WELCOME

package Minesweeper;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.scene.text.FontPosture;

import javafx.scene.text.FontWeight;

import javafx.scene.text.Text;

import javafx.stage.Stage;

public class Welcome extends Application{

@Override

public void start(Stage primaryStage){

Pane pane = new Pane();

pane.setPadding(new Insets(100, 100, 100, 100));

Text text1 = new Text(600, 40, " M I N E S W E E P E R G A M E S ");

Text text2 = new Text(530,60, " Y O U R G A M I N G W O R L D ");

text1.setFont(Font.font("Platino", FontWeight.BOLD,

FontPosture.ITALIC, 15));

text1.setFill(Color.BLACK);

text2.setFont(Font.font("Comic Sans MS", FontWeight.BOLD,

FontPosture.REGULAR, 15));

text2.setFill(Color.BROWN);

HBox hBox = new HBox();

hBox.setPadding(new Insets(580, 580, 540, 470));

hBox.setSpacing(10);

hBox.setAlignment(Pos.BOTTOM\_CENTER);

ImageView imageAbout = new ImageView(new Image("http://icons.iconarchive.com/icons/zhoolego/material/24/Games-icon.png"));

Button btAbout = new Button(" H O W T O P L A Y ? ");

btAbout.setGraphic(imageAbout);

ImageView imageGame = new ImageView(new Image("http://icons.iconarchive.com/icons/hopstarter/soft-scraps/24/Games-icon.png"));

Button btGame = new Button("G A M E");

btGame.setGraphic(imageGame);

ImageView imageSuggest = new ImageView(new Image("http://icons.iconarchive.com/icons/ccard3dev/dynamic-yosemite/24/Utilities-Feedback-Assistant-icon.png"));

Button btfb = new Button("S U G G E S T I O N");

btfb.setGraphic(imageSuggest);

hBox.getChildren().addAll(btAbout, btGame, btfb);

pane.getChildren().add(getHBox());

pane.getChildren().add(text1);

pane.getChildren().add(text2);

pane.getChildren().add (hBox);

btAbout.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

AboutUs about= new AboutUs();

about.start(primaryStage);

}

});

btfb.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Feedback fb= new Feedback();

fb.start(primaryStage);

}

});

btGame.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Game Game= new Game();

Game.start(primaryStage);

}

});

Scene scene = new Scene(pane);

primaryStage.setTitle("Hello, Welcome!"); // Set the stage title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show();

primaryStage.centerOnScreen();// Display the stage

}

public static void main(String[] args) {

launch(args);

}

public HBox getHBox() {

HBox hBox = new HBox(95);

hBox.setPadding(new Insets(100, 800, 200, 300));

ImageView imageView = new ImageView(new Image("http://i.imgur.com/X02uxgK.png"));

hBox.getChildren().add(imageView);

return hBox;

}}

Game

package Minesweeper;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.HBox;

import javafx.scene.layout.Pane;

import javafx.stage.Stage;

public class Game extends Application{

@Override

public void start(Stage primaryStage){

Pane pane = new Pane();

pane.setPadding(new Insets(80, 80, 80, 80));

HBox hBox = new HBox();

hBox.setPadding(new Insets(580, 580, 540, 470));

hBox.setSpacing(10);

hBox.setAlignment(Pos.BOTTOM\_CENTER);

ImageView imageEasy = new ImageView(new Image("http://icons.iconarchive.com/icons/icons8/windows-8/24/Numbers-1-Black-icon.png"));

Button btEasy = new Button(" Beginner ");

btEasy.setGraphic(imageEasy);

ImageView imageMedium = new ImageView(new Image("http://icons.iconarchive.com/icons/icons8/windows-8/24/Numbers-2-Black-icon.png"));

Button btMedium= new Button(" Intermediate");

btMedium.setGraphic(imageMedium);

ImageView imageHard = new ImageView(new Image("http://icons.iconarchive.com/icons/icons8/windows-8/24/Numbers-3-Black-icon.png"));

Button btHard = new Button(" Expert ");

btHard.setGraphic(imageHard);

ImageView imageHome = new ImageView(new Image("http://icons.iconarchive.com/icons/icons8/windows-8/24/Very-Basic-Home-icon.png"));

