

Unity 引擎下多人在线网络游戏的设计与开发^①



刘贤梅, 刘俊, 贾迪

(东北石油大学 计算机与信息技术学院, 大庆 163311)

通讯作者: 刘俊, E-mail: 1633209750@qq.com

摘要: 游戏数字产业在数字经济发展中扮演着重要的角色, 目前对于单机游戏的设计与开发比较多, 多人在线 3D 网络游戏涉及的相对较少. 本文结合网络通信原理, 以 Unity3D 为基础, 采用 MVC 架构, 设计并实现了多人在线动作角色扮演游戏 (ARPG). 针对在线玩家场景的一致性, 提出了位置同步、武器同步、伤害同步等通信协议. 游戏界面是否友好, 很大程度上影响玩家的用户体验, 同时也影响着游戏的整体进度, 对此设计出一种通用的 UI 框架. 由于存在网络延迟, 系统很难做到实时的、精确的玩家位置同步, 因此本文采用预测式位置同步方法, 有效减少了游戏运行时的延迟误差.

关键词: 网络游戏; 通信协议; UI 框架; 预测式位置同步

引用格式: 刘贤梅, 刘俊, 贾迪. Unity 引擎下多人在线网络游戏的设计与开发. 计算机系统应用, 2020, 29(5): 103-109. <http://www.c-s-a.org.cn/1003-3254/7407.html>

Design and Development of Multiplayer Online Games Based on Unity Engine

LIU Xian-Mei, LIU Jun, JIA Di

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163311, China)

Abstract: The game digital industry plays an important role in the development of the digital economy. At present, there are many designs and developments for stand-alone games, but relatively few for multiplayer 3D online games. Combined with the principle of network communication, based on Unity3D, and using MVC architecture, this study designs and implements multiplayer online Action Role-Playing Game (ARPG). For the consistency of online player scenes, Communication protocol, such as position synchronization, weapon synchronization, damage synchronization, etc., has been proposed. Whether the game interface is friendly or not, would greatly affect the user experience of the player, and also affect the overall progress of the game. A general UI framework is designed for it. Due to the network delay, the system is difficult to achieve real-time and accuracy. The player's position is synchronized, so this study uses the predictive position synchronization method, which effectively reduces the delay error during game running.

Key words: online game; communication protocol; UI framework; predictive location synchronization

随着虚拟现实技术和数字媒体技术的不断发展, 游戏产业的关注度也越来越高, 数字游戏经济在整体经济中正占据着一定的比重. 目前开发游戏的引擎平台很多, 而 Unity 因其所具有的跨平台性、易操作

性、性能优良等特点受到了市场的青睐^[1]. 本文利用 Unity3D 游戏开发引擎设计了一款多人在线对战游戏. 其中包括行走控制、人工智能系统、界面系统、服务端框架、客户端框架、房间系统、战场系统等多项内

① 基金项目: 黑龙江省高等教育教学改革研究项目 (SJGY20180079、SJGZ20180010); 东北石油大学研究生教育创新工程项目 (JYCX_07_2018)

Foundation item: Higher Education Teaching Reform Research Program of Heilongjiang Province (SJGY20180079, SJGZ20180010); Innovative Engineering Project of Graduates Education of Northeast Petroleum University (JYCX_07_2018)

收稿时间: 2019-10-11; 修改时间: 2019-11-07; 采用时间: 2019-11-20; csa 在线出版时间: 2020-05-07

容. 多个玩家可以通过在线注册、在线登录、房间选择等流程进入游戏, 最终体验在线 3D 网络游戏的娱乐.

