

Unity3D 网络游戏实战

罗培羽 著

Multiplayer Game

With Unity3D

- 深入讲解Unity3D网络游戏开发的原理和实现细节，全面剖析游戏网络框架设计。
- 实战性强，以多人坦克对战游戏案例贯穿全书，详述网络游戏的开发过程，帮助读者掌握商业游戏的设计思路和实现方法。



Unity3D 网络游戏实战

罗培羽 著



Multiplayer Game Development With Unity3D



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Unity3D 网络游戏实战 / 罗培羽著. —北京: 机械工业出版社, 2016.10
(游戏开发与设计技术丛书)

ISBN 978-7-111-54996-3

I. U… II. 罗… III. 游戏程序 - 程序设计 IV. TP317.6

中国版本图书馆 CIP 数据核字 (2016) 第 236591 号

Unity3D 网络游戏实战

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 和 静

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2016 年 11 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 28.25 (含 0.5 印张彩插)

书 号: ISBN 978-7-111-54996-3

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东



为什么要写这本书

笔者在玩到好玩的游戏时，总会希望有朝一日自己也能够做出经典的游戏作品。然而随着玩家欣赏水平的提高和网络游戏的发展，早期游戏的简陋画面再也不能吊起玩家的胃口，游戏大作再也不是一个人花几个星期就能完成的事情。Unity3D、虚幻4等引擎也正因为能够帮助开发者较快地制作出炫酷的游戏产品而备受青睐。

笔者曾与小伙伴们一起开发了一款仙剑同人游戏《仙剑5前传之心愿》，它是第一款由玩家开发完成的仙剑3D同人游戏，也是第一款采用即时战斗的仙剑游戏。那种只为圆一个游戏梦想，无条件付出的日子实属难忘。该作品就是使用Unity3D开发的，品质还算精良，读者可以登录 pal5h.com 下载。



《仙剑5前传之心愿》游戏截图

以前笔者主要关注单机游戏，也总能够找到很多资料进行学习，很快就能做出成果。然而由于猖獗的盗版市场及国外优秀作品的打击，国内单机游戏公司纷纷转型开发网络游戏，大部分游戏公司都在使用 Unity3D 引擎开发网络游戏。因此对于开发者而言，学习 Unity3D 网络游戏开发至关重要。

然而市面上的 Unity3D 教程，大多是介绍单机游戏开发方法的，就算涉及网络，也只是简单带过。如果读者想要制作当今热门的网络游戏，特别是手机网络游戏，单单参考市面上的书肯定是不够的。如果想要到游戏公司求职，仅仅凭借这些知识，也不容易成功应聘。针对这一痛点，本书以制作一款完整的多人坦克对战游戏为例，详细介绍网络游戏的开发过程。书中还介绍了一套通用的服务端框架和客户端网络模块（它是商业游戏的简化版本），相信通过本书，读者能够掌握 Unity3D 网络游戏开发的大部分知识，也能够从框架设计中了解商业游戏的设计思路。

读者对象

根据用户的需求来区分，可能使用本书的读者如下。

- 游戏开发爱好者：想要自己制作一款游戏的人，作为自学参考书。
- 求职者：想要谋求游戏公司开发岗位的人，作为自学参考书。
- 职场新人：游戏公司程序员，作为自学参考书。
- 游戏公司：作为新人培训资料。
- 学校：可作为大专院校或游戏培训机构的实验教材。

如何阅读本书

本书先提出了一个明确的学习目标，即制作一款完整的多人对战游戏，然后逐步去实现它。全书涉及行走控制、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。在涉及相关知识点时，书中会有详细的讲

解。具体来说，本书分为如下 3 个部分。

第一部分“单机游戏”：第 1 章至第 5 章，主要介绍如何开发一款功能完整的坦克单机游戏。除了让坦克行走、开炮，还将介绍基于代码和资源分离的界面系统、敌人 AI。了解开发单机游戏知识，也是为接下来的网络开发学习奠定基础。

