

网络多人游戏 架构与编程

[美] Joshua Glazer Sanjay Madhav 著

王晓慧 张国鑫 译

MULTIPLAYER
GAME
Programming



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



Pearson

网络多人游戏 架构与编程

[美] Joshua Glazer Sanjay Madhav 著

王晓慧 张国鑫 译

MULTIPLAYER
GAME
Programming

人民邮电出版社
北京

图书在版编目(CIP)数据

网络多人游戏架构与编程 / (美) 格雷泽
(Joshua Glazer), (美) 马达夫 (Sanjay Madhav) 著;
王晓慧, 张国鑫译. — 北京: 人民邮电出版社, 2017. 10
ISBN 978-7-115-45779-0

I. ①网… II. ①格… ②马… ③王… ④张… III.
①网络游戏—游戏程序—程序设计 IV. ①TP317.63

中国版本图书馆CIP数据核字(2017)第137928号

版 权 声 明

Authorized translation from the English language edition, entitled Multiplayer Game Programming, 0134034309 by Joshua Glazer, Sanjay Madhav, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright © 2016 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2017.

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

-
- ◆ 著 [美] Joshua Glazer Sanjay Madhav
 - 译 王晓慧 张国鑫
 - 责任编辑 胡俊英
 - 责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京捷迅佳彩印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 23
 - 字数: 448 千字 2017 年 10 月第 1 版
 - 印数: 1—2 500 册 2017 年 10 月北京第 1 次印刷

著作权合同登记号 图字: 01-2016-2847 号

定价: 109.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

内容提要

网络多人游戏已经成为游戏产业的重要组成部分，本书是一本深入探讨关于网络多人游戏编程的图书。

全书分为13章，从网络游戏的基本概念、互联网、伯克利套接字、对象序列化、对象复制、网络拓扑和游戏案例、延迟、抖动和可靠性、改进的延迟处理、可扩展性、安全性、真实世界的引擎、玩家服务、云托管专用服务器等方面深入介绍了网络多人游戏开发的知识，既全面又详尽地剖析了众多核心概念。

本书的多数示例基于C++编写，适合对C++有一定了解的读者阅读。本书既可以作为大学计算机相关专业的指导教程，也可以作为普通读者学习网络游戏编程的参考指南。

作者简介

Joshua Glazer是Naked Sky Entertainment的创始人之一和CTO。Naked Sky Entertainment是一个独立的游戏开发工作室，开发游戏机和电脑游戏，如*RoboBlitz*、*MicroBot*、*Twister Mania*和最近手机端的游戏*Max Axe*和*Scrap Force*。作为Naked Sky团队的领导人之一，他为许多外部的项目提供咨询，包括Epic Games公司的虚幻引擎（Unreal Engine）、Riot Games公司的《英雄联盟》（*League of Legends*）、THQ公司的《毁灭全人类》（*Destroy All Humans*）的特许经营，还包括艺电（Electronic Arts）、Midway、微软（Microsoft）和派拉蒙电影公司（Paramount Pictures）等公司的很多其他项目。

Joshua同时也是南加州大学（USC）的兼职讲师，在那里他很喜欢讲授关于多人游戏编程和游戏引擎开发的课程。

Sanjay Madhav是南加州大学（USC）的高级讲师，讲授许多编程和电子游戏编程的课程。他的主打课程是从2008年就开始讲授的本科生游戏编程课。此外，他还讲授许多其他课程，包括游戏引擎、数据结构和编译器开发。他同时也是*Game Programming Algorithms and Techniques*的作者。

在来到南加州大学（USC）之前，Sanjay在许多电子游戏开发公司做程序员，包括艺电（Electronic Arts）、Neversoft和Pandemic Studios。他开发过的游戏有《荣誉勋章：血战太平洋》（*Medal of Honor: Pacific Assault*）、《托尼霍克极限滑板8》（*Tony Hawk's Project 8*）、《指环王：征服》（*Lord of the Rings: Conquest*）和《破坏者》（*The Saboteur*），大部分都有某种形式的网络多人版。

译者简介

王晓慧，博士，就职于北京科技大学工业设计系。主要研究方向为虚拟现实、大数据与信息可视化、人工智能、情感计算、人机交互等，在国内外学术期刊和会议上发表论文20余篇。主持国家自然科学基金、北京市社会科学基金、北京市科技计划项目及中国博士后科学基金等国家、省部级项目4项，主持中央高校基本科研业务费、CFF-腾讯犀牛鸟创意基金等项目5项，参与973项目、国家自然、国家社科等项目6项，获得软件著作权2项，出版译著《精通Unreal游戏引擎》。邮箱地址是xiaohui0506@foxmail.com。

