
更新日志

2020-11-28

环境的配置

- 今天配完了编译的环境和一些参考代码的阅读
- 配置：32位的eclipse的安装的环境的配置和写字板环境的配置
- 仓库建立

2020-11-29

使用GUI对初始界面的设计的完成，比较粗糙

- 选择菜单的方向
- 选择完毕的开始按钮
- 界面有待完善...

2020-11-30

确认笔的数据和文件的写入

- 确认笔的压力，笔的角度，笔的旋转角.....
- 能够将笔的一些信息写入创建的csv文件中
- 测试写入的结果是否正确
- 未知信息的考量

2020-12-05

界面的跟新和一些操作数据测试

- 重写界面，在选择开始按钮后显示写字面板
- 使用cello.tablet.JTabletCursor包来读出笔的各个属性值，目前以笔的压力为测试，后续完善
- 写字板的简单实现，测试为主
- 添加鼠标的监听，在鼠标点击后记录，在鼠标拖动时给出反应
- 后续需要在鼠标拖动时显示线条

2020-12-06

写字面板的编写和遇到的Bug

- 按下开始按钮后打开写字板
- 打开写字板后拖动鼠标显示线条 **-> Bug：在写这部分的时候由于没有将初始点进行更新，所以画出的线条是从一个点向外散射的线条,而使用 `Graphics` 画出的又是一个个小点，所以用 `Graphics2D` （已解决）**
- 使用写字板进行一些简单的测试 **-> Bug：以压力为测试条件，但无法从写字板读出笔的压力值（已解决）**
- 正常获取压力值 **-> 应该是使用 `PenValue` 里面的 `Pressure` 里面的函数**
- 能够将每次测试的笔尖压力记录在相应的文件中

2020-12-07

在写字面板的操作和Bug

- 打开写字板，画一条线 **-> Bug：当开始画线后会出现开始按钮**（已解决）

2020-12-09

对与界面新要求

- 在初始界面需要有用户ID和组数Block的数据输入
- 需要有一个选择练习按钮
- 需要在界面中新增选择模式按钮
- 开始按钮 **->** 按照初始设计不变，但需要调整位置

对写字板的新要求

- 写字界面分为传统写字板界面和优化后的写字界面

传统写字板界面要求

- 在传统写字板中加入颜色选择和粗细选择按钮
- 最好能显示测试者当前的颜色和粗细以及应该选择的颜色和粗细
- 在写字板中需要加入两个颜色区域，用于提醒测试者在该区域进行颜色选择和笔的粗细选择
- 记录测试者在做颜色选择和粗细选择中的所花的时间

还有三个与传统界面的做对比的写字界面

- 待讨论.....

2020-12-18

重新对用户初始的登陆页面进行设计

- 新加**用户ID**的输入框
- 新加**实验组数**的输入框
- 新加**练习**选择框
- 但界面的排列还有待改善，目前只考虑方法的实现

完善笔的信息记录

- 获取笔的角度
- 获取笔的旋转角

界面逻辑需要完善

- 当用户选择**开始**后，登录界面需要被写字板覆盖

传统写字板类的创建

- 新添加一个传统写字板的类，用于实现相应功能

2020-12-19

对登录界面修改

- 解决**2020-12-07**时画线会出现开始按钮的Bug

- 当用户选择开始按钮后，登录界面被写字界面覆盖
- 将四个写字模式的按钮添加到界面

传统写字界面

- 添加两个菜单栏，分别是颜色和像素
- 添加颜色和像素的下拉菜单
- 画线的逻辑需要修改

2020-12-20

传统写字界面的实现

- 通过用户选择下拉菜单栏，来实现对画笔颜色的选择 **-> Bug:一旦切换为某个颜色，所有之前的线条都会变成当前选择的颜色**
- 像素中下拉菜单中对画笔的修改暂未实现

点类

- 为了方便对点进行信息处理，建立点类，**也是为了解决切换颜色后所有线条会变为当前颜色的Bug**
- 可以保存当前点的信息
- 通过读取点的位置来画线 **-> Bug:会出现当重画一条线时，新的线会和旧线连接（已解决）**
- 对点的颜色逻辑处理需要修改，可以做到对线条颜色进行实时处理

2020-12-21

传统写字界面部分实现

- 通过下拉菜单的形式来对画笔的颜色和粗细进行选择的功能正常实现
- 在写字界面加入两个实验区域，一个为颜色测试区域，一个为像素测试区域
- 当线条进入颜色测试区域时，会提示待选颜色
- 像素测试区域暂未实现
- 颜色测试区域 **-> Bug:当画笔进入到测试区域的时候会出现提示颜色乱跳的情况 -> 分析原因：可能是判断逻辑里有问题，到时候再测试一下 -> 问题确认：随机数生成位置不对**
- 提示颜色乱跳问题 **-> 解决办法：将随机数作为属性放入到点类中，当发现点进入到颜色测试区域，会获得该点当时被赋得随机数，作为颜色提示的依据**

