《Java软件开发基础》实验报告

实验名称：2048游戏

姓名：吴越 学号：PB21000004 日期：2022.12.3

实验环境：CPU :Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz;内存：4.00 GB;操作系统：Windows 10;软件平台：VS code

1. 需求分析：

实验的基本功能要求如下：

1. 登录游戏。
2. 界面友好，需包括操作规则提示，当前得分，最高分等基本信息。
3. 4*4布局。
4. 使用reset操作重新开始一轮游戏。
5. 使用鼠标和键盘操作游戏界面。
6. 将所有玩过本游戏的用户的信息保存在文件中，能够使用排行榜功能显示用户排名及分数。

综合分析，可以总结出如下2个基本功能：

1. 2048游戏主体功能
2. 登录保存记录功能

其中两个基本功能又有对应的小功能，如：

1. 2048游戏主体功能包括：
(a) 2048游戏界面
(b) reset操作重新开始
(c) 操作规则提示
(d) 显示当前得分等信息
(e) 排行榜功能

2. 登录保存记录功能包括：
(a) 登录注册界面
(b) 基本的登录功能
(c) 基本的注册功能
(d) 文件记录玩家最高分

以上便是2048游戏的基本需求分析。

2. 实现类结构

根据面向对象程序设计的思想，整个2048游戏主体功能可分为3个对象：

1. 玩家用户
2. 游戏界面系统

3. 游戏规则系统

三者的关系为： 第一类对象（玩家对象）负责接受用户输入，并告知第二类对象（界面对象）游戏布局的变化，游戏界面对象接收到了游戏中数据的变化就要负责在屏幕上面显示出这种变化，同时利用第三类对象（规则系统）来对游戏进行是否结束的判定。