Button btHome = new Button(" Return to HOME ");

btHome.setGraphic(imageHome);

hBox.getChildren().addAll(btEasy, btMedium, btHard,btHome);

pane.getChildren().add(getHBox());

pane.getChildren().add (hBox);

btEasy.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Easy pst = new Easy();

try {

pst.start(primaryStage);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btMedium.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Medium Medium = new Medium();

try {

Medium.start(primaryStage);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btHard.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Hard dsrt= new Hard();

try {

dsrt.start(primaryStage);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btHome.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

Welcome wc= new Welcome();

wc.start(primaryStage);

}

});

Scene scene = new Scene(pane);

primaryStage.setTitle("Game Play"); // Set the stage title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show(); // Display the stage

}

public HBox getHBox() {

HBox hBox = new HBox(95);

hBox.setPadding(new Insets(80, 10, 200, 300));

ImageView imageView = new ImageView(new Image("http://i.imgur.com/zCwUrXH.png"));

hBox.getChildren().add(imageView);

return hBox;

}

public static void main(String[] args) {

launch(args);

}}

About US

**package** Minesweeper;

**import** javafx.application.Application;

**import** javafx.event.ActionEvent;

**import** javafx.event.EventHandler;

**import** javafx.geometry.Insets;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.BorderPane;

**import** javafx.scene.layout.HBox;

**import** javafx.scene.layout.VBox;

**import** javafx.stage.Stage;

**import** javafx.scene.image.Image;

**import** javafx.scene.image.ImageView;

**public** **class** AboutUs **extends** Application {

@Override // Override the start method in the Application class

**public** **void** start(Stage primaryStage) {

HBox hBox = **new** HBox(15);

hBox.setPadding(**new** Insets(15, 150, 15, 150));

hBox.setStyle("-fx-background-color: pink");

Button btback= **new** Button ("Back");

btback.setOnAction(**new** EventHandler<ActionEvent>() {

@Override **public** **void** handle(ActionEvent e) {

Welcome wel= **new** Welcome();

wel.start(primaryStage);

}

});

// Create a border pane

BorderPane pane = **new** BorderPane();

// Place nodes in the pane

pane.setTop(getVBox());

pane.setCenter(btback);

// Create a scene and place it in the stage

Scene scene = **new** Scene(pane);

primaryStage.setTitle("About MinesWeeper Game!"); // Set the stage title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show(); // Display the stage

primaryStage.centerOnScreen();

}

**public** VBox getVBox() {

VBox vBox = **new** VBox(15);

vBox.setPadding(**new** Insets(20, 5, 5, 5));

Label con=**new** Label ("Good Luck!");

con.setStyle("-fx-font-weight: bold");

vBox.getChildren().add(con);

Image image = **new** Image("http://icons.iconarchive.com/icons/aha-soft/free-large-boss/72/Security-Guard-icon.png");

Label email = **new** Label("1.The purpose of the game is to open all the cells of the board which do not contain a bomb.\n"

+ "2.Once you open the game stage, time will be count to know how much time you take to open all cell without the bomb."

+ "3.You lose if you set off a bomb cell.\n"

+ "4.Every non-bomb cell you open will tell you the total number of bombs in the eight neighboring cells.\n"

+ "5.When you successful open all cell that without a bomb,you will get pop up message that tell you are the winner.\n"

+ "6.When you accidently open a cell that contain a bomb,you will get pop up message that tell you are lost.\n"

+ "7.You can click back button to move to another stage or go to main page."

+"\n\nHappy mine hunting!-APPLE PEN TEAM");

email.setGraphic(**new** ImageView(image));

vBox.getChildren().add(email);

**return** vBox;

}

/\*\*

\* The main method is only needed for the IDE with limited

\* JavaFX support. Not needed for running from the command line.

\*/

}

EASY

**package Minesweeper;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Timer;**

**import java.util.TimerTask;**

**import javax.swing.JOptionPane;**

**import javafx.application.Application;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.scene.Parent;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.TextField;**

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

**import javafx.scene.layout.StackPane;**

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

**import javafx.scene.paint.Paint;**

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

**import javafx.scene.text.Font;**

**import javafx.scene.text.Text;**

**import javafx.stage.Stage;**

**public class Easy extends Application {**

**private static final int TILE\_SIZE = 60;**

**private static final int W = 400;**

**private static final int H = 400;**

**private static final int X\_TILES = W / TILE\_SIZE;**

**private static final int Y\_TILES = H / TILE\_SIZE;**

**private Tile[][] grid = new Tile[X\_TILES][Y\_TILES];**

**private Scene scene;**

**private Button btback;**

**private int win, open;**

**private Parent createContent() {**

**Pane root = new Pane();**

**root.setPrefSize(W, H);**

**TextField txt = new TextField();**

**// long start = System.currentTimeMillis();**

**StopWatch st = new StopWatch();**

**win = 0; open = 0;**

**btback= new Button ("Back");**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**//if number < 0.2 it will generate as location that have bomb**