第二部分“网络原理”：第 6 章至第 8 章，主要介绍网络通信的原理，开发客户端的网络模块和服务端程序框架。这套框架具有较高的通用性，可以运用在多种游戏上。

第三部分“网络游戏”：第 9 章至第 12 章，主要讲解房间系统和同步系统的逻辑实现，将单机坦克游戏改造成多人对战的网络游戏。

作为实例教程，本书偏重于例子中涉及的知识点。如果读者想要深入了解某些内容，或者了解实现某种功能的更多方法，建议在阅读本书的过程中多多查询相关资料，以便做到举一反三。

本书提供了所有示例的源码和素材，读者可以在作者提供的网盘中下载这些源码。由于网盘具有不稳定性，笔者不能保证多年后网盘地址还有效。若读者发现网盘地址失效，可以发送邮件到笔者的邮箱，笔者将会把最新的下载地址发给您。

下载地址：<http://pan.baidu.com/s/1c18esDE>，密码：9inz

笔者邮箱：aglab@foxmail.com

勘误和支持

由于笔者水平有限，写作的时间也很仓促，书中难免会出现一些错误或不准确的地方，恳请读者批评指正。如果读者发现了书中的错误，或者有更多的宝贵意见，欢迎邮件交流，笔者很期待能够听到你们的真挚反馈。

致谢

2013 年 8 月份，笔者在筹备出版第一本图书《手把手教你用 C# 制作 RPG 游

戏》时，就已经着手规划这本介绍网络游戏技术的图书了。经过两年多的积累，本书渐渐成型。2016年2月，在机械工业出版社华章公司杨绣国编辑的帮助下，本书的出版事项提上了日程。

在此，我要感谢父母，有了他们的支持才有笔者义无反顾的前行。

感谢黄剑基和蒙屿森，他们作为本书的第一批读者，给了笔者不少可行的建议。

感谢一同开发《仙剑5前传之心愿》的唛玥、吾辈名妖、杨凯云等人，这段不为名利的奋斗历程使笔者永生难忘。

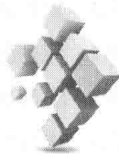
感谢郑志铭、卢阳飞、许远帆、林文佳、梁浩林等人在本书写作过程中给予的诸多鼓舞。

愿与诸位一同努力，造就顶级游戏产品。

每一款游戏都是梦想与智慧的结晶！

罗培羽

2016年6月于广州



前言

第1章 掌握Unity3D基本元素 / 1

- 1.1 最简单的游戏 / 2
 - 1.1.1 Unity3D 的界面构成 / 2
 - 1.1.2 在场景中创建立方体 / 3
 - 1.1.3 编写第一个程序 / 4
 - 1.1.4 测试游戏 / 6
 - 1.1.5 总结 / 6
- 1.2 导入资源 / 6
 - 1.2.1 从本地导入素材 / 7
 - 1.2.2 从本地导入包文件 / 7
 - 1.2.3 从 AssetStore 导入 / 8
- 1.3 山体系统 / 10
- 1.4 灯光 / 12
- 1.5 材质 / 14
 - 1.5.1 什么是材质 / 14
 - 1.5.2 如何创建材质 / 15
 - 1.5.3 Mesh Renderer 组件 / 16
 - 1.5.4 着色器 / 16
- 1.6 预设 / 19
 - 1.6.1 制作预设 / 19
 - 1.6.2 预设的实例化 / 19
 - 1.6.3 使用预设的例子 / 20

- 1.7 声音 / 22
 - 1.7.1 音源 / 22
 - 1.7.2 接收器 / 23
 - 1.7.3 简单播放器 / 23
- 1.8 GUI / 24
 - 1.8.1 GUI 绘图基础 / 24
 - 1.8.2 编写 HelloWorld 程序 / 25
 - 1.8.3 绘制登录框 / 25
- 1.9 场景 / 26
 - 1.9.1 创建场景 / 27
 - 1.9.2 场景切换 / 27
- 1.10 导出游戏 / 28

