PROJECT REPORT

NAME : MUHAMAD FARHAN RAIS BIN NOOR MOHAMED

ID : 20001518

LECTURER : DR. NORDIN BIN ZAKARIA

Concrete Classes :

1) SystemManager
2) Player
3) StartPage
4) GameBoard
5) EndPage
6) ControllerStartPage
7) ControllerGameBoard
8) ControllerEndPage
9) BoxList
10) Box
11) OShape
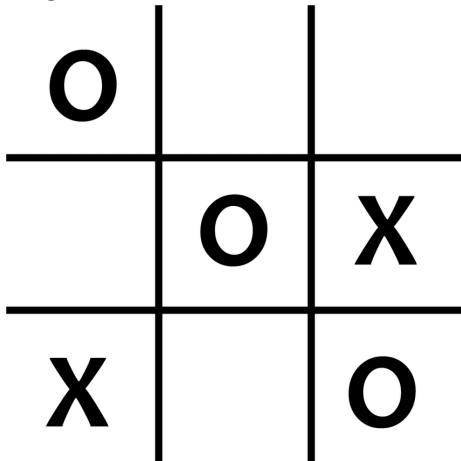12) Xshape

Abstract Classes :

1) Pages
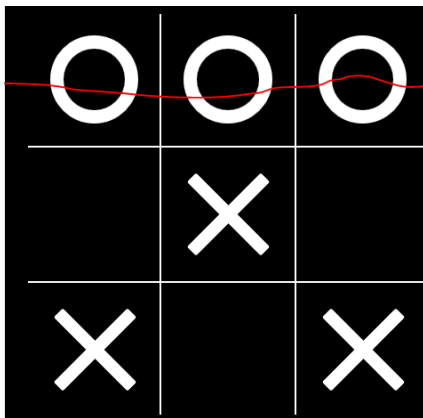2) Shape

Interface :

1) Controller

Game Overview

This is a game of TicTacToe that is played by two players. The players will have to key in their details on the start page. The start page will have text fields for the names of player 1 and player 2. Then, there will be a toggle button for shapes that each player have to choose. Both player must choose different shape or otherwise there will exception called. Then, the player can click start button to start the game. A gameboard page will be shown. This page will have 9 boxes. Player 1 will start first as indicated by a turn label. Player 1 can click on any box they want and the box will be highlighted. The highlighted box will have red border colour. Once the box is highlighted, player can press any button on the keyboard to put their chosen shape on the box, as long as there is no shape yet on the highlighted box. Then, the other player can do the same to put his shape on whichever box he prefers. Once there is a winner, the game will move to next page or end page. This page will have result label that will indicate the winner of previous match. Then, players can decide to end the game, by pressing the exit button, or replay again by clicking on the retry button. They can play as much as they want, and if they decide to stop the game, they can click exit. In the middle of the match, player can exit the game by clicking the close button on the top right corner. This will terminate the program.

Rules of Winning

1) Diagonal Win

2) Horizontal Win

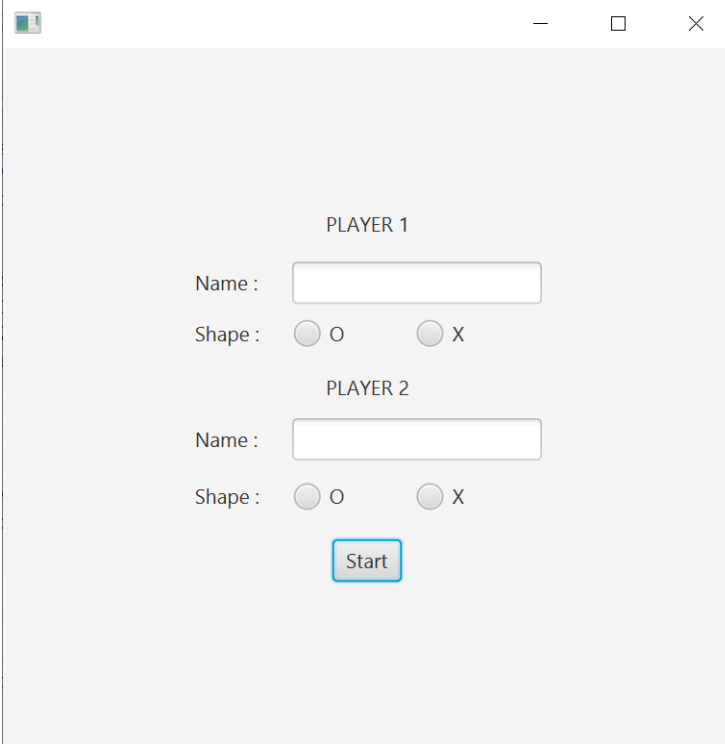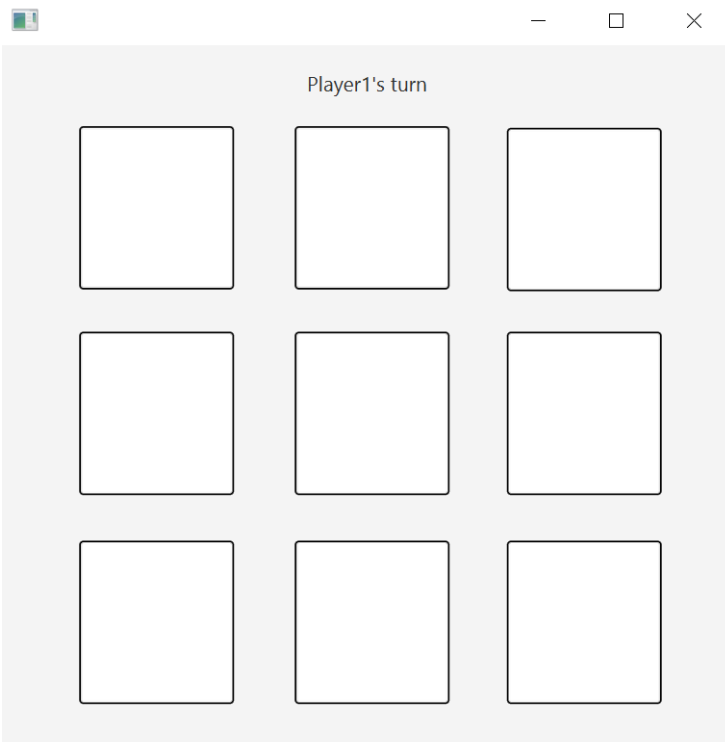
3) Vertical Win
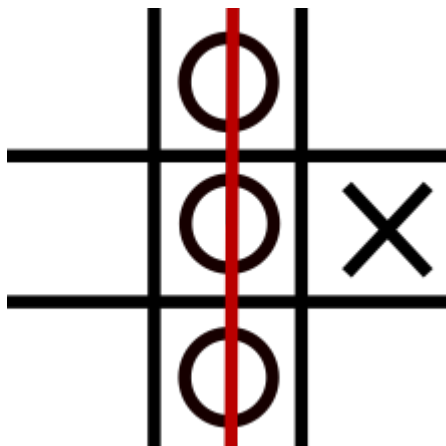
Flow of the Game

