厦門大學

信息学院软件工程系

《JAVA 程序设计》实验报告

实验 11

姓名：黄勖

学号：22920212204392

学院：信息学院

专业：软件工程

完成时间：2023.5.11

一、实验目的及要求

- 熟悉 JavaFX

二、实验题目及实现过程

实验环境：Windows 10 21H2、jdk17、javafx scene builder、utf-8 编码

题目一

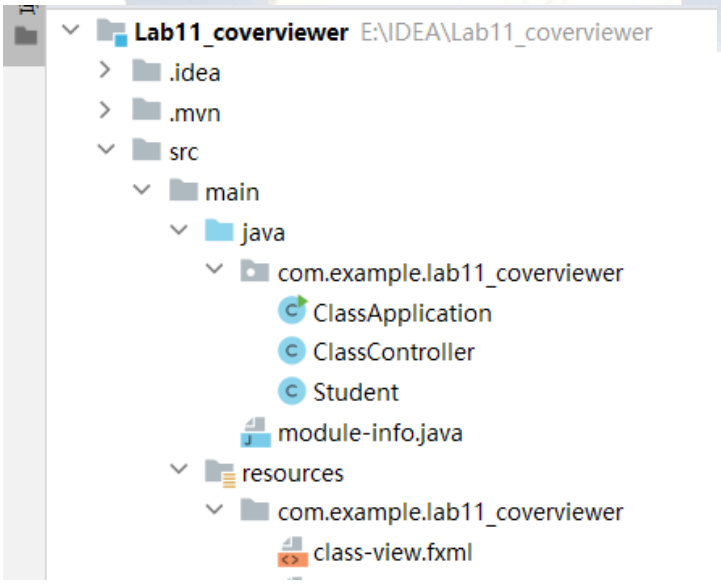
（一） 实验题目

◆ 设参照 CoverViewer 这个例子写一个程序，管理班上所有人员。其中，左侧 ListView 中显示“学号 姓名”，并按学号排序，右边显示左边所选中的学生的基本信息（照片，姓名，学号，联系电话，邮箱）。右边窗口中有四个按钮：新增、查询、上一个、下一个。
若点击“下一个”按钮，则显示下一个学生信息；
若点击“上一个”按钮，则显示上一个学生信息。
若点击“新增”按钮，则出现输入窗口（对话框），提示用户输入相关信息；

