# 宠物小精灵对战系统 POTTLE 说明文档

目录

[宠物小精灵对战系统 POTTLE 实验报告 1](#_Toc48119897)

[一、 设计思路 1](#_Toc48119898)

[（1）、宠物小精灵 1](#_Toc48119899)

[（2）、用户与平台交互 1](#_Toc48119900)

[（3）、对战系统 2](#_Toc48119901)

[二、 模块划分 2](#_Toc48119902)

[三、 模块具体设计 2](#_Toc48119903)

[三、 数据结构及全局变量 5](#_Toc48119904)

[（1）、类定义 5](#_Toc48119905)

[（2）、重要数据结构 6](#_Toc48119906)

[（3）、重要全局变量 6](#_Toc48119907)

[四、 具体操作流程 6](#_Toc48119908)

[（1）具体流程功能介绍： 7](#_Toc48119909)

[（2）战斗操作示例 7](#_Toc48119910)

[五、 评价与改进意见 8](#_Toc48119911)

## 设计思路

### （1）、宠物小精灵

设计宠物小精灵基类，包括种类（力量型、肉盾型、防御型、敏捷型）、等级、经验值、攻击力、防御力、生命值、攻击间隔，以及战斗中临时的项属性等等变量供战斗使用；成员函数除了对应各种属性的getter和setter函数以外，还有攻击函数（虚函数，每个种类精灵进行不同的攻击）、被攻击函数，成长函数、图形化界面属性相关函数。

设计力量型、肉盾型、防御性、敏捷型四个子类，包括种族（也就是名称、如皮卡丘、喷火龙）属性；成员函数主要是改写基类中的攻击函数，返回信息函数（将获取到的信息拼接为字符串返回供集中输出）。

### （2）、用户与平台交互

平台采用socket通信，采用TCP方式建立连接，服务器端socket最多同时接受四个连接，server.exe作为POTTLE 对战平台的服务器，而客户端client.exe作为用户操作面板。平台在用户账号的基础上提供平台服务，以界面为流程单位供用户进行操作，每个用户可以注册唯一用户名的账号，并设置密码，在从服务器下线后可再次登录，注册成功的用户可以随机获得三只初始精灵，失去所有精灵则系统再随机分配已知初始精灵。用户在线时，也可以获取到其他用户的信息（具体在操作流程中介绍）。

设计用户类，包括用户名、密码、勋章、胜率，拥有精灵列表，登录状态，通信套接字，对战请求信息等等基本属性，成员函数包括登录操作相关函数，设置对战请求、精灵相关操作等等。

### （3）、对战系统

平台为用户提供总计四种对战方式，对战对象可选服务器或用户，对战模式可选升级赛和决斗赛。对象为服务器则可以由玩家挑选服务器精灵列表中的一只精灵与之对战，并选出自己的出站精灵；对象为用户则要等待服务器匹配两名选择了同样对战方式的玩家进行战斗。升级赛中，获胜精灵获得合理的经验值；决斗赛中，获胜方获得失败方出站精灵，若被服务器在决斗赛中击败，则由服务器在玩家精灵中随机挑选三只（玩家精灵不足三只则全部选中），由玩家在此之中挑选一个送出。

战斗系统在图形化界面和命令行界面同步进行战斗模拟，战斗由两只参赛精灵完成，战斗结果由精灵属性以及不确定因素决定（一定概率暴击和闪避）。战斗结束后，进行结算，战斗信息被返回到客户端。

