

22 游戏观测者——远程与调试器

游戏观测：

在实际开发中，我们可能会遇到各种各样的 bug，或者是遇上各种各样的优化问题。观测游戏运行时的状态将有效帮助我们解决这些问题。

.

一、远程

- 显示运行时场景树下的节点结构
- 显示节点相关的属性，可以进行修改与编辑。

二、调试

- 显示碰撞区域
- 显示寻路系统
- 等等

三、调试器

①错误

- 用红色文字提示代码书写错误，或反馈运行时发生的程序报错。
- 用黄色文字向开发者做出警告，被警告的代码一般是书写不够严谨的代码，他们不会直接破坏游戏的运行，具体的警告是否要处理，由代码编写者根据实际情况自行决定。
- 大部分情况下，错误的出现会破坏游戏的运行，使得游戏崩溃或出现图像和物理处理方面的 bug。

②性能分析器

- 以毫秒为单位，实时报告游戏中物理帧和空闲帧的间隔时间，以及物理、画面、音频、脚本处理所消耗的时间。

③visualprofiler

- 显示 CPU 与 GPU（处理器与显卡），进行画面渲染时每帧执行各项渲染功能时所消耗的时间。

④监视

- 显示游戏画面的刷新帧数，空闲帧的间隔时间和物理处理时间。
- 显示游戏的占用的内存，最大内存和消息队列最大占用内存。

- 显示游戏中的 Object、资源、节点和场景树之外的节点数目。
- 显示光栅、视频、物理、音频相关的状态情况。

⑤显存

- 显示游戏中各个资源占用的显存大小

⑥其他

- 帮助识别我们点击的 Control 类节点
- 可以显示此节点在场景树下的位置

⑦网络分析器

- 制作联网游戏时，显示 Godot 的流量数据。

四、断点与暂停

暂停：暂停运行中的项目，观测程序状态

断点：在脚本中设置断点，当代码运行到断点时进入暂停状态。进入断点后可以逐行运行代码，观察运行内部的状态。

第二部分结语：

- 第二部分的教程，主要是介绍了 Godot 运行的机制和 Godot 中节点以外的重要对象，这些知识，可以帮助我们更有条理的建立起 Godot 的框架。大部分人初学结束这部分的感受是云里雾里，这是正常的。
- 第三部分将会介绍具体的节点功能，这会比第二部分好理解很多，但是我建议我每介绍一个节点，大家自己去浏览一下这个节点的描述部分，并过目一下节点的属性和方法。