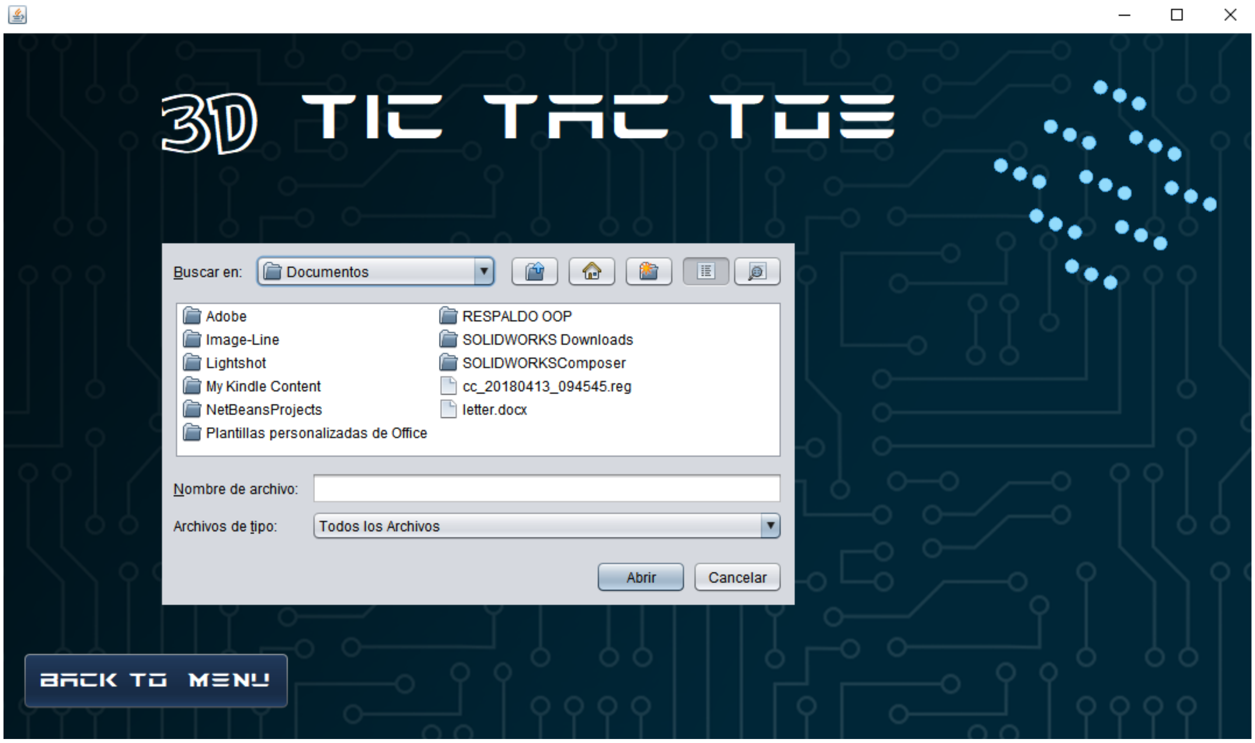


3D TIC TAC TOE

player2 turn: Robert

player 1 : Dan

player 2: Robert



Buscar en: Documentos

Nombre de archivo:

Archivos de tipo: Todos los Archivos

Individual Project Proposal:
3D Tic-Tac-Toe

Introduction:

Tic-Tac-Toe is an ancient that dates back to Ancient Egypt around 1300 B.C., Since then, this game has been played by different cultures through history. In 1952, Tic-Tac-Toe was the first computer game to be played. It was developed by AS Douglas for the ESDAC (Electronic Delay Storage Automatic Calculator), the world's first ever programmable computer. [1]

The normal way of winning a 2D Tic-Tac-Toe game is by completing a 3-in-a-row with the same symbols (X´s or O´s) in a vertical, horizontal or diagonal way.

[1] Anonymous (2018). History of Noughts and Crosses. [online] Ubergames.co.uk. Available at: http://www.ubergames.co.uk/noughts-and-crosses-history.html [Accessed 4 Mar. 2018].

Purpose:

The main purpose of a Tic-Tac-Toe game is to entertain people and let them have fun. It could also be used to develop a strategic and logical mindset that is able to figure out many possible different combinations and think about the possible movements of the opponent. In the case of a 3D-Tic-Tac-Toe it can also have the purpose of developing a Visual-spatial ability by understanding a 3D-Space and the possible movements the player and opponent can do within it.

Features:

- It will allow 2 users to play a 3D tic-tac-toe game.
- The user can introduce their name/nickname for the program to save their game results with their name.
- The players can and save/open their game results which will be stored in a file.
- It will have a graphical user interface.

