**Vscode mingw下opencv及多线程C++socket—Apple的学习笔记**

目的：由于vslam都是用c++开发的，所以要多练习c++11，所以就自己设计项目来练手

立项：（适用于0.1版本）

名称：Display

功能：一个client将数据集中的png图片一张张读取，并且通过tcp socket协议传输给server端进行显示。

特殊控制功能：Server端的按键s是控制socket传输数据的开关，按键j是控制是否将当前传输的图片数据实时显示的开关。

工具链：coding用Vscode，编译用CMake+MinGw8.1（环境搭建可以参考我之前的文章）

数据集：在client端是读取png的数据集（我用的是TUM数据集，由于比较大无法上传，读者可以选用自己的数据集作为输入）

工程共享路径: <https://github.com/AppleCai/Display>

**一，0.1版基本设计说明：**

* 1. Server端：

1.1.1主进程下有2个线程—一个是key按键扫描，另一个是server的socket通信传输数据及imshow显示图片。

1.1.2添加了锁，用于按钮s键暂停socket传输的功能，此时会挂起server线程。当再次按s键后，在key线程中可以触发server线程唤醒

1.1.3由于有特殊控制功能，所以添加了策略模式的设计，可以选择不同的控制显示的方式。

1.2 client端：

用作配合server的测试code，忽略设计。

**二，后续更新版本说明：**

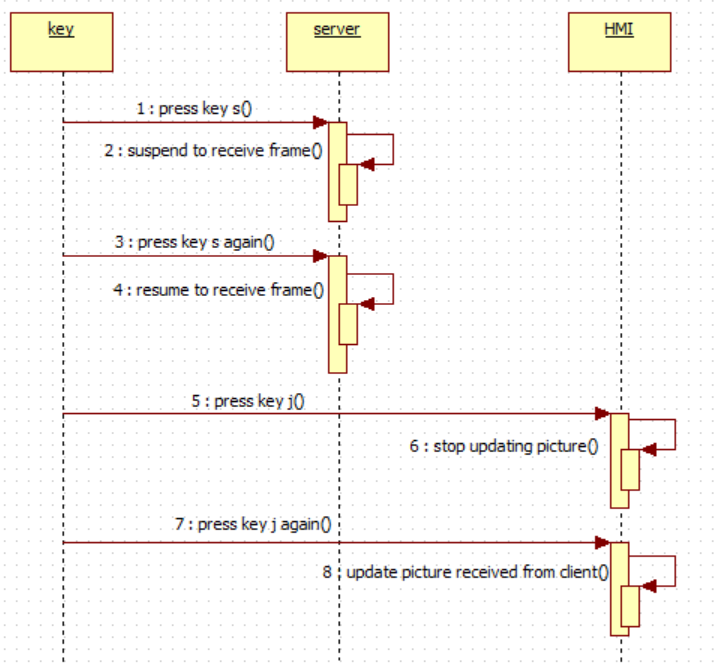
计划改进和添加的功能如下：

2.1优化client和server间的通信协议。添加crc校验及各类出错处理及退出。

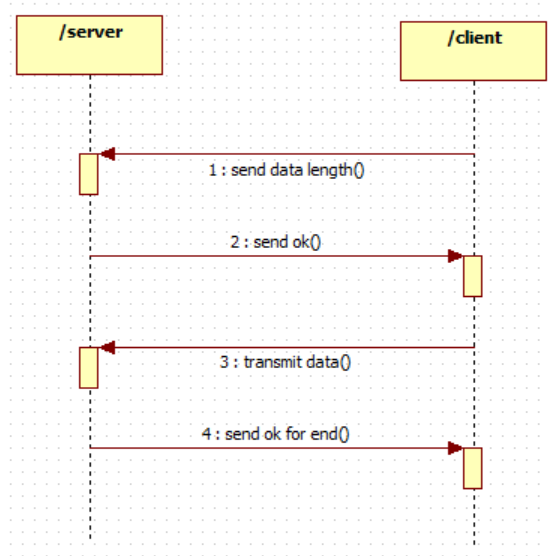
* 1. Socket添加心跳包监控及重连接尝试。
  2. HMI显示和socket传输分离为2个线程。
  3. HMI图片显示添加ROI区域显示张数及叠加特质logo。
  4. 添加对图片的监视作用，添加弹窗报警及logDTC记录。（目的是使用进程间通信D-bus和共享内存）
  5. Server端切换到linux上。
  6. 添加守护进程。
  7. 进行性能测试，进行优化。

