

海康威视2018软件精英挑战赛

线上复赛截止：2018.06.12

本届大赛试题以无人机AI对战为背景，参赛选手以自建队伍为单位，在给定地图的三维空间与其他限定条件下，运送货物的对战比赛，每场比赛中获得价值最多者获胜。复赛赛题在初赛题目基础上增加条件，已在官网公布。答案提交于**6月12日中午12:00**截止。**2018年6月6日开放提交复赛代码调试。**

[报名参加](#)

[目录](#)

海康威视2018软件精英挑战赛题目

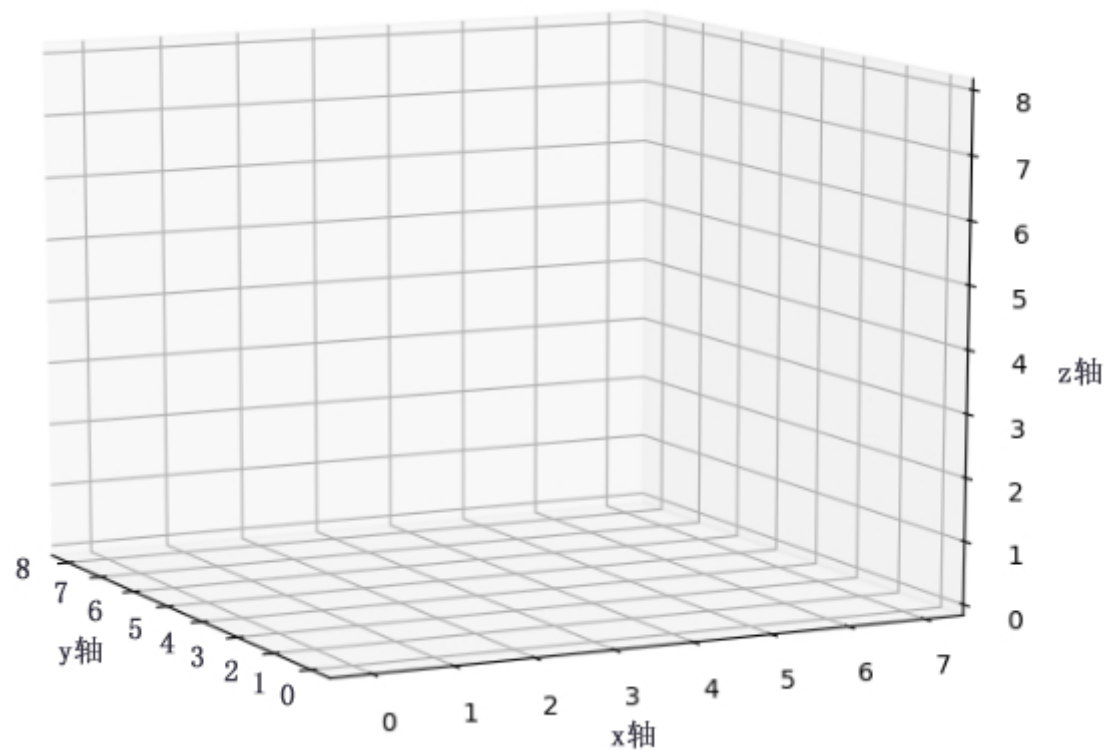
复赛题目

背景介绍

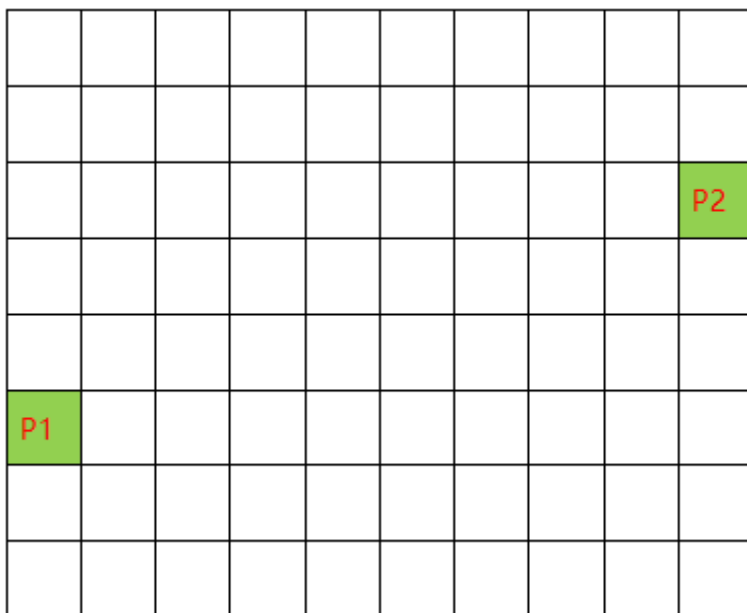
海康机器人行业级无人机业务依托海康威视视频技术的积累，进行跨技术领域的深度整合，以视频图像处理为基础、以产品安全为基石、以智能应用为导向、以满足行业需求为目标，自主研发了雄鹰系列无人机及保障低空空域安全的无人机干扰器，以丰富的产品为客户提供有针对性的行业解决方案，立足安防，专注行业。题目以无人机为主题考察参赛队伍的综合能力。这个夏天，来一起AI对战吧！