第2章 坦克控制单元 / 31

- 2.1 导入坦克模型 / 31
 - 2.1.1 导入模型 / 31
 - 2.1.2 调整尺寸 / 32
 - 2.1.3 材质和贴图 / 33
- 2.2 行走控制 / 34
 - 2.2.1 基础知识 / 35
 - 2.2.2 上下左右移动 / 37
 - 2.2.3 转向和前后移动 / 38
- 2.3 相机跟随 / 40
 - 2.3.1 数学原理 / 41

- 2.3.2 跟随算法 / 42
 - 2.3.3 设置跟随目标 / 46
 - 2.3.4 横向旋转相机 / 47
 - 2.3.5 纵向旋转相机 / 50
 - 2.3.6 滚轮调节距离 / 51
 - 2.4 旋转炮塔 / 52
 - 2.4.1 坦克的层次结构 / 52
 - 2.4.2 炮塔 / 56
 - 2.4.3 炮管 / 58
 - 2.5 车辆行驶 / 61
 - 2.5.1 Unity3D 的物理系统 / 62
 - 2.5.2 车轮碰撞器 / 65
 - 2.5.3 控制车辆 / 69
 - 2.5.4 制动(刹车) / 73
 - 2.6 轮子和履带 / 74
 - 2.6.1 轮子转动 / 74
 - 2.6.2 履带滚动 / 77
 - 2.7 音效 / 79
- ### 第3章 火炮与敌人 / 82
- 3.1 发射炮弹 / 82
 - 3.1.1 制作炮弹 / 82
 - 3.1.2 制作爆炸效果 / 83
 - 3.1.3 炮弹轨迹 / 84
 - 3.1.4 坦克开炮 / 86
 - 3.2 摧毁敌人 / 88
 - 3.2.1 坦克的控制类型 / 88
 - 3.2.2 坦克的生命值 / 89
 - 3.2.3 焚烧特效 / 89
 - 3.2.4 坦克被击中后的处理 / 90
 - 3.2.5 炮弹的攻击处理 / 92
 - 3.3 准心 / 94
 - 3.3.1 概念和原理 / 94
 - 3.3.2 计算目标射击位置 / 96
 - 3.3.3 计算实际射击位置 / 101
 - 3.3.4 绘制准心 / 103
 - 3.4 绘制生命条 / 104
 - 3.4.1 生命条素材 / 105
 - 3.4.2 绘制生命条 / 105
 - 3.5 击杀提示 / 107
 - 3.5.1 谁发射了炮弹 / 107
 - 3.5.2 谁被击中 / 107
 - 3.5.3 显示击杀提示 / 108
 - 3.6 炮弹的音效 / 110
 - 3.6.1 发射音效 / 110
 - 3.6.2 爆炸音效 / 111
- ### 第4章 人工智能 / 113
- 4.1 基于有限状态机的人工智能 / 113
 - 4.1.1 有限状态机 / 113
 - 4.1.2 分层有限状态机 / 115
 - 4.2 程序结构 / 116
 - 4.2.1 AI类的结构 / 116
 - 4.2.2 在 Tank 中调用 / 118
 - 4.3 搜寻目标 / 119
 - 4.3.1 搜寻规则 / 119
 - 4.3.2 坦克标签 / 119
 - 4.3.3 主动搜寻算法 / 120
 - 4.3.4 被动搜寻算法 / 122
 - 4.3.5 调试 / 123
 - 4.4 向敌人开炮 / 124
 - 4.4.1 电脑控制的方式 / 124
 - 4.4.2 炮塔炮管的目标角度 / 125
 - 4.4.3 调试程序 / 125
 - 4.4.4 开炮 / 126
 - 4.5 走向目的地 / 128
 - 4.5.1 路点 / 128
 - 4.5.2 路径 / 128
 - 4.5.3 根据场景标志物生成路径 / 130
 - 4.5.4 给 AI 指定路径 / 131