## 模块划分

**主模块**：初始化及基本服务器线程创建

**等待客户连接模块**：接受客户端连接请求，为每一个客户端单独分配流程推进模块

**流程推进模块**：完成初始化，根据用户的输入和当前所处模式进行模式变化和信息发送（具体流程推进模式介绍在下方**四、具体操作流程**和文件夹内**流程图.png**中）

**对战匹配模块**：完成战斗双方的匹配

**战斗模拟模块**：完成平台战斗的模拟

## 三、模块具体设计

**主模块：**

**初始化及基本服务器线程创建**

{

初始化；

开启等待匹配对战线程；

开启等待客户端连接线程；

阻塞；

}

**等待客户连接模块：**

**接受客户端连接请求，为每一个客户端单独分配流程推进模块**

{

初始化；

While（true）

{

接受连接请求；

分配流程推进模块；

}

}

**流程推进模块：**

**完成初始化，根据用户的输入和当前所处模式进行模式变化和信息发送**

{

While（true）

{

长时间未收到信息退出；

收到信息；

If（当前模式是xxx）

{

进行操作；

变换模式；

Continue；

}

If（当前模式是xxx）

{

进行操作；

变换模式；

Continue；

}

……………………

}

}

**对战匹配模块：**

**完成战斗双方的匹配**

{

While（true）

{

for（用户列表头；不到尾；++）

{

If（用户有服务器对战需求）

{

战斗模拟；

}

}

for（用户列表头；不到尾；++）

{

If（用户有用户对战需求）

{

加入对战等候区；

}

If（对战等候区满两位选手）

{

对战模拟；

Break；

}

}

}

}

**战斗模拟模块：**

**完成平台战斗的模拟**

{

战斗初始化；

While（双方战斗临时生命值均大于零）

{

If（a精灵距离攻击剩余时间为0）

a攻击b；

if（b临时生命值为0）

break；

If（b精灵距离攻击剩余时间为0）

b攻击a；

}

战斗结算；

传输信息；

}

## 数据结构及全局变量

### （1）、类定义

**宠物精灵类：**

变量：精灵种类、等级、经验值、攻击力、战斗临时攻击力、防御力、战斗临时防御力、生命值、战斗临时生命值、攻击间隔、战斗临时攻击间隔、距离下一次攻击剩余时间、攻击是否被影响、防御是否被影响、攻击速度是否被影响、图像存储位置。

函数：构造函数、攻击函数（虚函数）、被攻击函数、经验增长、等级增加（虚函数）、战斗准备函数、攻击倒计时更新、setter：图像存储位置；getter：种族（虚函数）、等级、经验值、生命值、临时生命值、攻击力、临时攻击力、防御力、临时防御力、攻击间隔、临时攻击间隔、距离下次攻击剩余时间、图像存储位置、精灵战斗信息字符串、精灵信息字符串。

**派生精灵类（力量型精灵类、肉盾型精灵、类防御性精灵类、敏捷型精灵类）：**

变量：种族。

函数：构造函数、攻击函数、等级增加函数、getter：种族、精灵信息字符串、精灵战斗信息字符串。

**用户类：**

变量：用户名、密码、套接字、用户拥有精灵向量、在线状态、升级赛请求状态、决斗赛请求状态、对战对象选择、出战精灵编号、服务器出战精灵选择、获胜场次、总场次、与服务器决斗失败送出精灵选择列表。

函数：构造函数、验证密码函数、更新胜率函数、添加精灵、移除精灵、获得随机精灵函数、setter：登录状态、战斗模式、战斗对象、战斗选择信息；getter：所有精灵信息字符串、制定编号精灵、在线情况、用户名、战斗模式信息、套接字、精灵数量、战斗选择信息、战斗对象信息用户信息字符串、高等级精灵数目。

友元函数：int in\_lost\_select(std::vector<int> v, int)检查用户所输入编号是否在与服务器决斗失败送出精灵选择列表之中、User\* check\_username(std::string username)检查用户名是否已存在服务器列表中。

必要性：函数需要对用户的用户名、与服务器决斗失败送出精灵选择列表进行操作，从而进行编号选择可行性进行检查、用户名合理性检查。

**服务器操作类：**

变量：套接字。

函数：构造函数、getter：服务器监听套接字。

友元函数：int InitServer(SOCKET\* server\_socket)初始化套接字、void init\_images()初始化图形化界面图像、void init\_server\_user()初始化服务器用户、void server\_accept\_thread()服务器等待客户端连接请求、void server\_handle\_thread(SOCKET\* connect\_socket)服务器处理客户端操作（流程推进）、void wait\_for\_competitor()服务器匹配用户对战请求、int battle\_thread(User\* a, int a\_id, User\* b, int b\_id)服务器战斗模拟。

必要性：对套接字进行初始化、绑定等等SOCKET操作，此外对服务器用户、图形化界面的初始化也要用到套接字、在服务器等待客户端连接请求、处理客户端操作、匹配对战、战斗模拟也都需要与客户进行通信，因此要用到套接字。

友元函数：各种枚举类型以及布尔类型转字符串函数：std::stringto\_string(Poke\_type );

std::string to\_string(Strength\_poke\_race);

std::string to\_string(Health\_poke\_race);

std::string to\_string(Defense\_poke\_race);

std::string to\_string(Agility\_poke\_race);

std::string to\_string(bool);

必要性：进行SOCKET通信时要以字符串的形式进行信息传输，因此需要将一次发送的信息拼接为一整个字符串，友元函数在各种返回信息字符串的操作中提供信息构造辅助，有效精简代码。

