RflySim3D 控制台命令接口实验原理

1. 文件目录

例程目录：[安装目录]\RflySimAPIs\3.RflySim3DUE\0.ApiExps\e2_CommandAPI\

序号	实验名称	简介	文件地址
1	三维场景交互接口 RflySim3D 控制台命令接口实验	通过 RflySim3D 控制台命令接口对场景中的不 同对象进行实时控制的方法。	1.command\Readme.pdf
2	三维场景交互接口 RflySim3D 内置场景模型导览实验	列举平台可以选择的三维模型和场景，如何使用导出的列举数据。	2.enumerate\Readme.pdf

2. 总体说明

交互设计的需求

控制台命令这种人机交互设计的起源可以追溯到早期的计算机系统，当时图形用户界面（GUI）还不普及，用户需要通过文本命令来与系统交互。随着技术的发展，GUI 逐渐成为主流的用户界面，但是控制台命令依然保留了一些优势，如：

- 灵活性：控制台命令可以直接修改和查询系统的各种参数和状态，而不需要通过菜单或按钮等图形元素。

- 高效性：控制台命令可以一次执行多个操作，而不需要多次点击或拖动鼠标等输入设备。

- 兼容性：控制台命令可以在不同的平台和设备上运行，无需考虑屏幕分辨率、颜色深度、输入方式等因素。

因此，控制台命令这种设计被广泛应用于游戏和仿真等领域，作为一种与用户界面相辅相成的开发和调试工具。

虚幻引擎中的控制台命令是一种强大的工具，可以让用户在运行时对游戏或应用进行控制、调试和优化。控制台命令可以通过键盘输入或脚本执行，可以方便地查看、添加、删除和修改，可以根据不同的需求和场景进行灵活的使用。主要应用包括以下几个方面：

- 调试和测试：控制台命令可以帮助开发者在运行时检查和修改游戏或应用的各种状态，如物理、渲染、声音、输入、网络等。通过控制台命令，可以快速地定位和解决问题，或者模拟不同的情况进行测试。

- 优化和分析：控制台命令可以显示和调整游戏或应用的性能参数，如帧率、内存占用、CPU 和 GPU 使用率、对象数量等。通过控制台命令，可以监测和优化游戏或应用的运行效率，或者分析其资源消耗。

- 交互和控制：控制台命令可以让用户在运行时改变游戏或应用的行为和表现，如视角、画面效果、操作方式、难度等。通过控制台命令，可以增加游戏或应用的可玩性和灵

活性，或者适应不同的需求和喜好。

- 扩展和创造：控制台命令可以实现一些额外的功能和效果，如生成和销毁对象、播放动画和声音、执行脚本和命令等。通过控制台命令，可以扩展和创造游戏或应用的内容和体验，或者实现一些特殊的仿真目的。

虚幻引擎控制台的工作流

首先，要实现控制台命令，需要有一个控制台窗口，也就是用户可以输入和查看命令的界面。虚幻引擎中，控制台窗口是由 SConsoleWidget 类定义的，它是一个继承自 SCompoundWidget 的自定义控件，包含了一个文本输入框、一个文本输出框和一个候选列表。用户可以通过按下特定的按键（如~或 Tab）来打开或关闭控制台窗口，也可以通过上下箭头来浏览历史命令或候选命令。

其次，要实现控制台命令，需要有一个控制台管理器，也就是负责注册、存储、查找和执行命令的核心类。虚幻引擎中，控制台管理器是由 IConsoleManager 接口定义的，它是一个单例类，可以通过 GEngine->GetConsoleManager()来获取实例。控制台管理器维护了一个 TMap 类型的变量，用来存储所有注册的控制台对象，其中键是控制台对象的名称，值是控制台对象的指针。控制台对象是由 IConsoleObject 接口定义的，它是所有控制台命令、变量和别名的基类，包含了一些公共的属性和方法。控制台对象有三种派生类：IConsoleCommand、IConsoleVariable 和 IConsoleAlias，分别对应了控制台命令、变量和别名的实现。

