

# 实验报告

电信学部 计试 81 李昱志  
2185011803

## 一、实验目标及功能

基于 java 实现多人在线画板功能和你画我猜小游戏功能,允许服务器端修改画板颜色,实现你画我猜功能时能让服务器端选择所画目标,并能判断客户端所给答案是否正确,正确时将答案存入本地文件 `output.txt` 文件中。

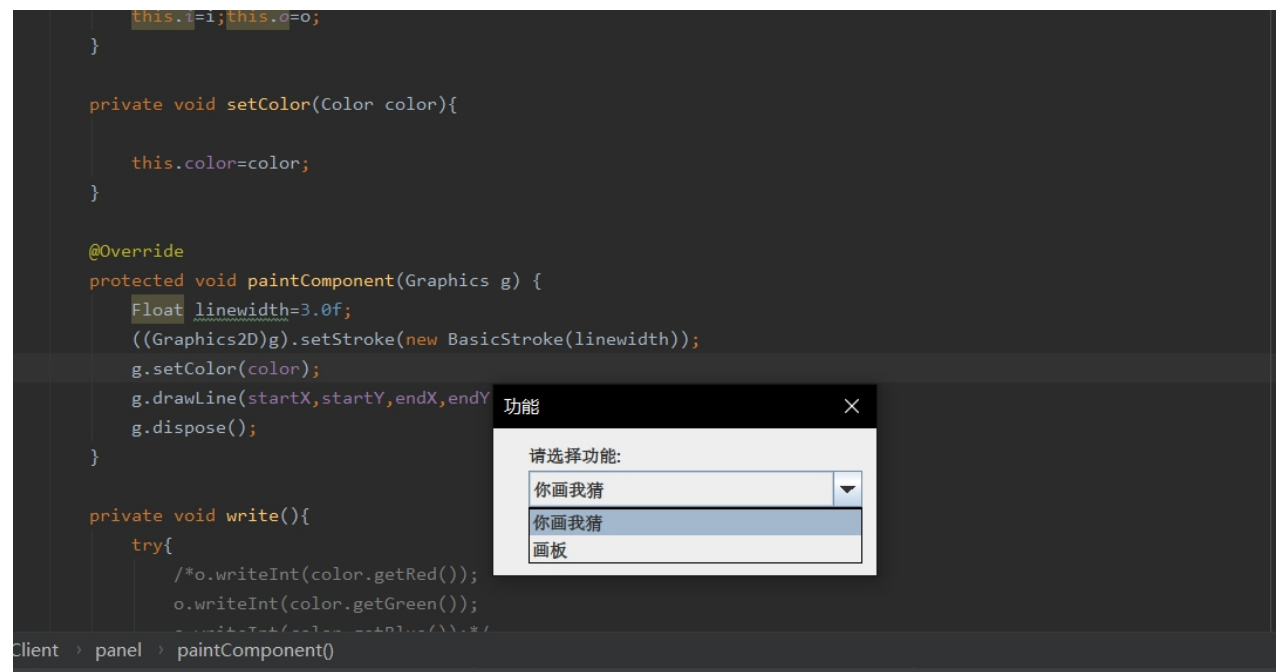
## 二、实现过程

利用 `socket` 类实现网络数据传输,用 `inputstream` 和 `outputstream` 实现用户与服务器端的交互,利用实时传递点坐标并同步绘图的方式实现双方的鼠标所画在对方屏幕上的重现,此外,为实现颜色更改,服务器端的输出增加代表颜色的数据。每一次的作画数据传输用 `writeInt` 方法进行输出,使用 `readInt` 方法读取输入并进行相应的操作,每次输出均为鼠标处于按下状态时所处的坐标位置,服务器端增加当前颜色数据,开始读取时读取坐标作为划线起始坐标,将下一组输入数据作为终点坐标,画线完成后则将终点坐标作为新起始坐标,并将下一组输入作为终点坐标,使用传递点的方式依据是点动成线,即使每两点之间所作均为直线,点数量充分大时也能实现曲线作画。在进行作画时,本机的作画与接受传入数据的复制作画互不冲突,可同时进行。在服务器端,利用多线程可实现同时同时与多个客户端交互,并以线程池进行优先级的把握和线程的调控,理论上应当能实现不同客户端之间的交互,但由于本机开发环境限制与条件制约,我只能以本机,即服务端作为一个客户端模拟交互,通过适当修改被注释掉的代码,能实现多客户端之间的交互。由于为了调试方便,我最开始仅

