

网络文字对战游戏协议文档

Author:201220131胡增杰

Date:2023/4/16

1. 系统的功能和参与者

这个系统包括一个服务器端和多个客户端，多个客户端可以与这个服务器建立连接。

客户端的功能：

指令	功能
login	登录用户
logout	退出登录
view	查看当前所有在线玩家及其状态，Vacant表示在线非对战，Battling表示正处于对战中
request	向别的玩家请求对战
clear	清屏，并且打印指令的提示信息
exit	退出程序
yes/Yes/YES/y/Y	当有别人向本客户端请求对战时，输入这些指令中的任何一个，可以接受对战
no/No/NO/n/N	当有别人向本客户端请求对战时，输入这些指令中的任何一个，可以拒绝对战
r, s, p	当与人对战时的作战选择，分别表示rock, scissors, paper

服务器的功能：

1. 监控用户登录。当有用户登录时，检查用户名是否重复，如不重复则登录成功，如发生重复则登录失败。在服务器端界面显示登录成功或失败的提示信息，返回给客户端登录成功或失败的报文，向当前其他在线玩家通知该新用户的登录。
2. 监控用户退出登录。在服务器端界面显示退出登录成功或失败的提示信息，返回给客户端退出登录成功或失败的报文，向当前其他在线玩家通知该用户的退出。
3. 支持用户查看当前所有在线用户及其状态。当用户执行request时，服务器返回给客户端所有在线用户及其状态信息。
4. 处理用户间的对战请求。当有用户向其他用户请求对战时，交换双方的请求和应答信息，维护用户的状态。
5. 管理用户间的对战信息交换，并即时返回给双方对战结果的详细信息。

2.协议格式

服务器和客户端之间交换的数据报文只有一种统一的格式。该报文的长度固定为82个字节。

报文结构如下: (使用C语言代码展示出来)

```
1  #pragma pack(1)
2  typedef struct PKT{
3      uint16_t type; //2 bytes
4      union{
5          struct { //for 0x01xx, 0x02xx, 0x03xxx
6              char srcusername[NAME_SIZE]; //19 bytes
7              char dstusername[NAME_SIZE]; //19 bytes
8              struct { //6 bytes, for 0x03xx
9                  uint8_t srcopt; // 1-rock, 2-scissors, 3-paper
10                 uint8_t dstopt;
11                 uint8_t srcblood;
12                 uint8_t dstblood;
13                 uint8_t winlose; //result of this round, 1-win, 2-lose, 3-tie
14                 uint8_t gameover;
15             } battledetail;
16             char useless[36];
17         } info;
18         struct { //19+1+30*20 = 80 bytes, for 0x04xx,
19             char logusername[NAME_SIZE];
20             uint8_t usernum;
21             User users[USER_NUM];
22         } userlist; //for 0x04xx
23     };
24 } PKT;
```

各字段的详细含义如下:

type字段

(说明下标中client->server表示只在客户端发送给服务器是出现, server->client表示只在服务器发送给客户端时出现)

取值	含义
0x0101	client->server, 请求登录
0x0102	server->client, 登录成功
0x0103	server->client, 登录失败
0x0104	client->server, 请求退出登录
0x0105	server->client, 退出登录成功
0x0106	server->client, 退出登录失败
0x0201	client->server, 请求和其他玩家对战, 由对战发起方发送给服务器
0x0202	server->client, 服务器转发对战请求, 发送给对战请求的目的方
0x0203	client->server, 对战请求的目的方接受请求
0x0204	client->server, 对战请求的目的方拒绝请求
0x0205	server->client, 告知对战请求的发起方, 目的方接受你的请求
0x0206	server->client, 告知对战请求的发起方, 目的方拒绝你的请求
0x0207	server->client, 报错给对战请求方, 目的方不存在或者没有登录
0x0208	server->client, 报错给对战请求方, 目的方正在处于对战中
0x0209	server->client, 报错给对战请求方, 请求方没有登录
0x020a	server->client, 发送给对战请求的目的方, 通知其游戏开始, 和0x0205的报文同时发送
0x0301	client->server, 客户端发送给服务器端自己的对战选择
0x0302	server->client, 服务器返回给对战双方战斗结果
0x0401	client->server, 客户端想查看所有在线用户及其状态
0x0402	server->client, 服务器返回给客户端所有在线用户及其状态
0x0403	server->client, 服务器通知在线玩家有玩家登录
0x0404	server->client, 服务器通知在线玩家有玩家退出登录

info字段

该字段仅在type==0x01xx, 0x02xx, 0x03xx或者0x0401时才有效, 且info内的useless仅做填充目的, 并无实义。

当type==0x01xx时, srcusername, 表示客户端登录或者退出登录的用户的名称, 其他字段都无效

当type==0x02xx时, srcusername, 表示自己的用户的名称, dstusername表示对方的对手的名称

当type==0x03xx时, 各字段含义如下:

- srcusername和dstusername分别表示自己和对手的名字；
- srcopt和dstopt分别表示自己和对方的选择，1代表rock，2代表scissors，3代表paper；
- srcblood和dstblood分别是自己和对手当前血量数值，初始值为3，每次对战输者血量减1，赢者血量不变，平局双方血量都不变；
- winlose表示当前这一局对战中自己的胜负，1表示自己胜，2表示自己输，3表示平局；
- gameover表示和对手的整个对战是否结束，0表示没有结束，1表示对战结束且自己赢，2表示对战结束且自己输。

当type==0x0401时，srcusername表示当前用户的名称，也就是想要查看所有在线玩家及其状态的用户的名称。

userlist字段

该字段仅当type==0x0402, 0x0403, 0x0404才有效

当type==0x0402时，logusername没有被使用，usernum表示所有在线玩家的数量，userlist用来存储所有在线玩家及其状态，userlist是结构为User的数组，User结构的定义如下，name为玩家的名称，state表示玩家的状态，0表示不在线，1表示在线但是没在对战中，2表示在线且正在和别的用户对战。

```

1  #define NAME_SIZE 19
2  #pragma pack(1)
3  typedef struct User{ //20 bytes
4      char name[NAME_SIZE];
5      uint8_t state;
6  } User;

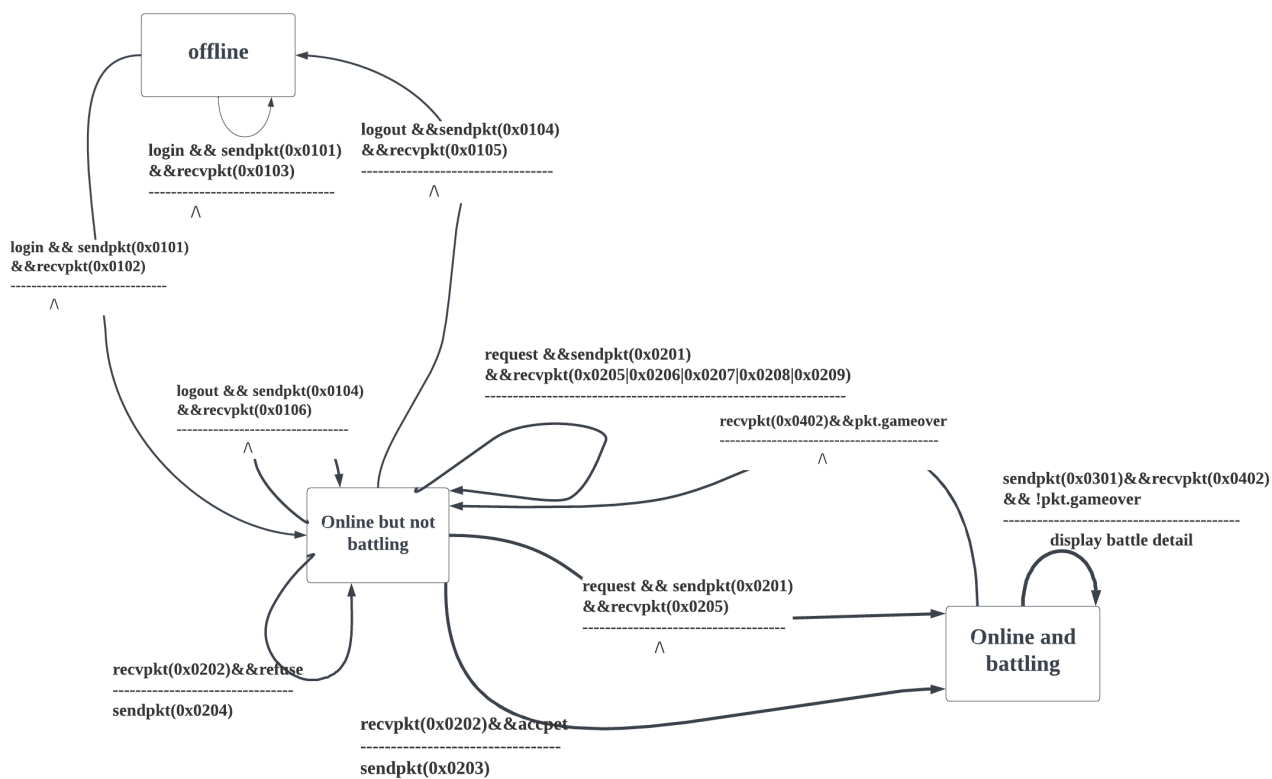
```

当type==0x0403,0x0404时，logusername表示登录或者退出的用户的名称。

3.协议运行图例

下面给出客户端运行这个协议时的状态转换图，由于服务器主要负责整体数据的维护，故不再给出状态图。

客户端运行这个协议时的状态转化如下：



说明：图中的sendpkt(0x0104),表示发送了一个type为0x0104的数据包给服务器，rcvpkt(0x0106)表示从服务器接收到了一个type为0x0106的数据包。对于每个状态的转化条件，虚横线“-----”上方的表示转换的触发条件，虚横线下方的表示转换时执行的动作。