ゲーム体験向上のための動的難易度調整向け報酬設計

Design of rewards for dynamic difficulty adjustment to improve the game experience

久保田 和市 (Waichi Kubota) *

法政大学情報科学部コンピュータ科学科
waichi.kubota.9x@stu.hosei.ac.jp

Abstract

Modern games have static difficulty levels that create an imbalance between the player's skill and the difficulty of the game. As a result, players cannot achieve flow, a state of sustained enthusiasm. In previous research, the difficulty is dynamically adjusted by switching the learning steps of reinforcement learning, to maintain the balance between player's skill and difficulty. However, it is trained on a single Q-Table and is not suitable for learning games with many states and actions. Therefore, the aim is to improve user experience by adjusting the difficulty using deep reinforcement learning. This paper proposes a reward shaping that uses deep reinforcement learning to achieve dynamic difficulty adjustment so that learning is stable. As a result of implementing this method in 'VizDoom' and experimenting against built-in bots with static difficulties, it was found that deep reinforcement learning with reward shaping can be used to dynamically adjust the difficulties.

1 はじめに

近年のゲームでは、ゲームを始める前に難易度を設定する固定的難易度調整が良く用いられる。固定的難易度調整とは、初級、中級、上級、のシンプルな三択や、より詳細なオプションがある場合はその中から選択し、難易度を設定する。しかし、ゲームによって難易度の数や感じ方の違いがある点や、このような固定的難易度設定では難易度が固定されてしまい、変動しないため、ユーザー体験を損なう可能性がある。根拠として、ユーザー体験に深く関わるゲームの面白さ [1] や、ミハイ・チクセントミハイが提唱したフロー理論 [2] がある。フローとは、タスクの難易度と人のスキルのバランスが良いとき、時間を忘れるほど熱中し集中する状態を指す。ゲームの面白さには複数の要素があるが、フロー理論と合わせると難易度が特に重要になる。また、Penelope Sweetser [3] が述べているように、ゲームの難易度とプレイヤースキルのバランスや、目標の明確さとフィードバックがユーザー体験の重要な要素になる。そこで、ユーザー体験を向上させるため、フロー理論や面白さで重要な難易度に注目する。

この難易度に着目した研究として、Dynamic Difficulty Adjustment (DDA) [4] という難易度を動的に調整するものがある。その中でも今回は、Skilled Experience Catalogue (SEC) [5] という、強化学習 [6] の学習段階を使用した DDA を参考にする。しかし、SEC ではある 1 つの学習のみしか行っていないため他の要素を考慮できない問題や、複数タスクを学習する場合に学習が安定しない問題があった。

本論文では、強化学習手法の一つである Dueling Deep Q-Network [7] を使用し、学習曲線が滑らかに増加する学習を実現するための Reward Shaping を行う。まず、そのゲームで重要なタスクの調査と動的難易度調整で使用できる学習となるような報酬のスケーリングを行った。そして、スケーリングを実装し学習曲線が滑らかに増加するかどうかや、その学習を使用して動的難易度調整が可能かどうかの実験を行った。検証には、コンピューターゲームのジャンルの 1 つである First Person Shooter (FPS) のデスマッチという純粋な撃ち合いを楽しむゲームルールを使用する。難易度は、ゲーム内で対戦相手となる Non-Player Character (NPC) の強さを変更することで調整する。使用環境は、ZDoom という FPS ゲームを元に作られた Python での深層強化学習に適した環境である、VizDoom [8] を使用する。

2 関連研究

動的難易度調整の研究は今までにもされてきた。その 1 つに強化学習の学習段階を使用する手法がある。Skilled Experience Catalogue (SEC) [5] は、強化学習の学習段階を一定区間ごとにポリシーストレージへ保存し、保存した学習段階を閾値を用いて上下させることで相手とのスキルレベルに一致するように調整し、難易度調整を行った。この手法を First Person Shooter の 1 対 1 で戦闘するデスマッチルールに実装し実験

