

## ----- scene viewport -----

》》》做了两件事：设置视口和设置相机比例

》》》为什么要设置 m\_ViewportSize 为 glm::vec2 而不是 ImVec2 ?

因为后面需要进行 != 运算，而 ImVec2 没有这个运算符的定义，只有 glm::vec2 有这个运算符的定义。

```
template<typename T, qualifier Q>
GLM_FUNC_QUALIFIER GLM_CONSTEXPR bool operator!=(vec<2, T, Q> const& v1, vec<2, T, Q> const& v2)
{
    return !(v1 == v2);
}
```

所以需要 ImVec2 接收 GetContentRegionAvail 返回的 ImVec2 类型的 panelSize，然后将两者进行比较。

```
ImVec2 panelSize = ImGui::GetContentRegionAvail();
if (m_ViewportSize != *(glm::vec2*)&panelSize)
{
    // ...
}
```


》》》发现一个问题

```
ImVec2 panelSize = ImGui::GetContentRegionAvail(); // 获取面板大小
if (m_ViewportSize != *(glm::vec2*)&panelSize)
{
    m_ViewportSize = { panelSize.x, panelSize.y }; // 及时更新视口大小
    m_Framebuffer->Resize(m_ViewportSize.x, m_ViewportSize.y);
    m_CameraController.Resize(m_ViewportSize.x, m_ViewportSize.y);
}

ImTextureID textureID = (void*)m_Framebuffer->GetColorAttachmentRendererID();
ImGui::Image(textureID, ImVec2{ m_ViewportSize.x, m_ViewportSize.y }, ImVec2{ 0,1 }, ImVec2{ 1,0 });
```



其中，无论对 m\_Framebuffer 是否调用 Resize，其渲染结果和响应好像都是一样的，并没有什么影响（实际上这应该对图像的分辨率有一定影响，但为何我没有发现什么明确特征？）。而且不调用 Framebuffer->Resize 的话，调整窗口大小的时候图像并不会出现闪烁的现象。（所以说闪烁正是因为帧缓冲对纹理附件的刷新而导致的）


》》》另一个问题


**@KennyTutorials** 4年前

Im found a little bug when we double click on title bar / border, then engine is crashed and say that framebuffer isn't complete. Maybe this bug only on my engine? I think we just destroy the viewport window, but at same time trying drawing framebuffer on this window.

当我们双击标题栏/边框时，我发现了一个小错误，然后引擎崩溃并说帧缓冲区不完整。也许这个错误只出现在我的引擎上？我认为我们只是销毁视口窗口，但同时尝试在此窗口上绘制帧缓冲区。



 回复


 3 条回复

**@FlukierJupiter** 4年前

use the ImGuiTabBarFlags\_NoTooltip flag when creating the imgui window to prevent the window collapsing



创建 ImGui 窗口时使用 ImGuiTabBarFlags\_NoTooltip 标志以防止窗口折叠


 回复

**@FlukierJupiter** 4年前

you could just return from the invalidate function if the width or height is zero, before creating the frame buffer

如果宽度或高度为零，则在创建帧缓冲区之前，您可以从无效函数返回

 2 回复

**@KennyTutorials** 4年前

**@FlukierJupiter** Thanks for help! It works)

**@FlukierJupiter** 感谢您的帮助！有用)

》》》值得一提的是，相机的纵横比更新函数参数需要为 float 类型的，而不是 uint 类型，否则会导致窗口尺寸过小时无渲染结果。

```
void OrthoGraphicCameraController::Resize(float width, float height)
{
    // ...
    m_AspectRatio = width / height;
    UpdateViewport();
}
```

-----ImGui Layer Events-----  
» » » »