- 4.5.5 操控坦克 / 132
- 4.5.6 调试程序 / 136
- 4.6 使用 NavMesh 计算路径 / 137
 - 4.6.1 NavMesh 的原理 / 137
 - 4.6.2 生成导航图 / 137
 - 4.6.3 生成路径 / 140
- 4.7 行为决策 / 143
 - 4.7.1 巡逻状态 / 144
 - 4.7.2 进攻状态 / 145
 - 4.7.3 调试 / 146
- 4.8 战场系统 / 147
 - 4.8.1 单例模式 / 147
 - 4.8.2 BattleTank / 148
 - 4.8.3 战场逻辑 / 148
 - 4.8.4 敌我区分 / 150
 - 4.8.5 出生点 / 151
 - 4.8.6 坦克预设 / 152
 - 4.8.7 开启一场两军对峙的战斗 / 152
 - 4.8.8 战场结算 / 154
 - 4.8.9 开始战斗 / 155

第5章 代码分离的界面系统 / 157

- 5.1 Unity UI 系统 / 157
 - 5.1.1 创建 UI 部件 / 158
 - 5.1.2 Canvas 画布 / 159
 - 5.1.3 EventSystem / 161
 - 5.1.4 RectTransform / 162
 - 5.1.5 其他 UGUI 组件 / 164
 - 5.1.6 事件触发 / 165
 - 5.1.7 简单的面板调用 / 165
- 5.2 制作界面素材 / 167
 - 5.2.1 标题面板和信息面板 / 167
 - 5.2.2 制作预设 / 168
- 5.3 面板基类 PanelBase / 168
 - 5.3.1 代码与资源分离的优势 / 168

- 5.3.2 面板系统的设计 / 169
- 5.3.3 面板基类的设计要点 / 169
- 5.3.4 面板基类的实现 / 170
- 5.4 面板管理器 PanelMgr / 172
 - 5.4.1 层级管理 / 173
 - 5.4.2 打开面板 OpenPanel / 174
 - 5.4.3 关闭面板 ClosePanel / 176
- 5.5 面板逻辑 / 176
 - 5.5.1 标题面板 TitlePanel / 176
 - 5.5.2 信息面板 InfoPanel / 178
- 5.6 调用界面系统 / 179
 - 5.6.1 界面系统的资源 / 179
 - 5.6.2 界面系统的调用 / 179
- 5.7 胜负面板 / 181
 - 5.7.1 面板素材 / 181
 - 5.7.2 面板逻辑 / 181
 - 5.7.3 面板调用 / 183
- 5.8 设置面板 / 184
 - 5.8.1 面板素材 / 184
 - 5.8.2 面板逻辑 / 185
 - 5.8.3 面板调用 / 186

第6章 网络基础 / 188

- 6.1 七层网络模型 / 189
 - 6.1.1 应用层 / 190
 - 6.1.2 传输层 / 190
 - 6.1.3 网络层 / 190
 - 6.1.4 数据链路层 / 191
 - 6.1.5 物理层 / 191
- 6.2 IP 与端口 / 192
 - 6.2.1 IP 地址 / 192
 - 6.2.2 端口 / 192
 - 6.2.3 C# 中的相关类型 / 193
- 6.3 TCP 协议 / 193
 - 6.3.1 TCP 连接的建立 / 193

- 6.3.2 TCP 的数据传输 / 195
- 6.3.3 TCP 连接的终止 / 195
- 6.4 Socket 套接字 / 196
 - 6.4.1 Socket 连接的流程 / 196
 - 6.4.2 Socket 类 / 196
- 6.5 同步 Socket 程序 / 198
 - 6.5.1 新建控制台程序 / 198
 - 6.5.2 编写服务端程序 / 199
 - 6.5.3 客户端界面 / 202
 - 6.5.4 客户端程序 / 203
- 6.6 异步 Socket 程序 / 205
 - 6.6.1 BeginAccept / 205
 - 6.6.2 BeginReceive / 205
 - 6.6.3 Conn (state) / 206
 - 6.6.4 服务端程序 (主体结构) / 208
 - 6.6.5 服务端程序 (Accept 回调) / 210
 - 6.6.6 服务端程序 (接收回调) / 211
 - 6.6.7 开启服务端 / 212
 - 6.6.8 客户端界面 / 212
 - 6.6.9 客户端程序 / 213
 - 6.6.10 调试程序 / 215
- 6.7 MySQL / 216
 - 6.7.1 配置 MySQL 环境 / 216
 - 6.7.2 建立 MySQL 数据库 / 218
 - 6.7.3 MySQL 基础知识 / 218
 - 6.7.4 MySQL 留言板服务端程序 / 220
 - 6.7.5 调试程序 / 222
- 6.8 类的序列化 / 223
- 6.9 定时器 / 225
- 6.10 线程互斥 / 226
- 6.11 通信协议和消息列表 / 228
 - 6.11.1 通信协议 / 228
 - 6.11.2 服务端程序 / 229
 - 6.11.3 消息列表 / 229
 - 6.11.4 客户端场景 / 230