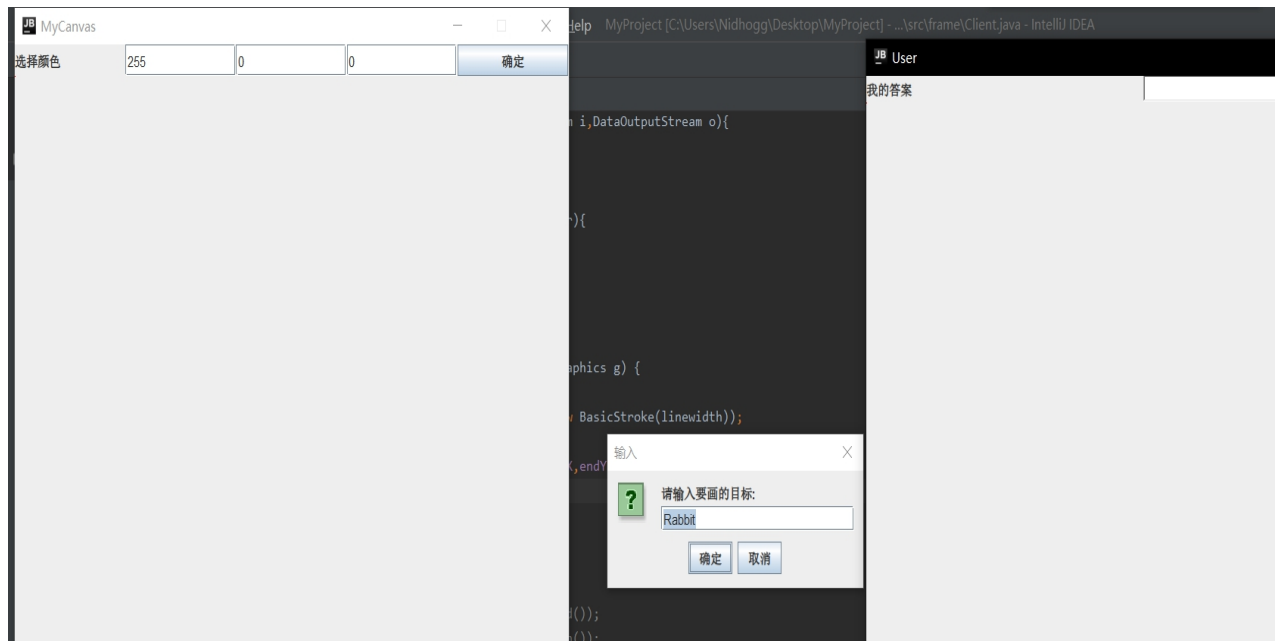
仅将服务器与单个客户端相连，因此服务器的 io 仅仅设置了单一接口，为实现服务器端与多客户端之间的信息传递，仅需设置多组 io 接口与不同的客户端相连，适当修改服务器线程的代码，然后将点阵分别输出即可，在源代码中可以看到关于这一部分的被舍弃代码（因为难以找到多个客户端连接，难以进行调试，因此被注释掉了）。

