

# 1.资源限制

单方面程序最大可用 CPU 为 5 个核，最大可用内存为 10GB。超过则视为程序异常。

# 2.编码格式

报文采用 json 协议组成的字符串格式，**json 内容+换行符\n 为一个通信报文，不要遗漏\n**，否则会被认为消息不完整，最终判定超时(**超时时间 1 秒**)。需要注意的是，**客户端不要对输出做缓冲，如有 flush 之类的接口尽量每次发送完应答都调用下**，以避免客户端默认的输出缓冲而产生超时的情况。

# 3.比赛环境

硬件：

公司开发机：CPU：12 核，内存 32G，硬盘 400G，显卡：无，**不连接外网**。

系统：

OS 内核：Linux TENCENT64.site 3.10.104-1-tlinux2，

发行版：Tencent tlinux release 2.2 (Final)

建议大家也使用 tlinux2.2 的机器来开发。

开发语言不限制，通过脚本拉起程序，只要支持在后端能拉起来跑，通过标准输入输出进行协议交互就行。

第三方库：

所有第三方库版本采用 Tencent tlinux release 2.2 发行版提供的默认的第三方库以及版本。平台侧不帮助安装任何其他第三方库。若需要调用第三方库，则在程序包中自行打包或预先静态编译好。

特别声明：

参赛提供的程序包需要**离线运行**，严禁程序通过网络调用内网/外网远程接口。

已经支持的执行环境

GCC 4.8.5

Python 2.7.5(通过 python 调用)

Python 3.4.5(通过 python3 调用)

Java 1.8.0

PHP 5.4.16

Go 1.6.3

Perl 5.16.3

Node.js v8.12.0

## 4.SVN 提交格式

请把代码和可执行程序交到你的 SVN 下的 trunk 目录下，在 trunk 目录下需要包含(文件严格区分大小写)：

make.sh，编译脚本，有些语言如 python 不需要编译，写一个空文件即可

PlayerName.txt (内容为战队的名字)，**#建议提供，方便前端展示**

run.sh **用于执行程序。**

... 其他文件和目录。

组委会通过 svn 拉取完代码之后,通过普通用户(无 root 权限)执行./make.sh 进行代码编译,编译完成之后务必确保这个目录中的 run.sh 可以拉起程序(还是这个普通用户),否则会被判负,后果自负。

## 5.对战平台使用说明

对战平台由 c++开发,需要用 tlinux2 内核的机器来执行。

1. 将压缩包放入任意目录,然后解压 tar zxvf robotserver.tgz,解压之后会得到三个文件:

|---bin #二进制程序目录

|---robotclient #非常简单的参赛跑测程序,供运行参考

|---run.sh #启动脚本,供参考

|---robotserver #对战平台

|---safeshell #不用管

|---log #对战的过程日志

2. 执行 cd bin; ./robotserver [玩家 1 目录] [玩家 2 目录]

示例: ./robotserver . .

a)robotserver 接受 2 个参数,分别是两个参与 PK 的玩家程序的目录, **目录下必须要有**

**一个 run.sh 的运行脚本**。因为本目录下有一个简单的参赛程序 run.sh,所以参数一和参数二都用.当前目录。

b)这 2 个参数目录可以根据需要换成任意其他路径,只要路径下包含可执行文件 run.sh。

c)若玩家程序与 robotserver 正常通信,且初始化机器人位置无误,则对战开始后,前台会显示出地图。若初始化错误,未显示地图,则请查看 log 目录下的 robotserver 日志。



#### 4. 对战过程展示

a) 访问开发网的机器：<http://9.77.5.36> 可以查看对战过程，进入后点击 **开始游戏**，会弹出对话框，选择对战平台执行完生成的流水文件 **oplog\_时间戳\_uuid.log** 即可展示。

