深層強化学習による人間補助を行う格闘ゲーム AI の作成

山本　拓実† 　清　雄一† 　田原　康之† 　大須賀昭彦†

† 電気通信大学情報理工学域　〒 182-8585 東京都調布市調布ヶ丘

E-mail: †yamamoto.takumi@ohsuga.lab.uec.ac.jp, {seiuny, tahara, ohsuga}@uec.ac.jp

あらまし　対戦型ゲームにおいて, ゲームが上手くない人は負けやすくなってしまう. それによって, 何度も対戦に負けてしまい, ゲームに楽しみを見いだせず, そのゲームから離れてしまう人もいる. 解決策としては, 対戦相手に合わせて手加減をするゲーム AI が挙げられるが, 全ての場面で自然な動きができるとは言えず, ゲーム状況にそぐわない動きをしたり, プレイヤーが明らかな手抜きを感じてしまったり等の問題が生じる. 本研究では, その問題の解決策として, プレイヤーの操作を部分的に補助する深層強化学習によるゲーム AI を提案する. 実験環境は FightingICE を取り上げた. 被験者に完成した AI を用いた時と用いなかった時で残り体力, スコアの比較を行った所, 平均的には作成した AI を使った方が体力, スコアが上がる傾向が見られた. 人間による主観評価では, 楽しさの観点では, サポート AI を使った時の方がゲームをより楽しめる傾向にあるという結果となった.

キーワード　深層強化学習, 格闘ゲーム, コンピュータゲーム, 人と AI の協力

1　はじめに

1.1　背　　景

近年, 深層強化学習技術を現実世界に適用させるという試みが盛んに見られる [1]. しかし, 現実世界は時間的に連続で, 複雑な空間であることから, AI が安全に正しく動作しづらいという問題が挙げられる. 深層強化学習エージェントが複雑な状況であるゲーム内で目的を果たすことができれば, 現実世界への深層強化学習の適用に貢献できることから, 深層強化学習を用いたゲーム AI の研究が盛んである. 特に, 人間すらも凌駕する強いゲーム AI の研究に力が注がれている. 比較的単純なゲームである Atari2600 [2] の内, 強化学習のベンチマークとされている 57 個のゲームでは, Deep Mind 社によって開発された MuZero [3] や Agent57 [4] が超人的なスコアを獲得するという成果を挙げている. また, 格闘ゲームのような複雑なゲームにおける AI の研究も行われている [5].

ゲーム AI の開発では, 強さ以外の要素にも注目され始めており, 人に教える, 人を楽しませる, 人と協力する等のタスクが扱われ始めている [6] [7]. 例えば, Shi らによって, プレイヤーの勝率に合わせて手加減をし, 人を楽しませる深層強化学習を用いた囲碁 AI の研究がされた事例がある [8].

対戦型ゲームにおいて, ゲームが上手くない人は負けやすくなってしまう. それによって, 何度も対戦に負けてしまい, 楽しみを見いだせず, そのゲームから離れてしまうという人もいる. 元々人を楽しませることが目的の一つであるゲームに楽しみが見いだせなくなってしまうのは好ましくない. 解決策には, 上記に記した手加減をしてくれるような AI を用いることが挙げられる. しかし, Shi らによると, 手加減をしてくれる AI にもゲーム状況にそぐわない不自然な動き, 明らかな手抜きを感じる等の問題が生じる時もあると示されている. また, 対戦相手を手加減をする AI で固定していなければならず, プレイヤーが飽きてしまう可能性も考えられる.

1.2　目　　的

本研究では, 対戦相手の強さを変えずに AI の支援によってプレイヤーが勝利に近づくこと, プレイヤーを支援する AI を用いた時にプレイヤーが手加減 AI との対戦よりも楽しんでもらうことを目的とする.

対戦型ゲームには, 研究用格闘ゲームである FightingICE [9] を取り上げ, プレイヤーの操作を部分的に補助する深層強化学習による AI を導入する. この研究が成し遂げられれば, ゲームが不得意な人が, プロのプレイヤーと張り合える場面や知らなかった行動を AI が行ってくれることで新たなプレイスタイルを知る場面が生まれ, ゲームの楽しさをさらに感じさせることができる可能性もある. また, 格闘ゲームの敗因が体力 (HP) の減少であることから, AI には防御に着目して学習を行わせる.

1.3　FightingICE

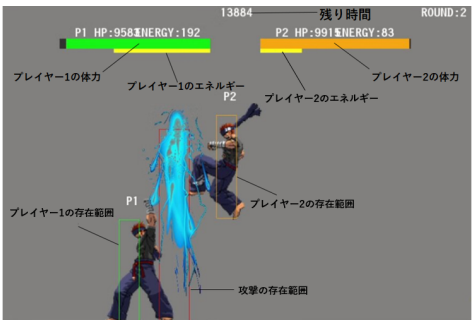

図 1　FightingICE のゲーム画面

FightingICE は, 立命館大学理能コンピュータエンタテインメント研究室が開発した研究用の格闘ゲームである. FightingICE

のゲーム画面を図 1 に示す．このゲームは，1 ラウンドで，60 秒間に，プレイヤー 1，2 が自身の HP を維持しつつ，相手の HP を減らすために，定められた 40 の行動を駆使して戦う 2D 対戦型格闘ゲーム環境である．プレイヤーの HP は自然数で設定することができる．また，いくつかの攻撃は，エネルギーをいくらか消費しなければならない．攻撃を実行すると，攻撃毎に定められた攻撃範囲が発生し，攻撃範囲内に相手の存在範囲が重なると，その相手にダメージを与える事ができる．最終的には，1 ラウンド終了時までに，相手プレイヤーよりも多くの HP を持つあるいは相手の HP を 0 にする事でプレイヤーは勝利することができる．エージェントが行動決定をする際に，ゲーム側からエージェントへ 0 から 1 の値で表現されたゲーム状況が送りだされ，それを基にエージェントは行動決定を行う．ゲーム状況は，両プレイヤーの HP，エネルギー量，座標，状態，スピード，遠距離攻撃で出現する物体の座標等から構成されている．

