ADVANCED PROGRAMMING

ACADEMIC YEAR 2021-2022

REPORT TIC TAC TOE

DOMENICO ERRIQUEZ

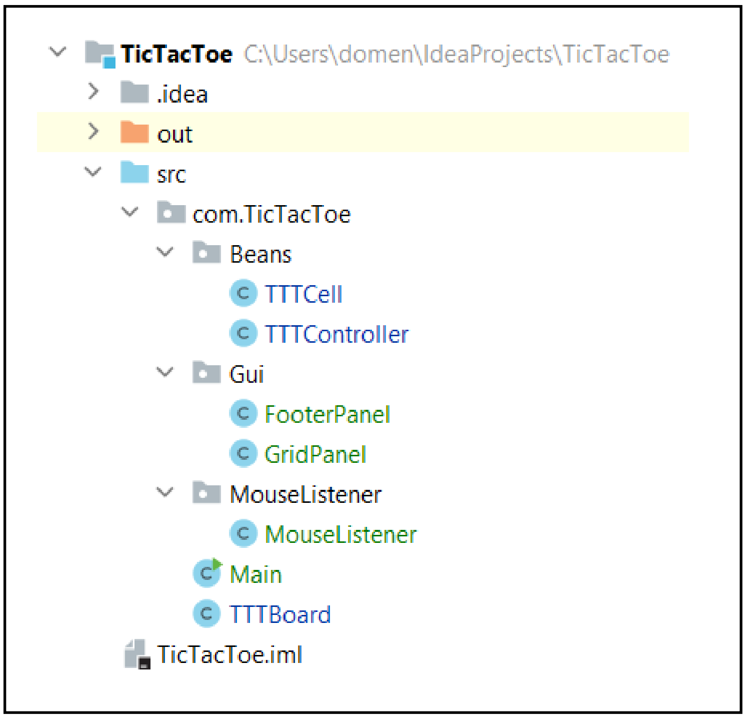
University of Pisa

Department of Computer Science

1. Introduction

This report describes the main design decision about the implementation with Java of the Tic Tac Toe game.

2. Structure of the program



```
TicTacToe C:\Users\domen\IdeaProjects\TicTacToe
  .idea
  out
  src
    com.TicTacToe
      Beans
        TTTCell
        TTTController
      Gui
        FooterPanel
        GridPanel
      MouseListener
        MouseListener
      Main
      TTTBoard
  TicTacToe.iml
```

Figure 1: Structure of the program

The program is structured by 4 main packages:

- Beans package: it contains the beans used in this project, **TTTCell** and **TTTController**
- Gui package: it contains the Jpanel used for the graphic user interface like **GridPanel** and **FooterPanel**
- MouseListener package: it contains the **MouseListener**
- TicTacToe package: it contains the Main class and the **TTTBoard** class

3. Beans package

3.1 TTTCell

The TTTCell class represents a cell of the board. It extends a JPanel and it’s made by two JButton: buttonX, button0.
Each cell has a state that can be “Initial”, “X” or “0”.

- **Initial** it’s the default state, it means that cell has not yet been clicked by the Player X or Player 0.
- **X** it means that the player clicked on the buttonX so the cell’s state is set to X.
- **it** means that the player clicked on the buttonO so the cell’s state is set to O

The layout of the cell is a GridbagLayout where the buttonX is placed in row 0 and column 0 and the button0 in row 1 and column 0
To change the value of the state it has been implemented a function that changes the state of the function only if the TTTController allows this change. Once the change is allowed the listeners get notified.

3.2 TTTController

The TTTController has the job to check if all the moves made by the players are correct. The TTTController extends JLabel and is represented by a label that indicates the status of the game.
It is used a Boolean variable “InitialGame” to check if the game just started and this means that both players didn’t make a play, so both of them are allowed to make the next move.
It is used a Boolean variable “ActivePlayer0” to know who player’s turn is, if this variable is true then the Player0 is allowed to make a move, if it is false then the PlayerX is allowed to make a move.
TTTController implements two listener interfaces:

- VetoableChangeListener: It’s triggered whenever a player tries to change the state of a TTTCell, so whenever they click a JButton. It checks if the move is allowed, so in case the next turn is of the PlayerX and the Jbutton clicked is the Button0(dually for PlayerO) then this move will be forbidden, and it will through an exception saying, “It’s not your turn”.

