CO 225: Software Construction

Lab 06 : Event Handling and MVC Approach

Ranasinghe S.M.T.S.C.

E/18/285

MVC Approach

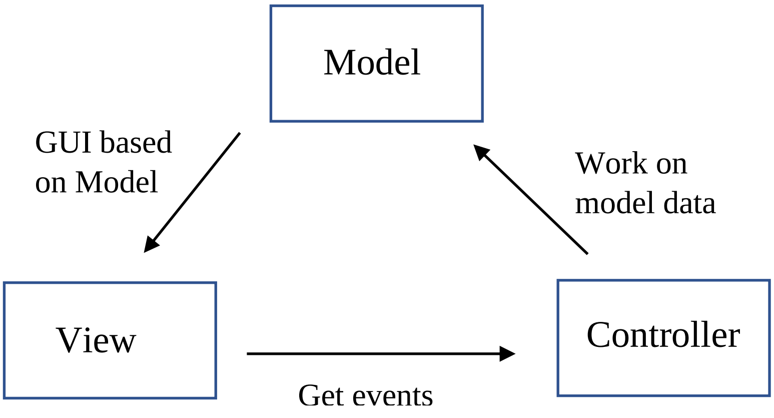
In lab 06 we build Tic Tac Toe game using java MVC architecture.

In this application,

Model – Store data about variables, get signals from controller and check for winner

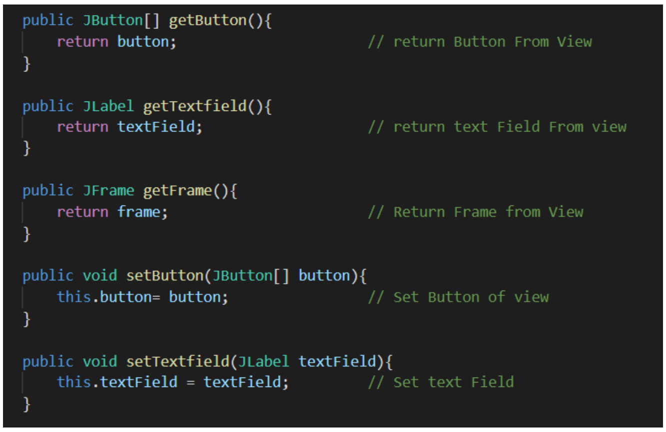
View – View contains all GUIs of game that show to user. It is showing details based on model and changes that happens in controller

Controller – Here controller is doing event handling of model and build interface between model and view



VIEW

View has GUI of the program. So, also it has its methods to change the view base on model data. Instance of view is created in Main method for execution. Instance of Model is in View.

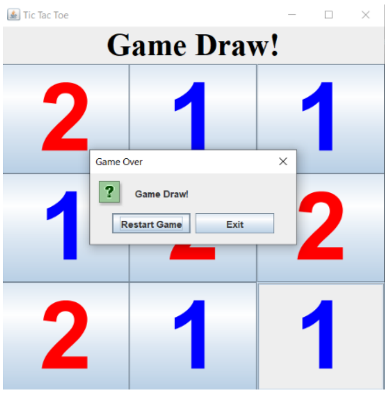


```
public JButton[] getButton(){
    return button;                    // return Button From View
}

public JLabel getTextfield(){
    return textField;                 // return text Field From view
}

public JFrame getFrame(){
    return frame;                     // Return Frame from View
}

public void setButton(JButton[] button){
    this.button= button;              // Set Button of view
}

public void setTextfield(JLabel textField){
    this.textField = textField;       // Set text Field
}
```

These methods used to get and set view based on Model’s data.