### 三、运行结果

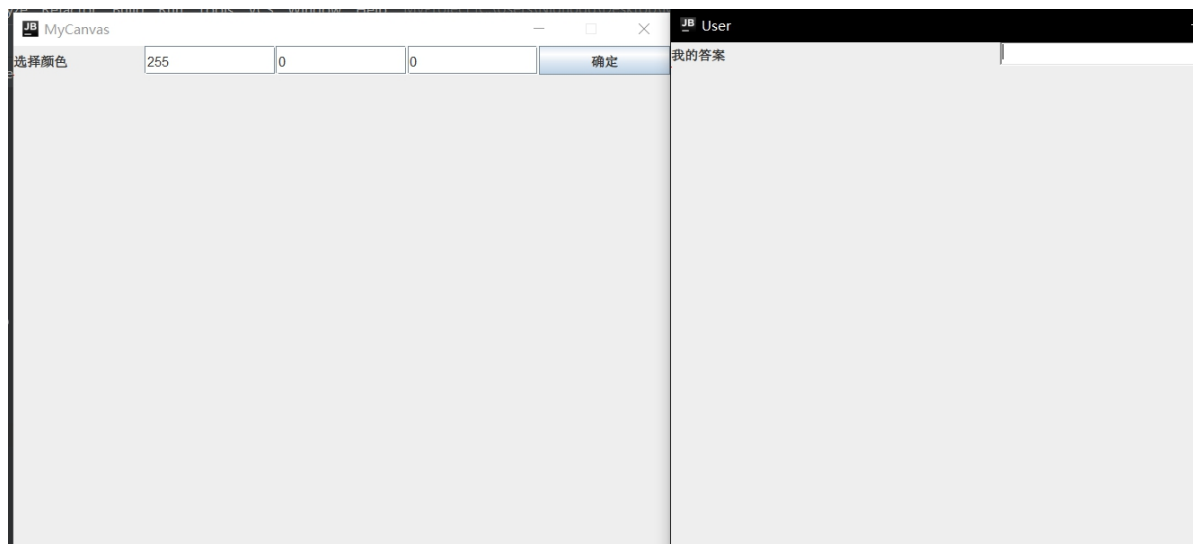
#### 1、你画我猜功能的实现：

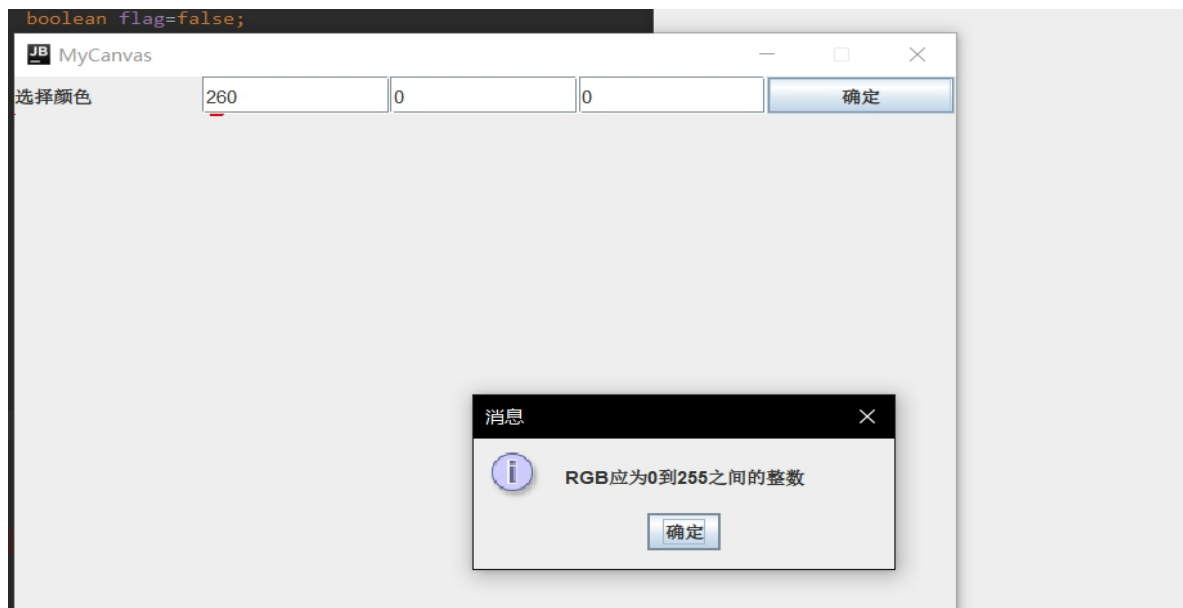


这里可选择功能，我们选择你画我猜



由服务端设置所画目标，客户端回传的答案与目标一致时才退出，并将正确答案记录在本地 **output.txt** 文件中。左为服务器端，右为客户端。此处如果点击取消、关闭或是输入空串将会断开连接，并在服务器端以 **err** 流输出“没有获取到所画目标”。

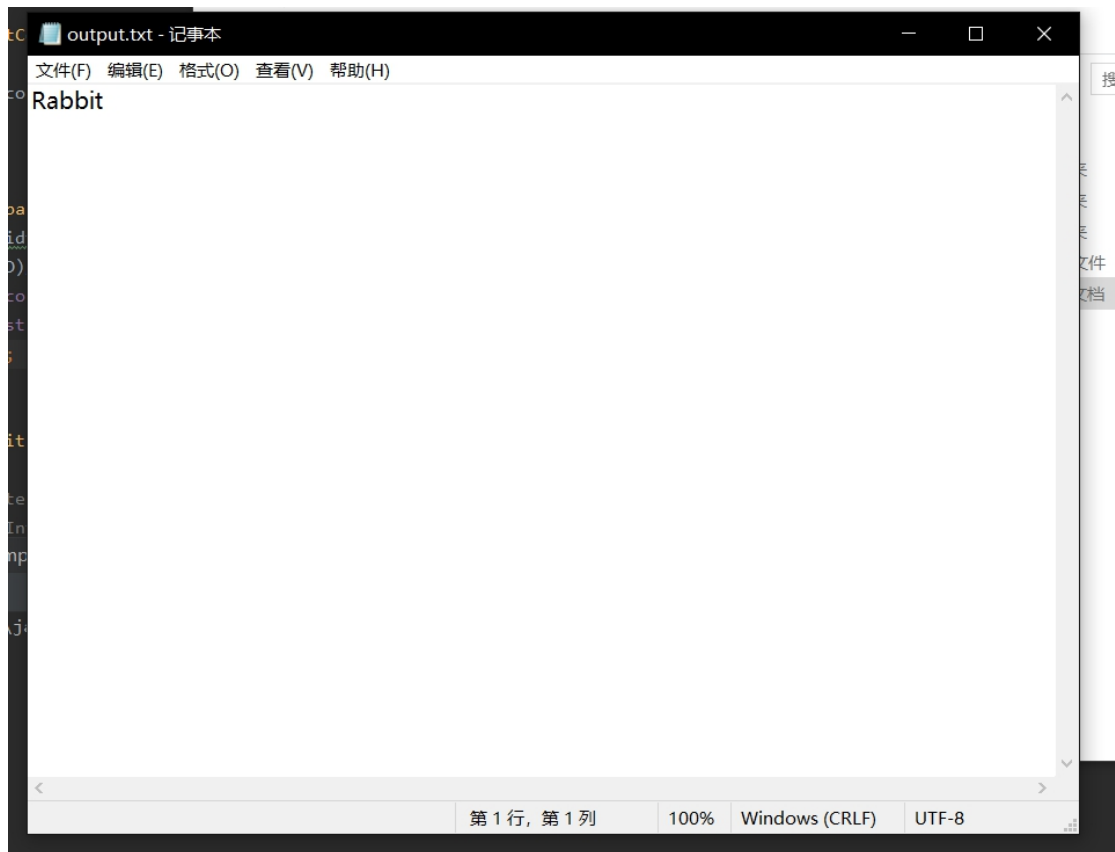




服务器端可选择颜色，在输入 RGB 数值后点击确定即可更改颜色。

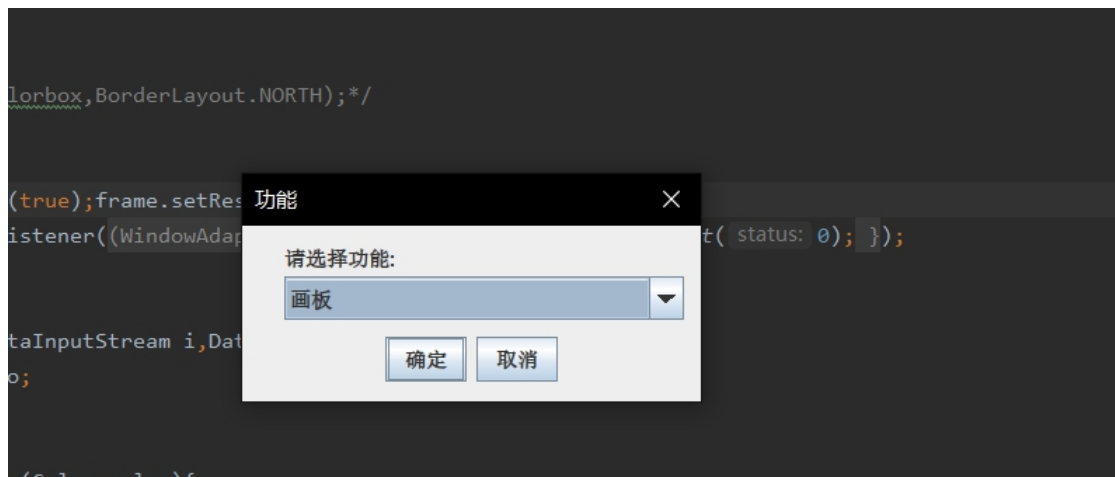
仅有服务器端可以作画，客户端除输入答案外没有任何操作响应。

这里，如果输入超出 RGB 范围的数时会抛出一个异常，出现一个错误提示对话框，增强程序健壮性。

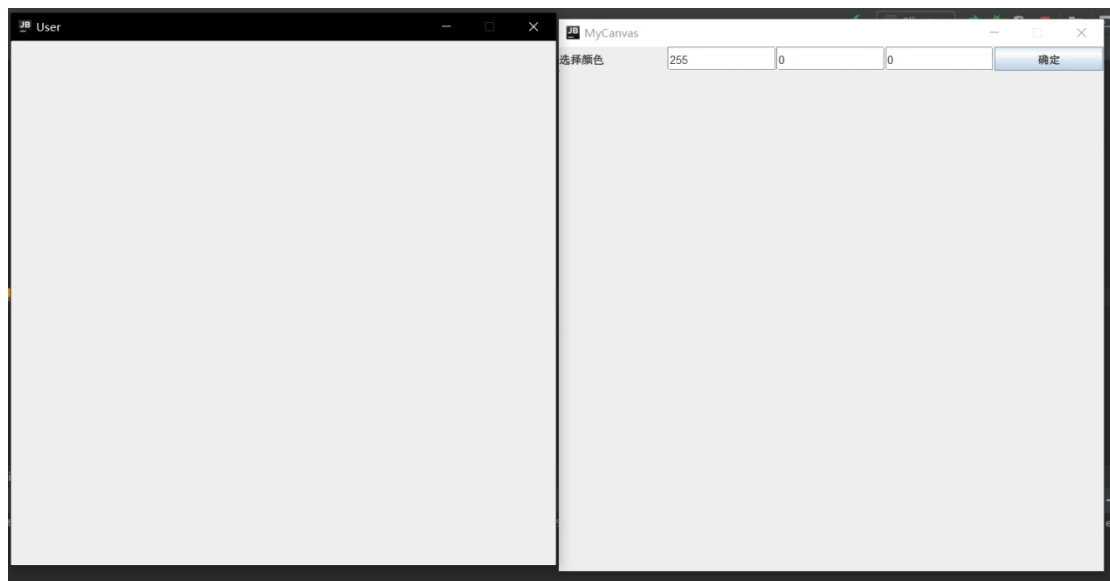


上为客户端输入 **Rabbit** 后（与目标一致）程序退出，服务器端在本地的 **output.txt** 文件中记录的正确答案。

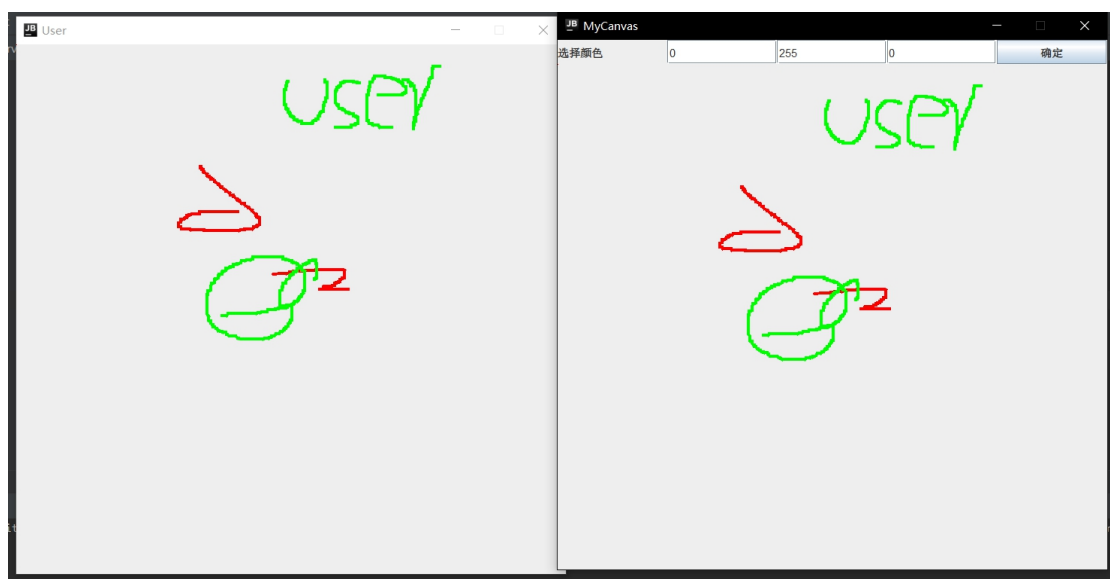
## 2、画板功能的实现：



这里我们选择画板功能。



进入界面后不再要求选择目标，客户端也没有答案文本框，但仅有服务器端能选择颜色，将代码进行适当的修改能够实现双方均可选择颜色。同样，RGB 超出范围时会出现错误提示。



在服务器做出颜色的改变后客户端的颜色也会随之改变，双方均可作画，且在本地所画会在对方的屏幕上同步显示，该功能没有固定的退出条件，用户点击关闭窗口即可断开与服务器的连接。

## 四、优化与改进

由于开发环境限制，我无法调试面对多个客户端输入请求时的程序，

