package application.File;

import javafx.event.EventHandler;

import javafx.event.ActionEvent;

import application.Main.Main;

import javafx.application.Application;

import javafx.application.Platform;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.HBox;

import javafx.scene.paint.Color;

import javafx.stage.Stage;

import javafx.scene.text.Font;

public class UserSelect extends Application {

Stage stage=new Stage();

private UserSelectShow show=new UserSelectShow();

private Integer NucleusAmount;原子核数目

private Integer NucleusRightLocation;原子核右边界范围

private Integer NucleusLeftLocation;原子核左边界范围

private Integer NucleusUpLocation;原子核上边界范围

private Integer NucleusDownLocation;原子核下边界范围

private Integer NeutronVelocity;中子速度

private Integer neutrondirections;中子方向，0-左，1-右

private String getAmount;

private String getVelocity;

private String getRightLocation;

private String getLeftLocation;

private String getUpLocation;

private String getDownLocation;

private String getDirection;

@Override

public void start(Stage primaryStage) {

try {

primaryStage.setTitle(用户选择);

Scene scene = new Scene(new Group(), 450, 350);

选择原子核数目

Label Amount=new Label(请输入原子核数目：);

Amount.setTextFill(Color.web(#FF76a3));

Amount.setFont(Font.font (Verdana, 20));

TextField nucleusAmount = new TextField ();

选择中子速度

Label Velocity =new Label(请输入中子速度);

Velocity.setTextFill(Color.web(#FF76a3));

Velocity.setFont(Font.font(Verdana,20));

TextField neutronVelocity=new TextField();

选择原子核位置

Label Location =new Label(请选择原子位置);

Location.setTextFill(Color.web(#FF76a3));

Location.setFont(Font.font(Verdana,20));

Label LocationRL =new Label(请选择原子核左右边界范围);

LocationRL.setTextFill(Color.web(#FF76a3));

LocationRL.setFont(Font.font(Verdana,20));

TextField RightLocation=new TextField();

TextField LeftLocation=new TextField();

Label LocationUD =new Label(请选择原子核上下边界范围);

LocationUD.setTextFill(Color.web(#FF76a3));

LocationUD.setFont(Font.font(Verdana,20));

TextField UpLocation=new TextField();

TextField DownLocation=new TextField();

选择中子方向

Label Direction =new Label(请选择中子方向);

Direction.setTextFill(Color.web(#FF76a3));

Direction.setFont(Font.font(Verdana,20));

TextField NeutronDirection=new TextField();

确定按钮

Button btn = new Button(确定);

HBox hbBtn = new HBox(10);

hbBtn.setAlignment(Pos.BOTTOM\_RIGHT);

hbBtn.getChildren().add(btn);

btn.setOnAction(new EventHandlerActionEvent() {

@Override

public void handle(ActionEvent e) {

原子核数目

getAmount=nucleusAmount.getText();

if(getAmount!=null&&!getAmount.equals()) {

NucleusAmount=Integer.parseInt(getAmount);

}

show.setNucleusAmount(NucleusAmount);

中子速度，横纵坐标速度一样

getVelocity=neutronVelocity.getText();

if(getVelocity!=null&&!getVelocity.equals()) {

NeutronVelocity=Integer.parseInt(getVelocity);

}

show.setNeutronVelocity(NeutronVelocity);

上下左右边界范围

getRightLocation=RightLocation.getText();

if(getRightLocation!=null&&!getRightLocation.equals()) {

NucleusRightLocation=Integer.parseInt(getRightLocation);

}

getLeftLocation=LeftLocation.getText();

if(getLeftLocation!=null&&!getLeftLocation.equals()) {

NucleusLeftLocation=Integer.parseInt(getLeftLocation);

}

getUpLocation=UpLocation.getText();

if(getUpLocation!=null&&!getUpLocation.equals()) {

NucleusUpLocation=Integer.parseInt(getUpLocation);

}

getDownLocation=DownLocation.getText();

if(getDownLocation!=null&&!getDownLocation.equals()) {

NucleusDownLocation=Integer.parseInt(getDownLocation);

}

show.setNucleusLocation(NucleusLeftLocation, NucleusRightLocation, NucleusUpLocation, NucleusDownLocation);

中子方向，0-左，1-右

getDirection=NeutronDirection.getText();

if(getDirection!=null&&!getDirection.equals()) {

neutrondirections=Integer.parseInt(getDirection);

}

show.setNeutronDirection(neutrondirections);

try {

show.showWindow();

} catch (Exception e1) {

TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

GridPane布局

GridPane grid = new GridPane();

grid.setAlignment(Pos.CENTER);

grid.setVgap(10);

grid.setHgap(10);

grid.setPadding(new Insets(5, 5, 5, 5));

将信息加入布局中

grid.add(Amount, 0, 0);

grid.add(nucleusAmount, 1, 0);

grid.add(Velocity, 0, 1);

grid.add(neutronVelocity, 1, 1);

grid.add(Location, 0, 2);

grid.add(LocationRL, 0, 3);

grid.add(RightLocation, 1, 3);

grid.add(LeftLocation, 1, 4);

grid.add(LocationUD, 0, 5);

grid.add(UpLocation, 1, 5);

grid.add(DownLocation, 1, 6);

grid.add(Direction, 0, 7);

grid.add(NeutronDirection, 1, 7);

grid.add(hbBtn, 1, 8);

Group root = (Group) scene.getRoot();

root.getChildren().add(grid);

primaryStage.setScene(scene);

primaryStage.show();

} catch (Exception e) {

e.printStackTrace();

}

}

public void showWindow() throws Exception {

start(stage);

}

public static void main(String[] args) {

launch(args);

}

}