  - Composed of buttons
  - Tic-Tac-Toe 2D grids that represent a layer
  - A 3D image of the whole cube for the user to look at and have a better idea of the 3d space in which the game is being played.
  - Displays the person who currently has to put their symbol. ○ Has an exit button.
  - Has a start menu with play, exit and rules button ○ Has a window that displays the rules.
  - Has a window to introduce the user's names.
  - Has a final window that displays results.
- The players will be able to select where to put their symbol by clicking on a certain space in the Tic-Tac-Toe grid.
- It will include colors to differentiate layers.
- It will tell who wins and who loses or if it was a tie.

Rules:

- The game can only function if only 2 real users are going to play it.
- User 2 will always be the X's.
- User 1 will always be the O's.
- A user can only select where to put a symbol if it is his/her turn.
- A user can only select where to put a symbol one-at-a-time.
- After one user has finished selecting where to put their symbol, then the turn of the other user starts.
- If no one completes a 3 in-a-row for their specific figure, a tie is declared and no one wins

- **To win one user has to complete 3 in-a-row for their specific figure (x or o)**

In a 3D Tic-Tac-Toe game there are 49 possible ways of winning:

There are 8 3-in-a-rows in each level. This would be the normal Tic-Tac-Toe *Note: All views are viewed from above without rotating the cube.*

X		
X		
X		

Layer 1

	X	
	X	
	X	

Layer 1

		X
		X
		X

Layer 1

X	**X**	**X**

Layer 1

X	**X**	**X**

Layer 1

X	**X**	**X**

Layer 1

		x
	x	
x		

Layer 1

x		
	x	
		x

Layer 1

Now, as there are 3 layers, the total ways of winning until now are: 3x8=24
There are also 9 vertical 3-in-a-rows that need the three layers of the cube.

1)

x		

Layer 1

x		

Layer 2

x		

Layer 3

2)

	x	

Layer 1

	x	

Layer 2

	x	

Layer 3

3)

		x

Layer 1

		x

Layer 2

		x

Layer 3

4)

x		

Layer 1

x		

Layer 2

x		

Layer 3

5)

	x	

Layer 1

	x	

Layer 2

	x	

Layer 3

6)

		x

Layer 1

		x

Layer 2

		x

Layer 3

7)

x		

Layer 1

x		

Layer 2

x		

Layer 3

8)

	x	

Layer 1

	x	

Layer 2

	x	

Layer 3

9)

		x

Layer 1

		x

Layer 2

		x

Layer 3

The total ways of winning until now are: 24+9=33

The next diagrams are the rest of possible combinations to complete 3-in-a-rows:

Diagonal 2D (Viewed from above). Times it could be done in the whole cube: 12
Examples:

x		

Layer 1

	x	

Layer 2

		x

Layer 3

		x

Layer 1

	x	

Layer 2

x		

Layer 3

x		

Layer 1

x		

Layer 2

x		

Layer 3

x		

Layer 1

x		

Layer 2

x		

Layer 3

The total ways of winning until now are: 33 + 12= 45

Diagonal 3D (Viewed from above) Times it could be done in the whole cube: 4

x		

Layer 1

	x	

Layer 2

		x

Layer 3

		x

Layer 1

	x	

Layer 2

x		

Layer 3

x		

Layer 1

	x	

Layer 2

		x

Layer 3

		x

Layer 1

	x	

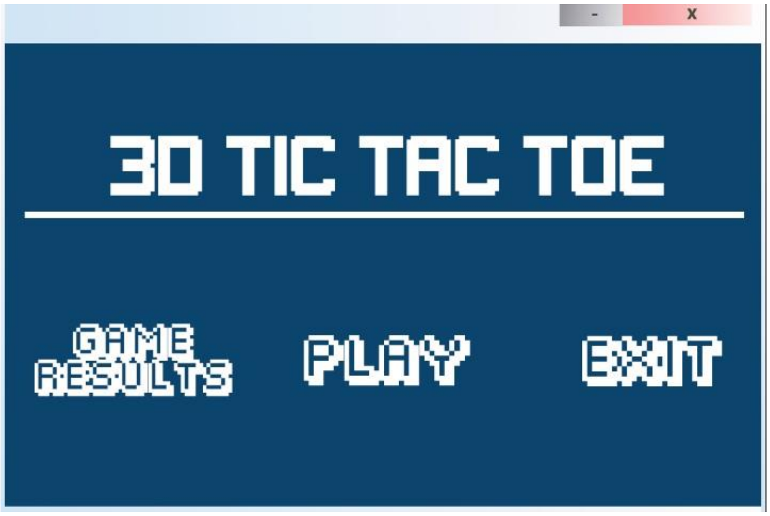
Layer 2

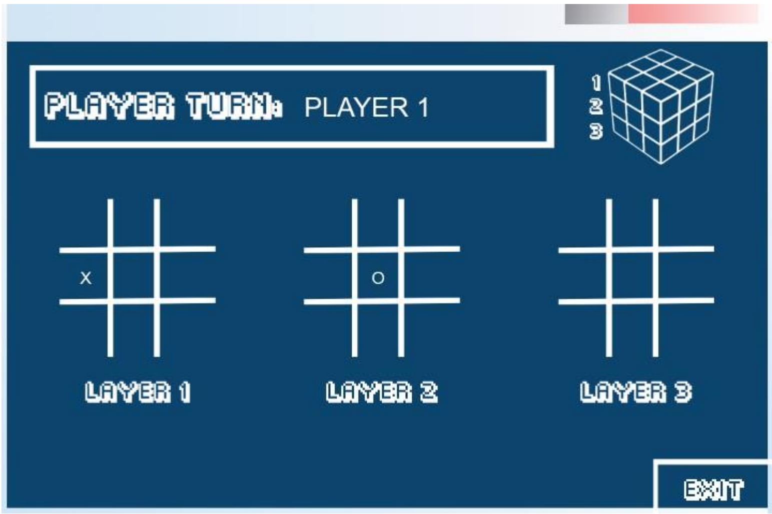
x		

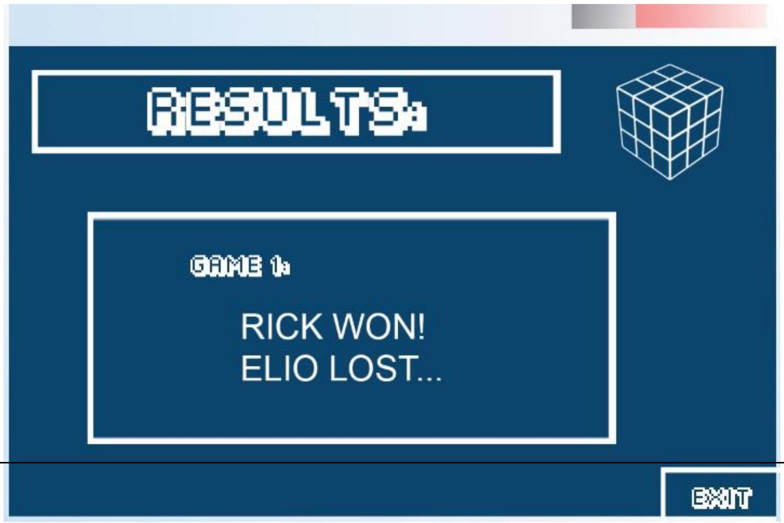
Layer 3

The final total ways of winning are: 45+4= **49**

How will it look:

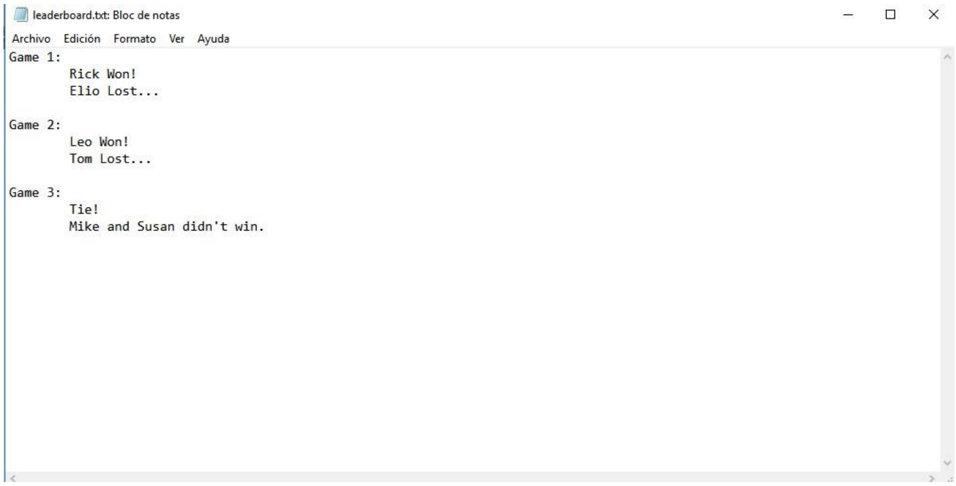



PLAYER TURN: PLAYER 1


RESULTS:

GAME 1:

RICK WON!
ELIO LOST...


leaderboard.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda

```
Game 1:
        Rick Won!
        Elio Lost...

Game 2:
        Leo Won!
        Tom Lost...

Game 3:
        Tie!
        Mike and Susan didn't win.
```

- This changed to the Open and Choose File in the final version. = Serialization and Deserialization.