### （2）、重要数据结构

extern std::vector<User\*> users;//用户向量，存储系统注册的用户

以向量线性存储系统的所有注册成功用户

extern std::vector<IMAGE\*>poke\_attack\_images;

extern std::vector<IMAGE\*>poke\_defense\_images;

extern std::vector<IMAGE\*>poke\_attack\_images\_cover;

extern std::vector<IMAGE\*>poke\_defense\_images\_cover;

以向量线性存储图形化界面的所有图像路径

### （3）、重要全局变量

extern std::mutex users\_mutex;//用户向量锁

多线程对用户向量进行操作，利用互斥量避免冲突

extern User\* server\_user;//服务器用户

与服务器对战，对战对象为特殊的服务器用户

……

各种图像对象

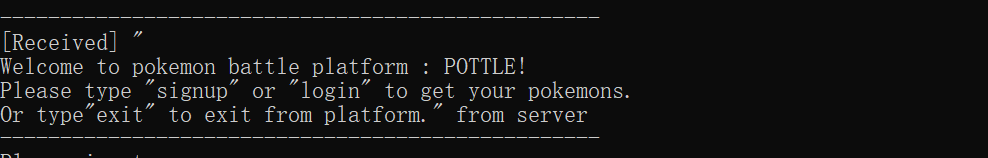
## 具体操作流程

要进行对战，请首先打开Pokemon.exe， 随后可以最多同时打开四个client.exe， 每个打开的client.exe相互独立。

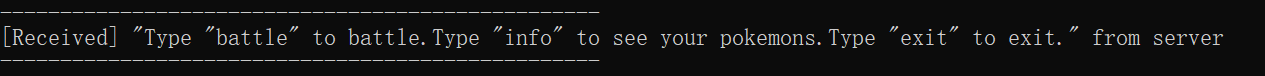
打开client.exe（客户端）后，看到“Please input”字样代表成功进入到POTTLE平台，请根据客户端收到的信息在客户端进行操作

