

対戦格闘ゲームにおけるゲームAIや操作法の違いが プレイヤーの感じる面白さに与える影響の分析

石原 誠¹ 宮崎 泰地¹ 原田 智広¹ ターウォンマツト ラック^{1,a)}

受付日 2016年2月18日, 採録日 2016年9月6日

概要: 本稿では, 対戦格闘ゲームにおけるゲーム AI (AI) や操作法がプレイヤーの感じる面白さに与える影響について分析する. 対戦格闘ゲームには, キーボードなどの指先による操作と, Kinect を用いて体の動きで操作する方法がある. プレイヤーがいずれの操作においても楽しく対戦格闘ゲームをプレイするためには, プレイヤーと互角に戦うような AI が必要である. また, それを実現させるためには強さのある程度持った AI が必要である. 本稿では, UCT をノード選択における戦略としたモンテカルロ木探索, ルーレット選択, ルールベースの手法を組み合わせることで, 先述した AI を開発する. この AI をベースにし, UCT の評価関数を改変することによってプレイヤーに合わせて強さを調整 (難易度調整) する AI を開発する. そして, AI や操作法がプレイヤーの感じる面白さに与える影響を, キーボード, Kinect のそれぞれの操作において分析する. 対戦格闘ゲームの国際 AI 大会のプラットフォームとして利用されている FightingICE を用いた被験者実験より, 難易度調整はプレイヤーがより楽しんで対戦格闘ゲームをプレイするための重要な要素であり, 特に Kinect において顕著な効果が示された.

キーワード: 対戦格闘ゲーム, 難易度調整, MCTS, FightingICE, Kinect

Analysis of Effects of AIs and Interfaces to Players' Enjoyment in Fighting Games

MAKOTO ISHIHARA¹ TAICHI MIYAZAKI¹ TOMOHIRO HARADA¹ RUCK THAWONMAS^{1,a)}

Received: February 18, 2016, Accepted: September 6, 2016

Abstract: In this paper, we analyze effects of AIs and interfaces to players' enjoyment in fighting games. There are two input interfaces in fighting games. One is finger-control interface such as the keyboard or gamepad, and the other is body-movement-control interface like Kinect. In order to have players enjoy playing fighting games in both input interfaces, AIs are need that evenly fight against their opponent human players. To implement such AIs, it is also necessary to have sufficiently strong AIs to be based upon. In this paper, first, we attempt to make a latter AI, called pAI, by combining MCTS with UCT (used in MCTS's selection criteria), roulette selection, and rule-base. Next, based on pAI, by changing its UCT evaluation function, we develop eAI, an AI that dynamically adjusts its strength to that of its current player in the game. Finally, we analyze effects of both AIs and keyboard as well as Kinect interfaces to players' enjoyment. The results of our experiments using FightingICE, a fighting game platform recently used in a number of game AI competitions, show that adjustment of AIs' strength is an important factor for the player to play the game with more fun.

Keywords: fighting game, adjustment of difficulty, MCTS, FightingICE, Kinect

1. はじめに

対戦格闘ゲームは, プレイヤーやゲーム AI (以下, AI と呼ぶ) がキャラクターを操作し, 様々な攻撃や回避などのアクション (以下, 行動と呼ぶ) を駆使することで対戦相手

¹ 立命館大学情報理工学部知能エンターテインメント研究室
Intelligent Computer Entertainment Laboratory, College of
Information Science and Engineering, Ritsumeikan Univer-
sity, Kusatsu, Shiga 525-8577, Japan

^{a)} ruck@is.ritsumeikan.ac.jp

に勝つことを目的とした、リアルタイムゲームの一種である。ここでは、AIはゲーム内のキャラクターの行動を制御するコンピュータプログラムとして定義する。対戦格闘ゲームの対戦方式は、大きく2つに分けることができる。1つは、2人のプレイヤーが互いにキャラクターを操作して戦うプレイヤーVSプレイヤー(PvP)である。もう1つは、1人のプレイヤーがAIと戦うプレイヤーVSエネミー(PvE)である。本稿では、これらの2つの対戦方式のうちPvEに着目する。

対戦格闘ゲームには、キーボードやゲームパッドなどといった体を動かさずに指先でプレイするような入力インタフェースと、Microsoft Kinect [1] といった体を動かしてプレイする入力インタフェースがある。対戦格闘ゲームは、STREET FIGHTER [2] や鉄拳 [3] など、前者を用いたものが主流となっている。しかし、同インタフェースを用いたゲームは子供の肥満と相関がある、と報告されている [4]。それに対して、Kinectを用いた対戦格闘ゲームは体を動かしてプレイするため、プレイヤーの肥満解消といった健康促進の面で期待できる。しかし、素早い反応が必要な対戦格闘ゲームにおいて、Kinectによる操作では、プレイヤーの反応速度や操作の正確性がキーボードやゲームパッドと比べて劣ることが予想される。それにより、プレイヤーはAIの攻撃に素早く対処することができず一方的に攻撃を受けるといったプレイヤーがつねに不利な状況が続き、対戦格闘ゲームを楽しくプレイできなくなる恐れがある。それを防ぐためにも、Kinectを用いた対戦格闘ゲームでは、AIがより重要な役割を持つ。

対戦格闘ゲームのAIに関して、Kinectが入力インタフェースであることを前提とした研究はさかんではない。しかし、キーボードを前提とした研究はさかんに行われている。近年では、相手に対して自分が有利になるよう行動を選択するAIの研究例として、ルールベース(学習を必要としない、あらかじめ決められたルールにのみ従う)の行動を、自分が持つ複数のルールベースの行動の集合の中から選ぶAI [5] や、Dynamic Scripting [6] を用いるAI [7] などがあげられる。しかし、これらの研究は強さのみに着目しており、楽しさの観点では着目していない。対戦格闘ゲームにおけるAIは、ゲームのエンタテインメント性の向上において重要な要素の1つである。プレイヤーに楽しく対戦格闘ゲームをプレイしてもらうためにも、プレイヤーを楽しませるようなAIが必要である。

本稿では、対戦格闘ゲームにおけるAIや操作法がプレイヤーの感じる面白さに与える影響を分析する。プレイヤーの感じる面白さに影響を与える要素として、特にAIの強さの観点に着目する。まず、UCT (Upper Confidence bounds applied to Trees) [8] をノード選択における戦略としたモンテカルロ木探索 (Monte-Carlo Tree Search: MCTS) [9] をベースとした、強さのある程度を持ったAI (以降、Potent AI; pAI と呼ぶ) を開発する。次に、それをベースにし、

プレイヤーの技量に合わせた強さになるように、ゲーム中で動的に自身の強さを調整 (以降、難易度調整と呼ぶ) するようなAI (以降、Entertaining AI; eAI と呼ぶ) を開発する。そして、pAI と eAI とを、キーボードとKinectとの2つの入力インタフェースごとの被験者実験によって比較し、AIや操作法がプレイヤーの感じる面白さにどのような影響を与えるのか分析する。

2. 関連研究

文献 [10] では、プレイヤーが楽しんでゲームをプレイするのに必要な要素と、ゲームデザインの方法をゲームフロー (図 1) を用いて述べている。図 1 において、横軸がプレイヤーのゲームに対する技量、縦軸がゲームの難易度を表している。プレイヤーの技量とゲームの難易度が図 1 中の“FLOW ZONE”内、つまりFLOW曲線に従うことによって、プレイヤーはゲームを楽しんでプレイできる。よって、プレイヤーの技量に合うようにゲームの難易度を調整する必要がある。これは、ゲームデザインだけでなく、AIの観点でも同様のことがいえる。