* Supervisor: Prof. Katunobu Itou

した。実験では、訓練フェーズにおいて、行動に対する報酬の期待値の表である、単一の Q-Table を用いた強化学習で、銃を敵に向けて撃つことに対してだけを学習し、学習データは 100 デスごとに保存した。そして、実際に利用する際は NPC の強さを増減させるために、キルとデスの差を閾値として学習データを上下させ、相手とのスキルレベルが一致するような調整を行った。実験の内容として、「SEC を使用した NPC」対「5 つの固定レベルの NPC」で戦わせた。その結果として、全てのレベルでキルデス比をほぼ 1.0 にキープすることができたため、バランスがうまく調整されたと結論付けた。この結論から、エージェントのパフォーマンスが悪い状態から良い状態になる前提があれば、学習段階を上下させることでエージェントのパフォーマンスを変更し、バランス調整を行えることがわかった。しかし、この調整では銃を敵に向けて撃つだけの学習しかしていないため、他の戦略を考慮する余地が十分にあるとしていた。また、人間相手に実験していないため、動的難易度調整がに人間に有効であるかどうかが不明である。さらに、動的難易度調整時に初期のレベルを 0 から始めるとしていたため、初期難易度が低すぎるという問題もあった。

複数タスクを考慮するため、ニューラルネットワークを使用する強化学習の研究も多い。その強化学習を深層強化学習と呼び、最も基礎的なものとして Deep Q-Network (DQN) がある。そして、この学習の効率を向上させた Dueling DQN [7] という手法が存在する。この学習方法は、基本形である DQN のニューラルネットワークを 2 つに分割し、状態価値と行動価値をそれぞれのネットワークで求めるものである。この手法は、どのような行動をとっても価値がほとんど変わらないような状態において学習を促進することができ、実際に性能が向上したことを示した。

一人称視点のゲームで強化学習を行える環境はいくつかあり、SEC [5] では Unreal Tournament 2004 + Pogamut 3 という環境を使用した。しかし、一人称視点における強化学習のシステムレビュー [9] によると Unreal Tournament 2004 + Pogamut 3 という環境は古く、現代ではほぼ使われておらず現代で使う環境として良くないと記されている。そのため、関連研究の環境を再現するために重要な要素を取り出し、それらが存在し現代で使われている環境を使用する。ここで重要な要素として、学習する相手がいること、ゲーム中のステータスが取得できること、学習を分割して保存できることがあげられる。現代の研究によく使用されている環境として、VizDoom [8] や Unity ML-Agents [10] という環境があげられている。その中でも、Unity ML-Agents より VizDoom の方が使用されていると示した。

3 準備

3.1 Deep Q-Network

深層強化学習である Deep Q-Network (DQN) [11] は、強化学習に使用されていた Q 学習 [12] にニューラルネットワークの考え方を含めた手法である。Q 学習では Q 値 (期待値) の表を更新していく仕組み上、連続的な状態を表そうとすると膨大な数の状態数となり、学習を行うことが現実的に難しい。一方 DQN では、Q 値の推定にニューラルネットワークを使用して、 Q 値の近似関数を得ることで、複雑な状態の定義を行うことも可能となる。具体的には、状態行動価値関数をニューラルネットワークの近似関数で求め、その状態時の行動毎 Q 値を推定できれば、最も Q 値の高い、最善である行動が分かる。具体的な流れとして、状態 s を入力層、行動 a を出力層のノードとなるニューラルネットワーク (図 1) を使用し、Q 値 Q(s,a) を計算する。

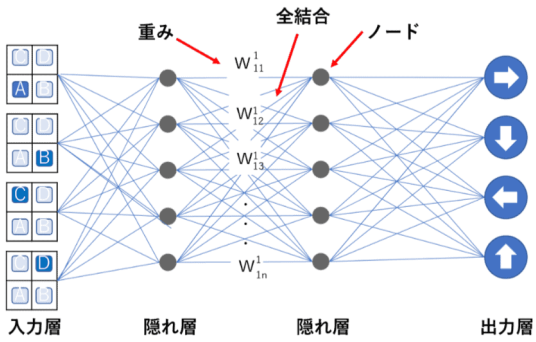

図 1　Deep Q-Network

3.2 Dueling Deep Q-Network

Dueling Deep Q-Network (DDQN) [7] は Deep Q-Network のネットワークを、状態評価と行動評価の 2 つに分離し (図 2)、それぞれのネットワークで状態価値と行動価値を求める手法である。この手法を用いることで、完全に積みな状態、すなわちどのような行動を取っても結果があまり変わらない状態の学習が促進される。

