

令和5年度 卒業研究

カードゲームの多人数戦において
プレイレベルを変更して対戦できる
AI の研究

Research on AI that allows you to change the play
level and compete in multiplayer card games

函館工業高等専門学校

生産システム工学科情報コース

中野 夢絃

指導教員 東海林 智也

目次

第 1 章 序論

1. 1 Abstract

1. 2 研究背景・目的

第 2 章 多人数戦の現状

2. 1 多人数戦とは

2. 2 プレイレベルとは

2. 3 多人数戦の課題

第 3 章 ゲームの内容

3. 1 ルールについて

3. 2 プログラムについて

第 4 章 AI

4. 1 想定するシステムについて

4. 2 考察

第 5 章 今後の展望

第 6 章 謝辞

第 7 章 参考文献

第 1 章 序論

1.1 Abstract

Among 1v1 trading card games, there are many that develop the rules and develop special formats for multiplayer matches with 3 to 4 players in an elimination format. In multiplayer battles in such card games, there is a problem that there is a difference in the level of play (how to enjoy the game) among the players, resulting in a game experience that does not satisfy everyone. The purpose of this research is to investigate to what extent the concept of play level can be expressed using AI and algorithms.

1.2 研究背景・目的

1vs1 のトレーディングカードゲームの中には、そのルールを発展させて、3～4 人での負け抜け形式の多人数戦を特殊フォーマットとして展開させているものも少なくない。そうしたカードゲームの多人数戦において、プレイヤー間でのプレイレベル（楽しみ方）にずれが生じ、全員が満足するようなゲーム体験にならない、という問題が挙げられている。

プレイレベルの概念が AI やアルゴリズムでどこまで表現できるのかを調査することが本研究の目的である。

第 2 章 多人数戦の現状

この章では多人数戦とプレイレベルについて詳しく説明を行う。

2.1 多人数戦とは

「多人数戦」とは、通常 1vs1 で行われるカードゲームを、多人数（多くの場合 1vs1vs1vs1）でのサバイバル形式にしたものである。多人数戦ならではの観点から競技性が高くなりづらく、経験の浅いプレイヤーでもカジュアルな楽しみ方ができるフォーマットとして注目されている。

多人数戦では 1vs1 の対戦にはない特色がある。例えば 1vs1 では、自分が強い動きをとって相手の行動を制限することが、基本的にはそのまま自分の勝ちに直結する。しかし多人数戦では、強すぎる行動をとることで、残り 3 人のプレイヤーから「この人をゲームに残しておいては勝てないな」と思われてしまい、一人狙いされてしまいそのまま負けてしまう、と言ったことも多い。このような事態を避けるために程よい強さの動きを目指すことは「ヘイトコントロール」と呼ばれ、交渉や協力などの指標にもなる。

2.2 プレイレベルとは

「プレイレベル」とは、多くの多人数戦で掲げられている楽しみ方の指標である。

多人数戦のプレイヤーの中には、競技として楽しみたい人や、カジュアルに楽しみたい人などが混在している。そのような楽しみ方の異なるプレイヤーが何も知らずに対戦してしまうと、双方がやりたいゲームができずつまらないゲーム体験になってしまう。

そのようなことを防ぐため、プレイレベルは、対戦相手と勝負を通して自分がどんな楽しみ方をしたいかを表す指標が用いられている。プレイレベルの例として、例えば表1に示すように、勝利よりもその場を楽しむことに重きを置いたレベル1から、勝利を1番に迫及するレベル3まで分類しているゲームがある[1]。各レベルは、ヘイトコントロールの行い方や勝ちに向かうための過程などで差別化される。

表1 プレイレベルの例

レベル1 みんなで楽しむことを目標に据える。
レベル2 楽しみつつ、勝利も目指す。
レベル3 勝利のための駆け引きや強力なプレイを楽しむ。

2.3 多人数戦の課題

現状の多人数戦ではプレイヤー間、プレイレベルの認識のずれが問題となっている。そもそもプレイングに確実な正解が見出しにくいカードゲームにおいて、プレイの指標というものは曖昧な部分が多く、ゲーム前にすり合わせを行ったとしても、認識を合わせられないケースも多い。そこで本研究では、AI のプレイレベルの調整のアプローチを調査し、実際の間人同士のゲームでの解決方法を探る。

第 3 章 ゲームの内容

本研究で開発するカードゲームは、既存のカードゲーム多人数戦を AI に理解しやすいように簡略化させたものを想定している。この章ではゲームのルールとプログラムについて解説する。