张国鑫，博士，蓝因（北京）科技有限公司CEO。多年来一直从事虚拟现实、三维模型处理和人工智能相关的研究和工 作，发表SCI论文多篇，取得发明专利4项，获得软件著作权3项，曾获得国家技术发明奖二等奖，教育部技术发明奖一等奖。邮箱地址是zgx@lanetechs.com。

致谢

首先，我们感谢 Pearson 的整个团队指导完成这本书，包括我们的执行编辑 Laura Lewin，劝说我们聚到一起写这本书；我们的助理编辑 Olivia Basegio，协助我们保证过程的顺利进行；我们的开发编辑 Michael Thurston，为提升内容质量出谋划策。我们还要感谢整个制作团队，包括制作编辑 Andy Beaster 和 Cengage 公司。

我们的技术评论家 Alexander Boczar、Jonathan Rucker 和 Jeff Tucker 为保证这本书的准确性帮了很大的忙。感谢他们在百忙之中抽出时间审阅这本书。最后，我们感谢维尔福软件公司（Valve Software）允许我们写关于 Steamworks SDK 的知识并审读了第 12 章。

来自 Joshua Glazer 的致谢

感谢 Lori 和 McKinney 给我无限的理解、支持、爱和微笑。你们是我最好的家人。由于写这本书，我陪你们的时间少了，但是现在，我终于完成了！感谢父母的养育之恩和对我的无限关爱，让我的英文能和代码一样流畅。感谢 Beth 做了很多不可思议的事情，有时还帮我照看我的猫。感谢大家庭在我写这本书时给我的支持和信任。感谢 Charles 和 Naked Sky 所有的程序员让我时刻保持警惕，在我犯愚蠢错误的时候及时指出。感谢 Tian 和 Sam 将我带到游戏产业中来。感谢 Sensei Copping 教会我如何有效工作。当然，感谢 Sanjay 把我带上了南加州大学（USC）的讲台，与我一起完成这个事情。如果没有你的智慧和冷静，我不能完成这本书，更不用说你写了这本书的一半。（再一次感谢 Lori，以防你错过了第一句话！）

来自 Sanjay Madhav 的致谢

作者的著作数量与致谢的长度有关。因为在上一本书中我写了很长的致谢，所以这次我写得短一点。我首先感谢我的父母和妹妹。同时，感谢我在南加州大学（USC）信息技术专业（Information Technology Program）的同事。最后，感谢 Joshua Glazer 同意讲授“多人游戏编程”这门课程，因为如果不是因为这门课，这本书就不会诞生。

前言

今天，网络多人游戏已经成为游戏产业的一个重要组成部分，玩家的数量和投入的资金数额都是惊人的。到2014年为止，《英雄联盟》(League of Legends)月活跃用户已达到6700万。在写这本书的时候，2015年DoTA 2的世界锦标赛的奖金已经超过1600万美元。《使命召唤》(Call of Duty)系列经常在发布后的头几天就打破10亿美元的销售额，其大受欢迎的部分原因在于它的多人模式。即使有的游戏曾经是单人模式，现在也都包含了网络多人成分，例如《侠盗猎车手》(Grand Theft Auto)。

本书深入探讨用于网络多人游戏编程的所有重要知识。首先介绍网络的基本知识——互联网是如何工作的，如何将数据发送给其他计算机。建立基本知识架构之后，本书讨论用于游戏数据传输的基本知识——如何准备用于网络传输的游戏数据，如何更新网络中的游戏对象，如何组织参与游戏的计算机。接着，本书论述如何补偿不可靠性和网络延迟，如何将游戏代码设计得同时具有可扩展性和安全性。第12章和第13章讲述如何集成玩家服务，以及使用云托管专用服务器，两个内容在今天的网络游戏中非常重要。

本书采用非常实用的方式。大部分章节不仅讨论概念，而且通过真实的代码让读者了解到网络游戏的工作原理。本书的配套网站提供两个不同游戏的源代码，一个是动作游戏，另一个是即时战略游戏(real-time strategy, RTS)。为了适应各个章节的内容演进，本书从头到尾会包含这两个游戏的多个版本。

本书中的大部分内容是基于南加州大学(University of Southern California, USC)的多人游戏编程课程的。正因为如此，本书包含的是被验证的行之有效的多人游戏编程学习方法。虽然这样说，但是本书不仅用于大学课程，书中的方法对于任何对网络游戏编程感兴趣的人来说都是有价值的。

