import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.SwingConstants;

/\*\*

\* This class initializes the screen size and takes the input values for first

\* and last name. Also, takes input of the no of asteroids. Start Button

\* initiates the game. It also disposes the JFrame window when it closes.

\*

\* @author Ram Reddy & Eric Feng

\* @version 5.24.20

\*/

public class StartScreen extends JFrame {

private JTextField getNumberObstacles, firstName, lastName;

private JLabel welcome, labelForObstacles, labelForFirstName, labelForLastName;

private JPanel startPanel;

private JButton buttonToStart;

private int numberOfObstacles;

/\*\*

\* Player Enter First and Last name and enters the number of asteroids.

\*/

public StartScreen() {

// Set the initial screen size

this.setSize(600, 250);

// Add First, Last, No of obstacles to the screen

firstName = new JTextField(20);

lastName = new JTextField(20);

labelForFirstName = new JLabel("Enter First Name:");

labelForFirstName.setFont(new Font("Comic-sans", Font.PLAIN, 20));

firstName.setFont(new Font("Comic-sans", Font.PLAIN, 25));

lastName.setFont(new Font("Comic-sans", Font.PLAIN, 25));

labelForLastName = new JLabel("Enter Last Name:");

labelForLastName.setFont(new Font("Comic-sans", Font.PLAIN, 20));

getNumberObstacles = new JTextField(20);

getNumberObstacles.setFont(new Font("Comic-sans", Font.PLAIN, 25));

labelForObstacles = new JLabel("Number of Asteroids:");

labelForObstacles.setFont(new Font("Comic-sans", Font.PLAIN, 20));

// Add Start button on the screen.

buttonToStart = new JButton("Press to Begin");

buttonToStart.setFont(new Font("Comic-sans", Font.PLAIN, 25));

buttonToStart.setBackground(Color.GRAY);

buttonToStart.setForeground(Color.BLACK);

startPanel = new JPanel();

// Title of the game

welcome = new JLabel("Destroy the Asteroids!");

welcome.setFont(new Font("Comic-sans", Font.PLAIN, 30));

// Grid Layout

GridLayout gLayout = new GridLayout(0, 2);

this.add(welcome, BorderLayout.NORTH);

startPanel.setLayout(gLayout);

startPanel.add(labelForFirstName);

startPanel.add(firstName);

labelForFirstName.setHorizontalAlignment(SwingConstants.CENTER);

startPanel.add(labelForLastName);

startPanel.add(lastName);

labelForLastName.setHorizontalAlignment(SwingConstants.CENTER);

startPanel.add(labelForObstacles);

startPanel.add(getNumberObstacles);

labelForObstacles.setHorizontalAlignment(SwingConstants.CENTER);

welcome.setHorizontalAlignment(SwingConstants.CENTER);

this.add(buttonToStart, BorderLayout.SOUTH);

this.add(startPanel, BorderLayout.CENTER);

this.centerWindow();

buttonToStart.addActionListener(new ActionListener() {

// add an action listener

// This method uses action listeners and checks no of asteroids

public void actionPerformed(ActionEvent e) {

if (e.getSource() == buttonToStart) {

if (!getNumberObstacles.getText().equals("")) {

// makes sure user enters a value

try {

numberOfObstacles = Integer.parseInt(getNumberObstacles.getText());

} catch (NumberFormatException e1) {

JOptionPane.showMessageDialog(null, "Please enter the number");

}

} else {

JOptionPane.showMessageDialog(null, "Please enter the number of Asteroids");

}

if (numberOfObstacles > 0) {

new AsteroidGame(numberOfObstacles);

}

}

}

});

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setVisible(true);

}

/\*\*

\* Close window

\*/

private void closeWindow() {

this.dispose();

}

// center the window based on the dimensions of the screen size.

private void centerWindow() {

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

int width = this.getSize().width;

int height = this.getSize().height;

int x = (dim.width - width) / 2;

int y = (dim.height - height) / 2;

this.setLocation(x, y);

}

/\*\*

\* Returns the player's first name

\*

\* @return String firstName - First name of the player

\*/

public String getFirstName() {

return firstName.getText();

}

/\*\*

\* Return the player's last name

\*

\* @return String lastName - Last name of the player

\*/

public String getLastName() {

return lastName.getText();

}

}