文献 [11] では、囲碁や将棋のAIに関してプレイヤーを楽しませるために必要な要素と、そのためのアプローチについて述べている。プレイヤーを楽しませるためには、プレイヤーの技量に合わせてAIが難易度調整をすることで、プレイヤーとよい勝負、もしくはAIが負けてあげることが求められる。同文献では、その流れをゲームフローを用いて説明している。そのゲームフローを対戦格闘ゲームのAIに適用したものを図 2 に示す。難易度調整といっても、同図に示すようにAIが無抵抗でプレイヤーに負けたり、大きく勝っているときに明らかな手加減と思われる行動をとったりしてしまうと、プレイヤーは興ざめしてしまう。よって、プレイヤーと互角に戦うような難易度調整をするAIが必要となる。また、それを実現させるためには、強さのある程度を持ったAIが必要である。

対戦格闘ゲームのAIの難易度調整に関する研究として、

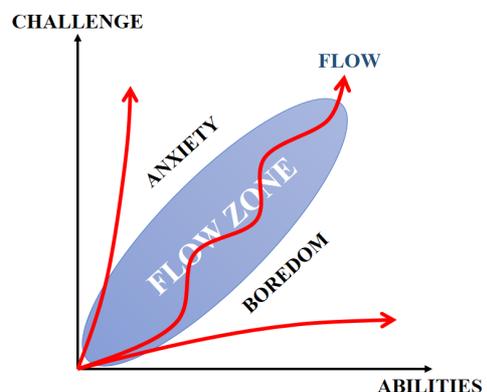


図 1 ゲームフロー
Fig. 1 Game flow.

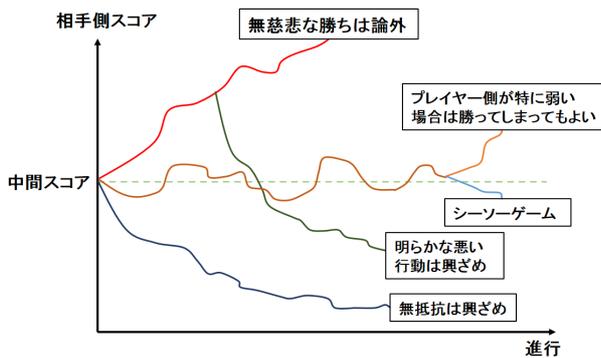


図 2 対戦格闘ゲームにおけるゲームフロー
Fig. 2 Game flow in fighting games.

文献 [12], [13] があげられる。文献 [12] では、 k 近傍法 [14] とシミュレータを用いた、プレイヤーの技量に合わせた対戦格闘ゲーム AI の動的難易度調整について述べている。 k 近傍法を用いてプレイヤーの行動を予測し、AI のとりうる行動と予測した行動に対してシミュレーションを行い、それぞれの行動について評価値を得る。また、AI がゲームに勝っているかどうかを対戦スコアによって判断する。もし勝っていれば評価値が最小のものを、負けていれば最大のものを得た行動を次の AI の行動とし、動的難易度調整を行っている。しかし、同文献では対戦スコアのみを指標とした動的難易度調整を行っており、プレイヤーを楽しませるような観点での試みは行っていない。また、AI どうしによる対戦結果より有用性を評価しており、人間がプレイヤーの場合に関しては考えられていない。

文献 [13] では、対戦格闘ゲーム内における一連の行動の流れに着目し、プレイヤーごとのプレイの振舞いに適応し、プレイヤーの技量に合わせた難易度調整を行う AI を提案している。プレイヤーに合わせた難易度調整という点で本研究と類似しているが、同文献では 1 試合が終了した時点でプレイヤーの技量に合わせるように難易度を調整している。つまり、その試合の間は難易度が固定されている。これに対し、本研究では試合中にプレイヤーの技量に合わせて動的に難易度を調整することを目指しており、この点で同文献とは異なる。

3. MCTS

MCTS は、乱数を用いたシミュレーションを何度も行うことにより近似解を求めるモンテカルロ法と、ゲーム木探索を組み合わせた手法である [9]。以降、3 章と 4 章において MCTS を適用した AI を“自分”、対戦相手を“相手”と呼ぶこととする。MCTS は図 3 に示すように、選択、拡張、シミュレーション、逆伝播の 4 つのステップからなる。同図において、各エッジが自分の行動、各ノードがゲームの状態を表している。以下にそれぞれのステップについて述べる。

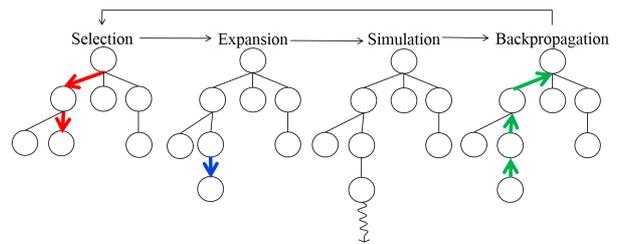


図 3 MCTS による探索の概要
Fig. 3 An overview of MCTS.

• STEP 1 選択

選択基準に従って、ルートノードから末端ノードに到達するまで子ノードを選択する。この選択基準として最もよく使われているものが UCB1 であり、 i 番目のノードの評価値は式 (1) で表される。

$$UCB1_i = \bar{X}_i + C \sqrt{\frac{2 \ln N_i^p}{N_i}} \quad (1)$$

式 (1) において、 C はバランスパラメータ、 N_i^p は i 番目のノードの親ノードの探索回数、 N_i は i 番目のノードの探索回数である。また、 \bar{X}_i は i 番目のノードにおける評価値の平均であり、式 (2) で表される。

$$\bar{X}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} eval_j \quad (2)$$

式 (2) において、 $eval_j$ は j 回目のシミュレーションで得られた評価値である。

このように、UCB1 を木探索に適用したアルゴリズムを UCT と呼ぶ [8]。UCT は囲碁や将棋などのゲーム AI のアルゴリズムに広く利用されており、特に囲碁では顕著な成果をあげている [15]。

• STEP 2 拡張

STEP 1 で末端ノードに到達後、その末端ノードの探索回数が閾値 N_{max} を超えた場合、そこから新たなノードを生成する。

• STEP 3 シミュレーション

末端ノードから、ゲームの終了もしくは一定時間 (回数) が経過するまでシミュレーションを行い、評価値を得る。

• STEP 4 逆伝播

STEP 3 で得られた評価値を、末端ノードからルートノードに到達するまで親ノードへと伝播させる。その過程で、該当するすべてのノードにおいて、選択基準に用いた評価式の各値を更新する。

MCTS は、あらかじめ設定した時間、もしくは回数を経過するまでこの 4 つのステップを繰り返し実行する。その後、評価基準に従って、ルートノードから最も評価が良かったノードにつながるエッジ (行動) を最終的に選択する。この評価基準には、評価値が最も高かったノード、最も探索されたノードなどといった基準がある [16]。本稿

では、ノードの選択基準に $UCB1$ を適用した MCTS を、UCT と定義する。

4. 開発 AI

本章では、AI や操作法がプレイヤーの感じる面白さに与える影響を分析するために開発した AI について述べる。初めに強さのある程度持った AI (pAI) の手法 (4.1 節) を述べ、その手法をベースとした難易度調整を行う AI (eAI) の手法 (4.2 節) を述べる。