- 6.11.5 客户端程序 / 231
- 6.11.6 调试 / 236

第7章 服务端框架 / 238

- 7.1 服务端架构 / 238
 - 7.1.1 总体架构 / 238
 - 7.1.2 游戏流程 / 239
 - 7.1.3 连接的数据结构 / 240
 - 7.1.4 数据库结构 / 241
 - 7.1.5 项目结构 / 241
- 7.2 数据管理类 DataMgr / 243
 - 7.2.1 数据表结构 / 243
 - 7.2.2 角色数据 / 244
 - 7.2.3 Player 的初步版本 / 244
 - 7.2.4 连接数据库 / 245
 - 7.2.5 防止 sql 注入 / 246
 - 7.2.6 Register 注册 / 247
 - 7.2.7 CreatePlayer 创建角色 / 249
 - 7.2.8 登录校验 / 250
 - 7.2.9 获取角色数据 / 251
 - 7.2.10 保存角色数据 / 252
 - 7.2.11 调试 / 253
- 7.3 临时数据 / 255
- 7.4 网络管理类 ServNet / 256
 - 7.4.1 粘包分包现象 / 256
 - 7.4.2 粘包分包的处理方法 / 256
 - 7.4.3 Conn 连接类 / 257
 - 7.4.4 ServNet 网络处理类 / 260
 - 7.4.5 ReceiveCb 的粘包分包处理 / 261
 - 7.4.6 发送消息 / 264
 - 7.4.7 启动服务端 / 265
 - 7.4.8 调试 / 265
- 7.5 心跳 / 267
 - 7.5.1 心跳机制 / 267
 - 7.5.2 时间戳 / 268

- 7.5.3 使用定时器 / 269
- 7.5.4 心跳协议 / 270
- 7.5.5 调试心跳协议 / 270
- 7.6 协议 / 271
 - 7.6.1 协议基类 / 272
 - 7.6.2 字符串协议 / 273
 - 7.6.3 字节流协议 / 274
 - 7.6.4 字节流辅助方法 / 276
 - 7.6.5 使用协议 / 278
 - 7.6.6 调试 / 280
- 7.7 中间层 Player 类 / 282
 - 7.7.1 登录流程 / 282
 - 7.7.2 下线 / 282
 - 7.7.3 Player 类的实现 / 283
- 7.8 消息分发 / 285
 - 7.8.1 消息处理的类 / 285
 - 7.8.2 消息处理类的实现 / 286
 - 7.8.3 反射 / 287
- 7.9 注册登录 / 289
 - 7.9.1 协议 / 289
 - 7.9.2 注册功能 / 290
 - 7.9.3 登录功能 / 291
 - 7.9.4 登出功能 / 292
 - 7.9.5 获取分数功能 / 293
 - 7.9.6 增加分数功能 / 293
 - 7.9.7 输出服务端信息 / 294
 - 7.9.8 Main 中的调用 / 294
 - 7.9.9 测试用客户端 / 295
 - 7.9.10 调试 / 297