因此本程序仅实现了服务器端和客户端之间的双向通信，但在注释掉的代码的基础上稍加修改调试，能在主体框架不变的情况下，实现功能集成和多用户交互，本程序的主要思想是通过传递点列的方式来实现远程共享，将一个窗口的画面以点坐标作为信息载体传递，并在另一台机器上实现复刻，传输的数据量小，结构简单，在需要同时向多个客户端传输数据的情况下能极大减轻服务器端的压力，这是本程序的优点之一，此外，用户界面有很大的改进空间，当前的颜色选择方式较为细致，但对普通用户而言（对色域要求不高）不太友好，从我注释掉的代码中可以找到利用下拉菜单实现由用户选择固定的某几种常用颜色的颜色选择方式，并可添加清屏功能，擦除功能，撤回功能等，这些附加功能在本程序的框架下实现并不困难。

## 五、心得与体会

在考虑使用何种方式实现屏幕共享时，我曾考虑过传输窗口的实时画面，这是最容易想到的方法，但是采用这种方式实现麻烦，传输数据量大，且复现画面所需要的开销大，最终在经过反复的思考和实践后，我才决定采用点阵传输的方式，这种方式充分利用了点动成线的思想，每两点之间只需调用 `drawline` 函数绘出直线，只要实时传递点阵，便能达到画出任意曲线的效果，这说明，在程序设计时，我们应深入思考问题确定算法，抓住问题的本质，这样在最后设计时代码会简化许多，在考虑问题时，往往最容易想到的仅仅是对于人类思维而言较简单的方法，用于程序设计代码实现则会相当复杂甚至难以实现，我们应当从机器的角度去思考如何“教会”计算机去解决这个问题，计算