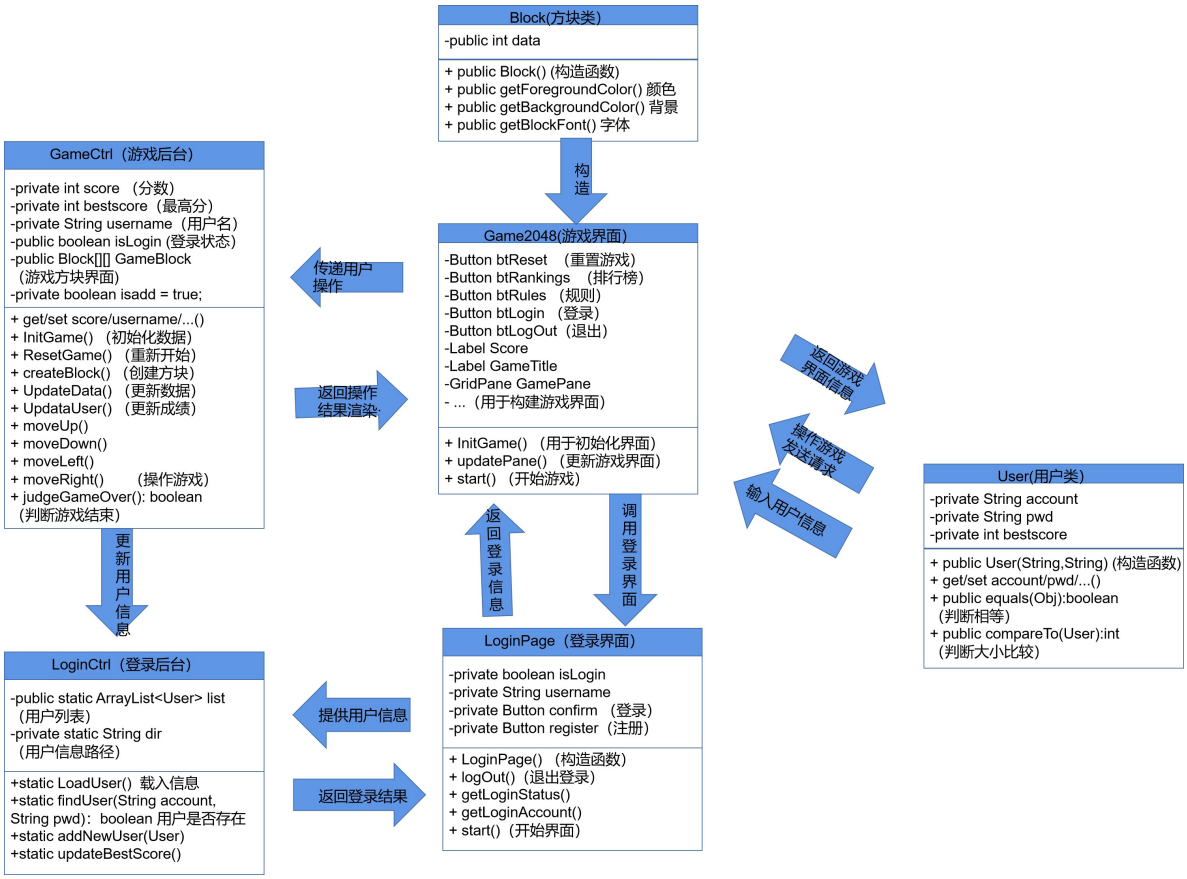
此外，对于2048游戏主体本身，又可分出组成游戏的方块对象等等。

登录保存记录功能也可以分为3个对象：

1. 玩家用户
2. 登录界面系统
3. 信息处理系统

三者的关系为： 第一类对象（玩家对象）通过在第二类对象（界面对象）输入自身信息，将相应信息发送给第三类对象（处理系统）进行处理，来对进行玩家对象本人信息的储存与读取。

综上分析可以构造出对应的类，具体的UML图如下：



根据上图，分别构造各个类即可将问题分解。

由于课堂上主要介绍了javaFX的基础使用，本游戏也使用javaFX窗口控件实现。

3. 核心代码说明：

(1)2048游戏控制滑动

```
// 操作游戏上移
public boolean moveUp() {
    for (int i = 0; i < 4; i++) {
        for (int j = 1, index = 0; j < 4; j++) {
            if (GameBlock[i][j].data > 0) {
                // 遍历各元素 该方向有相同的就合并，分数改变。是0则直接替换
                if (GameBlock[i][j].data == GameBlock[i][index].data) {
```

```
                    score += GameBlock[i][index++].data <<= 1;
                    GameBlock[i][j].data = 0;
                    isadd = true;
                } else if (GameBlock[i][index].data == 0) {
                    GameBlock[i][index].data = GameBlock[i][j].data;
                    GameBlock[i][j].data = 0;
                    isadd = true;
                } else if (GameBlock[i][++index].data == 0) {
                    GameBlock[i][index].data = GameBlock[i][j].data;
                    GameBlock[i][j].data = 0;
                    isadd = true;
                }
            }
        }
    }
    return isadd;
}
```

算法分析：

这是控制游戏界面上滑的函数。思路就是遍历每一个不为0的方块，考虑其上方的方块：若相同则合并，否则若为0则直接替代。因为在如(2,2,2,2)的情况下，操作一次应当变为(4,4,0,0)，因此这里引入了参数index来限制一次只有两个方块合并，已合并的不会再次合并。若上滑成功则将isadd设置为true，指操作后需要生成新的随机方块。否则不需生成。这样就实现了上滑的操作。其它左，右，下滑操作与之同理。

(2) 2048游戏主体核心代码

```
GamePane.setOnKeyPressed(e -> {
    // 按下“上”键
    if (e.getCode() == KeyCode.UP) {
        updatePane();
        Ctrl.moveUp();
        Ctrl.createBlock();
        Ctrl.UpdateData();
    }
    // ...
});

private void updatePane() {
    Score.setText("" + Ctrl.getScore());
    if (login.getLoginStatus() == true)
        BestScore.setText("" + Ctrl.getBestScore());
    else
        BestScore.setText("Null");

    if (Ctrl.judgeGameOver()) {
        Alert alert = new Alert(AlertType.INFORMATION);
        //...
    }
}
```

