

游戏服务器及通信协议

1、游戏服务器

网络版的游戏服务器是一个 Socket 通信服务器。Socket 也称为“套接字”，是一组基于 TCP/IP 协议的网络接口 API，可以通过双向的通信连接实现数据交换，其基本工作流程如下图 1。

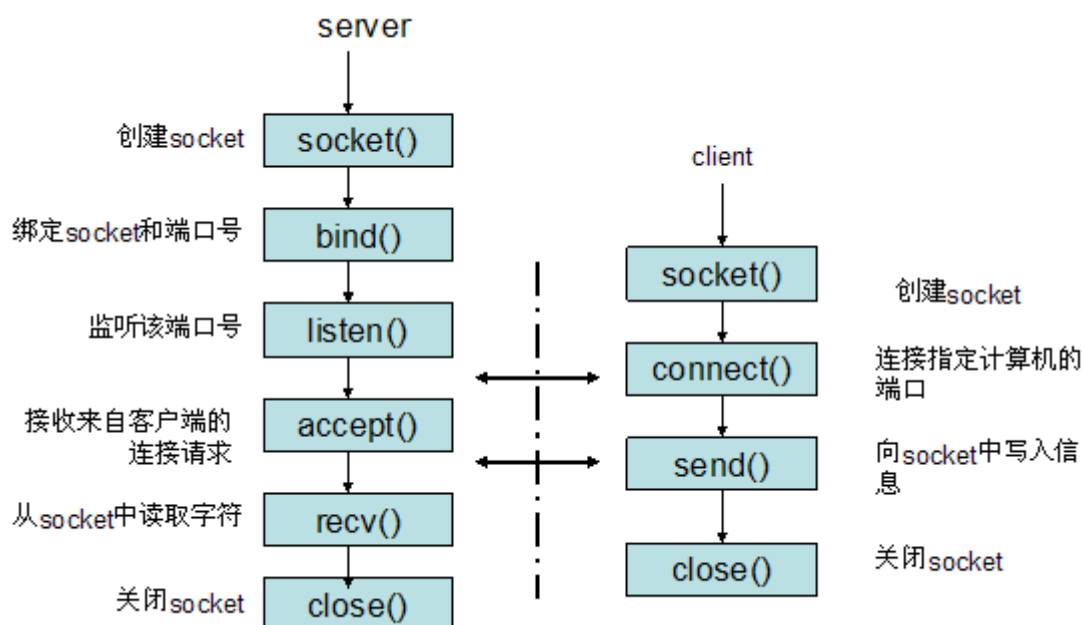


图 1 Socket 通信工作流程

运行服务器需要 Python3 环境。切换到 server 目录下，执行如下命令，启动游戏服务器。

```
python server.py
```

server 目录下的 config.txt 中保存了游戏服务器的配置信息，包括 IP 地址和 Socket 端口号，一回合等待最大时间等，可以根据需要进行修改。

2、通信协议

游戏通信协议使用 JSON（JavaScript Object Notation）数据格式。JSON 是一种轻量级的数据交换格式，易于阅读和编写，同时也易于机器解析和生成。JSON 采用完全独立于语言的文本格式，但是也使用了类似于 C 语言家族的习惯（包括 C、C++、C#、Java、JavaScript、Perl、Python 等）。这些特性使 JSON 成为理想的数据交换语言。

联网对战时，需要与对方客户端 ClientB 或者服务器 Server 交换的信息有：

- ✓ 向服务器声明自己处于等待配对状态，并请求配对同一游戏的对手。
- ✓ 当前回合对手的动作：棋子移动的轨迹（旧坐标->新坐标）。而棋盘的绘制和刷新、判定胜负、走棋规则等都是由游戏客户端来进行判定实现的。
- ✓ 向服务器主动声明退出本局游戏。

还有几种额外的信息需要发往服务器（可选），这些协议能够让服务器获得更加完善的信息：

- ✓ 游戏结束时，本客户端在当前游戏局的胜利（失败）信息。
- ✓ 其它信息等。

下面说明每种交互动作的过程：

2.1.向服务器请求加入游戏

向服务器申请加入游戏时，需要声明自己的游戏昵称。昵称为不超过 10 个字符的字符串，可由客户端自行决定是否显示己方昵称和对手昵称。

申请加入游戏时，向服务器 Server 发送如下格式的 json 信息，表示昵称为[小明]的同学希望加入对战，示例如下：

```
{
  "type": 0,
  "msg": {
    "name": "小明"
  }
}
```

