



# 2020 首届中国人工智能搅蛋大赛

赛前培训会>

江苏省人工智能学会

2020年9月27日

01

## 大赛简介

大赛背景、大赛目的、赛程赛制、奖项设置、组织机构等

02

## 竞赛规则

大赛比赛规则和晋级规则等

03

## 竞赛平台

大赛竞赛平台使用说明及参赛demo示例讲解

04

## 互动答疑

大赛启动会、宣传报名、组织竞赛、人机大赛、颁奖仪式等



## ➤ 大赛简介

- 01-1. 大赛官网
- 01-2. 竞赛目的
- 01-3. 赛程赛制
- 01-4. 奖项设置
- 01-5. 组织机构



# ➤ 大赛简介——大赛官网



大赛官网: [gd.jsai.org.cn](http://gd.jsai.org.cn)



大赛主宣传海报

## ➤ 大赛简介——大赛目的及赛程赛制



### 竞赛目的

人工智能在各种棋牌游戏和人类对战，其意义不在于输赢本身，更重要的是人们对这类游戏都耳熟能详，能够通过比赛了解到人工智能的最新进展，同时也希望人工智能在棋牌类游戏成功和突破能够启发其他方面的研究和应用，并能将人工智能技术创新应用到更多行业和领域，推动人工智能技术和产业发展。

举办首届“中国人工智能掬蛋大赛”主要目的，一是借助人们喜闻乐见的“掬蛋游戏”，向社会普及人工智能知识；二是通过“掬蛋”算法大赛，发掘算法精英人才，推动人工智能算法研究和应用；三是通过“掬蛋”算法大赛，进一步扩大江苏（淮安洪泽）影响力，带动江苏（淮安洪泽）文化旅游经济发展。



### 赛程赛制



赛事阶段	日期
报名	9月22日~10月15日
初赛	10月17日~10月18日
决赛	10月25日
人机大赛	10月29日
颁奖典礼	10月30日



## ➤ 大赛简介——奖金设置



奖金设置分为攒蛋大赛奖金及人机大赛奖金设置

奖项	数量	奖金	总奖金	备注
特等奖	1	¥ 50000	¥ 50000	第一名
一等奖	3	¥ 20000	¥ 60000	第2-4名
二等奖	4	¥ 10000	¥ 40000	第5-8名
三等奖	8	¥ 5000	¥ 40000	第9-16名
合计:			¥ 190000	

攒蛋算法大赛奖金设置

人机大赛奖项另外公布

## ➤ 大赛简介——组织机构



### 指导单位

江苏省科学技术协会 江苏省工业和信息化厅 淮安市洪泽区人民政府

### 主办单位

江苏省人工智能学会

### 承办单位

淮安市洪泽区工业和信息化局 江苏省人工智能学会机器学习专委会 江苏省大数据安全与智能处理重点实验室

### 协办单位

淮阴工学院 复杂系统智能控制与决策国家重点实验室

江苏省掇蛋运动协会 淮安掇蛋文化协会 淮安市洪泽区棋牌协会 淮安市计算机学会

### 技术支持

江苏万维艾斯网络智能产业创新中心有限公司



**指导委员会：**指导委员会由主办（承办、协办）单位领导、掇蛋游戏裁判等组成，主要负责制定大赛赛程，指导和协调大赛各项工作开展。

**评审委员会：**评委由人工智能领域的著名学者、专家组成，主要负责制定算法评审标准及具体评审工作。

**裁判委员会：**大赛裁判工作委托江苏省掇蛋运动协会组建裁判委员会，负责大赛裁判工作。



## ➤ 竞赛规则

---

01-1. 单局胜负规则

01-2. 初赛

01-3. 决赛



## ➤ 竞赛规则——单据胜负规则

### (一) 掇蛋游戏规则

本次竞赛所采用的掇蛋规则遵循2017年国家体育总局发布的《淮安掇蛋竞赛规则》（不完全相同）；

### (二) 掇蛋算法大赛单局胜负规则

- 1、打到等级A双上的一方团队获胜（记为一局），胜方积2分，败方积0分。（不存在平局，若50次洗牌后双方等级相同，则重新进行洗牌一次后根据等级高低判断）；
- 2、根据掇蛋规则，各队打到等级A，累计3局不“双上”则从等级2开始打；
- 3、根据座位随机分配结果，规定初始打1号位的等级2；
- 4、每支队伍将与不同对手进行多场对局，根据积分排名决定是否晋级。

