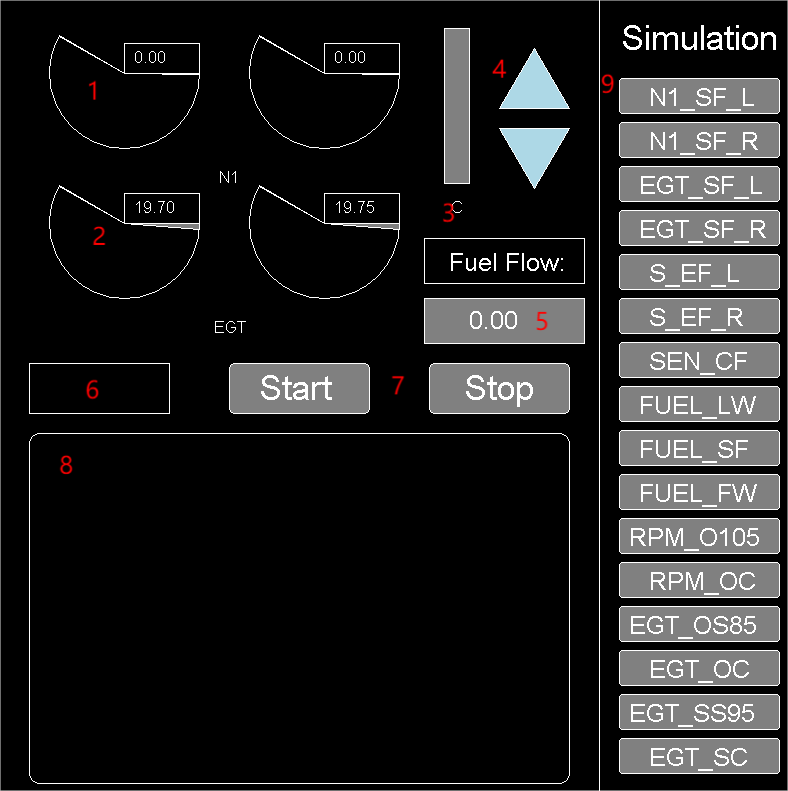
**高级语言程序设计（进阶）**

**期末作业报告**

**2353721 贾博睿**

**一、项目基本情况介绍**

该项目为虚拟发动机模拟测试界面，该界面主要包含以下部分：



1.N1仪表：显示发动机转速的仪表

2.EGT仪表：显示发动机温度的仪表

3.C仪表：显示剩余油量的仪表

4.增加/减少推力按钮：用于增加和减少发动机推力

5.Fuel Flow仪表：显示油量消耗速度的仪表

6.状态显示表：显示RUN或START状态

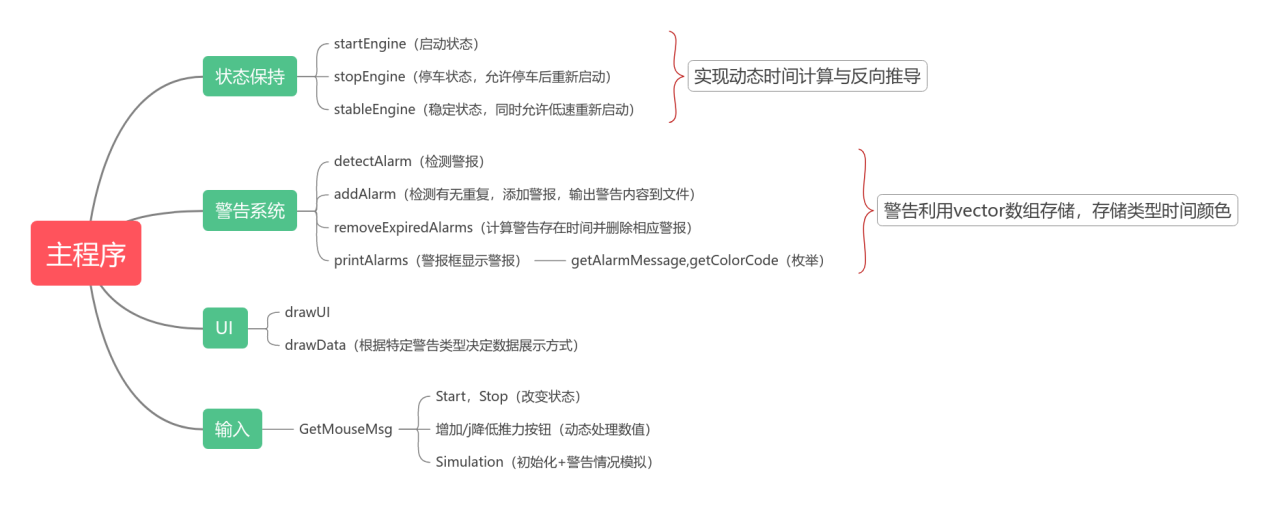
7.Start/Stop按钮：启动/停止发动机

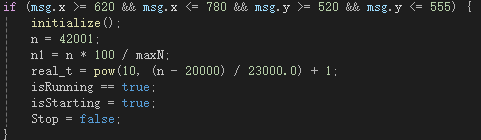
8.警告显示框：显示当前警告内容

9.警告模拟按钮：模拟所有类型的警报

在实现所有基础想和进阶项的基础上，本项目还实现了油量表，稳定状态发动机重启，便捷的警告模拟等一系列内容。