2 関 連 研 究

2.1 Deep Q Network

強化学習の手法の一つに Q 学習 [10] が挙げられる．Q 学習では，ゲーム状況に対応する Q 値という各行動の価値が格納されている Q-Table を用いて，エージェントが行動決定を行い，実行した行動から得られた報酬で Q 値を更新していく．時刻 t における状況 s_t で，行動 a_t を行った際の Q 値を $Q(s_t, a_t)$ とする．また，エージェントが状況 s_t の際に，行動 a_t をとり，報酬 r_t が得られたとする．時刻 t の次の時刻である時刻 $t+1$ の状況 s_{t+1} における全ての行動集合 a に対する Q 値 $Q(s_{t+1}, a)$ を用いて，$Q(s_t, a_t)$ は式 (1) で更新される．α，γ は学習率，割引率という 0 から 1 の定数である．即時報酬に注目して学習を行う際は，α の値を大きくすれば良い．将来の報酬に注目して学習を行う際は，γ の値を大きくすれば良い．

$$Q(s_t, a_t) \leftarrow Q\ (s_t, a_t) + \alpha(r_t + \gamma \underset{a}{max} Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (1)$$

Q 学習をリアルタイムゲームのように連続的で複雑なゲームに用いると，考えられる状況が多すぎてしまい，Q-Table が数多く存在するゲーム状況に対応しきれなくなってしまう．Mnih らは，Q-Table を Deep Neural Network(DNN) で近似した手法である Deep Q Network(DQN) [11] を提案した．DQN では，DNN にゲーム状況を入力し，出力された全行動の Q 値を用いてエージェントが行動決定する．そこで，得られた報酬を用いて，Q 学習由来の以下の損失関数 L で DNN を学習していき，DNN を Q-Table に近似する．

$$L = \mathrm{E}[(r_t + \gamma \underset{a}{max} Q(s_{t+1}, a) - Q(s_t, a_t))^2] \quad (2)$$

式 (2) より，DQN の学習には，ゲームから得られる (s_t, a_t, r_t, s_{t+1}) のデータが必要となる．ゲームの進行によって入手できるこれらのデータを使って順に学習をしていくとすると，最後に与えられたデータを重点にして学習を行ってしまうことから，古いデータに対しての学習が疎かになってしまう．そこで，Mnih らは DQN に Experience Replay [12] を用いた．Experience Replay では，最新のデータを学習に使用するのではなく，ランダムにいくつか取り出した過去のデータを DNN に学習させることで，学習データの偏りを無くし，学習データを効率的に使用する．また，Mnih らは，DQN の学習時に，ゲーム内で実際に行動するエージェントの DNN と同構造のターゲットネットワークを用いた．これによって，DNN の学習によって変動してしまう Q 値に対応して学習することができる．

図 2 が DQN の学習の流れである．DNN から出力される Q 値に則って，エージェントがゲーム内で行動を行い，得られたデータをバッファに保存する．エージェントが行動を行う度に，バッファ内からランダムにミニバッチというデータを取り出し，ミニバッチ学習を行う．定められた回数の学習が行われると，ゲーム内で実際に行動するエージェントの DNN のパラメーターがターゲットネットワークのパラメーターに共有される．また，学習時の行動決定時には，エージェントに様々なデータを集めさせる為に，ε-greedy 法を用いることもある．ε-greedy 法は，確率 ε でエージェントにランダムな行動をさせて，より効率的に探索を行わせる方法である．

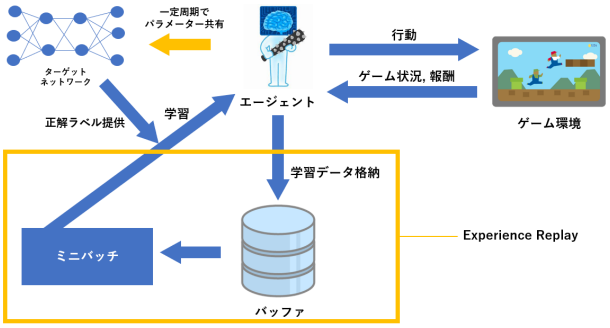

図 2　DQN の学習の流れ

2.2 Hybrid Reward Architecture

DQN では，1 つの出力部分に対して得られた報酬で学習を行っていた．Caruana によって，下層を共有し，いくつかの種類に対する予測を出力し，学習を同時に行うマルチタスク学習 [13] が提案され，様々な課題を別々に学習するよりも，同時に学習する方が，DNN の性能が良くなるという可能性を示した．Seijen らによって，報酬を何種類かに分け，それら全てに対してマルチタスク学習を行う DQN である Hybrid Reward Architecture(HRA) が提案された [14]．報酬を 2 種類考慮する場合，得られた 2 種類の報酬とそれぞれのヘッドで出力された Q 値を用いて式 (2) で学習していく．また，出力層ではそれぞれのヘッドに固有の重みをかけて足された値が出力される．状況 s において行動 a を選ぶ際に出力される Q 値 $Q(s, a)$ は，i 番目のヘッドの Q 値 $Q_i(s, a)$ と i 番目のヘッドの重み w_i を用いると以下のようになる．

$$Q(s, a) = \sum_{i=1}^{2} w_i Q_i(s, a) \quad (3)$$

Takano らは，攻撃用，防御用に分けた 2 種の報酬を用いた HRA によるダブルヘッド DQN を FightingICE に適応させた [15]．さらにダブルヘッド DQN と通常の DQN による AI を，2017 年の FightingICE の AI コンペティション出場 AI 等

13 種と比較すると, 通常の DQN の AI のコンペティション基準の評価が 6 位だったことに対し, ダブルヘッド DQN の AI は 4 位という好成績を残した. このことから FightingICE において HRA による AI が通常の DQN の AI よりも高性能であることが示された.

