**MAIN JAVA**

package com.company;

import javafx.application.Application;

import javafx.application.Platform;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.Pane;

import javafx.stage.Stage;

public class Main extends Application {

private Controller controller;

@Override

public void start(Stage primaryStage) throws Exception{

FXMLLoader loader=new FXMLLoader(getClass().getResource("game.fxml"));

GridPane rootGridPane =loader.load();

controller = loader.getController();

controller.createPlayground();

MenuBar menuBar = createMenu();

menuBar.prefWidthProperty().bind(primaryStage.widthProperty());

Pane menuPane = (Pane) rootGridPane.getChildren().get(0);

menuPane.getChildren().add(menuBar);

Scene scene=new Scene(rootGridPane);

primaryStage.setScene(scene);

primaryStage.setTitle("Connect Four");

primaryStage.setResizable(false);

primaryStage.show();

}

private MenuBar createMenu(){

//File Menu

Menu fileMenu = new Menu("File");

MenuItem newGame = new MenuItem("New game");

newGame.setOnAction(event -> controller.resetGame());

MenuItem resetGame = new MenuItem("Reset game");

resetGame.setOnAction(event -> controller.resetGame());

SeparatorMenuItem separatorMenuItem=new SeparatorMenuItem();

MenuItem exitGame = new MenuItem("Exit game");

exitGame.setOnAction(event -> exitGame());

fileMenu.getItems().addAll(newGame, resetGame, separatorMenuItem, exitGame);

//Help menu

Menu helpMenu = new Menu("Help");

MenuItem aboutGame = new MenuItem("About Connect4");

aboutGame.setOnAction(event -> aboutConnet4());

SeparatorMenuItem separator=new SeparatorMenuItem();

MenuItem aboutMe = new MenuItem("About Me");

aboutMe.setOnAction(event -> aboutMe());

helpMenu.getItems().addAll(aboutGame, separator, aboutMe);

MenuBar menuBar = new MenuBar();

menuBar.getMenus().addAll(fileMenu, helpMenu);

return menuBar;

}

private void aboutMe() {

Alert alert =new Alert(Alert.AlertType.INFORMATION);

alert.setTitle("About the Developer");

alert.setHeaderText("Jinkani Avinash");

alert.setContentText("I love to play around with code and create games. " +

"Connect 4 is one of them. In free time I like to spend time with nears and dears." +

"It is awesome.");

alert.show();

}

private void aboutConnet4() {

Alert alert =new Alert(Alert.AlertType.INFORMATION);

alert.setTitle("About Connect Four");

alert.setHeaderText("How To Play");

alert.setContentText("Connect Four is a two-player connection game in which the players" +

" first choose a color and then take turns dropping colored discs from the top into a sevencolumn, " +

"six-row vertically suspended grid. The pieces fall straight down, occupying the next +"+"available space within " + "the column. The objective of the game is to be the first to form a horizontal, vertical, or"+

"diagonal line of four of one's own discs. " +

"Connect Four is a solved game. The first player can always win by playing the right moves.");

alert.show();

}

private void exitGame() {

Platform.exit();

System.exit(0);

}

private void resetGame() {

//TODO

}

public static void main(String[] args) {

launch(args);