（二） 实现过程

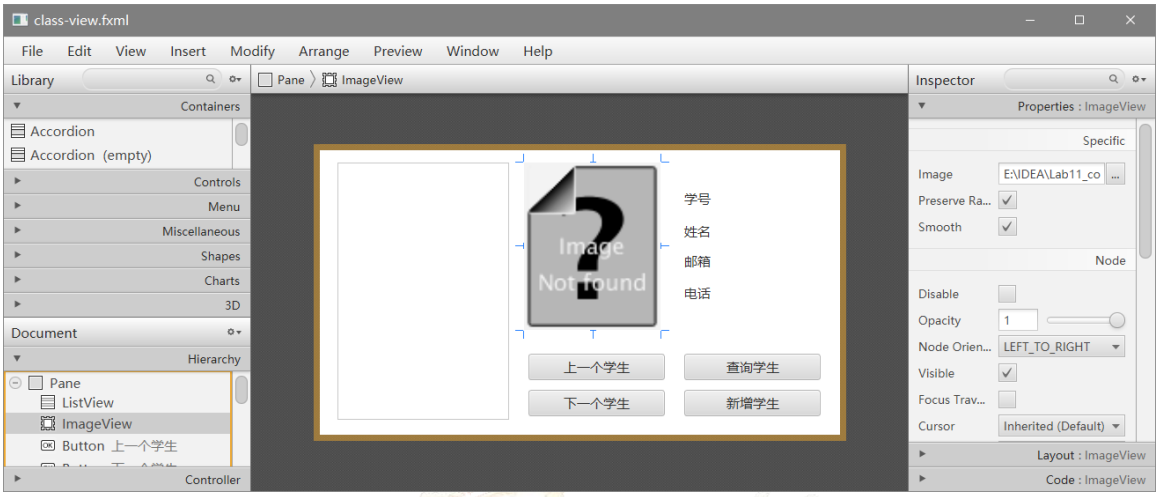
思路：

1、新建 JavaFx Project；

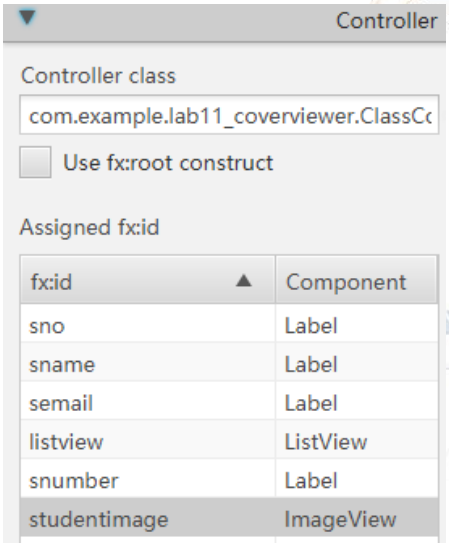


2、新建 fxml 文件，Open with SceneBuilder；

3、拖动组件，按照图片制作界面；



4、配置 controller 以及对应组件 fxid；



5、修改窗口标题以及窗口大小。

class-view.fxml | ClassController.java | ClassApplication.java | Student.java

```java
package com.example.lab11_coverviewer;

import ...

2 个用法
public class ClassApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(ClassApplication.class.getResource( name: "class-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), v: 591, v1: 323);
        stage.setTitle("班级学生管理系统");
        stage.setScene(scene);
        stage.show();
    }

    0 个用法
    public static void main(String[] args) { launch(); }
}
```

6、编写 Student 类，实现学生基本信息的存储（包括照片信息）；

class-view.fxml | ClassController.java | ClassApplication.java | Student.java

```java
package com.example.lab11_coverviewer;

10 个用法
public class Student {
    4 个用法
    private String id;
    4 个用法
    private String name;
    3 个用法
    private String image;
    3 个用法
    private String phonenumber;
    3 个用法
    private String email;
    5 个用法
    public Student(String id,String name,String image,String phonenumber,String email) {
        this.id = id;
        this.name = name;
        this.image = image;
        this.phonenumber=phonenumber;
        this.email=email;
    }
    1 个用法
    public String getName() {
        return name;
    }
```

7、（最主要）编写 Controller；

①加载 javafx 组件

```
    @FXML
    private ListView<Student> listview;
    8 个用法
    private ObservableList<Student> studentList;
    5 个用法
    private int currentChoose = 0;
    2 个用法
    @FXML
    private ImageView studentimage;
    2 个用法
    @FXML
    private Label sno;
    2 个用法
    @FXML
    private Label sname;
    2 个用法
    @FXML
    private Label snumber;
    2 个用法
    @FXML
    private Label semail;
```

②编写上下切换条目的点击事件

```
    @FXML
    public void upClick() {
        listview.getSelectionModel().select( i: listview.getSelectionModel().getSelectedIndex()-1);
    }

    1 个用法
    @FXML
    public void downClick() {
        listview.getSelectionModel().select( i: listview.getSelectionModel().getSelectedIndex()+1);
    }
```

③编写窗口初始化加载的方法，设置初始显示的学生

```
public void initialize() {
    studentList = FXCollections.observableArrayList(
            new Student( id: "2292021003", name: "谢顶",   image: "xd.jpg", phonenumber: "13367467334", email: "hyx666@ilovemaoli.com"),
            new Student( id: "2292021002",  name: "甜所浩二",  image: "haoer.PNG", phonenumber: "1145141919810", email: "cybdsb@ilovemaoli.com"),
            new Student( id: "2292021001",  name: "蔡徐坤",  image: "ikun.jpg", phonenumber: "13184626018", email: "ikun@chicken.ok"),
            new Student( id: "2292021004",  name: "桥本环奈",  image: "qbhn.jpg", phonenumber: "18725472845", email: "sodayo@qaq.co.jp")
    );
    // 考写法: studentList.sort((o1, o2) -> o1.getID().compareTo(o2.getID()));
```

④实现按学号排序

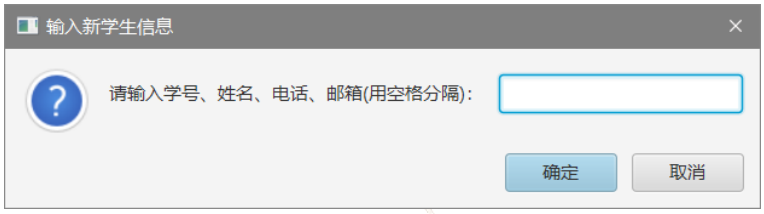
```
studentList.sort(Comparator.comparing(Student::getID));
```