2.3 人と協力するゲーム AI

近年における, 人と協力するゲーム AI の研究は, Overcooked [16] のようなマルチプレイヤーゲームで行われている [7]. Strouse らの研究では, これらのエージェントの学習法として Self-Play(SP), Population-Play(PP), Behavioral cloning play(BCP), Fictitious Co-Play(FCP) がマルチエージェントの学習法として紹介された [17]. SP は, 学習エージェントのコピーを他プレイヤーとして使用し, エージェントを学習させていく方法である. PP は, 異なる方策を持つエージェントに並列で学習を行わせ, データを共有して学習していき, それらのエージェントを方策に基づいて選択する多様的なエージェントを作り上げる方法である. BCP は, 人間のプレイデータを用いて, 教師有り学習で模倣学習された AI を他プレイヤーとして用いることで, 強化学習エージェントを学習させる方法である. FCP は, いくつかの SP エージェントに独立に学習を行わせ, それぞれ学習内で, 協力すべき他エージェントに最適な動きができるように学習させ, 集めたデータでエージェントの学習を行わせる方法である.

このように, 人間と協力するゲーム AI の研究はマルチプレイヤーゲームで, 様々な学習法が提案されてきた. しかし, マルチプレイヤーゲームではなく, プレイヤーが操作するキャラクターに直接協力を行うゲーム AI に着目されてきたことは殆どなく, 特に格闘ゲーム AI に関しては, 調べた限りでは行われていない. また, 人間と AI が扱える行動が異なるので, マルチプレイヤーゲームにおけるゲーム AI とは異なる方法で AI を学習させなければならない.

3 提案手法

3.1 システムの概要

サポート AI がプレイヤーの操作に介入する際のシステムの概要を図 3 に示す.

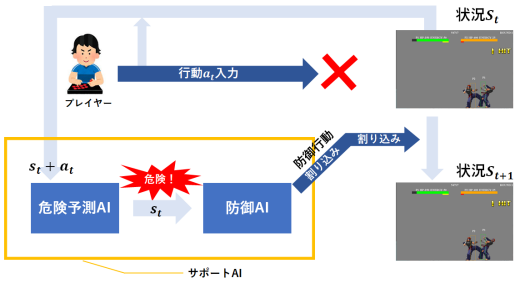

図 3 システムの全体構成

本研究では, 問題解決のために 2 つの AI を導入し, これらをまとめてサポート AI とする. 1 つ目は, プレイヤーの操作に介入して代わりの操作を行う AI である. 2 つ目は, 防御 AI へプレイヤーの操作に介入するタイミングを通知する AI である. 本研究ではこれらをそれぞれ防御 AI, 危険予測 AI とする.

時刻 t においてプレイヤーが行動 a_t を入力した際に, 危険予測 AI が状況 s_t と a_t を入力として受け取る. それらの入力から, その状況がプレイヤーにとって危険と判断された場合は, 危険予測 AI が防御 AI へ通知を行う. その後, 防御 AI が入力としてゲーム状況 s_t を受け取り, 最適な行動を選択した後, プレイヤーの操作に介入する.

3.2 防御 AI

3.2.1 報酬設計

防御 AI はできるだけ攻撃を受けないような行動を学習しなければならない. よって, 1 つ目の報酬 R_{HIT} を以下のように設定した.

$$R_{HIT} = \begin{cases} -100 & (\text{ダメージを受けた時}) \\ 0 & (\text{ダメージを受けなかった時}) \end{cases} \tag{4}$$

また, 相手キャラクターに接近された際に, プレイヤーが操作するキャラクターにダメージが入りやすいことを考えると, できるだけ相手との距離が離れるような行動を学習することも重要であると考えた. よって, 2 つ目の報酬 $R_{distance}$ を以下のように定めた. d は, 相手キャラクターとプレイヤーのキャラクター間の距離である. また, d を計算するために用いる座標は, 0 から 1 の間の間で表現されているものを用いる.

$$R_{distance} = -\frac{1}{0.01 + d} \tag{5}$$

3.2.2 ネットワークアーキテクチャ

3.2.1 項で示した 2 種の報酬 R_{HIT} と $R_{distance}$ で防御 AI の学習をさせていく. よって, 本研究では, 防御 AI に出力部が 2 つの HRA によるダブルヘッド DQN を使用する. R_{HIT} と $R_{distance}$ で学習するヘッドをそれぞれダメージヘッド, 距離ヘッドとする.

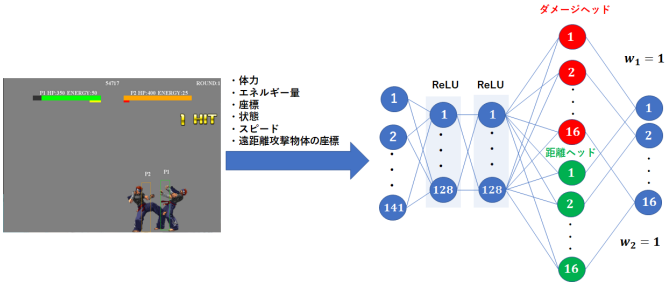

図 4 防御 AI の DNN の概要図

この DNN は出力層以外は全結合層で構成されており, 1.3 節で示した 141 個の情報を入力とする. また, 第 2 層, 第 3 層では, 活性化関数に ReLU 関数を用いた, ユニット数が 128 である全結合層を用いている. 第 3 層は, ダメージヘッド, 距離ヘッドとそれぞれ全結合している. それぞれのヘッドのユニット数は共に 16 である. また, それぞれのヘッド固有の重みは 1 とした. 本研究では, プレイヤーが実行できる行動を 16 種類とした.

