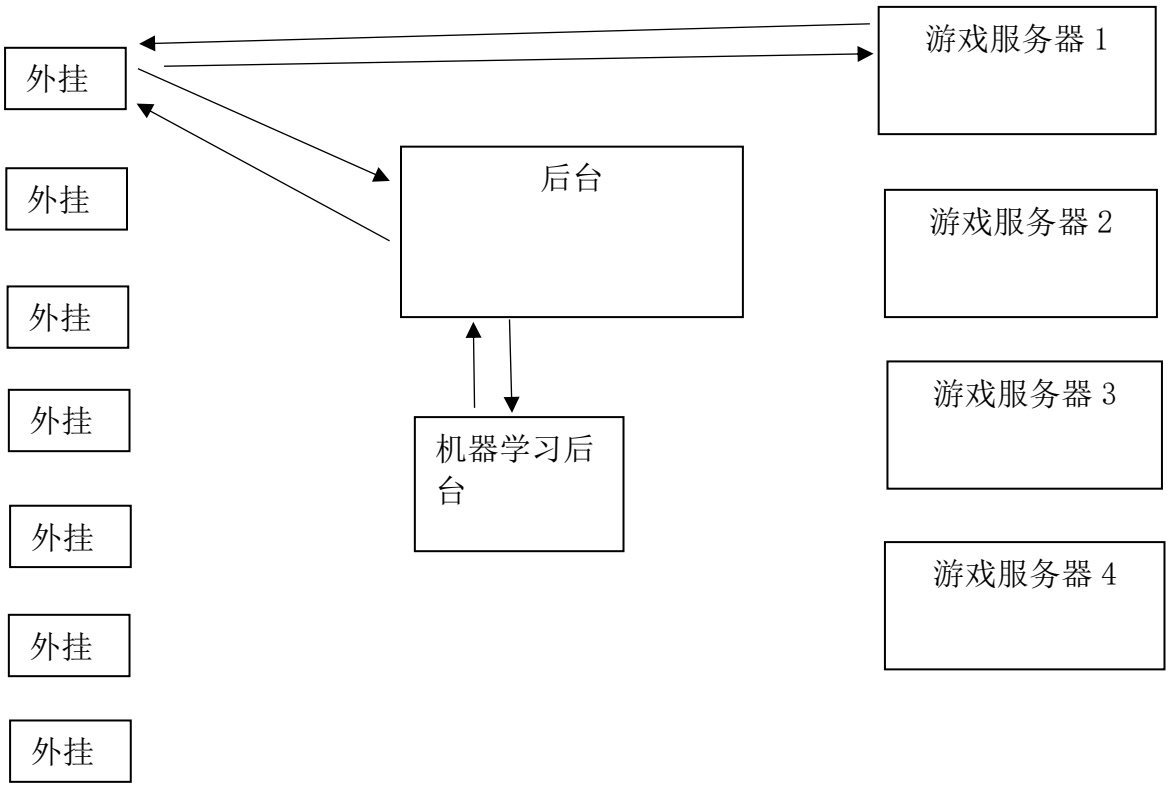
1、项目概况

业务逻辑图



- 游戏为打牌游戏，每个外挂为一个玩家客户端，有自己的 ip
- 游戏服务器是其他棋牌游戏公司运营的，后台可通过工具获取游戏服务器地址列表
- 外挂连接游戏服务器前，先请求后台，获取要连接的游戏服务器 ip
- 外挂出牌做决策时，先将牌局信息发给后台，后台再请求机器学习后台接口获得出牌决策，然后将决策返回给外挂，外挂再和游戏服务器交互

后台要实现的功能

- 服务器负载均衡，让每个游戏服务器上连接的外挂数量均衡
- 外挂客户端上线时向其提供账号密码以及游戏服务器地址端口信息（未完成）
- 向人工智能后台请求出牌策略
- 发现在两个外挂在同一个游戏房间的情况,打牌中请求出牌策略时加入其他外挂的牌信息
- 信息展示页面，包括游戏服务器列表，外挂客户端列表，人类玩家列表，各牌局信息

2、游戏服务器相关

2.1、游戏服务器信息上报接口

简要描述：

- 由外挂工具每个一定时间上传一个服务器信息给后台

请求 URL：

- http://127.0.0.1:8080/postServerList

请求方式：

- POST

参数：

参数名	必选	类型	说明
serverId	是	string	房间 ID
serverName	是	string	房间名称
serverIp	是	string	房间 IP
serverPort	是	string	房间端口
onLineCnt	是	string	在线人数
minEnterScore	是	string	最低进入分数
maxEnterScore	是	string	最高进入分数
maxConnectCnt	是	string	最大连接数

