命名很重要能加强可读性，

数据类型：首字母大写，每个单词首字母大写

变量命名可以：前缀（作用域+（类型））+（类型）+名字

cocos2d-x中的类型的变量可以放在最后如：CCLabelTTF \*m\_pNickNameLabel

m\_, m\_p

常量以k开头

为便于界定，除前缀外，每个单词的首字母要大写

|  |  |
| --- | --- |
| 无 | 局部变量 |
| m\_ | 类的成员变量（member） |
| sm\_ | 类的静态成员变量（static member） |
| s\_ | 静态变量（static） |
| g\_ | 外部全局变量（global） |
| sg\_ | 静态全局变量（static global） |
| gg\_ | 进程或动态链接库间共享的全局变量（global global） |

|  |  |
| --- | --- |
| **前缀** | **说明** |
| st | 结构体（struct） |
| p | 指针型变量和迭代子（pointer） |
| r | 引用（reference），此前缀对于常引用（const reference）来说可以省略 |
| arr | 数组（array） |
| str | string字符串(std::string) |
| map | std::map |
| v | std::vector |
| b | 布尔型变量（bool） |
| f | 浮点型变量（float） |
| e | 枚举型变量（enumeration） |

**常量名**

* #define宏定义的所有常量单词字母大写，单词之间必须用下划线隔开，不允许出现非全球通用的缩写。
* Enum定义的常量，遵循cocos2d-x的命名规范，以k为前缀的匈牙利命名法，每个单词首字母大写即可。
* 其他const等方法定义的常量，同上一条。

**函数名**

* 函数名一般是一个动宾结构，从意义上来看是doSomething。第一个单词要小写，其他单词首字母大写。函数名不包含前缀。
* 全局或者static的成员函数的所有单词首字母大写。
* protected的成员函数要加\_前缀。
* private的成员函数要加\_\_前缀。

**继承**

* 继承除第一个基类不做限制，从第二个往后必须只能是接口。不允许继承多个非接口类，即使是继承接口，也不宜过多。如果一定要继承多个可实例化类，使用组合，或修改自己的设计。
* 继承的写法如下：

**class** LocalClickSprite

: **public** cocos2d::CCSprite

, **public** cocos2d::CCTargetedTouchDelegate

{

……

适当注释

尽量不要写超过100行以上的函数