2020-12-22

传统写字界面更新

- 开始对像素测试区域进行实验，发现原始逻辑存在问题，初始的解决办法不太适用，开始尝试使用手写一个对测试区域的监听来实现对进入测试区域的对应操作，同时，也为了提示方法的优化
- **问题：** 当界面重绘时，发现有些组件不能在重绘界面出现 **-> 解决办法，将原来的界面分割为两个区域，分别进行不同的操作**

2020-12-23

传统写字界面完成

- 设置在左边界面的中提示标签位置
- 写监听，按照触发事件提示相应动作
- 监听用户当前的笔的颜色和像素，当用户切换颜色和像素时，也跟着变化
- 后面就该写进入测试区域的监听了 **-> 当用户进入颜色测试区域时，在左边的提示区域会显示应该切换的颜色，同理，当进入像素测试区域时，会在左边提示应该切换的像素**
- 对测试区域的监听不能使用现成的监听，还是得自己写一个对应的监听，这样的话，传统写字界面的功能就大致写完了
- 不用监听，直接可以使用鼠标滑动的那个监听器
- 传统界面大致完成

新增要求--->登陆界面标签修改

- 将 **四个写字板** 改为 **四个模式**，细化 **除传统写字板外另外三个模式，比如实列化 -> 实列化模式,PAT实列化**

新增要求--->传统写字界面修改

- 将测试区域修改为长条形
- 去掉默认的颜色和像素在选择菜单中，保证三个颜色对应三个像素
- 优化颜色提示和像素提示
- 数值记录

2020-12-31

修改传统写字界面

- 修改便签的名称
- 修改测试区域 **->** 将测试区域修改为长条形
- 修改颜色初始的选择菜单，将选择的黑色去掉，将画笔的颜色默认设为黑色
- 修改像素初始的选择菜单，将画笔的初始像素设为1个像素
- 明确模式中所要记录的数据
- 在传统界面的提示区域优化工作未完成

2021-01-01

左边的界面提示信息美化

- 左边区域中提示信息以文本框的形式出现，而不是纯文字
- 左边区域的显示情况待讨论，暂时先以这样显示
- 开始尝试将数据保存在 `csv` 文件中，可能会花的时间有点久，因为要记录的数据比较多

2021-01-02

传统写字界面的修改

- 对于用户的提示信息需要修改，到时候确定后再进行修改，不然重复修改太麻烦了

信息的记录

- 将信息记录，但要保存的信息需要重新选择
- 按照要求记录信息 **出现的问题-> 文字记录在文件中变成乱码 -> 解决办法：修改格式，将其改为 `GBK`**
- 但文字写入的格式需要修改，在每个列标题中加入 `","` 作为 `csv` 文档的分隔符
- **Bug: -> 最后一个方位角无法记录在文件中 -> 解决办法：把这个数据记录两次**

2021-01-03

信息记录

- 能够直接获取的量已经可以正常记录
- 但其他的实验数据的保存还要在代码中添加

- 当用户进入到任意区域内表示开始测试，这时候开始记录实验数据

写字界面测试区域修改

- 减小两个颜色测试区域之间的间隔

写字界面的键盘监听

- 当按下一个键后，清空界面，开始下一次测试
- **Bug： -> 键盘监听无响应 解决办法：除了要使用 `addKeyListener` ，还要让其成为 焦点**

2021-01-04

绘画时间信息记录

- 信息记录存在问题，笔的间隔太短时会出现0秒的情况，而且时间上的算数运算也存在问题
- 将结果保留两位小数就可以得到正确的结果 **原因 -> 在计算 / 运算的时候，结果自动保留为整数，而且是向下取整，这就导致了精度的丢失，而产生较大的误差**

2021-01-06

目标颜色和目标像素的信息记录

- 将目标颜色的信息和目标像素的信息记录在csv文件中
- **注意：** 信息要在进入区域后就开始记录，否则当笔迹出了测试区域后，提示信息会变为空，如果在最后记录的话就只有一个空的字符串

错误切换次数

- 在实验结束后目标颜色与实际颜色做比较，目标像素与实际像素做比较，记录不同的次数

误触发错误

- 当还没有进入测试区域时，用户就开始切换颜色或者像素此时，误触发数加一

一组实验的操作次数

- 当用户结束一次操作后，次数加一

2021-01-07

返回界面设置

- 做完所有组后返回登陆界面
- **Bug ->** 但是在设置关闭当前的画线界面，再打开新的登陆界面会出现两个登陆界面，一个是之前的，一个是新打开的
- **Bug ->** 当使用新的界面登录后会出现上一次的信息还保留在这一次的实验中
- 这个做完所有的次数重新返回到实验的登录界面感觉要花挺长时间的，因为自己写的还是不如意

2021-01-10

登录界面选择模式的修改

- 传统界面的样式不变
- 其他三个模式细分为 `P-T-A` 的不同选择模式
- 不同模式中记录的数值还是一样的
- 对原始的登录界面的每个插件的位置还要重新设置
- **Bug ->** 当将所有模式的选择按钮都在登录界面布局好了之后，除了传统模式，其它在同一排的模式均是倾斜 **-> 解决办法：由于采用的是绝对布局，只能自己把setBound的四个参数慢慢调**

