Lab13

TA:

- 14302010005@fudan.edu.cn 姜卓立
- 14302010040@fudan.edu.cn 武多才
- 14302010042@fudan.edu.cn 何培剑

1 摘要

1. 学习并掌握JavaFX界面制作与面板布局
2. 学习FXML的使用
3. 学习scene builder的使用
4. 学习使用Git和GitHub管理代码
5. 使用JavaFX实现Project图形界面

2 使用JavaFX实现迷宫的图形界面

2.1 绘制迷宫

2.1.1使用ImageView和GridPane绘制

ImageView是JavaFX中Node的子类，并能够加载图片并显示出来，使用ImageView绘制迷宫最好配合GridPane来指定格子的位置，实例代码如下：

```
private int[][] mapIndex = {
            { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
            { 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },
             ...
            { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, };
public void start(Stage primaryStage) {
        GridPane root = new GridPane();
        Scene scene = new Scene(root, 940, 705);
        for (int i = 0; i < mapIndex.length; i++) {
            for (int j = 0; j < mapIndex[0].length; j++) {
              Image img = null;
                if (mapIndex[i][j] == 0) {
                   img = new Image(getClass().getResourceAsStream("space.png"));
                } else {
                   img = new Image(getClass().getResourceAsStream("wall.png"));
                }
                   ImageView imv = new ImageView();
                   imv.setImage(img);
                   root.add(imv, j, i);
            }
        }
        primaryStage.setScene(scene);
        root.requestFocus();
        primaryStage.show();
}
```

每一个迷宫格子都用ImageView来表示，并用GridPane的add方法放到pane的指定位置。

2.1.2 使用Canvas绘制迷宫

使用Canvas绘制迷宫，迷宫的绘制主要用到GraphicsContext.drawImage()方法，GraphicsContext是JavaFX的绘制器，示例如下：

```
public class MyCanvas extends Canvas {
    // 游戏地图
    private GraphicsContext gContext;
    ...
    public MyCanvas(double width, double height) {
        super(width, height);
        ...
        gContext = getGraphicsContext2D();
    }

    public void draw() {
        int mapWidth = mapIndex[0].length;
        int mapHeight = mapIndex.length;
        for (int y = 0; y < mapHeight; y++) {
            for (int x = 0; x < mapWidth; x++) {
                if(mapIndex[y][x]==0){
                    gContext.drawImage(spaceImg, 0, 0, tileWidth, tileHeight, x * tileWidth, y
                            * tileHeight, tileWidth, tileHeight);
                }else {
                    gContext.drawImage(wallImg, 0, 0, tileWidth, tileHeight, x * tileWidth, y
                            * tileHeight, tileWidth, tileHeight);
                }
            }
        }
    }

}
```

通过继承Canvas类，实现自己的Canvas，并在此类中定义GraphicesContext变量，即可绘制图片。

注意canvas不能直接放入场景Scene中，而需先将canvas放入某个容器如Panel中，在将Pane放入Scene，才可正常显示。

与ImageView相比，Canvas绘制更加随意，可在任意位置绘制图形，可自行用Canvas尝试任务行走的动画。

2.2 通过Event让人物动起来

为需要处理的事件编写相应的事件处理程序被称为事件驱动，事件驱动是很多程序（尤其是游戏）中常用的程序运行以及编程方式。例如：控制人物移动可通过键盘驱动，当键盘按下'W'时，触发键盘事件，人物向上移动。JavaFX提供了使用监听器为事件驱动提供了编程模式。

通过控件如panel自带的setOnKeyPressed便可监听键盘按下事件，代码示例如下：

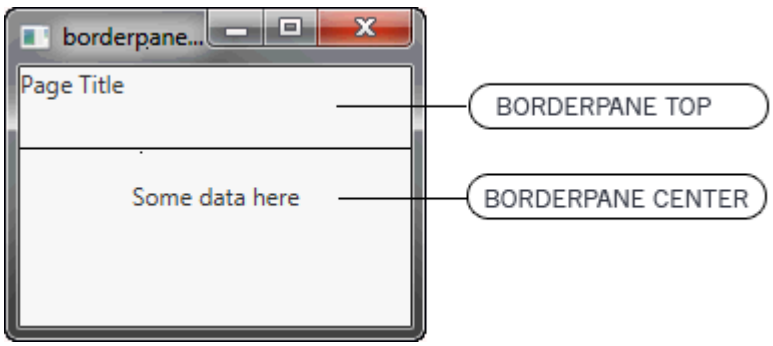
```
        panel.setOnKeyPressed(new EventHandler<KeyEvent>() {
            public void handle(KeyEvent event) {
                switch (e.getCode()) {
                    case W:
                        moveUp();
                        break;
                    case A:
                        moveLeft();
                        break;
                    case S:
                        moveDown();
                        break;
                    case D:
                        moveRight;
                        break;
                    case B:
                        back();
                        break;
                    case K:
                        killMonster();
                        return;
                }
            }

        });
```

通过调用panel的setOnKeyPressed方法为该panel设置键盘事件监听器，KeyEvent.getCode方法获得键盘输入，当键盘输入不同按键时，执行不同操作。