⑤添加 listener

```
listview.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newValue) -> {
    if (newValue != null) {
        Image image = new Image(getClass().getResourceAsStream(newValue.getImage()));
        studentimage.setImage(image);
        sno.setText("学号: "+newValue.getID());
        sname.setText("姓名: "+newValue.getName());
        snumber.setText("电话: "+newValue.getPhonenumber());
        semail.setText("邮箱: "+newValue.getEmail());
    }
});
listview.getSelectionModel().select(currentChoose);
```

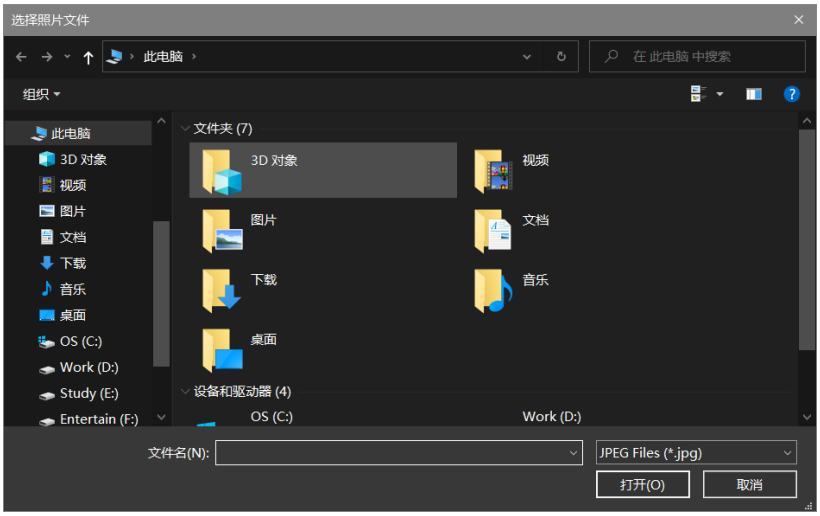
⑥编写添加学生的方法，使用对话框输入信息并添加到 listview

```
77      public void addClick() {
78          // 弹出输入对话框
79          TextInputDialog dialog = new TextInputDialog();
80          dialog.setTitle("输入新学生信息");
81          dialog.setHeaderText(null);
82          dialog.setContentText("请输入学号、姓名、电话、邮箱(用空格分隔)：");
83          Optional<String> result = dialog.showAndWait();
```