算法分析：

这是实现2048游戏的核心代码。利用setOnKeyPressed监听用户按动键盘。当用户按下指定的按键时（如操作上滑）则执行下列操作：

1. 更新游戏界面 判断游戏是否结束。

2. 游戏未结束，后台控件控制游戏上滑。
3. 继续产生新的方块
4. 更新后台的相应数据

这样就实现了2048游戏的基础功能。

(3)登录功能核心代码

```
//LoginPage.java
confirm.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent arg0) {
        // 用户密码对应，登陆成功
        if (LoginCtrl.findUser(userAccount.getText(), userPwd.getText())) {
            isLogin = true;
            username = userAccount.getText();
            Alert alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("成功提示");
            //...
            primaryStage.close();
        }
        // 密码错误
        else {
            Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("错误提示");
            //...
        }
    }
});
// LoginCtrl.java
public static boolean findUser(String account, String pwd) {
    return list.contains(new User(account, pwd));
}
```

算法分析：

这里实现登录的验证功能，当用户确认登陆时，调用了后台findUser函数，这里利用了list.contains();方法来判断用户是否存在以及密码是否正确。若返回结果为true则登录成功，将login状态设为true并设置用户名。否则弹出错误提示。这样就实现了基本的登录功能。

4.实现过程：

思路明确后，在实际的实现过程中遇到了部分难题，解决方案分别如下：

(1)排行榜功能的实现

排行榜的实现要求按照用户的最高分进行排序，但我们用户列表中储存的为用户User对象。之前打算通过自己编写函数将User对象按照最高分排序，但实际操作较为困难且复杂。通过查阅相关资料了解到可以通过重写CompareTo方法实现排序，具体如下：

```java
// 重写compareTo方法,为了实现根据分数排序。
@Override
public int compareTo(User u) {
    return u.getBestScore() - this.getBestScore();
}
//...
Collections.sort(list);
```

即通过重写CompareTo方法，并调用import java.util.Collections中的sort方法，即可实现将User对象按照最高分排序的功能。

(2) 键盘操作按钮的问题

在游戏过程中，是使用键盘操作方块界面，鼠标点击按钮来操作游戏。但是最开始遇到一个问题：在鼠标点击按钮后，方块界面对接下来的键盘操作就完全没有反应了。之前以为是代码有问题，但通过查阅相关资料了解到要通过调用requestFocus()，如：

```java
btReset.setOnMouseClicked(e -> {
    Ctrl.ResetGame();
    //...
    GamePane.requestFocus();
});
```

就可以在点击按钮后继续监听键盘操作。

此外在使用键盘操作时还会遇到一个问题：键盘操作上下左右键会将光标移动到界面的按钮上面，导致键盘仍然无法被监听，通过查阅相关资料了解到要通过调用setFocusTraversable(false);函数，如：

```java
btLogout.setOnMouseClicked(e -> {
    //...
});
btLogout.setFocusTraversable(false);
```

就能在操作键盘时不会将光标移动到按钮上。这样问题就得到了解决。

(3) 读写用户文件数据的问题

首先读取文件是总是提示无法找到文件，查阅资料得知VScode由于环境配置的问题，在读取文件时要使用绝对路径，不然无法读取文件。因此使用绝对路径，可以运行。

在储存用户信息时，我们使用的格式如下：

account pwd score
... ...

这样写入文件比较方便，但读取时如何才能读取对应位置的信息？查阅相关资料得知：可以使用BufferedReader读取行，然后用字符串.split方法将数据分隔，这样就可以分别读取不同位置的信息，如下：

```
BufferedReader reader = new BufferedReader(new FileReader(new File(dir)));
String line = null;
while ((line = reader.readLine()) != null) {
    String[] data = line.split(" ");
    User user = new User(data[0], data[1]);
    user.setBestScore(Integer.parseInt(data[2]));
    list.add(user);
}
reader.close();
```

之后的问题时如何更新最高分 即如何修改文件指定位置的数值。查阅资料了解到Accessible可以实现这一功能，但具体的使用不太了解。因此我采用依次读取各行，当读到需要修改的那行时修改数据，最后将数据全部重新写入文件的方法，如下：

