HITCRT 视觉组竞培营代码编写规范

整理：方思安

2022 年 6 月 24 日

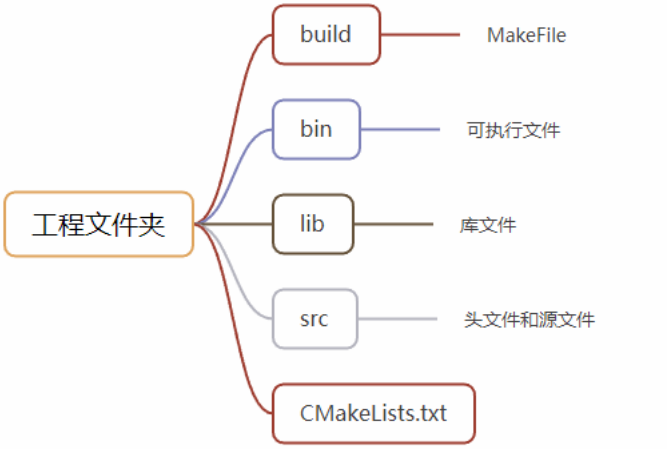
赠言

Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.

—John F. Woods, 1991, in a post to the comp.lang.c++ newsgroup

一、工程规范

1.[推荐]推荐工程目录结构如下：可按需添加其它文件夹。



（注：src 文件夹内不同模块放在不同的文件夹内，便于添加子模块）

2.[强制]所有文件采用 UTF-8 编码。

二、风格规范

1.[强制]使用 clang-format 格式化代码，风格统一为 Google，缩进宽度设为 4。

2.[强制]除 main.cpp 外其它源文件都有对应的同名头文件，用于申明接口，函数声明必须放在头文件中，实现必须放在源文件中。

3.[强制]缩进采用四个空格，若要使用 Tab 请先在编辑器里面将 Tab 设置为四个空格。

4.[强制]代码中禁止使用 using namespace cv 之类的全局命名空间，而是使用域解析符例如 cv::Mat; 这样的好处是可以避免名称冲突以及代码可读性强。

5.[强制]函数签名中形参顺序：输入在前，输出在后。

6.[强制]类的申明顺序依次为：public，protected，private。

7.[强制]除 for 等特殊语句，一行内不得有多条语句。

8.[强制]对 for，while，if，if…else…，if…else…if…等语句始终使用花括号，即使只有一条语句。

9.[强制]对于 if…else…，if…else…if…等语句如果语句中的一部分使用了花括号，那

么其他部分必须使用花括号。例如：

```
if()
  {
    …
  }
else
  …    //错误，前面使用了花括号 else 后面也应该使用花括号。
```

10．[强制]空的循环体必须使用"{}"而不是一个分号。例如：

```
while();   //错误，空循环体不能用分号表示。
while(){};  //正确，用"{}"表示空的循环体。
```

11．[强制]头文件必须加#pragma once 进行保护，用于替代#ifndef。

12．[推荐]向 vector 中增加元素之前，若已知元素个数，先 reserve 预留容量，减少 table grow cost。

13．[推荐]尽量避免使用基本数据类型，如数组，以 STL 等库中封装过的数据类型代替。

14．[推荐]尽量用智能指针(shared_ptr 或 unique_ptr)代替普通指针，免除内存泄漏等问题的困扰。

三、命名规范

1．[强制]对任何对象（文件、变量、函数）命名时，除循环变量等外，使用有意义的名字；除附录中收录的或极常用的缩写外，命名中使用完整的单词。

例如：int numErrors; //正确，使用有意义的命名

```
int n;        //错误，使用无意义的名字
```

2．[强制]C++文件以.cpp 结尾，头文件以.h 结尾。

3．[强制]主文件命名为 main.cpp。

4．[强制]文件名与类名相同使用大驼峰，例如：MyUsefulClass.cpp；文件夹名用小写加下划线，例如：aim_assist 。

5．[强制]代码中使用 hitcrt 作为命名空间名称。

6．[强制]宏常量字母全部大写，单词间以"_"隔开。

7．[强制]类，结构体，枚举名命名为大驼峰型，枚举值全大写并用"_"隔开，例如：

```
class UrlTable { ...};
enum UrlTableErrors
{
  OK = 0,
  ERROR_OUT_OF_MEMORY,
  ERROR_MALFORMED_INPUT,
};
```

8．[强制]普通函数和成员函数名均采用驼峰命名法，成员函数无需加前缀，例如：

```
getArmors();
setImageWidth();
```

9．[强制]除特例情况外，变量名一律采用驼峰命名法，类的成员变量以"m_"开头，静态变量以"s_"开头。

例：

```
smallArmor //局部变量
```

```
m_smallArmor //成员变量
s_smallArmor //静态变量
```

10. [强制]特例 1：全局变量和常量，变量名全大写，用下划线分割。

例：

```
int IMAGE_WIDTH;
const double GRAVITY_ACCELERATON = 9.8;
```

11. [强制]特例 2：从参数表中读入加载的静态变量，实际功能与全局变量一致的，也按照全大写、下划线分割形式命名。

例：

```
Param::GRAVITY_ACCELERATION
```

12. [强制]特例 3：若使用智能指针或封装过的类名中含有 Ptr 的类，变量名末尾加 Ptr；普通指针不加前缀/后缀。

例：

```
std::shared_ptr<hitcrt::Armor> armorPtr; //智能指针
std::shared_ptr<hitcrt::Armor> m_armorPtr; //智能指针成员变量
ICameraPtr cameraPtr; //含 Ptr 后缀的类名
char * fileName; //普通指针
```

13. [推荐]尽量少使用全局变量和外部变量，以静态变量代替。

14. [推荐]类型和变量名一般为名词：如 fileOpener、numErrors；函数名通常是指令性的，如 openFile()、setNumErrors()。

四、代码注释

1. [强制]使用 Doxygen 格式书写源文件注释，可用 vscode 或 clion 自动生成模板。

2. [强制]函数实现前必须有注释，格式如下：

```
/**
*@brief <简述>
*@param <参数说明>
*@return <返回值>
*@author <作者姓名>
*@mail <作者邮箱>
*/
```

3. [推荐]函数声明及全局变量声明前添加注释简要说明用途。

4. [推荐]在类定义前面加上注释说明类的功能和用法。

五、版本控制

1. [强制]git 备份周期：本地 git 每天都需要进行备份，码云 git 每星期一次，并在 README.md 中简要说明进度，指导会定期检查。本地 git 的 commit 次数不要太多，非代码或内容上的小修改（加减空格、删无用注释）不要反复 commit，否则 commit 太长不便于查找。

2. [强制]每次 commit 写清楚修改了什么，或达到了什么样的结果。

3. [强制].gitignore 中将编译产物(build 文件夹和 bin 文件夹)、大文件(.jpg 和.avi)和 IDE 生成的配置文件夹(.vscode 和.clion)忽略，禁止将大文件上传码云。