

judger的使用说明

judger的使用说明

- 1、环境说明
- 2、使用说明
 - 使用方法一：仅用于AI和AI对战
 - 使用方法二：使用测试模式

1、环境说明

python版本：经测试，**python 3.7.6以上版本**可用。python 3.7.6以下版本未经测试，由于基本没有用到新特性，因此预计python3以上基本都是可用的，如果使用过程中出现了问题，建议将版本升级到3.7.6及以上。

依赖库：除python原生支持的库之外，还需要您安装**websockets库**（在安装python和pip工具包的前提下）

```
1 | pip install websockets
```

2、使用说明

judger用来让选手进行简单的本地测试用，提供了简陋的本地人人、人机、机机对战功能。

需要注意的事情是，为了加强评测安全性和增强选手的可调试性，以及出于性能方面的考虑，网站评测端进行了从架构到代码方面的彻底重构，因此只兼容了本地测试用的judger的接口和通信协议。由于本地环境和评测环境可能存在差异，本地judger和评测时使用的在输入输出限制和空间限制等方面略有不同，因此本地judger测试通过并不能代表交到网站上就绝对没有问题，但是使用本地judger测试可以帮助您在交到网站上之前避免绝大多数的错误。

使用方法一：仅用于AI和AI对战

直接使用命令行参数启动，格式如下（切到judger.py和rserver.py对应的目录下）

```
1 | python judger.py <启动游戏逻辑指令> <启动AI0指令> <启动AI1指令> <配置信息> <生成录像文件路径>
```

这里启动游戏逻辑和启动AI的指令格式说明如下：

- 如果是编译出来的可执行文件，直接使用相对路径或者绝对路径即可
- 如果是python等解释型语言，则将启动指令中的空格换成+号，比如把python main.py变成python+main.py

需要注意的是：如果使用相对路径，应该使用相对judger.py的路径；由于+和空格的替换，因此路径名中请不要含有+或者空格。

配置信息是已经废弃的协议，因此这里随便输入个字符串即可；**生成录像文件**路径同样的，如果使用相对路径，则需要使用相对judger.py的路径

一个可行的示例如下(直接从游戏包下载里面的文件结构，将example_py.zip解压到其目录下，如果您变更了文件结构，可能需要自行修改)：

```
1 python judger.py python+../main.py python+../example/main.py
python+../example/main.py miracle replay.bin
```

在使用本指令后, 您可能会发现如下错误:

```
1 Traceback (most recent call last):
2   File "../example/main.py", line 10, in <module>
3 Traceback (most recent call last):
4   File "../example/main.py", line 10, in <module>
5     from card import CARD_DICT
6   File "C:\Users\andy\desktop\codes\example\card.py", line 53, in
  <module>
7     from card import CARD_DICT
8     DATA = json.load(open("Data.json", "r"))
9   File "C:\Users\andy\desktop\codes\example\card.py", line 53, in
  <module>
10 FileNotFoundError: [Errno 2] No such file or directory: 'Data.json'
11     DATA = json.load(open("Data.json", "r"))
12 FileNotFoundError: [Errno 2] No such file or directory: 'Data.json'
```

不用慌张, 这同样是因为文件相对路径引起的问题, 您只需要将example解压出来得到的Data.json文件拷贝一份到game/Judger目录下即可。

在使用命令行参数启动之后, 您会发现终端里输出了一些含有type和success之类的东西, 这是软件工程单元测试的时候使用的一些废弃协议, 不用在意它。

最终在游戏结束后, 会出现类似如下的输出:

```
1 {"0": 30003, "1": 26003}
```

这表示0号玩家和1号玩家获得的分数, 有关分数的定义请参考游戏规则说明文档(tutorial.pdf)

同样的, 按照上面样例指令的含义所示, 在游戏结束后您可以在Judger目录下找到replay.bin, 也就是本地测试的录像文件, 使用说明参考播放器的离线模式说明。

由于AI和逻辑的标准输入输出都被重定向到管道了, 因此不建议您本地使用标准输入输出进行调试, 这可能会给您带来不必要的麻烦。几种可行的输出中间变量的调试方案如下(同样适用于人机):