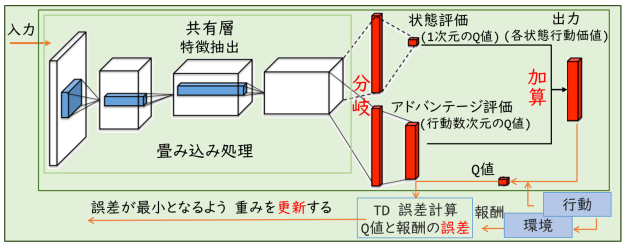

図 2　DDQN を用いた学習の流れ

学習の流れは図 2 である。まず、学習に必要な経験である状態として行動前のダウンサンプリング済みフレーム、エージェントの行動、行動に応じた報酬、次の状態としてダウンサンプリング済みフレームをメモリに保存する。次に、メモリに保存された経験をランダムに選出し、ネットワークの学習を行う。メモリに保存してある状態を共有層のネットワークに入

力し、入力された状態を処理する。次に、状態評価とアドバンテージ (行動) 評価 の 2 つに分離したネットワークでそれぞれの Q 値を近似する。次に、それぞれの Q 値を合計することで状態行動価値とし、実際の報酬の差、TD 誤差の計算を行う。最後に、この誤差が最小となるようネットワークの重みを更新する。

3.3 VizDoom

VizDoom [8] は、id Software が開発した Doom という First Person Shooter をベースにしたゲームを深層強化学習用に API を用意した環境である。VizDoom を対象としたゲーム AI についても多く研究されており、2017 年の Conference on Computational Intelligence in Games (CIG) ではこのゲームを対象とした Visual Doom AI Competition [13] という大会が開催された。VizDoom は、ゲーム AI 実装用にゲーム内の敵の位置や、フレーム内に表示されるオブジェクトなどといった環境内のパラメータを取得できる環境であるため、強化学習を比較的容易に実装することが可能である。本研究では、この環境を用いて実装を行った。

図 3　VizDoom のプレイ画面

3.4 Reward Shaping

様々な報酬を用意する際に報酬の尺度が異なる場合や、極端に大きい報酬がある場合、学習が不安定になり歪んでしまうという問題がある。そこで、Deep Q-Network [11] 内にある報酬のクリッピングという手法を用いる。報酬のクリッピングとはすべての報酬を一定の範囲に制限することで、不安定な学習を抑制させ、収束性を向上させる。

また、報酬がスコアだけに基づいている場合、報酬がゼロでない状態とアクションのペアが非常にまばらになり、エージェントが有利なアクションを学習するのが非常に難しくなる。そこで、学習プロセスをスピードアップするための中間報酬を含む報酬関数の修正 [14] を行う。中間報酬を導入することで、まばらな報酬の問題を解決し学習が高速化する。

これらの Reward Shaping はそれぞれ優位性を示したが、報酬の大きさによって学習が安定しない問題や、重要なタスクを見失い、適切なポリシーの学習ができず、学習が安定しないという問題がある。また、弱い状態から強い状態になるまでの学習の幅が狭いと、十分な難易度が作成できない。そして、動的難易度調整をするためには学習曲線が右肩上がりになる必要があるため、本研究では報酬のスケーリングを行いこれを解決する。

4 提案手法

本研究では、Glavin ら [5] の研究の発展として、FPS ゲームのデスマッチルールで Dueling Deep Q-Network (DDQN) を用いた動的難易度調整を行うため、学習曲線が右肩上がりになるような Reward Shaping を提案する。さらに、先行研究では動的難易度調整の初期値を 0 としていたが、初期難易度が低すぎることがあるため、初期値を手動で設定する。この提案手法の目的は、学習曲線が右肩上がりになるような Reward Shaping によって初期値設定を含めた動的難易度調整を行い、ユーザー体験を向上させることである。