了解更多海康机器人业务进入[海康机器人官网](#)

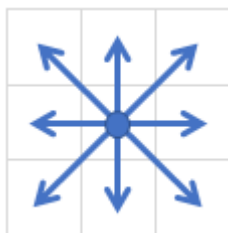
题目描述



在一个三维空间中，如上图所示，空间中的每个点对应一个坐标 $P(x,y,z)$ 其中 x 、 y 、 z 为非负整数。 x 和 y 为水平 z 为垂直坐标。当 $z=0$ 时，就是我们所说的地面，如下图所示：



在地面上存在二个停机坪P1和P2，停机坪上各有 n ($n > 0$)架规格不一的无人机。无人机可以上下以及水平移一步移动一个格子（包括斜着也是移动一个格子，例如下图中对角线的移动），同一时刻只能向一个方向移动，当前无人机的坐标为 $P(4,4,4)$ 则下一时刻无人机可以移动到 $P(4,4,5)$ 、 $P(4,4,3)$ 、 $P(3,4,4)$ 、 $P(4,3,4)$ 、 $P(5,4,4)$ 、 $P(5,5,4)$ 、 $P(3,5,4)$ 、 $P(5,3,4)$ 、 $P(3,3,4)$ 以及 $P(4,4,4)$ 。即无人机空间上不能水平和垂直同时运动。可以向四周运动，如下图所示



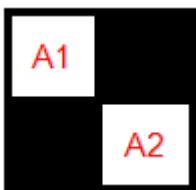
由于无人机低空水平飞行存在很大的安全隐患，因此无人机必须上升到一定高度 H_{low} ($H_{low} > 0$ ，包括 H_{low} 后，才能水平飞行。同时无人机由于他自身性能问题，飞行的最大高度为 H_{high} ($H_{high} > H_{low}$ ，包括 H_{hi} 不同规格的无人机载重量 L ($L > 0$) 不同。(同一时刻 一个无人机只能载一个货物，即使载重量有多余也不能载货物)。

在某一时刻，水平地图上会出现不同价值的物品，(所有的物品 $z=0$ ，目的地也是 $z=0$ ，同一地点物品持续时间出现多个，物品不会出现在建筑物内)，物品的价值为 V ($V > 0$)，质量为($G > 0$)，最大等待时间 T ($T > 0$ ，物品在出现的时间为 t ，那么在 $t+T$ 之后，物品将消失，不能再运送，如果在物品消失之前被拾起，则物品不会消失。)以及物品需要运送到的目的地 $D(x, y)$ 。当无人机将物品送到目的地后，将获得物品的价值 V ，物品不能在本次大赛的通讯协议JSON中，物品出现的时间用 $start_time$ 表示，持续时间用 $remain_time$ 表示(该值固定，随着时间减少)，当前剩余时间用 $left_time$ 表示，每过一个单位时间，该值会减少1。例如：某物品出现的时间为 t ，持续时间为5，那么第一次出现 $start_time=t$, $remain_time=5$, $left_time=5$, 下一个单位的时候 $start_time=t+1$, $remain_time=5$, $left_time=4$ 。直到 $left_time=0$ 。注意 $left_time=0$ 的时候，不能捡起物品。

选手可以用自己获取到的价值去购买无人机，以增加自己的运行效率，新购的无人机会在下一秒出现在自己的位置上。

在三维空间中，存在建筑物，以及雾区，为了简化题目建筑物以柱型方式出现，即垂直方向建筑物大小不变。体积可以忽略，同时雾区也是矩形方式出现。无人机不能碰到建筑物，否则无人机将损坏不能再使用。对于雾区，战双方不能看到对手的无人机。

