## 客户端架构说明



图1.客户端架构图

客户端分为7大模块，各自负责相应的逻辑处理与消息通信，各模块的功能职责具体如下：

**Manager：**

公用信息管理，包括4个脚本，其中

CommonInformation：包含与服务端一致的命令信息，以及用户UI显示的基本信息（血量、经验、金钱等）

CommonMethod：游戏脚本公用的方法

GameStart：用于将用户信息显示的UI控制，以及游戏结果显示

Switch：场景切换

**Enemy：**

敌人控制，包括4个脚本，其中

EnemyNetwork：从网络消息队列中获取怪物的游戏逻辑指令，分发给各怪物脚本进行怪物控制

NEnemyHealth：对怪物的生命进行控制，包括血条显示，以及动画转换

NEnemyMove：执行相应的移动操作

NEnemyShoot：执行怪物的攻击动作

**Player：**

玩家控制，包括6个脚本，其中：

MagicFire：用于控制魔法箭的攻击逻辑

PlayerHealth：用于控制玩家的生命值，受伤与死亡效果

PlayerMovement：用于控制玩家移动与动画，以及移动指令的发送

PlayerShooting：用于控制玩家射击，以及射击指令的发送

UIFollow：用于显示玩家的姓名

**Network：**

网络通信，包括3个脚本，其中：

Login：游戏登录界面，负责登录信息的发送

Regiser：游戏注册界面，负责注册信息的发送

NetworkHandler：网络通信主模块，负责游戏消息的发送与接收

**Camera：**

镜头调整，包括2个脚本，其中：

Aim：负责准心的绘制与对焦

MouseLookView：负责玩家的转向，镜头移动

**Trap：**

陷阱逻辑，包括3个脚本，其中：

TrapDamage：负责陷阱的触发，并发送相应的触发消息给服务端

TrapDetect：陷阱重叠放置检测，避免陷阱的重叠放置

TrapManager：负责两类陷阱的切换，购买，放置

**UI：**

UI控制，包括12个脚本，其中：

CreateRoom：负责房间创建消息的发送与显示

DropdownRoom：从服务器端获取所有的房间名

EnterRoom：负责进入相应房间的发送与显示

Multi：多人模式选择

OK：消息确认

Quit：用户退出游戏消息的发送与显示

QuitRoom：用户退出房间

ReStart：重新挑战的消息发送

Single：单人模式选择

TabSelect：tab进行UI按钮切换

UISwitch：各个Canvas间的切换

WaitingRoom：房间信息的更新

## 服务端架构说明



图2.服务端架构图

服务端分为3大模块，各自负责相应的逻辑处理与消息通信，各模块的功能职责具体如下：

**Common：**

公用信息与服务，包括6个脚本，其中：

Conf：包含与客户端一致的命令信息

Enemy：包括一个基类Enemy以及三个子类，Bear、Bunny和Ellephant，负责怪物的升级、陷阱伤害以及与玩家的各种交互操作

Events：消息类集合，负责各类消息的建立与识别

gameService：服务端较为重要的类，针对不同的消息进行不同的逻辑处理，包括用户登录、注册、房间创建、离开房间、游戏开始、游戏结束、陷阱购买等消息处理。

Pathfinding：负责自动寻路，将客户端导出的导航网格信息读取，利用A\*算法进行怪物的寻路，以及是否到达终点的逻辑判断

Player：用户类，包括用户基本的信息，升级以及陷阱购买处理

**Common\_server：**

游戏辅助，包括2个脚本，其中：

Database：游戏数据存储数据库，帮助获取用户数据，辅助进行登录、注册验证，以及用户信息存储

Timer：定时器类，添加了CallNumEvery类，用做每隔一段时间调用方法，可设置调用次数

**Server：**

服务器主要功能，包括2个脚本，其中：

Manager：负责游戏数据的映射管理

GameServer：游戏服务器主循环，处理用户各类消息以及网络连接的建立与断开，定时处理怪物逻辑，包括是否到达终点，是否追逐玩家，是否攻击玩家，是否死亡。游戏关卡的管理，包括怪物生成与销毁，游戏结果通知。