

COM244: OBJECT ORIENTED PROGRAMMING

TIC TAC TOE PROJECT REPORT



PREPARED BY

AYÇA NUR VANLI

1790128

Syllabus

Project Milestones

1. View Step
   1. Creating JFrame
   2. Adding Buttons
      1. Exit Button
      2. X/O buttons
      3. Try Again Button
      4. Passive-Active Buttons
   3. Customizing Frame
2. Controller Step
   1. Action Handling
      1. X/O Action
      2. Try Again Action
      3. Exit Action
   2. Win Action
   3. Changing Players
3. Model Step
   1. Initializing Game Necessities
4. Creating MVC Architecture
   1. Creating Main Class
      1. Creating Model Object
      2. Creating View Object
      3. Creating Controller Object
5. Making Relations
   1. Between View – Controller
   2. Between Controller-Model
   3. Between Model-View
6. Winning Game
   1. Reqiurements
   2. Actions That Will Be Taken
7. UML Diagrams
   1. Class Diagram
   2. Sequence Diagram
   3. Use-Case Diagram
   4. Activity Diagram

1. View Step

I created a package form named View for the View part of the Model-View-Controller architecture.

1.A. Creating Frame

Firstly , I decided that creatinf the user interface of hte program would be more fun to do soi did the GUI part of the program first. I created a JFrame named View.

1.B. Adding Buttons

I started to add buttons:

* 9 button for 3x3 matrix to X/O placements
* 1 button for exit
* 1 button for try again
* 2 button for Player X’s and Player O’s score
* 2 button for players passive/ active status

1.C. Customizing JFrame

I add JLabel for the headline of the game. I named it as “Tic Tac Toe Game”. Then i customized the colors of buttons, color of texts, font and size of the texts.

* Color (mostly): Light Pink
* Font: Verdana
* Size (mostly): 60

2. Controller Step

I created a package form named Controller for the Controller part of MCV.

2.A. Action Handling

With the help of listeners, in controller part i took actions acordingly to the specific event.Therefore, i listen the events and assained an event for it.

2.A.i. X/O Action

If a player click on the 3x3 matrix buttons, then according to the players state (passive/active) there will ve placed an X or O sign in that buttons.

2.A.ii. Try Again Action

If a player decides to reset the game and start all over again then only thing that he must do is clilcking “Try Again” button. Therefore, the scores of the players and also the 3x3 will be cleaned and ready to play again.

2.A.iii. Exit Action

If a player decides to exit the game then, there will be a JPanel message dialog popped up and ask for confirmation to exit as “Do you want to exit?”. If the answer is “Yes” then game will be shut down. Otherwise game will continute to where it was left.

2.B. Win Action

If any of the player complete its sign (X/O) consecutively for 3 button, that means that player is win and their score will be incremented 1. Also there will be a JPanel message diaolog that says “Player X/O win!” popped up.

2.C. Changing Players

When a active status player is done with its action, then the sign will be change to other player and also the player’s status will be passive.

3. Model Step

I created a package form named “Model” fort he Model part of the MVC.

3.A. Initiliazing Game Necessities

I initilized a varibles called

* oCount
  + Usage: keep count of the Player O’s total score
* XCount
  + Usage: keep count od the Player X’s total score
* oString
  + Usage: keep tracking the Player O’s game status
* xString
  + Usage: keep tracking og Player X’s game status
* startGame
  + Usage: keeping the who is in the active status

And also getters&setters for them.

4. Creating MVC Architecutre

The major part of this Project is making a collected system that used 3 seperate section called Model, View, Controller.

4.A. Creating Main Class

In a seperate package and class called “Main”, i linked three different componenets of the MCV architecture.

4.A.i. Creating Model Object

In Main class i create model object as:

Game game= **new** Game();

4.A.i. Creating View Object

In Main class i create view object as:

View view = **new** View();

4.A.i. Creating Model Object

In Main class i create controller object with the parameters model and vieew as:

Controller controller = **new** Controller(game,view);

5.Making Relations

Linking with the functions and allowing the updates.