最后，要实现控制台命令，需要有一个控制台代理，也就是负责处理控制台窗口的输入和输出的辅助类。虚幻引擎中，>控制台代理是由 FConsoleManager 类定义的，它是一个实现了 IConsoleManager 接口的类，同时也继承了 FOutputDevice 类。控制台代理重写了 FOutputDevice 类的 Serialize 方法，用来向控制台窗口的文本输出框打印信息。控制台代理还有一个 Exec 方法，用来解析和执行控制台窗口的文本输入框中的命令。Exec 方法会首先检查输入的字符串是否为空或以特殊字符开头，如果是则直接返回。然后，它会将输入的字符串分割成多个子字符串，按照空格或分号进行分割。对于每个子字符串，它会调用 ProcessUserInput 方法，该方法会首先尝试从控制台管理器中查找与子字符串匹配的控制台对象，如果找到了，则根据控制台对象的类型，执行相应的操作。例如，如果控制台对象是一个命令，则调用其 Execute 方法，如果控制台对象是一个变量，则调用其 Set 方法，如果控制台对象是一个别名，则递归地调用 Exec 方法。如果没有找到匹配的控制台对象，则调用 FWorldDelegates::OnConsoleCommand 方法，该方法会遍历所有的世界代理，并尝试在其内部执行子字符串。如果仍然没有成功地执行子字符串，则打印一条错误信息。

虚幻引擎的控制台变量[4]

控制台变量 可用于保存一些状态信息，可通过控制台进行修改或查看。游戏内控制台还支持自动完成和列举由控制台管理器注册的控制台命令和控制台变量（控制台命令 Help

或 DumpConsoleVariables）。因此需要避免使用老旧的 Exec 接口。中间点中的控制台管理器将控制所有内容和更多内容（如用户输入历史）。

优先级是一个 32 位的整数，用来标识不同来源的控制台变量设置的相对顺序。优先级越高，表示该设置越晚被应用，也就越可能覆盖之前的设置。例如，如果一个控制台变量被 consolevariables.ini 设置为 10，然后在命令行上被设置为 20，那么最终的值将是 20，因为命令行的优先级高于 consolevariables.ini。优先级的设定基于以下原则：

- 控制台变量的初始值（由构造函数指定）应该是最低的优先级，这样可以让其他来源的设置生效。

- 预定义的配置文件（如 Scalability.ini，SystemSettings.ini，DeviceProfiles.ini 等）应该具有中等的优先级，这样可以让用户根据不同的平台和性能需求进行调整。

- 用户自定义的配置文件（如 consolevariables.ini，ProjectSettings.ini 等）应该具有较高的优先级，这样可以让用户覆盖预定义的设置，实现更个性化的控制。

- 命令行参数应该具有最高的优先级，这样可以让用户在启动游戏时临时修改某些设置，或者测试不同的参数的效果。

- 代码中的设置（如通过代码调用的 IConsoleManager::SetConsoleVariable()）应该具有比命令行略低的优先级，以便在特殊情况下强制修改某些设置，但同时也允许用户通过命令行进行覆盖。

- 控制台输入（无论是编辑器中的控制台窗口，还是游戏中的控制台键）应该具有最高的优先级，这样可以让用户在运行时实时地改变某些设置，或者恢复到默认值。

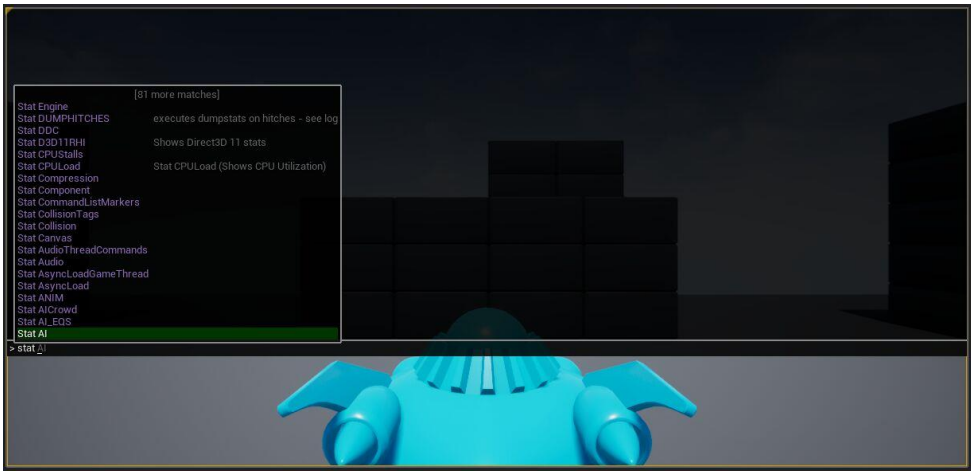
虚幻引擎的控制台命令

控制台命令是一条被发送到引擎的字符串，通常由用户在游戏内的控制台中输入。而引擎能以某种方式辨识并进行回应（如控制台/日志响应，修改内部状态等）。控制台变量是一种特殊的控制台命令，用于存储一些值，可以通过控制台进行修改或查看。