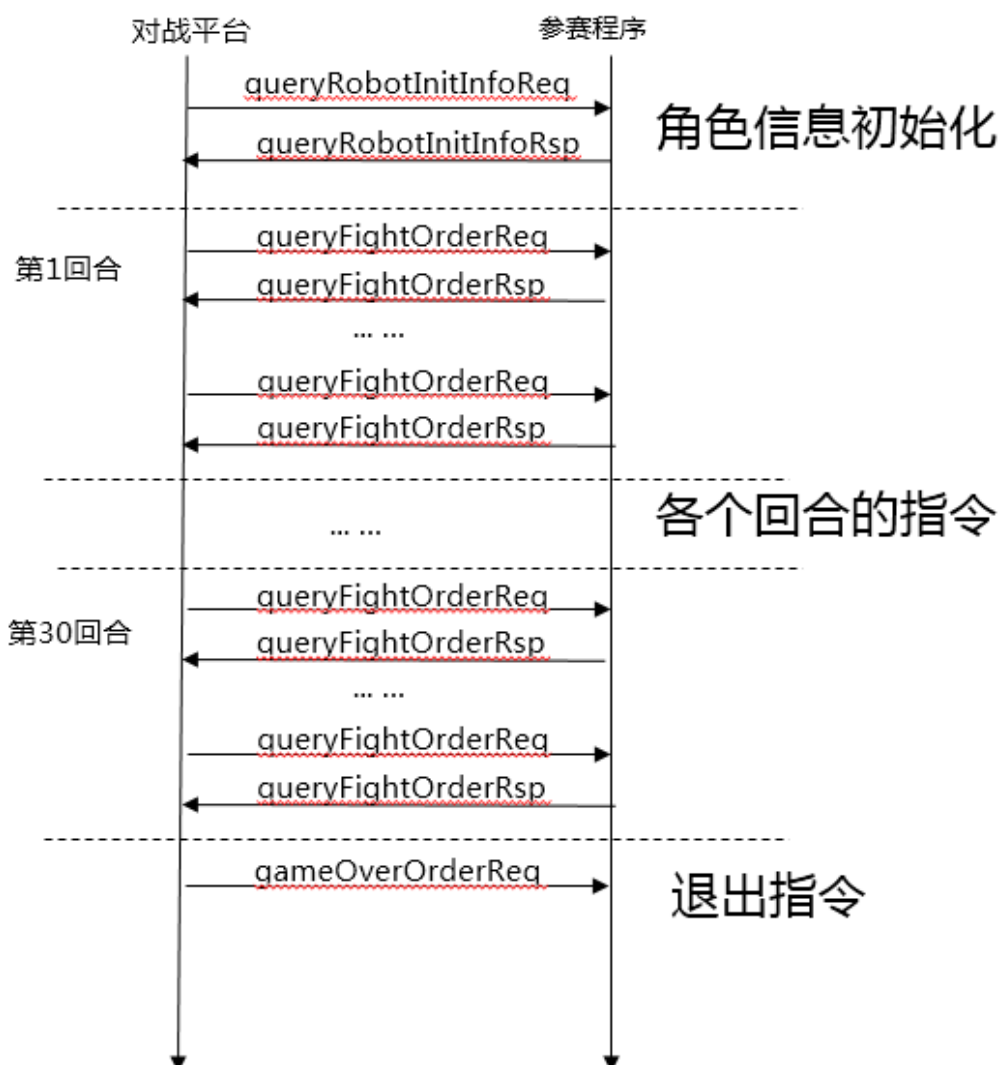
b) 建议采用 chrome 浏览器，其他浏览器会出现兼容或者卡顿问题。部分老的 PC 也可能因为配置比较差，可能出现卡顿，可以换成新的笔记本或者性能更好的机器查看。

#### 5. 调试过程中的易犯错误和注意事项：

a) **不要在代码中加 printf**：由于程序的 stdout 已被重定向用于收发包，printf 或者其他与收发包无关的向 stdout 输出的行为会扰乱收发包的解析。如要打印调试信息，请使用 stderr 通道或采用日志文件形式。

b) **记得在每个包后面\n换行符，记得 flush。**

## 6.交互流程



如上图所示：

游戏交互流程如下：

### 1. 初始化阶段：

对战平台向玩家发送 `queryRobotInitInfoReq`，查询玩家的初始化信息，参赛程序收到这个请求之后返回自身初始化的机器人布局列表(注意玩家一和玩家二处于不同的初始化布局位置)，通过 `queryRobotInitInfoRsp` 返回应答。

## 2. 游戏进行阶段：

每个回合，对战平台会向玩家发送  $n$  次 `queryFightOrderReq`，其中  $n$  为目前该回合玩家仍然存活的单位数量。玩家每次通过 `queryFightOrderRsp` 给出平台一个当前回合未行动单位的移动路线和攻击坐标（在 `queryFightOrderReq` 的请求报文中完整记录了本回合已经移动过的机器人）。

## 3. 游戏结束阶段：

30 个回合后，平台会向玩家发送 `gameOverOrderReq`，收到这个报文之后参赛程序主动退出（不需要应答）。

# 7. 通信协议说明

## 6.1 通用的 json 头部

### 6.1.1 请求包

```
{
  "interface": {
    "interfaceName": "" #接口类型，目前支持 3 种，queryRobotInitInfoReq，
queryFightOrderReq，gameOverOrderReq
  },
  "player": 1, #参赛程序分配成玩家 1 还是玩家 2，由对战平台随机初始化
  "round": 0, #第几回合
  "seq": 0, #序号，应答的时候原样返回
  "timestamp": 1536655826, #应答的时候原样返回
}
```

```
    "version": "1.0" #应答的时候原样返回
}
```