5.A. Between View and Controller

Adding controller components and has-a relationship is a must for controller to take actions so i made a function called registerController(Controller controller) so that view has a controller.

5.B Between Controller and Model

Since Controller’s Constructure has a Model ( Game) and View paramters there is an already a link that combine them so there wasnt any necessery axtions to take.

5.B. Between Model and View

When data in Model part is change the View must also be uptaded so there must be a link for it. Because of that reason i decline a registerGame(Game game) so that View has a updated Model each time.

6. Winning Game

This part is explaining the meaning of the what win means on this project

6.A. Requirements

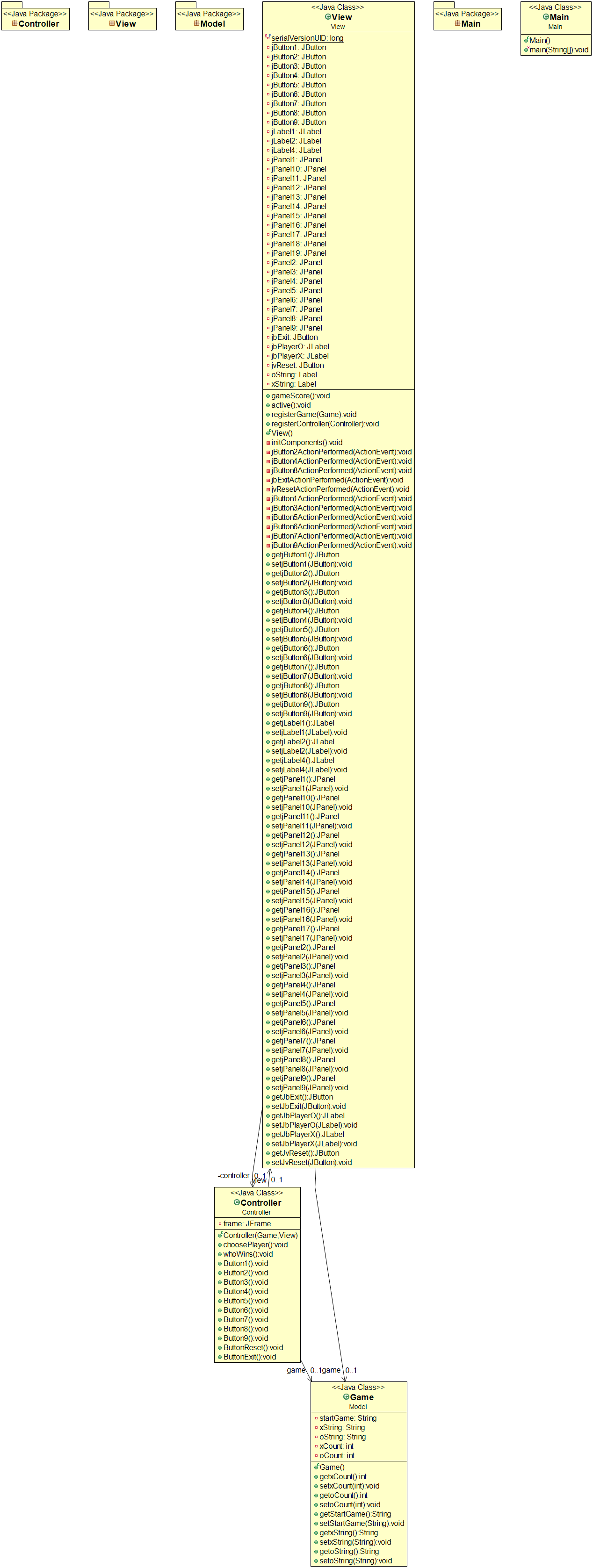
A consecutive 3 same sign button is considered as a success. It could be