目前, 游戏产业正从传统的 2D/2.5D 游戏、单机游戏向新兴的 3D 游戏和在线网络游戏发展. 而 3D 游戏和网络游戏的兴起就伴随着技术和算法的改变. 传统的 2D 路径搜索方式已经不能适用于 3D 场景, 因此需要改变或重新设计相应的寻路算法. 游戏中的界面系统在游戏中占据了重要的地位, 目前的 UI 设计与交互费时费力, 影响游戏的开发进度, 急需设计出一种 UI 框架来减少界面系统的开发周期. 对于网络游戏而言, 最重要的就是玩家场景的一致性和实时性. 一致性就需要建立对应的网络通信协议, 而实时性就要减少网络传书中的延迟误差.

针对上述问题, 本文使用 Visual Studio 2017 开发平台搭建服务器, 实时进行数据传输与交换; 采用 Unity 引擎完成客户端的开发, 使用 MySQL 数据库存储用户数据. 通过通信协议的搭建、UI 框架的设计、预测式位置同步方法的创建、NPC 的应用等完成了整个游戏的开发, 满足了玩家的娱乐心理.

1 游戏设计

1.1 游戏背景

本游戏是一款西式魔幻风格的动作角色扮演游戏 (ARPG), 主要以传说系列、伊苏系列等经典 RPG 游戏的感觉和理念而创作. 游戏故事主要讲述的是两个部落的种族人民为了各自的利益, 寻找陆地资源, 在泰雷诺亚大陆上拼杀、抢夺资源的故事.

1.2 游戏功能

为了能够满足玩家的娱乐心理, 同时也使游戏充满趣味性, 所以设计了如下功能:

1) 注册登录: 玩家首先需要进行玩家注册, 录入玩家信息后系统会自动将信息添加到数据库中. 注册成功后, 在登录界面输入玩家信息进入游戏.

2) 房间创建和加入: 玩家登录成功后, 可以加入已经创建的房间中, 通过点击加入按钮加入, 若房间人数已满, 则加入失败; 反之加入成功, 等待游戏开始. 玩家也可以自己创建房间, 等待其他玩家的加入.

3) 游戏运行: 游戏运行的过程中, 场景里除了玩家之外, 还加入了怪物 NPC. NPC 会有自身的一套行为,

它们会在固定的路线上进行巡逻、攻击、躲避等行为, 增强游戏的乐趣.

4) 玩家控制: 玩家在游戏中通过鼠标控制角色的移动, 或者通过鼠标点击的方式让角色自动寻路. 在漫游的过程中捡取补给、增添装备、强化技能. 当遇见敌方玩家或者怪物时, 释放不同的角色技能来消灭对方.

5) 特效、音效: 玩家攻击、释放技能、死亡时, 系统会给出相应的特效或者音效, 使游戏的画面感和代入感增强.

1.3 系统整体架构

图 1 描述了本文所提出的游戏系统结构, 多人在线网络游戏系统包含客户端和服务端, 客户端包含接口、数据、界面、资源、网络、房间和战斗管理系统. 服务器端处理游戏运行时产生的数据, 包括玩家数据、房间数据和战场数据等, 同时对网络通信进行管理. 游戏支持手机、PC 等多种运行终端, 并且具有良好的自适应性.

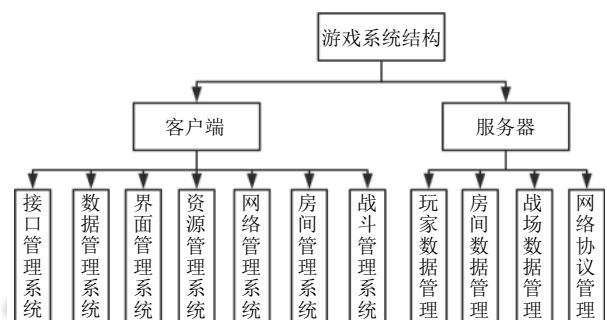


图 1 游戏系统结构图

系统经过人物建模和场景建模后进行场景集成, 将集成好的场景文件导入到 Unity3D 中, 完成游戏场景的搭建. 之后完成数据库的设计, 服务器、客户端的制作. 各个模块相互独立、耦合度低, 便于后期的维护, 能够提升软件开发的效率.