### 线上竞赛

**赛程安排：** 10月17日—10月25日

**竞赛平台：** JSAI人工智能掼蛋平台（以下简称“掼蛋平台”）为大赛官方指定赛期唯一专用竞赛平台，掼蛋平台分为在线平台和离线平台两种。其中，在线平台供参赛团队正式竞赛使用，离线平台供参赛团队测试训练使用。

**参赛要求：** 已完成报名的参赛团队根据大赛官网“参赛指南”，编写人工智能算法程序（以下简称“掼蛋机器人”）通过在线平台参加竞赛，大赛组委会要求参赛团队编写两个掼蛋机器人（允许两个掼蛋机器人为同一程序），并保证掼蛋机器人可在离线平台上平稳运行。

**赛程分解：** 线上竞赛分为初赛、决赛两个阶段，大赛组委会将按期组织进行，并根据比赛结果及时公示晋级团队名单。

**竞赛结果：** 竞赛结果通过官网菜单栏【掼蛋平台】—【排行榜】公开。



### 晋级规则

**赛程安排：** 10月17日—10月25日

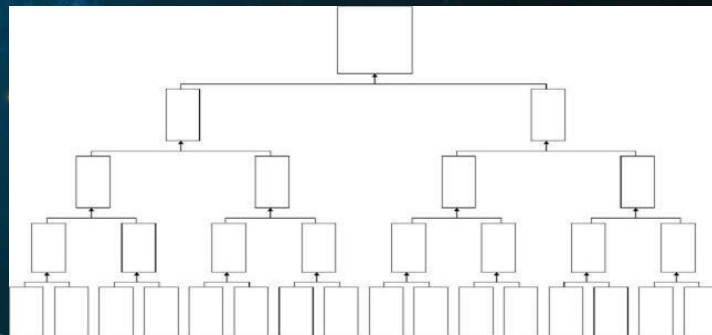
#### 1、初赛（10月17日、10月18日）

初赛采用分组循环淘汰赛制，通过在线撮蛋平台将所有参加初赛的参赛团队随机分组，每组由N支参赛团队组成，各小组内进行循环赛，各小组内根据总积分进行排名（积分相同，根据胜场数、胜负关系、胜局数、升级数的优先级顺序等进行排名），每组排名前N/2名的参赛团队晋级决赛，组内其余队伍被淘汰。（每组参赛团队数量视实际报名参赛团队数量而定）

最终决出**32个**参赛团队晋级决赛，并邀请参加人机大赛。

#### 2、决赛（10月25日）

决赛采用淘汰赛制，通过在线撮蛋平台将参加决赛的参赛团队重新进行随机匹配，进行两两比赛，胜者晋级，败者淘汰，最终决出冠军团队及其他获奖团队（获奖数量及奖金数额见奖项设置）。



## ➤ 竞赛规则——人机大赛

	AI	
AI	桌	AI
	AI	

观摩赛

观摩赛形式（暂定）

选前4名的捣蛋机器人参赛，播放其决赛阶段比赛过程或现场进行比赛，邀请捣蛋主持人进行解说。

...

	人	
AI	桌	AI
	人	

人机对抗赛

人机对抗赛形式（暂定）

真人团队（2人）对抗机器人团队（2个）。举行淘汰赛，共设置32桌，每桌1队机器人和1队真人。

...