第8章 客户端网络模块 / 300

- 8.1 网络模块设计 / 300
 - 8.1.1 整体架构 / 300
 - 8.1.2 监听表 / 301
 - 8.1.3 类结构 / 301

- 8.2 委托 / 302
 - 8.2.1 使用委托 / 302
 - 8.2.2 示例 / 302
 - 8.2.3 操作符 / 303
- 8.3 MsgDistribution 消息分发 / 304
 - 8.3.1 MsgDistribution 的成员 / 304
 - 8.3.2 DispatchMsgEvent / 305
 - 8.3.3 AddListener / 306
- 8.4 Connection 连接 / 307
 - 8.4.1 Connection 的成员 / 307
 - 8.4.2 连接服务端 / 309
 - 8.4.3 关闭连接 / 309
 - 8.4.4 异步回调 / 310
 - 8.4.5 消息处理 / 311
 - 8.4.6 发送数据 / 311
 - 8.4.7 心跳机制 / 312
- 8.5 NetMgr 网络管理 / 313
- 8.6 登录注册功能 / 314
 - 8.6.1 界面资源 / 315
 - 8.6.2 登录面板功能 / 316
 - 8.6.3 注册面板功能 / 319
- 8.7 位置同步的服务端程序 / 320
 - 8.7.1 协议 / 321
 - 8.7.2 场景 / 321
 - 8.7.3 协议处理 / 324
 - 8.7.4 事件处理 / 324
- 8.8 位置同步的客户端程序 / 325
 - 8.8.1 客户端资源 / 325
 - 8.8.2 客户端程序 / 326
- 8.9 调试框架 / 331

第9章 房间系统 / 334

- 9.1 游戏界面 / 335
 - 9.1.1 登录面板 / 335
 - 9.1.2 注册面板 / 336

- 9.1.3 提示面板 / 337
 - 9.1.4 UGUI 的滑动区域 / 338
 - 9.1.5 房间列表面板 / 340
 - 9.1.6 房间面板 / 342
 - 9.1.7 创建预设 / 343
 - 9.2 协议设计 / 344
 - 9.3 提示框的功能实现 / 346
 - 9.4 登录注册的功能实现 / 348
 - 9.4.1 登录面板的功能 / 348
 - 9.4.2 GameMgr / 349
 - 9.4.3 注册面板的功能 / 350
 - 9.4.4 调试 / 351
 - 9.5 房间列表面板的功能 / 352
 - 9.5.1 获取部件 / 353
 - 9.5.2 开启监听 / 354
 - 9.5.3 刷新成绩栏 / 355
 - 9.5.4 刷新房间列表 / 355
 - 9.5.5 刷新按钮 / 357
 - 9.5.6 加入房间 / 357
 - 9.5.7 新建房间 / 358
 - 9.5.8 登出 / 359
 - 9.5.9 测试面板 / 360
 - 9.6 房间面板的功能 / 360
 - 9.6.1 获取部件 / 361
 - 9.6.2 监听 / 362
 - 9.6.3 刷新列表 / 362
 - 9.6.4 退出按钮 / 364
 - 9.6.5 开始战斗 / 365
 - 9.6.6 测试面板 / 366
- 第10章 房间系统服务端 / 368**
- 10.1 玩家数据 / 368
 - 10.2 房间类 / 370
 - 10.2.1 数据结构 / 370
 - 10.2.2 添加玩家 / 371
 - 10.2.3 删除玩家 / 372
 - 10.2.4 更换房主 / 373
 - 10.2.5 广播消息 / 373
 - 10.2.6 输出房间信息 / 374
 - 10.3 房间管理器 / 374
 - 10.3.1 数据结构 / 374
 - 10.3.2 创建房间 / 375
 - 10.3.3 离开房间 / 376
 - 10.3.4 输出房间列表 / 376
 - 10.4 玩家消息处理 / 377
 - 10.4.1 查询成绩 GetAchieve / 377
 - 10.4.2 获取房间列表 GetRoomList / 377
 - 10.4.3 创建房间 CreateRoom / 378
 - 10.4.4 加入房间 EnterRoom / 379
 - 10.4.5 获取房间信息 GetRoomInfo / 380
 - 10.4.6 离开房间 LeaveRoom / 380
 - 10.5 玩家事件处理 / 381
 - 10.6 调试 / 382
- 第11章 战场系统 / 386**
- 11.1 协议设计 / 386
 - 11.2 开始战斗 / 388
 - 11.2.1 客户端战场数据 / 389
 - 11.2.2 获取阵营 / 390
 - 11.2.3 清理场景 / 390
 - 11.2.4 开始战斗 / 391
 - 11.2.5 产生坦克 / 391
 - 11.2.6 服务端战场数据 / 394
 - 11.2.7 服务端条件检测 / 395
 - 11.2.8 服务端开启战斗 / 395
 - 11.2.9 服务端消息处理 / 396
 - 11.2.10 调试程序 / 397
 - 11.3 三种同步位置方案 / 398
 - 11.3.1 瞬移式位置同步 / 399
 - 11.3.2 移动式位置同步 / 400