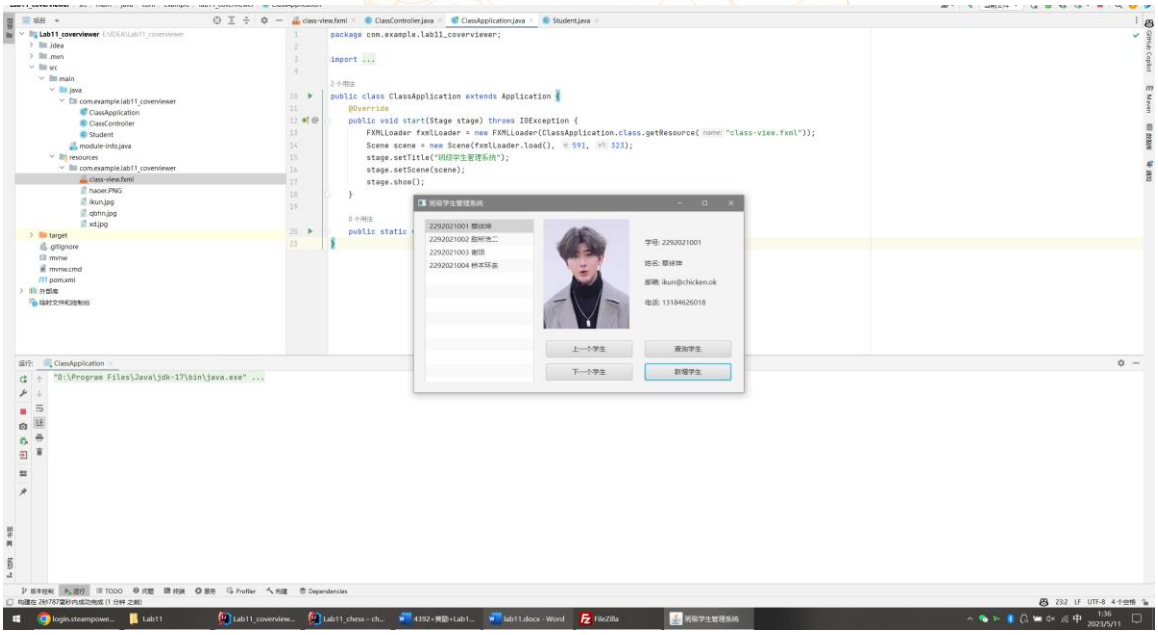
⑦使用文件选择对话框添加照片文件

```
85          result.ifPresent(input -> {
86              String[] inputs = input.split( regex: " ");
87              if(inputs.length == 4){
88                  //获取照片文件路径
89                  Alert alert1 = new Alert(Alert.AlertType.INFORMATION);
90                  alert1.setTitle("输入成功");
91                  alert1.setHeaderText("请上传学生照片");
92                  alert1.showAndWait();
93                  FileChooser fileChooser = new FileChooser();
94                  fileChooser.setTitle("选择照片文件");
95                  fileChooser.getExtensionFilters().addAll(
96                      new FileChooser.ExtensionFilter( s: "JPEG Files", ...strings: "*.jpg")
97                  );
98                  File selectedFile = fileChooser.showOpenDialog( window: null);
99                  if(selectedFile != null){ // 将照片文件复制到当前目录下
100                     try {
101                         Path source = Paths.get(selectedFile.getPath());
102                         Path fxmlPath = Paths.get(getClass().getResource( name: "class-view.fxml").toURI());
103                         Path targetDir = fxmlPath.getParent();
104                         Files.createDirectories(targetDir);
105                         String targetFileName = inputs[1] + ".jpg";
106                         Path target = targetDir.resolve(targetFileName);
107                         System.out.println(target);
108                         Files.copy(source, target, StandardCopyOption.REPLACE_EXISTING);
109                         Student newStudent = new Student(inputs[0],inputs[1],targetFileName,inputs[2],inputs[3]); // 创建新学生对象
110                         studentList.add(newStudent);
111                         studentList.sort(Comparator.comparing(Student::getID));//按学号排序
112                         currentChoose = studentList.indexOf(newStudent);
113                         listview.getSelectionModel().select(currentChoose);
114                     } catch (IOException | URISyntaxException ex) {
115                         Alert alert2 = new Alert(Alert.AlertType.ERROR);
116                         alert2.setTitle("添加失败");
117                         alert2.setHeaderText("请尝试重新添加");
118                         alert2.showAndWait();
119                         ex.printStackTrace();
120                     }
121                 }
122             }
```