防御 AI が扱える行動も同様にそれらの 16 種とするため, ヘッド部分, 出力層のユニット数も 16 とした.

3.3 危険予測 AI

3.3.1 報酬設計

危険予測 AI も, プレイヤーの HP を削られないような行動を学習しなければならない. しかし, 防御 AI と同じように R_{HIT} を報酬として設定してしまうと, 総合的に大きなダメージを受けない攻撃に対しても反応してしまい, プレイヤーの操作に介入しすぎてしまうことも考えられる. よって, 危険予測 AI の 1 つ目の報酬として, 受けたダメージを報酬とした以下の $R_{defence}$ を用いる. $HP_t^{self}, HP_{t+1}^{self}$ はそれぞれ報酬計算前と報酬計算時のプレイヤーの HP であり, 0 から 120 の間で表現される.

$$R_{defence} = HP_{t+1}^{self} - HP_t^{self} \tag{6}$$

$R_{defence}$ の報酬のみを指針として学習すると, 将来の報酬を最大化するために, 防御 AI に全ての操作を委ねるという事態が発生してしまう. よって, プレイヤーの操作を評価するために, 以下の報酬 R_{attack} を新たに設定する. $HP_t^{opp}, HP_{t+1}^{opp}$ はそれぞれ報酬計算前と報酬計算時の相手プレイヤーの HP であり, 0 から 120 の間で表現される.

$$R_{attack} = HP_t^{opp} - HP_{t+1}^{opp} \tag{7}$$

3.3.2 ネットワークアーキテクチャ

3.3.1 項で示した 2 種の報酬 $R_{defence}$ と R_{attack} で危険予測 AI の学習をさせていく. 危険予測 AI でも, 防御 AI と同様に HRA によるダブルヘッド DQN を使用する. $R_{defence}$ と R_{attack} で学習するヘッドをそれぞれ防御ヘッド, 攻撃ヘッドとする. また, 危険予測 AI において, HRA によるダブルヘッド DQN を使用する意図は, 性能向上のみならず, $R_{defence}, R_{attack}$ の Q-Table を近似した後, プレイヤーが操作しやすいように出力層の前の重みを調節するためである.

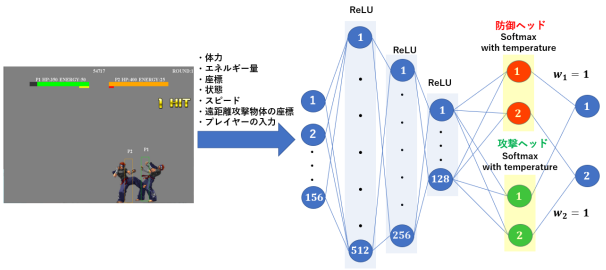

図 5 危険予測 AI の DNN の概要図

入力層には, 防御 AI の入力で用いた 141 個の情報に加え, プレイヤーの 16 種類の行動を 0, 1 で表現している 15 個の情報も追加で入力される. よって, 入力層のユニット数は 156 個である. それ以降は第 4 層まで全結合層となっており, ユニット数は第 2 層から順に, 512, 256, 128 である. 活性化関数には, ReLU 関数を用いている. また, 第 4 層は, 防御ヘッド, 攻撃ヘッドそれぞれに全結合している. 両ヘッドのユニット数は 2 であり, 行動は「防御 AI に行動を委ねる」, 「プレイヤーの入力を実行する」の 2 種類である. 各ヘッドにおいて Q 値の大きさに大きな差が生まれてしまうことから, 正則化のために活性化関数にソフトマックス関数を用いる. しかし, 防御ヘッドの Q 値が負の値となることから, 通常のソフトマックス関数では, 2 つのユニットの出力にほとんど差が無くなってしまう. よって, 本研究では, 第 5 層の両ヘッドの活性化関数に温度付きソフトマックス関数を用いる. それぞれのヘッド固有の重みは 1 とした.

4 実　　験

4.1 実験準備

4.1.1 FightingICE の HP 設定

FightingICE の AI コンペティション公式ルールでは, キャラクターの最大 HP は 400 と定められている. しかし, HP の推移をより確認しやすくするために, 実験時の FightingICE におけるキャラクターの最大 HP を全て 500 に統一した. 実験時の使用キャラクターは, ZEN に固定した.

4.1.2 FightingICE におけるキャラクターの操作

FightingICE は, AI を動かすための格闘ゲーム環境である. よって, そのままでは, プレイヤーが FightingICE のキャラクター操作を行うことができない. まず, キーボード入力で, キャラクターを動かせるように FightingICE の改変を行った. また, キーボード操作でキャラクターを満足に操作するのは難しい. よって, ゲームコントローラーからの入力をキーボード入力に変換することで, プレイヤーが FightingICE のキャラクターを操作できるように FightingICE の改変を行った. 使用したゲームコントローラーは, DUALSHOCK 3 [18] である.

4.2 実験 1:防御 AI の学習

ハイパーパラメータは表 1 のように定めた.