本书的目标读者

虽然附录 A 包含了本书中所使用到的现代 C++ 的一些内容，但是本书假设读者已经熟悉 C++。本书进一步假设读者已经掌握在计算机课程中学习的标准数据结构。如果您不熟悉 C++，或者想复习一下数据结构，有一本非常好的参考书——*Programming Abstraction in C++*，作者 Eric Roberts。

本书还假设读者已经了解如何编写单人游戏。读者应该熟悉游戏循环、游戏对象模型、向量数学和基础游戏物理。如果读者不熟悉这些概念，建议先学习游戏编程入门书——*Game Programming Algorithms and Techniques*，作者 Sanjay Madhav。

正如之前所提到的，本书既适用于大学课程，也适合想要学习网络游戏编程的程序员。即使是没有接触过网络游戏的游戏程序员，也能在这本书中找到大量有用的知识。

本书中使用的约定

代码通常使用定点字体。小的代码片段可能出现在段落内，也可能是单独的一段：

```
std::cout << "Hello, world!" << std::endl;
```

长的代码段以代码清单的形式展现，如清单 0.1 所示。

清单 0.1 代码清单示例

```
// Hello world program!  
int main()  
{  
    std::cout << "Hello, world!" << std::endl;  
    return 0;  
}
```

为了提高可读性，示例代码与编程环境中的显示基本一致。

在本书中，读者会看到一些段落标记为注释、小窍门、边栏和警告。下面分别举例说明。

注释：

注释包含与正文上下文分离的有用信息，应该仔细阅读。

小窍门：

当在你的游戏代码中实现特定的系统时，小窍门将提供一些有用的提示。

警告：

警告非常重要，因为它包括一些常见的陷阱和需要注意的问题，同时包括解决这些问题的方法。

边栏

边栏通常包含与章节主要内容关系不太密切的更多讨论。这些内容针对各种各样的问题提供一些有趣的见解，但是其包含的内容往往不是章节教学目标所必需的。

为什么使用 C++

本书中大量使用 C++，是因为 C++ 是游戏产业事实上的编程语言，被游戏引擎程序员广泛使用。尽管有的引擎允许游戏中的大部分代码使用其他语言，如 Unity 中使用 C#，但是要谨记，这些引擎的底层代码仍然是用 C++ 写的。因为本书从头开始讲解编写网络多人游戏，所以使用大部分游戏引擎所使用的语言。话虽这么说，即使读者使用其他语言编写游戏网络代码，其所有的核心思想仍然是一样的。尽管如此，还是建议读者学会 C++，不然书中的示例代码对读者来说可能就没有多大意义了。

为什么使用 JavaScript

尽管刚开始的时候 JavaScript 是为了支持 Netscape 浏览器被匆忙赶出的脚本语言，它已经发展成为标准化、功能全面、有些函数式风格的语言。它在客户端语言中的流行帮助它跨越到了服务器端。在服务器端，它的第一级过程、简单的闭包语法和动态类型的性质，使其对于开发事件驱动的服务非常有效。JavaScript 重构困难，性能比 C++ 差，因此很难成为下一代游戏前端开发的首选语言。

对于服务器端来讲，这不是一个问题，因为扩大服务规模就像向右边拖动滑块这么简单。第 13 章中服务器端的例子使用 JavaScript，理解这部分内容需要了解该语言的一些知识。截至本书写作的时候，JavaScript 已经成为 GitHub 上最活跃的语言之一，其占比接近 50%。仅仅为了赶时髦而跟随趋势并不是一个好主意，但是使用当下最流行的语言编程肯定有它的好处。

配套网站

本书的配套网站提供了本书中所使用示例代码的链接。同时，网站还包括勘误表，幻灯片的链接和用于高校的示例教学大纲。

目录

第1章 网络游戏概述	1
1.1 多人游戏的简要历程	1
1.1.1 本地多人游戏	1
1.1.2 早期网络多人游戏	1
1.1.3 多用户网络游戏	2
1.1.4 局域网游戏	3
1.1.5 在线游戏	3
1.1.6 大规模多人在线游戏	4
1.1.7 移动网络游戏	4
1.2 星际围攻：部落	5
1.2.1 平台数据包模块	7
1.2.2 连接管理器	7
1.2.3 流管理器	8
1.2.4 事件管理器	8
1.2.5 ghost 管理器	8
1.2.6 移动管理器	9
1.2.7 其他系统	9
1.3 帝国时代	10
1.3.1 轮班计时器	11
1.3.2 同步	12
1.4 总结	13
1.5 复习题	13