	9段	
冠军AI	桌	冠军AI
	9段	

表演赛

表演赛形式（暂定）

选获得冠军的捣蛋机器人与蛋位最高的真人团队进行表演赛，并邀请捣蛋主持人和AI算法专家进行解说。

人员设备规模：每桌配4台电脑，真人选手2人，2个捣蛋机器人，裁判1人。

真人选手选择：蛋位5段（中间段位）至9段之间





## ➤ 竞赛平台

---

01-1. 离线平台

01-2. 在线平台

## ➤ 竞赛平台——离线平台

参赛第一步，运行平台，运行示例代码，熟悉离线平台的使用方法和示例代码的编写方法

1. 使用环境：Windows系统下，Linux与Mac环境将于10.07对外发布
2. 运行方法：在大赛官网下载压缩包解压后，双击运行即可
3. 常见问题解答：运行之后一闪而过即闪退

解决方案：将缺少的Qt5Core.dll添加到系统的PATH环境变量中即可

步骤1：在已有Python3环境下，打开命令行，输入命令：`pip install pyqt5`

步骤2：将路径 `~\Lib\site-packages\PyQt5\Qt\bin` 添加到系统的PATH环境变量

例如Python安装目录为D:\python3

则进将上述路径中的波浪号替换为安装目录，即D:\python3\Lib\site-packages\PyQt5\Qt\bin

例如使用Anaconda3在Example环境下安装的pyqt5，则目录为D:\Anaconda3\envs\Example\Lib\site-packages\PyQt5

若使用的是Anaconda3默认环境，则目录为D:\Anaconda3\Lib\site-packages\PyQt5

步骤3：再次双击运行离线平台即可

注：若无python环境，采用其他语言进行编码，亦可从Qt5官网下载安装，最后找到Qt5Core.dll所在目录，并将其添加到系统的PATH环境变量即可



## ➤ 竞赛平台——离线平台

出现如右侧所示两个即为运行成功

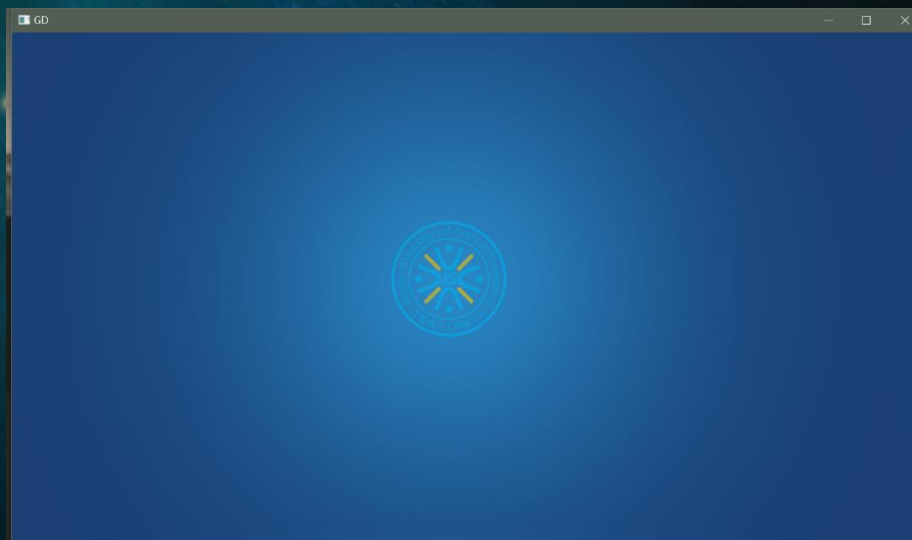
上方所示界面为命令行界面，选手可在该界面下看见本地服务器的通信信息，选手可在此处见到所有程序与服务器的通信信息。

下方所示界面为可视化界面，选手在该界面下可以直观的看见整个算法程序对战的过程。

两个程序切勿在开始离线训练时关闭其中任何一个，否则会产生严峻的后果。



```
D:\Far\gui\game.exe
[1 200909 23:57:03 game:41] [From Server]: Ready for connect.
```