1.4 游戏设计难点

本文在线网络游戏设计的难点在于游戏服务器端的设计, 服务器系统架构图如图 2 所示. 服务器端包含一个总服务器和多个分组服务器, 系统为每一个创建的游戏房间分配一个分组服务器, 它负责当前房间的消息处理和转发. 总服务器负责玩家的登录注册以及分组服务器的管理.

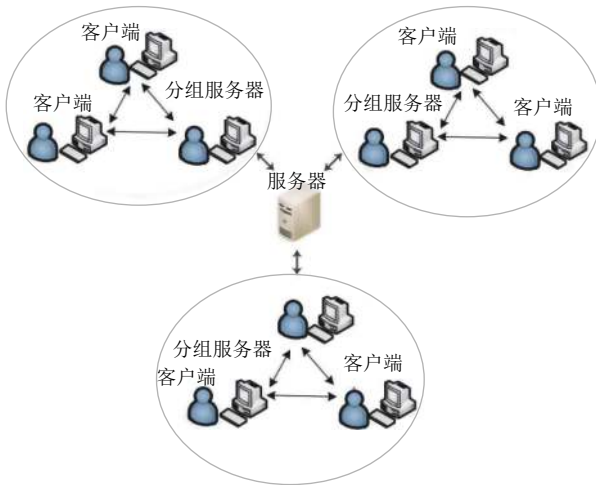


图2 服务器系统结构图

2 关键技术及系统实现

2.1 网络通信协议构建

通信协议是指双方实体完成通信或服务所必须遵循的规则和约定。通过通信信道和设备互连起来的多个不同地理位置的数据通信系统,要使其能协同工作实现信息交换和资源共享,它们之间必须具有共同的语言。交流什么、怎样交流及何时交流,都必须遵循某种互相都能接受的规则。通信协议的构建可以让通信双方通过制定好的规则来进行数据的传输。在网络游戏的设计中,网络协议是最重要的,是整个游戏的核心。客户端之间、客户端与服务器之间通过协议完成消息的传输,保证游戏场景的一致性。

为了提高开发效率及控制成本,设计一种便捷、清晰、高效的协议格式。本文设计了两种通信协议格式,图3是客户端向服务器发送的协议格式,图4是服务器向客户端发送的协议格式。

Length	RequestCode	ActionCode	Data
--------	-------------	------------	------

图3 客户端向服务器发送的协议格式

Length	ActionCode	Data
--------	------------	------

图4 服务器向客户端发送的协议格式

协议设计的思想是按照游戏中客户端的请求类型进行设计的。游戏中大致有3种请求类型:用户类型、房间类型和战斗类型,也就是协议格式中的 RequestCode,

根据请求类型细分成具体的请求方式,也就是格式中的 ActionCode。Length 和 Data 分别代表协议长度和协议内容。这种协议的设计简明清晰,数据小、解析快,服务器端可快速解析并转发,从而提升游戏运行速率。

当客户端向服务器发送请求时,协议格式如图3所示。例如当进行登录请求时,需要将玩家的用户名(id)和密码(password)传送至服务器,服务器接收后以此作为凭据查询数据库检查用户名和密码是否存在。

当服务器向客户端返回请求结果时,协议格式如图4所示,ActionCode 与之前客户端发送请求中的 ActionCode 相对应,例如之前的登录请求,服务器首先解析协议,然后查询数据库,若能匹配到请求中的用户名和密码,则返回 string 类型的字符串 Success,表示成功,否则返回 Fail 表示失败。客户端收到服务器的返回结果时,首先解析协议,解析出协议中的各个字段,当解析到的结果为 Success 时,表示登录成功,进入到房间界面。否则,系统给出提示:登录失败。