**Tile tile = new Tile(x, y, Math.random() < 0.2);**

**grid[x][y] = tile;**

**if (!tile.hasBomb) win++;**

**root.getChildren().add(tile);**

**}**

**}**

**txt.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(txt);**

**txt.setEditable(false);**

**txt.setText(String.valueOf(st.getTime()));**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**Tile tile = grid[x][y];**

**if (tile.hasBomb)**

**continue;**

**long bombs = getNeighbors(tile).stream().filter(t -> t.hasBomb).count();**

**if (bombs > 0)**

**tile.text.setText(String.valueOf(bombs));**

**}**

**}**

**TimerTask task = new TimerTask() {**

**@Override**

**public void run() {**

**txt.setText(String.valueOf(st.getTime()));**

**}**

**};**

**Timer time = new Timer();**

**time.schedule(task, 1000l, 1000l);**

**btback.setTranslateX(X\_TILES/2 \* TILE\_SIZE);**

**btback.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(btback);**

**// root.getChildren().add(new TextField());**

**// System.err.println(win);**

**return root;**

**}**

**private class StopWatch {**

**long start;**

**public StopWatch() {**

**start = System.currentTimeMillis();**

**}**

**private long getTime() {**

**long now = System.currentTimeMillis();**

**return (now - start) / 1000;**

**}**

**}**

**private List<Tile> getNeighbors(Tile tile) {**

**List<Tile> neighbors = new ArrayList<>();**

**// ttt**

**// tXt**

**// ttt**

**//BFS concept**

**//clockwise coordinate**

**int[] points = new int[] {**

**-1, -1,**

**-1, 0,**

**-1, 1,**

**0, -1,**

**0, 1,**

**1, -1,**

**1, 0,**

**1, 1**

**};**

**for (int i = 0; i < points.length; i++) {**

**//make sure**

**int dx = points[i];**

**int dy = points[++i];**

**//set the real position**

**int newX = tile.x + dx;**

**int newY = tile.y + dy;**

**if (newX >= 0 && newX < X\_TILES**

**&& newY >= 0 && newY < Y\_TILES) {**

**neighbors.add(grid[newX][newY]);**

**}**

**}**

**return neighbors;**

**}**

**private class Tile extends StackPane {**

**private int x, y;**

**private boolean hasBomb;**

**private boolean isOpen = false;**

**private Rectangle border = new Rectangle(TILE\_SIZE - 2, TILE\_SIZE - 2);**

**// p = border.getFill();**

**private Text text = new Text();**

**public Tile(int x, int y, boolean hasBomb) {**

**border.getFill();**

**this.x = x;**

**this.y = y;**

**this.hasBomb = hasBomb;**

**border.setStroke(Color.LIGHTGRAY);**

**text.setFont(Font.font(18));**

**text.setText(hasBomb ? "X" : "");**

**text.setVisible(false);**

**// border.setFill(null);**

**getChildren().addAll(border, text);**

**setTranslateX(x \* TILE\_SIZE);**

**setTranslateY(y \* TILE\_SIZE);**

**setOnMouseClicked(e -> open());**

**}**

**public void open() {**

**if (isOpen)**

**return;**

**if (hasBomb) {**

**// System.out.println("Game Over");**

**JOptionPane.showMessageDialog(null, "Game Over", "Game Over", JOptionPane.ERROR\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**isOpen = true;**

**text.setVisible(true);**

**border.setFill(null);**

**open++;**

**// System.err.println(win+ " " +open);**

**if (open == win) {**

**JOptionPane.showMessageDialog(null, "You are winner!", "Completed...", JOptionPane.INFORMATION\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**if (text.getText().isEmpty()) {**

**getNeighbors(this).forEach(Tile::open);**

**}**

**}**

**}**

**@Override**

**public void start(Stage stage) throws Exception {**

**scene = new Scene(createContent());**

**btback.setOnAction(new EventHandler<ActionEvent>() {**

**@Override public void handle(ActionEvent e) {**

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

**Game.start(stage);**

**}**

**});**

**stage.setScene(scene);**

**stage.show();**

**}**

**public static void main(String[] args) {**

**launch(args);**

**}}**

MEDIUM

**package Minesweeper;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Timer;**

**import java.util.TimerTask;**

**import javax.swing.JOptionPane;**

**import javafx.application.Application;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.scene.Parent;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.TextField;**