表 1 防御 AI のハイパーパラメーター

ハイパーパラメータ	値
ミニバッチサイズ	32
リプレイバッファサイズ	50000
ターゲットネットワークの更新頻度	300
割引率	0.9
学習率	0.001

学習時には, 防御 AI に ε-greedy 法による行動決定を行わせた. ε は, 13000 回の行動決定で, 1 から 0.1 になるように線形に変形させた. 対戦相手には, 2015 年 FightingICE の AI コンペティション優勝 AI である Machete を採用した. エネルギーを使わなければならない行動を実行する際に, エネルギーが足りないと, 何も実行できない. また, その際にダメージを受けてしまうと, これらの技に対する Q 値が下がってしまう. よって, これらの行動を選択時に, エネルギーが足りない場合は, Q 値を極めて小さい値とすることで, 無駄に実行することが無いようにした. 以上の条件の基で, 防御 AI を 300 エピソード学習させた. また, HRA の有効性の確認のため, 同じハイパーパラメーター, 報酬でシングルヘッド DQN でも同様の実験を行っ

た．学習時の HP の直近 30 エピソード分の平均値を取ったグラフを図 6 に示す．

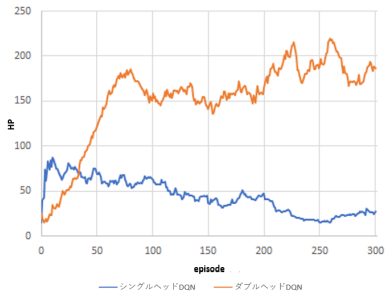

図 6　シングルヘッド, ダブルヘッド DQN の HP の推移

4.3　実験 2:危険予測 AI の学習

ハイパーパラメーターは表 4.3 のように定めた．

表 2　危険予測 AI が学習時のハイパーパラメーター

ハイパーパラメーター	値
ミニバッチサイズ	64
リプレイバッファサイズ	50000
ターゲットネットワークの更新頻度	300
割引率	0.5
学習率	0.001
ソフトマックス関数の温度	0.2

4.2 節同様に, 学習時には ε-greedy 法による行動決定を行わせ, 対戦相手は Machete とした．また, 危険時に危険予測 AI が通知を送る相手には, 4.2 節で学習させた防御 AI を採用した．

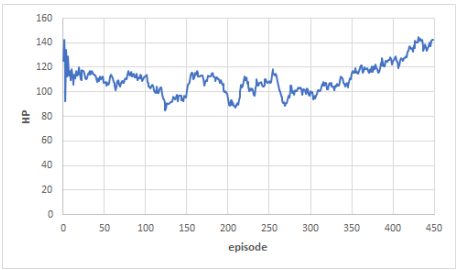

図 7　危険予測 AI 学習時の HP の推移

以上の条件の基で, 危険予測 AI を 300 エピソード学習させた．学習時の直近 45 エピソード分の HP の平均値を取ったグラフを図 7 に示す．

学習後, プレイヤーが快適に操作できるように, ヘッドから出力層への重み, ソフトマックス関数の温度の調整を行った．それらの値は表 3 のように設定した．

表 3　危険予測 AI のヘッドの重みとソフトマックス関数の温度

ハイパーパラメーター	値
防御ヘッドの重み	0.25
防御ヘッドのソフトマックス関数の温度	0.4
攻撃ヘッドの重み	0.75
攻撃ヘッドのソフトマックス関数の温度	10

4.4　実験 3:被験者による実験

4.4.1　実験内容

20 代の男女 15 人の被験者にサポート AI を用いた実験を行い, HP や式 (8) によるスコアの比較で行う定量評価とアンケートによる主観評価を行った．スコアは対戦終了時の自身の HPHP_{self} と相手の HPHP_{opp} を用いて式 (8) で計算される．

$$score = \frac{HP_{self}}{HP_{self} + HP_{opp}} \times 1000 \qquad (8)$$

アンケートの内容は以下のようにした．

「普段の格闘ゲームプレイ時間について」

- 大乱闘スマッシュブラザーズ, 鉄拳, ストリートファイター等の格闘ゲームを何時間ほど行ったことがあるか．

「対戦について」

(a) プレイ中どの程度サポート AI に助けてもらったと感じられたか．

(b) プレイ中どの程度サポート AI の介入を邪魔だと感じなかったか．

(c) サポート AI 有りと無しでは, どちらが楽しめたか．

「手加減 AI との比較」

- 手加減 AI があったとする．対戦相手の強さに合わせ試合中動的に手加減 AI の強さが変化する．しかし,「プレイヤーが有利になると手加減 AI が急に強くなる」,「プレイヤーが不利になりすぎると手加減 AI の攻撃や防御頻度が急に少なくなる」こともあるとする．この時

①この手加減 AI にサポート AI 無しで戦った時

②敵の強さは変えずに今回のサポート AI を使って戦った時

はどちらが楽しいと思えるか (点数が高い程, ②を肯定しているとする.).

「手加減 AI の比較」における手加減 AI のデメリット 2 点は, 池田の研究で挙げられていた手加減 AI の課題 [6] を参考に設定した．「対戦について」と「手加減 AI との比較」は 5 段階評価で評価してもらい, 値が大きいほど, サポート AI が良いという結果になるように設定した．また,「対戦について」と「手加減 AI の比較」については印象的な場面・理由等もあれば被験者に記述してもらった．

実験は**「普段の格闘ゲームプレイ時間について」**を尋ねた後に, FightingICE に搭載されている AI である BCP と Machete のそれぞれに対してサポート AI を用いないで 9 戦, サポート AI を用いて 9 戦を行ってもらった．対戦相手が変わるごとに**「対戦について」**を尋ね, 最後に**「手加減 AI との比較」**について尋ねた．

4.4.2　実験結果

被験者の格闘ゲームプレイ時間

アンケートの結果, 本実験で集まった被験者の格闘ゲームプレイ時間の割り当ては, 0 時間以上 50 時間未満の人数が 9 人, 50 時間以上 500 時間未満の人数が 4 人, 500 時間以上の人数が 2 人であった．本実験ではそれぞれのプレイ時間に該当するグ

ループをを初心者, 中級者, 上級者とする.

BCP との対戦における定量評価の結果

被験者毎の BCP と対戦した際の平均 HP, 平均スコア , 勝利数, を図 8, 図 9, 図 10 に示す.