1.6 延伸的阅读资料	14
第2章 互联网	15
2.1 起源：分组交换	15
2.2 TCP/IP 模型	17
2.3 物理层	18
2.4 链路层	18
2.5 网络层	22
2.5.1 IPv4	23
2.5.2 IPv6	36
2.6 传输层	37
2.6.1 UDP	38
2.6.2 TCP	39
2.7 应用层	48
2.7.1 DHCP	49
2.7.2 DNS	49
2.8 NAT	50
2.9 总结	56
2.10 复习题	57
2.11 延伸的阅读资料	58
第3章 伯克利套接字	61
3.1 创建 Socket	61
3.2 API 操作系统差异	63
3.3 socket 地址	66
3.3.1 类型安全	68
3.3.2 用字符串初始化 sockaddr	70
3.3.3 绑定 socket	73
3.4 UDP Socket	74
3.5 TCP Socket	79
3.5.1 通过连接的 socket 实现发送和接收	80
3.5.2 类型安全的 TCP Socket	82
3.6 阻塞和非阻塞 I/O	84
3.6.1 多线程	85
3.6.2 非阻塞 I/O	86

3.6.3 Select	88
3.7 其他Socket选项	92
3.8 总结	94
3.9 复习题	94
3.10 延伸的阅读资料	95
第4章 对象序列化	97
4.1 序列化的需求	97
4.2 流	100
4.2.1 内存流	101
4.2.2 字节存储次序的兼容性	105
4.2.3 比特流	109
4.3 引用数据	115
4.3.1 内联或嵌入	116
4.3.2 链接	117
4.4 压缩	119
4.4.1 稀疏数组压缩	120
4.4.2 熵编码	121
4.4.3 定点	123
4.4.4 几何压缩	125
4.5 可维护性	127
4.5.1 抽象序列化方向	127
4.5.2 数据驱动的序列化	129
4.6 总结	132
4.7 复习题	133
4.8 延伸的阅读资料	134
第5章 对象复制	135
5.1 世界状态	135
5.2 复制对象	135
5.2.1 对象创建注册表	139
5.2.2 一个数据包中的多个对象	143
5.3 朴素的世界状态复制方法	144
5.4 世界状态中的变化	148
5.5 RPC作为序列化对象	155
5.6 自定义解决方案	158

5.7 总结	159
5.8 复习题	160
5.9 延伸的阅读资料	160
第6章 网络拓扑和游戏案例	161
6.1 网络拓扑	161
6.1.1 客户端-服务器	161
6.1.2 对等网络	163
6.2 客户端-服务器的实现	165
6.2.1 服务器和客户端的代码分离	166
6.2.2 网络管理器和欢迎新客户端	167
6.2.3 输入共享和客户端代理	172
6.3 对等网络的实现	177
6.3.1 欢迎新对等体和开始游戏	179
6.3.2 命令共享和锁步回合制	181
6.3.3 保持同步	187
6.4 总结	192
6.5 复习题	192
6.6 延伸的阅读资料	193
第7章 延迟、抖动和可靠性	195
7.1 延迟	195
7.1.1 非网络延迟	195
7.1.2 网络延迟	198
7.2 抖动	199
7.3 数据包丢失	201
7.4 可靠性: TCP 还是 UDP	202
7.5 数据包传递通知	204
7.5.1 标记传出的数据包	205
7.5.2 接收数据包并发送确认	206
7.5.3 接收确认并传递状态	211
7.6 对象复制可靠性	216
7.7 模拟真实世界的条件	224
7.8 总结	226
7.9 复习题	227

7.10 延伸的阅读资料	228
第8章 改进的延迟处理	229
8.1 沉默的客户终端	229
8.2 客户端插值	231
8.3 客户端预测	233
8.3.1 航位推测法	236
8.3.2 客户端移动预测和重放	238
8.3.3 通过技巧和优化隐藏延迟	243
8.4 服务器端回退	244
8.5 总结	245
8.6 复习题	246
8.7 延伸的阅读资料	246
第9章 可扩展性	247
9.1 对象范围和相关性	247
9.1.1 静态区域	248
9.1.2 使用视锥	249
9.1.3 其他可见性技术	250
9.1.4 不可见时的相关性	252
9.2 服务器分区	253
9.3 实例化	255
9.4 优先级和频率	255
9.5 总结	256
9.6 复习题	257
9.7 延伸的阅读资料	257
第10章 安全性	259
10.1 数据包嗅探	259
10.1.1 中间人攻击	259
10.1.2 在主机上的数据包嗅探	262
10.2 输入验证	263
10.3 软件作弊检测	264
10.3.1 维尔福反作弊系统	265
10.3.2 典狱长反作弊系统	266
10.4 保护服务器	267