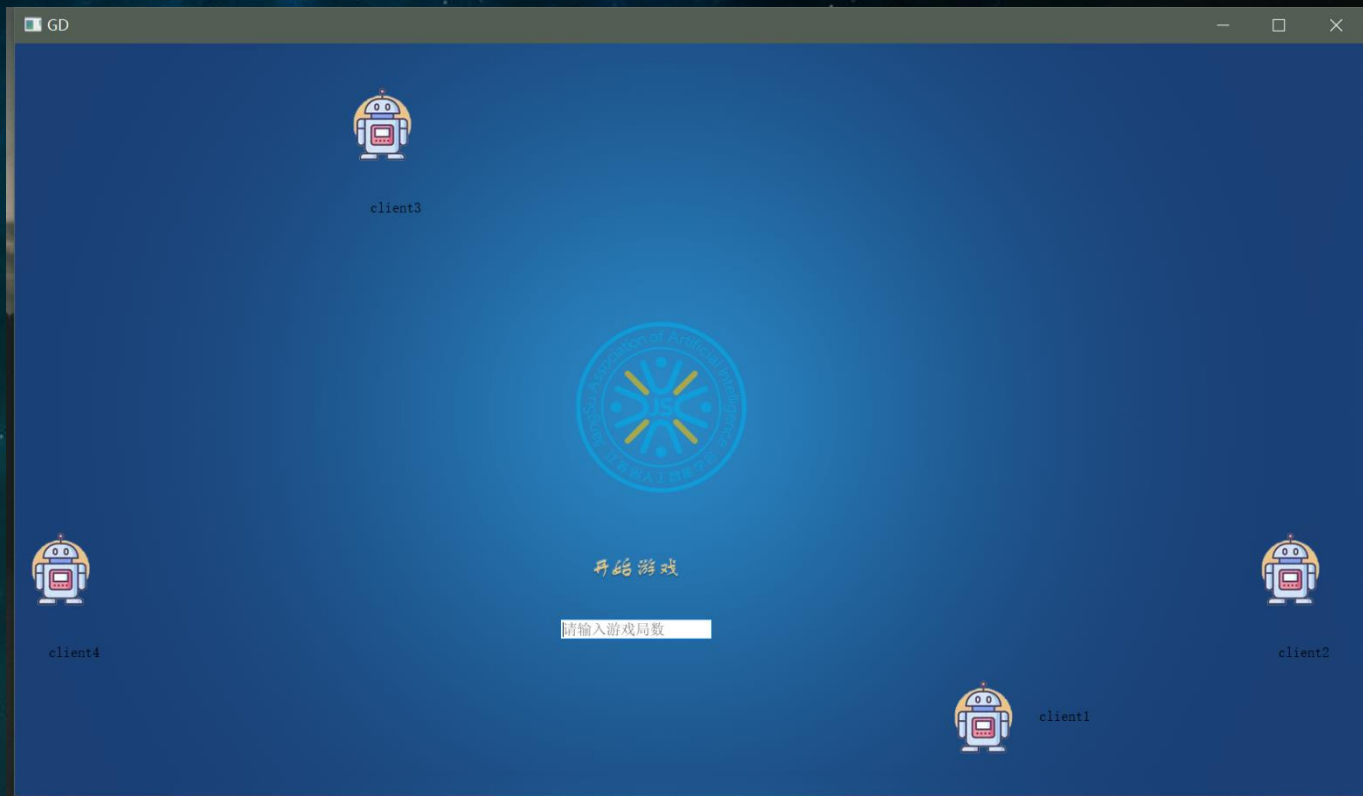
## ➤ 竞赛平台——离线平台

### 离线平台识别算法程序的方式

算法程序与离线平台之间建立websocket连接，离线平台固定连接地址前缀为ws://127.0.0.1:23456/game/，算法程序需要提供自身的标识供离线平台进行识别，例如其标识为client1，则连接地址则为ws://127.0.0.1:23456/game/client1