3 FXML

FXML是一种基于XML的语言，提供了将构建用户界面和应用程序逻辑代码分离的结构。这种对表现和应用逻辑的分离对Web开发者来说非常有吸引力，因为他们可以使用Java组件来构建一个用户界面但不需要掌握那些获取和填充数据的代码，FXML是一种很典型的解耦结构。

例如要实现一个简单的例子



Java代码：

```
public void start(Stage stage) throws Exception {
    BorderPane border = new BorderPane();
    Label toppanetext = new Label("Page Title");
    border.setTop(toppanetext);
    Label centerpanetext = new Label ("Some data here");
    border.setCenter(centerpanetext);
    Scene scene = new Scene(border, 300, 275);
    stage.setScene(scene);
    stage.show();
}
```

FXML代码：

首先需要在start方法中配置：

```
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("fxmlSample.fxml"));
        Scene scene = new Scene(root, 300, 275);
        stage.setTitle("FXML Welcome");
        stage.setScene(scene);
        stage.show();
    }
```

然后在fxml中编写：

```
<BorderPane fx:controller="sample.FxmlSample" xmlns:fx="http://javafx.com/fxml">
  <top>
      <Label text="Page Title"/>
  </top>
  <center>
      <Label text="Some data here"/>
  </center>
</BorderPane>
```

使用FXML的一个好处是场景图的结构更为清晰明了，这使得开发组能更为方便地创建和维护一个可测试的用户界面，而使用java代码则很难直观的看出组件与组件之间的层次关系。