系统是基于 Socket 的通信方式,协议的发送是调用 Socket 中的 Send 函数。图5是服务器和客户端通信的时序图。

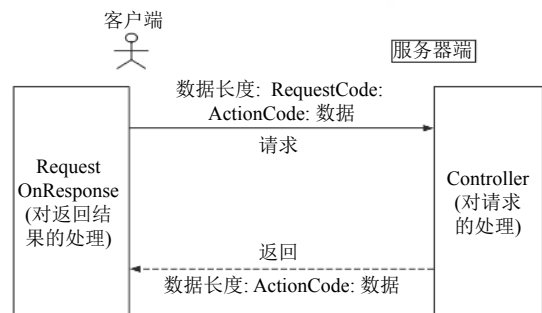


图5 服务器与客户端的通信时序图

2.2 UI 框架设计

UI 设计包括对游戏的人机交互、操作逻辑、界面美观的整体设计。好的 UI 设计不仅让游戏变得更有乐趣、更有品味,更让游戏的操作变得舒适简单、愉悦自由。因此设计出一种通用的 UI 框架对于提升游戏的开发进度有着很大的作用。本文的 UI 框架设计方案是:首先,每一个界面对应一个面板 (Panel), 每一个面板对应一个类。在这个类中编写面板的功能,再设计一个界面管理器,用它来控制界面的显示和关闭。界面管

理器中有两个基本方法: 分别是 Open 和 Close.

所有的面板类继承一个面板基类, 面板基类中有 OnInit、OnShow、OnClose 等 3 个固定的方法.

当调用界面管理器中的 Open 方法时, 界面管理器会依次调用对应面板中的 OnInit 和 OnShow. 在这两个函数中可以添加一些初始化方法, 包括按钮事件监听、网络消息监听等. 在调用的 Close 方法时, 界面管理器会调用面板类的 OnClose 方法, 在 OnClose 可以

添加一些资源释放的功能, 例如取消网络消息监听. 在 OnInit 方法中设置 skinPath 和 layer, skinPath 代表皮肤路径, 系统会根据该路径动态加载 Resources 中的界面资源. layer 代表层级, 可以设置面板的优先级, 确定面板的上下层级关系. 例如提示面板会显示在其他功能面板之上, 应该设置为最高优先级.

界面管理器 PanelManager 和面板基类的逻辑结构如图 6 和图 7 所示.

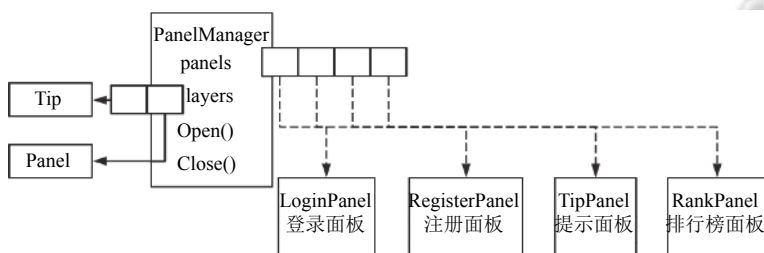


图 6 界面管理器逻辑结构图

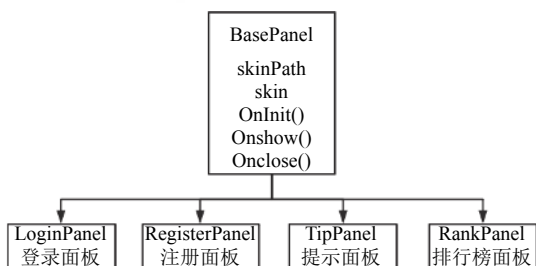


图 7 界面基类逻辑结构图

界面管理器中的 panels 是一个字典, 它会保存所有已经打开的面板, 以便阻止重复打开, 方便实现关闭功能.layers 是层级列表, 维护每个面板的层级.

每个面板通过动态加载资源, 然后进行实例化. 每个面板都有自己单独的面板类, 继承自面板基类, 面板基类如图 7 所示. 在 OnShow() 方法中进行组件的初始化, 例如找到登录面板上的登录按钮组件, 之后添加登录按钮的事件监听, 当玩家点击登录按钮时, 向服务器发送登录协议.

