

网络多媒体交互式娱乐3D游戏引擎项目开发大纲

[日期：2006-10-30]

来源：www.mscenter.edu.cn/prj/files/310/ 作者：

[字体：大 中 小]

网络多媒体交互式娱乐3D游戏引擎项目总纲

Online Multimedia Interactive Entertainment 3D Game Engine Project Report

一、 前言

如今我国的网络游戏业已经进入了一个发展的高峰期，随之而来的是大量的网络游戏的投入制作。然而目前我国游戏制作的现状是十分不尽如人意的，其中一点就体现在游戏引擎的制作上。我们通过业内人士了解到现在我国拥有自主知识产权的游戏引擎的游戏制作公司仅有三家。而面对即将来临的网络游戏大潮，这种情形对本土游戏业的发展是十分不利的。

因此我们希望自主开发一个游戏引擎，并为此做了大量的准备。在学校的技术力量以及我们希望取得的微软公司的技术经费的支持下，我们有信心做好这个项目。

二、 需求分析

首先该引擎是只针对客户端的。虽然网络游戏既要有客户端，又要有服务器端，但是由于服务器端的程序与客户端截然不同，而且服务器和客户主要是交换数据，只要该引擎只要拥有接收和发送数据的能力，便可以在很大程度上与服务器兼容，由此我们完全可以将其作为独立的项目来开发。

作为网络游戏的引擎，首先要包括传统的单机游戏的元素，如画面、声音、控制。然后要有网络通信的元素。在可能的情况下可以提供简单游戏逻辑这样更上层的功能。对于引擎性能的要求，由于网络游戏传输速率的问题，所以不可能像单机游戏一样流畅，但在一般情况下至少不能出现明显的停顿和延迟的现象。关于引擎向上的接口，应该做到功能强大而又简洁明快，以便于游戏设计者的使用。

三、 总体设计

该引擎主要可以分为如下几个独立的模块：**System**（系统）、**Console**（控制台）、**Support**（支持）、**Render Core**（渲染核心）、**Interface**（接口）、**Tools/Data**（工具/数据）、**Communication**（通信）以及**Simple Game Logic**（简单游戏逻辑/暂定）。

System(系统)是引擎与机器本身做通信交互的部件。系统(system)是引擎与机器本身做通信交互的部件。在平台移植时，系统则是唯一需要做主要更改（扩加代码）的地方。系统又分为若干个子系统，其中包括：**Graphics**（图形）、**Input**（输入）、**Sound**（声音）、**Timer**（记时器）、**Configuration**（配置）。主系统负责初始化、更新、以及关闭所有的子系统。

Graphics Sub-System（图形子系统）是利用**Direct Graphics**或是**software rendering**（软件渲染）实现。或

是把这些API给予支持，然后抽象出一个“图形层”并将其置与实现API之上，这将给了客户开发人员或是玩家更多的选择，以获取最好的兼容性、最佳的表现效果。

Input Sub-System（输入子系统）需要把各种不同输入装置（键盘、鼠标、游戏手柄[Joystick]）的输入触发做统一的控制接收处理（透明处理）。用户与玩家可以非常自由地切换输入装置，通过不同的输入装置来获取统一的行为将变的很容易。

sound system（声音子系统）该子系统功能非常简洁明了，就是负责载入、播放声音。

3D游戏引擎中很多出色的表现都是基于**Timer sub-system**（时间子系统）的。（游戏里）任何东西都是通过时间触发来做移动变化。

Configuration（配置系统）位于所有子系统的顶端。它负责读取配置记录文件，命令行参数，或是实现修改设置。在系统初始化以及运行期间，所有子系统都将一直与它保持通讯。切换图象解析度，色深，定义按键（key bindings），声音支持选项，甚至包括载入游戏，该系统将这些实现显得格外的简单与方便。玩家与开发人员也能很方便地通过该子系统选择他们喜欢的运行方式。

Console（控制台）在运行时改变引擎的设置，而不需要重启。开发期间输出调试信息它非常的有效。引擎在运行期间，一旦发现了一个错误，不必立即退出程序；通过控制台，可以做些非常轻便的控制，并将这个错误信息打印出来。假如不希望最终用户看见或是使用该控制台，也可以非常方便地将其disable。

Support（支持）在引擎中任何地方都将被使用到。该系统包含了引擎中数学成分，存储管理器，文件载入器，数据容器等基础与底层。它将复用到更多别的相关模块中去。

Render Core（渲染核心）负责管理、处理和显示图形图像及相关资源。

重要的是将渲染器组件制作得基化（based）与干净（clean）。

首先可以确定的是将拥有不同的模块来完成不同的任务，渲染器可拆分为以下几个部份：可见裁减（Visibility）、碰撞检测与反馈（Collision Detection and Response）、摄像机（Camera）、静态几何体（Static Geometry）、动态几何体（Dynamic Geometry）、粒子系统（Particle Systems）、布告板（Billboarding）、网格（Meshes）、天空体（Skybox）、光线（Lighting）、雾（Fogging）、节点着色（Vertex Shading）和输出（Output）。

其中每一个部分都得需要一个接口来方便地实现改变设置（settings）、位置（position）、方向（orientation）、以及其他可能与系统相关的属性配置。

Interface（接口）作为引擎与游戏制作之间的控制介质，能使代码设计变得更清晰，应用起来也会更舒服。它能使游戏引擎具有非常好重用性，通过设计架构游戏逻辑（game logic）的脚本语言（scripting language）也能使开发变的更方便，也可以将游戏代码置入库中。

Tools/Data（工具/数据）。如3d模型编辑器，关卡编辑器，另外如加工游戏数据的工具。

Communication（通信）用于与服务器交换数据。支持游戏协议的封装，游戏数据的打包、解包，以及利用套接口发送和接收数据。

阅读：次
录入：pangmingyong

【推荐】 【打印】

相关新闻

本文评论 [全部评论](#)

发表评论



- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
- 本站管理人员有权保留或删除其管辖留言中的任意内容
- 本站有权在网站内转载或引用您的评论
- 参与本评论即表明您已经阅读并接受上述条款

点评：

姓名：

发表

[加入收藏](#) | [网站地图](#) | [网站搜索](#) | [关于我们](#) | [联系我们](#)

免责声明：本站所有资源均仅供从事教育游戏研究使用，如果侵犯了你的权益，请通知我们更正。

联系邮箱：eegamennu@yahoo.com.cn

南京师范大学教育游戏研究中心 Copyright © 2006 - 2007

中国教育技术协会信息技术教育专业委员会支持

Powered by [DET,NJNU](#)