4.1 pAI

pAI を実現させるため、UCT、ルーレット選択、ルールベースの 3 つの手法を組み合わせた。以下にそれぞれの手法について述べる。

4.1.1 対戦格闘ゲームにおける UCT

対戦格闘ゲームにおける UCT (以下、Fighting Game UCT: FG-UCT と呼ぶ) の流れを図 4 に示す。通常の UCT では、ノードがゲームの状態を表し、1 つのエッジ (行動) で 1 手先のノードにつながっている。これに対し、FG-UCT ではルートノードのみが現在のゲームの状態を表し、それを除く各ノードは自分の行動を表す。ここでは、攻撃、移動、防御動作を入力してからその動作が終了し、次の動作が実行可能になるまでの一連のまとまりを、行動として扱う。また、各エッジは単にノードどうしのつながりを表し、行動が終了した (再び別の行動が実行可能になった) 時点で、子ノードの行動に遷移する。つまり、図 4 で用いられているゲーム木は、自分の実行する行動のスケジュールを表している。本稿では、自分と相手の体力 (以下、HP と呼ぶ)、エネルギー量、位置、行動、試合の残り時間を含む情報を、ゲームの状態と定義する。

pAI は、図 4 における 4 つのステップを設定時間 T_{max} の間繰り返し実行する。設定時間経過後、ルートノードから最も探索回数が多かったノード (行動) を、最終的に選択する。以下に、図 3 から変更した各ステップの処理について述べる。

- STEP 1' 選択

FG-UCT において、式 (2) における $eval_j$ は式 (3) によって定義される。

$$eval_j = (afterHP_j^{my} - beforeHP_j^{my}) - (afterHP_j^{opp} - beforeHP_j^{opp}) \quad (3)$$

式 (3) において、 $beforeHP_j^{my}$, $beforeHP_j^{opp}$ と $afterHP_j^{my}$, $afterHP_j^{opp}$ はそれぞれ j 回目のシミュレーション前後の自分、相手の HP である。つまり、自分の HP を減少させず、相手の HP をより減少させるほど、評価が高くなる。

本ステップにおいて、式 (1) の第 1 項を式 (4) を用いて正規化した後、式 (1) の第 2 項と足し合わせた値 $UCB1'_i$ が一番大きいノードを、末端ノードに到達するまで選択していく。

$$\bar{X}'_i = \frac{\bar{X}_i - \bar{X}_{min}}{\bar{X}_{max} - \bar{X}_{min}} \quad (4)$$

式 (4) において、 \bar{X}_{max} , \bar{X}_{min} はそれぞれ i 番目のノードと同じ木の深さにある全ノードの $UCB1$ の第 1 項の最大値、最小値である。

- STEP 2' 拡張

3 章の STEP 2 の条件を満たし、かつ末端ノードにおける木の深さが閾値 D_{max} 未満だった場合、そこから新たに生成可能なノードをいっせいに生成する。存在するノードがルートノードのみ (FG-UCT の開始時) の場合、STEP 2' の条件を無視し例外的に生成可能なノードをいっせいに展開する。

- STEP 3' シミュレーション

ルートノードから今いる末端ノードまでに通ったノードの行動と、ルーレット選択によって選ばれた相手の行動 (4.1.2 項) を用いて T_{sim} 間シミュレーションを行い、式 (3) における $eval_j$ を算出する。このとき用いる自分と相手の行動は最大 5 つであり、通ったノードの行動やルーレット選択によって選ばれた相手の行動が 5 つに満たない場合、それぞれの可能な行動からランダムに 5 つになるまで選択する。

- STEP 4' 逆伝播

STEP 3' で得られた $eval_j$ を、末端ノードからルートノードに到達するまで親ノードへと伝播させる。その過程で、該当するすべてのノードの $UCB1$ を構成する各項を更新する。

4.1.2 ルーレット選択

従来の UCT では、シミュレーション時に相手の行動をランダムに決定し、評価を行う。しかし、相手によっては特定の行動しか実行せず、ゲーム中でまったく実行されない行動が存在する可能性がある。そのような場合においても、UCT は相手が実行しない行動まで含めてランダムに相手の行動を決定するため、正しく評価できない可能性がある。よって、シミュレーション時に相手が実際に実行した行動の頻度を加味し、不必要な行動に対してのシミュレーションを防ぐ必要がある。

この問題に対して、シミュレーション時に用いる相手の

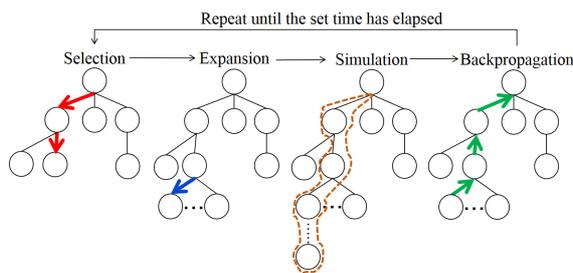


図 4 FG-UCT の探索の概要

Fig. 4 An overview of FG-UCT.

行動を、実行頻度を基準としたルーレット選択によって確率的に決定することで解決を図る。相手の各行動 $action_i$ に対する選択確率 $P(action_i)$ は式 (5) で表される。

$$P(action_i) = \frac{N_{action_i}}{\sum_{i=1}^M N_{action_i}} \quad (5)$$

式 (5) において、 N_{action_i} は $action_i$ の累積の実行回数、 M は行動の種類総数である。つまり、実行頻度が多い相手の行動ほど高い確率で選択され、反対に実行頻度が少ない行動は選択される確率が低くなる。これによって相手の行動を絞ることにより、その相手の行動に最も対抗できる自分の行動を選びやすくしている。

本稿では、ルーレット選択に用いる相手の行動の履歴は、現試合の開始から FG-UCT の開始前までとする。また、相手の行動を記録する際、その時点での相手との水平距離によって近距離、中距離、長距離の 3 つのリストに分けて記録する。そして、シミュレーションを行う時点での相手との水平距離によって、前述した 3 つのリストを使い分ける。これによって、状況に応じて相手の行動を絞ることにより、その相手の行動に最も対抗できる自分の行動をさらに選びやすくしている。本稿では、近距離を相手との水平距離が DIS_{short} 以下のとき、長距離を DIS_{long} より大きいとき、中距離を DIS_{short} より大きく DIS_{long} 以下のときと定義している。

4.1.3 ルールベース

対戦格闘ゲームには、キャラクタごとに必殺技が備わっていることが多い。必殺技は他の技と比べ非常に大きなダメージを与えることができ、それだけでゲームの状況をひっくり返すことが可能である。必殺技による大逆転を防ぐため、相手在必殺技を使用可能な状態になった場合、優先的に必殺技を回避するようなルールベースの行動を設定する。しかし、一定時間が過ぎても必殺技を使用しなかった場合は、FG-UCT による行動選択に戻る。

4.1.4 アルゴリズム

pAI のアルゴリズムを、Algorithm 1 に示す。Algorithm 1 内の変数について、 $spSkillFlag$ は相手の必殺技の実行フラグ、 $bestAction$ は実行する行動、 $state$ はゲームの状態、 $avoidActionTime$ は回避行動を始めてからの経過時間を、それぞれ表している。また、関数について、 $acquireState()$ は現在のゲームの状態を取得、 $canSpSkill()$ は相手在必殺技を実行できる条件（保持エネルギー量など）を満たしているかの確認、 $runUCT()$ は FG-UCT の実行、 $runMyAction()$ は行動の実行を、それぞれ表している。

4.2 eAI

本節では、4.1 節で述べた pAI のアルゴリズムの一部を改変し、プレイヤーに合わせた AI の動的難易度調整を実現する手法について述べる。文献 [11] でも述べられているように、プレイヤーを楽しませるためにはプレイヤーと互角に戦う