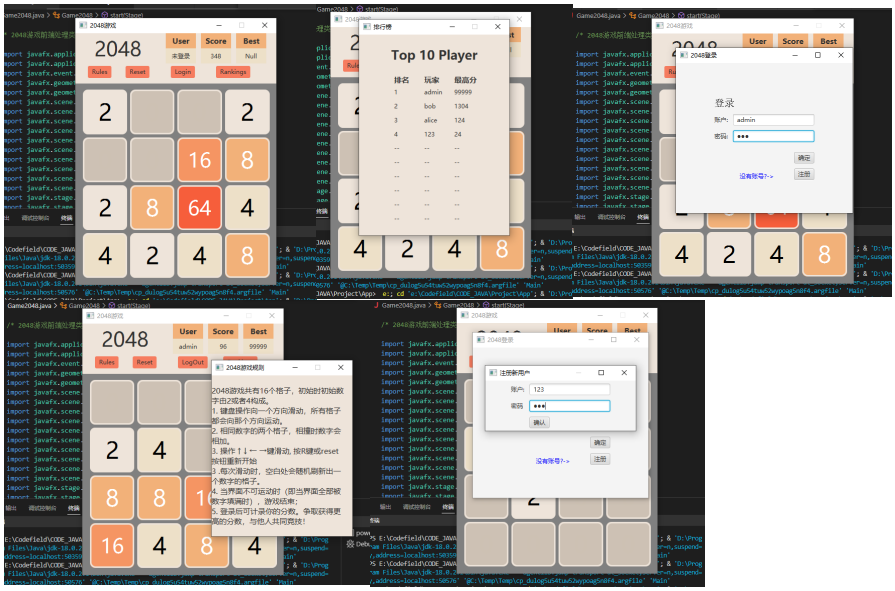
```
//读文件
BufferedReader reader = new BufferedReader(new FileReader(new File(dir)));
String line = null;
while ((line = reader.readLine()) != null) {
    String[] data = line.split(" ");
    // 依据用户名 找到对应分数并修改
    if (username.equals(data[0])) {
        bufAll.append(data[0] + " " + data[1] + " " + bestscore);
        bufAll.append(System.getProperty("line.separator"));
    }
    // 其他内容保留
    else {
        bufAll.append(line);
        bufAll.append(System.getProperty("line.separator"));
    }
}
reader.close();
// 写入文件
BufferedWriter writer = new BufferedWriter(new FileWriter(new File(dir)));
writer.write(bufAll.toString());
writer.flush();
writer.close();
```

这样就实现了对某一用户某一数据的更改，解决了这一问题。

5.实验结果

可见附件中的录屏exp.mp4

展示部分截图如下：



总结：

本次java实现2048游戏的实验是我第一次单独使用java语言编写一个相对较大的程序，因此难免存在许多缺陷与不足。在编写程序的过程中，我也遇到许多困难，许多挫折。但在努力一一克服之后，也给我带来了巨大的成就感。回想一下这次编程的收获，我觉得主要有以下几点：

1. 由于我之前没有学习过java的相关经历，平时使用java语言也比较少。因此这次实验帮助我提高了对java语言掌握的熟练度，也让我将课中所学真正转化为了实践。
2. java是面向对象设计的语言，而我之前学习的C语言是面向过程的编程语言。这次实验也开拓了我编程的思路，让我进一步掌握面向对象编程这一思想。
3. 这次实验也进一步提高了我的信息收集能力，让我在面对困难时有自主解决的能力。

总之，本次java实验让我受益匪浅。

另附：

源代码文件：./src

Game2048.java

GameCtrl.java

LoginPage.java

LoginCtrl.java

Block.java

User.java

GameFont.java

由于我使用的平台为VScode ，在不同环境下可能因为配置不同，运行有差异
读取文件时，若不使用绝对路径会找不到文件
可运行main.java作为辅助程序 并修改LoginCtrl中dir的绝对路径为当前路径。