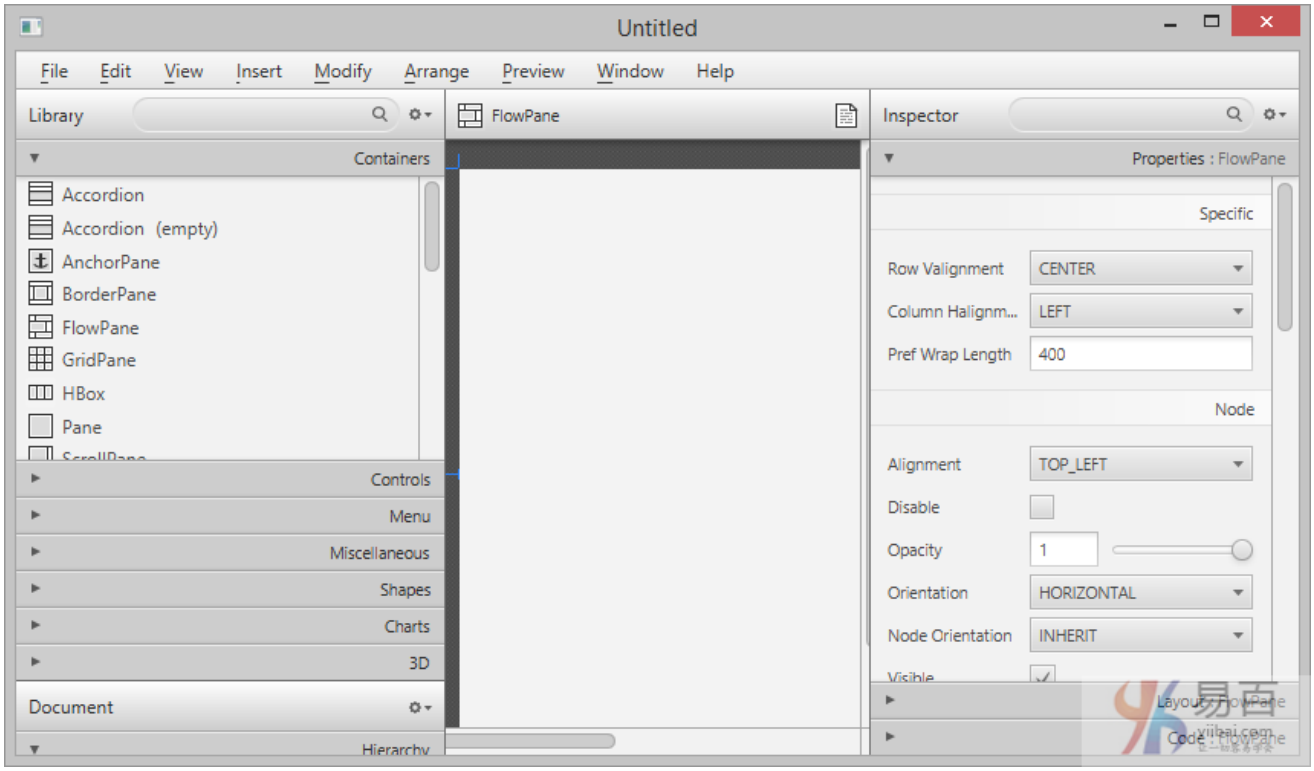
更多关于FXML编写，可参考官方文档或利用百度、Google搜索相关教程。

4 Scene Builder

4.1 简介

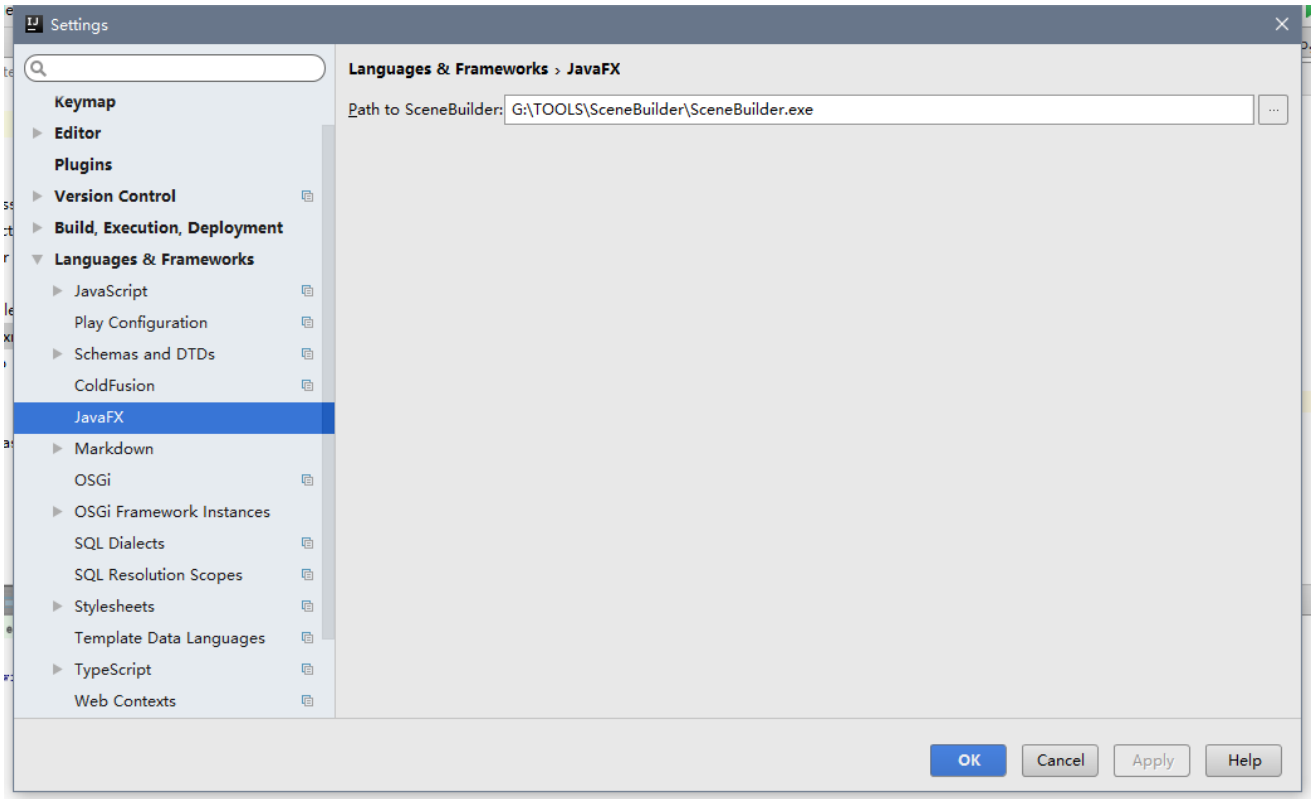
JavaFX Scene Builder是一种可视布局工具，允许用户快速设计JavaFX应用程序用户界面，而无需编码。用户可以将UI组件拖放到工作区，修改其属性，应用样式表，并且它们正在创建的布局的FXML代码将在后台自动生成。它的结果是一个FXML文件，然后可以通过绑定到应用程序的逻辑与Java项目组合。