2.3 NPC 设计

NPC 的加入可以使游戏更加具有刺激性, 使玩家能够产生兴趣^[2]. 本文的 NPC 设计采用的是动画状态机. 目前游戏中加入了两种怪物, 为此系统根据怪物的特点, 攻击力的强弱, 属性值的差异、技能的多少设计

了两种动画状态机.

图 8 表示的是一般怪物—哥布林动画状态机, 因其能力较弱, 所以加入了较少的动画和攻击技能. 图 9 展示的是 Boss 怪物的状态机, 由于属性强, 给其加入了多种攻击技能和多种状态动画.

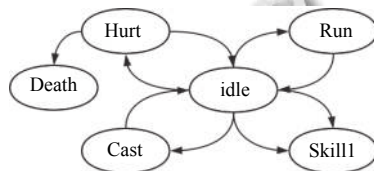


图 8 哥布林动画状态机图

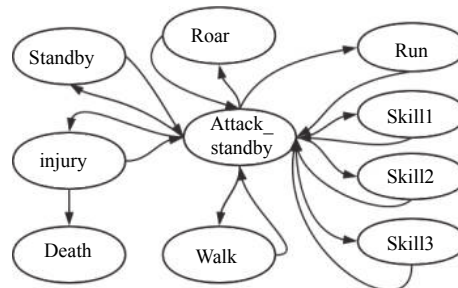


图 9 Boss 动画状态机图

系统设计的怪物有 3 种状态: 巡逻, 追赶玩家和战斗, 如图中的 Walk、Run 和 Attack. 当怪物距离玩家较远时, 怪物在固定的几个点来回缓慢行走 (Walk), 同时

实时观察自己周围是否有玩家进入;当怪物法线有玩家进入周围时,进入追赶玩家状态,朝着玩家的方向奔跑(Run);在追赶状态,当玩家跑远了,怪物发现玩家已经不在自己范围内时则返回巡逻状态;反之当追赶上玩家,即玩家在自己攻击范围内时则进入攻击状态,此时不再移动,而是朝向玩家播放攻击动画。

2.4 系统特效

一个比较成功的游戏中往往会有许多华丽炫酷的特效,包括场景特效和人物技能特效。本文的游戏中也加入了相应的特效设计。场景特效包括火焰、烟雾特效等,人物技能特效包括闪电光特效、刀光特效、烟雾魔法特效等。每一个特效都是由多种效果组合完成。图10展示的是游戏中的特效效果。



图10 特效效果图

2.5 游戏运行及操作控制

本文设计的游戏内存和内核的使用率低,对于GPU硬件的要求较高。基本运行和最高运行时的需求如表1。

表1 游戏运行需求

	游戏基本运行	游戏最高运行
CPU	单核 0.8 GHz 以上	单核 1 GHz 以上
RAM	>1 GB	>2 GB
GPU	Nvidia GeForce GT630 以上	Nvidia Quadro K620 以上

玩家可通过鼠标和键盘的方式控制角色,行走时可使用鼠标点击地面或者按 WASD 键控制人物在场景中的走动。同时, Q、R、T 按键为人物技能键, B 为背包键。

3 关键算法

3.1 导航网格寻路算法

在游戏对战的过程中,玩家角色需要在游戏场景中行走,因此需要解决角色的寻路问题。当玩家点击地面时,角色会自动走向点击的位置。由于 Unity3D 自带的寻路方式精确度不够,不能很好的满足玩家的心理。所以本文在此基础上进行改进,使之能够玩家的需求。