**二、整体设计思路**



主要的结构如上图，警报模拟是利用数值初始化+数值模拟实现的，例如下图：  


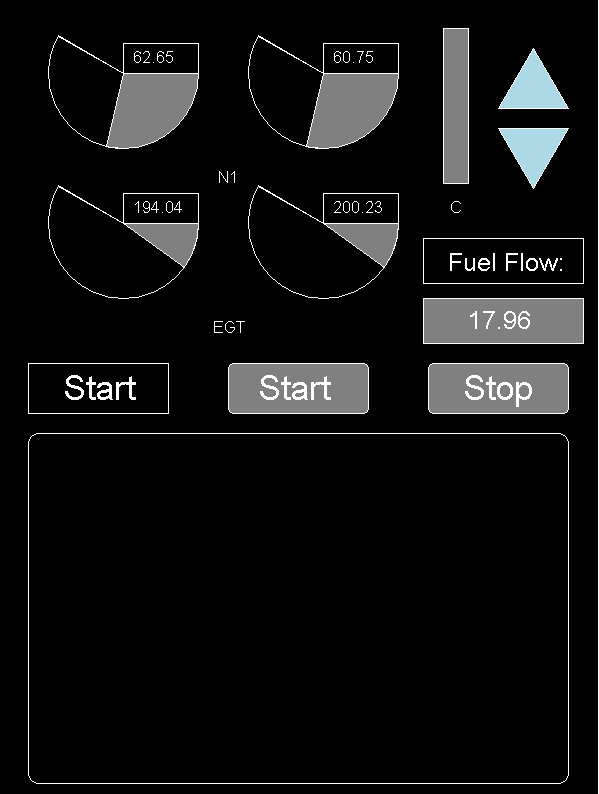
但也有不是这样的，比如说传感器无效的警告，数据还在但是传感器故障了，这种情况下我定义了一个另外的变量来控制（也就是说传感器故障不会在程序正常运行且不进行警告模拟的情况下出现）。

关于循环的主要逻辑，基本上就是：

绘制UI→处理点击→检测添加删除警报→绘制数据→发动机状态函数→时间增加并且结束双缓冲

**三、主要功能实现**

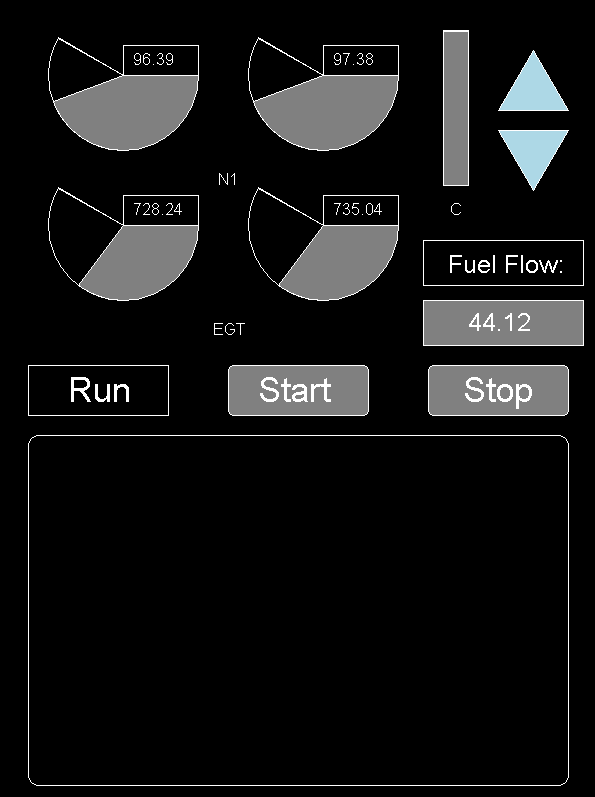
1. 引擎启动



规则：各个发动机转速以每秒104转的速度线性增加，燃油流速以每秒5单位的速度线性增加。该过程持续两秒。此后，燃油流速c按照对数函数速度增加，大致为V=42lg(t−1)+10；同时，转速N按照对数函数速度增加，大致为 N=23000lg(t−1)+20000；温度从T0（本题规定为20℃）开始按照对数速度增加，大致为T=900lg(t−1)+T0。该过程一直持续到转速达到额定转速的95%。