输入示例

```
  {
   "maxConnectCnt" : 60000,
   "maxEnterScore" : 100,
   "minEnterScore" : 100,
   "onLineCnt" : 199,
   "serverId" : 9001,
   "serverIp" : "120.25.165.108",
   "serverName" : "0.05 元斗地主场",
   "serverPort" : "9001"
}
```

返回示例

```
  {
    "code": 200,
    "msg": "SUCCESS",
    "data": ""
}
```

3、打牌相关

3.1、获取服务器信息接口

简要描述：

- 打牌客户端外挂进入游戏前获取游戏账号密码和一个服务器链接信息

请求 URL：

- http://127.0.0.1:8080/getConnInfo

请求方式：

- GET

参数：

无

返回示例

```
  {
    "code": 200,
    "msg": "SUCCESS",
    "data": {
            "uid":"2233322",
            "password":"1243",
            "serverIp":"192.168.1.1",
            "serverPort":"55"
            }
    }
```

返回参数说明

参数名	类型	说明
uid	String	用户登录 id

参数名	类型	说明
password	String	用户登录密码
serverIp	String	游戏服务器 ip
serverPort	String	游戏服务器密码

3.2、打牌接口

简要描述：

- 打牌客户端外挂打牌时和后台交互接口

请求 URL：

- http://127.0.0.1:8080/postCardPlayInfo

请求方式：

- POST

请求示例

```
  {
    "action": "4",//打牌动作 0->坐下，1->发牌，2->叫地主，3->加倍，4->出牌，5->展示结果
    "active_play": "0",//0->被动出牌，1->主动出牌
    "blind_cards": "82x",//3 张底牌
    "chair_id": "2",//玩家编号
    "last_play_id": "3",//上次出牌信息
    "lord_chair_id": "3",//地主编号
    "orig_cards": "d2KKQJTT998777665",//原始所有牌
    "play_cards": "3,456789TJ,2,0,1,0,3,K,2,0,1,2,3,x",//所有出过的牌，例如 3 号玩家为地主，456789TJ，其下家是 2 号玩家 pass，1 号玩家 pass，之后 3 号玩家出牌 K，以此类推
    "player": [{
        "chairid": "1",//该玩家编号
        "id": 0,//外挂客户端 id
        "ip_zone": "广东",//玩家地区
        "last_play_card": "2",//上次出的牌
        "level": "1",//玩家等级
        "multi": "3",//加倍情况
        "nick_name": "50670221",//玩家昵称
        "own_cards": "",//目前手上的牌
        "remain_count": "16",//剩余张数
        "role": "2",//2->农民，1->地主
```

```
        "score": "574",//目前分数
        "status": "2",//玩家状态
        "uid": "50670221",//玩家 id
        "unplayed_cards": "",//目前牌局还没有出的牌
        "uuid": "",//只有外挂有该字段，其余人类玩家没有
        "vip_level": "0"//玩家 vip 等级
    }, {
        "chairid": "2",
        "ip_zone": "西藏",
        "last_play_card": "",
        "level": "8",
        "multi": "0",
        "nick_name": "18598069",
        "own_cards": "",
        "remain_count": "17",
        "role": "0",
        "score": "854",
        "status": "2",
        "uid": "18598069",
        "unplayed_cards": "",
        "uuid": "",
        "vip_level": "0"
    }, {
        "chairid": "3",
        "ip_zone": "广东",
        "last_play_card": "",
        "level": "9",
        "multi": "0",
        "nick_name": "66864420",
        "own_cards": "",
        "remain_count": "17",
        "role": "0",
        "score": "1168",
        "status": "2",
        "uid": "66864420",
        "unplayed_cards": "",
        "uuid": "",
        "vip_level": "0"
    }],
    "server": {
        "id": "9011",
        "ip": "120.77.237.146",
        "name": "0.05 元斗地主场",
        "port": "9011"
```

```
    }
}
```

返回示例

```
{
    "code": 200,
    "msg": "SUCCESS",
    "data": "d"
}
```

4、人工智能交互接口

4.1、人工智能交互接口（暗牌）

简要描述：

- 本版本斗地主算法是根据已知信息（当前玩家手中牌，所有玩家出的牌以及未出的牌）用深度学习训练的算法。本算法考虑了玩家所处的位置（农民派 or 地主派，农民派的话农民为地主上家还是地主下家），玩家手中的牌，要管的牌为哪个玩家所处，出牌类型，以及主被动出牌等因素
- 暗牌表示该游戏桌子上只有该外挂一个玩家，无其他外挂

接口请求地址以及方式

- 网络通信是用 python 中的 socket 写的，data 为算法需要的信息，格式如下所示。data 传入服务器端后运行算法，输出为字符串，例如用上述 data 作为输入，输出则为：56789，若当前玩家不出则输出：0，若 data 数据有误则返回-1
- 如果只是调用算法，那么可以用下面的服务器地址和端口。 serverName = 'riseworlds.51vip.biz' #服务器地址 serverPort = 8000 #服务器端口

  请求示例

```
  {

    "ddz_playcard": "1001",//出牌，此为暗牌出牌标志位，为固定值
    "landlord": "3",//地主编号
    "to_play": "2",//当前出牌玩家编号
    "board_list": "3,456789TJ,2,0,1,0,3,K,2,0,1,2,3,x",//所有出过的
牌
```

```
    "xj_count": "16",//下家手上的牌的张数
    "card_array": "d2KKQJTT998777665",//
    "othercard": "22AAAAKQQQJJT9886554443333",//为出过的牌
    "activeplay": "0",//被动出牌
    "player": "3",//上轮出牌着编号
    "play": "x"//上轮出的牌
}
```

返回参数说明

- 返回的数据即你要出的牌如"kk",表示你要出一对 k

4.2、人工智能交互接口（明牌 ）

简要描述：

- 本版本斗地主算法是根据已知信息（当前玩家手中牌，所有玩家出的牌以及未出的牌）用深度学习训练的算法。本算法考虑了玩家所处的位置（农民派 or 地主派，农民派的话农民为地主上家还是地主下家），玩家手中的牌，要管的牌为哪个玩家所处，出牌类型，以及主被动出牌等因素
- 明牌表示该游戏桌子上发现其他外挂，请求出牌时要将该外挂玩家的牌信息加上

接口请求地址以及方式

- 网络通信是用 python 中的 socket 写的，data 为算法需要的信息，格式如下所示。data 传入服务器端后运行算法，输出为字符串，例如用上述 data 作为输入，输出则为：56789，若当前玩家不出则输出：0，若 data 数据有误则返回-1
- 如果只是调用算法，那么可以用下面的服务器地址和端口。 serverName = 'riseworlds.51vip.biz' #服务器地址 serverPort = 8000 #服务器端口

  请求示例

```
  {
   'ddz_playcard' : 1002,#出牌，此为明牌出牌标志位，是固定值
   'landlord' : 3, #地主编号
   'to_play' : 2, #当前出牌玩家
   'activeplay': 1, #主动出牌为 1，被动出牌为 0
   'player': '3', #上轮出牌者
   'card_array': '5677892', #当前玩家的牌
   'othercard': '3334455667888999TQQQKAAAA222xd', #未出的牌
```

```
    'play': '0', #上一轮出的牌，也是待管牌，假设地主 0 号玩家出了 TTTJ，轮到 1 号玩家出牌了，那么 play="TTTJ"
    'xj_count': 13, #当前玩家的下家手中牌的张数
    'board_list': '3,TTTJ,2,4JJJ,1,0,3,QKKK,2,0,1,0,3,34567,2,0,1,0', #所有出过的牌，例如 3 号玩家为地主，出牌 TTTJ，其下家是 2 号玩家，出牌 4JJJ，1 号玩家 pass，之后 3 号玩家出牌 QKKK，以此类推
    'card_xj':'3334455667888999',#当前玩家下家的牌
    'card_sj':''#当前玩家上家的牌
}
//如果当前玩家与其下家为一伙，那么 card_xj 就传当前玩家下家的牌，否则 card_xj 就为'0'或者'';
//如果当前玩家与其上家为一伙，那么 card_sj 就传当前玩家上家的牌，否则 card_sj 就为'0'或者'';
```

返回参数说明

- 返回的数据即你要出的牌如"kk",表示你要出一对 k

5、页面展示相关

5.1、游戏服务器列表页面

简要描述：

- 展示游戏服务列表

请求 URL：

- http://127.0.0.1:8080/getServerList

页面示例

房间ID	房间名称	房间IP	房间端口	在线人数	最低进入分数	最高进入分数	最大连接数	更新时间
9001	0.05元斗地主场	203.107.32.203	9001	132	100	100	60000	2019-08-25 06:57:48
9011	0.05元斗地主场	120.77.237.146	9011	207	100	100	60000	2019-08-28 11:22:41
9005	0.05元斗地主场	120.77.239.45	9005	162	100	100	60000	2019-08-25 08:27:49
9010	0.05元斗地主场	203.107.32.203	9010	142	100	100	60000	2019-08-25 07:47:48
9004	0.05元斗地主场	203.107.32.203	9004	112	100	100	60000	2019-08-25 06:07:47