}

}

CONTROLLER **FILE**

**package com.company;**

**public class contoller {**

**import javafx.animation.TranslateTransition;**

**import javafx.application.Platform;**

**import javafx.fxml.FXML;**

**import javafx.fxml.Initializable;**

**93**

**import javafx.geometry.Point2D;**

**import javafx.scene.control.\*;**

**import javafx.scene.layout.GridPane;**

**import javafx.scene.layout.Pane;**

**import javafx.scene.paint.Color;**

**import javafx.scene.shape.Circle;**

**import javafx.scene.shape.Rectangle;**

**import javafx.scene.shape.Shape;**

**import javafx.util.Duration;**

**import java.net.URL;**

**import java.util.ResourceBundle;**

**import java.util.List;**

**import java.util.ArrayList;**

**import java.util.stream.\*;**

**import java.util.Optional;**

**public class Controller implements Initializable {**

**private static final int COLUMNS =7;**

**private static final int ROWS=6;**

**private static final int CIRCLE\_DIAMETER =80;**

**private static final String discColor1 ="#24303E";**

**private static final String disccolor2 ="#4CAA88";**

**private static String PLAYER\_ONE ="Player One";**

**private static String PLAYER\_TWO ="Player Two";**

**private boolean isPlayerOneTurn = true;**

**private Disc[][] insertedDiscsArray = new Disc[ROWS][COLUMNS]; //FOR STRUCTURAL CHANGE**

**@FXML**

**public GridPane rootGridPane;**

**@FXML**

**public Pane insertedDiscsPane;**

**@FXML**

**public Label playerNameLabel;**

**@FXML**

**public TextField playerOneTextField, playerTwoTextField;**

**@FXML**

**public Button setNamesButton;**

**private boolean isAllowedToInsert = true;**

**public void createPlayground(){**

**94**

**setNamesButton.setOnAction(event -> {**

**PLAYER\_ONE = playerOneTextField.getText();**

**PLAYER\_TWO = playerTwoTextField.getText();**

**});**

**Shape rectangleWithHoles = createGameStructuralGrid();**

**rootGridPane.add(rectangleWithHoles,0,1);**

**List<Rectangle> rectangleList = createClickableColumns();**

**for(Rectangle rectangle: rectangleList){**

**rootGridPane.add(rectangle, 0, 1);**

**}**

**}**

**private Shape createGameStructuralGrid(){**

**Shape rectangleWithHoles = new Rectangle((COLUMNS + 1) \* CIRCLE\_DIAMETER, (ROWS + 1) \***

**CIRCLE\_DIAMETER);**

**for(int row=0; row< ROWS; row++){**

**for(int col = 0; col< COLUMNS; col++){**

**Circle circle =new Circle();**

**circle.setRadius(CIRCLE\_DIAMETER / 2);**

**circle.setCenterX(CIRCLE\_DIAMETER / 2);**

**circle.setCenterY(CIRCLE\_DIAMETER / 2);**

**circle.setSmooth(true);**

**circle.setTranslateX(col \* (CIRCLE\_DIAMETER + 5) + CIRCLE\_DIAMETER / 4);**

**circle.setTranslateY(row \* (CIRCLE\_DIAMETER + 5) + CIRCLE\_DIAMETER / 4);**

**rectangleWithHoles = Shape.subtract(rectangleWithHoles, circle);**

**}**

**}**

**rectangleWithHoles.setFill(Color.WHITE);**

**return rectangleWithHoles;**

**}**

**private List<Rectangle> createClickableColumns(){**

**List<Rectangle> rectangleList = new ArrayList<>();**

**for(int col =0; col < COLUMNS; col++){**

**Rectangle rectangle = new Rectangle(CIRCLE\_DIAMETER,(ROWS + 1) \* CIRCLE\_DIAMETER);**

**rectangle.setFill(Color.TRANSPARENT);**

**95**

**rectangle.setTranslateX(col \* (CIRCLE\_DIAMETER + 5) + CIRCLE\_DIAMETER / 4);**

**rectangle.setOnMouseEntered(event -> rectangle.setFill(Color.valueOf("#eeeeee26")));**

**rectangle.setOnMouseExited(event -> rectangle.setFill(Color.TRANSPARENT));**