3. 1 ルールについて

ゲームは人間が 1 人、AI が 3 人の計 4 人でのサバイバル形式とし、それぞれに与えられた持ち点が 0 になったプレイヤーから脱落していく。相手の持ち点を減らすカードを使用するためにはプレイポイントを必要とする。プレイポイントは初期値 1 で、自分のターンを迎えるたびに最大値を 1 増やし、上限までプレイポイントを回復する。後半になるほどできることが増えるので、序盤から終盤まで同じような状況になりにくいルールと考案採用した。

プレイヤーは手札をドロー、プレイする。ターンを終了するたびにプレイポイントを 1 増やし、カードを 1 枚引く。カードをプレイすると手札から消える。図 1 にプレイ画面を示す。

```
あなたの手札: ['5軸バニラ', '6軸バニラ', '9軸バニラ', '5軸バニラ', '8軸バニラ',  
'1軸バニラ', '7軸バニラ', '10軸バニラ', '6軸バニラ', '商人']  
あなたの PP: 6  
あなたの MP: 0  
アクションを選択してください (1: カードをプレイ, 2: ターンを終了, 0: ゲームをやめる): █
```

図 1 プレイ画面

3. 2 プログラムについて

本プログラムではデッキ構築は考慮せず、全員同じデッキを使用する。カードは図2のように連想配列を用いて管理している。

```
def score_hands(self, player_hands):
    score_hands = []
    for card_idx, cards_score in enumerate(player_hands):
        # print(card_idx, cards_score)
        # 連想配列に通知
        cards_score_set = str(cards_score).split("-")
        card_mark = cards_score_set[0]
        card_rank = cards_score_set[1]

        # コストを算出
        card_cos = 0
        if card_rank == "1輪バニョ":
            card_cos = 1
        elif card_rank == "2輪バニョ":
            card_cos = 2
        elif card_rank == "3輪バニョ":
            card_cos = 3
        elif card_rank == "4輪バニョ":
            card_cos = 4
        elif card_rank == "5輪バニョ":
            card_cos = 5
        elif card_rank == "6輪バニョ":
            card_cos = 6
        elif card_rank == "7輪バニョ":
            card_cos = 7
        elif card_rank == "8輪バニョ":
            card_cos = 8
        elif card_rank == "9輪バニョ":
            card_cos = 9
        elif card_rank == "10輪バニョ":
            card_cos = 10
        elif card_rank == "ワイニョ":
            card_cos = 2
        elif card_rank == "炎":
            card_cos = 1
        elif card_rank == "氷":
            card_cos = 2

        score_hands[card_idx] += card_cos
        score_hands.append(card_cos)
    print(score_hands)
```

図2 デッキプログラム

第 4 章 AI

AI については未実装なため、システムとして想定していたものと、類似例や自身のゲーム体験から考えた考察をこの章では記載する。

4. 1 想定するシステムについて

AI として 2 つのプレイレベルを想定している。プレイレベル強の AI は、ゲームにおいて AI 自身の勝利を目標とさせる AI とし、既存のゲームの AI のように一番勝利しやすいプレイを選択するように学習を進めていく。表 1 でいえばプレイレベル 3 を想定している。

プレイレベル弱の AI は、自身を含めた 4 人のプレイヤーの戦力差が著しく離れることがないようにプレイングをするように学習させる。表 1 でいえばプレイレベル 2 を想定している。

4. 2 考察

AI を学習させるに際して、AI がゲームの状況を理解するための様々な情報を数値化させる必要があると考える。具体的には、手札枚数、残り持ち点などが該当する。これらの値は、特に弱プレイレベルでは各プレイヤーの値の差が重要になってくると思われる。それが単純な差や倍率で判断できるかどうかについては、今後の検討すべき課題と思われる。

第 5 章 今後の展望

カードゲームの多人数戦において、プレイレベルの概念が AI やアルゴリズムでどこまで表現できるのかを調査すること目的に、研究で用いるカードゲームのルールを決め、プログラムを作成した。

ただし AI 部分については実装に至らなかったのが早急に実装する。またゲームとして分かりやすくなるように Python の GUI インターフェースも導入する。

第 6 章 謝辞

本研究を進めるにあたって、ご指導くださいました東海林先生や協力して頂いた研究室メンバーに感謝いたします。

第 7 章 参考文献

[1] デュエパーティー

<https://dm.takaratomy.co.jp/rule/dhueparty/>