2023年1月1日星期日

Iimgui事件

ImGuiIO& io = ImGui::GetIO();

Application& app = Application::Get();

io.DisplaySize = ImVec2(app.GetWindow().GetWidth(), app.GetWindow().GetHeight());

这段代码，imgui绑定了窗口

Imgui的事件必须窗口事件执行后，imgui才能捕捉

2022年12月31日星期六

窗口事件

1. Window.h文件

using EventCallbackFn = std::function<void(Event&)>;// 返回值是void的，参数是Event的调用形式

virtual void SetEventCallback(const EventCallbackFn& callback) = 0;

这里定义了EventCallbackFn 和SetEventCallback虚函数

2．WindowsWindow.h文件 windowwindow类继承Window

实现了

inline void SetEventCallback(const EventCallbackFn& callback)override { m\_Data.EventCallback = callback; }

并且有

struct WindowData {

std::string Title;

unsigned int Width, Height;

bool VSync;

EventCallbackFn EventCallback;

};

WindowData m\_Data;

将EventCallbackFn定义为一个属性

1. Application.h ---重点

std::unique\_ptr<Window> m\_Window;

指向Window类

在构造函数中，调用纯虚函数，设置

m\_Window->SetEventCallback(BIND\_EVENT\_FN(OnEvent));

EventCallbackFn绑定为OnEvent函数

而OnEvent函数是void返回值和Event参数，正好符合上面 EventCallbackFn 的function定义

void OnEvent(Event& e);

1. 在windowswindow.cpp中

WindowData m\_Data

glfwSetWindowUserPointer(m\_Window, &m\_Data); //设置了glfw的m\_data用户自定义的指针

然后在窗口事件获取

WindowData& data = \*(WindowData\*)glfwGetWindowUserPointer(window);

//

WindowCloseEvent event;

data.EventCallback(event);// 代表执行EventCallback函数，参数为event当前触发的事件，因为在3中OnEvent接受一个Event参数

每一个data. EventCallback(event); 意思是调用一次Application的OnEvent，OnEvent里面处理具体事件