Introduction to JAVA
Final Project Writeup

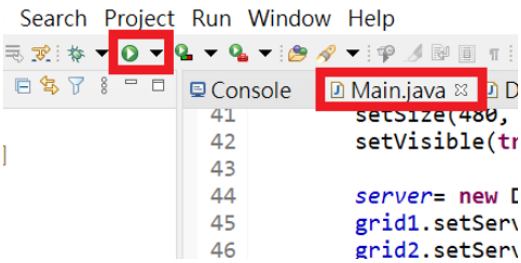
3D Tic Tac Toe

A game Implemented by JAVA

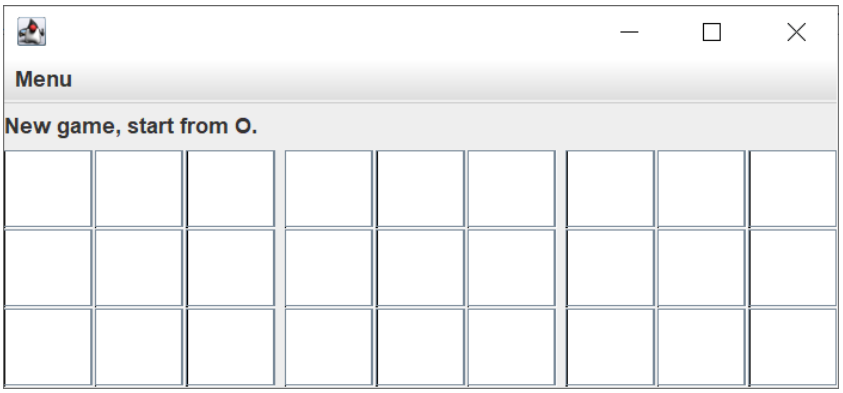
Roberto Hong Li-Yeh Yang

How to play?

1. Open Eclipse → File → Open Projects from File System…
   → Import the project "IntroJava_Final_zh2441_ly2173" from Directory
   → Go to src and open the "Main.java" → Run → Run As → 1 Java Application



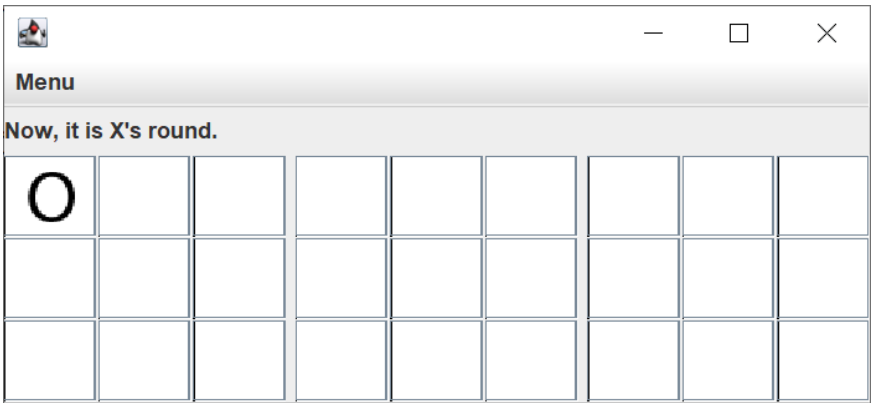
2. A GUI will pop up, then you can start playing



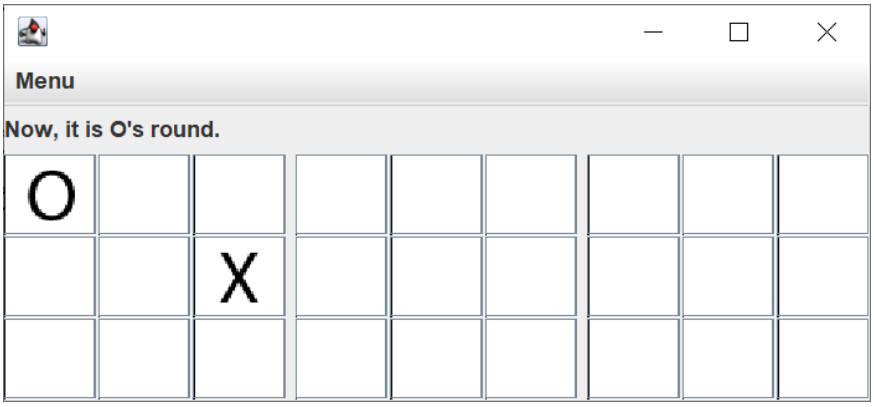
There are three 3×3 grids. Each represents a layer in a 3×3×3 cube.
- the leftmost grid represents the top layer of the cube,
- the middle grid represents the middle layer of the cube
- and finally the rightmost grid represents the bottom layer of the cube