* Horizontly
* Verticaly
* Across

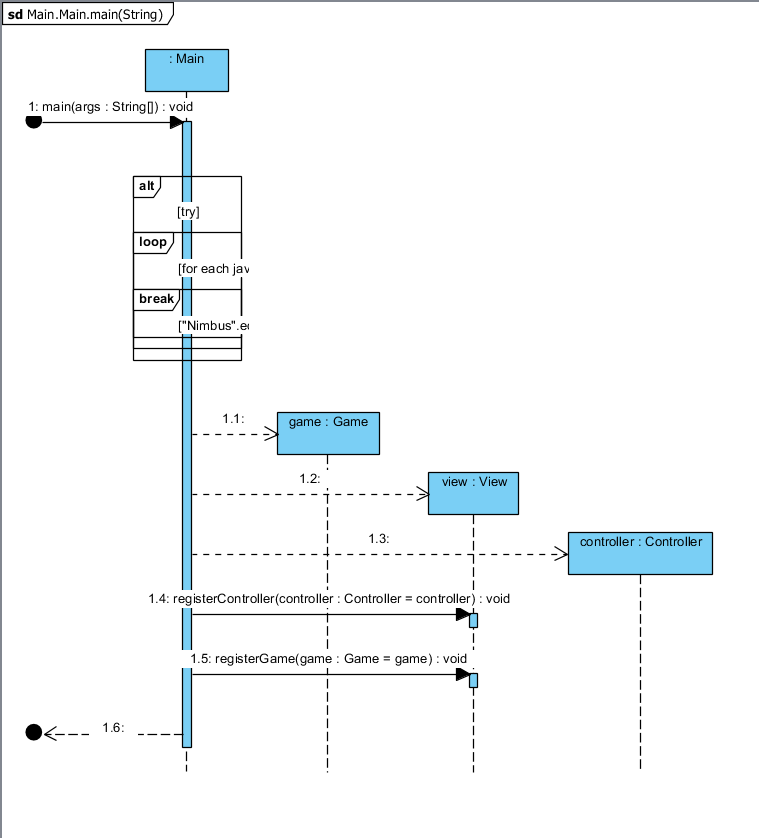
6.B. Actions That Will Be Taken

When a player is win, the score of that player will be incremented by 1 and also there will be popped up. That players succes will be highlighted with yellow color to give a brief explanation about the win.

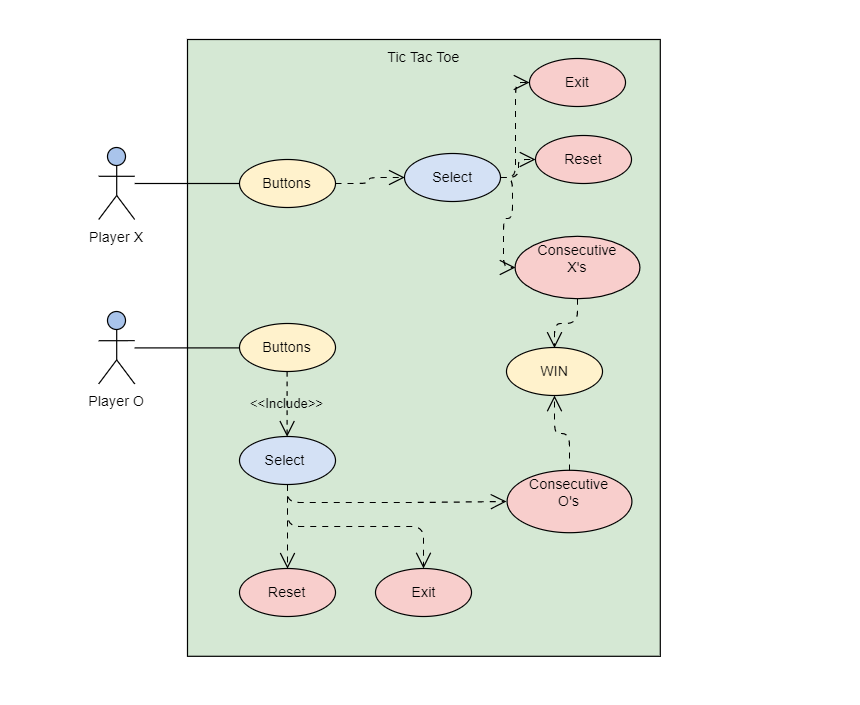
7.A. Class Diagram



7.B. Sequence Diagram



7.C. Use Case Diagram



7.D. Activity Diagram