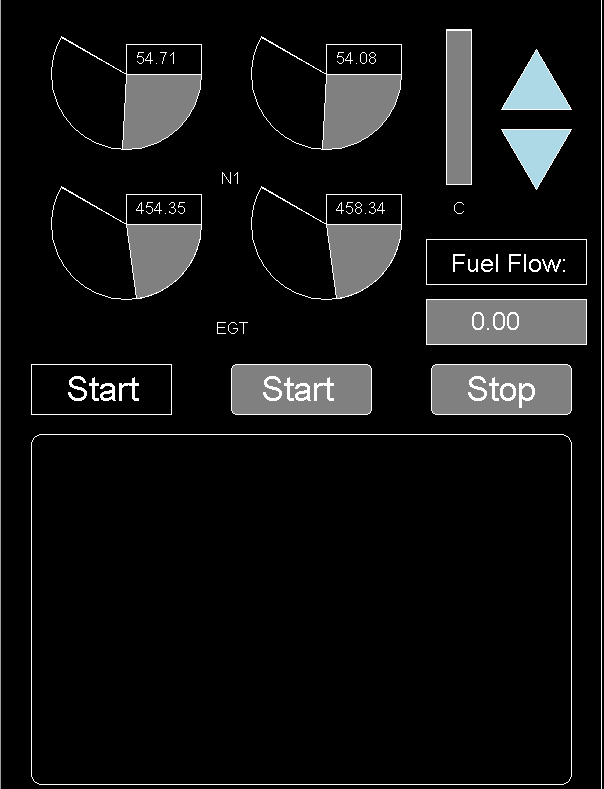
同时，我增加了追加内容：如果在稳定状态通过减小推力按钮让N1降低到95以下，那么可以再次按下Start按钮进入启动状态，而此时依旧遵循启动阶段的数值规则，但是如果直接按照初始时间来进行模拟，会出现n和t的突变，因此对于t分开进行了反向推演，以保证n和t在再启动之后能够平滑变化。