Algorithm 1 The algorithm of pAI

```

1:  $spSkillFlag \leftarrow true$  //必殺技の実行フラグ
2:
3: while  $true$  do
4:    $bestAction \leftarrow null$  //実行する行動
5:    $state \leftarrow acquireState()$  //ゲーム状態を取得
6:   // 相手在必殺技を実行可能かどうか
7:   if  $canSpSkill() \ \&\& \ spSkillFlag$  then
8:      $bestAction \leftarrow avoid$  //回避行動
9:     // 回避行動を 10 秒間続けているか
10:    if  $avoidActionTime > 10s$  then
11:       $spSkillFlag \leftarrow false$ 
12:    end if
13:  else
14:    //設定時間が経過するまで FG-UCT を繰り返し実行する
15:     $bestAction \leftarrow runUCT(state, T_{max})$ 
16:     $spSkillFlag \leftarrow true$ 
17:  end if
18:
19:   $runMyAction(bestAction)$  //行動を実行
20: end while

```

ような AI が求められる。それを実現させるために、式 (2) を改変した。改変後の式を (6) に表す。

$$\bar{X}_i = \sum_{j=1}^{N_i} \left(1 - \tanh \left(\frac{diffHP_j}{S} \right) \right) \quad (6)$$

式 (6) において、 S はスケールパラメータ、 $diffHP_j$ は j 回目のシミュレーション後の自分の HP と相手の HP の差の絶対値であり、式 (7) で表される。

$$diffHP_j = |afterHP_j^{my} - afterHP_j^{opp}| \quad (7)$$

ハイパボリックタンジェントの変数が 0 に近づく、つまり $diffHP_j$ が 0 に近づくほど高い評価が得られる。したがって、ゲームの状況にそぐわないような不自然な行動に関しては考慮せずに、eAI はゲームに負けているときは強い行動を、勝っているときは手加減した行動をとる。

5. 評価実験

本章では、AI や操作法の違いが、プレイヤーの感じる面白さにどのような影響を与えるかを分析するための評価実験について述べる。

5.1 実験環境

本節では、評価実験で使用した対戦格闘ゲーム FightingICE と、pAI、eAI に使用したパラメータについて述べる。

5.1.1 FightingICE

FightingICE は 2D の対戦格闘ゲームであり、対戦格闘ゲーム AI の性能を競う国際大会*1のプラットフォームに使用されている [17]。このゲームの大きな特徴は、人間の認知能力を模倣した、状態認知の遅れがシステムに組み込

*1 <http://www.ice.ci.ritsumei.ac.jp/~ftgaic/>

まれていることである。具体的には、システムが AI に対して、相手の位置や行動といった状態は 15 フレーム（約 0.25 秒）前のものしか取得できない、という制約を与えている。これによって、相手の攻撃を見て間髪いれず防御する、といった人間ではできないような行動を AI がとれないようにしている。

FightingICE は、1 試合が 1 ラウンド 60 秒の 3 ラウンドで構成されている。各キャラクタの HP の初期値は 0 に設定されており、攻撃を受けると際限なく減少していく。そして 60 秒が経過した時点で次のラウンドに移行し、HP が 0 にリセットされる。各ラウンド終了時のお互いの HP から、式 (8) によってスコアがラウンドごとに算出される。

$$Score_{my} = \frac{HP_{opp}}{HP_{my} + HP_{opp}} \times 1000 \quad (8)$$

HP_{my} , HP_{opp} はそれぞれ自分、相手の HP である。同式より、スコアの上限は 1,000 である。自分よりも相手の HP の方が小さい場合、自分の獲得スコアは 500 より大きくなる。反対に、相手の HP の方が大きい場合、500 より小さくなる。つまり、500 を超えるスコアを獲得した方の AI をそのラウンドでの勝者とする。

大会は、出場 AI の総当たり戦で行われる。各 AI は、同一の相手に対して、P1（ステージの左側）と P2（ステージの右側）の 2 つの初期サイドについてそれぞれ対戦を行う。2016 年大会ではラウンドの勝ち数の合計で順位が決定される。ただし、勝ち数が同数の場合は、獲得スコアの合計で順位を決定する。

本実験では、FightingICE version 2.00 を使用し、前述した大会ルールに則って行っている。また、状態認知の遅れに対応するために、pAI, eAI における `acquireState()` ではシステムにより提供された状態を基に一度シミュレーションを行い、15 フレーム進める。そして、そこで得られた状態をルートノードとする。これによって、遅れていない現在の状態を取得できるようになる。なお、このとき行うシミュレーション内では、自分と相手は新しい行動を起こさないこととした。

5.1.2 実験に用いたパラメータ

pAI, eAI に用いたパラメータを表 1 に示す。これらのパラメータは、予備実験を行い最適なものを使用している。なお、同表における DIS_{short} , DIS_{long} の値は実験で用いた FightingICE のステージ幅が 960 pixel であることを考慮して決定した。

5.2 実験方法

5.2.1 pAI の強さを示す実験（実験 1）

まず、4.1 節で述べた pAI の強さを示すための実験を行った。pAI を大会に参加させ、計 10 回行う。そして 10 回の大会における平均勝ち数によって、pAI の強さを示す。対戦相手には 2015 年大会に出場した 17 種類の AI を使用した。

表 1 実験に用いたパラメータ

Table 1 The parameters used in the experiments.

Notations	Meanings	Values
C	バランスパラメータ	0.42
N_{max}	訪問回数の閾値	7
D_{max}	ゲーム木の深さの閾値	3
T_{sim}	シミュレーション回数	128 frames
T_{max}	UCT の実行時間	16.5 ms
DIS_{short}	近距離の閾値	110
DIS_{long}	遠距離の閾値	160
S	スケールパラメータ	30

5.2.2 プレイヤの感じる面白さに与える影響を確かめる実験（実験 2）

eAI がプレイヤに合わせた難易度調整ができるか、また、AI や操作法の違いが、プレイヤの感じる面白さにどのような影響を与えるかを、被験者実験によって確かめる。被験者は健康な 20 人（男：18 人、女：2 人、平均年齢 22.1 ± 0.8 歳）を用いた。各被験者に対し、次に示す 2 つのシナリオで実験を行った。

● キーボードを用いたシナリオ

被験者はディスプレイが正面にくるように椅子に座る。始めに被験者に対して、キーボードによる操作方法について説明した後、以下のような手順で実験を行う。

(1) 練習として、何もしない AI と 1 試合（3 ラウンド）行う。

(2) pAI, eAI とそれぞれ 5 試合ずつ、計 10 試合行う。

ただし、手順 2 において pAI, eAI との対戦順序はランダムに選択する。

● Kinect を用いたシナリオ

Kinect は床から 1.2m の高さ、被験者から 2.5m 離れたところに設置する。被験者はディスプレイが正面にくるように立つ。また、Kinect を用いて FightingICE のキャラクタを操作するために、文献 [18] のシステムを用いた。始めに被験者に対して、Kinect による操作方法について説明した後、以下のような手順で実験を行う。

(1) 練習として、何もしない AI と 1 試合（3 ラウンド）行う。

(2) pAI, eAI とそれぞれ 5 試合ずつ、計 10 試合行う。

前シナリオと同様に、手順 (2) において pAI, eAI との対戦順序はランダムに選択する。しかし、本シナリオでは体を動かして AI を操作するため、被験者の疲労を考慮し、各試合間に 1 分の休憩時間を設ける。これは、ボクシングのルールに則っている。

実験 2 では、被験者に対して事前に AI に関する情報や、戦う AI の順番については知らせない。また、被験者の実験に対する順序効果を考慮し、実験を行う前に、被験者にキーボードを用いて 4.1.1 項で述べた手法からルーレット選択を除いたもの（シミュレーション時の相手の行動選択

表 2 アンケートの内容

Table 2 The questionnaire content.