传统写字界面的提示信息样式的修改

- 将原先的两个分离界面去掉，整合为一个界面
- 当前颜色和像素：将当前的颜色还是按照颜色块显示，像素以文字框的形式出现
- 提示切换颜色和像素：提示颜色和像素均以弹窗的样式出现 **->** 看看弹窗的有那些形式，之前也写过一个弹窗，但是会有异常（只是简单的测试一下）
- 将原来的两个测试区域合为一个测试区域，但任要做出两次的提示信息
- 这样的修改相当于把传统界面重新写一次
- 改着改着出现一堆Bug，明天再改

2021-01-11

颜色提示和像素提示

- 当用户的绘制进入到颜色和像素测试区域时，以弹窗的形式提示用户应该切换的颜色和像素，但需要注意的是，当弹窗出现时，用户的绘制会被打断，此时系统不会判定为鼠标抬起，所以需要在弹

窗出现时记录当前线条的绘制时间

颜色和像素的测试

- 一直都在想怎么让颜色和像素能够组合的出现，一种颜色对应是三个像素，而且每一组中都必须将所有的条件都测试一遍，我一直没有好的想法，所以留到现在才开始写
- **现想法：** 将所有的颜色和像素按对应情况分别放入一个颜色集合和像素集合，当每次测试的时候，将集合中的颜色和像素取出，显示给用户，并且将已经显示过的颜色和像素移除，直到集合为空，**缺点：** 所有的颜色和像素只能按顺序出现

定义一个新的类-CompleteExperiment

- 用来提供一次完整实验的实验所需要的信息和变量

传统界面

- 传统界面的完善在今天已经完成了，但还是有一些小的 **Bug** 需要修改一下 😄

2021-01-12

登录界面的底层逻辑修改

- 为了解决登录界面的一些小的问题，对登录的代码进行一些修改
- 对登录界面的排版和按钮位置做出一些修改 **->** 将ID，组数，选择的模式等一些组件采用组合的方式组装到面板中，比原来的情况更好查看
- 登录界面的优化完成

2021-01-13

开始对传统面板的逻辑进行优化

- 对数据的记录和初始化的问题需要解决
- 传统面板的界面优化 **->** 尽可能美观一点 **->** 选择一个好一点的布局
- 可以参照一下登录界面的布局
- 传统界面的整体结构完成，后面细化一下就可以了
- 标签排版 **->** 为了让界面更贴近PPT的样式，只能自己采用没有布局，采用坐标来精确排版了
- 但有一个问题就是把绿色区域合并了，那就是一进入测试区域就开始弹窗，还是缓冲后就开始弹窗

对传统界面的组数判定问题

- 当测试用户输入完组数时，应该把当前的组数做为后续的是否完成所有组数的测试条件，但目前只是把条件记录，并不能使用当前的信息
- **解决办法->** 当打开用户选择的模式时，将组数作为参数传入到选择的模式中，这样就可以在选择的模式中，使用该参数

当打开新的登录界面，原始数据还有存留的问题

- 当用户把所选的测试组数进行完，重新登录进行新一轮测试时，有的在上一轮的测试数据还保存在当前这一轮数据中，比如进行的实验组数，当上一轮的测试结束后，下一轮的实验组数是在上一轮进行的实验组数的条件下，继续进行 `+ 1` 操作
- **解决办法->** 当重新进行登录时，对相关数据进行初始化操作
- 传统界面全部完成，明天开始写实列化模式，冲冲冲 😆

2021-01-14

开始着手对压力实列化界面的书写

- 创建压力实列化类

对传统界面的弹窗修改

- 取消颜色提示和像素提示以弹窗的形式出现，改为在当前颜色下方展示
- 将颜色提示和像素提示改为随机组合的方式 😢

csv文件记录的数据修改

- 在原先的基础上，添加对颜色切换的时间和像素切换的时间 **->** 也就是将模式切换时间细分出来
- 将 `模式切换错误` 细分为 `当前实验` 的 `颜色切换错误` 和 `像素切换错误`
- `误触发数` 也一样按照模式切换错误那样来处理，为了保险起见，还是把误触发也分为 `颜色误触发` 和 `像素误触发`
- 将时间以毫秒为单位保存

传统界面的测试区域

- 在绿色测试区域加入一段缓冲区

2021-01-15

传统界面的修改完成

- 把昨天对传统界面的修改完善，因为昨天已经把大部分的修改完成，但测试的时候有一个小的 **Bug**，今天把它修改一下
- **Bug->** 当按下空格清空时，当进行第二次实验进入到颜色测试区域时，会导致颜色和像素提示一起出现，颜色提示为当前实验的目标颜色，而像素提示为上一次的像素提示
- **解决办法->** 当按下空格时，把颜色提示相关的和像素提示相关的隐藏，将 坐标 和 高度 都 设为0
- 终于把这个界面给写完了😭

开始着手写P-实列化界面

- 先把这个界面写出来，因为界面和传统界面一样，就是切换颜色和像素的方法不一样，这个是通过压力的值来实现颜色和像素的切换
- 这个界面最关键的还是通过对压力的实际值来对颜色和像素进行选择，这个到时候还得考虑一下
- 把这个写完后， T-实列化 和 A-实列化 就很容易了，就是参数不一样
- 终于get了一个好的方法来实现通过压力对颜色和像素进行选择 😉
- 明天开始把这个功能的框架先搭出来