对于建筑物(黑色部分表示建筑物)，无人机允许如下图所示的情况进行飞行，从A1位置飞行到A2位置。



复赛无人机说明

由于目前技术的限制，无人机的飞行过程中耗电量极大。为了方便做题，现在假设无人机空载的时候，耗电极低，可以忽略不计。载货时，耗电量跟货物的重量成正比，比例系数为1，例如货物重量为100，那么单位时间为100个单位电能。单位时间耗电量跟无人机有没有移动无关，即静止状态也需要耗能，包括无人机在地面的（停机坪除外）。不同的无人机电池容量不同。如果无人机在外面飞行过程中出现没电量（空载除外），即视为无人机坠毁（不论是否在地面）。选手可以在自己的停机坪对无人机进行充电，单位时间充的电量跟无人机机型有关（无人机只能在空载的时候进行充电，如果载有货物，不能进行充电，否则视为犯规）。停机坪可以同时多台无人机进行充电（充满电后，可以停留在停机坪）。初始无人机电量都为0，（包括比赛开始时，以及新购买的无人机），因此一开始需要充电（不打算运送货物除外）。

复赛对战协议JSON新增项说明

地图信息中init_UAV中新增剩余电量remain_electricity, 表示当前无人机还剩余的电量，由于一开始无人机电量为0，因此都为0。

无人机价格表增加电池容量 capacity, 和单位时间充电量 charge。

服务器发给参赛者下一步指令中 UAV_we和UAV_enemy中，增加剩余电量 remain_electricity, 表示当前机还剩余的电量。同时增加status字段值定义3（0表示正常，1表示坠毁，2表示处于雾区），表示充电。

选手发给服务器指令UAV_info中，增加剩余电字段 "remain_electricity"。

复赛新增项数据说明

1. 当无人机飞到己方停机坪后，就默认开始充电，不需要选手设置，由于载货时不能充电，因此载货的时候不能进入己方的停机坪。

2. 无人机剩余电量需要选手自己设置，例如某飞机（单位时间充电为1000）处于飞行载重状态，货物重量100，假如 $t=40$ 时，服务器发送过来的数据中，该架无人机剩余电量为500，那么这个时刻选手发送给服务器的数据中，剩余容量应该为400。同理对于充电状态，如果服务器发送过来当前剩余容量为1000，那么这个时刻选手发送给服务器的数据中，剩余电量为2000，（当然不能超过无人机的总容量，例如总容量为1500，那么该时刻返回的1500）。如果当前时刻已经充满，那么每次返回只要返回充满的电量即可。

3. 进入停机坪临界说明，假设某无人机容量为4000，单位充电量为2000， (x, y) 坐标为停机坪坐标，需向停机坪充电。当前服务器发送过来的数据 $z=1$ ，剩余电量为1000，你返回给服务器时的数据为 $z=0$ ，剩余电量3000，随后服务器返回你的数据为 $z=0$ ，剩余电量为3000；假如此时你还不打算飞走，你应该返回给服务器 $z=0$ ，剩余电量为4000。

4. 离开停机坪临界说明，假设无人机 x, y 坐标为停机坪坐标，还没有充满电，就要离开停机坪，当前服务器发送过来的数据 $z=0$ ，剩余电量为1000，你返回给服务器的数据为 $z=1$ ，剩余电量为1000。

5. 取货进入临界值说明, 假设无人机 x, y 坐标为货物坐标, 货物编号为50, 重量为100, 当前服务器发送过来的数据 $z=1$, 剩余电量为1000, 你返回给服务器的数据为 $z=0$, 剩余电量为900, 货物编号为50。

6. 取货物离开临界值说明, 假设无人机 x, y 坐标为货物坐标, 货物编号为50, 重量为100, 当前服务器发送过来的数据 $z=0$, 剩余电量为900, 你返回给服务器的数据为 $z=1$, 剩余电量为800, 货物编号为50。

7. 放置货物进入临界值说明, 假设 x, y 坐标为货物运送目标坐标, 货物编号为50, 重量为100, 当前服务器发送过来的数据 $z=1$, 剩余电量为1000, 你返回给服务器的数据为 $z=0$, 剩余电量为900, 货物编号为50。

8. 放置货物离开临界值说明, 假设 x, y 坐标为货物运送目标坐标, 货物编号为50, 重量为100, 当前服务器发送过来的数据 $z=0$, 剩余电量为1000, 货物编号为-1, 你应该返回给服务器的数据为 $z=1$, 剩余电量为1000, 货物编号为-1。