Metrics	Questions
試合の面白さ	今の試合は面白かったですか
相手の強さ	敵はあなたと比べてより強かったですか
相手の行動の自然さ	敵の動きは自然でしたか

がランダム)を適用した AI を相手に 5 試合行ってもらった。そこで得られた 5 試合の平均獲得スコア (以下、獲得スコアと呼ぶ) をもとに、グループ間に差が出ないように被験者を A, B の 2 つのグループに分けた。実験は、グループ A はキーボード, Kinect の順, グループ B は Kinect, キーボードの順に行った。

また、両シナリオにおいて各試合終了後に表 2 に示すアンケートに回答してもらった。アンケートは文献 [19] を参考に作成した。アンケートの内容は、本稿で着目した試合の面白さと AI の強さに加え、図 2 に示すように、相手が勝っているときにあからさまな不自然な行動をしないかどうかを確かめるため、行動の自然さについての項目を取り入れた。肯定的な回答ほど高い評価値になるように各項目に関して、「とてもそう思う」=5, 「ややそう思う」=4, 「どちらともいえない」=3, 「あまりそう思わない」=2, 「まったくそう思わない」=1 の 5 段階で評価してもらった。

6. 結果の考察

本章では、実験 1, 実験 2 を行った結果と、それぞれに対する考察について述べる。

6.1 実験 1 に対する結果と考察

FightingICE の大会を 10 回行った結果における、上位 3 つの AI の 1 大会の平均勝ち数を表 3 に示す。本実験では、pAI を含めた 18 個の AI の総当たり戦で行ったため、1 大会における各 AI がとりうる最大の勝ち数は 102 である。表 3 より、pAI は平均で 72.0 勝の成績を修めて 2 位という結果になった。同表の Machete, Nilmir4ri は、FightingICE の 2015 年大会においてそれぞれ 1, 2 位の AI である。これらのことから、4.1 節で述べた手法を適用することによって、強さのある程度持った AI が実現できたといえる。

6.2 実験 2 に対する結果と考察

本節では、被験者の pAI, eAI に対する獲得スコアとアンケートのそれぞれの観点から、結果と考察を述べる。なお、本節における結果の検定には対応なしのウィルコクソンの順位和検定 [20] を用いている。

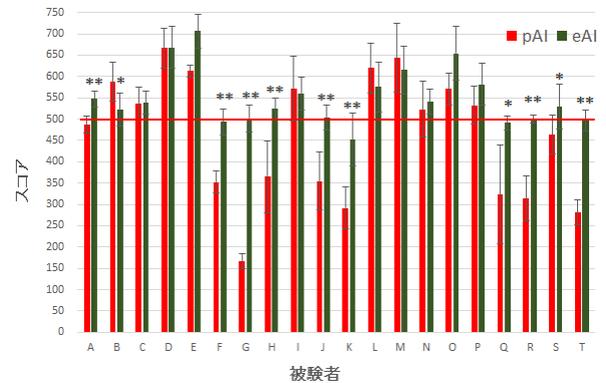
6.2.1 被験者の獲得スコアに関する結果と考察

キーボード実験における、被験者ごとの pAI, eAI に対する獲得スコアを図 5 (a), Kinect 実験における、被験者

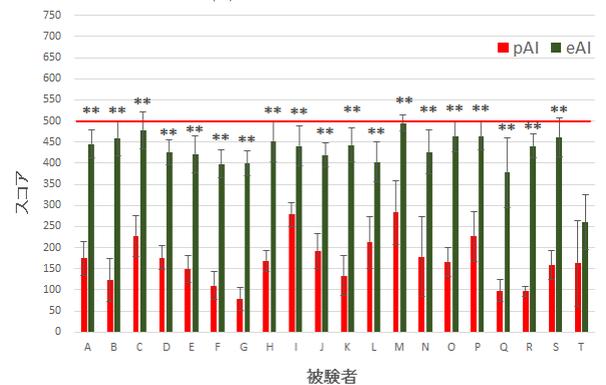
表 3 10 回の大会における上位 3AI の平均勝ち数

Table 3 Average # of wins of the top 3 AIs in 10 competitions.

AI Names	Average # of wins
Machete	76.9
pAI	72.0
Nilmir4ri	70.6



(a) キーボード実験



(b) Kinect 実験

図 5 15 ラウンド (5 試合) における各 AI に対する被験者の 1 ラウンドの平均獲得スコア. (a), (b) はそれぞれ入力インタフェースがキーボード, Kinect のときの結果である。また, *, ** はそれぞれ pAI と比べ, 有意水準 5%, 1% で差があることを示す

Fig. 5 The average round scores each player obtained against pAI and eAI in 15 rounds (5 games). (a) and (b) show the result when the input interface is keyboard and Kinect, respectively. The symbols * and ** show significant difference at 5%, 1% to pAI, respectively.

ごとの pAI, eAI に対する獲得スコアを図 5 (b) にそれぞれ示す。各図において、横軸が被験者、縦軸が獲得スコアを表している。獲得スコアは、500 に近づくほどその AI に対して互角の勝負をした、ということの意味する。図 5 (a) より、被験者 G や K といった、pAI に対して負けている被験者が、eAI に対してスコアが 500 に近い、もしくは 500 を超えるスコアを獲得していることが分かる。たとえば、被験者 G の場合、pAI に対する獲得スコアが 168 であり、互角の強さといえるスコアの 500 に対し 332 の差があった。一方、eAI に対する獲得スコアは 502 であり、差が明

らかに縮まっている．さらに、それらの被験者の獲得スコアについて、10 人の被験者に対して有意差が認められた．

反対に、被験者 D や E といった pAI に対して勝っている被験者に対しては、eAI は被験者 B を除いて有意差が認められなかった．たとえば、被験者 D の場合、pAI, eAI に対する獲得スコアはそれぞれ 667, 668 であり、両 AI とも差が変わっていないことが分かる．これは、eAI が負けているときは被験者に対して自分が有利な行動を行い、スコアを 500 に戻そうとする．つまり、pAI と同じような行動原理になっているためだと考えられる．

また図 5 (b) より、すべての被験者について、pAI に対する獲得スコアに比べ eAI に対する獲得スコアが 500 に近づいていることが分かる．たとえば被験者 M の場合、pAI に対する獲得スコアが 284 であったのに対し、eAI に対する獲得スコアは 495 であり、eAI と互角の勝負をしている．さらに、被験者 T を除くすべての被験者に対して有意差が認められた．

これらのことから、pAI に負けている被験者に対して難易度調整を行うことによって、互角に近い試合を実現できていることが示された．また、pAI に勝っている被験者に対しては難易度調整がうまくいかなかったことから、より有望な eAI を実現させるためにはさらなる強さを持つ AI が必要である、ということがいえる．

次に、図 5 (a), 図 5 (b) を比較すると、入力インタフェースがキーボードのときに比べ、Kinect のときでは被験者の両 AI に対する獲得スコアが明らかに減少していることが分かる．つまり、Kinect を用いた場合、プレイヤーはキーボードを用いた場合に比べ弱くなるということがいえる．これは、行動のキー入力の際、キーボードでは指先の動きだけでキーを入力することができるため、プレイヤーはそのときのゲームの状況に応じた素早い対処ができる．一方、Kinect ではキー入力にパンチやキックなどの大きな体の動きが必要なため、プレイヤーはキー入力に多くの時間を要する．その結果、行動が相手の後手に回ってしまい、先に攻撃を当てられるといった状況が頻繁に観察されたため、このような結果になったのだと考えられる．加えて、体を動かし続けることによる疲労の蓄積によって、プレイヤーのパフォーマンスが低下したのも要因の 1 つだと考えられる．