2021-01-16

开始对P-实列化界面的核心部分进行编写

- 新建 `PAExperimentPanel` 类，该类主要是将笔的参数变化通过动态的方式展现出来，主要是当用户进行颜色和像素选择时，进行直观的展示，方便用户的操作和选择
- 将压力值的实时变化以动态图像的形式展现出来
- 当达到压力值条件时，弹出颜色选择和像素选择菜单
- **Bug->** 在测试压力检测中使用 `repaint()` 方法调用 `PAExperimentPanel` 类中的 `paintComponent` 的方法时，竟然没有反应？导致动态图像不能显示出来
- OK，对上面这个**Bug**的解决 **->** 没有把 `PAExperimentPanel` 加入到当前页面中，所以才没有对 `repaint()` 方法产生反应
- 但是正常重绘后，无法对压力值进行图像般显示 **问题根源->** 是由于在 P-实列化 中有两个区域（一个是 **画线区域**，一个是把**压力进行动态显示的区域**）所以存在监听冲突，只能对一个区域进行监听 **解决办法->** 将原来那画线区域合并到压力的动态显示区域
- 现在对 P-实列化 的界面的功能已经大致实现了，还剩下当压力达到规定值后，将颜色和像素选择菜单弹出，同时还要对颜色和像素进行选择，笔的颜色和像素也要进行相应的改变
- 但是要对可以进行颜色和像素选择的区域进行划分，不能让这个动态图像在绘制过程中一直都在

2021-01-17

P-实列化界面选择菜单的技术要求

- 当压力到达预设值后，颜色和像素选择菜单可以弹出，但只是显示图像，还不能进行选择操作
- 当压力到达规定值后，原来的动态显示的压力图像关闭，打开选择菜单
- 对颜色和像素菜单的编写有了一些简单的灵感，但还是要来实践来测试是否可行
- 当用户在进行选择时应该暂时取消此时笔的绘画作用，避免在选择过程中也会有笔迹产生
- 当然了，但把所有的技术都实现了，还有些东西需要微调，这个等把功能都实现后再进行微调

2021-01-18

在压力到达指定值后，弹出菜单

- 菜单的位置需要是固定的 **->** 也就是说，当笔尖的压力在到达预设值时，此时应该记录下达到压力点的位置，做为菜单弹出的位置参考，同时当菜单弹出时，笔应该作为选择菜单栏的工具，而不再具备绘线功能，当选择完后，此时笔再获得绘线的功能
- 根据笔的位置来判断用户选择的是哪一个菜单栏的函数有一些问题，不能正确判断用户选择的菜单栏
- 菜单栏可以正常显示了，剩下的就是当用户抬笔后，当再次下笔时笔的压力值应该动态显示，在没有达到规定值时，选择菜单不显示，菜单的位置需要跟随鼠标的位置变化而变化
- 对用户抬笔操作后的刷新操作的逻辑有待考量 **Bug->** 当笔正常抬起时，在界面中还存在菜单选择菜单，按正常操作来说，这是不应该出现的 **解决办法->** 我感觉还是界面的刷新问题 **->** 问题原来是出在 `Timer` 这个类的使用方法上，每次应该是在笔尖按下时来调用它的 `restart()` 方法，在笔尖抬起时调用 `stop()` 方法
- **Bug->**在第一笔落下时，会出现一个压力的框图 **原因->** 对该框图最初定义的 `boolean` 值错误

2021-01-19

对P-实列化界面的颜色和选择菜单的优化

- 在最初的几天里，主要是实现了该界面功能的框架实现，现在是对它进行细致的优化，以达到使用的效果
- 在进行菜单栏的选择时，不用记录笔的轨迹

- 当画笔停留在颜色菜单中时，颜色分支菜单弹出，可以在分支菜单中选择想要的颜色，像素菜单也一样
- 但是触发选择分支的菜单条件还是没有想好，因为每个区域都是以坐标范围为参考的，所以这个要好好考虑一下，因为后续的颜色和像素赋值也需要通过这个判定
- 当坐标进入到颜色菜单区域时，在原来的基础上，将颜色菜单区域扩大 **->** 也就是扩大至包含颜色选择的分支菜单，这样当要进行颜色选择时，就不会出现坐标不在颜色区域时，颜色分支区域收回的情况了，像素分支菜单的处理也一样 **->** 这样的想法还是不行，还得想想其他办法 😭
- **解决办法：** 添加两个 `boolean` 类型的值，用来控制是否打开颜色和像素分支菜单，当打开颜色分支菜单时，像素分支菜单关闭，当打开像素分支菜单时，颜色分支菜单关闭
- **Bug：** 对颜色选择的结果和像素选择的结果的保存问题 **-> 原因：** 但颜色菜单被选择后，没有进行选择颜色的话就是,值就是 `-1` ，而在颜色分支选择菜单中，会对这个 `-1` 进行运算，最终导致结果为 `0` **-> 解决办法：** 将颜色和像素的变化条件修改为当有颜色被选择或者像素被选择时，颜色值和像素值才会产生相应的变化，这个**Bug** 我找了好久，总算解决了 😁
- **Bug：** 菜单框的粗细会随着像素的选择而变化 **-> 原因：** 当笔的粗细被选择后，由于使用的是同一个 `Graphics2D` ，所以在将像素改变后，所有的线条的粗细都会改变 **-> 解决办法：** 当每次根据当前点的像素绘画完线条后，将线条的像素变为原样
- 明天的主要任务应该就是对在 `P-实列化界面` 的实验信息记录了