9. 耗电完毕坠毁临界值说明, 假如 $t=t_1$ 时刻, 某无人机剩余电量100, 所载货物 $no=10$, 货物重量为100, 下一步, 如果顺利到达送货点, 则你应该发送给服务器 $z=0$, 货物编号10, 剩余电量0, 随后服务器会返回给你 $z=0$, 货物编号为-1, 剩余电量为0; 如果没有到达送货点, 假设为 $z=1$, 则你应该发送给服务器 $z=1$, 货物编号10, 剩余电量0, 随后, 服务器会返回给你该无人机坠毁。

10. 购买无人机说明, 假设某时刻 t_0 , 你发送"purchase_UAV": [{"purchase": "F1"}]购买无人机且货币充完, 下一步 t_1 , 服务器给你返回购买成功的无人机{no: noNew1, x: 停机坪 x 坐标, y: 停机坪 y 坐标, z: 0, remain_electricity: 0, status: 0}, 此时, 你可以进行充电, 也可以飞走。如果充电, 则发送z: 0, remain_electricity: 单次充电电量, 下一步服务器会确认, 并返回status: 3; 如果飞走, 则发送z: 1, remain_electricity: 0, 下一步服务器也会确认, 并返回status: 0。

对战方式

每次比赛有二只队伍进行对战，每张地图比赛二次（交换停机坪），在停机坪上无人机可以重叠出现，即没有碰撞，一旦离开停机坪，如果碰到其他无人机，则二架无人机都将损坏（因此离开停机坪也必须一架一架的离开碰撞后，无人机如果有载物品，则物品也一起消失。碰撞的方式有二种，

1. 同一时刻，同一位置同时出现多架无人机；如下图所示：前一时刻无人机位置如图1，下一时刻无人机在个位置，如图2。

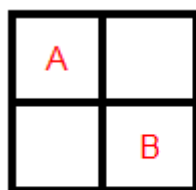


图 1

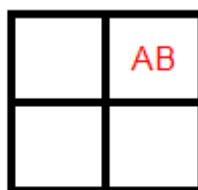


图 2

2. 无人机飞行过程中出现相遇。如下图所示 当前时刻无人机A和B在图1位置，下一个时刻变成了图2位置，相遇，类似的情况也包括对角线的交换。

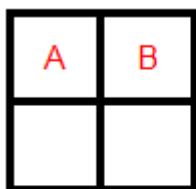


图 1

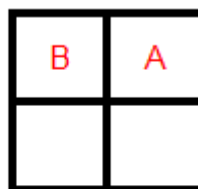
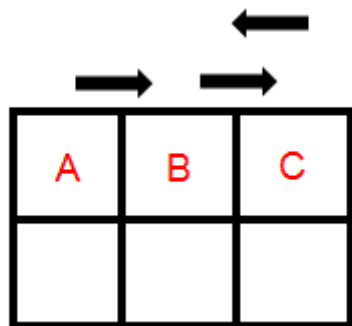


图 2

优先检测第二种情况，一旦飞机损坏，即刻从游戏中移除损坏的飞机，再检测第一种情况。例如，如下图所示，机A B向右飞行，无人机C向左飞行，那么下一时刻由于B和C存在相遇，因此先判断相遇，即B和C撞毁，从地图除，接下来判断位子重合，这个时候由于B和C已经消失，因此A无人机没有影响。



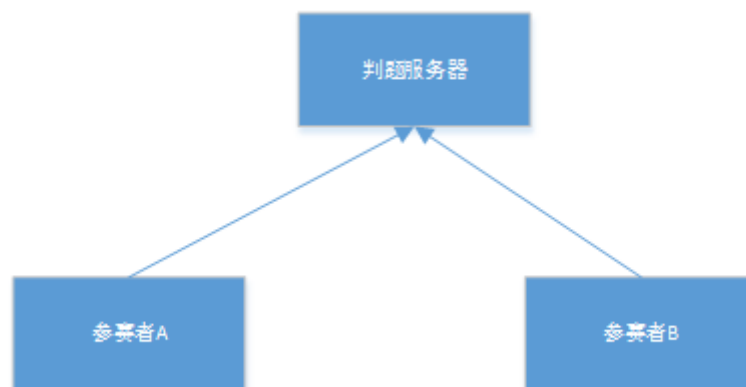
本次比赛不同于以往ACM，需要存在网络交互。参赛者的程序需要去链接服务器，（服务器IP，端口待定，链接，中间不能关闭socket，否则认为失败），链接到服务器后，服务器会发送地图信息给二个参赛者，发送信息后表示比赛开始，该时刻为0，参赛者程序收到地图信息，此时，参赛者不能移动无人机，不能充电；同时需参赛者将他们所控制的无人机时刻为0的位置发送给服务器，服务器接受到二个参赛者的信息后，做一些处理，并息合并分别发送给参赛者程序，此时时刻为1。同理参赛者需要将时刻为1的无人机等信息发送给服务器。以此：直到比赛结束。具体发送数据格式见协议说明。