1. 稳定状态



规则： N,T,V 均动态平稳。（可在 ±3% 波动）

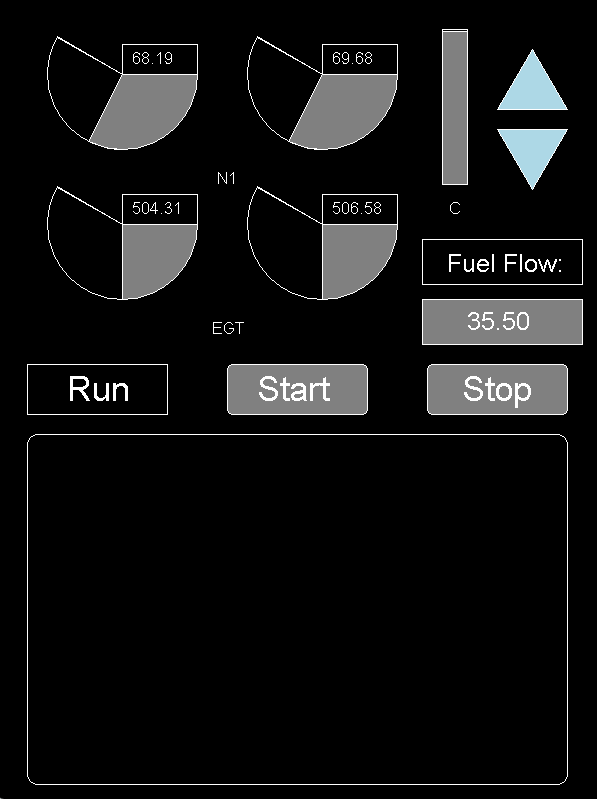
1. 停机状态



规则：燃油流速 V 直接归零，转速 N 和温度 T 以对数方式下降

同时，我追加了规则：在完全停机（即n归零）之后，可以再次按下Start重新启动。

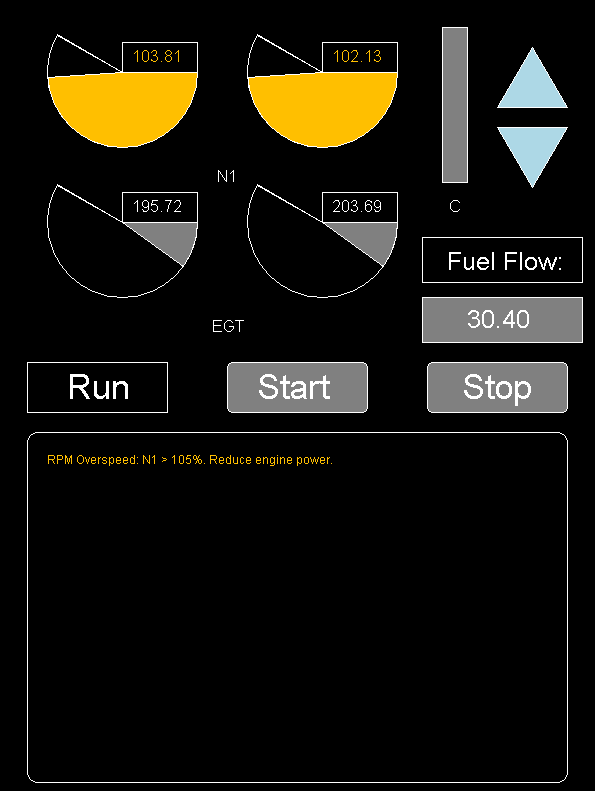
1. 增加/减少推力按钮



（该图是稳定状态按了几下减少推力按钮的状态）

每增加/降低一次推力，则燃油流速V增加/减少1单位每秒， N,T 在 3%∼5% 的范围内随机加/减。

1. 警报模拟



为每种警报设计了自己的逻辑，在初始化之后载入一个预先设定好的状态，这个截图是进行超温1模拟后的情况。（对于传感器故障，实现的效果是输出故障，但是实际上的n和t不变）

1. **开发中遇到的问题及解决方案**

（按时间顺序排列）

1. 在绘制扇形方面遇到困难

解决方案：自己去好好看了easyx的文档，把各种函数都记住了，包括后面使用的圆角矩形

1. 在实现启动与停机流程方面遇到了函数问题

解决方案：自己算了几个函数

1. 在处理增加/减小推力按钮方面不知道怎么处理启动和停机过程

解决方案：询问之后发现只能在稳定状态使用推力按钮

1. 当稳定状态通过操作按钮导致n1<95时，再启动的时候不知道怎么确定参数

解决方案：将启动状态控制t和n的参数和真实时间分离开，并且这两个函数使用不同的自变量控制，当出现稳定状态需要再启动情况时，通过n和t的真实值反推两个函数分别的自变量并且带入（即程序中的real\_t与real\_t1）

1. 在模拟警报时，如果不改变当前状态直接把数值强加到变量上会导致很多bug

解决方案：写了一个初始化函数initialize()，模拟除了传感器报错以外的任何警报时都直接初始化程序再进行模拟

1. 在模拟传感器报错的时候没法正确维护报错传感器涵盖的各种变量

解决方案：在绘制数据方面下功夫，在检测到传感器报错的时候直接输出错误的表盘，这样不会改变变量的实际值，符合了实际情况

1. 在绘制表盘的时候出现闪烁问题

解决方案：直接使用双缓冲技术，完美解决问题

1. 警告存在几个部分（名称、警告信息、存在时间）很难处理

解决方案：使用结构体来组织警告的结构，包括警告代号、颜色、存在时间，同时使用枚举结构简化了警告信息内容和警告颜色

1. **智慧编程工具在此次开发中的作用**

首先我必须说一句，在这个需要设计ui和逻辑复杂的项目里，让ai写整个项目根本不可能。我在整个项目里面用的最多的应该是copilot的tab一键补全剩余代码，这个确实很好用。剩下的就是询问建议，比如我最开始就没想到要给警告建立一个结构体，这个点子是gpt告诉我的。还有一些比较重复性的工作比如绘图里面的画框，输出文本啥的，都是扔给gpt干（但是有的时候gpt也干不好，参数还得我自己来）。总的来说，感觉ai不擅长复杂项目，但是对于单一问题不管多难都有迎战的能力。（但是gpt不知道有的时候为啥会输出完全不能用或者过时的答案，比如我之前问gpt用easyx库怎么画圆弧结果gpt直接给我了一个已经被easyx淘汰的函数出来）

1. **心得体会**
2. 在进行一个大型项目的时候必须要有先后次序，一般是先前端后后端，否则会难以入手，加长作业时间
3. 在一个大型项目中，变量命名必须要有逻辑，绝对不能随意命名，就算是bool值都不行，因为写到最后会很混乱
4. 在程序中不要滥用bool值，会导致全局变量太多难以管理
5. 在需要处理数据时，一定要注意临时变量的使用，临时变量可以解决很多问题
6. 全局变量修改后一定要搞清楚在流程中全局变量都应该是什么情况，必须心中有数，才不会出现很多问题，同时在某些时间也必须注意对于全局变量的初始化
7. 当需要创建一个含有多种信息的内容时，结构体是很方便的方法
8. 在处理函数定义的时候，最好把函数定义的部分分开来摆放，这样易于寻找特定的函数，防止造成混乱
9. 注意if和else if的区别，这两种结构的效果完全不同
10. 尽量把大段的代码全部封装为函数，便于管理
11. 在写代码的时候，一定要注意各个流程之间的逻辑和顺序关系，有时调换两个函数执行的顺序会导致很大的改变。