

G-103

# Kinect を用いた対戦格闘ゲームにおけるユーザの運動量を向上させる AI の探求

## Investigating Kinect-based Fighting Game AIs That Increase the Player's Amount of Exercise

石原 誠<sup>†</sup>      宮崎 泰地<sup>†</sup>      原田 智広<sup>†</sup>      ターウオンマット ラック<sup>†</sup>  
 Makoto Ishihara    Taichi Miyazaki    Tomohiro Harada    Ruck Thawonmas

**要約**: 本論文では, Kinect を用いた対戦格闘ゲームにおけるユーザの健康促進のための AI について述べる. ユーザの運動量を向上させるために, 対戦格闘ゲーム FightingICE においてユーザの運動量を向上させるために必要な, 対戦相手の AI の要素について分析する. 実験として, モンテカルロ木探索 (MCTS) の一種である UCT を適用した AI (UCTAI) と, UCTAI に  $k$  近傍法を適用した AI ( $k$ NN-UCTAI) との 2 つの AI をユーザと対戦させ, ユーザが使用した JUMP 行動の回数を比較する. 実験の結果, 被験者は  $k$ NN-UCTAI の方に対してより多くの JUMP 行動を行っていることが示された.

**キーワード**: Kinect; 対戦格闘ゲーム; games for health; UCT;  $k$  近傍法; FightingICE

### 1. はじめに

対戦格闘ゲームにおける AI は, ゲームの娯楽性の向上において, 重要な要素のひとつである. 近年, 対戦格闘ゲームの AI に対する研究が盛んになっている. 例えば[1]の論文では, 技のコンボに焦点を当て, ユーザが 1 人でも楽しんで対戦格闘ゲームをプレイできるような AI について述べている.

また, ユーザの健康促進を目的とした, Kinect を用いた対戦格闘ゲームの研究が注目されつつある[2]. Kinect を用いて対戦格闘ゲームのキャラクターを操作することで, 運動しながらゲームをプレイすることが出来, ユーザの健康促進に繋がる. 対戦格闘ゲームには多彩な技が用意されており, ユーザはそれぞれの技に対応したポーズを行ってその技を使用し, 相手と戦う. 多くの技を使うためには, 多くのポーズをする必要があり, ユーザの運動量も多くなる. そのため, 健康促進においてはユーザに多くのポーズをさせる AI が必要である.

本論文では, Kinect を用いた対戦格闘ゲームにおけるユーザの健康促進のための AI について述べる. 対戦格闘ゲーム FightingICE[3]において, ユーザの運動量を向上させるために必要な対戦相手の AI の要素, とりわけ JUMP 系の行動について着目し, 分析することが目的である. 実験として, モンテカルロ木探索 (MCTS) の一種である UCT を適用した AI (UCTAI) と, UCTAI に  $k$  近傍法を適用した AI ( $k$ NN-UCTAI) との 2 つの AI をユーザと対戦させ, ユーザが使用した JUMP 行動(JUMP, FOR\_JUMP)の回数を比較する.

### 2. 使用 AI のアルゴリズム

#### 2.1 UCT

UCT[4]は 2006 年に Kocsis らによって提案された. UCT において, 各子ノードの UCB(Upper Confidence Bounds)値は式(1)を用いて求められ, 最終的に UCB 値が最大となる子ノードを選択する. 末端ノードに到達後, その末端ノードの探索回数が予め決めた閾値を超えており, 且つゲーム木の深さが上限でない場合, そのノードから新たな子ノ

ードを生成し, そうでない場合, 乱数を用いたシミュレーションによる探索を行う.

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

UCTAI において,  $\bar{X}_i$  は  $i$  番目の子ノードにおける敵キャラクターの HP の変化量と自キャラクターの HP の変化量の差から計算される評価値の平均である. つまり, 自分の HP を減少させずに敵の HP を減少させるほど評価値が高くなる. また,  $C$  はバランスパラメータ,  $N_i^p$  は  $i$  番目の子ノードの親ノードにおける探索回数,  $N_i$  は  $i$  番目の子ノードにおける探索回数である.