当比赛结束时，获得的价值最多者获胜（这里的价值包括现有无人机价值+剩余价值），如果价值相同，那么所行时间最短的获胜，（运行时间指 服务器每次发送给参数者信息到接收到信息这段时间总和）。

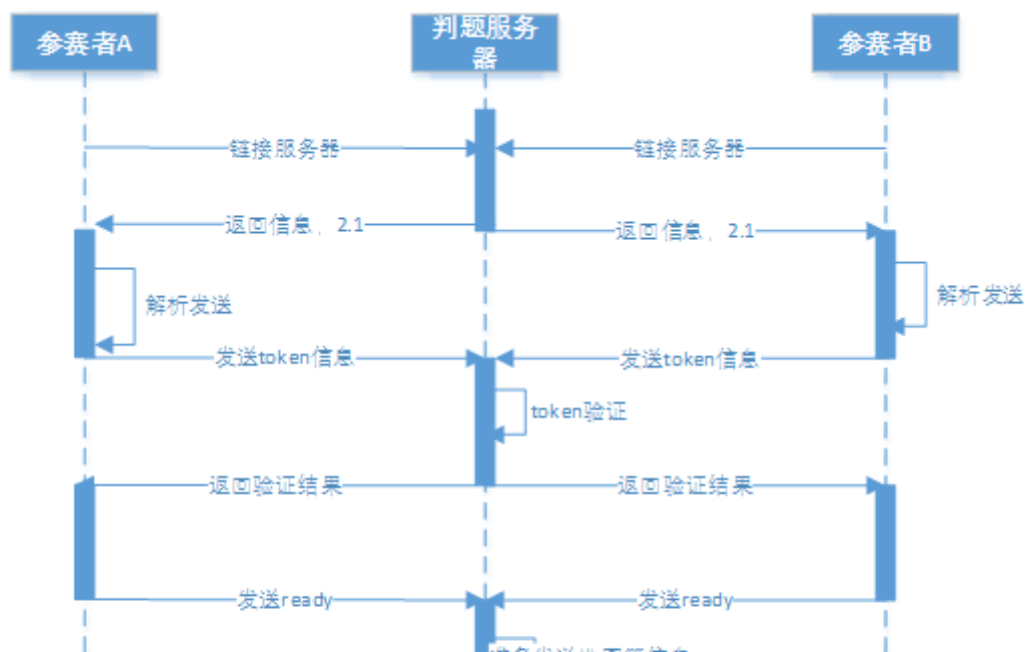
数据说明

比赛中关于链接服务器，以及协议的解析，我们已经提供相应的demo，参赛者可以直接拿demo来使用。题目的时间都是虚构的单位时间。为了使参赛者容易理解比赛过程中的流程，下面说明下这个过程的架构。

比赛过程中参赛者和服务器之间的链接如下（这部分demo中都已经实现，参赛者只要关注算法即可）：



对于他们之间的数据的时序图如下：





这个时候，选手如果返回无人机位置为 (0, 0, 1)，并同时设置物品编号，则认为操作错误。因为你是先离开0再拾取物品。对于无人机将货物送到目的地后，不需参赛者要设置goodsno为-1，服务器会自动将该字段设置无人机空闲时刻可以飞到任何地方，（除了建筑物内）。由参赛者自己决定。

整个地图中，最多物品个数不会超过256个，建筑物个数不会超过50个，雾区不会超过50个，无人机种类不超过10种。

比赛结束的总价值 = 未撞毁无人机的价值+赚取物品的价值。

算法建议说明

由于该题目比较开发，要想考虑全面非常复杂，因此比较合适的做法是（仅做参考）

- 1.比赛开始的时候，可以派出所有无人机，分散到地图的各个角落。
- 2.当物品出现后，派遣最近的无人机去拾取货物，然后根据货物的目的地进行路径规划。
- 3.送到货物，如果没有其他货物可以送，可以再次将无人机平均分散到地图上，如果有货物要接送，那么直接进行路径规划，去货物点拾取获取。
- 4.以上整个过程不要考虑对方无人机的干扰，否则问题又复杂化，会让选手无从下手。
- 5.如果上面的程序完成了，接下来可以加入对方信息，进行优化，例如用自己廉价无人机去撞毁对方昂贵无人机者带有重大价值的无人机等。