如右图所示，为四个算法程序成功连入离线平台的情况。选手可自行设定游戏局数，输入后点击开始游戏即可开始

**注意：**离线平台默认将第一个连入的算法程序和第三个连入的算法程序作为一队，将第二个和第四个连入的算法程序作为一队，不可更改。





## ➤ 竞赛平台——离线平台

编写掇蛋算法程序，从随机示例程序开始

如右图所示，为官方提供的随机示例算法程序，

即随机做出出牌动作

随机示例程序中包含4个函数：`__init__`、`opened`、

`closed`、`received_message`，分别代表程序初始化、

建立连接、连接关闭、接收到消息四个功能模块。

由于掇蛋是回合制类型游戏，算法程序无法主动

给离线平台发送消息，故只有在接收到离线平台

的消息时算法程序才可发送消息。

故选手在只需在`received_message`处编写代码即可，

其余部分无需关注

只有对消息格式熟悉了，才能够开始下一步的编

写，下面对消息格式进行解释

```
class ExampleClient(WebSocketClient):

    def __init__(self, url):
        super().__init__(url)
        self.name = "client1"

    def opened(self):
        pass

    def closed(self, code, reason=None):
        print("Closed down", code, reason)

    def received_message(self, message):
        content = json.loads(str(message))
        print(content)
        if "action_list" in content and content["action_list"]:
            card_type = random.choice(list(content["action_list"].keys()))
            print("Choose type ", card_type)
            rank = random.choice(list(content["action_list"][card_type].keys()))
            action = random.choice(list(content["action_list"][card_type][rank]))
            print("Choose action:", action)
            self.send(json.dumps({"action": action, "type": card_type, "rank": rank}))
```

## ➤ 竞赛平台——离线平台

平台与算法程序之间通过JSON数据格式来进行交互，整个游戏过程中，算法程序会收到3种平台所发来的JSON，（JSON是什么不再阐述），可以通过JSON字段中的type值来判断是何种类型。下面分别对3种JSON进行解释。

首先是第1种JSON，示例如下

```
{
  'type': 0,
  'self_rank': '2',
  'opponent_rank': '3',
  'current_rank': '3',
  'training_times': 10,
  'current_times': 1,
  'winners': [3, 2, 0, 1]
}
```

‘type’: 0 表示该条消息表示当前小局结束，该条消息会广播至所有算法程序，算法程序无需做出回应

‘self\_rank’: ‘2’ 表示当前我方所处的等级为2，opponent\_rank表示对方所处等级为3，current\_rank表示当前游戏所打的等级为3

‘training\_times’: 10 表示当前进行10局游戏，‘current\_times’: 1表示当前正在进行第一局游戏，只有再某一方打至A等级且双下后

‘current\_times’才会自增1，进入到下一局。

‘winners’:表示当前对局的玩家完牌的次序，若完牌次序为[1,3,0,2]即双下，那么0, 2的次序对积分不影响。



## ➤ 竞赛平台——离线平台

如右图所示，为算法程序会收到的第二种JSON数据格式。`'type': 1`表示该条消息为广播其他玩家所做出的动作，算法程序无需做出回应。

算法程序接收到该类型的消息时，可以首先对`current_stage`字段进行判断，`current_stage`的字段值为5表示当前处于进贡阶段。其次可以获取`current_player`的值，`current_player`的值为1表示当前1号位玩家进贡。进而可以从`action_performed`种获取进贡信息，`action_performed`表示进贡的牌为大王。结合上述字段的值即可简单判断当前游戏场面。

即`current_stage`表示当前游戏处于何阶段，`action_performed`表示当前正在进行的动作，`current_player`表示当前正在进行动作的player的座位号（座位号从0开始至3结束）

```
{
  'type': 1,
  'hand_cards': [
    [0, '2'], [3, '5'], [0, '6'], [1, '6'], [3, '7'], [0, '7'], [0, '7'], [2, '7'], [2, '7'],
    [2, '8'], [0, '8'], [0, '9'], [1, '9'], [0, '9'], [0, '10'], [0, '10'], [1, '10'],
    [3, 'Q'], [0, 'Q'], [3, 'K'], [3, 'K'], [2, 'A'], [3, 'A'], [2, 'A'], [2, '3'],
    [0, 'JOKER'], [1, 'JOKER']],
  'index': 0,
  'public': [
    {'rest': 27, 'play_area': []},
    {'rest': 26, 'play_area': [[0, 'JOKER']]},
    {'rest': 27, 'play_area': []},
    {'rest': 27, 'play_area': []}
  ],
  'current_player': 1,
  'current_stage': 5,
  'self_rank': '3',
  'opponent_rank': '3',
  'current_rank': '3',
  'action_list': None,
  'action_performed': {'action': [[0, 'JOKER']], 'type': 'tribute', 'rank': 'JOKER'},
  'current_times': 1,
  'training_times': 100
}
```

