

2019华为云首届鲲鹏开发者大赛-赛题说明文档

赛题：化鲲为鹏（探险）

1.1 背景介绍

北冥有鱼，其名为。之大，不知其几千里也，化而为鸟，其名为鹏；鹏之背，不知其几千里；怒而飞，其翼若垂天之云。

现在到了一年一度的化为鹏时期了，只有最强壮的鲲才可以成功化鹏。为了成功化鹏，这些鲲需要在北冥中进行比赛，获取积分，最后获取积分最多的，可以成功进化为鹏。

下面对北冥地图进行一些介绍。

1.2 地图规格

1.2.1 地图方向

X轴：表示第多少列，从左往右数

Y轴：表示第多少行，从上往下数

即：左上角为位置坐标为（0，0）

1.2.2 地图大小

地图大小支持（15x15）到（25x25）正方形地图。

1.3 地图元素

1.3.1 能量



能量是鲲得分的主要来源，每个能量携带对应的分值（分值范围1~5分）。

鲲移动到能量位置上，能量消失，该鲲携带的分数也将增加相应的分值。同时鲲所在的团队增加对应分值。

1.3.2 陨石



陨石是阻挡鲲移动的障碍物。

鲲遇到陨石，只能绕行。

1.3.3 虫洞



虫洞提供跳跃传送的功能，且以成对方式出现。

鲲必须从外部移动到虫洞上，传送功能才生效，传送到与之对应的虫洞上。

1.3.4 时空隧道



时空隧道提供快速连续移动的功能，共有四个方向。

鲲移动到时空隧道上，他会被移动到方向指向的下一个位置（即鲲永远不会踩在时空隧道上）。

时空隧道可以级联，一次移动多格。

下面对鲲之间的移动规则进行描述。

1.4 移动规则

1.4.1 分类

北冥中的鲲可以分为两大类，一类鲲遍体通红，性格暴躁，富有攻击性，称为赤鲲（beat），另一类鲲为蓝色，性格温和，善于思考，称为蓝鲲（think）。



1.4.2 收集能量

鲲移动到能量点上实现团队积分增加，该鲲携带的分值会相应增加。

1.4.3 视野范围

以鲲为中心点的一个方形视野，视野为X即可视范围为以鲲为中心的 $(2X+1) * (2X+1)$ 的方形。默认视野4，即视野范围9*9的方形空间。

1.4.4 消灭对手

场景一：在火（fire）模式下，赤鲲处于进攻方，蓝鲲处于防守方。在此模式下，赤鲲和蓝鲲移动到同一位置，即判定赤鲲消灭对手。

场景二：在水（water）模式下，蓝鲲处于进攻方，赤鲲处于防守方。在此模式下，蓝鲲和赤鲲移动到同一位置，即判定蓝鲲消灭对手。

消灭对手的鲲将获得对手身上携带的分值，并且所在团队获得对手携带的分值及额外10分的奖励。

1.5 游戏模式

1.5.1 水与火

游戏为回合制游戏，150回合做一次模式交换，交换瞬间完成。

即0~149回合处于一种模式，150~299回合处于另一种模式。

1.5.2 鲲移动

防守一方优先移动。（利于防守鲲躲避抓捕，增加保命能力）

同一方鲲移动，根据ID从小到大顺序移动。

多个鲲可以重叠。

1.5.3 能量生成

在地图上随机生成，分值范围在1-5之间。

1.5.4 鲲复活

每个团队都有一定的复活次数。

当有复活次数时，鲲被消灭后，随机选择重生点复活，继续进入游戏。

复活需要消耗4回合时间。

1.5.5 积分计算

团队积分只增加不存在减少，团队积分来源：