1) Start Page

PLAYER 1

Name :

Shape :   ◯ O   ◯ X

PLAYER 2

Name :

Shape :   ◯ O   ◯ X

Start

2) Gameboard Page

Player1's turn

3) End Page

RESULT

Player1 has WON!!!

Play Again

Exit

UML Diagram

TIC TAC TOE

SystemManager

+ player1 : Player
+ player2 : Player
+ stage : Stage
+ sp : StartPage
+ gb : GamePage
+ ep : EndPage
+ retry : boolean

+ start() : void
+ retry() : void
+ stop() : void

<<Abstract>>
Pages

+ player1 : Player
+ player2 : Player

+ <> run() : Scene

StartPage.fxml

GameBoard.fxml

EndPage.fxml

StartPage

+ ctrlStartPage : ControllerStartPage

+ run() : Scene

GameBoard

+ ctrlGameBoard : ControllerGameBoard
+ boxList : BoxList

+ run() : Scene
+ checkForWin(Player) : boolean
+ isComplete(ArrayList<dounle[]>) : boolean
+ isDiagonal(ArrayList<double[]>) : boolean

EndPage

+ ctrlEndPage : ControllerEndPage

+ run() : Scene

Player

+ name : String
+ shapeCode : String
+ shapePos : ArrayList<double>
+ chosenBox : Box
+ hasWon : boolean

+ reset() : void
+ clickBox() : void
+ putShape(Box) : void
+ printShapePos() : void

ControllerStartPage

+ player1 : Player
+ player2 : Player
+ p1NameField : TextField
+ p2NameField : TextField
+ p1ShapeCode : ToggleGroup
+ p2ShapeCode : ToggleGroup

+ start() : void
+ setPlayers() : void

<<Interface>>
Controller

+ setPlayers;

ControllerGameBoard

+ p1 : Player
+ p2 : Player
+ currPlayer : Player
+ boxList : BoxList
+ tempR : Rectangle
+ turn : String
+ turnCount : int

+ setTurnLabel(Label) : void
+ setBoxList(BoxList) : void
+ drawShapes(KeyEvent) : void
+ highlightBox(MouseEvent) : void
+ nextPlayer() : void
+ setPlayers() : void

ControllerEndPage

+ player1 : Player
+ player2 : Player
+ exitButton : Button
+ retryButton : Button
+ resultLabel : Label

+ setResultLabel(Scene) : void
+ retry() : void
+ exit() : void
+ setPlayers() : void

BoxList

+ numberOfBoxes : int
+ list : ArrayList<Box>

+ clearAll() : void
+ getById(String) : Box
+ unhighlightAll() : void

Box

+ rect : Rectangle
+ xshape : XShape
+ oshape ; OShape
+ hasShape : boolean
+ highlighted : boolean

+ highlight() : void
+ clear() : void
+ setShape(String) : void

<<Abstract>>
Shape

+ visibility : boolean

+ <>
setVisibility(boolean) : void

OShape

+ circle : Circle
+ shapeCode : String

+ setVisibility(boolean) : void

XShape

+ line1 : Line
+ line2 : Line
+ shapeCode : String

+ setVisibility(boolean) : void