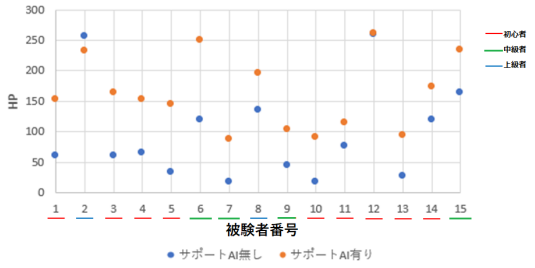

図 8　BCP と対戦した際の被験者毎の平均 HP

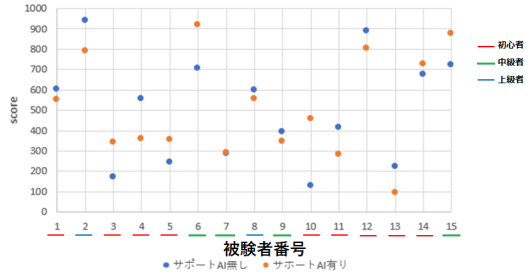

図 9　BCP と対戦した際の被験者毎の平均スコア

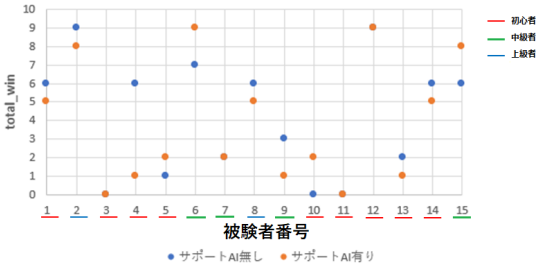

図 10　BCP と対戦した際の勝利数

また, BCP との対戦におけるサポート AI を使った時と使っていない時のそれぞれの全被験者の HP, スコアの平均, 合計勝利数を表 4 に示す. 平均 HP, 平均スコアは, 小数点以下を切り捨てた値である.

表 4　BCP との対戦における全被験者の測定結果

	平均 HP	平均スコア	勝利数 (回)
サポート AI 無し	98	504	63
サポート AI 有り	164	517	58

BCP との対戦における主観評価の結果

図 11 に BCP との対戦について被験者に尋ねたアンケートの結果を箱ひげ図で示す. 質問 (a) の第一, 第二, 第三四分位数は全て 4 である. 質問 (b) に関しては, 第二, 第三四分位数が共に 4 である.

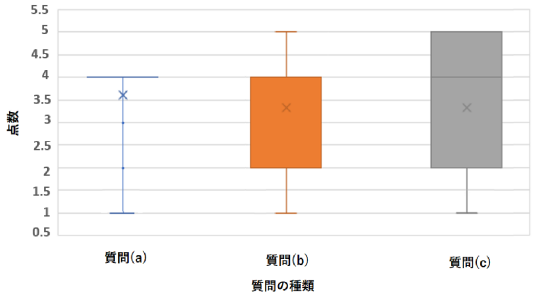

図 11　BCP との対戦実験におけるアンケート結果

Machete との対戦における定量評価の結果

Machete と対戦した際の体力とスコアの結果を図 12, 図 13, 図 14 に示す.

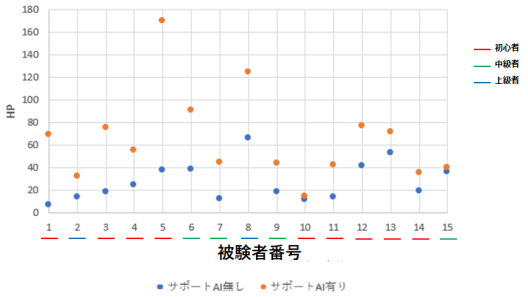

図 12　Machete と対戦した際の被験者毎の平均 HP

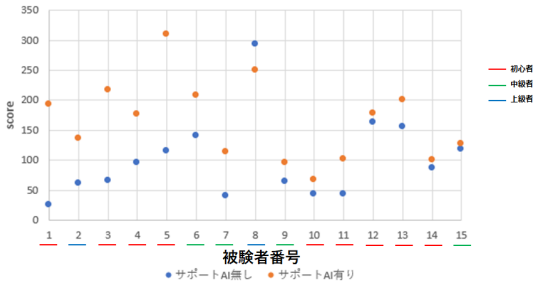

図 13　Machete と対戦した際の被験者毎の平均スコア

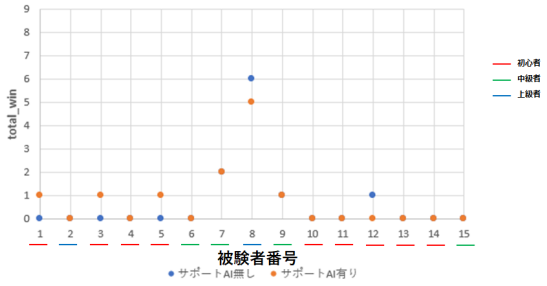

図 14　Machete と対戦した際の勝利数

また, Machete との対戦における表 5 にサポート AI を使った時と使っていない時それぞれの全被験者の体力, スコアの平均, 合計勝利数を示す. 平均値は, 小数点以下を切り捨てた値である.