3.1.1 三角导航网格的生成

导航网格是一个凸多边形的集合,凸多边形指其内部任意两点的连线都在其内部,每个单元在路径搜索的过程中高度独立。本文首先利用建模工具对地图进行处理,剔除障碍物所在区域,生成可行走区域的多边形网格地图。然后对多边形网格进行三角剖分,使寻路算法能够运用于多边形网格。本文的多边形网格地图三角剖分采用的是一种快速约束 Delaunay 三角剖分算法^[3]。算法如下:

输入: navmesh 多边形导航网格数据
输出: Delaunay 三角网格

1. 读取网格信息,保存边界和顶点信息;
2. 遍历边界边,作为起始约束边;
3. 如果顶点不为空;
4. 判断可见点;
5. 创建三角形外接圆和外接圆的包围盒;
6. 遍历外接圆包围盒的所有顶点;
7. 判断 DT 点;
8. 连接两点之间的线;
9. 返回三角形,返回第3步。

三角剖分的前后效果如图11所示。

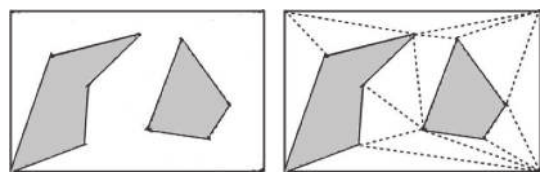


图11 三角剖分效果图

3.1.2 网格寻路

本文基于改进 A*算法^[4]的 3D 场景的自动寻路。网格寻路的效果如图12所示,蓝色区域为可行走区域,其他为不可行走区域。

3.2 预测式位置同步算法

由于存在网络延迟,很难做到实时的、精确的位置同步,只能通过各种方法来尽量减少误差^[5]。传统的

有瞬移式位置同步算法和移动式位置同步算法,但是游戏体验不好,误差比较大,实时性较差.所以本文采用预测式位置同步来提升用户的体验,减少误差.

在某些有规律可循的条件下,比如玩家匀速或者匀加速运动,我们能够预测玩家在接下来某个时间点的位置,让玩家提前走到预测的位置上去,这就是预测算法.

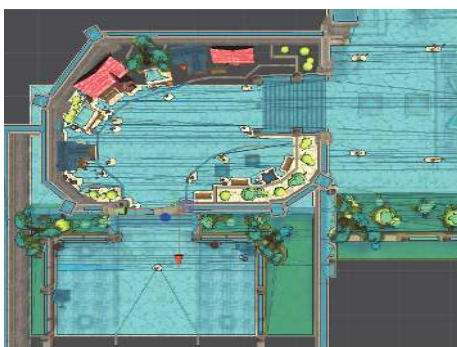


图 12 路径生成

假设玩家的移动速度保持不变,同步时间不变,玩家 2 可以预测玩家 1 的后续同步位置.如图 13 所示.玩家 1 经过 B 点时发送同步信息,玩家 2 根据 $s=vt$ 计算得出下一次同步时玩家 1 应移动到 C 点,然后以速度 v 向 C 点移动.此时的距离误差为 $v \cdot t$ (延迟).

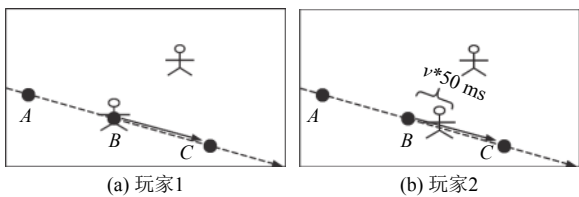


图 13 刚刚同步位置的距离误差

如图 14 所示,玩家 2 始终保持着速度 v 向预测位置靠近,距离误差一直是 $v \cdot t$ (延迟).若忽略网络延迟时间,则误差为 0.