UCTAI は, UCT を実行速度的に限界と考えられる 17ms の間実行し, 最終的に最も探索回数が多いノードを AI の行動として選択する.

#### 2.2 $k$ 近傍法

$k$  近傍法[5]とは, 過去のデータとそのデータのクラスを記録しておき, 新たなデータが与えられたとき, そのデータから最も近い  $k$  個のデータを選択し多数決によって新たなデータのクラスを決定する, という方法である.  $k$ NN-UCTAI において, 以下のようなアルゴリズムで自分の行動を決定する.

1. 相手が JUMP をしたときの, 自分の行動(ACTION)と相手との相対座標(RelativeX, RelativeY)をデータセットとして記録する.
2. データセットの個数が  $N$  より多いとき, 現在相対座標から一定距離内( $K\_DISTANCE$ )にあり, かつ最も近い  $k$  個のデータセットを選択する.
3. 選択したデータセット内の行動で多数決を取り, 一番多かった行動を次の自分の行動とする.

図 1 の例では, 自分の行動は ACTION\_B と決定される.

<sup>†</sup> 立命館大学情報理工学部

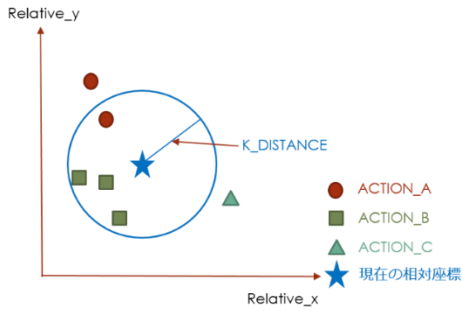


図 1. 記録されたデータの例

もし、選択したデータセットにおいて同じ個数の行動があった場合、その中からランダムで行動を決定する。また 2.において、データセットの個数が  $N$  以下、もしくは  $K\_DISTANCE$  内にデータが  $k$  個以下しかない場合、 $kNN$ -UCTAI は前述した UCT を用いて行動を決定する。

### 3. 実験

#### 3.1 環境

Kinect は床から 1.0m の高さ、被験者から 2.5m 離れたところに設置する。また、被験者は Kinect とモニターが正面にくるように立つてもらおう。

対戦格闘ゲームのプラットフォームは、今年の対戦格闘ゲーム AI の大会に用いられている FightingICE ver. 1.22 を使用する。ゲームルールは、大会のルールに準拠する。また、P-Tracker[2]を使用し、Kinect を入力インタフェースとして用いて FightingICE のキャラクターを操作する。

#### 3.2 実験に用いたパラメータ

本論文で行った実験では、UCT について、新たな子ノードを生成するときのノードの探索回数の閾値は 10、ゲーム木の深さは最大 2、 $C$  の値は 3 とした。また、 $k$  近傍法について、 $N$  の値は 5、 $k$  の値は 3、 $K\_DISTANCE$  の値は 85 とした。これらのパラメータは、予備実験において性能の良かった値を使用している。

#### 3.3 方法

被験者にはキーボードと Kinect を用いて FightingICE のキャラクターを操作し、UCTAI、 $kNN$ -UCTAI とそれぞれ対戦してもらう。最初に、被験者にキーボードを用いてそれぞれの AI と対戦してもらう。まず、被験者は何も行動をしない AI と 2 試合戦い、キーボードでの操作の練習をする。次に、被験者は UCTAI、 $kNN$ -UCTAI とそれぞれ 5 試合ずつ、計 10 試合対戦する。

キーボードでの対戦終了後、被験者には Kinect を用いてそれぞれの AI と対戦してもらう。被験者は、キーボードの時と同様に UCTAI、 $kNN$ -UCTAI とそれぞれ 5 試合ずつ、計 10 試合対戦する。なお、試合間の休憩時間は、ボクシングの標準休憩時間に則りすべて 1 分とする。また、この実験では、対戦する AI の順番はランダムで選ぶ。