表 5　Machete との対戦における全被験者の測定結果

	平均 HP	平均スコア	勝利数 (回)
サポート AI 無し	27	101	10
サポート AI 有り	66	165	11

Machete との対戦における主観評価の結果

図 15 に Machete との対戦について被験者に尋ねたアンケートの結果を箱ひげ図で示す．質問 (a) の第一, 第二, 第三四分位数は全て 4 である．質問 (b) に関しては, 第二, 第三四分位数が共に 4 である．

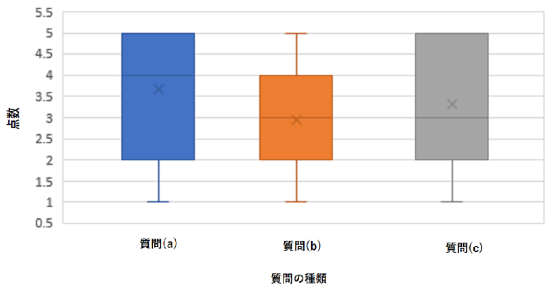

図 15　Machete との対戦実験におけるアンケート結果

「手加減 AI との比較」のアンケートについて

アンケート内の「手加減 AI との比較」のアンケート結果を図 16 に箱ひげ図で示す．

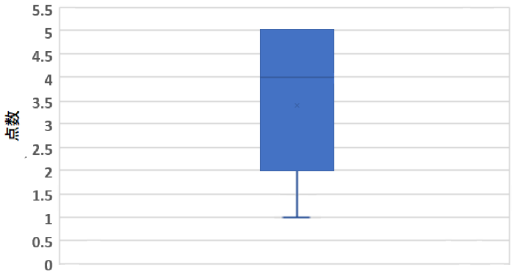

図 16　「手加減 AI との比較」のアンケート結果の箱ひげ図

主観評価で挙げられた質問ごとの印象点

BCP, Machete 両方において, 挙げられた印象についてはおおよそ似たような意見が集められた．以下によく見られた意見を示す．アンケートの「対戦について」では, BCP, Machete を使った両実験において似た記述が見られた．よって, 以下にまとめて表記する．特に, サポート AI に肯定的な意見を質問 (a), 否定的な意見を質問 (b) に記してもらった．また, エネルギーを使わなければならない比較的強力な攻撃を必殺技と表記することとする．

- 質問 (a)
  - 壁にハメられた時に脱出するのを助けてもらった．
  - 自動的にガードや回避をしてもらった場面があった．
  - 自動で距離を取ってもらい, 一息付けた．
  - 追加で攻撃をしてくれた
  - 必殺技を使い忘れていたが, サポート AI が使ってくれて思い出せた．
- 質問 (b)
  - 入力を受け付けてくれない (特に必要な場面で必殺技が出せない)
  - 進みたい方向と逆方向にジャンプや移動をさせられた
  - 必殺技を勝手に使われた
- 質問 (c)
  - 勝てなかった相手にギリギリ勝てそうな試合が増えて楽しかった．

アンケートの「手加減 AI との比較」では, 以下の様な意見が見られた．サポート AI に肯定的な意見を P.O. , 否定的な意見を N.O. と区別した．

- P.O.
  - 手加減されることがそもそも好きではない．
  - 手加減してくれる敵は見慣れている．自分をサポートしてくれる AI は, 新鮮である．
  - 敵の強さが途中で変わるとやりづらい気がする
- N.O.
  - 入力を受け付けてくれないことがかなりストレス

5　考　　察

図 6 より, ダブルヘッド DQN は, 最大体力の 4 割程とシングルヘッド DQN よりも多くの体力を維持できるようになり, 相手との距離を離すジャンプやバックステップ等の行動が頻繁に見られた．また, 4.4.2 項の**主観評価で挙げられた印象的な場面・理由**では, 相手との距離を離すという行動が壁に閉じ込められてしまうという現象から脱出できたり, 相手と距離をとれてプレイヤーが落ち着けたりと, それらの行動がプレイヤーを助ける場面があったことが分かる．このことから防御に特化したエージェントの作成はできたと考えられる．しかし, 不必要に必殺技を使われたという意見もあった事から, 不必要な必殺技の使用を防ぐべく, エネルギー量に対するペナルティを課す必要があった可能性がある．

図 7 より, 危険予測 AI の学習では, 体力に増加は見られたものの, わずか 40 程の増加であったことから, 正確に危険を予測できているとは言い難いと考えられる．これは, 防御 AI が取る行動によって, 更新される Q 値が一定ではないからと考えられる．また, ヘッドの重みやソフトマックス関数の温度の調整時には, 大きく値が変わってしまったことから, 今回の報酬設計では, 操作の快適さを求めるのも難しいと考えられる．特に**主観評価で挙げられた印象的な場面・理由**では, 必殺技使用時にプレイヤーの入力が実行できないという意見があったことから, エネルギー量に対するヘッドを追加し, 必殺技をより快適できるようにすべきであったと考えられる．また, 操作性の問題に関しては, 人間はランダムプレイヤーのような連続的な入力を行わない事から, 学習時にランダムプレイヤーを用いたことも関係していると考えられる．

表 4, 表 5 から, 被験者実験の定量評価では, BCP 戦におけるサポート AI 使用時のスコアの上昇度は, Machete 戦の時よりも小さいという結果となった．これは, サポート AI が, 学習

時に Machete との対戦しか行っていないことが原因と考えられる. 逆に Machete 戦においては, サポート AI 学習時の相手であったことから, サポート AI を用いた時のスコアの増加度が BCP と戦った時よりも増加している. また, 図 9 から, 被験者の半数以上のスコアが下がってしまうという結果になってしまった. 特に, サポート AI を使わなかった時にスコアが 400 点以上であり, サポート AI を使った時にスコアが下がった被験者は, 被験者番号 1, 2, 4, 8, 9, 11, 12 番の 7 人もいた. これは, BCP に対しての有効策を早めに理解してしまったことが原因と考えられる. その有効策が, 防御 AI の学習している行動と異なっている場合, サポート AI の介入が逆に被験者の行動の足を引っ張ってしまったのではないのかと考えられる.