## 6.1.2 应答包

```
{
  "interface": {
    "ans": {},
    "interfaceName": "" #接口类型，目前支持 3 种，queryRobotInitInfoRsp ,
    queryFightOrderRsp , gameOverOrderRsp
  },
  "player": 1, #原样返回
  "round": 0, #原样返回
  "seq": 0, #原样返回
  "timestamp": 1536655826, #原样返回
  "version": "1.0" #原样返回
}
```

## 6.2 初始化机器人信息 queryRobotInitInfoReq

参数程序收到这个请求根据自己的策略初始化己方的机器人布局列表

### 6.2.1 请求示例

```
{
  "interface": {
    "interfaceName": "queryRobotInitInfoReq" #初始化机器人列表
  },
  "player": 1, #根据 player 是 1 还是 2 将兵力部署在对应的区域
  "round": 0,
```



```

    "seq": 0,
    "timestamp": 1536655826,
    "version": "1.0"
}

```

## 6.2.2 玩家一应答示例

```

{
  "interface": {
    "ans": {
      "robotlist": [#数组形式的机器人列表
        {
          "columnid": 4, #机器人的列坐标

          "hp": 0, #机器人的血量，参赛程序调试的时候在设置环境变量
            skipsafecheck 为 1 的时候生效，正式比赛会被对战平台忽略

          "robotype": "AiBrain", #机器人的类型，此处为 AI 大脑

          "roleid": 0, #机器人的角色编号，同一个玩家的机器人编号要求唯一

          "rowid": 25 #机器人的行坐标
        }, {
          "columnid": 2,
          "hp": 0,

          "robotype": "OverLordSoldier", #霸王兵

          "roleid": 1,
          "rowid": 27
        }, {
          "columnid": 3,
          "hp": 0,

          "robotype": "SlungshotSoldier", #投石兵

          "roleid": 2,
          "rowid": 26
        }, {
          "columnid": 1,
          "hp": 0,

          "robotype": "ArmourSoldier", #铁甲兵

```

```

        "roleid": 3,
        "rowid": 25
    }}
},
"interfaceName": "queryRobotInitInfoRsp"
},
"player": 1,
"round": 0,
"seq": 0,
"timestamp": 1536659042,
"version": "1.0"
}

```

### 6.2.3 玩家二应答示例，注意列坐标

```

{
  "interface": {
    "ans": {
      "robotlist": [
        {
          "columnid": 35,
          "hp": 0,
          "robottype": "AiBrain",
          "roleid": 0,
          "rowid": 25
        }, {
          "columnid": 37,
          "hp": 0,
          "robottype": "OverLordSoldier",
          "roleid": 1,
          "rowid": 27
        }, {
          "columnid": 36,
          "hp": 0,
          "robottype": "SlungshotSoldier",
          "roleid": 2,
          "rowid": 26
        }, {
          "columnid": 38,
          "hp": 0,
          "robottype": "ArmourSoldier",
          "roleid": 3,
          "rowid": 25
        }
      ]
    }
  }
}

```

```

    },
    "interfaceName": "queryRobotInitInfoRsp" # queryRobotInitInfoReq 的应答
  },
  "player": 2,
  "round": 0,
  "seq": 0,
  "timestamp": 1536659042,
  "version": "1.0"
}

```

## 6.2.4 字段补充说明

"player": 1, #参赛程序被分配成玩家 1 还是玩家 2 , 玩家 1 将兵力部署在左下角 5\*5 的格子区域 ( 母巢坐标为 28,1 ) , 玩家 2 将兵力部署在右下角 5\*5 的格子区域(母巢坐标为 28,38)。

如兵力部署的位置出错 , 会被对战平台直接判负

**其他约束条件 , 如出错会被直接判负 :**

1. 所有的机器人角色 roleid 不能重复 , 0 号要求固定为 AiBrain , 编号范围是[0-24].
2. 机器人的位置必须在对应左右下角的 5\*5 区域 , 位置不能重叠。调试的时候将堆栈平台需要的环境变量 skipsafecheck 设置为 1 , 可以任意部署
3. 消耗的金额总金额不得超过 10000
4. 支持 4 种机器人 , 分别是 :AiBrain( AI 大脑 ) ,OverLordSoldier( 霸王兵 ) ,ArmourSoldier ( 铁甲兵 ) , SlungshotSoldier ( 投石兵 )