6.2.2 獲得スコアの時間的推移に関する結果と考察

キーボード実験における、pAI, eAI の各被験者に対する獲得スコアの時間的推移を図 6 (a), Kinect 実験における、pAI, eAI の各被験者に対する獲得スコアの時間的推移を図 6 (b) にそれぞれ示す．各図における被験者は、図 6 (a) では以下のように分けた各グループ内において、pAI に対する獲得スコアが中央値であるもの、図 6 (b) では pAI に勝っている、もしくは互角の勝負をしている被験者がいなかったため、全体において pAI に対する獲得スコアが中央値であるものを用いた．ここでは、グループ内の人数が偶

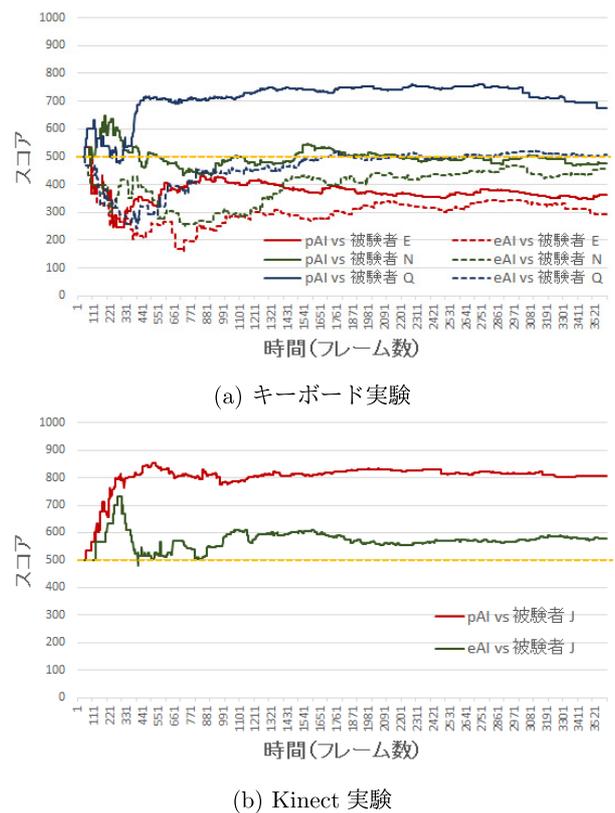


図 6 15 ラウンド (5 試合) における各代表的な被験者に対する各 AI の 1 ラウンドの平均獲得スコアの時間的推移．(a), (b) はそれぞれ入力インタフェースがキーボード, Kinect のときの結果である．また、(a) は被験者 E, N, Q, (b) は被験者 J が対戦相手である

Fig. 6 The temporal transition of average round scores each AI obtained against typical subjects in 15 rounds. (a) and (b) show the result when the input interface is keyboard (vs subjects E, N, and Q) and Kinect (vs subject J), respectively.

数だった場合、該当者のうち、より大きいスコアを獲得した方を中央値とした．

pAI に大きく勝っている

pAI に対して、獲得スコアが 550 より大きい被験者 7 人が該当．グループ内の平均獲得スコアは 610 ± 34 ．中央値は被験者 E が獲得した 613．

pAI と互角の勝負をしている

pAI に対して、獲得スコアが 450～550 の被験者 5 人が該当．グループ内の平均獲得スコアは 509 ± 29 ．中央値は被験者 N が獲得した 523．

pAI に大きく負けている

pAI に対して、獲得スコアが 450 未満の被験者 8 人が該当．グループ内の平均獲得スコアは 306 ± 59 ．中央値は被験者 Q が獲得した 323．

図 6 (a) より、被験者 Q に対して、eAI は 500 に近い獲得スコアで推移していることが分かる．また、被験者 E に対して、pAI, eAI ともに獲得スコアが つねに 500 未満で

表 4 全被験者におけるアンケートの平均評価
Table 4 The average evaluation score of the questionnaire in all subjects.

Questions	Interface	pAI	eAI
試合の面白さ	キーボード	3.34	3.56
	Kinect	2.90	3.48
相手の強さ	キーボード	3.52	2.66
	Kinect	4.64	3.55
相手の行動の自然さ	キーボード	3.60	3.33
	Kinect	3.74	3.12

推移していることが分かる。これらの結果と同様の傾向を示す獲得スコアの推移が、それぞれのグループに該当する全被験者において見て取れた。また、被験者 N に対して、pAI の方が eAI に比べ 500 に近い獲得スコアで推移していることが分かる。これは、被験者 N の技量と pAI の難易度が釣り合っており、難易度調整を行わなくても互角の勝負ができたためだと考えられる。この結果と同様の傾向を示す獲得スコアの推移が、pAI と互角の勝負をしているグループに該当する被験者のうち、pAI に勝っている被験者 (3 人) で見て取れた。pAI に負けている被験者 (2 人) では、被験者 Q のような大きな推移の差はないものの、eAI の方が 500 により近いところで獲得スコアが推移している傾向が見て取れた。このことから、pAI と互角以上の勝負をした被験者に対して、eAI は難易度調整を行えていないことが示された。

次に、図 6(b) より、被験者 J に対して pAI に比べ eAI の方が、500 に近い獲得スコアで推移していることが分かる。この結果と同様の傾向を示す獲得スコアの推移が、全被験者において見られた。

これらの結果より、pAI に負けている被験者に対して、eAI は入力インタフェースに関係なく、試合中で動的に難易度調整が行えていることが示された。

6.2.3 アンケートに関する結果と考察

アンケートの各質問項目における、全被験者の評価の平均値を表 4 に示す。評価に関して、試合の面白さ、相手の行動の自然さの項目については、評価値が 5 に近づくほど評価が高く、相手の強さの項目については、評価が 3 になるほど評価が高い。各質問項目について、それぞれ考察していく。

● 試合の面白さに関する結果と考察

表 4 の評価結果より、キーボード、Kinect 操作の両方において pAI より eAI の方が評価が高かった。さらに Kinect 実験に関しては、有意水準 1% で有意差が認められた。これは、図 5(b) から分かるように、被験者は pAI に対して大きく負けている。つまり、pAI の強さが被験者の技量に合っておらず、被験者がその試合に面白さを感じなくなってしまったためであると考えられる。一方、Kinect 操作時の被験者の eAI に対する平均獲得スコアは 429 である。こ

れより、eAI が被験者と互角、もしくはよい勝負をすることによって、被験者はその試合を面白く感じるということが考えられる。

また、キーボードと Kinect における eAI に対する評価から、入力インタフェースの違いによる試合の面白さの感じ方に有意差は認められなかった。これらのことから、AI の難易度調整を行うことによって、入力インタフェースに関係なく被験者に試合をより面白く感じさせることができた、ということがいえる。

● 相手の強さに関する結果と考察

表 4 の評価結果より、キーボード、Kinect 操作の両方において、被験者は pAI の方が eAI よりも強いと感じていることが分かる。どちらの操作においても有意水準 1% で有意差が認められた。この結果から、eAI は pAI に比べ難易度を下げている、ということがいえる。

また、Kinect 操作時の eAI に対する評価から、このシナリオにおいて eAI は被験者の技量よりも少し強いくらいに難易度調整を行っていた、ということがいえる。これは先にも述べたように、被験者の eAI に対する獲得スコアが 429 であったことから考えられる。