JavaFX Scene Builder的开发界面如下：

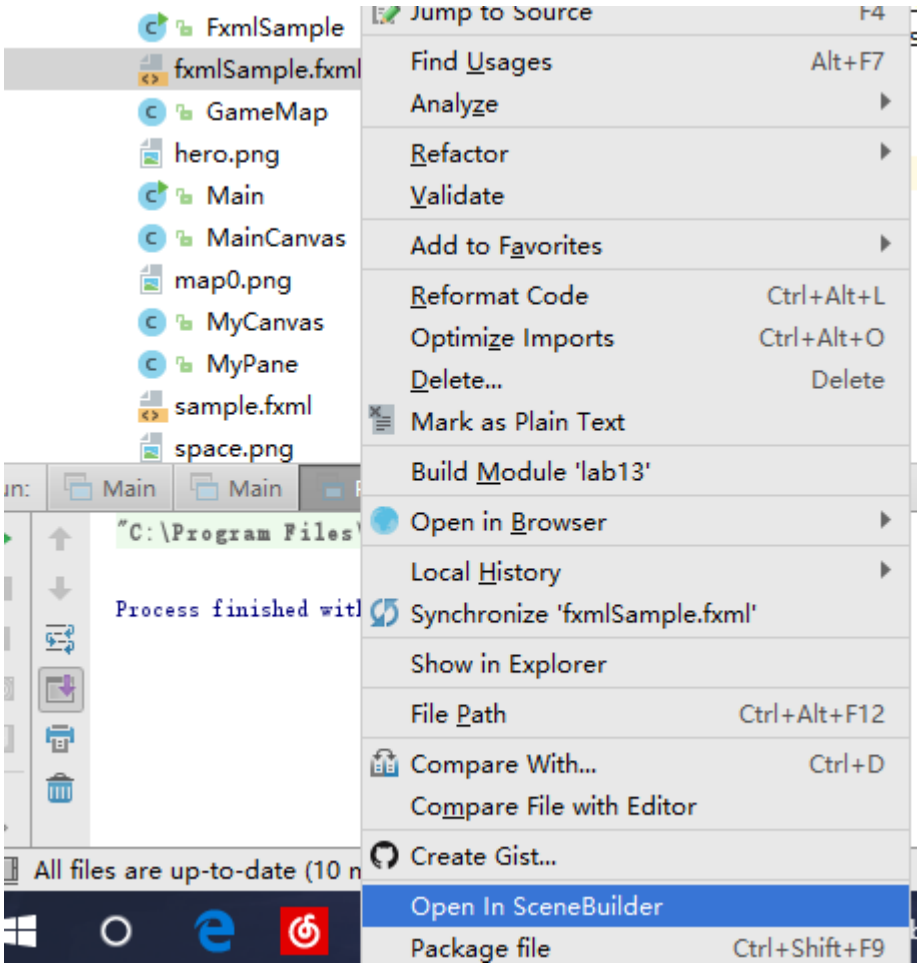


4.2 在IntelliJ中使用Scene Builder

首先将路径配置到IntelliJ中，



然后用IntelliJ打开Scene Builder



Scene非常适合用于做静态的界面布局，并且无需编写代码，但是对于动态界面，Scene Builder就显得不那么方便，特别是设计到逻辑相关，Scene Builder无法处理，因此对于迷宫Project，建议采用JavaFX代码的方式，更多关于Scene Builder请参考官方文档

5 Git与GitHub

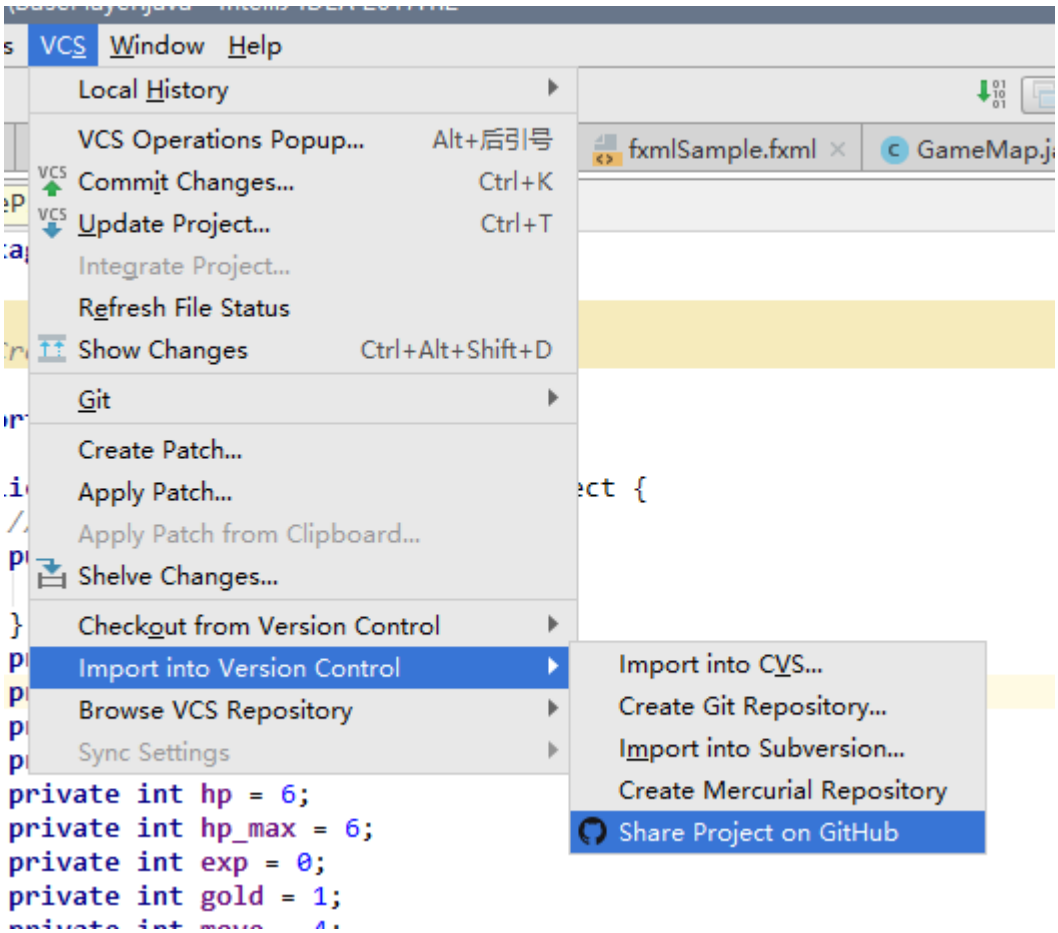
Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。