（三） 过程截图

最终结果（全屏截图）



选择上\下一个，可以实现切换。

也可以实现添加学生信息，包括照片文件。

此时添加后的学号也按照升序排序。

(拓展题目)题目二

（一） 实验题目

- 在上题若点击“查询” 按钮，则弹出对话框，提示用户输入要查询的学生学号，若能找到，则显示这个学生的信息，若找不到给用户提示找不到。

（二） 实现过程

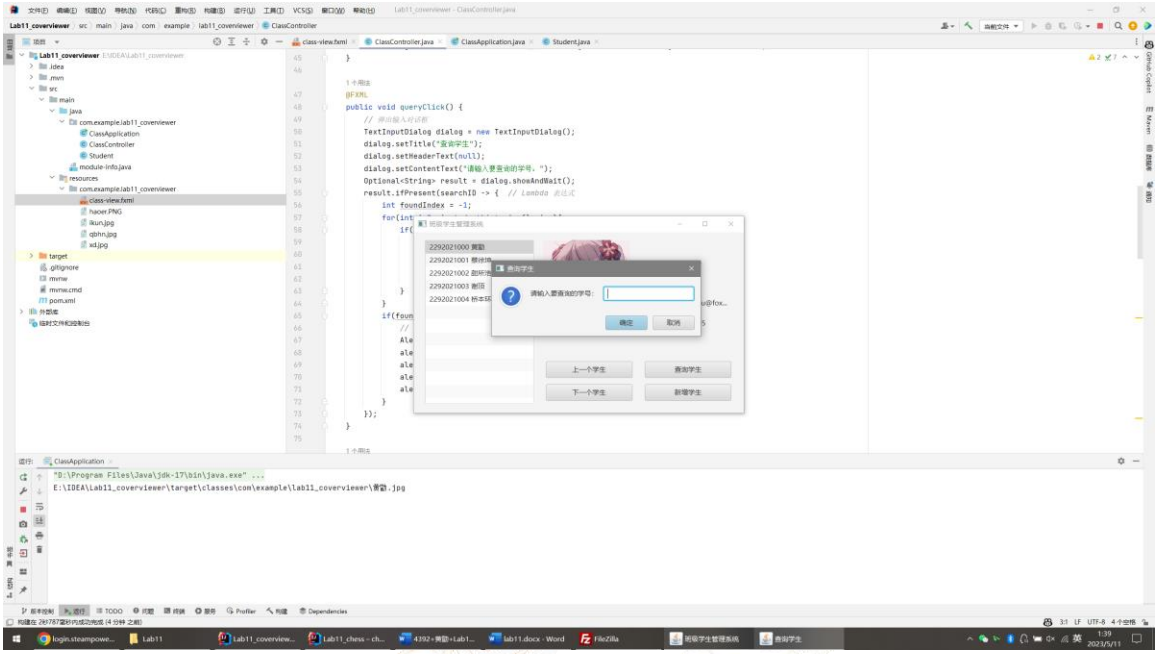
1. 添加查询按钮

2. 添加点击查询按钮的事件，弹出对话框输入学号

```
47      @FXML
48      public void queryClick() {
49          // 弹出输入对话框
50          TextInputDialog dialog = new TextInputDialog();
51          dialog.setTitle("查询学生");
52          dialog.setHeaderText(null);
53          dialog.setContentText("请输入要查询的学号：");
54          Optional<String> result = dialog.showAndWait();
55          result.ifPresent(searchID -> {  // Lambda 表达式
56              int foundIndex = -1;
57              for(int i=0; i<studentList.size(); i++){
58                  if(studentList.get(i).getID().equals(searchID)){
59                      foundIndex = i;
60                      currentChoose = foundIndex;
61                      listview.getSelectionModel().select(currentChoose);
62                      break;
63                  }
64              }
65              if(foundIndex == -1){
66                  // 弹出提示对话框
67                  Alert alert = new Alert(Alert.AlertType.ERROR);
68                  alert.setTitle("没有找到学生信息");
69                  alert.setHeaderText(null);
70                  alert.setContentText("没有找到匹配的学生信息！");
71                  alert.showAndWait();
72              }
73          });
74      }
```