● 相手の行動の自然さに関する結果と考察

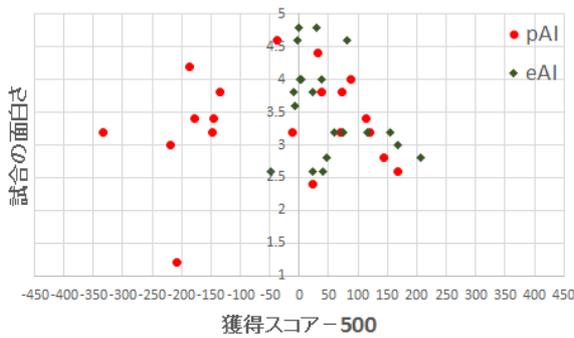
表 4 の評価結果より、キーボード、Kinect 操作の両方において、いずれも 3 を上回る結果となったが、被験者は pAI の方が eAI よりも動きが自然だったと感じていることが分かる。特に Kinect 操作時の評価については、有意水準 1% で有意差が認められた。これは、pAI はゲームの状況における最善な行動をつねに選択しているため、被験者はその行動をゲームの状況にあった自然な行動、と評価したと考えられる。一方、eAI は被験者の技量に合わせて難易度を調整するため、ゲームの状況にそぐわない行動、つまり手加減を行う。その手加減を行う際の行動として、eAI はプレイヤーとの距離が十分離れているにもかかわらず後退を繰り返すといった、あからさまな手加減をしていると判断できる行動が時折りあった。これにより、eAI は pAI よりも評価が低くなったということが考えられる。しかし、先にも述べたように、試合の面白さに関する評価が eAI の方が高かったことから、被験者は相手の動きが多少不自然であっても、互角の勝負ができた方が面白いと感じている、ということが考えられる。

6.2.4 獲得スコアと試合の面白さ、相手の強さとの相関の分析結果と考察

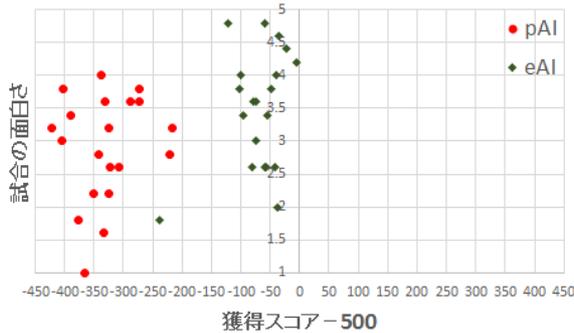
本項では、各被験者の両 AI に対する獲得スコアとアンケートにおける試合の面白さ、および獲得スコアと相手の強さの評価との相関について分析する。以下、本項で示す図において、横軸が被験者の獲得スコアから中間スコアである 500 を引いたもの、縦軸がアンケートの評価を表す。

● 獲得スコアと試合の面白さとの相関の分析結果と考察

キーボード実験における、被験者ごとの pAI, eAI に



(a) キーボード実験



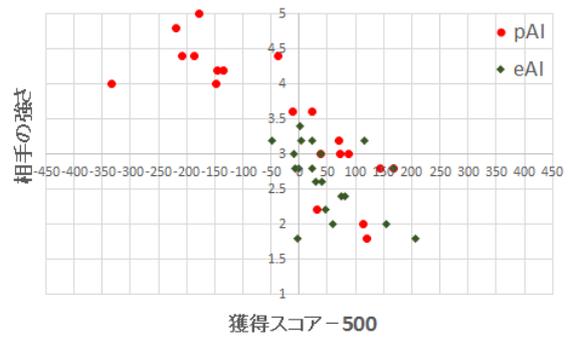
(b) Kinect 実験

図 7 各被験者の両 AI に対する平均獲得スコアと試合の面白さの平均評価との関係. (a), (b) はそれぞれ入力インタフェースがキーボード, Kinect のときの結果である

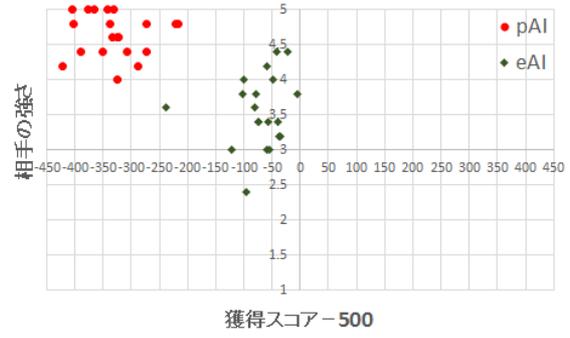
Fig. 7 The relationship between the average scores each player obtained against both AIs and the average evaluation scores about fun of the game. (a) and (b) show the result when the input interface is keyboard and Kinect, respectively.

対する獲得スコアと試合の面白さの平均評価との関係を図 7(a), Kinect 実験における, 被験者ごとの pAI, eAI に対する獲得スコアと試合の面白さの平均評価との関係を図 7(b) に, それぞれ示す. 図 7(a) より, キーボードが入力インタフェースの場合は, スコアの差が 0 に近いところではアンケートの評価が 3.5 ~ 5 の間に多く分布し, 差が開くほど多少ばらつきがあるものの評価が下がっていく傾向が確認できる. 獲得スコアが 500 未満のデータと, 500 以上のデータのそれぞれに対して, 獲得スコアと試合の面白さの平均評価との相関係数を算出した結果は 0.40, -0.45 であり, とともに弱い相関が示された. つまり, キーボードが入力インタフェースの場合は, 獲得スコアが 500 に近づくほどプレイヤはその試合を面白く感じ, 離れるほど面白く感じられなくなる, という弱い傾向がある.

また図 7(b) より, Kinect が入力インタフェースの場合は, スコアの差が -100 ~ 0 のあたりと -450 ~ -200 のあたりに分かれてアンケート評価が分布していることが分かる. また, それぞれの分布についてのアンケート評価は大きくばらついている. この 2 つの分布を比較してみると, スコアの差が 0 に近い分布の方が評価が高い傾向が確認で



(a) キーボード実験



(b) Kinect 実験

図 8 各被験者の両 AI に対する平均獲得スコアと相手の強さの平均評価との関係. (a), (b) はそれぞれ入力インタフェースがキーボード, Kinect のときの結果である

Fig. 8 The relationship between the average scores each player obtained against both AIs and the average evaluation scores about the opponent's strength. (a) and (b) show the result when the input interface is keyboard and Kinect, respectively.

きる. これは, 表 4 の Kinect での面白さの評価において, eAI に対する評価が pAI よりも高いこととつながる. 同図における, 獲得スコアと試合の面白さの平均評価との相関係数を算出した結果は 0.39 であり, 弱い相関が示された. つまり, Kinect が入力インタフェースの場合, 相手がプレイヤと互角かそれよりも強いとき, キーボードと同様の傾向が見られる.