客户端发送信息后，服务器返回以下格式信息：

1) 成功匹配到昵称为[张三丰]的对手，本局游戏的 ID 为 [690044a7-1400-4f2e-8003-4df850f21ac2]，并且是己方先手行动 side=0，side 为 0 或者 1，用于确定先后手和棋子颜色标记（约定红棋先手），think_time=30 表示一回合中一方最多使用时间是 30 秒，total_time=600 表示一局游戏中一方最多总共使用时间是 600 秒。示例如下：

```
{
  "status": 1,
  "counterpart_name": "张三丰",
  "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
  "side": 0,
  "think_time": 30,
  "total_time": 600
}
```

~~2) 等待超时，未匹配到对手，示例如下：←~~

```
+
"status": 0
+
```

2.2.棋子移动

当移动编号为 n 棋子的源坐标为(x1, y1)，移动后坐标为(x2, y2)时，处于动作回合的客户端 ClientA 应该主动向服务器 Server 发送如下格式的 json 信息，示例如下：

```

{
  "type": 1,
  "msg": {
    "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
    "side": 0,
    "num": 5,
    "src": {
      "x": x1,
      "y": y1
    },
    "dst": {
      "x": x2,
      "y": y2
    }
    "exp": "8 - 1 * 3"
  }
}

```

说明：其中的 game_id 和 side 取值同前，num 是所移动棋子的编号（示例为 5），src 和 dst 中分别表示棋子移动前后的坐标，exp 是棋子在进行单跨移动时所使用的四则运算表达式，不是单跨移动时可以为空字符串。

服务器 Server 转发 json 信息中的 **msg 部分内容** 给当前处在不处于动作回合的客户端 ClientB，ClientB 应该准备接收。

2.3.退出游戏

在匹配完成后的游戏过程中（根据客户端功能和执行者的交互动作确定），处在动作回合的客户端可以向服务器 Server 发送叫停和主动退出本局游戏的 json 信息。

1) 叫停游戏，当满足叫停规则时，客户端可以主动叫停游戏，此时向服务器发送的消息示例如下，**注意叫停后判断游戏胜负结果由客户端自行完成**。

```

{
  "type": 2,
  "msg": {
    "request": "stop",
    "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
    "side": 0
  }
}

```

2) 主动退出游戏，主动退出即代表认输，此时需要向服务器发送的消息示例如下：

```
{
  "type": 2,
  "msg": {
    "request": "quit",
    "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
    "side": 0
  }
}
```

说明：退出游戏时必须同时发送本局游戏的 ID 信息和己方的 side 值，叫停游戏和主动退出游戏的区别在于，叫停游戏的 request 取值为"stop"，主动退出游戏的 request 取值为"quit"。如果退出游戏的动作发生在游戏匹配完成之前，游戏的 ID 信息和己方的 side 值取空值。

服务器 Server 收到处在动作回合的客户端 ClientA 发送的退出游戏消息后，向当前处在不处于动作回合的客户端 ClientB，发出如下格式的退出游戏 json 信息：

```
{
  "status": 2,
  "request": "stop",
  "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
  "side": 0
}
```

说明：其中 request 为退出游戏的原因，也可以取值为"quit"。

客户端收到该退出游戏的消息后，先确认游戏的胜负，应该发送一条如下格式的消息回复服务器 Server（side 值是己方被分配的值）。然后，客户端应断开与服务器之间的 Socket 连接。

```
{
  "type": 3,
  "side": 0
}
```

2.4.服务器通知超时消息

在游戏对弈过程中，由于单步超时或一局总时间超时原因，客户端会接收到服务器如下格式的通知超时信息：

```
{
  "status": 3,
  "game_id": "690044a7-1400-4f2e-8003-4df850f21ac2",
  "side": 0
}
```

说明：side 值代表导致单步超时或一局总时间超时的一方，即为输的一方，例如 side=0 表示先手方（即红方）超时。

同样，客户端收到该超时消息，确认游戏的胜负后，应该发送一条如下格式的消息回复服务器 **Server**（**side** 值是己方被分配的值）。然后，客户端应断开与服务器之间的 **Socket** 连接。

```
{  
  "type": 3,  
  "side": 0  
}
```

若在实际过程中需要希望增加自己的协议，请及时提出。