/\*

\* 用动画形式显示连锁反应的过程和效果

\* 连锁反应是指原子核中质子、中子与电子的反应过程，是原子弹的实现原理。

\* (1)初始的原子核数目和位置由用户选择或随机生成。

\* (2)初始中子的速度和方向可以由用户选择

\* (3)在没有原子核的情况下，这些中子在封闭空间中自由运行（不用互相撞击）。

\*/

package application.Main;

import application.File.NeutronShow;

import application.File.RandomlyGeneratedShow;

import application.File.UserSelect;

import application.File.UserSelectShow;

import application.Help.Help;

import application.Model.NeutronModel;

import application.Model.NucleusModel;

import javafx.application.Application;

import javafx.application.Platform;

import javafx.scene.Scene;

import javafx.scene.control.Menu;

import javafx.scene.control.MenuBar;

import javafx.scene.control.MenuItem;

import javafx.scene.layout.BorderPane;

import javafx.scene.paint.Color;

import javafx.stage.Stage;

import javafx.event.EventHandler;

import javafx.event.ActionEvent;

public class Main extends Application {

private Stage primaryStage;

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

this.primaryStage = primaryStage;

this.primaryStage.setTitle("链式反应动画显示");

//背景

BorderPane root = new BorderPane();

Scene scene = new Scene(root, 450, 350, Color.WHITE);

// 菜单栏

MenuBar menuBar = new MenuBar();

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

root.setTop(menuBar);

//每个子菜单

Menu fileMenu = new Menu("File");

//用户选择

MenuItem selectMenuItem = new MenuItem("用户选择");

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

public void handle(ActionEvent e) {

try {

getUserSelect();

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

//随机生成

MenuItem randomMenuItem = new MenuItem("随机生成");

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

public void handle(ActionEvent e) {

try {

getRandomlyGeneratedShow();

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

//无原子核情况

MenuItem neutronMenuItem = new MenuItem("无原子核");

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

public void handle(ActionEvent e) {

try {

getNeutronShow();

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

fileMenu.getItems().addAll(selectMenuItem,randomMenuItem,neutronMenuItem);

//动画显示说明

Menu helpMenu = new Menu("Help");

MenuItem helpMenuItem = new MenuItem("动画说明");

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

public void handle(ActionEvent e) {

try {

getHelpShow();

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

helpMenu.getItems().add(helpMenuItem);

//退出

Menu exitMenu = new Menu("Exit");

MenuItem exitMenuItem = new MenuItem("退出");

exitMenuItem.setOnAction(actionEvent -> Platform.exit());

exitMenu.getItems().add(exitMenuItem);

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

primaryStage.setScene(scene);

primaryStage.show();

}

//跳转用户选择界面

public void getUserSelect() throws Exception {

UserSelect userselect =new UserSelect();

userselect.showWindow();

}

//跳转用户选择动画显示界面

public void getUserSelectShow() throws Exception {

UserSelectShow userselectshow =new UserSelectShow();

userselectshow.showWindow();

}

//跳转无原子核界面

public void getNeutronShow() throws Exception {

NeutronShow neutron=new NeutronShow();

neutron.showWindow();

}

//跳转随机生成动画界面

public void getRandomlyGeneratedShow() throws Exception {

RandomlyGeneratedShow random=new RandomlyGeneratedShow();

random.showWindows();

}

//跳转帮助界面

public void getHelpShow() throws Exception {

Help help=new Help();

help.showWindows();

}

}