## ➤ 竞赛平台——离线平台

其余字段，如`hand_cards`表示自己的手牌信息，  
每张牌由一个长度为2的数组组成，数组的第一个  
元素表示花色，第二个元素表示卡牌的点数，且  
黑桃红桃梅花方片分别对应0-3，如`[0, '2']`即表示  
黑桃2，特别地，`[1, 'JOKER']`表示大王，`[0, 'JOKER']`  
表示小王，`[0, 'PASS']`表示PASS

`index`表示自己的座位号；

`public`表示公共信息，是一个长度为4的数组，数  
组对应的下标即可获取到对应player的公共信息，  
如`public[0]`获取的即为座位号为0的player的公共  
信息，`'rest'`表示该player所剩余卡牌，

`'play_area'`表示该player的出牌区域。

Player做出的所有动作，都将展示在`play_area`中

```
{
  'type': 1,
  'hand_cards': [
    [0, '2'], [3, '5'], [0, '6'], [1, '6'], [3, '7'], [0, '7'], [0, '7'], [2, '7'], [2, '7'],
    [2, '8'], [0, '8'], [0, '9'], [1, '9'], [0, '9'], [0, '10'], [0, '10'], [1, '10'],
    [3, 'Q'], [0, 'Q'], [3, 'K'], [3, 'K'], [2, 'A'], [3, 'A'], [2, 'A'], [2, '3'],
    [0, 'JOKER'], [1, 'JOKER']],
  'index': 0,
  'public': [
    {'rest': 27, 'play_area': []},
    {'rest': 26, 'play_area': [[0, 'JOKER']]},
    {'rest': 27, 'play_area': []},
    {'rest': 27, 'play_area': []}
  ],
  'current_player': 1,
  'current_stage': 5,
  'self_rank': '3',
  'opponent_rank': '3',
  'current_rank': '3',
  'action_list': None,
  'action_performed': {'action': [[0, 'JOKER']], 'type': 'tribute', 'rank': 'JOKER'},
  'current_times': 1,
  'training_times': 100
}
```



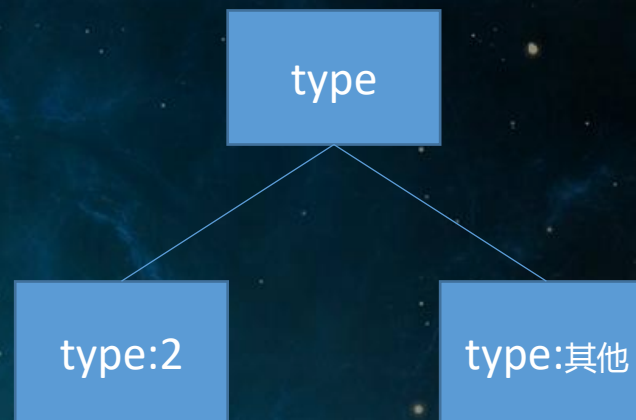
## ➤ 竞赛平台——离线平台

如右图所示，为算法程序会收到的第三种JSON数据格式。'type': 2表示该条消息为单播至该玩家的消息，算法程序需要做出回应。重复字段这里不在赘述。结合current\_stage、action\_performed和current\_player可以知道上家打出了PASS，目前最大的牌为[2, '8']，轮到我方做出动作。action\_list即表示当前情况下我方可以做出的动作，该action\_list仅提供给选手作为动作参考，选手可以自行生成动作。

```
{
  'type': 2,
  'hand_cards': [
    [2, '2'], [3, '2'], [3, '3'], [0, '3'], [0, '4'], [0, '5'], [2, '5'], [3, '6'], [2, '9'],
    [3, '9'], [2, 'J'], [3, 'K']],
  'index': 3,
  'public': [
    {'rest': 4, 'play_area': [[2, '8']]},
    {'rest': 5, 'play_area': [[1, 'K']]},
    {'rest': 13, 'play_area': [[0, 'PASS']]},
    {'rest': 12, 'play_area': []}],
  'current_player': 2,
  'current_stage': 2,
  'self_rank': 'A',
  'opponent_rank': '8',
  'current_rank': '8',
  'action_list': {'PASS': {'PASS': [[[0, 'PASS']]]}},
  'action_performed':
  {'action': [[0, 'PASS']], 'type': 'PASS', 'rank': 'PASS'},
  'current_times': 2,
  'training_times': 2
}
```