従来手法では、報酬の尺度を揃える Reward Clipping や、まばらな報酬を解決するために中間報酬を導入している。しかし、報酬の大きさによって学習が安定しないため、本研究では報酬のスケーリングを行うことでこれを解決する。具体的には、一番重要な報酬を大きく設定し強調し、重要なタスクに注目させる。また、中間報酬で与える負の報酬を小さくし、重要な学習を見失わないようにするといった手法を取る。

この手法を対戦ゲームである First Person Shooter のデスマッチルールに適応する。FPS ゲームのデスマッチルールは、敵をより多く倒した方が勝ちとなるゲームである。そのため、このルールで一番重要なタスクは勝敗に直接関係する「敵を倒す」ことである。そして、「敵を倒す」ためには「敵に弾を当てる」必要がある。なので、一番重要な「敵を倒す」タスクの報酬を大きくし強調する。また、中間報酬として「弾を当てる」タスクを設定する。また、敵を倒すために銃を撃った時、その弾が敵に当たらなければ、自身は弾を失い不利になる。そのため、敵に弾が当たらなかった場合はペナルティを与える。しかし、敵を倒すためには銃を撃つ必要があるため、このペナルティが大きいと銃を撃たない学習に陥る可能性がある。そのため、少ない報酬に設定することで重要なタスクを見失わず学習できるようにする。

また、弱い状態から強い状態まで、幅広く動的難易度調整で使用できる学習になるような報酬の値を設定する。報酬の値の設定は環境ごとに異なる。そのため、本研究では学習が進まない値から学習の伸びが同じくらいになる値までの範囲を実数で探す。

このように Reward Shaping を行い、DDQN を用いて VizDoom で学習する。学習時、動的難易度調整を行うためのデータを作成するため、エピソードごとに学習データを分割し保存する。そして、実際に利用する際、ユーザーに合った初期値を

設定、分割した学習データを動かすことで動的難易度調整を実現し、ユーザー体験向上を目指す。

5　実験

5.1　報酬スケーリング比較

提案手法を VizDoom に実装し学習することで、報酬のスケーリングが有効であるかどうかを確かめる。スケーリングは、一番重要なタスクである「敵を倒す」という行動を強調するため +25.0 という値に設定、このタスクを行うために必要な「敵に弾を当てる」という行動に +5.0 という中間報酬を与える。加えて、弾が当たらなかった場合のペナルティとして -0.5 という小さい中間報酬を与える。小さい報酬にすることで銃を撃たない学習になることを防ぐ。また、比較としてスケーリングしていない場合の実験も行う。スケーリングしていない場合の値は、「敵を倒す」が +5.0、「敵に弾を当てる」が +1.0、「敵に弾が当たらない」が -0.5 である。これらの値は VizDoom AI Competition 2017 で 1 位を獲得した Arnold というモデルのプログラムを参考にし、値を変化させた結果設定したものである。

5.1.1　結果

「3 分 (1550 steps) x 200 試合 (episodes)」で学習した結果を図 4 に示す。

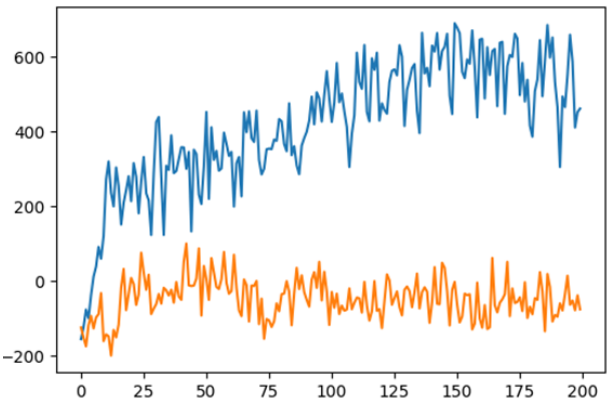

図 4　各エピソードの報酬

上の青線がスケーリングした場合、下のオレンジ線がスケーリングしていない場合のみの学習曲線となっている。この結果を見ると、スケーリングなしの報酬は学習が安定していないことが分かる。逆に、スケーリングありの場合、学習曲線がおおむね右肩上がりとなる学習ができたことが分かる。このことから、一番重要なタスクの報酬を強調設定し、小さな中間報酬を設定することが有用であることが分かった。また、中間報酬を小さく設定したため、重要なタスクを見失わず銃を撃たないという学習にならなかった。そのため、提案した Reward Shaping が有効的であると言える。