- 将中间变量的输出输出到文件里, 建议您每次输出到文件后及时进行flush, 否则可能会在AI进程结束后还有部分缓冲区内的消息没有写入文件内。
- 在judger.py中增加输出, 由于judger的标准输入输出流并没有被重定向, 因此您可以在judger中输出一些信息便于您的调试。因为某些历史原因, judger的代码非常的丑陋而且难读, 您可以只关注Player类里的write方法和Judger类里的write方法, 分别对应着游戏逻辑向AI发送的消息和AI向游戏逻辑发送的消息
- 在游戏逻辑的代码里将中间变量输出到文件里, 由于游戏逻辑的代码文件较多, 推荐您只关注main.py里的方法。

此外, 还存在修改judger内嵌AI和游戏逻辑, 使用调试工具等等方法来帮助您进行调试, 由于可能需要您做较大的代码改动或者对相关调试工具有一定的了解, 这里就不再赘述。

使用方法二: 使用测试模式

同样的, 在judger目录下, 输入以下指令

```
1 python judger.py test_mode
```

您可以进入简陋的用于测试的测试模式 (非常丑陋, 当时赶工测试赶出来的, 勿喷)

在进入测试模式后, 您可以输入**help**查看您可以使用的指令说明, 这里仅介绍下如何开启一局本地人机对局

对于启动AI, 您可以输入如下指令格式(格式检查较为严格, 多空格等情况也可能导致无效指令):

```
1 | 0 <index> <command>
```

其中的说明跟之前命令行启动的时候的指令说明类似, 表示启动的AI编号, 一个样例指令如下:

```
1 | 0 0 python+../../example/main.py
```

这里就启动了一个AI编号为0, 即先手

对于人类玩家, 您可以输入如下指令格式:

```
1 | 1 <index> <ip> <port> <room_id>
```

对于本地测试而言, 指人类玩家的编号, 指judger监听的ip地址和端口号, <room_id>同样是废弃的协议, 随便输入个数字即可.

如果是本地对战, 一个样例指令如下:

```
1 | 1 1 127.0.0.1 9010 1
```

输入指令之前, 请确保您对应的端口号没有被其他进程占用.

之后打开您的播放器, 在online mode里面输入

```
1 | <ip>:<port>/<room_id>/<name>/<seat>
```

其中<room_id>与上面指令相一致, 就是, 在本地测试中随便填写一个字符串即可, 一个可行的样例是(对应上面的指令):

```
1 | 127.0.0.1:9010/1/Aglove/1
```

在输入token后点击ok, 你会发现播放器弹出无法decode的错误提示, 您不需要惊慌, 只需要淡定的点下ok就可以继续连接了.

在连接成功后, 您的控制台上会有如下输出

```
1 | {"type": 1, "index": 1, "success": 1}
```

这个可以作为连接成功的标志

在双方玩家无论是AI还是人类都已经启动或者连接成功后, 您可以使用如下指令来**启动游戏**:

```
1 | 4 <command> <config> <replay>
```

是启动游戏逻辑的指令, 含义跟上面的一样; config是废弃的协议, 随便填写个字符串即可; 是生成录像的路径, 关于路径的说明跟上面的一样.

一个样例指令是:

在启动游戏后, 对局就开始了, 如果是人人或者人机, 您就可以在本地进行播放器相关的对战操作.

在对局结束后, 你可以在控制台上看到最后的分数, 跟命令行启动的分数含义是一样的. 后面可能会出现一些warning信息关于task的销毁的, 您不用理会他们. 同样的, 对局录像文件也会生成在你指定的路径下.

如果您想通过测试模式启动人机或者人人对战, 通过修改上面的指令也是可以做到的, 您甚至可以通过测试模式和您的好友远程联机, 只需要修改对应的并将其告知您的好友即可.

祝您使用愉快, 如果在使用过程中出现了一些问题, 请先仔细阅读本说明文档, 如果仍然无法解决您的问题, 请在QQ群里提出, 我们会很快为您处理.