- PropertyChangeListener: It’s triggered whenever a TTTCell’s state is changed. Bases on who change the TTTCell’s state it sets the variable ActivePlayer0 to true or false, it changes the text of the label showing whose turn is.

4. Gui package

4.1 GridPanel

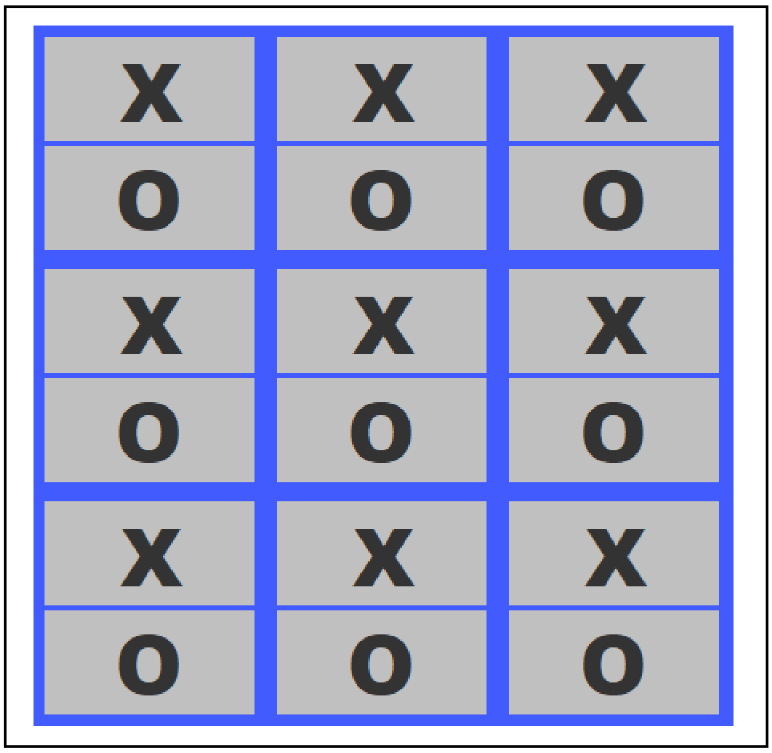


Figure 2: GridPanel

The gridpanel class extends a Jpanel and it represent the grid made by nine TTTCell. The layout used is GridLayout with 3 rows and 3 columns. To each TTTCell it adds the listeners and to each button of each cell it adds the

MouseListener.

4.2 FooterPanel

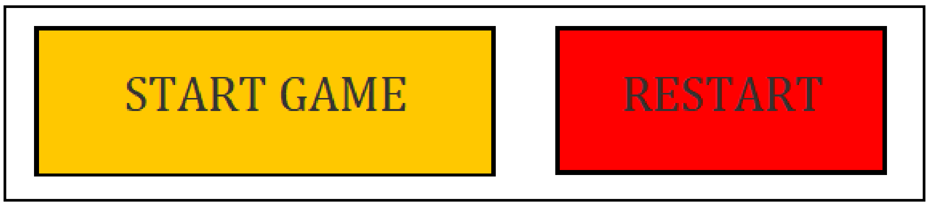


Figure 3: FooterPanel

The FooterPanel class extends a JPanel, it contains the TTTController bean and a Jbutton “Restart” used to reset all the values in order to start a new match. Its layout is a BoxLayout where the objects are positioned on the x axis.

5. MouseListener

The mouselistener class’s goal is to track if the players click on the TTTCell's buttons. In case there is a click on one of the buttons and the button is enabled it will try to change the state of the cell based on which button has been clicked. After changed the state in case there is not a winner it will change the button’s color, disable it and set the text of the button to an empty string.

6. TicTacToe package

TTTBoard class

The TTTBoard class extends a Jframe and it contains all the other components said in the previous chapters. Its layout is a BoxLayout where GridPanel and the FooterPanel are positioned on the Y axis.
It implements a propertyChangeListener that it’s triggered when the state

of a cell change. It checks if there is a tris or if the grid got full without a winner. In case there is a winner, it disables all the buttons of the cells, it changes the label of the TTTController showing who is the winner and it will color of green the buttons that led to the tris. In case the grid got full without winner it will disable all the buttons of the cells and it will set the label of the TTTController to “It’s a tie”.