

Matlab大作业

此Matlab程序实现了Audiovisual events capture attention/ Evidence from temporal order judgments (Van et al., 2008)文章的实验1。

此README在Github中阅读体验更佳（链接可以生效）

[Github地址](#)

[PCKit Github地址](#)

实验程序结构

- 实验主脚本为[main.m](#)
- [setUp.m](#)为程序启动时所作设置，包括收集被试信息，定义全局环境变量，隐藏鼠标，初始化PTB和PsychPortalAudio，打开窗口，呈现指导语等
- [setUpSound.m](#)为初始化听觉刺激，在PsychPortalAudio中填充所要播放的声音
- [Trial.m](#)是记录试次条件的，其中只有三个属性和构造函数，分别表示SOA，是否有tone，关键颜色变化在左/右
- [genTrials.m](#)即老师给的用于随机生成试次条件的函数
- [generateTrialCondition.m](#)是用于生成一个block中的所有试次条件
- [generateWithinTrialCondition.m](#)是用于生成一个试次中21次颜色变化的持续时间
- [closeDown.m](#)为程序结束时所还原设置（显示鼠标等）
- [MatlabExcelMac](#)是第三方写xls的JAVA包
- [data](#)是保存数据并包含数据分析的目录
- [DataAnalyzer.m](#)是拟合分析的脚本（包括画拟合图）
- [DataAnalyzer.sps](#)是在SPSS中做了四个重复测量方差分析的SPSS代码
- [PictureDrawer.m](#)是画三个重复测量方差分析图的脚本

实验数据结构

- [raw data](#)为我自己被试的数据
- [fitted.fig](#)是拟合图
- [JND_total.xlsx](#)是我组五人四种条件下的JND
- [PSS_total.xlsx](#)是我组五人四种条件下的PSS
- [JND_PSS_Slope.xlsx](#)是我组五人四种条件下的JND、PSS和Slope

注意事项

- setUp中引入了第三方JAVA包MatlabExcelMac，这是用于解决在Mac下Matlab自带函数xlswrite会出现的问题（会发生无法写入xls而写入csv的问题，csv中没有sheet，不适合本实验的结果记录），此三方库下载于MathWorks官方网站，用法与xlswrite几乎相同，只不过函数名是xlwrite
- 程序依赖PCKit，使用时需将PCKit add PATH入MATLAB
- 为防止出现不同电脑esc键编码的不同（有些电脑是esc有些电脑是escape），程序采用了try catch的方式来避免此问题

实验思路

- 实验思路主要体现在main.m中
- 由于本实验需要“同时”进行频繁的视觉刺激切换、听觉刺激播放、按键事件监听，而PTB自带的时间控制机制(WaitSecs())会阻塞进程，导致无法监听按键事件，故本实验采用“事件轮询”的机制处理问题，即每一个事件都包含两个部分，触发判断器和回调函数，程序在while循环中不断询问注册的每一个事件的触发判断器该事件是否触发，若触发则调用其注册的所有回调函数（同一个事件可以注册多个回调函数），事件可能是一次性的，也有可能是多次的，若一次性的则在第一次触发之后就被销毁。
- 考虑到“事件轮询”的机制与本实验无关，应该作为一个单独的模块，所以我单独抽取出这部分内容形成了PCKit，实现了高内聚低耦合的原则，PCKit也有单独的README可以参考。此外，PCKit作为一套框架，还面向对象封装了PTB的绘制函数，包括注视点（PCFixationPoint）、圆（PCOval）、矩形（PCRectangle），在使用时也比较省心。
- 本实验在PCRunloop中注册的事件的触发判断器分为两种，时间触发和按键触发，分别使用PCKit中PCTimeReachedFireJudgerBuilder和PCKeyboardPressedFireJudgerBuilder工厂方法构建，这两个函数的实质是返回一个闭包，这个闭包函数即是所需要的触发判断器。使用工厂方法可以极大地减少代码量，写一个就可以满足一类触发判断器，否则每个触发判断器都要单独写一个函数实在是太蠢了
- main.m中对于每一个block，每一个trial均实例化了一个PCRunloop（事件轮询的处理类），在PCRunloop中注册了所需要的事件，注册完后启动PCRunloop，即可完成一个试次的实验。每个试次的Runloop的逻辑如下：首先注册0s时呈现注视点的事件，此后使用一个循环注册所有的干扰刺激颜色变化的事件（都为时间事件），并且在关键颜色变化时额外注册三个事件，在关键颜色变化的同一个事件上注册声音播放事件，并且注册在其之后

0.125秒呈现第一个目标的事件，在其之后 $0.125+SOA$ 秒呈现第二个目标的事件。同时，在第一个目标呈现的回调函数中向runloop中加入键盘(z和m)监听事件，并且在键盘监听事件的回调函数中确保在一个键被按下之后两个键盘监听事件都会被注销掉。

- 采用事件轮询的机制也做到了解耦，不同的事件回调函数完全是分离的函数，不会在主程序中混成一团
- 所有绘图均采用了PCKit面向对象封装好的对象，包括PCFixationPoint、PCOval、PCRectangle，在绘制时只需调用对象的draw()方法完成绘制，将绘制逻辑移出主程序，进一步增加了程序可读性并做到了解耦

程序优点

- 低耦合高内聚
- 扩展性强（只要注册新事件就行了，真的超级方便）
- 稳定性好（Mac和Win平台均通过测试）
- 可读性好（变量命名都保证看变量名就知道这个变量是做什么的，回调事件命名也一样）

感想

- 由于PCKit需要接管所有的事件，所以PCKit的引入是侵入性的，不可能一部分实验用Runloop的形式来写一部分不以这个来写，这是比较致命的，但也确实没办法，Matlab本身对于异步的支持太弱，多线程则基本没有，只能这样做。同时PTB的函数也全都是同步的，包括在最后一个参数指定时间的。。。这个实在是让我有点不能理解。。。大概是OpenGL本身是不支持异步提交绘图事件的，所以PTB也没有做其他的包装
- 虽然只是个大作业，但是感觉要想写好真的不太容易，不过PCKit写好之后以后所有的实验程序都可以复用，包括封装好的绘图对象，以后写实验程序应该会比较轻松了~
- 在写程序的时候同组组员积极讨论，虽然没有互相借鉴代码但是也互相了解了对方的代码，大家在交流过程中都获得了一些灵感，特别感谢同组组员郝洋，席可颂，原显智，杨诗翰的共同努力~

Authored By 陈乐天
