

学 号： 2018218413

密 级： 公开

合肥工业大学

Hefei University of Technology

本科毕业设计

UNDERGRADUATE THESIS



类 型： 设计

题 目： 基于 Unity 的餐厅经营游戏的设计与开发

专业名称： 计算机科学与技术

入校年份： 2018 级

学生姓名： 欧林妍

指导教师： 张建军 教授

学院名称： 计算机与信息学院

完成时间： 2022 年 5 月

合 肥 工 业 大 学

## 本科毕业设计

### 基于 Unity 的餐厅经营游戏的设计与开发

学生姓名：欧林妍

学生学号：2018218413

指导教师：张建军 教授

专业名称：计算机科学与技术

学院名称：计算机与信息学院

2022 年 5 月

**A Dissertation Submitted for the Degree of Bachelor**

**Design and development of restaurant management  
game based on Unity**

By

Ou linyan

Hefei University of Technology

Hefei, Anhui, P.R.China

May, 2022

# 毕业设计独创性声明

本人郑重声明：所呈交的毕业设计是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在设计中作了明确的说明，并表示谢意。

毕业设计中表达的观点纯属作者本人观点，与合肥工业大学无关。

毕业设计作者签名：欧林妍 签名日期：2022 年 5 月 30 日

# 毕业设计版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用毕业设计的规定，即：除保密期内的涉密设计外，学校有权保存并向国家有关部门或机构送交设计的复印件和电子光盘，允许设计被查阅或借阅。本人授权合肥工业大学可以将本毕业设计的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计。

（保密的毕业设计在解密后适用本授权书）

学位论文作者签名：欧林妍 指导教师签名：张建华

签名日期：2022 年 5 月 30 日 签名日期：2022 年 5 月 30 日

## 摘 要

随着社会主义现代化、电子化程度的不断推进，游戏已经成为当代人群休闲娱乐的主要方式之一。除此之外，应运而生的还有游戏引擎，Unity 作为一款专业的游戏引擎，拥有集成开发平台，从众多游戏引擎中脱颖而出。本毕业设计采用 Unity 引擎作为平台，开发了一款可联机的餐厅经营游戏。

本文首先介绍了游戏的开发背景以及游戏开发中应用到的技术和方法，包括 Unity 的动画系统、资源商店、开发语言、网络通讯、数据库以及 Blender 建模工具的介绍。接下来文章介绍了游戏的需求和设计，除了需要准备游戏需要的资源外，为了实现游戏的联机，游戏还需要开发客户端与服务端两个工程。服务端需要接收客户端传送过来的消息并将消息转发给其他客户端，以实现不同客户端的消息同步。为了实现客户端与服务端的消息传递，设计需要定制消息的打包协议，方便消息的解析和数据获取。然后又从游戏资源准备、功能实现、以及动画和音频接入三个方面介绍了系统是如何实现的。在资源准备中，本设计使用 Blender 软件来完成游戏中 3D 模型的制作以及人物动画的制作，使用 UGUI 来制作游戏中的 UI 面板，游戏中的音频资源则由 Au 来进行录制。功能实现主要是按照游戏设计中客户端与服务端的划分来进行开发，客户端与服务端则根据功能划分为不同模块来进行开发，模块与模块之间通过中间模块来统一的进行调用与赋值，这样可以大大降低代码的耦合性，方便后期的调整与修改，代码的结构也更加整洁。

**关键词：**Unity；客户端；服务端；网络通讯；餐厅经营游戏；游戏开发

# ABSTRACT

With the continuous advancement of social modernization and electronization, games have become one of the main ways of leisure and entertainment for contemporary people. In addition, the game engine came into being. As a professional game engine, unity has an integrated development platform and stands out from many game engines. This graduation project uses unity engine as a platform to develop an online restaurant management game.

At first, this thesis introduces the background of game development and the techniques and methods used in game development, including Unity animation system, resource store, development language, network communication, database and Blender modeling tool introduction. Next, thesis introduces the requirements and design of the game, in addition to the resources of the need to prepare the game, in order to achieve the online game, the game also needs to develop two projects: the client and the server. To achieve message synchronization between different clients, the server needs to receive the messages sent by the client and forward the messages to other clients. In order to realize the message transmission between client and server, the design needs to customize the message packaging protocol to facilitate message parsing and data acquisition. Then it introduces how to realize the system from three aspects: game resource preparation, function realization, animation and audio access. In resource preparation, the design uses Blender software to complete the production of 3D model and character animation in the game, and it uses UGUI to produce UI panel in the game, and audio resources in the game are recorded by Au. The development of function realization is mainly based on the division of the client and server in the game design. The client and server are divided into different modules for development according to the functions. The modules are called and assigned uniformly through the intermediate modules. This can greatly reduce the coupling of the code, facilitate the later adjustment and modification, and make the structure of the code cleaner.

**KEYWORDS:** Unity; client; server; networking; restaurant game; cauldron;

# 目 录

1 绪论 .....	1
1.1 课题背景 .....	1
1.2 研究意义和目的 .....	1
1.3 主要工作 .....	1
1.4 本文组织结构 .....	2
2 相关技术和方法 .....	3
2.1 Unity 引擎 .....	3
2.1.1 Unity 引擎特点介绍 .....	3
2.1.2 动画系统 .....	3
2.1.3 资源商店 .....	3
2.2 c#语言 .....	3
2.3 网络通讯 .....	3
2.3.1 IP 和端口号 .....	4
2.3.2 TCP 协议的三次握手 .....	4
2.4 MySQL 数据库 .....	5
2.5 Blender 游戏建模 .....	5
2.6 本章小结 .....	5
3 需求分析与设计 .....	6
3.1 需求分析 .....	6
3.1.1 游戏资源 .....	6
3.1.2 数据存储 .....	6
3.1.3 服务端数据处理 .....	6
3.1.4 客户端逻辑处理 .....	6
3.1.5 网络通信 .....	7
3.2 游戏模块设计 .....	7
3.2.1 数据库 .....	7
3.2.2 服务端 .....	8
3.2.3 客户端 .....	9
3.2.4 网络通讯 .....	10

3.3 本章小结 .....	11
4 游戏的实现 .....	12
4.1 资源准备 .....	12
4.1.1 3D 模型以及动画 .....	12
4.1.2 UI 界面制作和管理 .....	14
4.1.3 音频录制 .....	16
4.1.4 数据库创建 .....	16
4.2 登陆和注册 .....	17
4.2.1 登陆 .....	17
4.2.2 注册 .....	18
4.3 房间管理 .....	19
4.3.1 创建房间 .....	19
4.3.2 加入房间 .....	20
4.4 游戏游玩 .....	22
4.5 动画和音频接入 .....	27
4.5.1 动画接入 .....	27
4.5.2 音频接入 .....	27
4.6 本章小结 .....	27
5 总结与展望 .....	29
5.1 总结 .....	29
5.2 展望 .....	29
参考文献 .....	30
致谢 .....	31



## 插图清单

图 2.1 三次握手原理图 .....	4
图 3.1 数据库模型设计 .....	7
图 3.2 服务端模块划分 .....	8
图 3.3 客户端模块划分 .....	9
图 3.4 共享工程模块目录 .....	10
图 3.5 消息传送协议 .....	10
图 4.1 餐厅场景建模 .....	12
图 4.2 人物建模 .....	13
图 4.3 人形骨架绑定 .....	13
图 4.4 UI 界面 .....	14
图 4.5 游戏音频资源 .....	16
图 4.6 数据库表结构 .....	16
图 4.7 开始界面 .....	17
图 4.8 登录界面 .....	17
图 4.9 注册界面 .....	18
图 4.10 创建房间等待加入界面 .....	19
图 4.11 房间列表界面 .....	20
图 4.12 房间加入成功界面 .....	21
图 4.13 等待房主开始游戏界面 .....	22
图 4.14 开始游戏倒计时界面 .....	23
图 4.15 玩家移动场景 .....	24
图 4.16 制作菜品场景 .....	25
图 4.17 游戏结算面板 .....	25
图 4.18 个人信息更新界面 .....	26
图 4.19 动画状态机 .....	27

## 表格清单

表 4.1 面板相对路径格式 .....	14
----------------------	----

# 1 绪论

## 1.1 课题背景

自上个世纪 50 年代，电子游戏诞生于美国以来，随着社会现代化、电子化程度的不断推进，物质层面日益得到满足的基础上，人们对于精神娱乐层面的需求不断提高，网络游戏基于其故事性、社会性和交流特性，已经成为当代人群休闲娱乐的主要方式之一。近年来，游戏产业市场规模与用户规模逐年增长，特色游戏市场逐步发展，行业整体向好发展。可以说在人们的日常生活中，除了工作、学习，玩自己喜欢的游戏正在成为一种时尚。而开发出一款大众喜欢的、高品质的休闲互动游戏，让人们在工作学习之余，享受游戏的快乐，是一款游戏真正成功的意义。