- 11.3.3 预测式位置同步 / 401
- 11.4 位置同步的服务端处理 / 402
- 11.5 位置同步的客户端处理 / 404
 - 11.5.1 发送同步信息 / 404
 - 11.5.2 网络同步类型 / 405
 - 11.5.3 预测目标位置 / 405
 - 11.5.4 向目标位置移动 / 408
 - 11.5.5 监听服务端协议 / 408
 - 11.5.6 调试 / 409
- 11.6 同步炮塔炮管 / 410
- 11.7 轮子和履带 / 411
- 第12章 炮火同步 / 413**
 - 12.1 炮弹同步 / 413
 - 12.1.1 协议设计 / 413
 - 12.1.2 服务端处理 / 414
 - 12.1.3 客户端发送同步信息 / 415
 - 12.1.4 客户端接收同步信息 / 416
 - 12.2 伤害同步 / 418
 - 12.2.1 协议设计 / 418
 - 12.2.2 服务端处理 / 418
 - 12.2.3 客户端发送伤害信息 / 420
 - 12.2.4 客户端接收伤害信息 / 420
 - 12.3 胜负判断 / 423
 - 12.3.1 协议设计 / 423
 - 12.3.2 服务端胜负判断 / 423
 - 12.3.3 服务端处理战斗结果 / 424
 - 12.3.4 客户端接收战斗结果 / 425
 - 12.4 中途退出 / 427
 - 12.5 完整的游戏 / 428



第 1 章

掌握 Unity3D 基本元素

随着玩家欣赏水平的提高，早期游戏的简陋画面再也不能吊起玩家的胃口。精美的画面需要较高的成本，游戏大作再也不是一个人花几个星期就能完成的事情。Unity3D、虚幻 4 等引擎因能够帮助开发者制作炫酷的游戏产品而备受青睐。Unity3D 是一款优秀的 3D 游戏引擎，可以开发各种各样的游戏，比如 MMORPG（多人在线角色扮演游戏）、赛车游戏、动作竞技、射击游戏等。在移动平台，Unity3D 的地位更是举足轻重，目前市面上大部分 3D 手机游戏都是使用 Unity3D 开发的。

本书将通过一款完整的坦克游戏实例，介绍使用 Unity3D 制作网络游戏的方法和技巧，本书所使用的 Unity3D 版本是 5.2.1，读者可以使用高于此版本的 Unity3D 打开示范工程。事实上，大可不必太在意版本的区别，同一大的版本（比如 5.x.x）它们的功能是大同小异的。

本章将讲解 Unity3D 的一些基础知识，如果读者对 Unity3D 没有太多接触，那么看完本章就算入门了。下载安装 Unity3D，跟我一起踏上征程吧！每一款游戏都是梦想与智慧的结晶！

1.1 最最简单的游戏

第一次打开 Unity3D 时，将看到 Create a project（新建项目）的界面。点击 New Project 将跳转到如图 1-1 所示的创建项目界面，填写新项目的名称（Project name）和路径（Location）后点击 Create project 按钮，进入 Unity3D 的主界面。