Demo说明

以C++demo为例，由于整个流程的架构demo都已经给出，因此只需要关注AlgorithmCalculationFun函数，每一步服务器demo都会调用该函数，参数者只需要根据自己的算法填写flyplane结构体即可。对于java和python demo也是类似。

C++ Demo: <https://git.acmcoder.com/hikvision/UAVGoodsDemoForCPP>

Java Demo: <https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForJava8>

Python2 Demo: <https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython2>

Python3 Demo: <https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython3>

协议说明

本次比赛评测时，会通过命令行启动您的程序，并附有3个命令行参数，第一个参数是对战控制器IP，第二个参数是对战控制器端口，第三个参数是对战校验Token；待5月中下旬开始调试时，可在我的对战中获取，获取后自行在动程序并传入参数调试；正式比赛时，这3个参数由服务器自动生成。

本次比赛网络数据传输采用json格式，对于json格式不懂的同学，可以网上查找相应的资料，我们也会在提包里供相应的解析代码。数据发送和接受的格式为 前面8个字节表示json长度， 接下来为json数据，整个数据包长度为8+json的长度。

“前8个字节表示JSON长度”的说明：此8字节中的每个字节均为ascii码，表示长度对应的字符串的每一位，不前面补零。

例如JSON长度为1820，先将1820转换为字符串，长度为四位，然后再前面补四个0，即变成字符串“0000182”，然后在socket中进行传递。

服务器发送给参赛者的json格式如下（赛题包中有相应的文件）：

参考<https://cdn.acmcoder.com/assets/hikvision2018/droneaisocket.html#jump6>

参赛者发送到服务器的json格式如下：

参考<https://cdn.acmcoder.com/assets/hikvision2018/droneaisocket.html#jump5>

关于Socket传输协议格式和JSON的示例，请参考：<https://git.acmcoder.com/hikvision/UAVRobotSimple>

通讯以及json解析框架DEMO参考：

1、C++：

<https://git.acmcoder.com/hikvision/UAVGoodsDemoForCPP>

2、Java:

<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForJava8>

3、Python:

<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython3> 或

<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython2>

数据限制

题目中所有的数据都为非负整数，不存在浮点计算。

所有地图的长宽 $0 < W, L \leq 200$ ，高度 $0 < H \leq 200$ 。

无人机飞行高度 $0 < h_{low} \leq h_{high} < H$

所有数据计算不会超过32位整数。

一架无人机上同时只能运载一种货物。

比赛说明

初始所拥有的无人机信息会在比赛命令开始时给出，货物信息在出现时才会给出；

每一步最长运行时间：服务器发送给参赛者后，参赛者必须在1秒内把结果返回给服务器，否则认为认输。初赛阶段由于是运行在选手个人电脑上，所以没有此1秒的限制，正式比赛时，选手的最后一次提交程序将会运行在核16G的服务器上，此时才有1秒的限制。一次比赛不超过15分钟。

目录

比赛结束时间, 分三种情况:

1.所有物品都运送完毕或消失;

- 2.地图上没有无人机，无人机已全部撞毁（包括有价值购买无人机，但实际并没有购买）；
- 3.比赛超过15分钟。特别地，假如地图中已经没有未被拾起的物品，则已经在飞机上的物品必须在 $T=$ 地图的长的时间内运送完毕，但是比赛总时间还是不能超过15分钟。

参数者每次计算的数据必须要确保准确，如果出现与实际不匹配，那么认为认输。不匹配的情况有如下几种：

- 1.无人机运行轨迹不正确，例如出现跳跃。（允许无人机撞向建筑物，或者其他无人机，这个判断由服务器做）
- 2.载重量不符合要求，例如载重量小的无人机去运送超过它载重的物品；
- 3.当前剩余价值不够的情况下，购买无人机；
- 4.移动已经撞毁的无人机；
- 5.一些不符合题意的操作。

初赛晋级规则

如何进入复赛：

- 1、如果最终有效代码数大于500，则随机两两分组进行PK，每组胜者直接进入下一轮；
- 2、如果最终有效代码数小于等于500，则随机每15个代码组成为1组，组内进行两两PK，赢一次积1分，组内排50%的进入下一轮。

什么是最终有效代码：

截止日12:00前，已经编译通过，且创建对战房间，通过与机器人或自己对战，能获取到货物价值的代码中的个。

PK方式：

一张地图，随机停机坪进行一局，然后交换停机坪进行一局，两局累加综合进行比较。

当比赛结束时，获得的价值最多者获胜（这里的价值包括现有无人机价值+剩余价值），如果价值相同，那么所行时间最短的获胜，（运行时间指 服务器每次发送给参赛者信息到接收到信息这段时间总和）。

