

20 游戏管理者——MainLoop 与 SceneTree

一、SceneTree（场景树）与 MainLoop（主循环）

①场景树的作用

- 场景树是游戏的管理者，它负责进行 Godot 的内置服务器与节点的沟通工作。
- 内置服务器是 Godot 中的各种模块，包括计时系统、物理模拟系统、图像绘制系统等。
- 节点是 Godot 中最基本最常用的开发组件。

②主循环与场景树的关系

- 场景树类继承自主循环类，场景树是在主循环的基础上，对节点管理进行了扩写。
- 虽然存在主循环类，但游戏运行时只存在场景树对象。

二、MainLoop 的功能

①程序启动后，程序一定会创建一个主循环对象（虽然这个对象一般情况是他的子类场景树），它包含了**初始化、空闲帧同步回调、物理帧同步回调等方法**。这个类中不包含与节点相关的具体操作，我们使用 Godot 制作游戏时也几乎不会遇到需要自行编写 MainLoop 的情况。

②空闲处理与物理处理（Idle Process 与 Physic Process）

- 物理处理：Godot 游戏程序中会内置一个物理服务器，用于处理游戏世界内的各种物理计算。在每一次**物理计算之前**，这个服务器都会给予主循环一次参与计算的机会，这就是主循环中的物理处理。默认情况下，物理服务器一秒内会进行 60 次运算，因此主循环中的物理处理也会每秒进行六十次处理。
- 空闲处理：在 Godot 游戏程序相对“空闲”（当某些内置服务器运行结束的时候）的时候执行此处理。引擎会尽可能快的利用空闲时间来绘制新的游戏图像。

③节点中的空闲处理与物理处理

- `_process(delta)` : 空闲帧回调函数 `delta` 表示距上次空闲帧调用的时间
- `_physics_process(delta)` : 物理帧回调函数 `delta` 表示距上次物理帧调用的时间
- `call(函数名)` : 立即调用函数
- `call_defer(函数名)` : 到下个空闲帧调用函数

三、SceneTree 的功能

一、场景树

- 场景树继承自 Mainloop。
- 游戏运行时，将生成一个场景树对象，这个场景树对象将控制游戏中节点

二、场景树的功能

节点内部的 `get_tree()` 函数可以获取场景树

①组（Group）功能

- 组：将若干节点归纳为一组节点，使用场景树命令同组节点调用函数或修改属性。
- 节点的组操作：使得节点加入 / 退出组、判断此节点是否处于某小组内
- 场景树的组操作：命令同一组内的节点调用函数 | 判断某组是否存在 | 获取某组内的所有节点

②游戏暂停

- 当场景树进入暂停后，节点会根据自己的暂停模式来调节自己的状态。
- 节点的暂停模式：继承 | 正常暂停 | 暂停时运行 | 总是运行 | 从不运行
- 场景树暂停操作：修改 `pause` 属性

④场景树的其他功能

- 游戏退出
- 创建计时器
- 创建 Tween 动画
- 获取节点总数
- 设置网络连接
- 等等

三、场景树与 viewport 节点

当游戏开始加载主场景时，场景树会创建一个 viewport 节点，将主场景的根节点加载为 viewport 节点的子节点。下节课的内容将围绕 viewport 节点来展开介绍。