（三） 过程截图

最终结果（全屏截图）



输入学号即可自动切换到该学生：

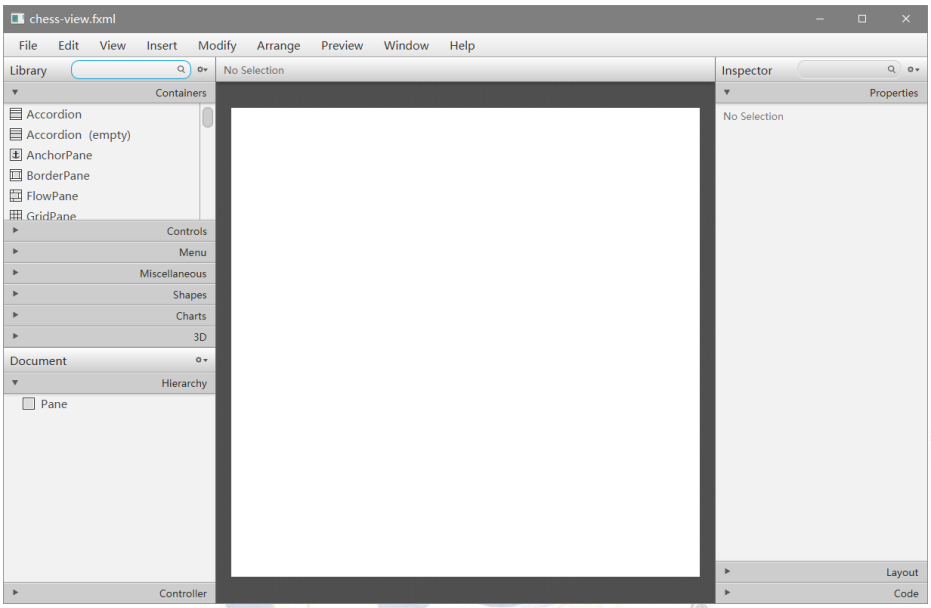
（拓展题目）题目三

（一） 实验题目

- 做一个五子棋程序，实现基本界面，当用户单击棋盘上的适当位置时实现黑白子的交替放置。

（二） 实现过程

思路：1）Scene builder 完成界面设计，设置 fxid、配置点击 Pane 事件



2）设置窗口标题

```
2 个用法
public class chessApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(chessApplication.class.getResource( name: "chess-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load());
        stage.setTitle("五子棋游戏");
        stage.setScene(scene);
        stage.show();
    }
```

3）配置控件

```
24 个用法
12      private int[][] point = new int[15][15];
        4 个用法
13      private boolean isBlack = true;
        7 个用法
14      @FXML
15      private Pane board;
```

4）绘画棋盘，利用 board.getChildren().add()循环划线，更为方便

```
153 public void initialize() {
154     //绘画棋盘
155     for (int i = 0; i < 15; i++) {
156         board.getChildren().add(new Line( v: 30 + 40 * i,  v1: 30,  v2: 30 + 40 * i,  v3: 590));
157         board.getChildren().add(new Line( v: 30,  v1: 30 + 40 * i,  v2: 590,  v3: 30 + 40 * i));
158     }
159 }
```

5）在 controller 里编写对应的脚本。

①重新开始方法，用于游戏结束后重置棋盘

```
16  void restart() {
17      board.getChildren().clear();
18      isBlack = true;
19      for (int i = 0; i < 15; i++) {
20          board.getChildren().add(new Line( v: 30 + 40 * i,  v1: 30,  v2: 30 + 40 * i,  v3: 590));
21          board.getChildren().add(new Line( v: 30,  v1: 30 + 40 * i,  v2: 590,  v3: 30 + 40 * i));
22      }
23      for (int i = 0; i < 15; i++) {
24          point[i] = new int[15];
25      }
26  }
```

②addchess 方法，用于轮流放置棋子

```
28 @    void addChess(MouseEvent mouseEvent) {
29          double x=mouseEvent.getX();
30          double y=mouseEvent.getY();
31
32          if (x < 30 || x > 590 || y < 30 || y > 590) return;
33
34          //计算落子位置
35          int row = (int) ((y - 30) / 40);
36          int col = (int) ((x - 30) / 40);
37
38          //判断是否已经有棋子
39          if (point[row][col] != 0) return;
40
41          //放置棋子
42          Circle piece = new Circle( v: 30 + col * 40,  v1: 30 + row * 40,  v2: 18);
43          if (isBlack) {
44              piece.setFill(Color.BLACK);
45              point[row][col] = 1;
46          } else {
47              piece.setFill(Color.LIGHTGRAY);
48              point[row][col] = 2;
49          }
50          board.getChildren().add(piece);
```