2021-01-20

P-实列化界面的完善

- 将P-实列化界面中需要记录的信息在相应位置进行设置，保证能够写入到 `csv` 文件中
- 设置压力调用条件的压力值需要确定
- **Bug：** 当第一次达到压力值后，第二次会很容易达到目标压力值 **原因：-> ** 应该是在第一次的基础上在加上了当前压力值 **解决办法：-> ** 在抬笔之后对 `tablet` 进行初始化
- `P-实列化` 界面完成

2021-01-21

记录信息的修改

- 对模式切换时间的记录方式修改，更贴近实际一点，待修改 **->** 修改为任务提示出现时间到切换完成结束这段时间
- 压力，方位角和倾斜角的具体记录待讨论

P-实例化

- 模式切换压力值的数值定义有待讨论

T-实例化的编写

- 开始对 T-实例化 界面进行编写，大致和 P-实例化 差不多，应该就是参数不同，一个是压力一个是倾斜角

2021-01-22

将模式切换时间记录方式改变

- 新加四个变量，分别记录颜色和像素切换的开始和结束时间
- 对 PenData 类中的逻辑进行一些修改

T-实列化界面的编写

- 先将 T-实例化界面 大致界面写出来，因为都是实例化的界面，所以大部分都是一样的内容，就只有核心部分不一样
- 该界面的核心部分的编写还有待考虑，先看看类似的程序看看有那些可以借鉴的
- 对倾斜角的动态界面的函数进行编写
- 对显示倾斜角的动态界面进行角度测试，目前能够正常显示能够触发菜单的角度区域，同时界面展示为扇形

2021-01-23

T-实例化界面完成

- T-实例化 界面已经完成，剩下的问题应该就是在测试过程中的问题，这都是隐藏的 **Bug** ，到时候再解决

A-实列化界面的编写

- 开始对 A-实列化 界面进行编写
- 先将实例化相同部分的先进行编写，后续再将该部分不同的部分再进行独立编程
- 对 A-实例化 界面的核心区域进行编写，也就是获得当前的方位角，并对当前方位角来进行颜色和像素值的切换，同时还要将方位角的位置进行实时图形的展示，使用户可以直观的观察和进行相应的展示

2021-01-24

T-实例化界面的方位角实时显示界面的编写

- 开始对方位角的实时显示界面进行实现，要求能够随着笔的实际方位进行实时体现
- 方位角的范围有点问题，这个需要讨论，方位角的动态显示需要斟酌
- 方位角的变换也需要验证和计算来保证结果的准确性，这样才能正常实时显示当前方位角的图像

对颜色和像素切换合法判断的优化

- 添加两个 `boolean` 的变量来对颜色和像素切换的情况来进行判断当前的切换是否合法 **->** 当颜色和像素提示出现后，那么当前的切换就是合法的，否则就是误触发，在一次颜色和像素切换实验结束后，要将这两个变量重新变为 `false`

2021-01-25

修改P-实例化中压力触发值

- 在原来的触发压力条件下进行修改，因为原来的压力值设置的太小，当进行切换时，很容易就会达到切换条件

A-实例化界面

- 依旧是对 `A-实例化` 界面中方位角的动态显示界面进行编写
- 倾斜角的动态界面可以正常显示，可以随着用户持笔的不同方位来进行冬天展示
- 但在倾斜角的动态展示图像有些细节许需要修改，以及检查方位角的值是否正确，反馈是否也是正确的
- 倾斜角界面的角度展示存在问题，对于角度的展示以及常用区域的显示问题都需要解决

2021-01-26

对A-实例化界面的方位角问题

- 方位角界面的展示一直存在问题，展示的效果一直不是很正常，所以对它的逻辑部分进行一下分析和逐步演示，按照结果进行相应的修改
- 方位角的展示测试 **->** 显示没有问题，输入的角度都可以正常显示，角度之前的范围也准确
- 方位角的数据测试 **->** 通过对方位角旋转一周后来查看方位角的具体记录方式 **->** 发现笔的方位角和系统记录的方位角是相反的，即系统是逆时针来表示 `0-360` ，而笔的方位角是顺时针来表

示 `0-360` 的，所以之前的显示才一直不正确
- 对方位角的常用区域和非常用的角度表示方式改为 **笔的角度表示方式**
- `A-实例化` 界面完成 😄

2021-01-27

开始对P-离散化界面进行编写

- 离散化界面的编写和实例化界面有相同部分，但是菜单选择的样式和实例化不一样
- 对压力值进行细分，分配给颜色和像素
- 离散化界面的颜色和像素选择的情况和实例化不一样，因此对于选择部分要进行独立编写