3. Now, players can take turns placing the mark on the grids.

   Nought goes first

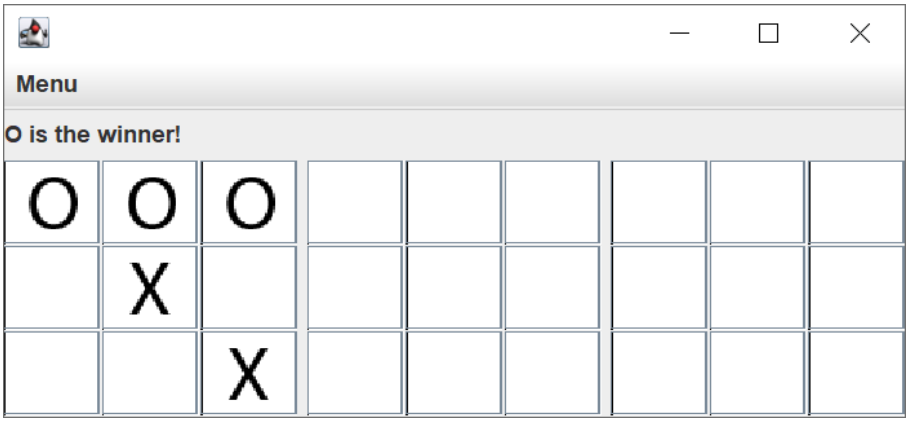


Then cross

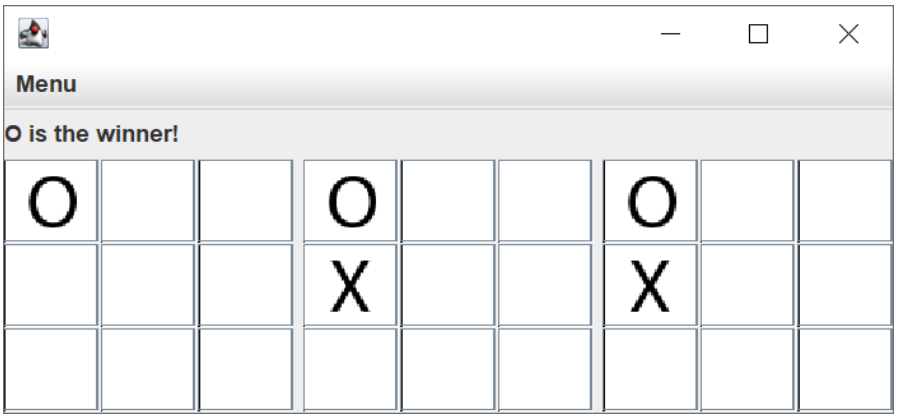


4. Winning Condition:
   The winner is the player who forms a line first.
   Remember to think 3-dimensionally, the following are some of examples:

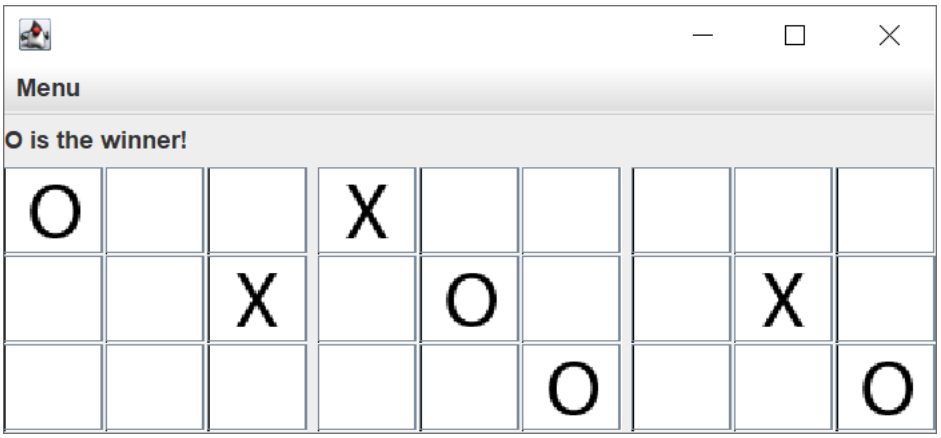
   - 3 marks form a line on a layer of the cube.



   - 3 marks through the cube at the same coordinate.

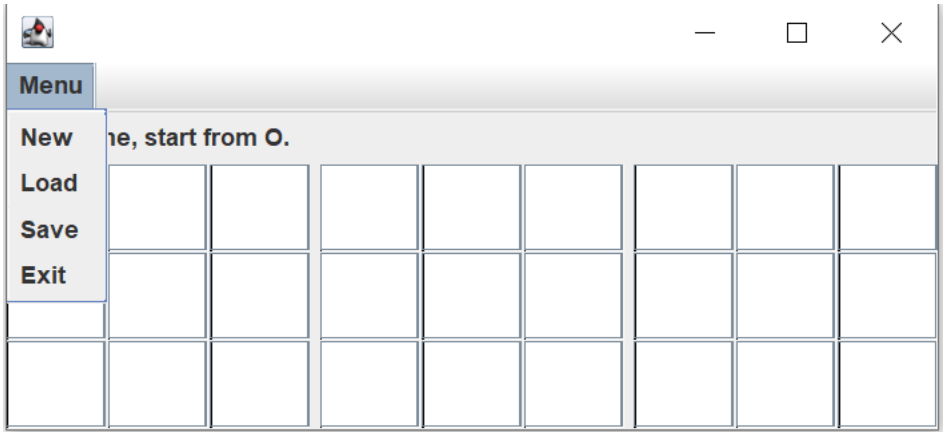


- 3 marks go diagonally through the cube.



5. Menu bar:

The pull-down menu contains functions below:
- New: start a new game
- Load: retrieve the game from the last saving point
- Save: save the current unfinished game
- Exit: exit the program



Planned to do

1. ***JAVA GUI libraries and Event Listener:***
   - Extending JFrame, JPanel, JButton libraries to create the game window, menu bar and grid
   - Adding listener to each button for updating the status (blank, O or X) of the space when there is a mouse click

2. ***Multithreading:***
   - Creating 3 threads to process each layer in the cube
   - Passing data between threads to examine the winning-condition

3. ***Server and Client:***
   - Creating a server which receives or sends game data from or to clients when save or load button is pressed

4. ***Database:***
   - Setting up a database to store the serialized game data

Actually did

We have three .java files in this project: Main, DataServer ,and GridPanel.

1. ***Main:***
   - Implements GUI, Event Listener, Client
   - It sets up frame, label, panels, and menu.
   - Each button in menu is added an event listener which calls corresponding methods to perform New, Load, Save, and Exit functions.

2. ***GridPanel:***
   - Implements GUI, Event Listener, and Multithreading
   - It sets up a 3×3 grids consisting of 9 buttons.
   - Each button can listen the mouse click and mark the place.
   - It also runs as a thread and works independently. When checking winning-condition, they access data of other layers through DataServer thread.

3. ***DataServer:***
   - Implements Server, Multithreading and Database
   - It runs as a thread keeping track of latest data of 3 GridPanel threads.
   - It also serves to connect JDBC for storing and retrieving game data.
   - It uses SQL statements to insert, update or select data in database.

In summary, the implementation of this game includes advanced topics such as GUI, Multithreading, Database, etc. We completed this project according to our proposal.