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

**import javafx.scene.layout.StackPane;**

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

**import javafx.scene.paint.Paint;**

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

**import javafx.scene.text.Font;**

**import javafx.scene.text.Text;**

**import javafx.stage.Stage;**

**public class Medium extends Application {**

**private static final int TILE\_SIZE = 40;**

**private static final int W = 500;**

**private static final int H = 500;**

**private static final int X\_TILES = W / TILE\_SIZE;**

**private static final int Y\_TILES = H / TILE\_SIZE;**

**private Tile[][] grid = new Tile[X\_TILES][Y\_TILES];**

**private Scene scene;**

**private Button btback;**

**private int win, open;**

**private Parent createContent() {**

**Pane root = new Pane();**

**root.setPrefSize(W, H);**

**// root.setPrefSize(root.getPrefWidth(), root.getPrefHeight());**

**TextField txt = new TextField();**

**// long start = System.currentTimeMillis();**

**StopWatch st = new StopWatch();**

**win = 0; open = 0;**

**btback= new Button ("Back");**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**Tile tile = new Tile(x, y, Math.random() < 0.2);**

**grid[x][y] = tile;**

**if (!tile.hasBomb) win++;**

**root.getChildren().add(tile);**

**}**

**}**

**txt.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(txt);**

**txt.setEditable(false);**

**txt.setText(String.valueOf(st.getTime()));**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**Tile tile = grid[x][y];**

**if (tile.hasBomb)**

**continue;**

**long bombs = getNeighbors(tile).stream().filter(t -> t.hasBomb).count();**

**if (bombs > 0)**

**tile.text.setText(String.valueOf(bombs));**

**}**

**}**

**TimerTask task = new TimerTask() {**

**@Override**

**public void run() {**

**txt.setText(String.valueOf(st.getTime()));**

**}**

**};**

**Timer time = new Timer();**

**time.schedule(task, 1000l, 1000l);**

**btback.setTranslateX(X\_TILES/2 \* TILE\_SIZE);**

**btback.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(btback);**

**// root.getChildren().add(new TextField());**

**System.err.println(win);**

**// root.setPrefSize(root.getPrefWidth(), root.getPrefHeight());**

**return root;**

**}**

**private class StopWatch {**

**long start;**

**public StopWatch() {**

**start = System.currentTimeMillis();**

**}**

**private long getTime() {**

**long now = System.currentTimeMillis();**

**return (now - start) / 1000;**

**}**

**}**

**private List<Tile> getNeighbors(Tile tile) {**

**List<Tile> neighbors = new ArrayList<>();**

**// ttt**

**// tXt**

**// ttt**

**int[] points = new int[] {**

**-1, -1,**

**-1, 0,**

**-1, 1,**

**0, -1,**

**0, 1,**

**1, -1,**

**1, 0,**

**1, 1**

**};**

**for (int i = 0; i < points.length; i++) {**

**int dx = points[i];**

**int dy = points[++i];**

**int newX = tile.x + dx;**

**int newY = tile.y + dy;**

**if (newX >= 0 && newX < X\_TILES**

**&& newY >= 0 && newY < Y\_TILES) {**

**neighbors.add(grid[newX][newY]);**

**}**

**}**

**return neighbors;**

**}**

**private class Tile extends StackPane {**

**private int x, y;**

**private boolean hasBomb;**

**private boolean isOpen = false;**

**private Paint p;**

**private Rectangle border = new Rectangle(TILE\_SIZE - 2, TILE\_SIZE - 2);**

**// p = border.getFill();**

**private Text text = new Text();**

**public Tile(int x, int y, boolean hasBomb) {**

**p = border.getFill();**

**this.x = x;**

**this.y = y;**

**this.hasBomb = hasBomb;**

**border.setStroke(Color.LIGHTGRAY);**

**text.setFont(Font.font(18));**

**text.setText(hasBomb ? "X" : "");**

**text.setVisible(false);**

**// border.setFill(null);**

**getChildren().addAll(border, text);**

**setTranslateX(x \* TILE\_SIZE);**

**setTranslateY(y \* TILE\_SIZE);**

**setOnMouseClicked(e -> open());**

**}**

**public void open() {**

**if (isOpen)**

**return;**

**if (hasBomb) {**

**// System.out.println("Game Over");**

**JOptionPane.showMessageDialog(null, "Game Over", "Game Over", JOptionPane.ERROR\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**isOpen = true;**