### （1）具体流程功能介绍：

首先可以注册或登录

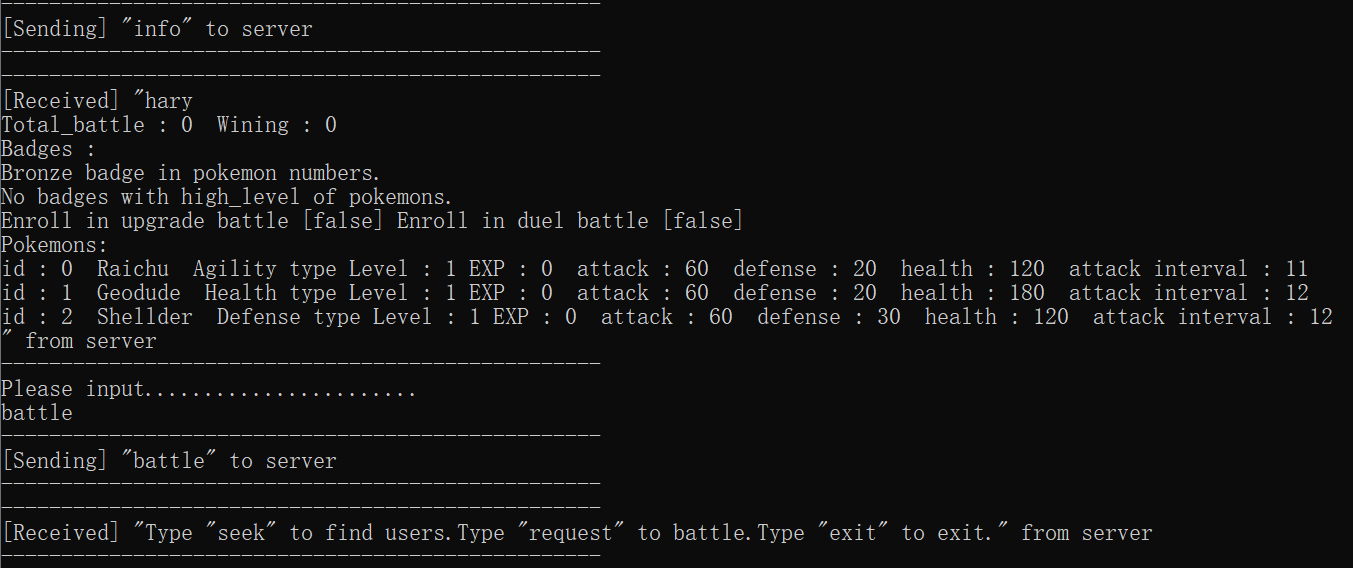


成功注册或登录后进入到**主界面，**



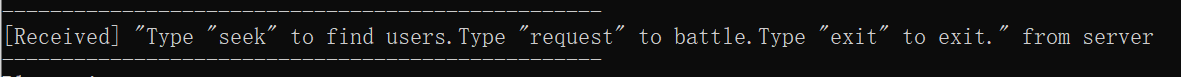
输入“battle”进入**对战操作界面**

输入“info”获取自己的信息和所有精灵

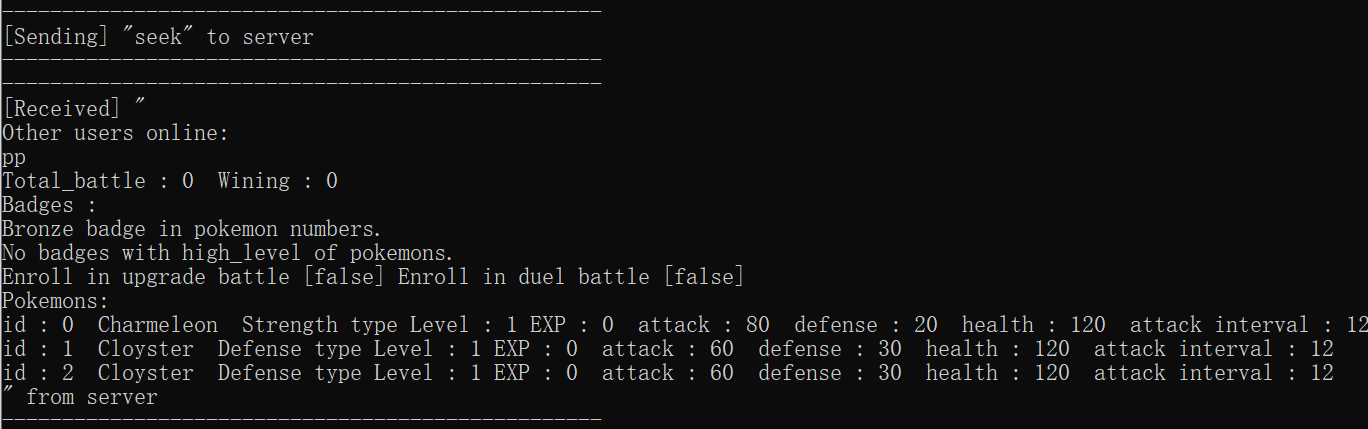


输入“exit”退出登录

进入**对战操作界面**



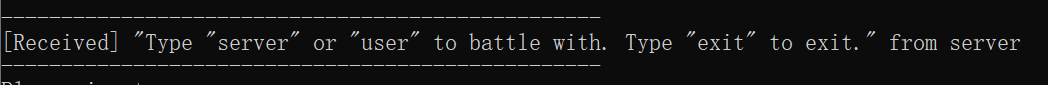
输入“seek”查询是否有其他用户在线（client.exe）以及对战请求



输入“request”进入**对战对象选择界面**

输入“exit”返回主界面

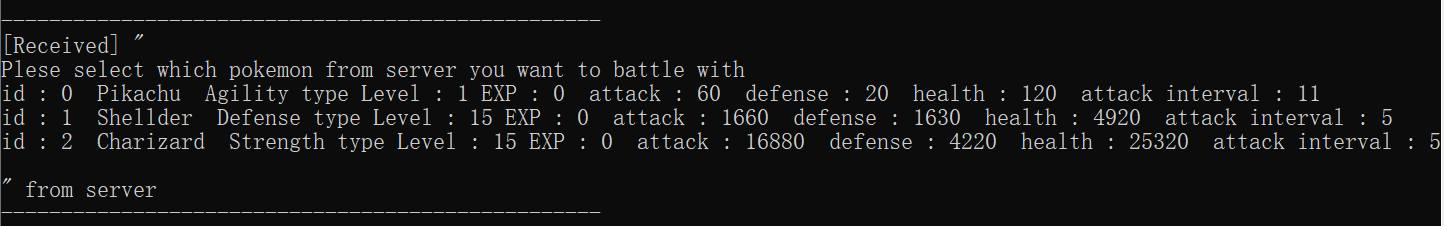
进入**对战对象选择界面**



输入“server”与服务器对战，进入**服务器选择界面**

输入“user”与其他用户对战（client.exe），进入**战斗模式选择界面**

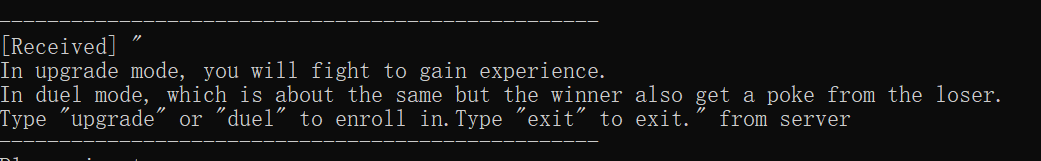
进入**服务器选择界面**



在服务器发送来的服务器精灵列表中选择自己要对战的精灵编号并发送

（发送对战服务器精灵编号后）

进入**战斗模式选择界面**



输入“upgrade”参加升级赛，胜者精灵获得经验值

输入“duel”参加决斗赛，

胜者精灵获得经验且获胜用户从失败用户手中得到击败的精灵

如果胜者为服务器，则向失败用户提供一个包含用户至多三个精灵的列表，有用户选择送出哪只精灵

输入“exit”返回**主界面**

（选择战斗模式后）

进入**出战精灵选择界面**



输入服务器发送来的用户精灵列表，发送一个出战精灵的编号

输入“exit”取消对战，返回**主界面**

（发送出站精灵编号后）

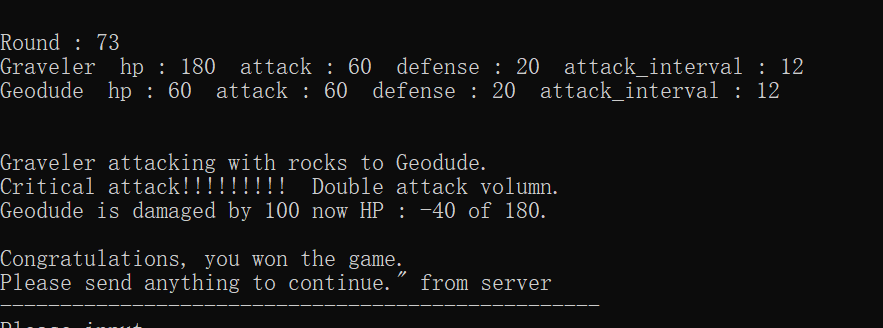