**final int column = col;**

**rectangle.setOnMouseClicked(event -> {**

**if(isAllowedToInsert) {**

**isAllowedToInsert = false;**

**insertDisc(new Disc(isPlayerOneTurn), column);**

**}**

**});**

**rectangleList.add(rectangle);**

**}**

**return rectangleList;**

**}**

**private void insertDisc(Disc disc, int column){**

**int row = ROWS -1;**

**while(row>=0){**

**if(getDiscIfPresent(row,column) == null)**

**break;**

**row--;**

**}**

**if (row<0) //If it is full, we cannot insert anymore disc**

**return;**

**insertedDiscsArray[row][column] = disc; //For Structural Change: For developers**

**insertedDiscsPane.getChildren().add(disc); //for players**

**disc.setTranslateX(column \* (CIRCLE\_DIAMETER +5) + CIRCLE\_DIAMETER / 4);**

**int currentRow = row;**

**TranslateTransition translateTransition = new TranslateTransition(Duration.seconds(0.5), disc);**

**disc.setTranslateY(row \* (CIRCLE\_DIAMETER + 5) + CIRCLE\_DIAMETER / 4);**

**translateTransition.setOnFinished(event -> {**

**isAllowedToInsert = true;**

**if(gameEnded(currentRow,column)) {**

**gameOver();**

**return;**

**}**

**96**

**isPlayerOneTurn = !isPlayerOneTurn;**

**playerNameLabel.setText(isPlayerOneTurn? PLAYER\_ONE : PLAYER\_TWO);**

**});**

**translateTransition.play();**

**}**

**private boolean gameEnded(int row, int column) {**

**List<Point2D> verticalPoints = IntStream.rangeClosed(row - 3, row + 3)**

**.mapToObj(r -> new Point2D(r, column))**

**.collect(Collectors.toList());**

**List<Point2D> horizontalPoints = IntStream.rangeClosed(column - 3, column + 3)**

**.mapToObj(col -> new Point2D(row, col))**

**.collect(Collectors.toList());**

**Point2D startPoint1 = new Point2D(row - 3,column + 3);**

**List<Point2D> diagonal1Points = IntStream.rangeClosed(0, 6)**

**.mapToObj(i -> startPoint1.add(i, -i))**

**.collect(Collectors.toList());**

**Point2D startPoint2 = new Point2D(column - 3,column + 3);**

**List<Point2D> diagonal2Points = IntStream.rangeClosed(0, 6)**

**.mapToObj(i -> startPoint2.add(i, i))**

**.collect(Collectors.toList());**

**boolean isEnded = checkCombinations(verticalPoints)**

**|| checkCombinations(horizontalPoints)**

**|| checkCombinations(diagonal1Points)**

**|| checkCombinations(diagonal2Points);**

**return isEnded;**

**}**

**private boolean checkCombinations(List<Point2D> points) {**

**int chain=0;**

**for (Point2D point: points) {**

**int rowIndexForArray = (int) point.getX();**

**int columnIndexForArray = (int) point.getY();**

**Disc disc = getDiscIfPresent(rowIndexForArray, columnIndexForArray);**

**if (disc != null && disc.isPlayerOneMove == isPlayerOneTurn){**

**chain++;**

**if(chain==4){**

**return true;**

**}**

**} else {**

**chain = 0;**

**97**

**}**

**}**

**return false;**

**}**

**private Disc getDiscIfPresent(int row, int column) {**

**if(row >=ROWS || row< 0 || column >= COLUMNS || column< 0)**

**return null;**

**return insertedDiscsArray[row][column];**

**}**

**private void gameOver() {**

**String winner = isPlayerOneTurn ? PLAYER\_ONE : PLAYER\_TWO;**

**System.out.println("Winner is:" +winner);**

**Alert alert = new Alert(Alert.AlertType.INFORMATION);**

**alert.setTitle("Connect Four");**

**alert.setHeaderText("The Winner is "+winner);**

**alert.setContentText("Want to play again?");**

**ButtonType yesBtn = new ButtonType("Yes");**

**ButtonType noBtn = new ButtonType("No, Exit");**