如何提交

本次大赛使用git进行代码管理，请在<https://git.acmcoder.com/hikvision>自行注册git账号进行代码版本管理。

注册完git账号后，请在“用户设置”-->“授权应用”-->“生成新的令牌”进行令牌的生成，大赛系统将会使用。

注意，生成的令牌只会显示一次，请生成后立刻拷贝进行妥善保存，已经用于提交代码评测的token请不要删除会影响您的评测。

 控制面板 工单管理 合并请求 发现 +

帐户设置

个人信息

头像设置

修改密码

邮箱地址

SSH 密钥

帐户安全

仓库列表

组织列表

授权应用

删除帐户

管理个人操作令牌

生成新的

您可以使用这些已生成的令牌来操作 Gogs API。



HikvisionUAVGoodsAI

增加于 May 01, 2018 — 上次使用在 May 01, 2018

删除



Travis-CI

增加于 Sep 05, 2017 — 没有最近活动

删除



drone

增加于 Aug 29, 2017 — 上次使用在 Aug 29, 2017

删除

大赛将于5月中下旬左右开始代码的提交，届时，请输入令牌，即可选择对应的仓库和分支进行提交。

注意：git账号和大赛账号不必一样，为了不泄漏您的git账号，我们在大赛系统中使用令牌进行代码的提交。

建议由队长新建一个私有仓库，并将队员加入为协作者。

对于要提交大赛系统进行评测的分支，建议通过分支保护功能保护起来，以免push时不小心被覆盖。

分支保护功能在您的代码仓库下，“仓库设置”-->“管理分支”-->“保护分支”

目录

 lifubang / test

 取消关注

1

 点赞

0

 派生

 文件

 工单管理 0

 合并请求 0

 Wiki

 仓库

仓库设置

基本设置

管理协作者

管理分支

管理 Web 钩子

管理 Git 钩子

管理部署密钥

默认分支

默认分支是被用于代码提交、合并请求和在线编辑的基准分支。

master

▼

更新

保护分支

保护分支不被强制推送、意外删除和限制代码提交白名单。

选择一个分支...

▼

master

test

编程语言

支持C，C++，java，python。允许使用第三方开源库，使用多线程等特性。

C

GCC版本：7.2.0

MAKE版本：GNU Make 4.1

APT源：<https://mirrors.aliyun.com/ubuntu/>

操作系统(64位)：Ubuntu 16.04.3 LTS下使用docker运行buildpack-deps:artful镜像。

示例及注意事项请查看：<https://git.acmcoder.com/hikvision/UAVGoodsDemoForC>

- 1、你的依赖包安装请写在可执行文件./configure中，注意使用apt-get install -y，以免被交互打断；
- 2、如果你没有额外依赖包，也要准备一个空的可执行文件./configure；
- 3、请提供Makefile进行编译生成，生成的目标文件名必须是main.exe，备注：ubuntu也可以是以.exe结尾。
- 4、评测命令：./main.exe 39.106.111.130 4010 ABC123DEFG

参数1：39.106.111.130，是裁判服务器的IP，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；

参数2：4010，是裁判服务器允许您连接的端口，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；

参数3：字符串，是本次运行的令牌，以便校验您的资格，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度生成。

C++

G++版本：7.2.0

MAKE版本：GNU Make 4.1APT源：<https://mirrors.aliyun.com/ubuntu/>

操作系统(64位)：Ubuntu 16.04.3 LTS下使用docker运行buildpack-deps:artful镜像

示例及注意事项请查看：<https://git.acmcoder.com/hikvision/UAVGoodsDemoForCPP>

- 1、你的依赖包安装请写在可执行文件./configure中，注意使用apt-get install -y，以免被交互打断；
- 2、如果你没有额外依赖包，也要准备一个空的可执行文件./configure；
- 3、请提供Makefile进行编译生成，生成的目标文件名必须是main.exe，备注：ubuntu也可以是以.exe结尾。

4、评测命令：./main.exe 39.106.111.130 4010 ABC123DEFG

参数1：39.106.111.130，是裁判服务器的IP，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；
参数2：4010，是裁判服务器允许您连接的端口，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；
参数3：字符串，是本次运行的令牌，以便校验您的资格，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度生成。

Java

Java版本：1.8.0_162

mvn版本：3.5.3

mvn镜像：<http://maven.aliyun.com/nexus>

操作系统(64位)：Ubuntu 16.04.3 LTS下使用docker运行maven:3.5.3-jdk-8镜像

示例及注意事项请查看：<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForJava8>

- 1、不要修改pom.xml中的build节内的任何内容；
- 2、入口程序在./src/main/java/Main.java，不要变更main函数所在的文件；
- 3、我们最后的打包命令是：mvn clean install package