IConsoleManager 是一个接口，允许对控制台命令进行访问和操作。IConsoleManager 有多个方法来创建、注册、注销、查找和执行控制台命令。例如，RegisterConsoleCommand 方法可以创建一个控制台命令，指定其名称、描述和委托函数，用于定义当命令被执行时要做什么。FindConsoleVariable 方法可以根据名称返回一个指向控制台变量的指针。ProcessUserConsoleInput 方法可以接受一个字符串，并将其作为控制台命令执行。要使用 IConsoleManager，需要包含"ConsoleManager.h"头文件，并通过调用 IConsoleManager::Get()获取一个全局控制台管理器对象的引用。然后，可以在这个对象上调用控制台管理器的方法。

虚幻场景编辑器的控制台窗口[3]

PIE 控制台（PIE Console） 是一种游戏内控制台，允许你输入命令并显示性能数据，启用或禁用引擎功能，或者执行其他操作。



RflySim3D 自定义全局函数

RflySim3D 内置的全局函数是一些可以在控制台终端中输入或者通过代码调用的功能命令，用于实现 RflySim3D 相关的各种操作。下面以 RflyReqObjData 为例进行简要的解释：

- RflyReqObjData(int opFlag, FString objName, FString colorStr)：请求场景中某个物体的数据，opFlag 是一个整数，表示请求的操作类型，objName 是一个字符串，表示物体的名称，colorStr 是一个字符串，表示物体的颜色。该函数可以用于获取场景中物体的位置、姿态、速度、加速度等信息，并且可以用不同的颜色标记物体。支持的操作类型有：

- 0: 请求物体的位置（单位：m）

- 1: 请求物体的姿态（单位：度）

- 2: 请求物体的速度（单位：m/s）

- 3: 请求物体的加速度（单位：m/s^2）

- 4: 请求物体的角速度（单位：rad/s）

- 5: 请求物体的角加速度（单位：rad/s^2）

- 6: 取消请求

3. 关键功能的实现

RflySim3D 内置了一些全局的命令，可以完成 RflySim3D 相关的大部分功能，在 RflySim3D 的程序窗口中按下键盘的波浪号（~），可以打开控制台终端，这里可以触发 UE 引擎本身的一些内置命令，还可以触发 RflySim3D 平台内置的命令。这些命令相当于一个个的全局函数。

其中，RflySim 内置命令提供了对不同场景对象、相机、地图等进行操作和调整的功能。主要包括屏幕提示信息显示、地图/视角切换、模型/动画预览及控制、场景中各类数据回传等。

而 UE 引擎自带命令用于监测并平衡各种渲染参数以优化仿真性能。这里性能优化的核心原则包含两点：渲染效率和计算性能。计算性能：减少 CPU 计算次数，把能放在 GPU 中计算的部分放在 GPU 中并行计算；渲染效率：减少每一帧的绘制次数。限制帧率、调

整后处理质量、阴影质量等参数，可以在渲染效率方面平衡画面质量和性能需求。同时，通过修改仿真的运行速度和监测各进程的反馈时间，可以评估和优化计算性能。

- 完整的 RflySim3D 控制台命令调用方法参见[1] 5、命令行控制接口
- 详细 RflySim3D 控制台命令调用示例参见 Readme.pdf

4. 相关文献

[1]. ..\..\API.pdf

[2]. IConsoleManager | Unreal Engine Documentation

[3]. 运行和模拟 | 虚幻引擎 4.26 文档 (unrealengine.com)

[4]. 虚幻引擎 C++中的控制台变量 | 虚幻引擎 5.4 文档 | Epic Developer Community (epicgames.com)

[5].

附加资源

官方文档：RflySim 官方文档：https://rflysim.com/doc/zh/

社区交流：加入 RflySim 技术交流群：951534390