GitHub则是运用了Git技术的一个面向开源及私有软件项目的托管平台，是最广为人知的代码管理平台。

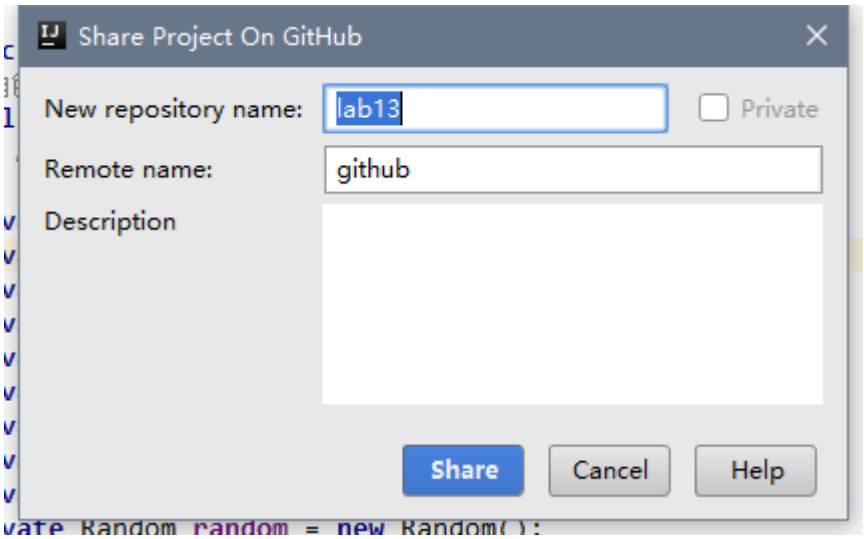
使用IntelliJ的GitHub工具

IntelliJ IDEA自带了GitHub相关的工具，要在IntellIj内使用要到GitHub网站上注册一个自己的GitHub账号

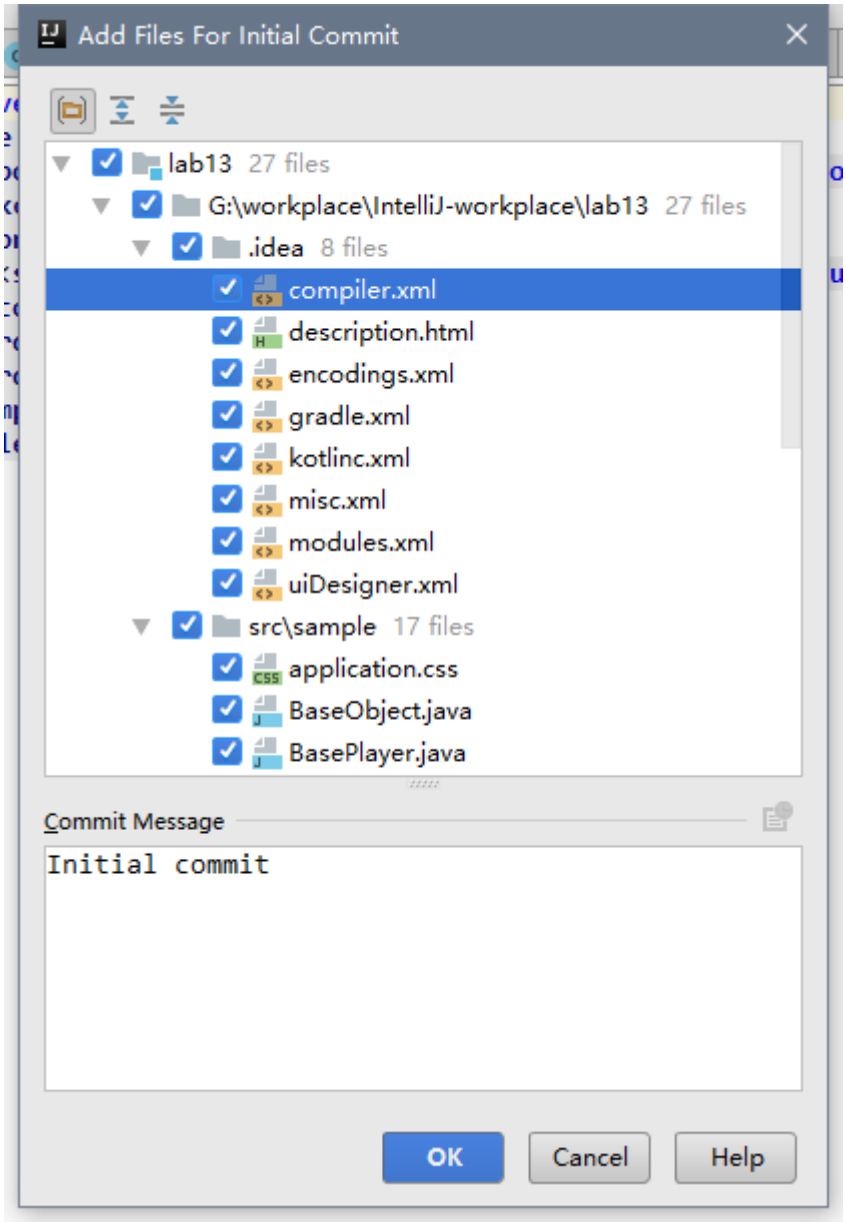
在IntelliJ里找到Share Project on GitHub