随着游戏行业的发展，游戏引擎也随之而生，市面上常见的游戏引擎有 Unity、Cocos、Laya、UE4 等，游戏引擎凭借着自身集成了游戏开发中需要的功能而受到游戏开发者的喜爱，大大提高了开发者的效率。而 Unity 作为一款专业的游戏引擎，拥有内置集成开发平台，并且具备跨平台功能，能高效开发中小型游戏的特点，在众多的游戏引擎中脱颖而出，它为游戏开发提供了一种全新的解决方案。正是因为 Unity 的这些优点，毕业设计决定基于 Unity 来开发一款餐厅经营类游戏。

## 1.2 研究意义和目的

餐厅经营类游戏主要考验的是玩家对时间的分配能力，游戏中玩家需要根据客人进入餐厅的先后顺序来合理的为客人服务。目前在各大游戏平台上热卖的《胡闹厨房》凭借着清新的卡通画风、“上手即玩”的游戏特性，博得了广大玩家的好评。而本次毕业设计也是从《胡闹厨房》中得到的灵感，希望能够通过结合本科阶段学到的知识，开发出一款属于自己的餐厅经营游戏。

游戏的开发包括了客户端与服务端的网络通讯、数据存储、游戏模型动画以及 UI 美术资源的制作，其用到的技术和方法本身就是对本科学习成果的检验。

## 1.3 主要工作

本游戏使用 Unity2020.3.3f1c1 版本引擎进行开发，游戏场景建模、游戏人物动画设计采用 Blender 软件绘制，UI 界面使用 UGUI 设计，脚本使用 C#语言编写，

使用的是 SQL 数据库来进行数据的存储，使用 Socket 来进行网络通讯。

在游戏开发初期，搭建好服务端和客户端的框架，实现两者之间的信息传递，然后使用 Blender 进行游戏场景的建模以及游戏人物动画的设计。在游戏基础框架搭建好之后，根据游戏进行的顺序进行开发，在开发的过程中及时进行测试和修复，并使用 git 来进行仓库管理，方便开发过程中回退等操作。

#### **1.4 本文组织结构**

本文介绍了一款基于 Unity 实现的联机的餐厅经营游戏，整体的论文规划如下：

第一章为绪论部分，介绍了课题背景，研究意义和目的，论文的准备工作和论文架构，以及游戏中的工作内容。

第二章介绍了开发中使用到的技术和方法，包括 Unity 游戏引擎，编程语言 c#，网络通讯，MySQL 数据库以及 Blender3D 建模软件。

第三章进行了游戏的需求分析以及模块设计介绍。

第四章从资源准备、登陆和注册、房间管理、游戏游玩、以及动画和音频接入这几个方面详细介绍了游戏的实现过程。

第五章为总结与展望部分，对本次游戏开发所完成的工作进行总结，分析不足之处以及可改进部分，并对游戏后续的开发进行展望。

## 2 相关技术和方法

### 2.1 Unity 引擎

#### 2.1.1 Unity 引擎特点介绍

Unity 是由 Unity Technologies 开发的一个能轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎。其功能强大、界面简洁、可移植性的特性，受广大游戏开发者的青睐。

#### 2.1.2 动画系统

Unity 拥有一个丰富而复杂的动画系统，支持导入动画或者在 unity 内直接创建动画，动画将会被编入一个类似于流程图的结构化系统中(Animator Controller)，Animator Controller 充当“状态机”，负责控制当前播放哪组动画，这种控制可由动画参数来进行控制，从而实现在代码中修改参数来实现对动画切换的控制。

#### 2.1.3 资源商店

Unity 的资源商店有着种类众多的游戏资源，包括 3D、2D、插件、音频、模版等，本次课设使用到的 DOTween 插件和天空盒就来源于资源商店，这大大节约了开发者搜集和准备资源的时间，同时对初学者也十分友好，初学者只需要专注于自己感兴趣的方向，其他方面的资源可以在资源商店直接进行获取。

### 2.2 c#语言

C#是由 C 和 C++衍生出来的一种安全的、稳定的、简单的、优雅的面向对象编程语言。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET 开发的首选语言。因此，Unity 为了方便和跨平台，选择了 c#作为主要的开发语言。

### 2.3 网络通讯

游戏采用 TCP/IP 协议来进行网络通讯，以实现客户端与服务端之间的消息传递。

### 2.3.1 IP 和端口号

IP 地址：用来标志网络中的一个通信实体的地址。通信实体可以是计算机，路由器等。

端口号：IP 地址用来标志一台计算机，但是一台计算机上可能提供多种应用程序，使用端口来区分这些应用程序。端口是虚拟的概念，并不是说在主机上真的有若干个端口。通过端口，可以在一个主机上运行多个网络应用程序。端口范围 0-65535, 16 位整数。

### 2.3.2 TCP 协议的三次握手

在 TCP/IP 协议中, TCP 协议提供可靠的连接服务, 采用三次握手建立一个连接。

第一次握手：建立连接时, 客户端发送 syn 包 ( $\text{syn}=j$ ) 到服务器, 并进入 SYN\_SEND 状态, 等待服务器确认。

第二次握手：服务器收到 syn 包, 必须确认客户的 SYN ( $\text{ack}=j+1$ ), 同时自己也发送一个 SYN 包 ( $\text{syn}=k$ ), 即 SYN+ACK 包, 此时服务器进入 SYN\_RECV 状态。

第三次握手：客户端收到服务器的 SYN + ACK 包, 向服务器发送确认包 ACK ( $\text{ack}=k+1$ ), 此包发送完毕, 客户端和服务器进入 ESTABLISHED 状态, 完成三次握手。

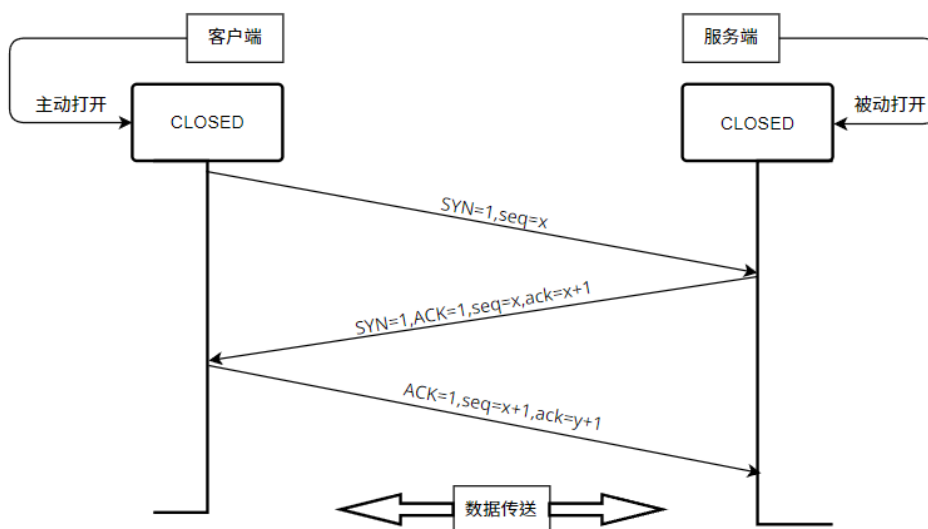


图 2.1 三次握手原理图

## 2.4 MySQL 数据库

数据库是计算机应用系统中的一种专门管理数据资源的系统。数据有多种形式，如文字、数码、符号、图形、图像及声音等，数据是所有计算机系统所要处理的对象。我们所熟知的一种处理办法是制作文件，即将处理过程编成程序文件，将所涉及的数据按程序要求组成数据文件，再用程序来调用，数据文件与程序文件保持着一定的关系。

游戏里使用数据库来储存玩家的账户信息，通过在服务器工程中引用 MySQL 中的 Mysql.Data 文件，即可使用 MySQL 语句进行数据库表的增删改查的操作。

## 2.5 Blender 游戏建模

Blender 包含了完整的多边形建模、曲线、曲面建模功能，修改器、雕刻、贴图展 UV 设定材质、烘焙场景、骨骼绑定、动画创作、粒子效果、物理仿真模拟、光线追踪引擎（Cycles）、相机追踪、图像后期合成等丰富、强大的功能。

游戏中使用 Blender 来制作餐厅的游戏场景，并利用 Blender 里面的人体骨骼来绑定模型来制作动画。

## 2.6 本章小结

本章介绍了毕业设计里面用到技术和方法，使用 Unity 游戏引擎来制作游戏的客户端，采用 C#语言，TCP/IP 来进行客户端与服务端之间的网络通讯，MySQL 作为数据库来存放游戏中的数据，Blender 软件进行游戏建模。下一章将介绍系统的需求分析和设计。

## 3 需求分析与设计

### 3.1 需求分析

#### 3.1.1 游戏资源

游戏需要准备以下资源：

3D 美术资源：餐厅场景模型、玩家人物模型、人物行走动画需要用 Blender 工具来进行制作。

2D 美术资源：开始界面、登陆界面、注册界面、房间列表界面、房间界面、游戏进行中界面、结果结算界面，这些界面通过 Unity 中的 UGUI 来进行制作。

音频资源：背景音乐、鼠标点击音效、菜品制作音效、得分音效。使用 Adobe Audition 音频软件来进行录制

#### 3.1.2 数据存储

游戏设计为一款可以远程联机的游戏，需要用数据库储存每名玩家的账户信息和战绩，方便游戏中对数据进行访问。因此，需要用数据库创建一张 user 表来储存账户名（username）和密码（password）和一张 result 表来储存玩家游戏总场数（totalcount）和最高得分（maxscore）。其中 result 需要设计一个外键 userid 与 user 表的 id 关联，使得每条战绩记录都与玩家账户相对应。

#### 3.1.3 服务端数据处理

为实现远程联机功能，需要设计一个服务端来同时处理多个客户端发来的请求，在服务器端进行数据处理，按照消息协议将处理后的数据返回到对应的客户端，以实现各个客户端的正常运行以及数据的同步。

同时，数据库也是由服务端来进行管理，包括数据的查询、添加、修改，都是在服务端进行操作。服务端在处理好数据后需要将数据按照协议规则进行打包，发送给客户端使用。

#### 3.1.4 客户端逻辑处理

客户端需要直接处理与玩家之间的交互。客户端在检测到玩家的操作输入后应及时进行响应，比如玩家的登陆、注册、创建房间、控制玩家移动、退出游戏等操作都需要客户端来进行处理，客户端如果需要服务端的数据支持，则需要向服务端



发起请求，获得服务端的响应后，将返回的数据解析后更新玩家的客户端数据。

### 3.1.5 网络通信

为了实现客户端之间的信息同步，应实现客户端与服务端的网络通讯。游戏中的数据处理统一放在服务端来进行处理，客户端在需要数据时，统一通过网络通讯向服务端发起数据请求，从而获得服务端发来的响应，因为在服务端是同一份数据，所以在不同客户端的消息是保持信息同步的。

客户端与服务端之间能实现通讯的前提是需要指定一份消息协议，这样消息才能通过这个规则进行打包以及解析处理。

## 3.2 游戏模块设计

### 3.2.1 数据库

根据数据库的需求分析，数据库模型设计如下：

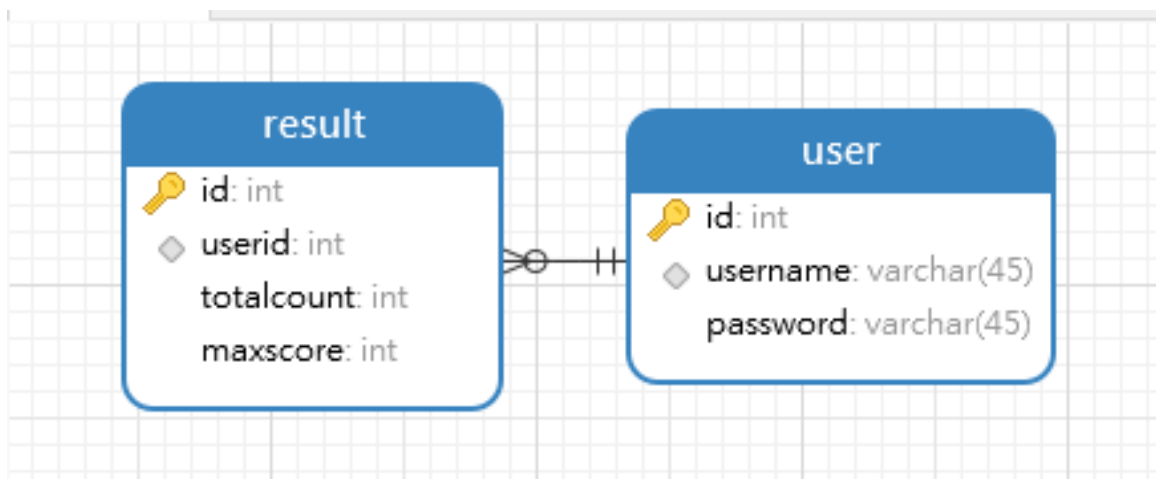


图 3.1 数据库模型设计

需要设计用户表和战绩表两张表。用户表用来储存用户名和密码，战绩表则储存对应的用户 id、总场数和最高得分。

### 3.2.2 服务端

根据游戏需求，服务端的模块划分如下：

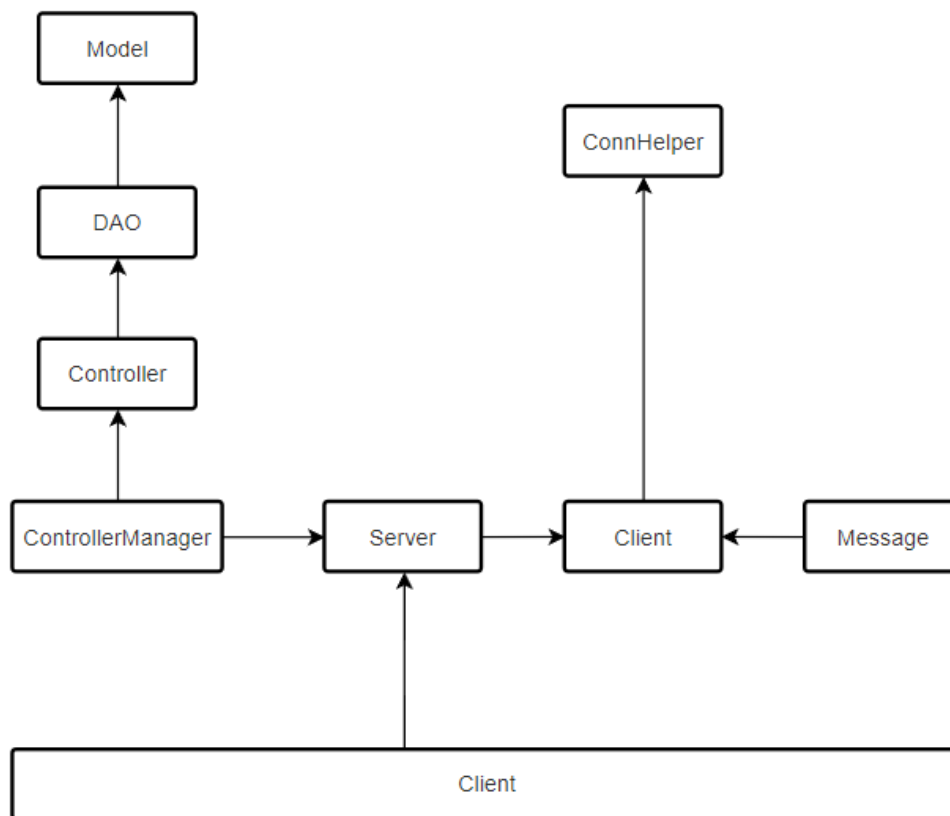


图 3.2 服务端模块划分

- (1) Server：用来创建 TCP 中的 Socket，监听客户端的连接。
- (2) ConnHelper：工具类，用来连接数据库。
- (3) Controller：不同的 Controller 分别负责处理客户端的不同类型请求，每当客户端的请求发送给 Server，Server 会调用相应的 Controller 来进行处理。
- (4) ControllerManager：管理所有 Controller。
- (5) Model：与数据库中的表对应，一个 Model 对应数据库中的一个表。
- (6) DAO（Data Access Object）：数据访问对象，用来操作数据库。
- (7) ConnHelper：用于将服务端与数据库进行连接。
- (8) Client：当某个 Client 向 Server 请求连接时，server 会为这个 Client 单独创建一个 Client 来进行 Client 和 Server 之间的收发消息。为每一个客户端都创建一个与服务端的连接，这样做能提高消息传送的效率。

(9) Message: 管理服务端客户端之间传送的消息, 包括消息的解析、读取以及打包等操作。

### 3.2.3 客户端

根据游戏需求, 客户端的模块划分如下:

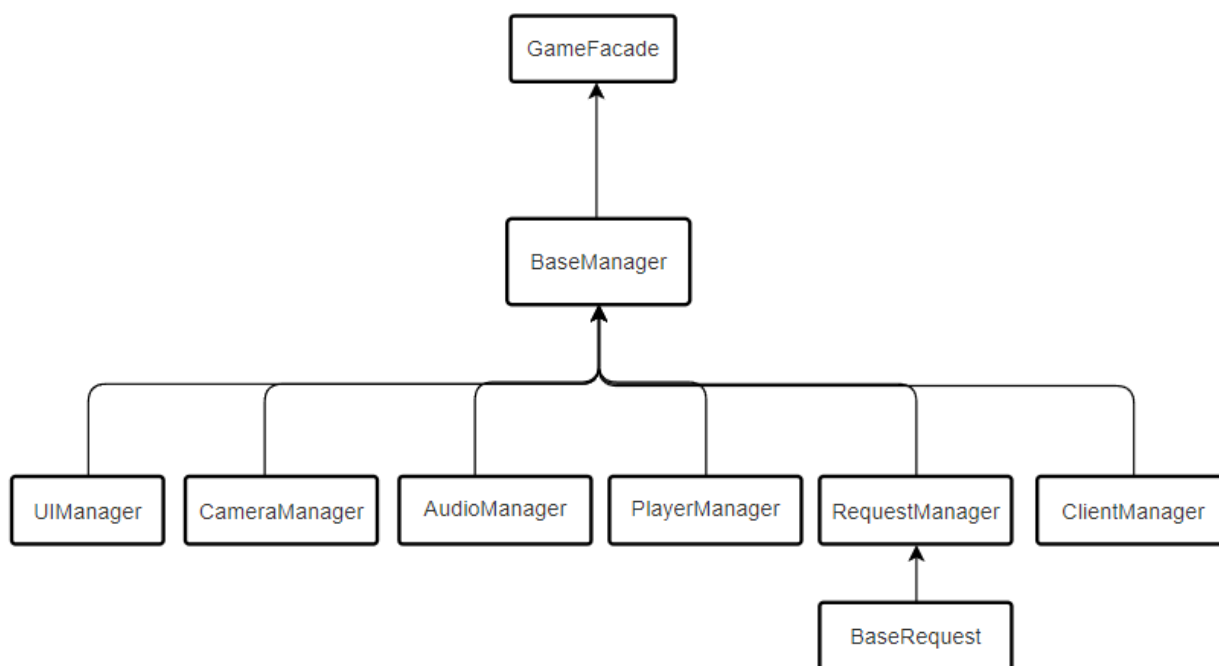


图 3.3 客户端模块划分

(1) GameFacade: 管理所有的 Manager, 作为不同 Manager 之间互相访问数据和方法的中介。

(2) UIManager: 管理游戏中的所有面板 UI。

(3) CameraManager: 管理游戏中的相机移动。

(4) AudioManager: 管理游戏中的音效的播放。

(5) PlayerManager: 管理游戏中的玩家, 包括玩家的生成以及初始化等操作。

(6) RequestManager: 储存和管理所有的 request 请求并对 request 进行处理。

(7) BaseRequest: 所有 Request 类的父亲, 用于存放 Request 类公用的属性和方法。

(8) ClientManager: 用来管理客户端跟服务器端的 Socket 连接。

### 3.2.4 网络通讯

#### (1) 创建共享工程

需要创建一个名为 Common 的共享工程，模块设计如下：

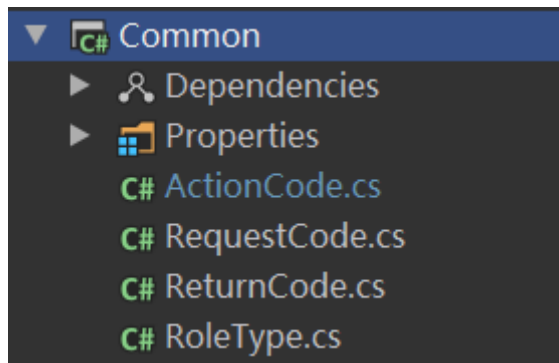


图 3.4 共享工程模块目录

其中 ActionCode 类通过枚举来储存所有的 ActionCode 类型；RequestCode 类通过枚举来储存所有的 RequestCode 类型；ReturnCode 类通过枚举储存客户端与服务端的响应类型；RoleType 类通过枚举来储存所有的角色类型。

#### (2) 设计消息协议

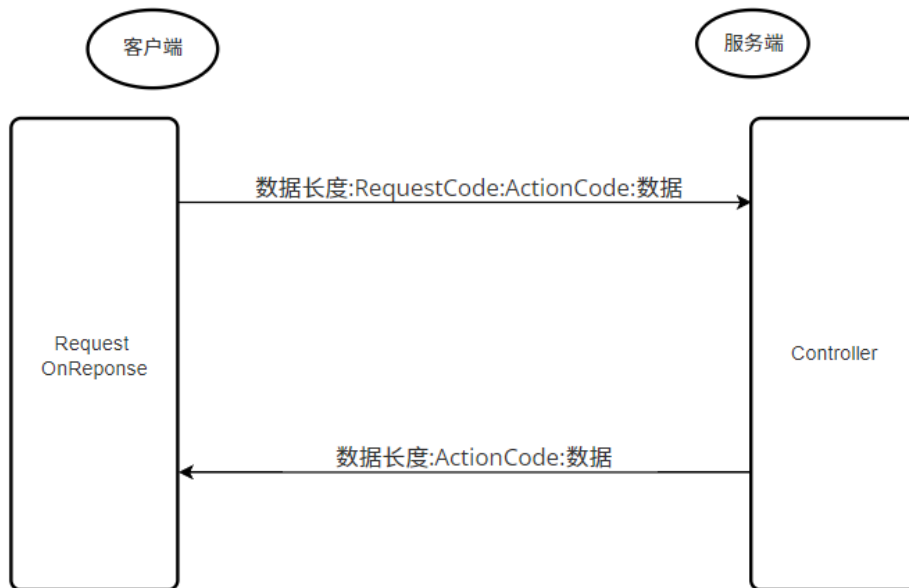


图 3.5 消息传送协议

通过服务端和客户端都有引用的共享工程，从而实现消息协议：在客户端与服务端之间进行消息通讯时，需要给消息制定规范：项目中将客户端向服务端发送的请求消息分为四段：数据长度 + RequestCode + ActionCode + 数据，首部用一个字节

的空间来存储要发送的消息总长度，接着是表示 Controller 类型的 RequestCode，然后是在 Controller 中需要执行的方法名 ActionCode。最后的具体需要传送的数据，例如登录操作时，客户端需要想服务器发送用户输入的用户名和密码来进行核验，其中的用户名和密码就是具体要发送的数据。ActionCode 同样也代表了客户端的响应函数名，在客户端使用 RequestManager 将所有的 ActionCode 用字典保存起来，因此服务端只需要传送 ActionCode，然后通过字典查找就能找到客户端的响应函数，从而服务端发送的消息只需要 ActionCode，而不需要传送 Controller 的类型 RequestCode 了，这是客户端与服务端两者的设计区别。

### **3.3 本章小结**

本章介绍了游戏的需求和设计。除了准备游戏需要的资源外，为了实现游戏的联机，游戏需要开发客户端与服务端两个工程，服务端需要接收客户端传送过来的消息并将消息转发给其他客户端，以实现不同客户端的消息同步。为了实现客户端与服务端的消息传递，需要制定消息的打包协议，方便消息的解析和数据获取。下一章将详细介绍系统的实现。

## 4 游戏的实现

### 4.1 资源准备

#### 4.1.1 3D 模型以及动画

考虑到项目开发时间不多，3D 建模采用的是 LowPoly 风格，在 Blender 中制作好模型后，搭配上 Unity 的渲染系统，虽然模型做的比较简易，但并不会降低玩家的游戏体验，因为 LowPoly 风格有着自己独特的美感，让游戏场景有种温馨的氛围。下面展示的是餐厅和人物的模型。

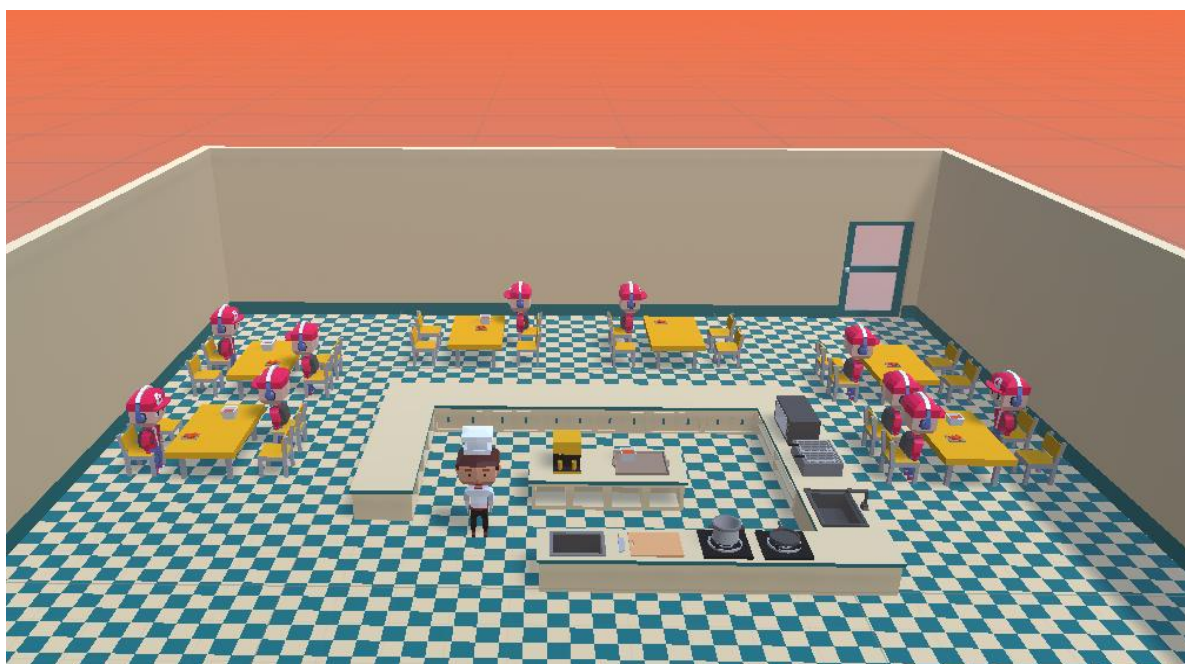


图 4.1 餐厅场景建模



图 4.2 人物建模

使用 blender 自带的人形骨架，将人形骨架绑定在人物模型上后，就可以通过控制骨架的位置来操作模型的动作，然后通过设计每一帧的动作，最后进行动画的播放时，每一帧的动作就会连贯的动起来，形成动画。

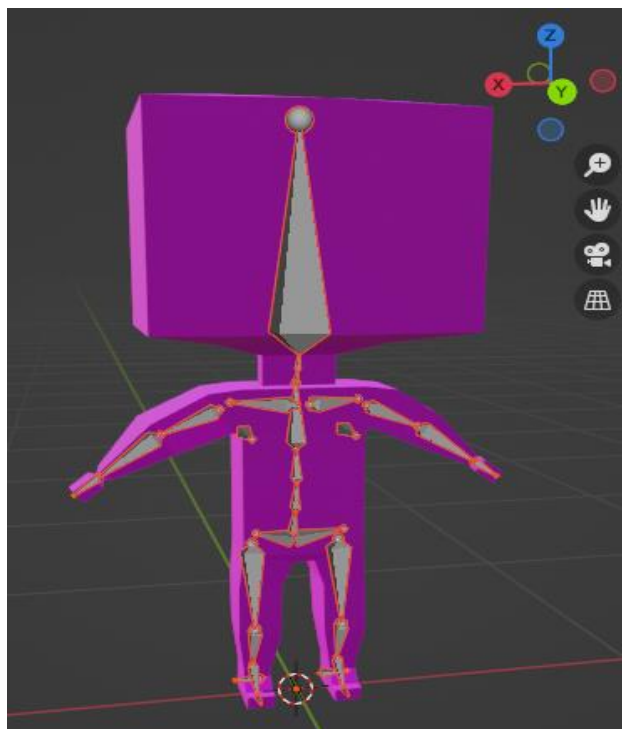


图 4.3 人形骨架绑定

4.1.2 UI 界面制作和管理

下面是使用 UGUI 制作的主要游戏 UI 界面：



图 4.4 UI 界面

Unity 中制作好后，可以拉入资源文件夹中作为预制体来进行加载。为方便项目中实例化面板，项目将界面的实例化方法放入了 UI 面板框架（UIManager 类）中进行统一管理，并使用 json 文件来储存所有面板预制体的相对路径，方便实例化面板时通过路径进行访问。下面为面板相对路径的格式。

表 4.1 面板相对路径格式

panelTypeString	path
Message	UIPanel/MessagePanel
Start	UIPanel/StartPanel
Login	UIPanel/LoginPanel
Register	UIPanel/RegisterPanel
RoomList	UIPanel/RoomListPanel
Room	UIPanel/RoomPanel
Game	UIPanel/GamePanel



UIManager 面板通过对 json 文件的读取，可以根据传入的面板类型，来得到需要实例化的面板来进行返回。

为了更好的管理游戏中的所有 UI 面板，UIManager 类来管理所有面板的进入和弹出，使用的是堆栈结构，因为游戏界面与堆栈的“先进先出”特性相符。把游戏中所有的 UI 界面用堆栈来储存，当有界面需要显示时，则将该页面进行入栈操作，而游戏中始终是显示堆栈最上面的界面，这样该页面因为在堆栈最上层，正好是需要展示的界面；而当页面需要返回时，则将页面从堆栈中弹出即可，这样上个被 Push 的界面就会位于堆栈的最上层，从而显示该页面，也就实现了返回上一个界面的功能。

对于 UI 界面的入场和出场动画，采用的是 DOTween 插件，使用该插件可以实现代码中平滑的控制场景中的 UI 的 Transform，从而可以平滑的控制 UI 的位置大小变化而实现用代码控制动画的进行。至于动画的调用时机，项目中将所有的 UIPanel 都继承自 BasePanel 类，BasePanel 中添加了 OnEnter（页面被显示出来时调用）、OnPause（页面暂停时调用）、OnResume（页面继续时调用）、OnExit（页面退出时调用）四个函数作为 UI 面板的生命周期，这四个函数统一由 UIManager 类来进行调用，而动画的入场动画和出场动画，分别在 OnEnter 和 OnExit 时进行调用即可，从而实现页面在进入时播放入场动画，页面消失时播放出场动画。

4.1.3 音频录制

以下是为游戏配置的音频资源，使用 Au 软件进行录制。

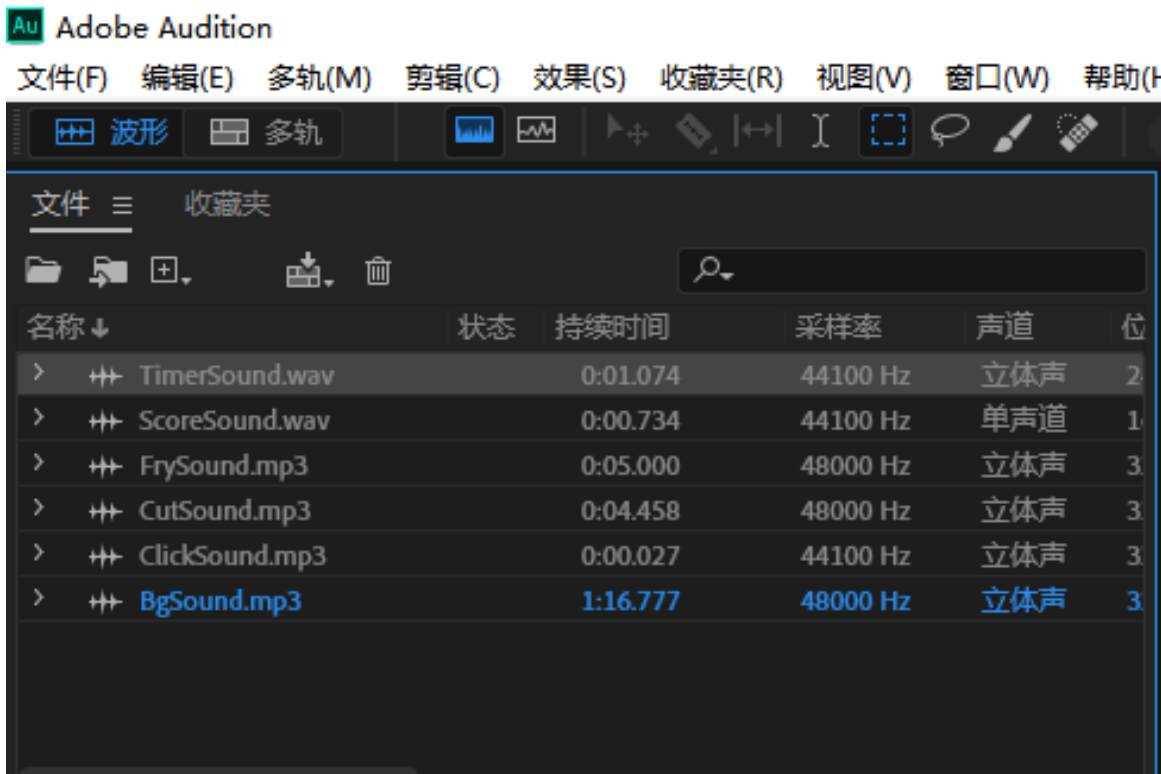


图 4.5 游戏音频资源

4.1.4 数据库创建

根据游戏需求，使用 MySQL 创建了游戏的数据库，数据库中创建了名为 user 和 result 的两张表，user 表来存放所有玩家用户名和密码，result 表用来存放玩家的进行游戏的总场数和最高得分，创建的两张表如下图所示。

result @gamedb (MySQL) - 表					user @gamedb (MySQL) - 表		
文件	编辑	查看	窗口	帮助	文件	编辑	查看
开始事务	文本	筛选	排序	导	开始事务	文本	筛选
id	userid	totalcount	maxscore		id	username	password
1	1	1	14		1	1	1
2	2	1	14		2	2	2
3	7	1	19		3	3	3
					4	lala	123
					5	4	4
					6	huhu	1
					7	???	1

图 4.6 数据库表结构

## 4.2 登陆和注册

### 4.2.1 登陆

进入游戏后，在开始界面右下角是一个登陆按钮，当玩家点击开始按钮后，会进入登陆面板。



图 4.7 开始界面

登录面板如下图所示，玩家可以选择输入用户名和密码进行登陆。



图 4.8 登录界面

对于登陆操作，在客户端，需要监听玩家输入的用户名和密码，在客户端可以对玩家的输入格式进行判断，如果玩家的输入格式有误则直接在客户端提示玩家重新输入，否则将玩家输入的数据进行打包发送给服务器处理。服务器在收到客户短发来的消息后，解析出玩家输入的用户名和密码，服务器则会在 user 表中进行查找，若不存在，则向客户端返回失败信息，若查找成功，则会在 result 表中获取该玩家的战绩，将该玩家的战绩打包返回给客户端。客户端在收到服务端的响应后，会对返回的消息类型进行判断，通过返回的结果控制界面是显示登陆成功还是失败。如果登陆成功，将会进入房间列表界面。

#### 4.2.2 注册

玩家选择注册按钮将进入注册界面进行新用户注册。



图 4.9 注册界面

对于注册操作，在客户端，需要监听玩家输入的用户名和密码，在客户端可以对玩家的输入进行检测，如果玩家输入为空或者两次密码输入不统一，则会直接在客户端提醒玩家重新输入，否则将玩家输入的数据进行打包发送给服务器处理。服务器在收到客户短发来的消息后，解析出玩家输入的用户名和密码，然后在 user 表中进行查找，如果存在同名，则向客户端返回失败信息，否则在 user 表中新增一条用户，向客户端返回成功信息。客户端在收到服务端的响应后，会根据返回的消息控制界面 UI 提示注册成功或者注册失败。

## 4.3 房间管理

### 4.3.1 创建房间

在登陆成功后, 玩家将会进入房间列表界面, 如下图所示, 房间列表左侧是个人信息界面, 包括玩家 id、总场数和玩家的最高得分。右侧则是当前可加入的房间列表。玩家可以点击右下角的创建房间按钮进行房间创建。

如果点击创建房间按钮, 客户端将会给服务端发起创建房间的请求。服务端在收到客户端发来的创建房间请求后, 会在 roomList 房间列表中新增一个 Room, 并向发来请求的客户端返回角色类型 (RoleType = Chef1), Chef1 角色类型表示当前客户端是房主。客户端在收到服务端创建房间成功的响应后, 就会控制房间面板入栈, 并将当前玩家的信息写入房间面板中, 队友的房间面板则显示等待玩家加入。



图 4.10 创建房间等待加入界面

### 4.3.2 加入房间

点击左下角的刷新按钮，右侧的房间列表会进行刷新，刷新后右侧房间列表将获得当前存在的房间，玩家可以点击房间列表中的任意房间中的加入按钮进行加入。



图 4.11 房间列表界面

点击刷新按钮后，客户端会通过 Request 类向服务端发起获得当前房间列表信息的请求。在服务端，当前所有的房间都在 Server 类里面进行管理，在收到客户端发来的房间列表获取的请求后，服务端会将 roomList 的信息进行打包返回给客户端。客户端在收到服务端返回的房间列表信息后，会根据数据打包的规则进行解析，然后将解析的数据写入房间列表面板中，从而使当前房间列表信息刷新。

在点击需要加入的房间后，客户端会将玩家的点击的房间的房主 id 进行打包发送给服务端，向服务端发起加入房间的请求。服务端在收到客户端发来的加入房间请求后，会在 roomList 中通过传送来的 id 进行查找，如果没有查找到房间，则返回给客户端该房间不存在的信息，否则将该客户端加入到查找到的房间中，并且将有玩家加入的消息广播给房主，让房主所在的客户端更新房间信息。同时也会给发送加入房间的客户端发回加入成功的响应。在收到服务端加入房间成功后的响应后，如果返



回的是加入成功，则会进入房间面板。房主所在的客户端在收到有玩家加入的响应后，会进行房间页面的更新。从而房主与非房主都会显示同样数据的房间信息面板。如果有玩家点击了退出房间按钮，则会向服务器端发起退出房间的请求，服务器端接收到退出请求后，判断如果是房主退出，则直接将房间销毁；如果是非房主退出，则将该玩家从房间的玩家列表中移除，处理完后将结果返回给客户端进行页面更新。



图 4.12 房间加入成功界面

#### 4.4 游戏游玩

玩家进入房间后，点击开始按钮后，游戏将会进入 3 秒倒计时，倒计时结束后，玩家可以通过键盘的上下左右来控制玩家移动，点击鼠标左键来进行菜品制作，制作完成后，菜品会朝着鼠标点击的方向发射，如果菜品到达顾客所在的餐桌即可得分。如果玩家完成了指定的菜品数量，则游戏结束，界面会弹出结算面板。

开始按钮点击后，客户端会向服务端发送开始游戏的请求。服务端在收到开始游戏的请求后，首先会判断发来开始游戏请求的是不是房主，如果不是房主，则会返回开始游戏失败的响应信息给客户端，反之，如果房主，则会向房间的其他玩家发送游戏开始的响应，同时将游戏开始成功的响应发送给房主。客户端在收到服务端返回的消息后，如果返回开始游戏失败，则会在游戏界面中提示“您不是房主，无法开始游戏！”。如果返回开始游戏成功，则会进入游戏界面。而房间的其他玩家也会收到服务端发来的开始游戏的广播消息，从而进入游戏界面。同时，在指定的位置生成玩家。在进入游戏界面时，相机上挂载的游戏脚本会控制相机移动，使相机从 UI 面板上跟随玩家聚焦到当前玩家，让玩家始终位于视野中央。

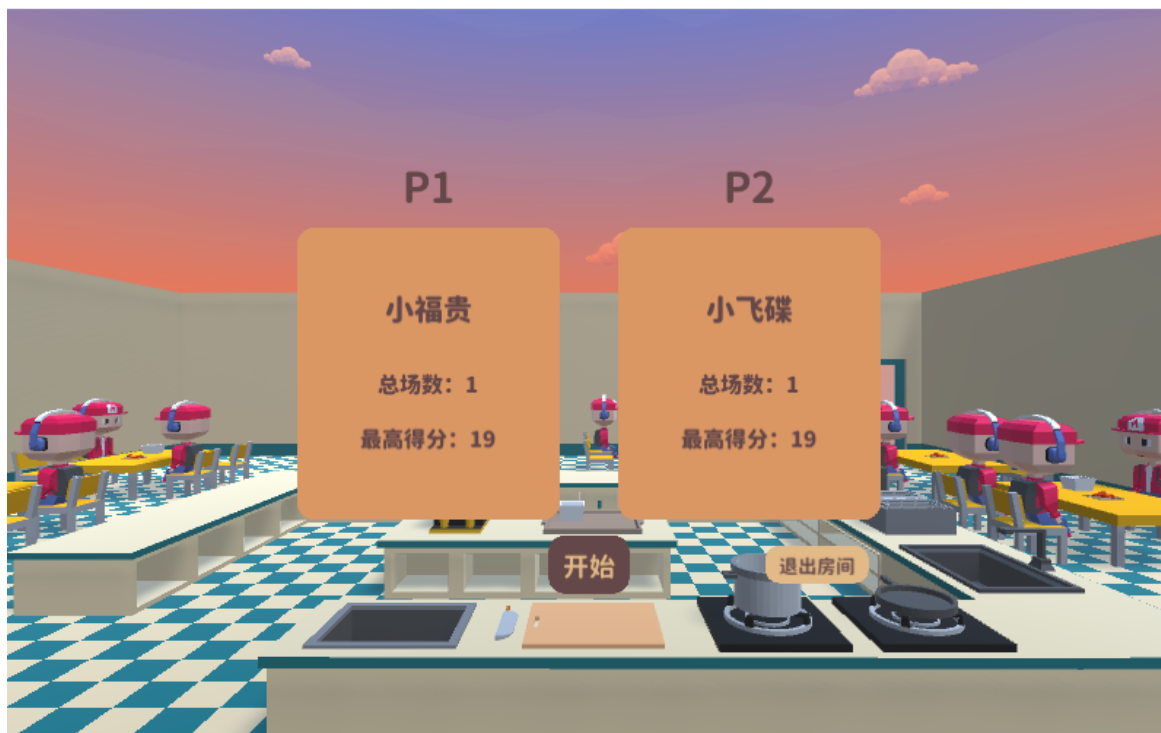


图 4.13 等待房主开始游戏界面

服务器端 CameController 在执行开始游戏的方法中，在确认房主开始游戏的同



时，也会开启一个 3 秒的倒计时，并将倒计时的时间数值打包发送给房间里的所有客户端。房间的客户端在收到服务端发来的时间数值后，会将获得的倒计时数值写入倒计时面板中并进行展示，从而房间内的玩家都会显示 3 秒的倒计时，同时在显示倒计时的时候播放倒计时音效。

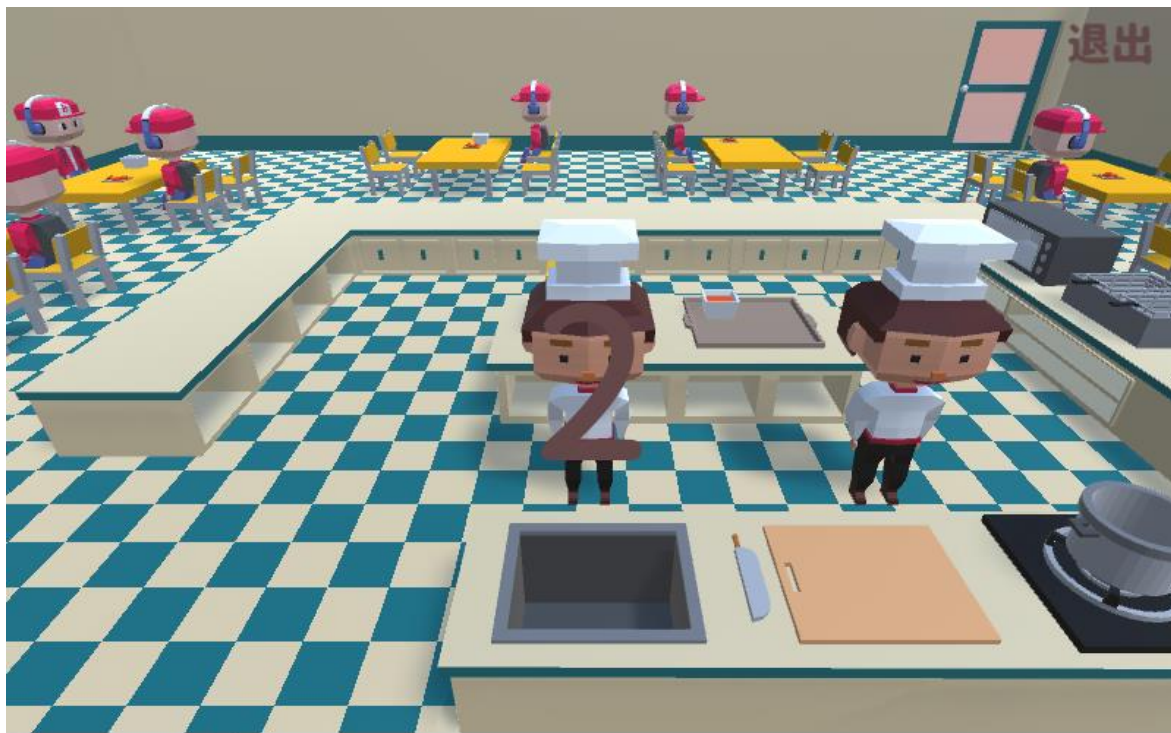


图 4.14 开始游戏倒计时界面

服务端在 3 秒的倒计时结束后,会给房间内的所有客户端发送开始游玩的消息。房间内的客户端在收到开始游玩的消息后,客户端会给当前玩家添加控制移动和制作菜品的脚本,从而使得玩家能通过键鼠输入来控制玩家移动和制作菜品。客户端通过在脚本的生命周期函数 `FixUpdate()` 函数中监听玩家的移动输入,在检测到玩家的移动输入时,除了控制当前客户端的玩家移动外,还会向服务端发起玩家移动的请求,以通过服务端将该玩家的移动信息转发给房间里的另一名玩家,另一名玩家所在的客户端通过服务端发来的消息来控制该名玩家的移动,从而实现在另一个客户端也能看到该名玩家进行移动了。



图 4.15 玩家移动场景

客户端在监测到玩家按下鼠标左键时,就会用脚本控制玩家制作菜品,并播放对应的制作音效,在制作完成后,生成的菜品会朝着之前玩家所点击的方位移动。在控制本地玩家制作菜品的同时,客户端还会向服务端发起制作菜品的请求,将本地的制作菜品信息传送到服务端,以通过服务端向房间内的另一名玩家发送该玩家制作菜品的信息,另一名玩家所在的客户端通过服务端发来的消息来控制该名玩家的制作菜品,从而实现房间中所有客户端菜品制作行为的同步。



图 4.16 制作菜品场景

做好的菜在移动的过程中，会对碰撞的物体进行检测，如果检测碰撞到了顾客，则表示上菜成功，同时播放得分音效，此时在客户端会随机生成（10， 20）内的得分，然后将分数发送给服务端进行得分处理，服务端在记录玩家上菜的总数达到要求后，就会向房间内广播游戏结束的信息，并将总得分返回给客户端，同时也会对战绩表进行更新。客户端在收到服务端发来的游戏结束消息后，会将服务端发来的分数写入得分面板，并在客户端进行展示。



图 4.17 游戏结算面板

在游戏中点击退出按钮或者给在结算面板点击任意位置都会退回到房间列表面板。而进入房间列表面板时，客户端通过之前收到的服务端的战绩数据对左侧个人信息面板进行更新。如下图所示。

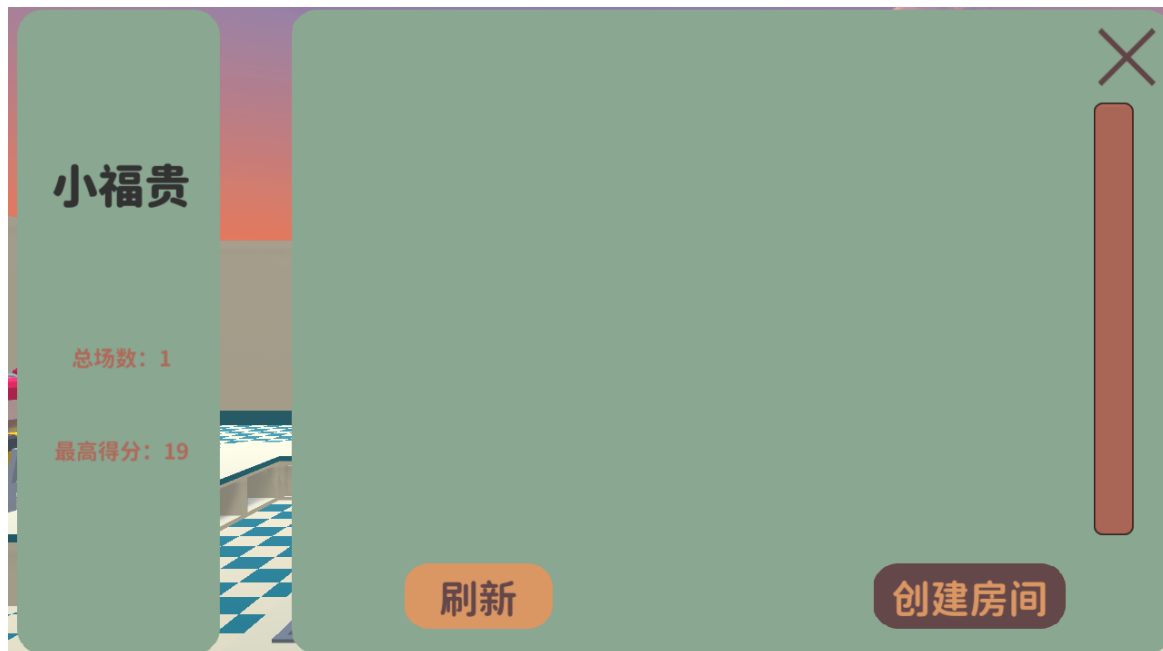


图 4.18 个人信息更新界面

## 4.5 动画和音频接入

### 4.5.1 动画接入

游戏中的动画接入与切换使用的是 unity 的动画状态机,玩家的动画包括 Idle、Walk、MakeFood 三种状态,使用动画状态机来进行动画的切换的话,则只需要给控制动画切换的参数进行赋值即可,如下面所示,如果需要 Grounded 切换到 MakeFood 状态的话,则只需要将参数 MakeFood 触发即可在 Grounded 状态下播放 MakeFood 动画。而 Grounded 是动画混合树,包含了 Idle 和 Walk 动画,它们的切换是根据 Forward 参数的大小变化来进行切换的,当 Forward 参数由 0 增长到 1 的过程中,玩家会从 Idle 状态逐渐切换到 Walk 状态,由 1 减少到 0 时,玩家会从 Walk 状态逐渐切换成 Idle 状态。而 MakeFood 和 Grounded 参数在脚本中会根据玩家的输入进行动态调整,而且实现了玩家的动画切换。

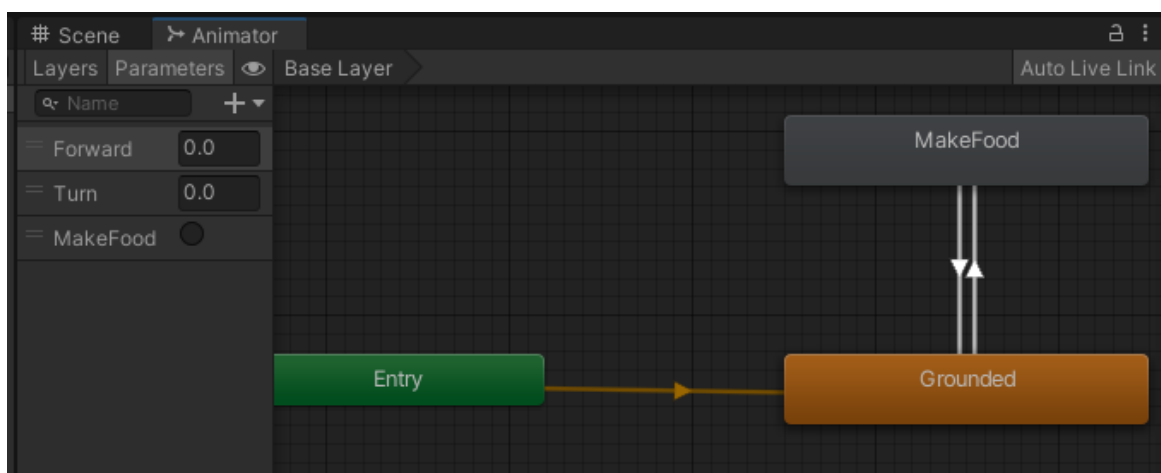


图 4.19 动画状态机

### 4.5.2 音频接入

游戏中的音效统一由 AudioManager 类来进行管理,其中的 PlayBgSound(string soundName)方法用来播放游戏的背景音乐,PlayNormalSound(string soundName)方法用来播放其他音效。因为背景音乐需要循环播放,所以将背景音乐的播放单独创建一个方法,避免与其他音效播放冲突。

## 4.6 本章小结

本章从游戏资源准备、登陆和注册、房间创建和加入、游戏游玩、以及动画和音



频接入这几个方面介绍了游戏是如何实现的。资源准备使用 Blender 软件来完成游戏中 3D 模型的制作以及人物动画的制作，使用 UGUI 来制作游戏中的 UI 面板，游戏中的音频资源由 Au 来进行录制。功能实现主要是按照游戏设计中客户端与服务端的模块划分来进行开发，不同模块负责不同的功能板块，模块与模块之间通过中间模块来统一的进行调用与赋值，这样可以大大降低代码的耦合性，方便后期的调整与修改，代码的结构也更加整洁。由于开发的是联机游戏，需要通过客户端向服务端发送当前玩家的状态，服务端收到同步消息后将处理转发给其他客户端，再由客户端来进行同步展示，从而实现房间中所有客户端的同步。动画的接入则是利用动画状态机来进行，通过修改状态机中的参数来实现动画与动画的切换。游戏中的音频则是由使用 AudioManager 类来统一管理。整体上达成了预期的需求，下一章将对本文进行总结与展望。

## 5 总结与展望

### 5.1 总结

本毕业设计就采用 Unity 引擎作为工具,开发了一款可以联机的餐厅经营游戏。在整体游戏的开发过程中,代码的整体结构和代码的具体实现都非常重要。游戏共创建了服务端、客户端、共享工程三个工程,服务端负责管理数据库,根据客户端的需求进行数据的计算和转发,客户端则根据服务端发送的数据来控制游戏中的数据变化,共享工程则是用来存放客户端和服务端共同使用的方法和变量。三个项目互相配合,实现了一款可以联机的餐厅经营游戏。

为了实现联机功能,服务端作为中介,负责接收客户端的状态并发送给房间内的其他客户端,实现房间内所有客户端之间的信息同步,以实现联机的效果。

在游戏的框架设计中,核心思想是用静态类来存放公用的变量和方法,也称“单例模式”。如果需要类与类之间方法的调用,则使用这个静态类作为中介,将方法放入静态类中,通过使用静态类来进行调用,这种做法大大降低了代码的耦合性,减少了代码结构调整时的工作量,代码的结构也更加整洁。

### 5.2 展望

目前项目还有以下部分需要优化:

游戏的玩法单一。目前游戏实现的功能是玩家制作菜品进行得分,而对于菜品的制作过程以及菜品的品类选择,都是后续优化中可以添加的功能。

客人的 AI 实现。目前游戏中顾客是没有 AI 行为的,后续可以给客人增加排队、入座等行为,可以大大增加玩家的游戏体验。

游戏特效还有待优化。某些画面并不都完美,特效表现不足,比如玩家的做菜过程中的特效过少,而做菜正是游戏中的主要玩法,后续可以对游戏特效进行优化。

## 参考文献

- [1] 路宜松. 基于 Unity 引擎的 2D 角色扮演游戏的设计与实现[D]. 沈阳理工大学, 2021. DOI:10.27323/d.cnki.gsgyc.2021.000043.
- [2] 章国雁. 基于 Unity3D 的多人在线游戏案例设计与实现[J]. 安徽水利水电职业技术学院学报, 2021, 21(03): 46-49.
- [3] Andrew Coggan. Unity Game Audio Implementation: A Practical Guide for Beginners[M]. Taylor and Francis: 2021.
- [4] Bai Ying. SQL Server Database Programming with Visual Basic.NET: Concepts, Designs and Implementations[M]. John Wiley & Sons, Inc.: 2020-06-10.
- [5] Thorn Alan. Blender to Unity Workflows[M]. CRC Press: 2018-09-01.
- [6] 史宝明, 贺元香, 李岚. Unity3D 随机寻路算法设计[J]. 吉林师范大学学报(自然科学版), 2022, 43(01): 128-133.
- [7] 陆春雨. 基于 Virtools 的双人联机合作游戏设计[D]. 北京林业大学, 2010.
- [8] 任胤颖. 电子角色扮演游戏审美研究[D]. 四川师范大学, 2021. DOI:10.27347/d.cnki.gssdu.2021.000707.
- [9] 朱杰. 基于 Unity3D 游戏人工智能的研究与应用[D]. 广东工业大学, 2020. DOI:10.27029/d.cnki.ggdgu.2020.000245.
- [10] 邓伟. Unity 3D 中单例模式的设计与应用[J]. 喀什大学学报, 2020, 41(06): 61-64. DOI:10.13933/j.cnki.2096-2134.2020.06.014.
- [11] 丁志耀. 网络游戏服务端设计[J]. 电子技术与软件工程, 2013(03): 30-31.
- [12] 吕艳阳. 多人联机对战游戏的设计与实现[J]. 电脑知识与技术, 2017, 13(34): 81-83+99. DOI:10.14004/j.cnki.ckt.2017.3834.



## 致谢

本论文是在张老师的指导下完成的，从论文的选题、开题答辩、中期答辩到现在的论文撰写，老师都进行了全面而细致的指导，在我毕业设计遇到困难时老师也给了我莫大的鼓励，让我沉下心来做，不要过于抓细节，首先把游戏做出来再考虑细节优化。正是有指导老师的帮助，让我少走了很多弯路。

另外，室友也给予了我很多帮助，在游戏开发过程中，遇到了困难，我会跟她们一起讨论如何解决，听取她们的意见和建议。还有我的家人，他们在生活工作上都给予了我无微不至的关心和支持，才有了我现在的论文研究成果。

最后，向在大学四年里陪伴我成长的家人老师还有同学们表示深深的感谢！

作者：欧林妍

2022 年 5 月 30 日