- 1、鲲采集的能量分值总数
- 2、鲲消灭对手获得的分值

1.5.6 上下半场

每半场时间为300回合

保证对战双方的公平性：

上半场，下半场鲲的出生点切换。

上半场模式顺序是先火后水。

下半场模式顺序是先水后火。

1.5.7 游戏结束

以下任意一条成立半场结束：

1、一方所有鲲都被消灭，并且这一方的复活次数为0

2、游戏回合数到达300回合

上下半场结束后，输出双方上下半场的分数及总分。

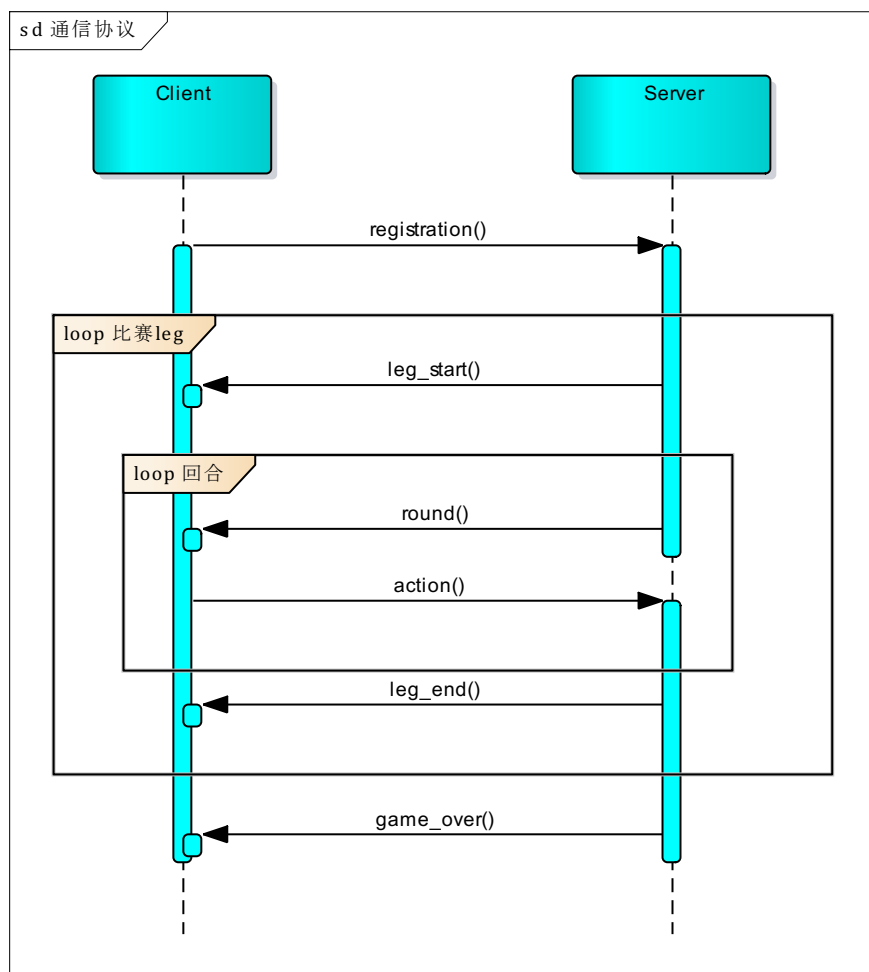
为了争取难得的化鹏机会，各只鲲需要各位参赛选手协助，给出行走路线，争取获取最多的机会。

1.6 交互接口

每位选手在自己的鲲鹏云服务器上进行开发、调试，然后将程序提交给大赛组，大赛组统一运行裁判程序来与选手程序进行交互。

具体交互协议如下：

1.6.1 通信协议流程图



Client为需要选手实现的客户端。

1. registration

客户端发送的注册消息。

2. leg_start

独立的子场（leg）开始消息。裁判程序给各参赛选手广播地图相关信息，以及选手的分组信息。

每个leg都重新开始，不受上一个leg结束状态，如双方已获点数、伤亡、地图等，的影响。

3. round

裁判程序给各参赛选手广播的当前回合的游戏相关的全部信息。

4. action

各参赛选手反馈在本回合准备执行的动作。

5. leg_end

本leg结束消息。裁判程序给各参赛选手广播比赛的结果，即各队的点数。

收到leg_end时请勿退出程序，务必等待下一个leg_start或game_over。

6. game_over

游戏结束消息

Server发送round消息后等待Client回复action消息的最大时长为**800毫秒**(调试时可配置)。

注意：服务器和客户端启动次序不定，客户端务必**支持等待socket连接**，推荐尝试连接**30秒**

1.6.2 消息

消息内容全部为文本字符，缺省都为小写字符串。格式为：消息长度+消息体。如下所示：

```
{
    "msg_name": "registration",
    "msg_data": {
        "team_id": 1000,
        "team_name": "test_demo"
    }
}
```

其中蓝色字体为消息体长度，黑色字体为消息体。

消息长度：指示消息体的长度（不包含长度本身），固定为5个字符，即消息体最长为65535个字符。高位不足补填0。

消息体使用JSON格式，详细格式要求请自行学习。提供基础版本实现供大家参考。

消息中所有字段全部为必选，使用[]表示数组为空。

下面是各消息体的说明。

1.6.3 registration

客户端发送的注册消息。

```
{
    "msg_name": "registration", #注册消息名
    "msg_data": {
        "team_id": 1001, #本队ID
        "team_name": "test", #本队名
    }
}
```