**三，0.1版设计图**

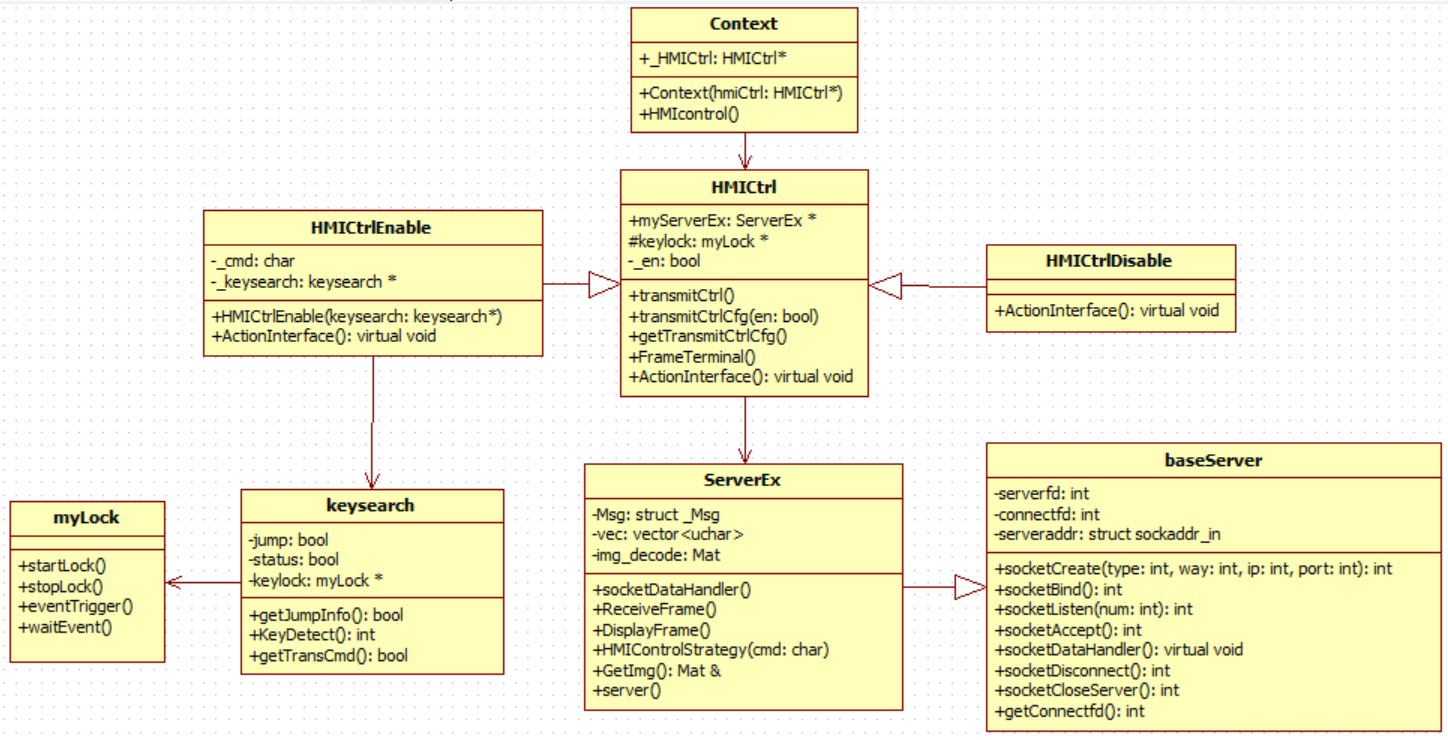
3.1 server端的3个对象关于按键的控制说明



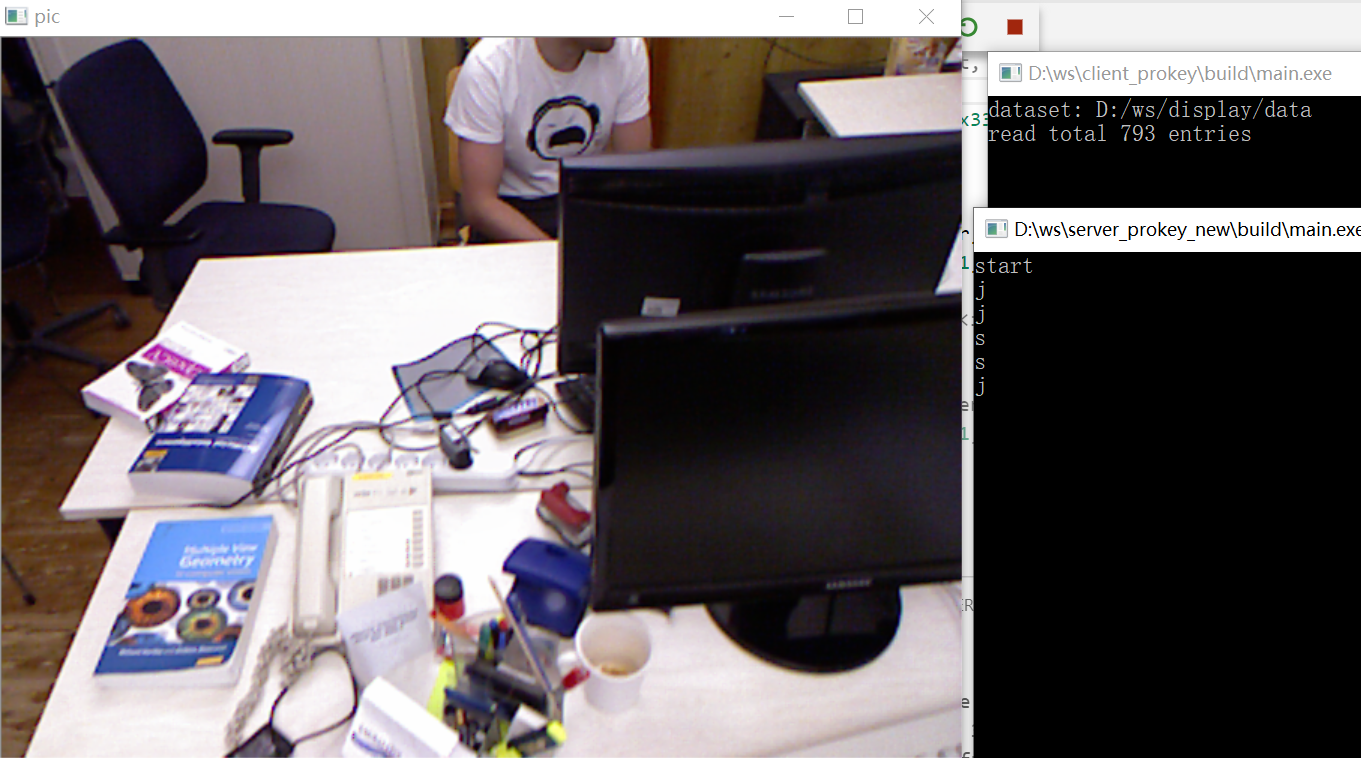
3.2 socket通信数据的应用协议说明



3.3 server端的类图



**四：运行效果**



**五：历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SN | Version | Author | Data | Description |
| 1 | 0.1 | Cai Ling | 2019/5/12 | initial version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |