

炉石兄弟 DIY 汇总

wjh 收录于 [Hearthbuddy](#)

□ 2020-03-15 □ 2024-02-11 □ 约 13400 字 □ 预计阅读 27 分钟

2021.3.27 新版智能SIM转移, CardDB_cardIDEnum.cs生成脚本

□ Python

```
1 # encoding: utf-8
2 import requests
3 import re
4 import json
5 import os
6 from tqdm import tqdm
7
8 ClassName = {
9     "DREAM": "梦境",
10    "WHIZBANG": "威兹班"
11 }
12
13 TypeName = {
14     "ENCHANTMENT": "附魔",
15     "HERO_POWER": "英雄技能"
16 }
17
18 CardSetName = {
19     "TB": "TB乱斗模式",
20     "HERO_SKINS": "HERO英雄皮肤和技能",
21     "THE_BARRENS": "BAR贫瘠之地的锤炼",
22     'SCHLOMANCE': "SCH通灵学园",
23     'BASIC': "CORE基本",
24     'BATTLEGROUNDS': "BGS酒馆战棋",
25     'BOOMSDAY': "BOT砰砰计划",
26     'BRM': "BRM黑石山的火焰",
27     'BLACK_TEMPLE': "BT外域的灰烬",
28     'DEMON_HUNTER_INITIATE': "BT恶魔猎手新兵",
29     'YEAR_OF_THE_DRAGON': "DRG巨龙降临",
30     "TGT": "AT冠军的试炼",
31     'GANGS': "CFM龙争虎斗加基森",
32     'CORE': "CORE基本",
33     'CREDITS': "CREDITS暴雪制作人员",
34     'EXPERT1': "EX经典",
35     'HOF': "HOF荣誉室",
36     'DALARAN': "DAL暗影崛起",
37     'DARKMOON_FAIRE': "DMF疯狂的暗月马戏团",
38     'DRAGONS': "DRG巨龙降临",
39     'NAXX': "NAX纳克萨玛斯",
```

```
40     'GILNEAS': "GIL女巫森林",
41     'GVG': "GVG地精大战侏儒",
42     'ICECROWN': "ICC冰封王座的骑士",
43     'UNGORO': "UNG勇闯安戈洛",
44     'LOOTAPALOOZA': "LOOT狗头人与地下世界",
45     'KARA': "KAR卡拉赞之夜",
46     'LOE': "LOE探险者协会",
47     'OG': "OG上古之神的低语",
48     'MISSIONS': "MISSIONS新手训练",
49     'TROLL': "TRL拉斯塔哈的大乱斗",
50     'TAVERNS_OF_TIME': "TOT时光酒馆",
51     'ULDUM': "ULD奥丹姆奇兵",
52     'VANILLA': "VAN经典模式"
53 }
54
55 file_list = requests.get("https://api.hearthstonejson.com/v1/").text
56 ver_list = re.findall("/v1/(\d+)/all/", file_list)
57 new_version = max(ver_list)
58 print(f"new_version: {new_version}")
59
60 print("loading global_json...")
61 global_json = requests.get('https://api.hearthstonejson.com/v1/strings/zhCN/GLOBAL')
62 global_data = json.loads(global_json)
63 assert global_data is not None
64 for name in global_data:
65     if "GLOBAL_CLASS_" in name:
66         class_name = name.replace('GLOBAL_CLASS_', '')
67         if class_name not in ClassName:
68             ClassName[class_name] = global_data[name]
69     if "GLOBAL_CARD_SET_" in name:
70         card_set = name.replace('GLOBAL_CARD_SET_', '')
71         if card_set not in CardSetName:
72             CardSetName[card_set] = global_data[name]
73     elif "GLOBAL_CARDTYPE_" in name:
74         card_type = name.replace('GLOBAL_CARDTYPE_', '')
75         if card_type not in TypeName:
76             TypeName[card_type] = global_data[name]
77
78 print("loaded global_json successfully!")
79
80 print("loading card_data...")
81 cardJson_data = ""
82 cardJson_File = f'{new_version}.json'
83 if os.path.exists(cardJson_File):
84     print("--file mode")
85     with open(cardJson_File, "r", encoding='utf-8') as f:
86         cardJson_data = f.read()
87     assert cardJson_data != ""
88 else:
89     cardJson_url = f'https://api.hearthstonejson.com/v1/{new_version}/all/cards.js'
```

```
90     print(f"--online mode({cardJson_url})")
91     cardJson_req = requests.get(cardJson_url, stream=True)
92     cardJson_byte = b''
93     pbar = tqdm(total=-1, unit='B', unit_scale=True)
94     for chunk in cardJson_req.iter_content(chunk_size=1024):
95         assert chunk != None
96         cardJson_byte += chunk
97         pbar.update(1024)
98     pbar.close()
99     cardJson_data = cardJson_byte.decode()
100    assert cardJson_data != ""
101    with open(cardJson_File, "w", encoding='utf-8') as f:
102        f.write(cardJson_data)
103
104    cardData_temp = json.loads(cardJson_data)
105    assert cardData_temp is not None
106    print("loaded card_json successfully!")
107
108    cardData = {}
109    for c in cardData_temp:
110        cardData[c['id']] = c
111
112    sim_path = os.path.join(os.getcwd(), "cards")
113    print("loading sim_data from", sim_path)
114    Sim_Context = []
115    Sim_CardID = []
116    Sim_text_idx = {}
117    Sim_id_idx = {}
118    for root, dirs, files in os.walk(sim_path):
119        for file in files:
120            card_id = file.replace('Sim_', '').replace('.cs', '')
121            with open(os.path.join(root, file), "r", encoding='utf-8') as sim_file:
122                sim_content = sim_file.read()
123                if "public" not in sim_content:
124                    continue
125                card_idx = len(Sim_Context)
126                Sim_Context.append(sim_content)
127                Sim_CardID.append(card_id)
128                Sim_id_idx[card_id] = card_idx
129                if card_id in cardData and 'text' in cardData[card_id]:
130                    Sim_text_idx[cardData[card_id]['text']['zhCN']] = card_idx
131    print("loaded " + str(len(Sim_Context)) + " old_sim_data successfully!")
132
133    enum_data = ""
134    if not os.path.exists('sim'):
135        os.mkdir('sim')
136
137    print("Creating sim file and CardDB_cardIDEnum.cs")
138    for cardid, c in cardData.items():
139        sim_data = ""
```

```

140     basic = ""
141     if 'type' in c and 'name' in c and c['type'] in TypeName:
142         # create enum data
143         type = TypeName[c['type']]
144         name_cn = c['name']['zhCN']
145         name = c['name']['enUS']
146         card_set = c['set']
147         cardtext_cn = ""
148         cardtext = ""
149         if 'cardClass' in c and 'cost' in c:
150             if type == '法术':
151                 basic = f"{ClassName[c['cardClass']] 费用: {c['cost']}"
152             if type == '随从' and 'attack' in c and 'health' in c:
153                 basic = f"{ClassName[c['cardClass']] 费用: {c['cost']} 攻击力: {c['attack']}"
154             if type == '武器' and 'attack' in c and 'durability' in c:
155                 basic = f"{ClassName[c['cardClass']] 费用: {c['cost']} 攻击力: {c['attack']} 耐久度: {c['durability']}"
156             if type == '英雄技能':
157                 basic = f"{ClassName[c['cardClass']] 费用: {c['cost']}"
158         enum_data += "/// <summary>\n"
159         enum_data += f"/// <para>{type} {basic}</para>\n"
160         enum_data += f"/// <para>{name}</para>\n"
161         enum_data += f"/// <para>{name_cn}</para>\n"
162         if 'text' in c:
163             cardtext_cn = c['text']['zhCN'].replace('\n', ' ')
164             cardtext = c['text']['enUS'].replace('\n', ' ')
165             enum_data += f"/// <para>{cardtext}</para>\n"
166             enum_data += f"/// <para>{cardtext_cn}</para>\n"
167         enum_data += "/// </summary>\n"
168         enum_data += f"{cardid} = {c['dbfId']},\n"
169         # create sim data
170         if type == '附魔':
171             continue
172         while True:
173             if cardid in Sim_id_idx:
174                 sim_data = Sim_Context[Sim_id_idx[cardid]]
175                 break
176             if 'text' in c and c['text']['zhCN'] in Sim_text_idx:
177                 idx = Sim_text_idx[c['text']['zhCN']]
178                 sim_id = Sim_CardID[idx]
179                 if cardid not in sim_id:
180                     sim_data = Sim_Context[idx]
181                     # fix
182                     sim_data = sim_data.replace(f'class Sim_{sim_id}', f'class Sim_{cardid}')
183                     break
184             sim_data = "using System;\nusing System.Collections.Generic;\nusing System.Linq;\nusing System.Text;\nusing HREngine.Bots\n{\n"
185             sim_data += "namespace HREngine.Bots\n{\n"
186             sim_data += f"\tclass Sim_{cardid} : SimTemplate /* {name_cn} {name} */\n"
187             if cardtext != "":
188                 sim_data += "\t\t/t/" + cardtext + "\n"
189                 sim_data += "\t\t/t/" + cardtext_cn + "\n"

```


搜索代码

□ C#

```
1 help.logg("cthunbonus: " + this.anzOgOwnCThunAngrBonus + " " + this.anzOgOwnCThunH
```

在下面添加一行

□ C#

```
1 help.logg("invoke: " + this.OwnInvoke + " " + this.EnemyInvoke);
```

搜索代码

□ C#

```
1 public void updateJadeGolemsInfo(int anzOwnJG, int anzEnemyJG)
2 {
3     anzOwnJadeGolem = anzOwnJG;
4     anzEnemyJadeGolem = anzEnemyJG;
5 }
```

在下面添加我们的自定义函数updateInvokedInfo

□ C#

```
1 public void updateInvokedInfo(int Owninvoke, int Ememyinvoke)
2 {
3     this.OwnInvoke = Owninvoke;
4     this.EnemyInvoke = Ememyinvoke;
5 }
```

然后修改silverfish_HB.cs文件 在

□ C#

```
1 Hrtprozis.Instance.updateJadeGolemsInfo(GameState.Get().GetFriendlySidePlayer().Ge
```

的下一行添加调用我们所写的的自定义函数updateInvokedInfo

□ C#

```
1 Hrtprozis.Instance.updateInvokedInfo(GameState.Get().GetFriendlySidePlayer().GetTa
```

最后修改Playfield.cs文件

搜索代码

□ C#

```
1 public int anzEnemyJadeGolem = 0;
```

在下面添加两行

□ C#

```
1 public int OwnInvoke = 0;
2 public int EnemyInvoke = 0;
```

搜索代码

□ C#

```
1 this.anzEnemyJadeGolem = prozis.anzEnemyJadeGolem;
```

在下面添加两行

□ C#

```
1 this.OwnInvoke = prozis.OwnInvoke;
2 this.EnemyInvoke = prozis.EnemyInvoke;
```

搜索代码

□ C#

```
1 this.anzEnemyJadeGolem = p.anzEnemyJadeGolem;
```

在下面添加两行

□ C#

```
1 this.OwnInvoke = p.OwnInvoke;
2 this.EnemyInvoke = p.EnemyInvoke;
```

搜索代码

□ C#

```
1 retval += this.anzOgOwnCThunAngrBonus + this.anzOwnJadeGolem + this.anzOwnElementa
```

修改为

□ C#

```
1 retval += this.anzOgOwnCThunAngrBonus + this.anzOwnJadeGolem + this.OwnInvoke + th
```

搜索函数

□ C#

```
1 public CardDB.Card getNextJadeGolem(bool own)
```

在这个函数上面添加getGalakrondInvoke函数

□ C#

```
1 public void getGalakrondInvoke(bool own)
2 {
3     int pos = own ? this.ownMinions.Count : this.enemyMinions.Count;
4     CardDB.Card DRG_238t14t3kid = CardDB.Instance.getCardDataFromID(CardDB.cardIDEX_238t14t3kid);
5     CardDB.Card DRG_238t12t2kid = CardDB.Instance.getCardDataFromID(CardDB.cardIDEX_238t12t2kid);
6     if (own) this.OwnInvoke++; else this.EnemyInvoke++;
```

```

7     TAG_CLASS cls = own ? this.ownHeroStartClass : this.enemyHeroStartClass;
8     switch(cls)
9     {
10        case TAG_CLASS.SHAMAN://萨满
11            this.callKid(DRG_238t14t3kid, pos, own);
12            break;
13        case TAG_CLASS.WARLOCK://术士
14            this.callKid(DRG_238t12t2kid, pos, own);
15            this.callKid(DRG_238t12t2kid, pos + 1, own);
16            break;
17        case TAG_CLASS.WARRIOR://战士
18            this.minionGetTempBuff(own ? this.ownHero : this.enemyHero, 3, 0);
19            break;
20        case TAG_CLASS.PRIEST://牧师
21            this.drawACard(CardDB.cardIDEnum.None, own, true);
22            break;
23        case TAG_CLASS.ROGUE://盗贼
24            this.drawACard(CardDB.cardIDEnum.DAL_741, own, true); //虚灵跟班
25            break;
26        default:
27            Helpfunctions.Instance.ErrorLog("[提示] 未知的英雄进行祈求");
28            break;
29    }
30    return;
31 }

```

还有调试相关修改，如果你不会调试，可以不进行修改 在**BoardTester.cs**文件

搜索代码

□ C#

```
1 int anzEnemyJadeGolem = 0;
```

在下面添加两行

□ C#

```
1 int OwnInvoke = 0;
2 int EnemyInvoke = 0;
```

搜索

□ C#

```
1 if (s.StartsWith("elementals: "))
```

在上面添加代码

□ C#

```
1 if (s.StartsWith("invoke: "))
2 {
3     String[] ss = s.Split(' ');
```



```

4     OwnInvoke = Convert.ToInt32(ss[1]);
5     EnemyInvoke = Convert.ToInt32(ss[2]);
6 }

```

搜索

□ C#

```
1 Hrtprozis.Instance.updateJadeGolemsInfo(this.anzOwnJadeGolem, this.anzEnemyJadeGol
```

在下一行添加

□ C#

```
1 Hrtprozis.Instance.updateInvokedInfo(this.OwnInvoke, this.EnemyInvoke);
```

到此修改结束，如果有卡要进行祈求则使用 `public void getGalakrondInvoke(bool own)` 函数
如有不懂的地方，或者我写的不明白的地方请评论留言。具体的文件位置可以看我另一篇文章。

炉石兄弟切换模式功能修复 3月20日

炉石发布会结束之后，兄弟的切换模式功能坏了 调试之后发现，问题主要存在于**Option**枚举被更新

解决方案 在主程序中替换新版**Option(Triton.Game.Mapping.Option)** 然后把
Triton.Bot.Logic.Bots.DefaultBot.DefaultBot中对应的**Option**的值也改过去，因为就修改枚举，调用的**Option**值不会变化。要修改的两个**Option**分别是 **IN_WILD_MODE = 199**
IN_RANKED_PLAY_MODE = 160

两种解决办法 1.直接用**DnSpy**修改IL代码（注意：直接用**DnSpy**修改代码会编译不通过） 2.
用**ILSpy**反编译之后将代码复制到**DnSpy**中（原因好像是**DnSpy**的内置**ILSpy**版本比较老）

附：新版Option(2020-03-18)

□ C#

```

1 public enum Option
2 {
3     INVALID,
4     CLIENT_OPTIONS_VERSION,
5     SOUND,
6     MUSIC,
7     CURSOR,
8     HUD,
9     STREAMING,
10    SOUND_VOLUME,
11    MUSIC_VOLUME,
12    GFX_WIDTH,
13    GFX_HEIGHT,
14    GFX_FULLSCREEN,
15    HAS_SEEN_NEW_CINEMATIC,

```

```
16    GFX_QUALITY,
17    FAKE_PACK_OPENING,
18    FAKE_PACK_COUNT,
19    HEALTHY_GAMING_DEBUG,
20    LAST_SCENE_MODE,
21    LOCALE,
22    IDLE_KICKER,
23    IDLE_KICK_TIME,
24    BACKGROUND_SOUND,
25    PREFERRED_REGION,
26    FORCE_SHOW_IKS,
27    PEGUI_DEBUG,
28    NEARBY_PLAYERS,
29    GFX_WIN_CAMERA_CLEAR,
30    GFX_MSAA,
31    GFX_FXAA,
32    GFX_TARGET_FRAME_RATE,
33    GFX_VSYNC,
34    CARD_BACK,
35    CARD_BACK2,
36    LOCAL_TUTORIAL_PROGRESS,
37    CONNECT_TO_AURORA,
38    SEASON_END_THRESHOLD,
39    RECONNECT,
40    RECONNECT_TIMEOUT,
41    RECONNECT_RETRY_TIME,
42    CHANGED_CARDS_DATA,
43    KELTHUZADTAUNTS,
44    GFX_WIN_POSX,
45    GFX_WIN_POSY,
46    PREFERRED_CDN_INDEX,
47    LAST_FAILED_DOP_VERSION,
48    TOUCH_MODE,
49    SHOWN_GFX_DEVICE_WARNING,
50    INTRO,
51    TUTORIAL_LOST_PROGRESS,
52    ERROR_SCREEN,
53    WELCOME_QUEST_VIEWS,
54    IKS_LAST_DOWNLOAD_TIME,
55    IKS_LAST_DOWNLOAD_RESPONSE,
56    IKS_CACHE_AGE,
57    CHEAT_HISTORY,
58    PRELOAD_CARD_ASSETS,
59    COLLECTION_PREMIUM_TYPE,
60    DEV_TIMESCALE,
61    IKS_LAST_SHOWN_AD,
62    SEEN_PACK_PRODUCT_LIST,
63    SHOW_STANDARD_ONLY,
64    SHOW_SET_ROTATION_INTRO_VISUALS,
65    SKIP_ALL_MULLIGANS,
```

```
66 IS_TEMPORARY_ACCOUNT_CHEAT,  
67 TEMPORARY_ACCOUNT_DATA,  
68 DISALLOWED_CLOUD_STORAGE,  
69 CREATED_ACCOUNT,  
70 LAST_HEAL_UP_EVENT_DATE,  
71 PUSH_NOTIFICATION_STATUS,  
72 DBF_XML_LOADING,  
73 HAS_SHOWN_DEVICE_PERFORMANCE_WARNING,  
74 SCREENSHOT_DIRECTORY,  
75 SIMULATE_CELLULAR,  
76 ASSET_DOWNLOAD_ENABLED,  
77 UPDATE_STATE,  
78 NATIVE_UPDATE_STATE,  
79 ASK_UNKNOWN_APPS,  
80 LAUNCH_COUNT,  
81 IS_INSTALL_REPORTED,  
82 FIRST_INSTALL_TIME,  
83 UPDATED_CLIENT_VERSION,  
84 UPDATE_STOP_LEVEL,  
85 MAX_DOWNLOAD_SPEED,  
86 STREAMING_SPEED_IN_GAME,  
87 AUTOCONVERT_VIRTUAL_CURRENCY,  
88 STREAMER_MODE,  
89 LATEST_SEEN_SHOP_PRODUCT_LIST,  
90 LATEST_DISPLAYED_SHOP_PRODUCT_LIST,  
91 RANK_DEBUG,  
92 DEBUG_CURSOR,  
93 SERVER_OPTIONS_VERSION,  
94 PAGE_MOUSE_OVERS,  
95 COVER_MOUSE_OVERS,  
96 AI_MODE,  
97 TIP_PRACTICE_PROGRESS,  
98 TIP_PLAY_PROGRESS,  
99 TIP_FORGE_PROGRESS,  
100 LAST_PRECON_HERO_CHOSEN,  
101 LAST_CUSTOM_DECK_CHOSEN,  
102 SELECTED_ADVENTURE,  
103 SELECTED_ADVENTURE_MODE,  
104 LAST_SELECTED_STORE_BOOSTER_ID,  
105 LAST_SELECTED_STORE_ADVENTURE_ID,  
106 LAST_SELECTED_STORE_HERO_ID,  
107 LATEST_SEEN_TAVERN_BRAWL_SEASON,  
108 LATEST_SEEN_TAVERN_BRAWL_SEASON_CHALKBOARD,  
109 TIMES_SEEN_TAVERN_BRAWL_CRAZY_RULES_QUOTE,  
110 SET_ROTATION_INTRO_PROGRESS,  
111 TIMES_MOUSED_OVER_SWITCH_FORMAT_BUTTON,  
112 RETURNING_PLAYER_BANNER_SEEN,  
113 LATEST_SEEN_TAVERN_BRAWL_SESSION_LIMIT,  
114 LAST_TAVERN_JOINED,  
115 LATEST_SEEN_FIRESIDE_BRAWL_SEASON,
```

116 LATEST_SEEN_FIRESIDEBRAWL_SEASON_CHALKBOARD,
117 LATEST_SEEN_SCHEDULED_DOUBLE_GOLD_VO,
118 LATEST_SEEN_SCHEDULED_ALL_POPUPS_SHOWN_VO,
119 LATEST_SEEN_SCHEDULED_ENTERED_ARENA_DRAFT,
120 LATEST_SEEN_SCHEDULED_LOGIN_FLOW_COMPLETE,
121 LATEST_SEEN_WELCOME_QUEST_DIALOG,
122 LATEST_SEEN_CURRENCY_CHANGED_VERSION,
123 LATEST_SEEN_ADVENTURE,
124 LATEST_SEEN_SCHEDULED_WELCOME_QUEST_SHOWN_VO,
125 LATEST_SEEN_SCHEDULED_GENERIC_REWARD_SHOWN_VO,
126 LATEST_SEEN_SCHEDULED_ARENA_REWARD_SHOWN_VO,
127 LAST_SELECTED_STORE_PACK_TYPE,
128 WHIZBANG_POPUP_COUNTER,
129 SET_ROTATION_INTRO_PROGRESS_NEW_PLAYER,
130 HAS_CLICKED_TOURNAMENT,
131 HAS_OPENED_BOOSTER,
132 HAS_SEEN_TOURNAMENT,
133 HAS_SEEN_COLLECTIONMANAGER,
134 JUST_FINISHED_TUTORIAL,
135 SHOW_ADVANCED_COLLECTIONMANAGER,
136 HAS_SEEN_PRACTICE_TRAY,
137 HAS_SEEN_HUB,
138 HAS_FINISHED_A_DECK,
139 HAS_SEEN_FORGE,
140 HAS_SEEN_FORGE_HERO_CHOICE,
141 HAS_SEEN_FORGE_CARD_CHOICE,
142 HAS_SEEN_FORGE_CARD_CHOICE2,
143 HAS_SEEN_FORGE_PLAY_MODE,
144 HAS_SEEN_FORGE_1WIN,
145 HAS_SEEN_FORGE_2LOSS,
146 HAS_SEEN_FORGE_RETIRE,
147 HAS_SEEN_MULLIGAN,
148 HAS_SEEN_EXPERT_AI,
149 HAS_SEEN_EXPERT_AI_UNLOCK,
150 HAS_SEEN_DECK_HELPER,
151 HAS_SEEN_PACK_OPENING,
152 HAS_SEEN_PRACTICE_MODE,
153 HAS_SEEN_CUSTOM_DECK_PICKER,
154 HAS_SEEN_ALL_BASIC_CLASS_CARDS_COMPLETE,
155 HAS_BEEN_NUDGED_TO_CM,
156 HAS_ADDED_CARDS_TO_DECK,
157 TIP_CRAFTING_UNLOCKED,
158 HAS_PLAYED_EXPERT_AI,
159 HAS_DISENCHANTED,
160 HAS_SEEN_SHOW_ALL_CARDS_REMINDER,
161 HAS_SEEN_CRAFTING_INSTRUCTION,
162 HAS_CRAFTED,
163 IN_RANKED_PLAY_MODE,
164 HAS_SEEN_THE_COIN,
165 HAS_SEEN_100g_REMINDER,

166 HAS_SEEN_GOLD_QTY_INSTRUCTION,
167 HAS_SEEN_LEVEL_3,
168 HAS_LOST_IN_ARENA,
169 HAS_RUN_OUT_OF_QUESTS,
170 HAS_ACKED_ARENA_REWARDS,
171 HAS_SEEN_STEALTH_TAUNTER,
172 FRIENDS_LIST_REQUEST_SECTION_HIDE,
173 FRIENDS_LIST_CURRENTGAME_SECTION_HIDE,
174 FRIENDS_LIST_FRIEND_SECTION_HIDE,
175 FRIENDS_LIST_NEARBYPLAYER_SECTION_HIDE,
176 FRIENDS_LIST_RECRUIT_SECTION_HIDE,
177 HAS_SEEN_HEROIC_WARNING,
178 HAS_SEEN_NAXX,
179 HAS_ENTERED_NAXX,
180 HAS_SEEN_NAXX_CLASS_CHALLENGE,
181 BUNDLE_JUST_PURCHASE_IN_HUB,
182 HAS_PLAYED_NAXX,
183 SPECTATOR_OPEN_JOIN,
184 HAS_STARTED_A_DECK,
185 HAS_SEEN_COLLECTIONMANAGER_AFTER_PRACTICE,
186 HAS_SEEN_BRM,
187 HAS_SEEN_LOE,
188 HAS_CLICKED_MANA_TAB,
189 HAS_SEEN_FORGE_MAX_WIN,
190 DEPRECATED_HAS_HEARD_TGT_PACK_VO,
191 HAS_SEEN_LOE_STAFF_DISAPPEAR,
192 HAS_SEEN_LOE_STAFF_REAPPEAR,
193 HAS_SEEN_UNLOCK_ALL_HEROES_TRANSITION,
194 SKIP_DECK_TEMPLATE_PAGE_FOR_CLASS_FLAGS,
195 HAS_SEEN_DECK_TEMPLATE_SCREEN,
196 HAS_CLICKED_DECK_TEMPLATE_REPLACE,
197 HAS_SEEN_DECK_TEMPLATE_GHOST_CARD,
198 HAS_REMOVED_CARD_FROM_DECK,
199 HAS_SEEN_DELETE_DECK_REMINDER,
200 HAS_CLICKED_COLLECTION_BUTTON_FOR_NEW_CARD,
201 HAS_CLICKED_COLLECTION_BUTTON_FOR_NEW_DECK,
202 IN_WILD_MODE,
203 HAS_SEEN_WILD_MODE_VO,
204 HAS_SEEN_STANDARD_MODE_TUTORIAL,
205 NEEDS_TO_MAKE_STANDARD_DECK,
206 HAS_SEEN_INVALID_ROTATED_CARD,
207 SHOW_SWITCH_TO_WILD_ON_PLAY_SCREEN,
208 SHOW_SWITCH_TO_WILD_ON_CREATE_DECK,
209 SHOW_WILD_DISCLAIMER_POPUP_ON_CREATE_DECK,
210 HAS_SEEN_BASIC_DECK_WARNING,
211 GLOW_COLLECTION_BUTTON_AFTER_SET_ROTATION,
212 HAS_SEEN_SET_FILTER_TUTORIAL,
213 HAS_SEEN_RAF,
214 HAS_SEEN_RAF_RECRUIT_URL,
215 HAS_SEEN_KARA,

```
216 HAS_SEEN_HEROIC_BRAWL,
217 SHOW_INNKEEPER_DECK_DIALOGUE,
218 HAS_HEARD_RETURNING_PLAYER_WELCOME_BACK_VO,
219 HAS_SEEN_RETURNING_PLAYER_INNKEEPER_CHALLENGE_INTRO,
220 DEPRECATED_HAD_ITEM_THAT_WILL_BE_ROTATED_IN_UPCOMING_EVENT,
221 SHOULD_AUTO_CHECK_IN_TO_FIRESIDE_GATHERINGS,
222 HAS_CLICKED_FIRESIDE_BRAWL_BUTTON,
223 HAS_CLICKED_FIRESIDE_GATHERINGS_BUTTON,
224 HAS_INITIATED_FIRESIDE_GATHERING_SCAN,
225 HAS_DISABLED_GOLDENS_THIS_DRAFT,
226 HAS_SEEN_FREE_ARENA_WIN_DIALOG_THIS_DRAFT,
227 HAS_SEEN_ICC,
228 LATEST_SEEN_ARENA_SEASON_STARTING,
229 LATEST_SEEN_ARENA_SEASON_ENDING,
230 HAS_SEEN_LOOT,
231 HAS_SEEN_LATEST_DUNGEON_RUN_COMPLETE,
232 HAS_SEEN_LOOT_CHARACTER_SELECT_VO,
233 HAS_SEEN_LOOT_WELCOME_BANNER_VO,
234 HAS_SEEN_LOOT_BOSS_FLIP_1_VO,
235 HAS_SEEN_LOOT_BOSS_FLIP_2_VO,
236 HAS_SEEN_LOOT_BOSS_FLIP_3_VO,
237 HAS_SEEN_LOOT_OFFER_TREASURE_1_VO,
238 HAS_SEEN_LOOT_OFFER_LOOT_PACKS_1_VO,
239 HAS_SEEN_LOOT_OFFER_LOOT_PACKS_2_VO,
240 HAS_JUST_SEEN_LOOT_NO_TAKE_CANDLE_VO,
241 HAS_SEEN_LOOT_IN_GAME_WIN_VO,
242 HAS_SEEN_LOOT_IN_GAME_LOSE_VO,
243 HAS_SEEN_LOOT_IN_GAME_MULLIGAN_1_VO,
244 HAS_SEEN_LOOT_IN_GAME_MULLIGAN_2_VO,
245 HAS_SEEN_LOOT_COMPLETE_ALL_CLASSES_VO,
246 HAS_SEEN_LOOT_BOSS_HERO_POWER,
247 HAS_SEEN_LOOT_IN_GAME_LOSE_2_VO,
248 HAS_SEEN_RANK_REVAMP_END_OF_GAME_WINS_REQUIRED,
249 HAS_SEEN_GIL_BONUS_CHALLENGE,
250 HAS_SEEN_PLAYED_TESS,
251 HAS_SEEN_PLAYED_DARIUS,
252 HAS_SEEN_PLAYED_SHAW,
253 HAS_SEEN_PLAYED_TOKI,
254 HAS_SEEN_BATTLEGROUND_BOX_BUTTON
255 }
```

深入探究炉石兄弟发现、抉择机制（第一篇：完美修复龙蹄的灾难选择） 3月25日

这是深入探究炉石兄弟发现、抉择机制的第一篇：**完美修复龙蹄的灾难选择** 先来看看龙蹄这张卡的牌面描述



那这个灾难究竟是怎么样的呢？我在网络上找了一张实战的图



可以发现这就是类似于抉择一样的选择卡牌之后使用对应的功能，关键代码如下

首先，通过调用 `makeChoice()` 函数来确定要执行的动作，然后使用一个 `switch` 语句根据返回的值选择相应的操作

- `case 0`：是选中间的一张牌
- `case 1`：是选最左边的一张牌
- `case 2`：是选最右边的一张牌

□ C#

```
1 await Coroutine.Sleep(555 + makeChoice());
2 switch (dirtychoice)
3 {
4     case 0: TritonHs.ChooseOneClickMiddle(); break;
5     case 1: TritonHs.ChooseOneClickLeft(); break;
6     case 2: TritonHs.ChooseOneClickRight(); break;
7 }
8 dirtychoice = -1;
```

再来看看 `makeChoice()` 相关的代码

首先检查 `dirtychoice` 变量的值，根据卡牌的属性（如是否为“发现”或“抉择”卡牌）设置 `choiceMode`。然后，根据当前时间和其他条件，调整 `dirtychoice` 的值以决定最终的选择。特别是，通过交换 `dirtychoice` 值0和1的逻辑，来调整选择的卡牌顺序。

□ C#

```
1 private int makeChoice()
2 {
3     if (dirtychoice < 1)
4     {
5         ...
6         if (sourceCard.GetEntity().HasReferencedTag(GAME_TAG.DISCOVER))
7         {
8             choiceMode = GAME_TAG.DISCOVER;
9             dirtychoice = -1;
10        }
11        else if (sourceCard.GetEntity().HasReferencedTag(GAME_TAG.ADAPT))
12        {
13            choiceMode = GAME_TAG.ADAPT;
14            dirtychoice = -1;
15        }
16        DateTime tmp = DateTime.Now;
17        ...
18        if (dirtychoice == 0) dirtychoice = 1;
19        else if (dirtychoice == 1) dirtychoice = 0;
20        int ttf = (int)(DateTime.Now - tmp).TotalMilliseconds;
21        if (ttf < 3000) return (new Random().Next(ttf < 1300 ? 1300 - ttf : 0, 310
22    }
```

```

23     else
24     {
25         Helpfunctions.Instance.logg("选择这张卡牌: " + dirtychoice);
26         return (new Random().Next(1100, 3200));
27     }
28 }

```

再来看看 ChooseOneClickRight() 函数

在 TritonHs.ChooseOneClickRight() 函数中，我们从右侧选择卡牌，这是通过获取所有可选卡牌的列表并选择列表末尾的卡牌来实现的，符合函数名称的直观理解。

□ C#

```

1 public static void ChooseOneClickRight()
2 {
3     List<Card> friendlyCards = ChoiceCardMgr.Get().GetFriendlyCards();
4     Client.LeftClickAt(Client.CardInteractPoint(friendlyCards[friendlyCards.Count
5 }

```

最后，对于龙蹄这样四张牌的选择，我对原有的选择逻辑进行了调整，以支持不仅仅是三张卡的选择，而是可以根据实际的卡牌数量动态选择。我的思路是直接调用点击卡牌方法，并且我决定去掉"内部函数0转成1，把1转成0，到外部调用的时反常识的顺序（012对应中左右）"这样的操作 于是我直接把内部函数的

□ C#

```

1 if (dirtychoice == 0) dirtychoice = 1;
2 else if (dirtychoice == 1) dirtychoice = 0;

```

直接删掉了，并且学着TritonHs.ChooseOneClickRight()函数来编写外部调用，来替换原来的switch语句

□ C#

```

1 List<Card> friendlyCards = ChoiceCardMgr.Get().GetFriendlyCards();
2 Client.LeftClickAt(Client.CardInteractPoint(friendlyCards[dirty]));

```

原本以为一切顺利，直到我发现兄弟使用了抉择卡牌就会报错。

问题表现为：

- 内部逻辑显示选择左侧卡牌，兄弟却执行的是选择了右侧卡牌。
- 内部逻辑显示选择右侧卡牌，兄弟就直接报错了。

经过仔细分析，我发现了问题的根源。问题出在了我们之前未曾注意到的代码分支上，这一分支负责处理抉择卡牌的延迟选择：

□ C#

```

1 if (dirtychoice < 1)
2 {
3     ...
4 }

```



```

5 else
6 {
7     Helpfunctions.Instance.logg("选择这张卡牌: " + dirtychoice);
8     return (new Random().Next(1100, 3200));
9 }

```

是的，问题就在于前面我们一直没有注意的第二个分支，他所做的事情是用于**抉择卡牌的延迟**，当抉择时，`dirtychoice` 就传入此函数时就已经不为 0，表示已处理，且抉择选项的编号不是从 0 开始的，而是从 1 开始，即 1 代表左侧卡牌，2 代表右侧卡牌。因为抉择卡牌只有两个选项，而按照原先的逻辑，返回 1 时程序错误地尝试选择第二张卡，返回 2 时则尝试选择第三张卡，从而导致错误。

为解决这个问题，我进行了一番调整。我重写了卡牌选择的逻辑，以确保程序能够正确处理抉择和发现机制：

□ C#

```

1 public void ChooseOneClick(int dirty)
2 {
3     if (dirty >= 0)
4     {
5         switch (dirtychoice)
6         {
7             case 0: TritonHs.ChooseOneClickMiddle(); break;
8             case 1: TritonHs.ChooseOneClickLeft(); break;
9             default:
10                {
11                    List<Card> friendlyCards = ChoiceCardMgr.Get().GetFriendlyCard
12                    if (friendlyCards.Count > dirty)
13                        Client.LeftClickAt(Client.CardInteractPoint(friendlyCards[
14                    else
15                        TritonHs.ChooseOneClickRight();//抉择
16                    break;
17                }
18            }
19        }
20    }

```

通过这种方法，我们能够确保当 `dirtychoice` 为 0 或 1 时，程序调用原有的函数进行选择；而对于其他值，则进行特殊处理，以区分是通过发现机制还是抉择机制进行的选择。

当然，只是修改了这个之后，还是会发现兄弟还是不会智能的选择卡牌，但是如果强制让其选择第四张牌已经不会报错了。这个只是我们修复炉石兄弟发现、抉择机制的一个小小的铺垫，实战中还需要进一步的修改。

敬请期待 第二篇：如何更完美的获取发现卡牌的来源（目前兄弟的抉择必须是由兄弟自己打出才能正常的选择，如果你在选牌的界面才打开兄弟则就会提示：没有获得卡牌数据，这对于想要追求完美的我来说不能忍！） **第三篇：**如何添加智能选牌数据库和修复类型识别问题

炉石新机制：休眠，流放 4月11日

休眠：双方玩家都无法触及。（感谢吧主） 流放：在手牌中的最左或最右侧使用时触发的额外效果。

最近有点忙，话不多说直接挂代码吧

Minion.cs中还有几处赋值需要新增，可以学习lifesteal进行增加

在**silverfish_HB.cs**的**getMinions**函数加入对**dormant**数据的读取

□ C#

```
1 m.dormant = entitiy.GetTag(GAME_TAG.DORMANT);
2 if (m.dormant > 0)
3 {
4     m.untouchable = true;
5     bool flag = false;
6     foreach (var en in entitiy.AttachedCards)
7     {
8         if (en.Id == "BT_737e")
9         {
10             m.dormant = en.GetTag(GAME_TAG.SCORE_VALUE_1) - en.GetTag(GAME_TAG.SCO
11             flag = true;
12             break;
13         }
14     }
15     if (!flag) Helpfunctions.Instance.ErrorLog("未能读取到休眠信息");
16 }
```

SimTemplate.cs中新增三个函数

□ C#

```
1 public virtual void onOutcast(Playfield p, bool own)
2 {
3     return;
4 }
5 public virtual void onOutcast(Playfield p, Minion m)
6 {
7     return;
8 }
9 public virtual void onDormantEndsTrigger(Playfield p, Minion m)
10 {
11     return;
12 }
```

这里解释一下，第一个**onOutcast**代表法术牌的流放事件 第二个**onOutcast**代表随从牌的流放事件，类似于战吼 **onDormantEndsTrigger**指代随从唤醒事件。

在**Playfield.cs**中的

□ C#

```
1 public void playACard(Handmanager.Handcard hc, Minion target, int position, int ch
```

函数下搜索

□ C#

```
1 c.sim_card.onCardPlay(this, true, target, choice);
```

在这一行之下添加

□ C#

```
1 //position从1开始, 可以自行测试
2 if (hc.position == 1 || hc.position == this.owncards.Count)
3 {
4     c.sim_card.onOutcast(this, true);
5 }
```

搜索

□ C#

```
1 public void placeAmobSomewhere(Handmanager.Handcard hc, int choice, int zonepos)
```

函数下搜索

□ C#

```
1 m.handcard.card.sim_card.getBattlecryEffect(this, m, hc.target, choice);
```

在这一行之下添加

□ C#

```
1 if (hc.position == 1 || hc.position == this.owncards.Count)
2 {
3     m.handcard.card.sim_card.onOutcast(this, m);
4 }
```

搜索

□ C#

```
1 public void triggerStartTurn(bool ownturn)
```

函数下搜索

□ C#

```
1 m.updateReadyness();
```

在这一行之下添加

□ C#

```
1 if (m.dormant > 0 && ownturn == m.own)
2 {
3     m.dormant--;
4     if (m.dormant == 0)
5     {
```

```

6         m.handcard.card.sim_card.onDormantEndsTrigger(this, m);
7     }
8 }

```

至此添加结束，接下来是我写的几张SIM Sim_BT_004.cs

□ C#

```

1 namespace HREngine.Bots
2 {
3     class Sim_BT_004 : SimTemplate /* 被禁锢的眼魔 Imprisoned Observer
4     {
5         //<b>Dormant</b> for 2 turns.When this awakens,deal 2 damage to allenemy m:
6         //<b>休眠</b>两回合。唤醒时，对所有敌方随从造成2点伤害。
7         public override void onDormantEndsTrigger(Playfield p, Minion m)
8         {
9             p.allMinionOfASideGetDamage(!m.own, 2);
10        }
11    }
12 }
13 }

```

Sim_BT_009.cs

□ C#

```

1 namespace HREngine.Bots
2 {
3     class Sim_BT_009 : SimTemplate /* 被禁锢的阳鳃鱼人 Imprisoned Sungill
4     {
5         //<b>Dormant</b> for 2 turns.When this awakens, summon two 1/1Murlocs.
6         //<b>休眠</b>两回合。唤醒时，召唤两个1/1的鱼人。
7         CardDB.Card kid = CardDB.Instance.getCardDataFromID(CardDB.cardIDEnum.BT_0
8         public override void onDormantEndsTrigger(Playfield p, Minion m)
9         {
10            p.callKid(kid, m.zonepos, m.own);
11            p.callKid(kid, m.zonepos + 1, m.own);
12        }
13    }
14 }
15 }

```

Sim_BT_121.cs

□ C#

```

1 namespace HREngine.Bots
2 {
3     class Sim_BT_121 : SimTemplate /* 被禁锢的甘尔葛 Imprisoned Gan'arg
4     {
5         //<b>Dormant</b> for 2 turns.When this awakens,equip a 3/2 Axe.
6         //<b>休眠</b>两回合。唤醒时，装备一把3/2的斧子。

```

```

7      CardDB.Card weapon = CardDB.Instance.getCardDataFromID(CardDB.cardIDEnum.C
8      public override void onDormantEndsTrigger(Playfield p, Minion m)
9      {
10         p.equipWeapon(weapon, m.own);
11     }
12
13 }
14 }

```

Sim_BT_127.cs

```

□ C#
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace HREngine.Bots
6 {
7     class Sim_BT_127 : SimTemplate /* 被禁锢的萨特 Imprisoned Satyr
8     {
9         //[x]<b>Dormant</b> for 2 turns.When this awakens, reducethe Cost of a ran
10        //[<b>休眠</b>两回合。唤醒时，使你手牌中的随机一张随从牌的法力值消耗减少（5）点。
11        public override void onDormantEndsTrigger(Playfield p, Minion m)
12        {
13            if (m.own)
14            {
15                foreach (Handmanager.Handcard hc in p.owncards)
16                {
17                    hc.manacost = Math.Max(0, hc.manacost - 5);
18                }
19            }
20        }
21    }
22 }
23 }

```

Sim_BT_211.cs

```

□ C#
1 namespace HREngine.Bots
2 {
3     class Sim_BT_211 : SimTemplate /* 被禁锢的魔喉 Imprisoned Felmaw
4     {
5         //[x]<b>Dormant</b> for 2 turns.When this awakens, attack a random enemy.
6         //[<b>休眠</b>两回合。唤醒时，随机攻击一个敌人。
7         public override void onDormantEndsTrigger(Playfield p, Minion m)
8         {
9             Minion sm = p.searchRandomMinion(m.own ? p.enemyMinions : p.ownMinions
10            if (sm == null) sm = (m.own) ? p.enemyHero : p.ownHero;

```

```

11         p.minionAttacksMinion(m, sm);
12     }
13
14 }
15 }

```

Sim_BT_305.cs

```

□ C#
1 namespace HREngine.Bots
2 {
3     class Sim_BT_305 : SimTemplate /* 被禁锢的拾荒小鬼 Imprisoned Scrap Imp
4     {
5         //<b>Dormant</b> for 2 turns.When this awakens,give all minions in your hand
6         //<b>休眠</b>两回合。唤醒时，使你手牌中的所有随从牌获得+2/+2。
7         public override void onDormantEndsTrigger(Playfield p, Minion m)
8         {
9             if (m.own)
10             {
11                 foreach (Handmanager.Handcard hc in p.owncards)
12                 {
13                     hc.addattack += 2;
14                     hc.addHp += 2;
15                 }
16             }
17         }
18     }
19 }
20 }

```

Sim_BT_934.cs

```

□ C#
1 namespace HREngine.Bots
2 {
3     class Sim_BT_934 : SimTemplate /* 被禁锢的安塔恩 Imprisoned Antaen
4     {
5         //[x]<b>Dormant</b> for 2 turns.When this awakens, deal10 damage randomly
6         //<b>休眠</b>两回合。唤醒时，造成10点伤害，随机分配到所有敌人身上。
7         public override void onDormantEndsTrigger(Playfield p, Minion m)
8         {
9             p.allCharsOfASideGetRandomDamage(!m.own, 10);
10        }
11    }
12 }

```

2020-05-16 记录对炉石兄弟的修复

今天刚到家，前几天就看到群里面闹得沸沸扬扬，大概是因为炉石使用mono2.0后而且导致了兄弟大面积不可用，本来以为是比较麻烦的问题，结果没想到二十几分钟就解决了，但是由于现在局势紧张，也确实不方便透露具体内容，所以我只能从技术的角度大致的说一下解决办法

首先是由于更新Mono.dll导致的偏移更新。 mono.dll -> mono-2.0-bdwgc.dll 还有相应的偏移需要更新，但是这里发现有个偏移地址是找不着的，但是经过研究似乎兄弟也没用到这个地址。

只修复上面那个还会有问题，经过调试可以定位到报错的位置，利用CE里面的结构体分析，可以快速的得到新的结构体偏移大小。修改后，测试可用。

最后是自己的一点感想 不希望看到留言邮箱让我发一份文件

1. 我希望对炉石兄弟更多的是一种技术上的研究而并非是当做一个工具，当然这个只是对于在我博客上学习的人来说的，对于大部分来说那肯定还是一个挂机刷金币的工具，就我自己而言研究炉石兄弟还是给自己带来蛮多的乐趣的，希望大家也能体会到这种乐趣。
2. 不要对大神产生依赖性，就像是大家期待的吧主修复，或者说贴吧里哪一位大神来修复，我认为等别人修复使用这肯定不是一个长久之计，除非你愿意掏钱到淘宝上买服务，只要是白嫖，那肯定要做好哪一天不能用的准备。与其等别人修复，还不如尝试着自己动手，我这里给出一个时间数据，我用19年10月左右的那个版本，从头到尾修复可用，差不多花费了三天的时间，这个时间比我想象中的要短了很多，我给出这个数据的目的是为了告诉大家，其实找到一个老版本修复一下并不是什么难事，起码没有想象中的那么难，现在贴吧里，各种博客里也充满了兄弟的资料，现在研究比起早期研究兄弟的人简单太多了，只要你肯花心思花时间去学习。由于一些个人原因，接下来的两三个月里，我对兄弟的研究肯定要停止一段时间了。
3. 这个博客的不是为了炉石兄弟而开设的，是作为一个个人博客的形象来呈现的，之所以现在的内容都关于炉石兄弟，因为这确实是我前段时间花了很大心思研究的东西，而之后我会分享自己其他方面的编程内容，炉石兄弟作为一个让自己入门C#的工具，其实到这里已经完成了我当时接触它时的使命了。

分享几个自用的用于整理卡牌数据的Python脚本 6月13日

修复换行问题，详情请见评论区

| 生成卡牌相关文件

□ C#

```
1 import html
2 import os
3 import shutil
4 #输入: C:\hs\sim\ 旧版的Sim文件
5 #输入: H:\CardDefs.xml 卡牌数据文件
6
7 #输出: C:\hs\CardDB_getSimCard.txt 生成的getSimCard文件
8 #输出: C:\hs\CardDB_cardIDEnum.txt 生成的cardIDEnum文件
9 #输出: C:\hs\newsim\ 新版的Sim文件, 保留旧版的Sim
```

```
10 TAG_CARDTYPE = {
11     0: "INVALID",
12     1: "GAME",
13     2: "PLAYER",
14     3: "英雄",
15     4: "随从",
16     5: "法术",
17     6: "附魔",
18     7: "武器",
19     8: "ITEM",
20     9: "TOKEN",
21     10: "英雄技能",
22     11: "BLANK",
23     12: "GAME_MODE_BUTTON",
24     22: "MOVE_MINION_HOVER_TARGET"
25 }
26
27 TAG_CLASS = {
28     0: "INVALID",
29     1: "巫妖王",
30     2: "德鲁伊",
31     3: "猎人",
32     4: "法师",
33     5: "圣骑士",
34     6: "潜行者",
35     7: "牧师",
36     8: "萨满祭司",
37     9: "术士",
38     10: "战士",
39     11: "梦境",
40     12: "中立",
41     13: "威兹班",
42     14: "恶魔猎手"
43 }
44 TAG_CARD_SET = {
45     0: "INVALID",
46     1: "TEST_TEMPORARY",
47     2: "0001CORE基本",
48     3: "0002EX经典",
49     4: "0003HOF荣誉室",
50     5: "MISSIONS新手训练",
51     6: "DEMO",
52     7: "NONE",
53     8: "CHEAT",
54     9: "BLANK",
55     10: "DEBUG_SP",
56     11: "PROMO",
57     12: "0012NAX纳克萨玛斯",
58     13: "0013GVG地精大战侏儒",
59     14: "0014BRM黑石山的火焰",
```



```
60     15:"0015AT冠军的试炼",
61     16:"CREDITS暴雪制作人员",
62     17:"0017英雄皮肤和技能",
63     18:"0018TB乱斗模式",
64     19:"SLUSH",
65     20:"0020LOE探险者协会",
66     21:"0021OG上古之神的低语",
67     22:"OG_RESERVE",
68     23:"0023KAR卡拉赞之夜",
69     24:"KARA_RESERVE",
70     25:"0025CFM龙争虎斗加基森",
71     26:"GANGS_RESERVE",
72     27:"0027UNG勇闯安戈洛",
73     1001:"1001ICC冰封王座的骑士",
74     1004:"1004LOOT狗头人与地下世界",
75     1125:"1125GIL女巫森林",
76     1127:"1127BOT砰砰计划",
77     1129:"1129TRL拉斯塔哈的大乱斗",
78     1130:"1130DAL暗影崛起",
79     1158:"1158ULD奥丹姆奇兵" ,
80     1347:"1347DRG巨龙降临",
81     1439:"1439WE狂野限时回归" ,
82     1403:"1403YOD迦拉克隆的觉醒",
83     1453:"BATTLEGROUND" ,
84     1414:"1414BT外域的灰烬" ,
85     1463:"1463BT恶魔猎手新兵",
86 }
87
88 fs = open(r"C:\hs\CardDB_getSimCard.txt", 'w', encoding='utf-8')
89 fb = open(r"C:\hs\CardDB_cardIDEnum.txt", 'w', encoding='utf-8')
90 cardname = ""
91 cardnamecn = ""
92 cardtext = ""
93 cardtextcn = ""
94 cardID = ""
95 enumID = 0
96 numberID = 0
97 CARDTYPE = 0
98 CLASS = 0
99 CARD_SET = 0
100 COST = 0
101 ATK = 0
102 HEALTH = 0
103 DURABILITY = 0
104 with open(r"H:\CardDefs.xml", 'r', encoding='utf-8') as f:
105     while 1:
106         line = f.readline()
107         if not line:
108             break
109         if r"</Entity>" in line:
```

```

110      #BATTLEGROUND TB
111      if CARD_SET != 1453 and CARD_SET != 1143:
112          fb.write("/// <summary>\n")
113          other = ""
114          print(cardID)
115          if TAG_CARDTYPE[CARDTYPE] == "法术" :
116              other += " " + TAG_CLASS[CLASS] + " 费用: " + str(COST)
117          if TAG_CARDTYPE[CARDTYPE] == "随从" :
118              other += " " + TAG_CLASS[CLASS] + " 费用: " + str(COST) + " 攻
119          if TAG_CARDTYPE[CARDTYPE] == "武器":
120              other += " " + TAG_CLASS[CLASS] + " 费用: " + str(COST) + " 攻
121          if TAG_CARDTYPE[CARDTYPE] == "英雄技能":
122              other += " " + TAG_CLASS[CLASS] + " 费用: "
123          fb.write("/// <para>" + TAG_CARDTYPE[CARDTYPE] + other + "</para>\n")
124          fb.write("/// <para>" + cardname + "</para>\n")
125          fb.write("/// <para>" + cardnamecn + "</para>\n")
126          fb.write("/// <para>" + cardtext + "</para>\n")
127          fb.write("/// <para>" + cardtextcn + "</para>\n")
128          fb.write("/// </summary>\n")
129          fb.write(cardID + " = " + str(numberID) + ",\n")
130          if TAG_CARDTYPE[CARDTYPE] != "附魔":
131              fs.write("case cardIDEnum." + cardID + ": return new Sim_" + (
132                  directory = "C:\\hs\\newsim\\" + TAG_CARD_SET[CARD_SET]
133                  if not os.path.exists(directory):
134                      os.mkdir(directory)
135                  if os.path.exists("C:\\hs\\sim\\Sim_" + cardID + ".cs"):
136                      shutil.move("C:\\hs\\sim\\Sim_" + cardID + ".cs", director
137                  else:
138                      with open(directory + "\\Sim_" + cardID + ".cs", 'w', encod
139                          sim.write("using System;\nusing System.Collections.Ger
140                          sim.write("namespace HREngine.Bots\n{\n");
141                          sim.write("\tclass Sim_" + cardID + " : SimTemplate //>
142                          sim.write("\t\t/" + cardtext + "\n")
143                          sim.write("\t\t/" + cardtextcn + "\n")
144                          sim.write("\t\t\n\t\t\n\t}\n}\n")
145
146          #print(cardname, cardnamecn, cardID, TAG_CARDTYPE[CARDTYPE], TAG_C
147          cardname = ""
148          cardnamecn = ""
149          cardtext = ""
150          cardtextcn = ""
151          cardID = ""
152          enumID = 0
153          numberID = 0
154          CARDTYPE = 0
155          CLASS = 0
156          CARD_SET = 0
157          COST = 0
158          ATK = 0
159          HEALTH = 0

```

```
160         DURABILITY = 0
161
162     if "<Entity CardID=\"" in line:
163         index1 = line.find("<Entity CardID=\"")
164         index2 = line.find("\"", index1 + 16)
165         if index1 == -1 or index2 == -1:
166             print(line)
167             exit(0)
168         cardID = line[index1 + 16: index2]
169         index1 = line.find("ID=\"", index2)
170         index2 = line.find("\"", index1 + 4)
171         if index1 == -1 or index2 == -1:
172             print(line)
173             exit(0)
174         numberID = line[index1 + 4: index2]
175
176     if "<Tag enumID=\"" in line:
177         enumIDl = line.find("<Tag enumID=\"")
178         enumIDr = line.find("\"", enumIDl + 13)
179         if enumIDl == -1 or enumIDr == -1:
180             print(line)
181             exit(0)
182         enumID = int(line[enumIDl + 13: enumIDr])
183         value1 = line.find("value=\"")
184         valuer = line.find("\"", value1 + 7)
185         if value1 == -1 or valuer == -1:
186             continue
187         value = int(line[value1 + 7: valuer])
188         #CARDTYPE
189         if enumID == 202:
190             CARDTYPE = value
191         #CLASS
192         if enumID == 199:
193             CLASS = value
194         #CARD_SET
195         if enumID == 183:
196             CARD_SET = value
197         #COST
198         if enumID == 48:
199             COST = value
200         #HEALTH
201         if enumID == 45:
202             HEALTH = value
203         #ATK
204         if enumID == 47:
205             ATK = value
206         #DURABILITY
207         if enumID == 187:
208             DURABILITY = value
209     if "<enUS>" in line:
```

```
210         if enumID != 185 and enumID != 184:
211             continue
212         index1 = line.find("<enUS>")
213         index2 = line.find("</enUS>", index1 + 6)
214         text = ""
215         while (index2 == -1):
216             text += line[index1 + 6:-1]
217             line = f.readline()
218             index2 = line.find("</enUS>")
219             index1 = -6
220         text += line[index1 + 6: index2]
221
222         if index1 == -1 or index2 == -1:
223             print("错误: " + line)
224             exit(0)
225         if enumID == 185:
226             cardname = html.unescape(text)
227         if enumID == 184:
228             cardtext = html.unescape(text)
229
230     if "<zhCN>" in line:
231         if enumID != 185 and enumID != 184:
232             continue
233         index1 = line.find("<zhCN>")
234         index2 = line.find("</zhCN>", index1 + 6)
235
236         text = ""
237         while (index2 == -1):
238             text += line[index1 + 6:-1]
239             line = f.readline()
240             index2 = line.find("</zhCN>")
241             index1 = -6
242         text += line[index1 + 6: index2]
243
244         if index1 == -1 or index2 == -1:
245             print("错误: " + line)
246             exit(0)
247         if enumID == 185:
248             cardnamecn = html.unescape(text)
249         if enumID == 184:
250             cardtextcn = html.unescape(text)
251     [/collapse]
252
253     [collapse title="转移PlayRequirement到新的CardDefs文件"]
254     #输入: H: \CardDefsold.xml
255     #输出: H: \CardDefs.xml
256
257     import html
258     playreq = {}
259     playname = {}
```

```
260 with open(r"H:\CardDefsold.xml", 'r', encoding='utf-8') as f:
261     for line in f:
262         if r"</Entity>" in line:
263             if cardID in playreq:
264                 playname[cardnamecn] = playreq[cardID]
265                 cardID = ""
266                 enumID = 0
267                 cardnamecn = ""
268             if "<Entity CardID=\"\" in line:
269                 index1 = line.find("<Entity CardID=\"")
270                 index2 = line.find("\"", index1 + 16)
271                 if index1 == -1 or index2 == -1:
272                     print(line)
273                     exit(0)
274                 cardID = line[index1 + 16: index2]
275
276             if "<PlayRequirement" in line:
277                 if cardID not in playreq:
278                     playreq[cardID] = ""
279                 playreq[cardID] += line
280
281             if "<Tag enumID=\"\" in line:
282                 enumIDl = line.find("<Tag enumID=\"")
283                 enumIDr = line.find("\"", enumIDl + 13)
284                 if enumIDl == -1 or enumIDr == -1:
285                     print(line)
286                     exit(0)
287                 enumID = int(line[enumIDl + 13: enumIDr])
288             if "<zhCN>" in line:
289                 if enumID != 185 and enumID != 184:
290                     continue
291                 index1 = line.find("<zhCN>")
292                 index2 = line.find("</zhCN>", index1 + 6)
293                 text = ""
294                 while (index2 == -1):
295                     text += line[index1 + 6:-1]
296                     line = f.readline()
297                     index2 = line.find("</zhCN>")
298                     index1 = -6
299                 text += line[index1 + 6: index2]
300                 if index1 == -1 or index2 == -1:
301                     print("错误: " + line)
302                     exit(0)
303                 if enumID == 185:
304                     cardnamecn = html.unescape(text)
305
306             #for key in playreq:
307                 #print(key)
308                 #print(playreq[key])
309 with open(r"H:\CardDefscopy.xml", 'w', encoding='utf-8') as fc:
```

```
310     with open(r"H:\CardDefs.xml", 'r', encoding='utf-8') as f:
311         for line in f:
312             if r"</Entity>" in line:
313                 if cardID in playreq:
314                     fc.write(playreq[cardID])
315                 else:
316                     if cardnamecn in playname and CARDTYPE != 6 and CARDTYPE != 3:
317                         print(cardID, cardnamecn)
318                         print(playname[cardnamecn])
319                         fc.write(playname[cardnamecn])
320                     cardID = ""
321             if "<Entity CardID=\"" in line:
322                 index1 = line.find("<Entity CardID=\"")
323                 index2 = line.find("\"", index1 + 16)
324                 if index1 == -1 or index2 == -1:
325                     print(line)
326                     exit(0)
327                 cardID = line[index1 + 16: index2]
328             fc.write(line)
329             if "<Tag enumID=\"" in line:
330                 enumIDl = line.find("<Tag enumID=\"")
331                 enumIDr = line.find("\"", enumIDl + 13)
332                 if enumIDl == -1 or enumIDr == -1:
333                     print(line)
334                     exit(0)
335                 enumID = int(line[enumIDl + 13: enumIDr])
336                 valuel = line.find("value=\"")
337                 valuer = line.find("\"", valuel + 7)
338                 if valuel == -1 or valuer == -1:
339                     continue
340                 value = int(line[valuel + 7: valuer])
341                 # CARDTYPE
342                 if enumID == 202:
343                     CARDTYPE = value
344             if "<zhCN>" in line:
345                 if enumID != 185 and enumID != 184:
346                     continue
347                 index1 = line.find("<zhCN>")
348                 index2 = line.find("</zhCN>", index1 + 6)
349                 text = ""
350                 while (index2 == -1):
351                     text += line[index1 + 6:-1]
352                     line = f.readline()
353                     fc.write(line)
354                     index2 = line.find("</zhCN>")
355                     index1 = -6
356                 text += line[index1 + 6: index2]
357                 if index1 == -1 or index2 == -1:
358                     print("错误: " + line)
359                     exit(0)
```

```
360         if enumID == 185:
361             cardnamecn = html.unescape(text)
362     [/collapse]
363
364     [collapse title="从指定目录转移有内容的Sim卡到指定位置(和1搭配使用)"]
365     import os
366     import shutil
367
368     def readFilename(file_dir):
369         for root, dirs, files in os.walk(file_dir):
370             return files, dirs, root
371
372
373
374     def findstring(pathfile):
375         fp = open(pathfile, "r", encoding='UTF-8') # 注意这里的打开文件编码方式
376         strr = fp.read()
377         # print strr.find("DoubleVec")
378         if (strr.find("public") != -1):
379             print('here?')
380             return True
381         return False
382
383
384
385     def startfind(files, dirs, root):
386         for ii in files:
387             # print(ii)
388             # if ii.endswith('.lua'):
389             try:
390                 if (findstring(root + "\\" + ii)):
391                     shutil.copy(root + "\\" + ii, "C:\\hs\\sim\\" + ii)
392                     print(ii)
393             except Exception as err:
394                 print(err)
395                 continue
396
397         for jj in dirs:
398             fi, di, ro = readFilename(root + "\\" + jj)
399             startfind(fi, di, ro)
400
401
402
403     if __name__ == '__main__':
404         default_dir = u"H:\\Hearthbuddy\\Routines\\DefaultRoutine\\Silverfish\\cards"
405         file_path = default_dir # th.expanduser(default_dir))
406         files, dirs, root = readFilename(file_path)
407         startfind(files, dirs, root)
408     [/collapse]
409
```

```
410 [collapse title="读取卡牌代码，输出策略中的卡牌特征函数"]
411 import base64
412 import win32clipboard as w
413 import win32con
414
415 def setText(aString):#写入剪切板
416     w.OpenClipboard()
417     w.EmptyClipboard()
418     w.SetClipboardData(win32con.CF_UNICODETEXT, aString)
419     w.CloseClipboard()
420
421 def read_varint(data):
422     shift = 0
423     result = 0
424     while True:
425         temp = data.pop(0)
426         result |= (temp & 0x7f) << shift
427         shift += 7
428         if (temp & 0x80) == 0:
429             break
430     return result
431
432 CardDB = {}
433 with open(r"H:\CardDefs.xml", 'r', encoding='utf-8') as f:
434     while True:
435         line = f.readline()
436         if not line:
437             break
438         if r"</Entity>" in line:
439             CardDB[numberID] = cardID
440             cardID = ""
441             numberID = 0
442
443         if "<Entity CardID=\"" in line:
444             index1 = line.find("<Entity CardID=\"")
445             index2 = line.find("\"", index1 + 16)
446             if index1 == -1 or index2 == -1:
447                 print(line)
448                 exit(0)
449             cardID = line[index1 + 16: index2]
450             index1 = line.find("ID=\"", index2)
451             index2 = line.find("\"", index1 + 4)
452             if index1 == -1 or index2 == -1:
453                 print(line)
454                 exit(0)
455             numberID = int(line[index1 + 4: index2])
456
457     while True:
458         try:
459             code = input()
```



```
460         codelist = list(base64.b64decode(code))
461     except:
462         continue
463     if len(codelist) == 0:
464         continue
465     reserve = read_varint(codelist)
466     if reserve != 0:
467         continue
468     version = read_varint(codelist)
469     if version != 1:
470         continue
471     format = read_varint(codelist)
472     heroes = []
473     num_heroes = read_varint(codelist)
474     for i in range(num_heroes):
475         card_id = read_varint(codelist)
476         heroes.append(CardDB[card_id])
477     cards = []
478     num_cards_x1 = read_varint(codelist)
479     for i in range(num_cards_x1):
480         card_id = read_varint(codelist)
481         cards.append([CardDB[card_id], 1])
482
483     num_cards_x2 = read_varint(codelist)
484     for i in range(num_cards_x2):
485         card_id = read_varint(codelist)
486         cards.append([CardDB[card_id], 2])
487
488     num_cards_xn = read_varint(codelist)
489     for i in range(num_cards_xn):
490         card_id = read_varint(codelist)
491         count = read_varint(codelist)
492         cards.append([CardDB[card_id], count])
493     str = "public List<CardDB.cardIDEnum> ArchetypeCardSet = new List<CardDB.cardIDEnum>{"
494     cardslen = len(cards)
495     for i in range(cardslen):
496         str += "\tCardDB.cardIDEnum." + cards[i][0]
497         if i != cardslen:
498             str += ",\n"
499     str += "};"
500     print(str)
501     setText(str)
```

对于炉石兄弟策略编写的一些感悟 7月30日

| 1. 关于惩罚

我看到很多策略的编写者，包括我刚入门的时候，喜欢用惩罚来调整兄弟出牌，因为当时我认为兄弟的AI算法就是一个简单的BFS搜索 + 剪枝，只会考虑当前场面最优解。但现在发现其实

兄弟并没有这么蠢，他是考虑了本回合 + 下一回合的操作(只要设置合理)，这样前瞻的思路其实完全够用。所以我认为惩罚的作用应该是用于调整大于两回合，兄弟预料不到的前瞻操作。比如：

1. 休眠随从的一些特效。
2. 一些卡牌为打出一些**Combo**而留牌。
3. 为了赚取更大场面优势而留牌(比如图腾潮涌，有图腾时直接打出是对场面有益的，但是为了更好的场面所以要通过惩罚留牌)。那么惩罚该怎么写呢？下面是我的一些感想，希望能给大家提供一定的帮助。之前看到弄的风风火火的破冰和火焰结界，但现在都没有看到一种合理的解决方法，这是为什么呢？我思考之后发现，很多人用惩罚去调整，反而效果适得其反。原因就是惩罚像是教小孩子的时候你对他施暴，而策略就是合理的教学，教给他怎么样才是对的。比如：火焰结界现在流行的版本添加的位置都是错误的，都是和猎人奥秘爆炸陷阱放在同一个地方，这个就属于模拟特效的问题，火焰结界是随从攻击后触发，而爆炸陷阱是攻击前触发，这两个放在一起，问题就出现了，兄弟认为用任意一个随从去攻击效果是一样的，反正都是无法造成伤害，而不是我们一般的思路，用攻击最高的随从去攻击。还有就是通过惩罚的一些弊端，比如你写惩罚，低血量的时候使用某个有治疗效果的卡牌，返回的是负数的惩罚值。其实返回负数的惩罚值就违背了惩罚的概念。当时这样写的时候只考虑到了低血量的时候使用可以快速的回血，符合我们平常打牌的思维，但是你没有想到的是，兄弟会想办法去构造这种低血量来为了得到这个负数的惩罚（比如火球术打脸），当然这种惩罚如果数额写小一点那还无所谓，因为会被策略中所调整，但是如果数额较大，比如30、50这种程度，那就非常不好了。所以我推荐惩罚值的编写最好是反常的思路，比如你正常思路中是低血量使用回血牌，而在惩罚值的编写中应当是高血量不使用回血牌，给一个正数的惩罚值，让兄弟知道这样是错误的，我认为这样才是正常的惩罚使用方法。当然这也还是有一些问题，我还是更加推荐使用策略中血量低于一定的血线则给出一定低的评分更好一些。

| 2. 关于策略

这里的策略指的是**behavior**文件夹中的策略文件，这些策略大多用来给场面一个合理的评分。那么有些兄弟就会说了，我的兄弟打出伏笔操作了，又不能通过惩罚来调整，那该怎么办呢？我这里有两种方法：

1. 检查SIM是否编写正确，是否按照真实情况去模拟了
2. 调整策略中的对战场的评分，比如要加强留牌，那么就给手牌数量的权重调大，如果要加强场面，那就给单个随从的价值调大。确定SIM的正确性是非常重要的，包括早期兄弟官方写的卡牌也是错误频出，问题就在于没有测试，我认为写一张卡牌首先要保障其语法的正确性，再要保证其模拟的正确性，这才是一张合理正确的卡牌。而出牌速度的问题也是因为模拟错误所导致的，正是因为模拟错误从而导致兄弟认为场面和模拟的不一致，而重新计算。而这个场面模拟是很严格的，就连随从的位置顺序都很重要，比如亡语：召唤一个XXX随从，这样的卡牌的召唤位置一定要是**m.zonepos - 1**，才是正确的模拟，包括利用**p.ownMinion.Count**的都是不正确的，甚至有些随从战吼召唤是在随从的一左一右的，这种位置关系也一定要考虑。否则重新计算的时间是非常难熬的。但是有些随机性的问题是难以模拟的，比如抽一张牌，按一下图腾召唤技能，这样的操作下一步是完全未知的。这种牌是兄弟非常难把控的，原因就在于兄弟只能固定一种随机操作而围绕这种随

机的操作的固定结果而计算出的定性结果，这样持续下去兄弟把这一回合的操作都计算完了，结果开始一操作就发现操作结果与模拟的不符合，然后又去重新计算。所以为什么兄弟不适合打弃牌术，我认为大概也是这个原因吧，这个问题来自于兄弟本身的算法架构，也很难修改。所以，盲目的追求操作速度其实意义没那么大，操作并没有那么长时间，真正长的时间应该是在大场面的计算上，而计算的问题正如我上面所说，应该要尽量写对结果可预料的卡牌编写正常的SIM，而对于具有随机性的卡牌也应该要考虑多种情况分类讨论。最简单的就是对斩杀的判定。

| 3. 关于 `_settings.txt`

其中的**maxwide**我想说明一下，这个参数表明的是计算的深度，也就是数值越大，计算的越深入，我认为如果随机性强的卡组，比如弃牌术，应该把这个数字调低一些，因为基于错误的随机模拟上的大量计算是没有意义的。而如果是随机性弱的卡组，比如奥秘法等，这个数字可以适当的调高。这个原因呢，仔细品味我这篇文章，就自然知道了。如果有疑问的可以在评论区提出，欢迎大家指出我的错误，一起讨论学习！

更新于 2024-02-11

CC BY-NC-SA 4.0