# [Window Messages] 4

1.std::map和std::unordered\_map, std::map使用二叉搜索树，它存储的键值对总是按键（key）排序的（可定义operator() > , <等 ）。而std::unordered\_map不保证任何特定的顺序，元素的位置由键的哈希值决定,搜索时间为常数

2.仿函数operator() 跟构造函数的区别

构造函数为特殊成员函数 在对象被创建的时候调用  
Object b(10); 构造函数  
b(10); 调用operator()成员函数

1. constexpr是C++11引入的一个关键字，用于指明表达式或对象的值在编译时就可以确定。这样，编译器就可以在编译时计算出这些值，提高运行时的效率

# [WM\_CHAR / Mouse] 5

1. WM\_CHAR 里面相对（WM\_KEYDOWN）区分大小写,且WM\_CHAR 是由TranslateMessage()产生的

# [Window Framework] 6

1. Window窗口封装类

ChiliWin.h

1. 主要就是包含Window头文件 #include <Windows.h>  
   由于Window.h包含超多命名空间 可在导入前取消定义 如取消Windows里面的Max Min宏定义  
   #define NOMINMAX

定义Windows sdk版本信息

// target Windows 7 or later

#define \_WIN32\_WINNT 0x0601

#include <sdkddkver.h>

1. Window接口类

WindowClass 注册 以及 取消注册 WinClass

Private static WindowClass wndClass; 在Cpp里面定义时候就会调用构造函数注册WindowClass  
Window::WindowClass Window::WindowClass::wndClass;

1. Hanky-panky
2. 构造函数最后加noexcept 主要是为了提高性能

# [Custom Icons / Error Handling]7

# [Keyboard / WM\_SYSKEYDOWN]8

1. 
2. Server Clinet

Server 作为Windows ，Clinet作为我们的Game

创建Kbd类来管理 关于键盘的消息 通过API来通信 这样的设计更具模块化

1. unsigned char keycode

来自WM\_KEYDOWN message , wParam 其所有值都在 1 Byte以内，等于8bit   
2的8次方 所以在unsigned 的情况下是 0-256 , 否贼为-128 -127

1. Windows 为 Keyboard 友原类，Windows 可访问Keyboard 私有函数

Keyboard 是Windows embedded object 生命周期绑定

1. VK\_MENU 表示ALT 为系统按键 like F10

case WM\_KEYDOWN:

case WM\_SYSKEYDOWN:

# [Mouse / Mouse Capture]9

1. VM\_MOUSE里

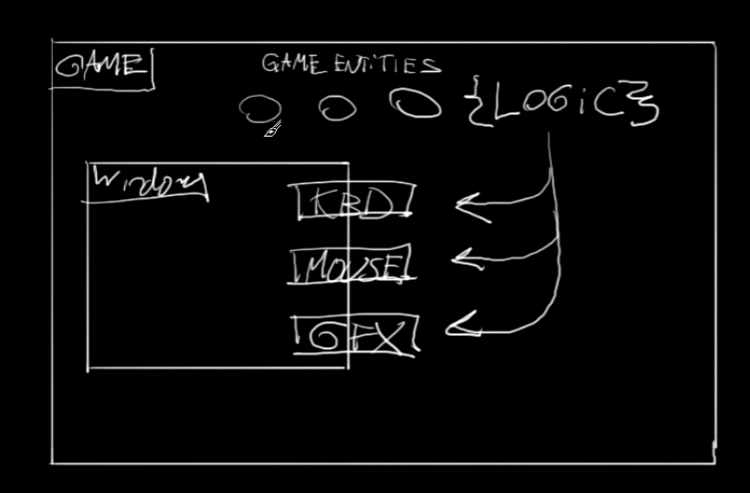
lParam表示 POINTS ; const POINTS pt = MAKEPOINTS(lParam);

wParam 表示鼠标中间键

GET\_WHEEL\_DELTA\_WPARAM(wParam) 大于表示WheelUp

1. kbd 跟 mouse Event的大致流程为  
   WinMain里面创建Window  
   Window不断获得消息，对应的bind到kbd 跟 mouse Event里面  
   WinMain主循环中通过wnd.kbd/wnd.mouse Read（）读出queue里面的Event根据  
   类型做出各种处理
2. 写入类为 ostringstream oss; oss.str();

# [App class / PeekMessage]10

1. 

创建一个APP类 来代表GameLogic

2. std::setprecision(1)<< std::fixed << t <<"s";  
std::setprecision(1) 浮点精度为1 3.1 std::fixed 固定输出精度 即使为0 两者一般连用

3.std::optional<> 可以返回一个不存在值 比如容器 返回optional<>类型也是OK 的不会冲突