**进入战斗状态**，此时客户端不可以向服务器发送信息，在系统匹配了用户的对手后将开始进行对战，对战后对战信息将发送给参战者，随后根据收到的信息进行操作即可返回主界面。











### （2）平台操作示例

1.**与服务器升级赛**（胜者精灵获得经验值）

依次输入

signup harry 123 battle request server (从接收信息里选个编号) upgrade （从接收信息里选个编号）

2.**与服务器决斗赛**（胜者精灵获得经验且获胜用户从失败用户手中得到击败的精灵.如果胜者为服务器，则向失败用户提供一个包含用户至多三个精灵的列表，有用户选择送出哪只精灵)

依次输入

signup harry battle request server (从接收信息里选个编号) duel （从接收信息里选个编号） （看接收信息指示输入）

3.**与用户升级赛**（请打开两个客户端作为两个客户，除用户名外同样的输入）：

依次输入

signup （写个用户名） 123 battle request user (从接收信息里选个编号) upgrade （从接收信息里选个编号）

4.**与用户决斗赛**

依次输入

signup （写个用户名） 123 battle request user (从接收信息里选个编号) duel （从接收信息里选个编号）

## 评价与改进意见

在完成实验要求的基本功能的基础上，增加了用户之间的两种对战模式，实现用户之间的竞技性；此外增加了图形化界面显示功能，在命令行的战斗信息输出基础上，将战斗模拟用图像显示出来，直观地反映精灵对战的情况，同时将各精灵派生类的特殊攻击方法形象化展现出来。

POTTLE对战平台提供了一个比较稳定和成熟的对战系统，为用户提供宠物精灵对战的各种服务，全面的且有创新地完成了实验要求。

POTTLE对战平台在图形化界面方面只有对战模拟的场景展示，客户端与服务器平台的交互只能通过客户端基于命令行输入的SOCKET通信来完成，在未来的系统升级中可以加入图形化界面的交互，比如基于输入框、按钮等等来实现SOCKET通信交互，实现操作更加便捷的对战系统。