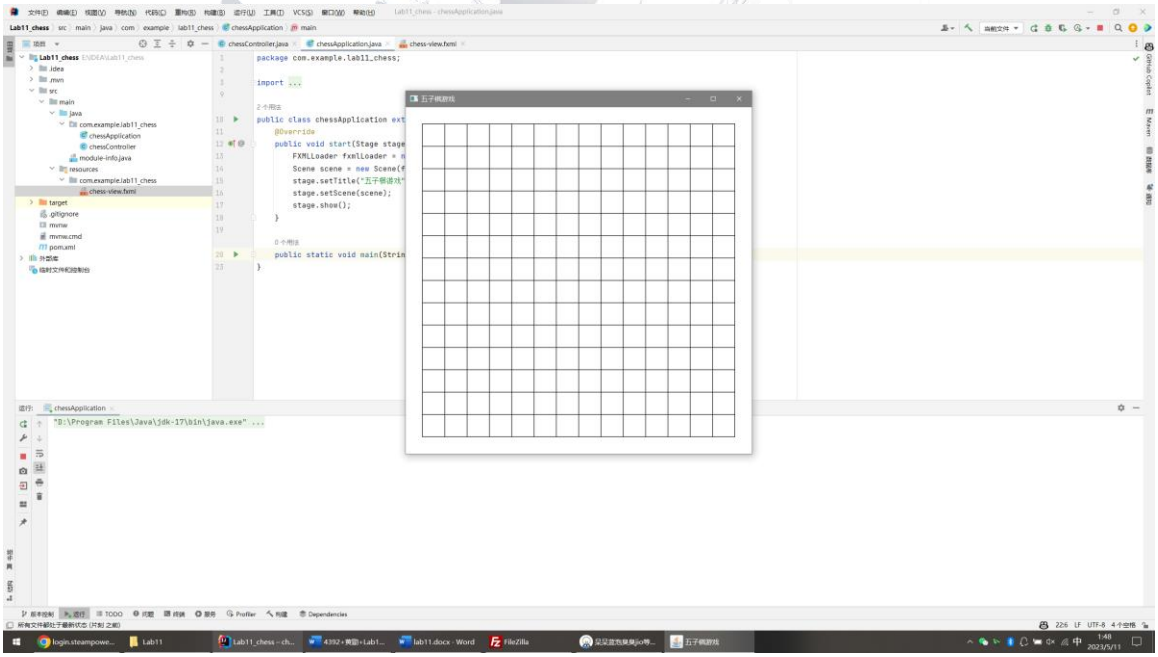
③判断输赢，循环判断棋盘落子

```
55          //判断输赢
56          int winner = checkWinner(row, col);
57          if (winner != 0) {
58              if (winner == 1) {
59                  Alert alert = new Alert(Alert.AlertType.INFORMATION);
60                  alert.setTitle("游戏结束");
61                  alert.setHeaderText(null);
62                  alert.setContentText("黑棋胜利");
63                  alert.showAndWait();
64              } else {
65                  Alert alert = new Alert(Alert.AlertType.INFORMATION);
66                  alert.setTitle("游戏结束");
67                  alert.setHeaderText(null);
68                  alert.setContentText("白棋胜利");
69                  alert.showAndWait();
70              }
71              restart();
```