4、评测命令：java -jar output/UAVGoodsAI.jar 39.106.111.130 4010 ABC123DEFG

参数1：39.106.111.130，是裁判服务器的IP，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；
参数2：4010，是裁判服务器允许您连接的端口，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；
参数3：字符串，是本次运行的令牌，以便校验您的资格，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度生成。

java -version

openjdk version "1.8.0_162"

OpenJDK Runtime Environment (build 1.8.0_162-8u162-b12-1-deb9u1-b12)

OpenJDK 64-Bit Server VM (build 25.162-b12, mixed mode)

mvn -version

Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-24T19:49:05Z)

Maven home: /usr/share/maven

Java version: 1.8.0_162, vendor: Oracle Corporation

Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre

Default locale: en, platform encoding: UTF-8

OS name: "linux", version: "4.4.0-105-generic", arch: "amd64", family: "unix"

Python2

Python版本：2.7.14

pip镜像：<https://pypi.tuna.tsinghua.edu.cn/simple>

操作系统(64位)：Ubuntu 16.04.3 LTS下使用docker运行python:2.7.14镜像

示例及注意事项请查看：<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython2>

1、入口main在main.py中，请不要变更此文件名，更不要将main入口变更至别的文件；

2、您的依赖包请写在Makefile2文件中，注意M大写；

3、Makefile3文件格式：

每行一个依赖包

依赖包格式：库名字==版本号，例如numpy==1.14.2

4、评测命令：python2 main.py 39.106.111.130 4010 ABC123DEFG

参数1：39.106.111.130，是裁判服务器的IP，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；

参数2：4010，是裁判服务器允许您连接的端口，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成

参数3：字符串，是本次运行的令牌，以便校验您的资格，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度成。

5、评测系统已经安装的依赖包

numpy==1.14.2

scipy==1.0.1

matplotlib==2.2.2

scikit-learn==0.19.1

simpleai==0.8.1

tensorflow==1.7.0

paddlepaddle==0.11.0

注意，由于正式评测时每位选手只有8核16G的服务器，请谨慎使用机器学习框架。

Python3

Python版本：3.6.5

pip镜像：<https://pypi.tuna.tsinghua.edu.cn/simple>

操作系统(64位)：Ubuntu 16.04.3 LTS下使用docker运行python:3.6.5镜像

示例及注意事项请查看：<https://git.acmcoder.com/hikvision/UAVGoodsAIDemoForPython3>

- 1、入口main在main.py中，请不要变更此文件名，更不要将main入口变更至别的文件；
- 2、您的依赖包请写在Makefile3文件中，注意M大写；
- 3、Makefile3文件格式：

每行一个依赖包

依赖包格式：库名字==版本号，例如numpy==1.14.2

4、评测命令：python3 main.py 39.106.111.130 4010 ABC123DEFG

参数1：39.106.111.130，是裁判服务器的IP，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成；

参数2：4010，是裁判服务器允许您连接的端口，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度程序生成

参数3：字符串，是本次运行的令牌，以便校验您的资格，您在本地调试时，可在“我的对战”中获取；正式评测时，由评测调度生成。

5、评测系统已经安装的依赖包

numpy==1.14.2

scipy==1.0.1

matplotlib==2.2.2

scikit-learn==0.19.1

simpleai==0.8.1

tensorflow==1.7.0

注意，由于正式评测时每位选手只有8核16G的服务器，请谨慎使用机器学习框架。

赛事交流讨论群

欢迎加入QQ群交流讨论大赛技术及赛制问题！

软件精英挑战赛全国1群：684146505

软件精英挑战赛全国2群：692600084

目录

关注海康威视招聘官方公众号获取大赛及校园招聘实时信息！

特别申明

1. 参赛作品必须是原创作品，未侵犯第三方知识产权。且该作品未在报刊、杂志、网站（含开源社区）及媒体公开发表，未申请专利或进行版权登记的作品，未参加过其他比赛，未以任何形式进入商业渠道。
2. 参赛者或参赛团体同意，获奖作品相关知识产权归海康威视所有。获奖者不以同一作品形式参加其他的比赛或转让给他方；否则，海康威视将取消其参赛、入围与获奖资格，收回奖品及并保留追究法律责任的权利。
3. 因参赛作品而引发的第三方侵权纠纷，由参赛者或参赛团体承担全部法律责任。