对PSExperimentPanel进行编写

- 对实例化界面进行了简单实现

2021-01-28

对实例化模式选择菜单方式的修改

- 实时检测笔的数值，当数值达到预设值后，显示菜单，此时用户对颜色和像素进行选择，以抬笔作为颜色和像素选择结束的依据
- 将实例化模式的按照要求进行修改
- 对最初的笔的压力，倾斜角和方位角的各值得动态展示取消，用于在离散化中使用
- 在整个实验中，对笔的压力，倾斜角和方位角进行实时监测，一旦达到预设值就展示颜色和像素选择菜单
- 解决在按下空格后笔的颜色和像素没有重新变为 **黑色** 和 **1像素**

离散化界面的函数的完善

- 解决一下离散界面调用函数的问题

2021-01-29

压力值离散化界面

- 压力值的离散界面的颜色和像素菜单选择模式和实例化不一样
- 实例化是通过预设压力值来作为打开菜单的判断依据，当笔尖的压力到达或者超过预设值后，打开颜色和像素选择菜单来对选择
- 离散化的选择是通过将不用的压力值区间进行细分，每一个压力区间作为颜色和像素的选择
- 可以将原先作为实例化压力动态展示的画面来做为离散化界面的菜单选择来使用

离散化界面的压力值菜单设计

- 将可以做为菜单的压力区间分为6部分，暂时按照整数来分配，因为原区间不能作为6等份均分
- 将颜色标签和像素提示嵌入压力区间 **->** 压力到达颜色区域 **->** 打开颜色选择子菜单，进行颜色选择，像素选择同理
- 对颜色和像素菜单进行区域划分，划分之后颜色和像素区域都太小了，要是在颜色和像素菜单划分子菜单，那就更小，不容易进行辨别，也更容易出现选择错误
- **Bug ->** 当进入到颜色选择菜单时，表示当前颜色的像素点会被覆盖点，因为对于颜色的选择是以颜色块的模式块的形式进行的选择 **解决办法->** 当进入到颜色和像素区域时，不在通过像素点来进行表示，而是通过用一个框来框住当前的二级菜单区域
- **Bug ->** 颜色选择块不会随着笔的走势跟着一个走 **解决方法->** 不断跟新颜色块的位置
- **Bug ->** 在笔画线的过程中，文字会出现错位的情况 **解决办法->** 是因为在达到该菜单的触发条件，没有对它进行相应的操作

2021-01-30

压力值离散化界面的二级菜单的设计

- 当压力到达到颜色或者像素菜单时，显示该菜单下的二级目录，并通过框选的形式来对用户当前的选择进行视觉反馈
- **Bug ->** 当压力值到达像素菜单时，二级目录没有展开 **问题原因 ->** 没有将宽度比例和菜单高度想乘
- 在对压力监听的过程中，对二级菜单进行移除和重组操作
- 设置二级菜单选择提示框
- 不在使用颜色标签，而是直接在画板上填充
- 二级菜单提示框完成，现在就是按下一个键后能够进行颜色和像素的切换
- 由于更改为一条线后，所以对每段线的绘制时间需要讨论

压力值离散化界面完成

- 还剩下实验数据的记录，因为和之前有一些不同，所以需要讨论，再做编写

对倾斜角离散化界面进行编写

- 完成倾斜角界面的外围程序

2021-01-31

倾斜角离散化界面核心程序的编写

- 对倾斜角的动态界面进行设计，问题和压力离散化一样，可用区间太小，而且要在两个不同的区域都加入颜色和像素的菜单，同时还要嵌入二级菜单
- 将倾斜角区域进行划分，为后面的一级菜单和二级菜单位置的设置做数据基础
- 对倾斜角的动态显示界面的大小进行相应的修改，为了满足用户对选择菜单栏的观察和选择
- 对倾斜角的动态界面的提示信息进行修改，显示为颜色和像素一级菜单，并且当达到某一菜单时，显示该菜单下的二级菜单
- 现在的问题是解决文字的倾斜显示，因为正常文字是平行显示的，而现在是要进行按一定角度倾斜显示

2021-02-01

P-离散化界面的二级菜单修改

- 在原来的界面中，当用户进入到颜色（像素）菜单中时，会在颜色（像素）菜单栏的范围中，展开二级菜单，不利于用户对颜色（像素）的选择，所以将其修改为当用户进入到颜色（像素）菜单栏中时，按下按键进入到二级菜单，二级菜单的展开范围时是整个菜单栏，方便用户的选择
- T-离散化界面也按这样修改
- A-离散化界面按原来编写，因为它的范围够做颜色和像素选择
- **Bug ->** 当在颜色菜单展开二级菜单时，像素没有移除 **问题根源：** 判断条件不完善
- 离散化界面修改完成

T-离散化界面核心界面的完善

- T-离散化界面的划分问题在讨论后，按照最后敲定的方案来进行编写
- 但是标签在角度的展示问题有待解决
- 核心部分框架完成，剩下的就是将标签带入，将二级菜单进行处理
- 对标签进行讨论，决定将标签的显示改为当角度进入到指定区域时，显示对应的提示文字
- 调整显示标签的位置，便签的位置要自己调一下