5.2　報酬値比較

本研究の Reward Shaping を用いた報酬の値を変更し、学習結果がなだらかになる報酬を探す。報酬の値は 5.1 章でスケーリングしていない「敵を倒す」が +5.0、「敵に弾を当てる」が +1.0 の値を等倍、1 倍、2 倍のように変更し確認する。実際に行った実験は学習が進まない最低値としてスケーリングしていない +5.0、 +1.0 とし、そこから 1, 2, 3, 4, 5, 10 倍 で実験を行った。また、図を作成する際報酬の最大値で割ることで正規化している。

5.2.1　結果

結果を図 5 に示す。

図 5　各報酬の学習結果

この結果から、報酬が等倍と 2 倍の場合はほとんど学習が進んでいないことが分かる。次に、報酬が 3 倍の場合は途中までは低迷しているが徐々に学習が進み、検証した中では一番なだらかな学習となっている。次に、報酬が 4 倍の場合は、学習は右肩上がりに見えるがかなり曲線が上下しており、あまり安定していない。最後に、報酬が 5 倍と 10 倍の場合は初期の段階で学習がかなり進み、そこからは右肩上がりではあるものの学習があまり進まないという結果になった。この中で動的難易度調整に使用するには、3 倍の「敵を倒す」が +15.0、「敵に弾を当てる」が +3.0 の値が一番適している。

5.3　動的難易度調整

難易度調整ができているかどうか、難易度が固定されている組み込み Bot との対戦で確認する。データは 5.1 章で学習したものを使用する。従来研究の SEC では、敵との対戦で最終的なキルデス比がほぼ 1.0 にキープされたことから、動的難易度調整ができていることを示した。そのため、同じくレベルが固定な敵と戦わせてキルデス比が 1.0 でキープされることを確認する。

5.4　結果

「3 分間 x 200 試合」「3 つの固定レベルの敵」と強化学習、同じく 3 つのレベルと動的難易度調整とで対戦した結果のキルデス比の推移をそれぞれ図 6,**??**に示す。また、それぞれのキルデス比の平均と分散の表をそれぞれ表 1, **??**に示す。

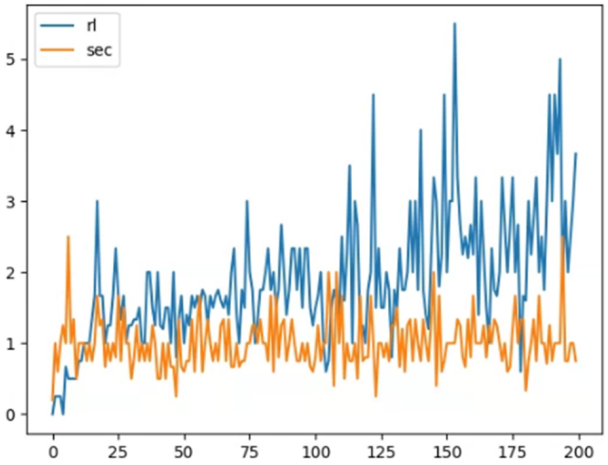

図 6　動的難易度調整と強化学習のキルデス比の推移

上の青線が深層強化学習を使用した場合と、下のオレンジ線が動的難易度調整を使用した場合のものである。これらの結果を見ると、深層強化学習を使用した NPC は徐々にキルデス比が向上し、強くなりすぎていることが分かる。逆に、動的難易度調整を使用した NPC はキルデス比が 1.0 付近でキープされている。

	平均	分散
強化学習	1.888917	0.812084
動的難易度調整	1.004655	0.134391

表 1　動的難易度調整と強化学習のキルデス比の平均と分散

続いて、キルデス比の平均と分散の表を見ると、動的難易度調整は平均がほぼ 1.0 付近となっており、分散も小さいことから、バランス調整が機能していることが分かる。これらのデータにより、深層強化学習を使用した学習データを用いて難易度調整を行えることが分かった。