机相对于人类的优势，即计算准确度和计算速度的优势应当充分发挥，在本程序中，若是以人类角度思考，并不会直接想到将曲线划分成点与点之间的直线连接的形式，只有从计算机的角度，从所用程序语言的角度，即利用 **java** 语言中特定类提供给程序员的方法、函数，才能有好的设计出现。编程能力的提高，不仅仅在于代码的堆砌，更在于对所用语言的熟悉，对各种结构，方法，**API** 使用的融会贯通，结合具体问题灵活分析。此外，在使用 **java** 语言编程的过程中，我更加感受到 **java** 语言鲜明的面向对象特征，以及利用 **java** 实现网络编程和多线程的方便之处，利用 **java** 自身提供的类对象及类方法，无需面对底层的各种复杂操作，便能很容易地实现高级功能，此外，**java** 的类继承和接口方式使得其相对于 **C++** 而言更加灵活，其自带的内存回收机制也使得我在编程的过程中能更专注于程序本身。

## 六、程序说明

压缩包中 **Server** 和 **mycanvas** 文件是服务器端运行所需的文件，**Client** 文件是客户端运行所需文件，如需运行，需将 **Client** 文件中初始化时的 **host** 设为服务器所在的 **ip** 地址，先运行服务器端，再运行客户端。