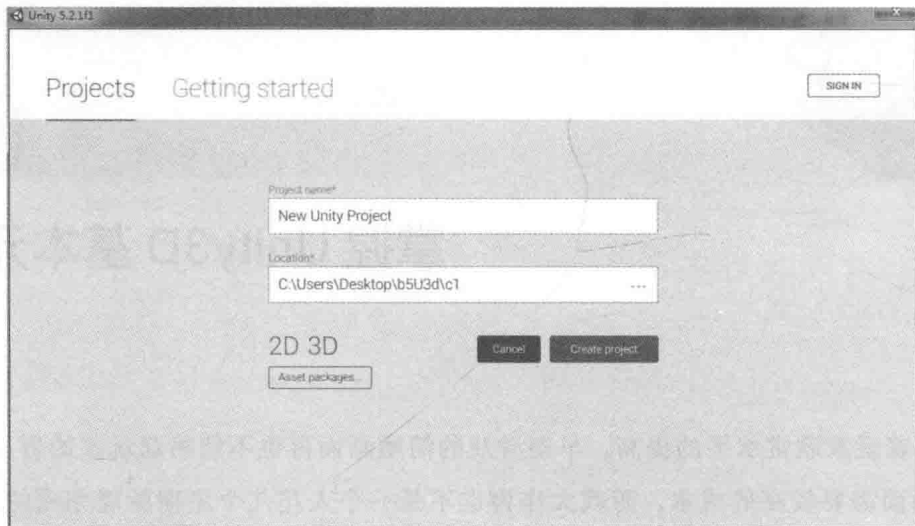


图 1-1 创建项目界面图

1.1.1 Unity3D 的界面构成

如图 1-2 所示，Unity3D 的基本界面并不复杂，4 个面板便囊括了常用的编辑功能，它们分别是：场景面板、属性面板、层次面板、项目面板。

在制作游戏之前，需要先熟悉这 4 个面板的用途，如表 1-1 所示。

表 1-1 Unity3D 主界面面板及功能

面板	功能
场景面板	Unity3D 中最常用的部分，场景中所有的模型、光源、摄像机、材质、音效等都显示在此面板上。在该面板中编辑游戏对象，包括旋转、移动、缩放等
属性面板	属性面板（检视窗口）可以显示游戏场景中当前选择对象的各种属性，包括对象的名称、标签、位置坐标、旋转角度、缩放、组件等
层次面板	显示场景中的所有物体
项目面板	列出游戏的所有资源，包括场景、脚本、三维模型、纹理、音频文件和预制组件等




图 1-2 Unity3D 主界面

1.1.2 在场景中创建立方体

让我们试着制作一款最最简单的游戏：场景中只有一个立方体，游戏开始后，立方体向右边移动，直到消失在“茫茫宇宙”之中。完成这一款游戏，只需要如下 3 个步骤即可。

- 1) 在场景中创建一个立方体。
- 2) 编写可以使立方体运动的程序。
- 3) 测试游戏。

点击菜单栏中的 **GameObject** → **3D Object** → **Cube**。完成后，层次面板和场景面板都会显示刚刚创建的立方体（如图 1-3 所示）。

可以通过菜单栏的  这几个按钮调整立方体的位置、旋转和尺寸（按钮的具体用途和效果分

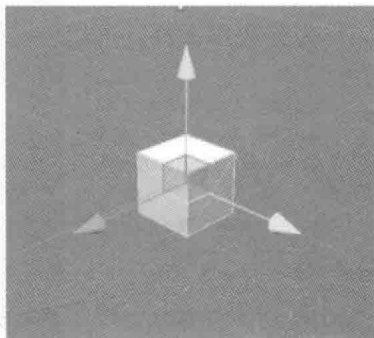
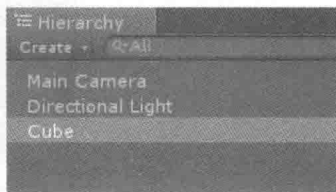


图 1-3 层次面板和场景面板中显示刚刚创建的立方体