**text.setVisible(true);**

**border.setFill(null);**

**open++;**

**System.err.println(win+ " " +open);**

**if (open == win) {**

**JOptionPane.showMessageDialog(null, "You are winner!", "Completed...", JOptionPane.INFORMATION\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**if (text.getText().isEmpty()) {**

**getNeighbors(this).forEach(Tile::open);**

**}**

**}**

**}**

**@Override**

**public void start(Stage stage) throws Exception {**

**scene = new Scene(createContent());**

**btback.setOnAction(new EventHandler<ActionEvent>() {**

**@Override public void handle(ActionEvent e) {**

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

**Game.start(stage);**

**}**

**});**

**stage.setScene(scene);**

**stage.show();**

**}**

**public static void main(String[] args) {**

**launch(args);**

**}}**

HARD

**package Minesweeper;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Timer;**

**import java.util.TimerTask;**

**import javax.swing.JOptionPane;**

**import javafx.application.Application;**

**import javafx.event.ActionEvent;**

**import javafx.event.EventHandler;**

**import javafx.scene.Parent;**

**import javafx.scene.Scene;**

**import javafx.scene.control.Button;**

**import javafx.scene.control.TextField;**

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

**import javafx.scene.layout.StackPane;**

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

**import javafx.scene.paint.Paint;**

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

**import javafx.scene.text.Font;**

**import javafx.scene.text.Text;**

**import javafx.stage.Stage;**

**public class Hard extends Application {**

**private static final int TILE\_SIZE = 40;**

**private static final int W = 700;**

**private static final int H = 700;**

**private static final int X\_TILES = W / TILE\_SIZE;**

**private static final int Y\_TILES = H / TILE\_SIZE;**

**private Tile[][] grid = new Tile[X\_TILES][Y\_TILES];**

**private Scene scene;**

**private Button btback;**

**private int win, open;**

**private Parent createContent() {**

**Pane root = new Pane();**

**root.setPrefSize(W, H);**

**TextField txt = new TextField();**

**// long start = System.currentTimeMillis();**

**StopWatch st = new StopWatch();**

**win = 0; open = 0;**

**btback= new Button ("Back");**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**Tile tile = new Tile(x, y, Math.random() < 0.2);**