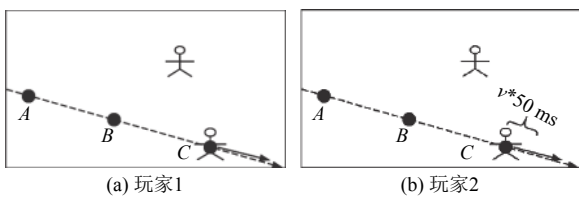


图 14 同步位置后的距离误差

玩家 1 不可能一直保持恒定的速度(包括速度的大小和方向).玩家 2 虽然可以计算最近两次同步信息的差值来获取较为准确的移动速度,但是误差还是存在.但总体上这种误差在可接受范围之内,平均误差依然为 0.

4 系统优化

游戏制作完成后,需要对系统进行优化以此来提升游戏运行的流畅度和玩家的游戏体验.为此,本文采用了以下几个技术来优化系统.

(1) LOD (Level Of Detail) 技术

LOD 是对模型建立了一个模型金字塔,根据摄像机距离对象的远近,选择使用不同精度的模型.它的好处是在适当的时候大量减少需要绘制的顶点数目^[6].图 15 展示了使用 LOD 技术后油桶从一个完整网格到简化网格,最后完全被剔除的例子.当距离摄像机较近的情况下使用 LOD0 层级的模型,即高精度模型;较远的情况下使用 LOD2 层级,即低精度模型.

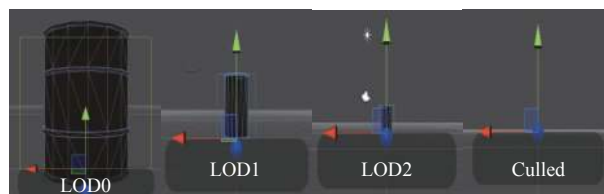


图 15 使用 LOD 技术效果图

(2) 遮挡剔除技术

遮挡剔除是当一个物体被其他物体遮挡住而不在摄像机的可视范围内时不对其进行渲染.遮挡剔除前后的效果如图 16 所示.使用该技术后,系统会自动剔除相机视野中被遮挡的物体.

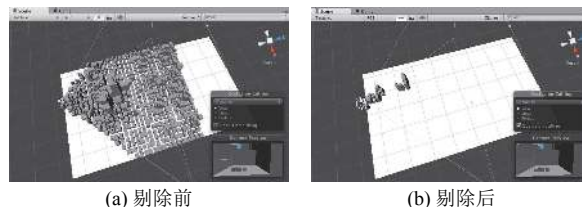


图 16 使用遮挡剔除技术效果图

(3) Lightmaps 技术

主要用于场景中整体的光照效果.这种技术主要

是提前把场景中的光照信息存储在一张光照纹理中,然后在运行时刻只需要根据纹理采样得到光照信息即可.与之配合的还有 Light Probes 技术.

5 结语

本文实现了多人在线网络游戏系统,其中包括行走控制、人工智能系统、界面系统、服务端框架、客户端框架、房间系统、战场系统等多项内容.游戏操作简单,界面美观,画面感强,在后期的体验中整体上满足玩家的需求.

后期的工作是在游戏中添加更多的功能模块,为游戏增加不同场景的地图和不同游戏角色,使玩家有更多的选择.

参考文献

- 1 吴晶晶,戴智超.基于 Unity3D 的多人在线网络游戏设计与开发.计算机系统应用,2017,26(10):133-138.[doi:10.15888/j.cnki.csa.006028]
- 2 翟涛.基于 Unity3D 的游戏智能行为体的研究与设计[硕士学位论文].沈阳:沈阳师范大学,2018.
- 3 朱昌龙,刘黎志.基于多边形导航网格地图的改进 A*算法.软件导刊,2019,18(1):17-21.
- 4 陈素琼,王惠来,向天雨.基于改进 A*算法的地图游戏寻径研究.重庆师范大学学报(自然科学版),2017,34(4):75-78.
- 5 陈俊锋.基于 Unity3D 的跨平台手机网络游戏的研究与实现[硕士学位论文].广州:中山大学,2013.
- 6 龚珏.基于 Unity3D 的移动游戏客户端框架设计与应用[硕士学位论文].武汉:华中科技大学,2018.