● 獲得スコアと相手の強さとの相関の分析と考察

キーボード実験における, 被験者ごとの pAI, eAI に対する獲得スコアと相手の強さの平均評価との関係を図 8(a), Kinect 実験における, 被験者ごとの pAI, eAI に対する獲得スコアと相手の強さの平均評価との関係を図 8(b) に, それぞれ示す. 図 8(a) より, キーボードが入力インタフェースの場合は, スコアの差が負の方向に開くほどアンケートの評価が 5 に近づき, 正の方向に開くほどアンケートの評価が 1 に近づく傾向が確認できる. 同図における, 獲得スコアと相手の強さの平均評価との相関係数を算出した結果は -0.79 であり, 強い相関が示された. つまり, キーボードが入力インタフェースの場合は, プレイヤは相手に対して大きく負けるほどその相手を強いと感じ, 大きく勝つほ

どその相手を弱く感じる、という強い傾向がある。

また図 8(b) より, Kinect が入力インタフェースの場合は, スコアの差が -200 より負の方向に開くところではアンケートの評価が 5 に近いところに多く分布し, スコアの差が 0 の付近では, アンケートの評価が 3 に近いところに多く分布していることが確認できる. 同図における, 獲得スコアと相手の強さの平均評価との相関係数を算出した結果は -0.72 であり, 強い相関が示された. つまり, Kinect が入力インタフェースの場合においても, プレイヤは相手に対して大きく負けるほどその相手を強いと感じ, 相手と互角の勝負をするほど, その相手をちょうどいい強さを感じる, という強い傾向がある.

7. おわりに

近年の対戦格闘ゲームは, キーボードやゲームパッドが入力インタフェースのものが主流であるが, 体を動かしながらプレイできる Kinect を入力インタフェースとしたゲームは健康促進に期待できる. いずれのインタフェースに対しても, プレイヤが楽しく対戦格闘ゲームをプレイするための要素の 1 つとして, プレイヤと互角に戦う AI があげられた. それを実現させるためには強さをある程度持った AI が必要であった.

本稿では UCT, ルーレット選択, ルールベースの手法を組み合わせることで強さをある程度持った AI を開発した. また, この AI をベースにし, UCT の評価関数を改変することによってプレイヤに合わせた難易度調整を図った. そして AI や操作法の違いが, プレイヤの感じる面白さなどのような影響を与えるかについて分析した.

評価実験の結果から, 前述した 3 つの手法を組み合わせることによって強さをある程度持った AI が実現できた. また, UCT の評価関数の改変によって, pAI に負けているプレイヤに対して, その技量に合わせた難易度調整が行えたことが示された. 加えて, アンケートの結果より, 難易度調整は特に Kinect による操作において, プレイヤがより楽しんで対戦格闘ゲームをプレイするための重要な要素であることが示された.

しかし, 難易度調整を行った AI は, 難易度調整を行っていない AI と比較して行動の自然さに対する評価が低かったため, 今後の課題として, あからさまな手加減だと感じられるような行動を防ぐことがあげられる. 加えて, pAI よりも強いプレイヤに対して適応できるように, さらなる強さを持った AI も開発していきたい.

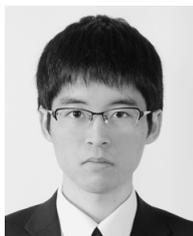
参考文献

- [1] Webb, J. and Ashley, J.: *Beginning Kinect Programming with the Microsoft Kinect SDK*, [E-book], Apress© (2012).
- [2] CAPCOM: ストリートファイターシリーズ公式サイト (オンライン), 入手先 (<http://www.capcom.co.jp/>)
- [3] game/content/streetfighter/) (参照 2016-01-19).
バンダイナムコエンターテインメント: TEKKEN OFFICIAL (オンライン), 入手先 (<http://www.tekken-official.jp/>) (参照 2016-01-19).
- [4] Vandewater, E.A., Shim, M.S. and Caplovitz, A.G.: Linking obesity and activity level with children's television and video game use, *Journal of Adolescence*, Vol.27, No.1, pp.71–85 (2004).
- [5] Sato, N., Tamsirikkul, S., Sone, S. and Ikeda, K.: Adaptive Fighting Game Computer Player by Switching Multiple Rule-based Controllers, *3rd International Conference on Applied Computing and Information Technology (ACIT 2015)*, pp.52–59 (2015).
- [6] Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I. and Postma, E.: Adaptive game AI with dynamic scripting, *Machine Learning*, Vol.63, No.3, pp.217–248 (2006).
- [7] Majchrzak, K., Quadflieg, J. and Rudolph, G.: Advanced Dynamic Scripting for Fighting Game AI, *Entertainment Computing-ICEC 2015*, pp.86–99 (2015).
- [8] Kocsis, L. and Szepesvari, C.: Bandit Based Monte-Carlo Planning, *Machine Learning, ECML*, pp.282–293 (2006).
- [9] 美添一樹: モンテカルロ木探索, 情報処理, Vol.49, No.6, pp.686–693 (2008).
- [10] Chen, J.: Flow in games (and everything else), *Comm. ACM*, Vol.50, No.4, pp.31–34 (2007).
- [11] 池田 心: 楽しませる囲碁・将棋プログラミング, オペレーションズ・リサーチ: 経営の科学, Vol.58, No.3, pp.167–173 (2013).
- [12] Nakagawa, Y., Yamamoto, K. and Thawonmas, R.: Online Adjustment of the AI's Strength in a Fighting Game Using the k-Nearest Neighbor Algorithm and a Game Simulator, *IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pp.494–495 (2014).
- [13] Moriyama, K. et al.: An Intelligent Fighting Videogame Opponent Adapting to Behavior Patterns of the User, *IEICE TRANSACTIONS on Information and Systems*, Vol.97, No.4, pp.842–851 (2014).
- [14] Smola, A. and Vishwanathan, S.V.N.: *Introduction to Machine Learning*, [E-book], Cambridge University Press (2014).
- [15] Gelly, S. et al.: The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions, *Comm. ACM*, Vol.55, No.3, pp.106–113 (2012).
- [16] Sephton, N., Cowling, P.I. and Slaven, N.H.: An Experimental Study of Action Selection Mechanisms to Create an Entertaining Opponent, *Computational Intelligence and Games (CIG)*, pp.122–129 (2015).
- [17] Lu, F. et al.: Fighting Game Artificial Intelligence Competition Platform, *IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, pp.320–323 (2013).
- [18] Paliyawan, P., Sookhanaphibarn, K., Choensawat, W. and Thawonmas, R.: Towards Universal Kinect Interface for Fighting Games, *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pp.332–333 (2015).
- [19] 井上雅之, 望月崇由, 坪井俊明: 3次元仮想空間における操作性とアバタデザインの評価と考察, 映像情報メディア学会誌, Vol.60, No.8, pp.1296–1306 (2006).
- [20] Wilcoxon, F., Katti, S.K. and Wilcox, R.A.: Critical Values and Probability Levels for the Wilcoxon Rank Sum Test and the Wilcoxon Signed Rank Test, *Selected tables in Mathematical Statistics*, Vol.1, pp.171–259 (1970).



石原 誠

2016年3月立命館大学情報理工学部知能情報学科卒業。現在、同大学大学院情報理工学研究科情報理工学専攻博士課程前期課程に在籍。対戦格闘ゲーム AI の研究に従事。



宮崎 泰地

2016年3月立命館大学情報理工学部知能情報学科卒業。同年4月より株式会社コナミデジタルエンタテインメントに勤務。在学中、対戦格闘ゲーム AI の研究に従事。



原田 智広

2012年電気通信大学大学院博士前期課程修了。同年同大学院博士後期課程入学。2012年日本学術振興会特別研究員 DC1 採択。2015年電気通信大学大学院博士後期課程修了。同年立命館大学情報理工学部の助教に着任、現在に至る。博士(工学)。進化計算、機械学習、知的エージェントの研究に従事。進化計算研究会会員。



ターウォンマツト ラック
(正会員)

1994年東北大学大学院工学研究科情報理工学専攻博士課程修了。博士(工学)。2004年4月より立命館大学情報理工学部知能情報学科教授。計算知能およびゲーム AI の研究に従事。