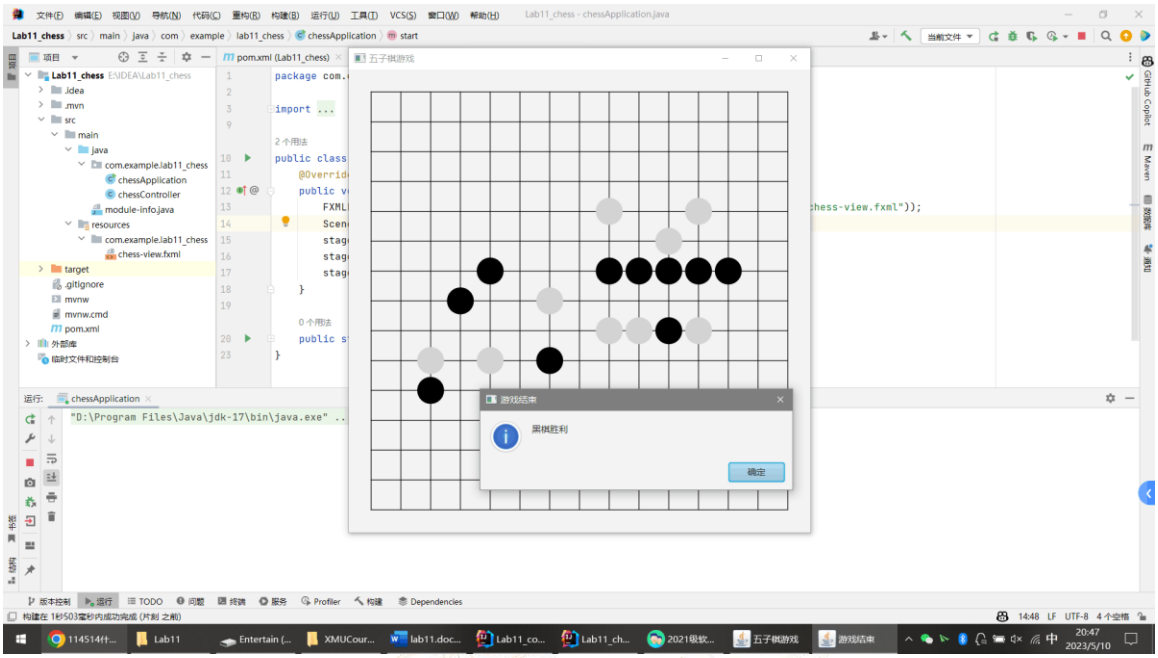
④从横向纵向斜对角轮流循环判断

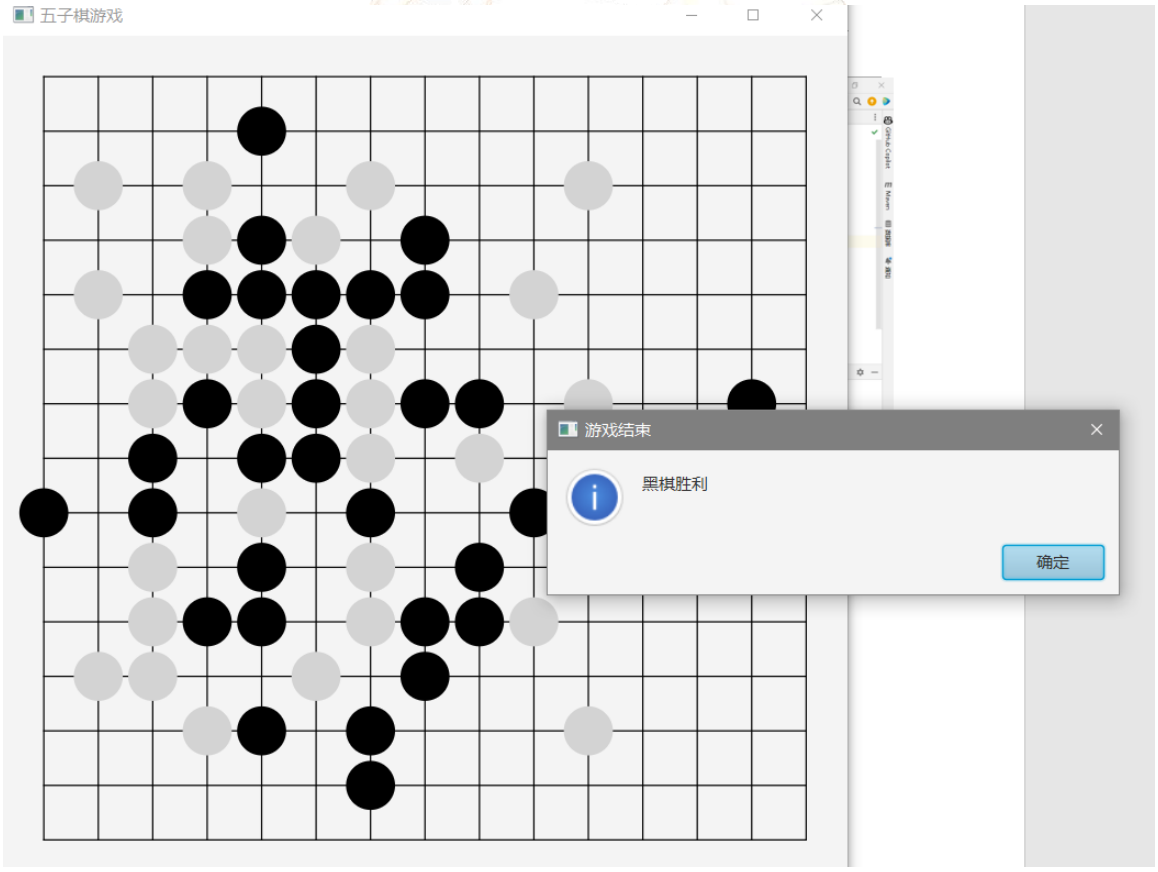
```java
public int checkWinner(int row,int col) {
    //横向
    int count = 1;
    for (int i = 1; i < 5; i++) {
        if (col - i >= 0 && point[row][col - i] == point[row][col]) {
            count++;
        } else {
            break;
        }
    }
    for (int i = 1; i < 5; i++) {
        if (col + i < 15 && point[row][col + i] == point[row][col]) {
            count++;
        } else {
            break;
        }
    }
    if (count >= 5) {
        return point[row][col];
    }
    //纵向
    count = 1;
```

（三） 过程截图



各种情况测试，均符合要求：


五子棋游戏
游戏结束
黑棋胜利
确定


五子棋游戏
游戏结束
黑棋胜利
确定

三、实验总结与心得记录

通过本次实验，我对 Java 使用 Scene builder 的使用有了更深入的了解。我学会了如何构建一个窗口等操作。在实验的过程中我为有功能的组件设置名字、编写了对应的方法，实现了需要的功能，这对我来说收获颇丰。