20201-02-02

T-离散化界面对提示菜单的完善

- 将颜色和像素菜单区域进行划分，加强用户的视觉体验
- 由于没有直接对扇形区域进行描边的函数，所以采用对扇形区域进行小角度填充颜色的办法，来表示一条线
- 对整个扇形区域的弧线描边
- 当用户进入到颜色和像素菜单时，取消箭头显示，改为颜色覆盖，提示用户进入的选择菜单区域
- 对T-离散化界面的二级菜单进行编写，当二级菜单展开后，对一级菜单进行覆盖，同时，重新对角度范围进行划分
- 二级颜色菜单的展开比较容易，只要在相应的扇形区域中，填充对应的颜色就可以了
- 而二级像素的菜单展开就比较废时间，因为你需要去调每个提示字的位置而这是不好控制的
- 剩下就是对二级菜单的确认了，这样T-离散化界面就写完了
- 对二级菜单的确认，当进入到二级菜单后，按键表示对当前颜色（像素）进行选择
- **Bug ->** 当在下半部分进行选择时，会出现像素的颜色同时展开二级菜单的情况 **原因 ->** 因为二级菜单切换的范围与一级菜单有重合，当在重合部分选择后，会打开这个位置的二级菜单 **解决办法 ->** 对二级菜单打开的逻辑进行优化
- T-离散化界面完成

2021-02-03

A-离散化界面的编写

- 对A-离散化界面的动态图像进行优化，主要还是一级菜单和二级菜单的嵌入问题，由于方位角的范围比较大，所以当二级菜单打开后，不再重新划分范围
- A-离散化界面交互界面编写完成，剩下的是对方位角的动态图像的展示，以及菜单的展示和选择
- 测试笔方位角和系统角度的区别
- 动态展示图像按照笔的方位角，箭头的方向要和笔的方向一致，所以 `A-实例化` 的界面也要做一下调整

2021-02-04

P-离散化界面的小细节

- 对菜单显示的位置要进行修改，不让会出现遮挡，因为界面是由两部分构成，不能跨区域显示

A-离散化界面的菜单部分的编写

- 根据系统和笔通道的不同性，做出相应的修改
- 将颜色的一级菜单和二级菜单进行范围划分，同时对像素的一级菜单和二级菜单进行相应的处理
- 在圆形区域里添加颜色一级菜单和像素一级菜单，分配在相应的区域内
- **BUg ->** 在测试的时候，颜色二级菜单在一定区域内会随着笔的转动而转动 **原因 ->** 处理箭头转动的时候会对所有标签的位置产生影响 **根源问题 ->** 是因为将 `graphics2D` 设置了旋转转换，所以会导致后面的标签会旋转
- 颜色二级菜单的区域分配完毕，当在一级菜单处选择后可以正常进入颜色二级菜单
- 像素二级菜单的区域分配完毕，当在一级菜单处选择后可以正常进入像素二级菜单

2021-02-05

A-离散化界面中标签的旋转问题

- 这个问题是由于为了显示笔的旋转，而将界面设置为了旋转转换，也就是说，此时界面会随着参数而旋转，而插入的标签也会随着界面的旋转而旋转，目前我还是没有想到什么好的解决办法，知识实在是有限，而网上对这类方法也没有详细的说明
- 所以我只能暂时改为进入到菜单区域后再显示颜色标签和像素标签，因为此时不会出现旋转转换，这是我能想到的办法
- 想再原图像上再加入一个箭头的界面，但是失败了，只能按照上面的想法来写了
- **对标签会随着箭头旋转问题的最终解决办法：** 取消原箭头的设计方法，采用 `Graphics2D` 中的 `fillArc()` 方法，因为改方法会按照参数的设计来进行颜色覆盖，所以当把参数 `arcAngle` 设置的相对较小时，会展现成线条
- A-离散化界面完成 😄

2021-02-06

对之前编写的项目进行逻辑检查

- 因为增量化和实例化界面相差不是很大，按照描述可以共用的地方应该很多，所以先检查之前编写的界面的数据记录部分
- 文件中的数据记录的检查，因为之前一直在写项目的界面部分，所以对文件的记录没有花太多精力，所以在把实例化模式和离散化模式编写完成后，开始对文件记录的部分进行检查
- 主要是检查实验所需要的数据的记录逻辑，以及数据记录的是否正确，感觉有一些部分需要修改

2021-02-07

文件记录按键的修改

- 由于离散化模式对按键的特殊要求，导致和原来的文件记录按钮产生冲突，所以对之前的按钮进行一些修改
- 同时继续检查数据的记录，对错误的地方进行修改
- 完善有些模式的数据记录
- 对传统模式和实列化模式的数据记录部分进行了修改
- 由于当时编写离散化界面的时候，大部分完成的是界面部分和功能的实现，对数据的记录几乎没有进行编写，而且在离散化界面中，落笔和抬笔操作均只进行一次操作，和之前的模式有很大的不同，所以数据记录部分就一直放在，现在开始进行数据部分的处理
- 对压力离散化界面的位置进行调整
- 对开始时间和结束时间进行修改