说明：注册阶段，如果一支队伍已经注册，另一支队伍无法连接服务器或者注册失败，注册阶段结束后游戏直接结束，已经注册的队伍胜利。

team_id: 本队ID。程序运行时输入的参数，为消息交互队伍识别Key值，测试时候可以随便输入一个整数，比赛时由裁判程序随机分配，请参见1.7节。

team_name: 本队名。程序签名，硬编码在程序中。长度最大为10字符，不支持中文和特殊字符。

1.6.4 leg_start

leg开始消息。裁判程序给各参赛选手广播地图相关信息，以及选手的分组信息。

```
{
    "msg_name": "leg_start", #leg开始消息名
    "msg_data": {
        "map": { #地图信息
            "width": 15, #地图宽度（X坐标）
            "height": 15, #地图长度（Y坐标）
            "vision": 3, #视野范围
        }
    }
}
```



```
"meteor": [          #陨石坐标
    {"x":1,"y":1},
    {"x":1,"y":4}
],
"tunnel": [          #时空隧道的坐标和方向
    {"x":3,"y":1, "direction":"down"},
    {"x":3,"y":4, "direction":"down"}
],
"wormhole": [        #虫洞的坐标和名称 a<->A b<->B
    {"name":"a",x":4,"y":1},
    {"name":"A",x":4,"y":4}
]
},
"teams":[            #各队及Tank ID
    {"id":1001,"players":[0,1,2,3],"force":"beat"},
    {"id":1002,"players":[4,5,6,7],"force":"think"}
]
}
}
```

"teams"中的id为团队的编号，全局唯一，参赛选手程序运行时所输入的参数。

"teams"中的"force"，团队的能力，赤鯢（beat）还是蓝鯢（think）。

"wormhole"：虫洞在名称为大小写字母之间传送，如虫洞a与虫洞A相互传送。地图保证成对出现。

1.6.5 round

裁判程序给各参赛选手广播的当前回合的游戏相关的全部信息，包括双方player的所有信

息。

每回合广播一次。

```
{
  "msg_name": "round",      #回合消息名
  "msg_data": {
    "round_id": 2,          #回合标识，要求参赛选手在响应消息回填
    "mode": "beat"         #本回合优势势力，在一定回合后切换
    "power": [              #能量的坐标与价值
      {"x": 5, "y": 2, "point": 1},
      {"x": 5, "y": 5, "point": 2}
    ],
    "players": [            #sleep表示该player是否处于睡眠
      {
        "id": 0, "score": 0, "sleep": 0, "team": 1001, "x": 0, "y": 1
      },
      .....                #如果有多个player信息的话
    ],
    "teams": [              #各队当前的分数、剩余复活次数
      {"id": 1001, "point": 0, "remain_life": 2},
      {"id": 1002, "point": 0, "remain_life": 3}
    ]
  }
}
```

"mode"表示本回合优势的能力。

"power"只包含视野范围内的矿点。

"players"包含自己，同时也包含视野范围内的敌人。

1.6.6 action

参赛选手反馈在本回合准备执行的动作。每个参赛选手每回合只反馈一条消息，包含所有player的动作请求。

```
{
  "msg_name": "action",      #动作请求消息名
  "msg_data": {
    "round_id": 2,          #回合标识, 回填
    "actions": [
      {
        "team": 1002, "player_id": 5,
        "move": ["up"],      #移动方向, 不动为空[]
      },
      ..... #如果有多个player动作需要反馈的话
    ]
  }
}
```

“move”指令：每步移动方向只能为up, down, right, left，每回合最多一步。不动为空[]

1.6.7 leg_end

leg结束消息

```
{
  "msg_name": "leg_end",
  "msg_data": {
```

```
"teams" : [  
    {  
        "id" : 1001,          #队ID  
        "point" : 770        #本leg的各队所得点数  
    },  
    {  
        "id" : 1002,  
        "point" : 450  
    }  
]  
}  
}
```

1.6.8 game_over

游戏结束消息

```
{  
    "msg_name" : "game_over"  
}
```

1.7 开发调试资源

每位选手在自己的鲲鹏云服务器上进行开发、调试，然后将程序提交给大赛组，大赛组统一运行裁判程序来与选手程序进行交互。

1.7.1 样例工程

```
试题工程  
├── run.bat          #系统启动脚本，包括client/server/ui三套程序，自动比赛  
├── submit.bat       #将client目录打包成比赛规定格式的zip包，用于提交作品  
├── client  
│   ├── gameclient.bat #选手程序比赛入口文件，务必存在  
│   └── bin  
│       └── client.exe #放置执行程序，由参赛者替换
```



gameclient.bat的入参依次为team_id server_ip server_port，比赛时由竞赛系统传入。

gameclient.bat的内容由参赛队伍自行实现，用于启动.exe/.jar/.py等，当前默认启动bin目录下demo.exe程序

demo_proj提供简单基础工程，请自行修改，替换client中对应程序或修改脚本启动

注：试题工程为了方便在windows上运行，提供的脚本为gameclient.bat，选手最终提交的程序会在linux环境下运行，因此需要提供的脚本未gameclient.sh，详见1.7.4，提交格式。