5.2、玩家列表

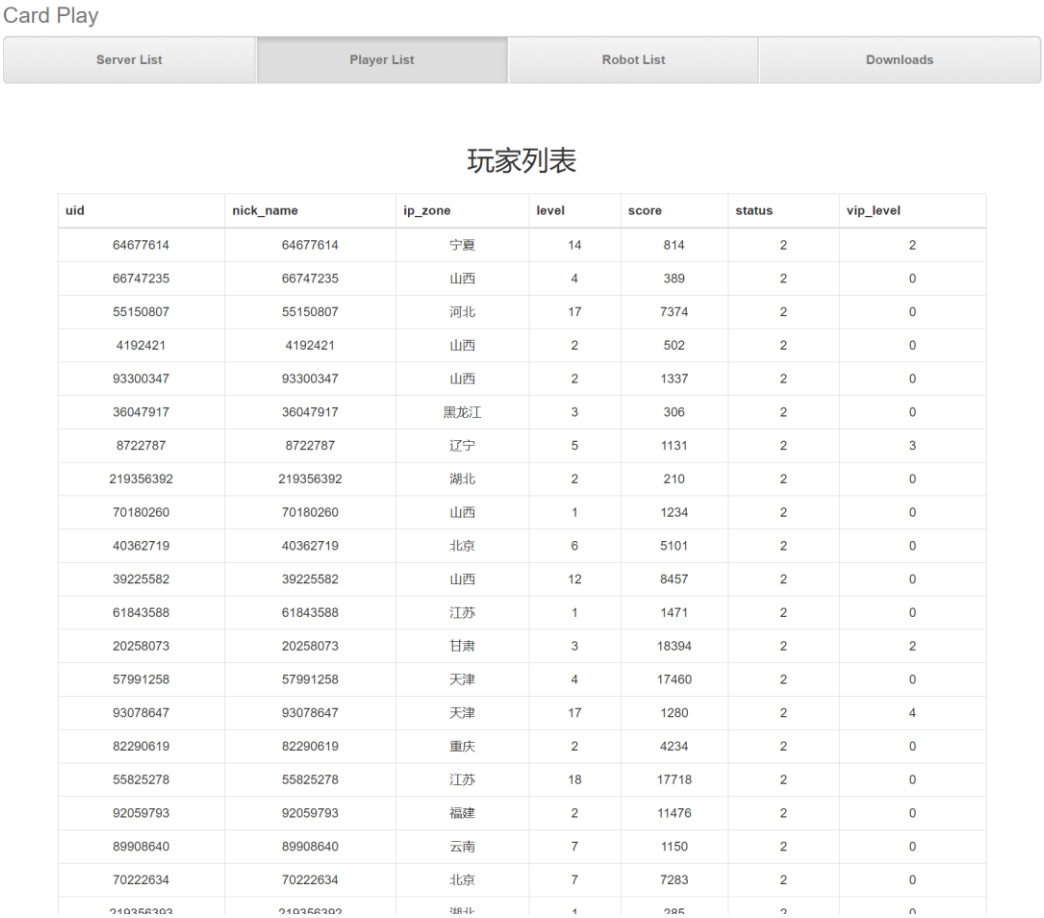
简要描述：

- 展示遇到过的人类玩家信息

页面 URL：

- http://127.0.0.1:8080/getPlayerList

页面示例



uid	nick_name	ip_zone	level	score	status	vip_level
64677614	64677614	宁夏	14	814	2	2
66747235	66747235	山西	4	389	2	0
55150807	55150807	河北	17	7374	2	0
4192421	4192421	山西	2	502	2	0
93300347	93300347	山西	2	1337	2	0
36047917	36047917	黑龙江	3	306	2	0
8722787	8722787	辽宁	5	1131	2	3
219356392	219356392	湖北	2	210	2	0
70180260	70180260	山西	1	1234	2	0
40362719	40362719	北京	6	5101	2	0
39225582	39225582	山西	12	8457	2	0
61843588	61843588	江苏	1	1471	2	0
20258073	20258073	甘肃	3	18394	2	2
57991258	57991258	天津	4	17460	2	0
93078647	93078647	天津	17	1280	2	4
82290619	82290619	重庆	2	4234	2	0
55825278	55825278	江苏	18	17718	2	0
92059793	92059793	福建	2	11476	2	0
89908640	89908640	云南	7	1150	2	0
70222634	70222634	北京	7	7283	2	0

5.3、外挂客户端列表

简要描述：

- 展示所有外挂客户端的信息

页面 URL：

- http://127.0.0.1:8080/getRobotPlayerList

页面示例

Card Play

Server List | Player List | Robot List | Downloads

机器玩家列表

uid	uuid	nick_name	ip_zone	level	score	status	vip_level	last_connect_ip	password	game_info	operation
220640444	dd7b7ee029df9247ee98d400ff6e804f	220640444	湖北	4	114	2	0	192.168.1.11		查看	删除
220640445	dd7b7ee029df9247ee98d400ff6e804f	220640445	湖北	3	194	2	0	192.168.1.11		查看	删除
222192651	fb4ddf3dd5febe4c7a97fe643417d31f	222192651	湖北	1	300	2	0	192.168.1.2		查看	删除
18598069	dd7b7ee029df9247ee98d400ff6e804g	18598069	西藏	8	854	2	0	192.168.1.11		查看	删除