図 11 より, BCP 戦において「対戦について」の全ての質問の中央値が 3 点よりも上回った. スコアはほとんど変わっていないものの, このような結果となったのは, 被験者がサポート AI を用いた時の方の HP がより多く残るようになったことが理由と考える. 図 15 から, Machete 戦において, 被験者が BCP の時よりは, サポート AI の動作を邪魔だと感じていることに関しては, Machete が BCP よりも攻撃的な AI であることが原因と考える. 被験者の BCP と Machete に対する勝利数の比較から, Machete の方が BCP より強いことは明らかである. このことから, BCP の時よりも Machete と戦っている時の方がプレイヤーが危機に瀕する場面が多いと考えられ, これによって, サポート AI の介入が BCP の時よりも増加するのではと考えられる. これは, 危険予測 AI がより正確に危険な場面を判断して介入するタイミングを減らすことや防御 AI がより適切な行動を選択することでより快適にサポート AI を使用してもらえるのではと考える.

主観評価で挙げられた印象的な場面・理由から, プレイヤーの入力をサポート AI が受け付けないことが, 「手加減 AI との比較」の評価を下げてしまっていることが分かる. しかし, 「手加減 AI との比較」のアンケート結果において, 中央値と平均値が 3 点以上であることから, この欠点が解決されれば, 手加減 AI と戦うよりもサポート AI を用いた方が楽しくゲームをできる人が過半数になるのではと考えることができる.

6 おわりに

本研究では, 研究用 2D 対戦型格闘ゲームである FightingICE を用いて, 人間のプレイを支援する深層強化学習エージェントの作成を試みた. 結果として, 学習時に対戦した Machete とプレイヤーが戦う際には, プレイヤーの平均 HP やスコアを上げることができた. しかし, プレイヤーが思った通りの動きができない時があるという問題が生じた. また, 学習に用いていない BCP との対戦に関しては, 主観評価で Machete 対戦時よりも高い評価を得てはいたものの, スコアがほとんど変わらなかったことから, 上手くプレイヤーを支援していたとは言い難い. しかし, 平均的には, 被験者はサポート AI に助けてもらっていると感じており, サポート AI を用いている時の方が, サポート AI を用いていない時よりも楽しくゲームができているということが分かった. 一方, 手加減 AI との比較については, 手加減 AI よりもサポート AI を用いた方が楽しく格闘ゲームを遊べるのではないかという可能性を示すことができた.

今後の展望としては, プレイヤーが入力した行動から何をしたいかを読み取り, より快適にプレイさせかつサポートできるような AI を作成することが挙げられる. そのために, 学習時のプレイヤーにある方策をもったプレイヤーを用いるべきであると考える. また, 今回の実験では, 学習時にサポート AI が 1 種類の対戦相手としか戦っていないことから, 今後は, 様々な対戦相手で学習を行わせ, サポート AI に普遍的な危機予測とその対処法をを学習してもらう必要があるのではと考える. また, 実際に手加減 AI を用いて比較を行えていないことから, 手加減 AI を導入したサポート AI との比較実験も行っていきたい.

謝辞 本研究は JSPS 科研費 JP21H03496, JP22K12157 の助成を受けたものです.

文 献

[1] 岸川大航, 荒井幸代「深層強化学習による自動運転の安心走行実現」, The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019
[2] Bellemare, M. G. et al., ”The arcade learning environment: an evaluation platform for general agents.” J. Artif. Intell. Res. 47, 253–279 (2013).
[3] Julian Schrittwieser et al., ”Mastering Atari, Go, chess and shogi by planning with a learned model” in Nature, 2020.
[4] Adri‘a Puigdom‘enech Badia et al., ”Agent57: Outperforming the human Atari benchmark”, 2020.
[5] Inseok Oh, Seungeun Rho, Sangbin Moon, Seongho Son, Hyoil Lee, Jinyun Chung, ”Creating Pro-Level AI for a Real-Time Fighting Game Using Deep Reinforcement Learning”, IEEE TRANSACTIONS ON GAMES, VOL. 14, NO. 2, JUNE 2022
[6] 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, 58(3):167-173, 2013-03-01
[7] Mesut Yang, Micah Carroll, Anca Dragan, ”Optimal Behavior Prior: Data-Efficient Human Models for Improved Human-AI Collaboration”, LIACS, 2022.
[8] Shi Yuan, Fan Tianwen, Li Wanxiang, 池田心, 「深層学習囲碁プログラムを用いた場合の手加減に関する研究」, 情報処理学会研究報告. GI, 研究報告ゲーム情報学, 2019-GI-41
[9] http://www.ice.ci.ritsumei.ac.jp/~ftgaic/(参照 2022-12-23)
[10] Christopher JCH Watkins, Peter Dayan. ”Q-learning”. Machine learning, 8(3-4):279–292, 1992.
[11] Volodymyr Mnih et al. , ”Playing Atari with Deep Reinforcement Learning”, NIPS Deep Learning Workshop 2013.
[12] LONG-JI LIN, ”Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching”, Machine Learning, 8, 293-321 (1992)
[13] Richard A. Caruana, ”Multitask Learning: A Knowledge-Based Source of Inductive Bias”, in ICML , 1993-6-27
[14] H. Van Seijen et al. , “Hybrid reward architecture for reinforcement learning, ”in Proc. Advances in Neural Information Processing Systems. 2017. pp. 5392- 5402.
[15] Yoshina Takano et al.,“Applying Hybrid Reward Architecture to a Fighting Game AI ”in IEEE, 2018
[16] Ghost Town Games, Overcooked, 2016. https://ghosttowngames.com/overcooked/ (参照 2022-12-22)
[17] DJ Strouse et al., ”Collaborating with Humans without Human Data”, Advances in Neural Information Processing Systems 34 (NeurIPS 2021), 2021.
[18] Sony,CECH-ZC2, https://www.sony.com/en/SonyInfo/design/gallery/CECH-ZC2/ (参照 2023-1-5)