1.7.2 单机游戏运行

直接运行样例工程中的run.bat即可开始战斗。

如果只需要回放之前的战斗，直接运行/ui/nw.bat即可，如果已经打开了UI，直接点击刷新按钮也可以重新回放。

log.txt提供运行日志，供分析行为及定位使用。

client中为测试程序，请自行替换为参赛作品。

1.7.3 JSON 库

通信协议采用JSON格式，各参赛队伍可以到网上自行下载各类JSON解析库。

1.7.4 提交格式

- 1、作品软件包命名规则统一为 队长华为云账号.zip
- 2、软件包内需要包含一个gameclient.sh文件，并且该文件可以使用鲲鹏CPU对应的linux操作系统中的bash指令运行，该程序需要支持3个连续的参数，分别为 队伍id（整数）、server ip地址（字符串）、port（数字）
- 3、作品运行的实体程序，由gameclient.sh调度，建议实体程序采用c语言、java、golang、python2或者python3完成
- 4、所有程序从启动至链接服务器可以允许有10分钟软件包依赖的准备时间，如开源软件安装等等，如果10分钟仍无法链接服务器，将会被判定为0分
- 5、由于gameclient.sh文件本身具备一定的开放性，请不要提交具备破坏、攻击性的作品，所有的作品都会在独立的虚拟机或容器中运行，出现虚拟机或容器被作品破坏、攻击的情况，则参赛资格会被取消
- 6、作品可以将日志输出至/var/log/battle.log，您对程序的执行结果有极大的疑问时，可以联系赛事主办方获取日志，确认你的程序运行是否正常

1.7.5 Server 相关

- 1、Server支持的启动参数：

注意：

- 启动Server时，需要在Server.exe同目录下放置地图文件map_demo.txt，否则启动失败。
- Server启动后，会在同目录下生成log.txt，data.csv，replay.txt，分别用于问题定位、积分统计、游戏回放。

- 2、Server启动：

如果Server启动成功，则会打印：

```
Wait Players !
39528!39584, Connection established.
Player[222] join game!
39528!39584, Connection established.
Player[111] join game!
```

如果Server启动失败，常见的问题有

- Bind失败，原因是ip+端口号被占用，此时换一个ip+端口号即可，可以通过netstat -ano查询本机已经占用的ip+端口号。

```
Wait Players !
ACE_Acceptor::ACE_Acceptor: address already in use
```

3、Server会记录收发消息的错误信息，参见error_log.txt:

4、Client掉线

Client掉线的主要原因是：Client程序异常或无法在规定时间内返回action消息。

连续10次未收到来自Client的action消息，判定为offline。

一方掉线，另一方继续运行，直到游戏结束。

1.7.6 地图格式

```
{
  "game": {
    "timeout": 800
  },
  "map": {
    "height": 15,
    "width": 15,
    "map_str": "
      a5.#.O...O.#.1b
      11.#.....#.11
      .....1..
      ....v...v....
      ####.....####
      X.....1.1.....X
      ....11.11.....
      .....5.....
      ....11.11.....
      X.....1.1.....X
      ####.....####
      .....^...^.....
      ..1.....
      11.#.....#.11
      B1.#.O...O.#.5A
      "
    "
  }
}
```

地图为JSON格式的文本文件，

game中包含了timeout超时时间配置，单位ms，如果为0表示不判断超时。

Map数据：

width表示宽多少列即x轴的范围

height表示搞多少行即y轴的范围，地图最大支持30×30

map_str为一组字符串具体描述：

- “.” 表示空
- “#” 表示陨石
- “^” “v” “<” “>” 分别表示时空隧道向上、向下、向左、向右
- “A” “a”，“B” “b”，“C” “c”，“D” “d” 分别表示4组虫洞。大写与小写字母相连，不能单独存在，也不能重复存在。
- 数字表示初始能量，取值1~5。例如“2”表示分值为2的能量点

- “O” / “X” 表示队伍出生点，地形是空地。

例如1队和2队比赛，上半场1队首先出生在A点，2队首先出生在B点。当存在多个A点时，1队队员将平均分配在各个A点上，同样存在多个B点时，2队队员也将平均分配在各个B点上。

1.7.7 运行打印

每一轮会显示团队的积分情况，同时会将当前执行地图输出在界面中。

```
Wait Players !
39528!39584, Connection established.
Player[222] join game!
39528!39584, Connection established.
Player[111] join game!
leg: 0 round: 0
team_id: 222 remain_life: 4 point: 0
team_id: 111 remain_life: 4 point: 0
-----
a5e#.0...0.#.13
11e#.....#.11
eee.....
...v...v...
####.....####
x.....e.....x
...11e11...
...ee5ee...
...11e11...
x.....e.....x
####.....####
...^...^...
...eee
11.#.....#e11
31.#.0...0.#e5A
-----
```

1.7.8 执行机

对预装软件或框架有特殊要求的，可提前告知试题组，并在提交程序时再次说明。