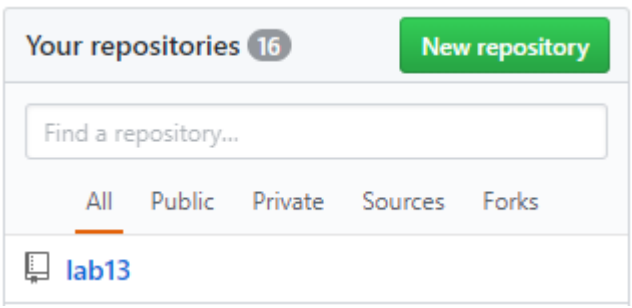
点击share：



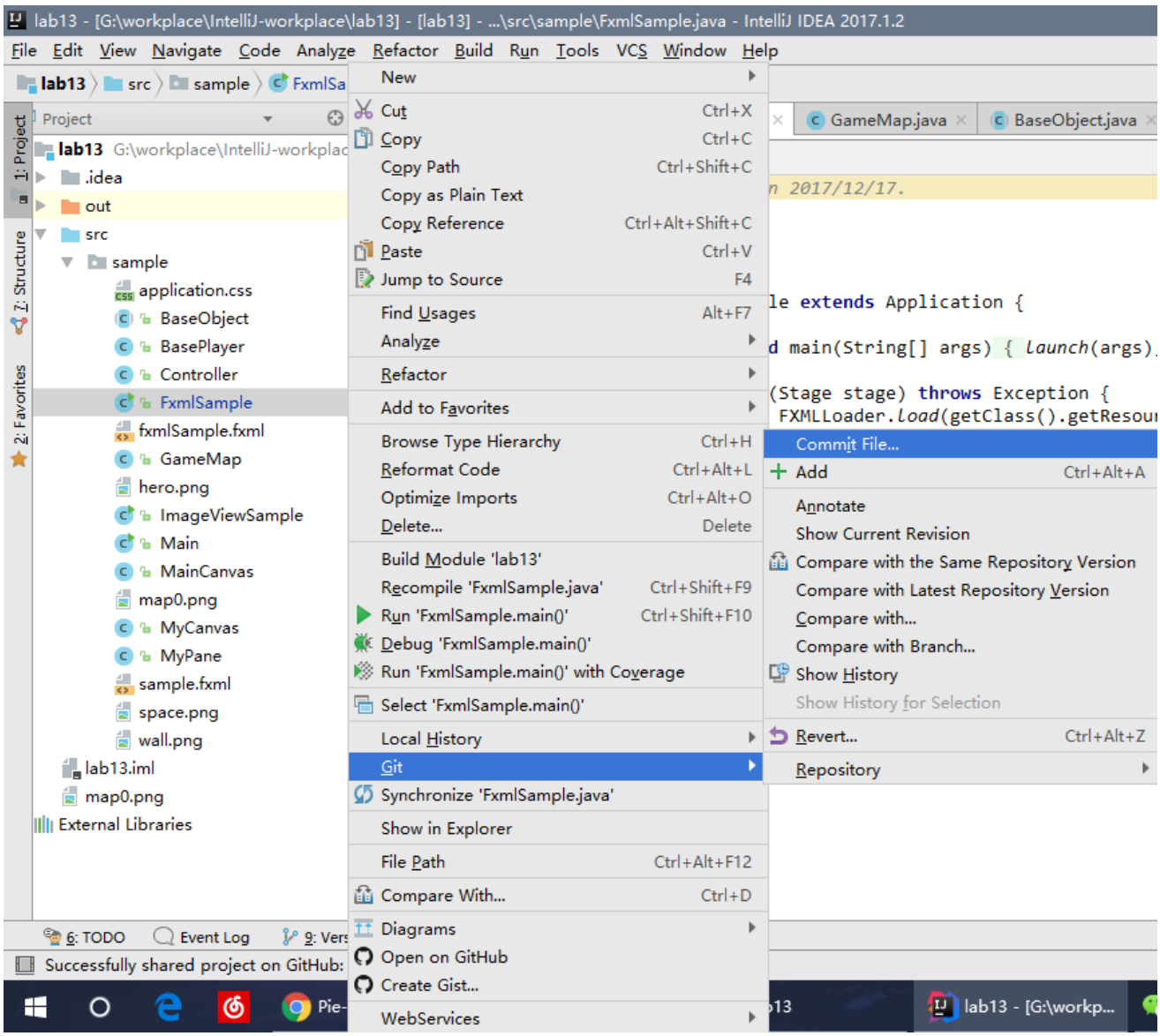
点击OK



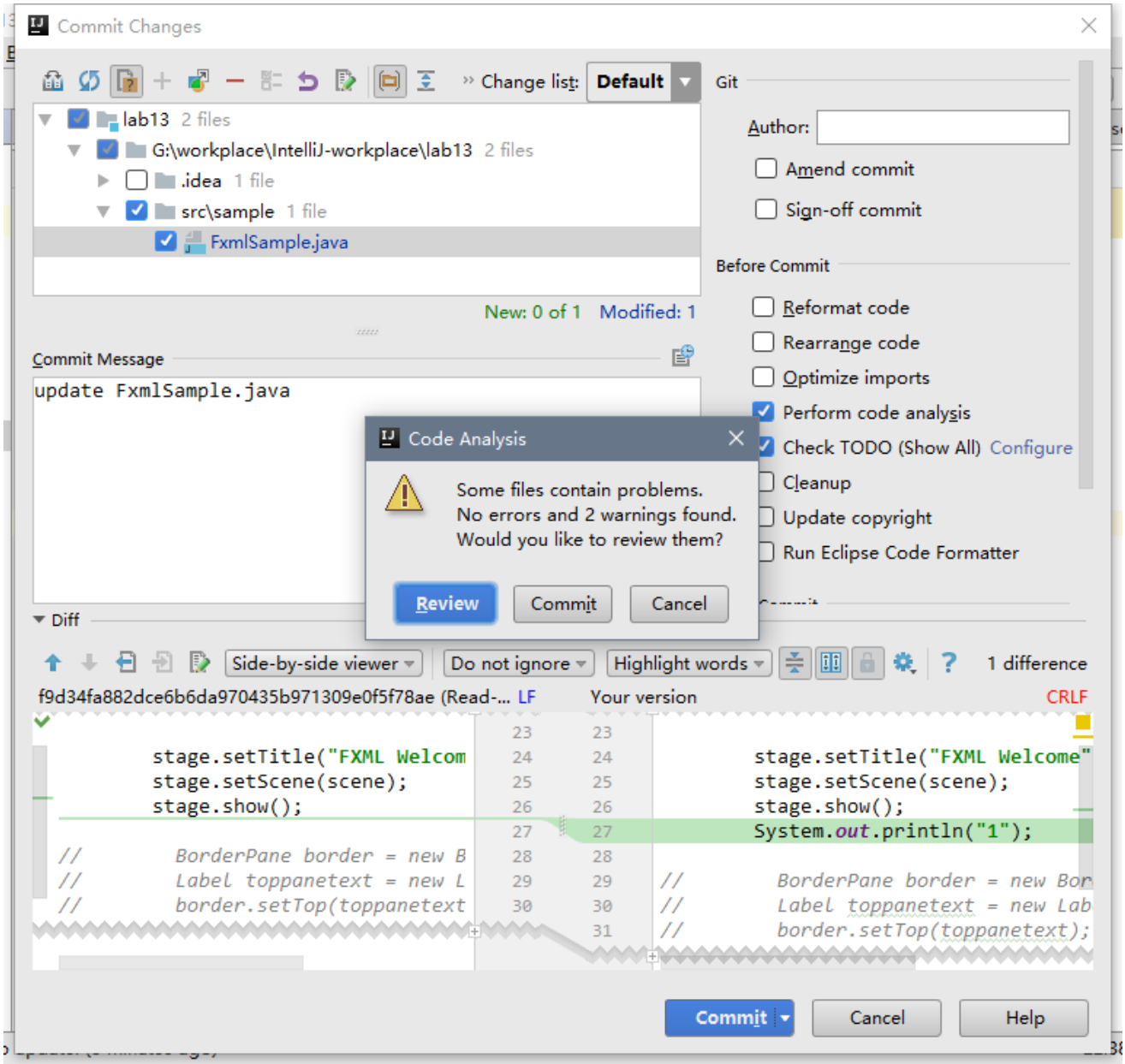
到网站上查看自己的 repositories



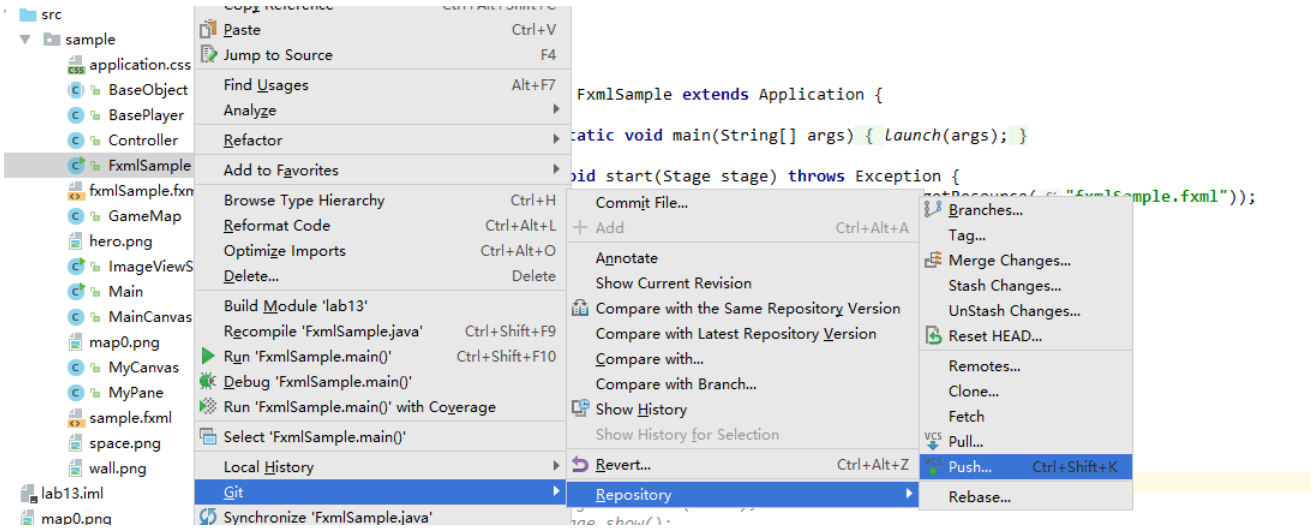
当代码修改或新增文件后，要更新到GitHub仓库，首先将代码提交到本地仓库



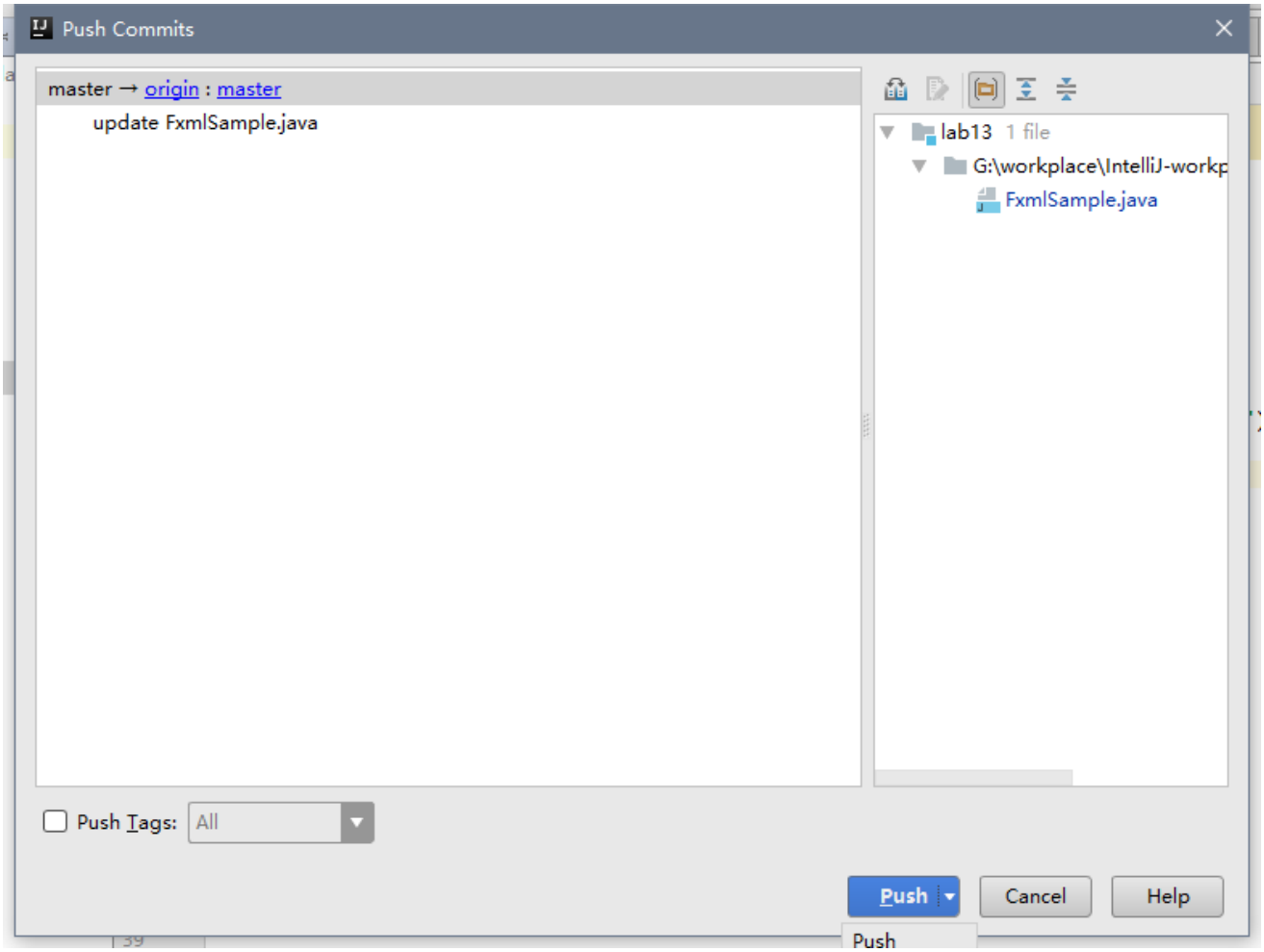
点击commit



然后在push到远程GitHub仓库



点击push



至此就将本地代码更新到了远程仓库

note:除了使用IDE自带工具来管理Git代码的方式之外，还可使用Git Bash，Git Bash是基于命令行的，通过输入git命令来实现如add、commit等操作，建议大家课外自学Git Bash的使用。

6 作业

本次Lab需要实现迷宫游戏的图形界面，并实现上下左右移动功能，只需实现一层迷宫即可。

在设计自己的Project时，注意面向对象的设计思想，如何才能更好的使Project模块化，例如将UI显示方式与迷宫逻辑作为两个模块分开。鼓励同学们多加思考，在编写前先做设计，达到一劳永逸的效果。

7 提交

1. 提交地址：**ftp://10.132.141.33/classes/17/171 程序设计A(戴开宇)/WORK_UPLOAD/lab13/**
2. 提交物：可运行程序，项目源码； 命名格式为lab13_[学号]， 例如：lab13_17302010001.zip
3. Deadline: 2017年12月24日23:59:59

8 声明

任何形式的作业都欢迎同学们相互讨论，但抄袭是严格禁止的。一旦发现抄袭行为，抄袭者和被抄袭者都以0分处理