详细的JSON数据格式说明可以通过官网的文档进行查阅。  
下面开始编写捣蛋算法程序的第一步，先判断何时轮到自己，需要自己做出动作选择。

如右图所示，可以有两种方式作为判断，通过type的值或者action\_list是否为空。算法程序只有在type的值为2时需要做出行动或者在action\_list非空时做出行动。





## ➤ 竞赛平台——离线平台

编写掇蛋程序的第二步，即如何去生成一个动作，或者说如何去表示一个动作。

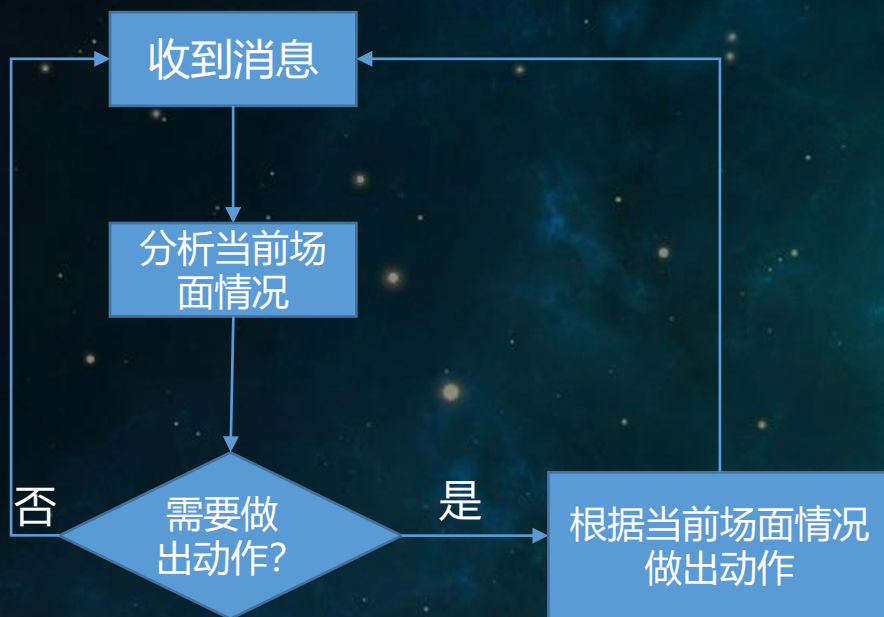
由于掇蛋有红桃配的存在，所以很多牌型如果没有上下文，则会引发歧义或者其他不必要的判断，如牌型A：红桃3、红桃3、红桃5、红桃5、黑桃4、黑桃4，当前等级打5，那么红桃5则为红桃配，在没有上下文的情况下，牌型A可以理解为333444或者334455，亦或者223344。为了省去不必要的判断，一个动作包含了action、type、rank三个字段值，如右图所示。Action表示该动作组成的卡牌，type表示该牌型的类型（钢板、三连对、三带二等），rank表示该牌型的点数。如26789，2为红桃配，可以产生56789和678910的两种顺子牌型，rank表示该牌型的起始点数。顺子的rank最小从A开始，最大到10；其他牌型同理。

特别地，我们规定PASS动作的表示固定为：{'rank': 'PASS', 'type': 'PASS', 'action': [[0, 'PASS']]}, 大小王的rank均为JOKER



## ➤ 竞赛平台——离线平台

知道了如何判断场面情况，何时该做出动作，动作如何表示，即可按照自己的想法开始攒蛋编写，一个一般的框架如下图所示：



当然如何科学地、智能地分析当前场面的情况并做出动作，则需要选手自行去考虑以上，就是离线平台的介绍和算法程序编写的一个大致步骤和框架



## ➤ 竞赛平台——在线平台

在线平台的消息数据格式与离线平台的大致相同，只不过从本地运行变成了连网对战，  
具体说明详见官网



➤ 互动答疑

---



谢谢观看