GUI : -

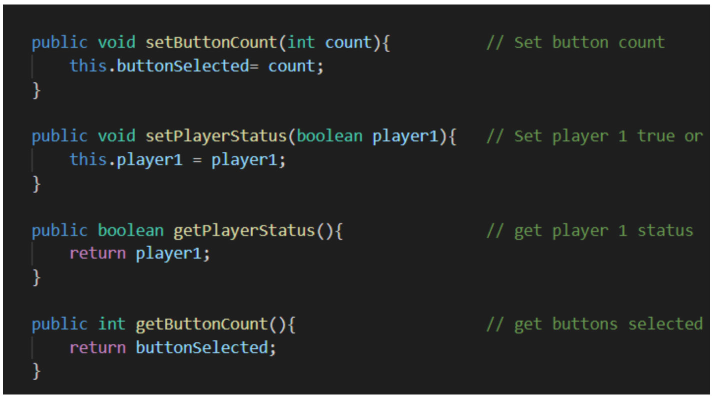
To get view, Main Class should be executed.



MODEL

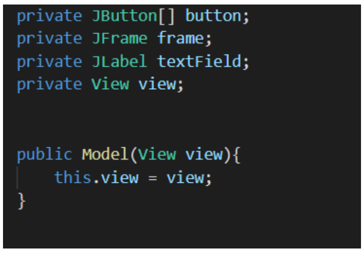
Model Contains variables that view is depended. It has its own getter and setter methods to set data on view and get data changes from controller. Rather than that it checks the winner of game by method.

It has instance of View (Create using Model’s Constructor).



```
public void setButtonCount(int count){          // Set button count
    this.buttonSelected= count;
}

public void setPlayerStatus(boolean player1){   // Set player 1 true or
    this.player1 = player1;
}

public boolean getPlayerStatus(){               // get player 1 status
    return player1;
}

public int getButtonCount(){                    // get buttons selected
    return buttonSelected;
}
```

Variables Used for View instance and Constructor

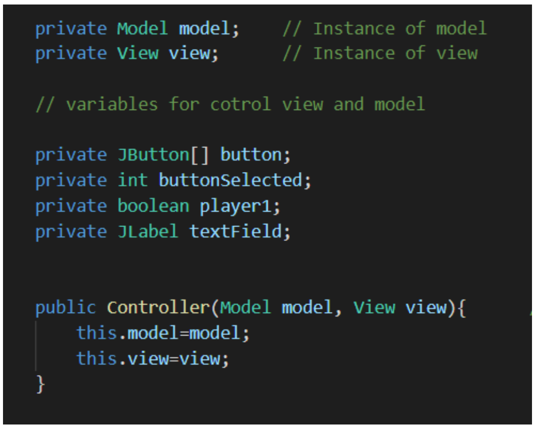


```
private JButton[] button;
private JFrame frame;
private JLabel textField;
private View view;


public Model(View view){
    this.view = view;
}
```

CONTROLLER

Controller do the main job for program. It checks changes on view that were done by users and send those changes to Model. For this program controller doing event handling of application.

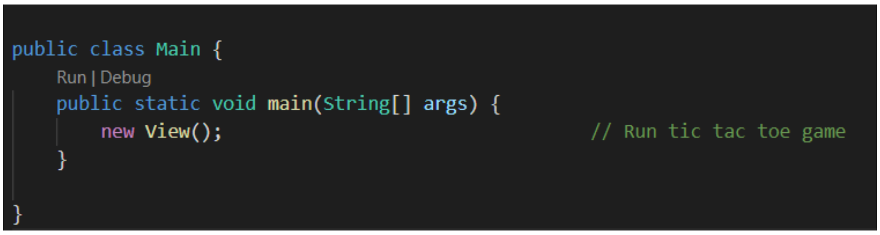
It has both View and Model instances in it (Created Through Constructor). And variables used in Model and View.



```
private Model model;    // Instance of model
private View view;      // Instance of view

// variables for cotrol view and model

private JButton[] button;
private int buttonSelected;
private boolean player1;
private JLabel textField;


public Controller(Model model, View view){
    this.model=model;
    this.view=view;
}
```

MAIN

Main Class contains instance of view. It executes program and do the job with support of View, Model and Controller.



```
public class Main {
    Run | Debug
    public static void main(String[] args) {
        new View();                                  // Run tic tac toe game
    }

}
```