## 6.3 对战请求指令 queryFightOrderReq

参赛程序收到这个指令选择一个本回合还没有移动或者攻击的机器人 ( havefight 为 0 的机器人 ) 进行移动和攻击 ; 请求包里面包含了自身和敌方所有的存活机器人信息 , 通过自己的

算法选择最优的移动和攻击策略。

### 6.3.1 请求包

```
{
  "interface": {
    "interfaceName": "queryFightOrderReq", #回合对战请求指令
    "para": {
      "attackbasebypeer": 0, #自己的母巢是否被对方攻破，攻破后也许可以不再守护
      "fightcount": 0, #本回合内第几次请求，可以忽略
      "haveattackpeerbase": 0, #自己是否已经攻破了对方的母巢，如已经攻破就不需
      "localrobotlist": [ #自己目前还存活的机器人列表
        {
          "columnid": 4, #列坐标
          "havefight": 0, #非常重要，如为 1 表示本回合已经发起过操作，不能再选
          "hp": 3000, #剩余的血量
          "hptocoin": 0, #可以忽略
          "isalive": 1, #肯定为 1，可以忽略
          "player": 1, #可以忽略
          "robottype": "AiBrain", #机器人类型
          "roleid": 0, #机器人编号
          "rowid": 25 #行坐标
        }, {
```

```

        "columnid": 2,
        "havefight": 0,
        "hp": 300,
        "hptocoin": 800,
        "isalive": 1,
        "player": 1,
        "robottype": "OverLordSoldier",
        "roleid": 1,
        "rowid": 27
    }, {
        "columnid": 3,
        "havefight": 0,
        "hp": 280,
        "hptocoin": 800,
        "isalive": 1,
        "player": 1,
        "robottype": "SlungshotSoldier",
        "roleid": 2,
        "rowid": 26
    }, {
        "columnid": 1,
        "havefight": 0,
        "hp": 450,
        "hptocoin": 600,
        "isalive": 1,
        "player": 1,
        "robottype": "ArmourSoldier",
        "roleid": 3,
        "rowid": 25
    }
}],

```

"peerrobotlist": [[#敌方目前还存活的机器人列表，格式与 localrobotlist 完全一](#)

[样，不重复解释](#)

```

{
    "columnid": 35,
    "havefight": 0,
    "hp": 3000,
    "hptocoin": 0,
    "isalive": 1,
    "player": 2,
    "robottype": "AiBrain",
    "roleid": 0,
    "rowid": 25
}

```

```

    }, {
        "columnid": 37,
        "havefight": 0,
        "hp": 300,
        "hptocoin": 800,
        "isalive": 1,
        "player": 2,
        "robottype": "OverLordSoldier",
        "roleid": 1,
        "rowid": 27
    }, {
        "columnid": 36,
        "havefight": 0,
        "hp": 280,
        "hptocoin": 800,
        "isalive": 1,
        "player": 2,
        "robottype": "SlungshotSoldier",
        "roleid": 2,
        "rowid": 26
    }, {
        "columnid": 38,
        "havefight": 0,
        "hp": 450,
        "hptocoin": 600,
        "isalive": 1,
        "player": 2,
        "robottype": "ArmourSoldier",
        "roleid": 3,
        "rowid": 25
    }
    }
},
"player": 1, #自己属于玩家几，初始化之后不会变的

"round": 1, #这是第几回合

"seq": 1,
"timestamp": 1536659042,
"version": "1.0"
}

```

### 6.3.1 应答包

```
{
  "interface": {
    "ans": {
      "attack": "23,17", #攻击哪个行列坐标，可以为空，表示不攻击，如"attack": ""

      "robottype": "OverLordSoldier", #发起攻击的机器人属于哪个兵种，可以不填

      "roleid": 2, #由哪个机器人发起的攻击，如没有选择合适的机器人，可以填写为
-1

      "walk": "RUUURU" #行走路径 L:左 R:右 U:上 D:下，也可以为空表示不需要移
动，如"walk": "" ;

    }, #先移动后攻击；移动，攻击字段根据实际情况都可以为空

    "interfaceName": "queryFightOrderRsp" # queryFightOrderReq 的应答
  },
  "player": 1,
  "round": 9,
  "seq": 123,
  "timestamp": 1536661758,
  "version": "1.0"
}
```

### 6.4 参赛程序退出指令 gameOverOrderReq

```
{
  "interface": {
    "interfaceName": "gameOverOrderReq" #收到这个指令参赛程序可以退出

  },
  "player": 1,
  "round": 30,
  "seq": 274,
  "timestamp": 1536661845,
  "version": "1.0"
}
```