**grid[x][y] = tile;**

**if (!tile.hasBomb) win++;**

**root.getChildren().add(tile);**

**}**

**}**

**txt.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(txt);**

**txt.setEditable(false);**

**txt.setText(String.valueOf(st.getTime()));**

**for (int y = 0; y < Y\_TILES; y++) {**

**for (int x = 0; x < X\_TILES; x++) {**

**Tile tile = grid[x][y];**

**if (tile.hasBomb)**

**continue;**

**long bombs = getNeighbors(tile).stream().filter(t -> t.hasBomb).count();**

**if (bombs > 0)**

**tile.text.setText(String.valueOf(bombs));**

**}**

**}**

**TimerTask task = new TimerTask() {**

**@Override**

**public void run() {**

**txt.setText(String.valueOf(st.getTime()));**

**}**

**};**

**Timer time = new Timer();**

**time.schedule(task, 1000l, 1000l);**

**btback.setTranslateX(X\_TILES/2 \* TILE\_SIZE);**

**btback.setTranslateY(Y\_TILES \* TILE\_SIZE);**

**root.getChildren().add(btback);**

**// root.getChildren().add(new TextField());**

**System.err.println(win);**

**return root;**

**}**

**private class StopWatch {**

**long start;**

**public StopWatch() {**

**start = System.currentTimeMillis();**

**}**

**private long getTime() {**

**long now = System.currentTimeMillis();**

**return (now - start) / 1000;**

**}**

**}**

**private List<Tile> getNeighbors(Tile tile) {**

**List<Tile> neighbors = new ArrayList<>();**

**// ttt**

**// tXt**

**// ttt**

**int[] points = new int[] {**

**-1, -1,**

**-1, 0,**

**-1, 1,**

**0, -1,**

**0, 1,**

**1, -1,**

**1, 0,**

**1, 1**

**};**

**for (int i = 0; i < points.length; i++) {**

**int dx = points[i];**

**int dy = points[++i];**

**int newX = tile.x + dx;**

**int newY = tile.y + dy;**

**if (newX >= 0 && newX < X\_TILES**

**&& newY >= 0 && newY < Y\_TILES) {**

**neighbors.add(grid[newX][newY]);**

**}**

**}**

**return neighbors;**

**}**

**private class Tile extends StackPane {**

**private int x, y;**

**private boolean hasBomb;**

**private boolean isOpen = false;**

**private Paint p;**

**private Rectangle border = new Rectangle(TILE\_SIZE - 2, TILE\_SIZE - 2);**

**// p = border.getFill();**

**private Text text = new Text();**

**public Tile(int x, int y, boolean hasBomb) {**

**p = border.getFill();**

**this.x = x;**

**this.y = y;**

**this.hasBomb = hasBomb;**

**border.setStroke(Color.LIGHTGRAY);**

**text.setFont(Font.font(18));**

**text.setText(hasBomb ? "X" : "");**

**text.setVisible(false);**

**// border.setFill(null);**

**getChildren().addAll(border, text);**

**setTranslateX(x \* TILE\_SIZE);**

**setTranslateY(y \* TILE\_SIZE);**

**setOnMouseClicked(e -> open());**

**}**

**public void open() {**

**if (isOpen)**

**return;**

**if (hasBomb) {**

**// System.out.println("Game Over");**

**JOptionPane.showMessageDialog(null, "Game Over", "Game Over", JOptionPane.ERROR\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**isOpen = true;**

**text.setVisible(true);**

**border.setFill(null);**

**open++;**

**System.err.println(win+ " " +open);**

**if (open == win) {**

**JOptionPane.showMessageDialog(null, "You are winner!", "Completed...", JOptionPane.INFORMATION\_MESSAGE);**

**scene.setRoot(createContent());**

**return;**

**}**

**if (text.getText().isEmpty()) {**

**getNeighbors(this).forEach(Tile::open);**

**}**

**}**

**}**

**@Override**

**public void start(Stage stage) throws Exception {**

**scene = new Scene(createContent());**

**btback.setOnAction(new EventHandler<ActionEvent>() {**

**@Override public void handle(ActionEvent e) {**

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

**Game.start(stage);**

**}**

**});**

**stage.setScene(scene);**

**stage.show();**

**}**

**public static void main(String[] args) {**

**launch(args);**

**}}**

FEEDBACK

package Minesweeper;

import Gui.DescriptionPane;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.HPos;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Node;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.image.ImageView;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

public class Feedback {

public TextField fb= new TextField();

public void start(Stage primaryStage) {

// Declare and create a description pane

DescriptionPane descriptionPane = new DescriptionPane();

GridPane pane = new GridPane();

pane.setPadding(new Insets(100, 100, 100, 100));

pane.setAlignment(Pos.TOP\_CENTER);

pane.setHgap(5.5);

pane.setVgap(5.5);

// Place nodes in the pane

pane.add(new Label("Name:"), 0, 0);

pane.add(fb, 1, 0);

GridPane gpane = new GridPane();

Button btsubmit = new Button("Submit Suggestion");

gpane.add(btsubmit, 0, 8);

gpane.add(descriptionPane, 0, 1);

GridPane.setHalignment(btsubmit, HPos.CENTER);

btsubmit.setOnAction(new EventHandler<ActionEvent>() {

@Override public void handle(ActionEvent e) {

System.out.println("Feedback :"+fb.getText()); // will be printed at console

Stage stage = new Stage(); // Create a new stage

// Set a scene with a button in the stage

stage.setScene(new Scene(new Button("Thank you for you time!"), 200, 100));

stage.show(); // Display the stage

}

});

// Set title, text and image in the description pane

descriptionPane.setTitle("Help Us!-APPLE PEN TEAM.\n");

String description = "Did you enjoy the game?\nHave u had seen and experienced any flaws in this game?"+

"\nWe welcome your suggestions and comments on this game.";

descriptionPane.setImageView(new ImageView("https://s-media-cache-ak0.pinimg.com/564x/3d/04/ba/3d04baca35bfef842d35c1f54ef1e1c4.jpg "));

descriptionPane.setDescription(description);

descriptionPane.setBottom(fb);

// Create a scene and place it in the stage

Scene scene = new Scene(gpane, 950, 500);

primaryStage.setTitle("Suggestion"); // Set the stage title

primaryStage.setScene(scene); // Place the scene in the stage

primaryStage.show(); // Display the stage

primaryStage.centerOnScreen();

}

/\*\*

\* The main method is only needed for the IDE with limited

\* JavaFX support. Not needed for running from the command line.

\*/

}