**alert.getButtonTypes().setAll(yesBtn, noBtn);**

**Platform.runLater(() -> {**

**Optional<ButtonType> btnClicked = alert.showAndWait();**

**if(btnClicked.isPresent() && btnClicked.get() == yesBtn) {**

**//...user chose YES so RESET the game**

**resetGame();**

**}else {**

**//...user chose NO so EXIT the game**

**Platform.exit();**

**System.exit(0);**

**}**

**});**

**}**

**public void resetGame() {**

**insertedDiscsPane.getChildren().clear(); //remove all inserted disc from pane**

**for(int row = 0; row < insertedDiscsArray.length; row++){ //structurally make all elements back to**

**null**

**for(int col = 0; col < insertedDiscsArray[row].length; col++ ){**

**insertedDiscsArray[row][col] = null;**

**98**

**}**

**}**

**isPlayerOneTurn = true;**

**playerNameLabel.setText(PLAYER\_ONE);**

**createPlayground();**

**}**

**private static class Disc extends Circle {**

**private final boolean isPlayerOneMove;**

**public Disc(boolean isPlayerOneMove) {**

**this.isPlayerOneMove = isPlayerOneMove;**

**setRadius(CIRCLE\_DIAMETER / 2);**

**setFill(isPlayerOneMove? Color.valueOf(discColor1): Color.valueOf(disccolor2));**

**setCenterX(CIRCLE\_DIAMETER/2);**

**setCenterY(CIRCLE\_DIAMETER/2);**

**}**

**}**

**@Override**

**public void initialize(URL location, ResourceBundle resources) {**

**}**

**}**

**}**

**FXML**

**<?xml version="1.0" encoding="UTF-8"?>**

**<?import javafx.geometry.\*?>**

**<?import javafx.scene.control.\*?>**

**<?import javafx.scene.layout.\*?>**

**<?import javafx.scene.text.\*?>**

**<GridPane fx:id="rootGridPane" style="-fx-background-color: #D9F7F0;"**

**xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1"**

**fx:controller="com.avinash.connectfour.Controller">**

**<columnConstraints>**

**<ColumnConstraints />**

**92**

**<ColumnConstraints maxWidth="313.0" minWidth="223.0" prefWidth="227.0" />**

**</columnConstraints>**

**<rowConstraints>**

**<RowConstraints minHeight="10.0" prefHeight="30.0" />**

**<RowConstraints />**

**</rowConstraints>**

**<children>**

**<Pane GridPane.columnSpan="2" />**

**<Pane fx:id="insertedDiscsPane" prefHeight="400.0" prefWidth="200.0" GridPane.rowIndex="1" />**

**<VBox style="-fx-background-color: #2B3B4C;" GridPane.columnIndex="1" GridPane.rowIndex="1">**

**<children>**

**<TextField fx:id="playerOneTextField" promptText="Player One Name">**

**<VBox.margin>**

**<Insets left="15.0" right="15.0" top="35.0" />**

**</VBox.margin></TextField>**

**<TextField fx:id="playerTwoTextField" promptText="Player Two Name">**

**<VBox.margin>**

**<Insets left="15.0" right="15.0" />**

**</VBox.margin></TextField>**

**<Button fx:id="setNamesButton" mnemonicParsing="false" prefHeight="26.0" prefWidth="229.0"**

**text="Set Names">**

**<VBox.margin>**

**<Insets bottom="5.0" left="15.0" right="15.0" top="5.0" />**

**</VBox.margin></Button>**

**<Label fx:id="playerNameLabel" alignment="CENTER" prefHeight="18.0" prefWidth="307.0"**

**text="Player One" textFill="WHITE">**

**<font>**

**<Font name="System Bold" size="29.0" />**

**</font>**

**<VBox.margin>**

**<Insets top="70.0" />**

**</VBox.margin></Label>**

**<Label alignment="CENTER" prefHeight="18.0" prefWidth="318.0" text="Turn" textFill="WHITE">**

**<font>**

**<Font size="26.0" />**

**</font></Label>**

**<Region prefHeight="200.0" prefWidth="200.0" VBox.vgrow="ALWAYS" />**

**</children>**

**</VBox>**

**</children>**