2021-02-08

对增量化模式的编写

- 增量化模式和实例化模式在程序结构上有很大的相似之处，所以先进行相似部分的编写，然后根据不同的模式下的要求进行独立编写
- 数据记录方面，因为之前已经检查过了，所以这部分没有什么问题
- 在增量化模式中添加一个提示信息，提示内容为即将进入到菜单选择
- 对增量化模式中添加一些该模式独有的变量

所有的模式均编写完成

- 今天就把所有的模式都编写完成 😄
- 对该项目操作的注意事项可以参考说明书
- 但还有个练习模式作用未知

2021-03-02

对实例化模式进行修改

- 将之前颜色提示文字改为颜色块提示
- 将提示菜单改为在右侧显示

数据记录方式的修改

- 实验开始的时间由原来的 **年-月-日-小时-分钟-秒-毫秒**，改为 **小时-分钟-秒-毫秒**
- 压力的记录改为整数形式 **Bug：->** 明明设置的是int型，而且数据在记录的过程中也是int型，但是存储在csv文件后却变成了小数
- **问题根源：->** 原来是之前在压力值后的区分符敲错了，本应该是，，但敲成了．，所以后面的数字就跟着压力值了，而这也就是之前为什么倾斜角没有记录的原因 😄

对于测试区域的细节进行调整

- 在绿色测试区域中添加虚线，以区分 颜色测试区域 和 像素测试区域，**注意：尽量做到均等份**
- 将提示文字做到与虚线对齐，方便测试者查看

2021-03-03

对误触发的判定条件的修改

- 将颜色误触发和像素误触发合为一个误触发，因为当提示条件未出现时，做出菜单调出操作，此时就认定未一个误触发操作
- **Bug:->** 当完成一组操作后，按照要求，一组应该是只有 **9** 个分组操作，但是会出现 **10** 组操作 **解决办法:->** 在一组实验做完后，重新生成测试随机数

2021-03-04

对记录数据的修改

- 将颜色误触发和像素误触发合并为一个误触发
- 将压力，倾斜角和方位角细分为两部分

增量化的敲定

- 对增量化概念的细分和调整，增加时间范围检测，和实例化产生差别

2021-03-05

对增量化模式的修改

- 修改捕获压力的时间间隔，和其他模式的时间间隔做出区分
- 在该模式下，记录笔的信息的值改为当提示信息第一次出现的时候记录当前的值
- 对达到增量化的条件做出修改

实列化模式的数据记录

- 实例化模式的数据记录为在画线过程中所有点数据的平均值

数据记录的Bug

- 除了实例化之外的所有模式在数据记录部分都出现报错
- **原因：->** 因为在实例化部分新加了一个求平均数的数据，而求平均数最后是除以一个点数的，除了实例化部分对点数有记录，其他模式对点数都没有记录，所以导致点数为0，变成了分母，出现了错误

2021-03-16

对增量化模式的判断进行修改

- 新加一个数组，按每350ms进行一次检测，在这7个数据中，只要达到了规定值就进行菜单的切换
- 一组每达到切换的条件就对下一组进行判断，直到达到切换条件

对保存的数据名称进行修改

- 将 **一组实验次数** 改为 **实验组编号**
- 其他的不变

2021 04-02

对文件的修改

- 在文件的保存目录中新增 **偏移线** 和 **ID**
- **偏移线**的定义和保存有待讨论，暂时先不写
- 对**ID**的保存很简单，就是读文件有多少行，保存最后一行的行数，进行 **+1**运算
- 但是这样的建议文件还是不要太大，耗时会比较久，因为对于获取文件的行数需要从头开始遍历到最后一行，现在还没有什么好的优化办法来直接获取

2021-04-12

偏移量数据的保存

- 将绘制过程中所有点的偏移量以绝对值的形式保存在文件中
- 还需要对原来数据的保存形式进行一些修改
- 对新数据的保存进行一些测试

2021-04-13

数据文件的保存形式的修改

- 对偏移量数据的保存格式进行修改，每一个数据一个空间，纵向排列
- 采集点的时间暂定为 `350ms` ，到时候在根据采点的个数进行修改，大致为每一次操作采集 **10~20个点**

对倾斜角离散化的修改

- 原倾斜角菜单的设置会出现选择菜单被上一个界面遮挡的情况，所以对倾斜角的菜单进行一些调整

2021-04-23

对偏移量数据记录形式的修改

- 在最终文件的数据保存中，将之前记录的所有点改为记录一次测试的偏移量平均值
- 平均值保留两位小数

2021-04-26

对偏移量精度的修改

- 最初的保留的两位小数，修改为保留三位小数

保存偏移量的数据格式

- 整数的末尾要和小数一样，在末尾补零

- 其实在程序中会是做出显示，例如 **13.000** ，但在文件中确实 **13** 应该是 **CSV** 文件格式问题，所以数字后面的零直接被忽略了
- **解决办法：** 将数字转换成字符，通过判断字符长度来决定是否添加 **后置0**，很妙，这个方法太复杂了，没有实现
- **另一个解决办法：** 直接在后面加制表符 **\t**就行了，直接解决这个问题