### 4. 結果

各 AI に対する、被験者の 1 試合間の JUMP 行動の平均実行回数を図 2 に示す。被験者は  $kNN$ -UCTAI に対して、UCTAI よりも多くの JUMP 行動を出していることが分かる。つまり、 $kNN$ -UCTAI は被験者に JUMP 行動をさせるような自分の行動を選択しており、 $k$  近傍法が機能しているといえる。

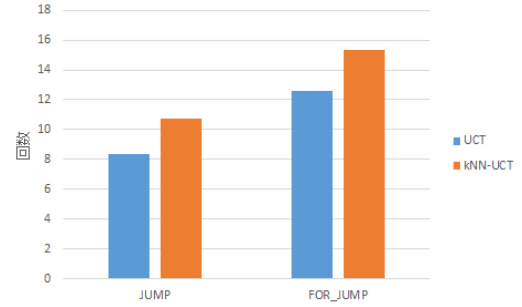


図 2. JUMP 行動の平均実行回数

表 1. UCTAI に対する 1 試合間の平均 JUMP 行動実行回数とスコア

被験者	JUMP 行動回数	スコア
A	16.2	262
B	40.8	413
C	8.4	325

表 2.  $kNN$ -UCTAI に対する 1 試合間の平均 JUMP 行動実行回数とスコア

被験者	JUMP 行動回数	スコア
A	14.4	308
B	54.6	553
C	9.2	307

また、3 人の被験者における UCTAI と  $kNN$ -UCTAI に対する、1 試合間の平均 JUMP 行動回数とそのときのスコアを、それぞれ表 1,2 に示す。被験者 B は被験者 A,C に比べて 1 試合間に JUMP 行動を多用しており、それによって A,C よりも獲得スコアが大きくなっている。これは、キーボードでの対戦の際に、被験者 B が両 AI に対して JUMP 行動が有効であると知り、Kinect 操作でも同じような手法を取ったからだと推測される。

### 5. おわりに

本論文では、Kinect を用いた対戦格闘ゲームにおける、ユーザの健康促進のための AI について述べた。実験の結果、UCT と  $k$  近傍法を適用した  $kNN$ -UCTAI は、UCT のみを適用した UCTAI よりもユーザに多くの JUMP 行動をさせており、被験者に対して  $kNN$  が機能していることが示された。今後の研究では、JUMP 行動以外の行動についても着目し、ユーザの運動量を向上させるための要素についてさらに深く探究、分析していく予定である。

#### 参考文献

- [1] K. Moriyama et al., "An Intelligent Fighting Videogame Opponent Adapting to Behavior Patterns of the User", IEICE TRANS. INF. & SYST., VOL.E97-D, NO.4, pp842-851, April, 2014.
- [2] P. Paliyawan, K. Sookhanaphibarn, W. Choensawat and R. Thawonmas, "Body Motion Design and Analysis for Fighting Game Interface," accepted for presentation at the 2015 IEEE Conference on Computational Intelligence and Games (CIG 2015), Tainan, Taiwan, Aug.31 - Sep. 2, 2015.
- [3] <http://www.ice.ci.ritsumeai.ac.jp/~ftgaic/> (2015/7/22 参照)
- [4] L. Kocsis and C. Szepesvari. "Bandit Based Monte-Carlo Planning", ECML, pp. 282-293, Berlin, Germany, September, 2006.
- [5] K. Yamamoto, S. Mizuno, C. Yin Chu, and R. Thawonmas, "Deduction of Fighting-Game Countermeasures Using the k-Nearest Neighbor Algorithm and a Game Simulator", 2014 IEEE Conference on Computational Intelligence and Games (CIG 2014), Aug.26-29, 2014.