さらに、この動的難易度調整を人間相手と対戦させた結果、固定難易度の敵と比べ、レベル上昇に伴い撃ち合いが激しくなる行動を取るようになった。さらに、難易度の初期値を学習時に収集したキルデス比を参考に設定することで初期難易度が低い問題が改善され、ユーザー体験が向上した。

6　考察

本研究では SEC [5] の手法を発展させたため、学習段階においてスキルレベルは学習時間の増加とともに単調に成長するという暗黙の仮定に依存している。そのため、深層強化学習を用いた学習が右肩上がりになるよう、報酬のスケーリングを行う Reward Shaping を行った結果、右肩上がりに安定した学習結果となった。これは重要なタスクを強調、負の報酬を小さくすることで、エージェントが重要なタスクを見失わなかったためうまくいったと考える。

報酬の大きさを変えることで学習が滑らかになった。これは、報酬が大きすぎるとその方向のみに学習しやすくなり、小さすぎると重要なタスクを見失うため、大きすぎず小さすぎないという間の報酬を選択することで滑らかになったと考える。また、固定難易度の敵と動的難易度調整の戦闘結果、k/d がほぼ 1.0 に安定した。これは、学習の段階を保存し、段階を増減させることで、動的難易度調整が機能したことを示している。さらに、人間プレイヤーと動的難易度調整の戦闘で体験が向上した。これは、難易度上昇に伴い敵が好戦的となり、戦闘の質が向上した点と、初期値設定により、敵が弱すぎることなくゲームスタートできたからだと考える。そして、今回のゲームルールが敵と戦闘することを目的としているため、体験向上に直接響いたと考えた。

今回の Reward Shaping は重要なタスクを強調しているため、複数タスク学習時、他のタスクを学習する妨げとなり他のタスクを学習できない可能性がある。ゲームには、アイテム収集、攻撃回避、マップ内の移動といったタスクが存在しており、考慮するべきタスクが多い。今回は、重要なタスクを「敵を倒す」として設定したが、ゲーム内の他のタスクを学習する場合は Reward Shaping の見直しが必要になる可能性は十分にある。

また、今回は武器を固定していたが、敵を倒す、敵に弾を当てるという報酬を与えているため、他の武器でも適応できる可能性は高い。しかし、弾を撃ってから当たるまでの時間が長い武器は、報酬がかなり遅延するため学習は難しくなると予想される。当たるまでが早い武器のみのゲームなどでは、複数武器に特化した学習データをそろえることでより多様なプレイスタイルに対応できるようになる可能性は高い。

深層強化学習を用い、動的難易度調整用に Reward Shaping をした結果、動的難易度が可能な学習を行うことがで来たことを示している。今回提案した手法は明示的に重要なタスクが存在する場合、様々なゲームジャンルでも有用である可能性は高い。今回は VizDoom 環境で検証し、FPS ゲームにおける重要なタスク、「敵に弾を当て倒す」という学習でその成功の可能性を示した。

7　おわりに

本論文では、動的難易度調整のための Dueling Deep Q-Network (DDQN) を用いた、右肩上がりに安定する学習の Reward Shaping と初期難易度設定を変更する提案をした。本研究の目的は、ユーザー体験を向上するための報酬設計を用いた動的難易度調整を可能にすることである。また、動的難易度調整に使用できるような滑らかな学習曲線を描くために報酬のスケーリングを行い、深層強化学習を行うことでユーザー体験を向上させることである。実験の結果から学習は右肩上がりになり、動的難易度調整で使用できることが示された。さらに、報酬の値を大きすぎず小さすぎ内容に設定することで、学習曲線が滑らかになり、より細かい難易度を作成可能となった。ま

た、難易度の初期値を設定することで、初期の難易度が低すぎる問題も改善することができた。

今回の Reward Shaping は、重要なタスクを抽出しその報酬を強調していた。しかし、重要なタスクとは別の学習を行いたい時、この報酬が干渉し他のタスクを学習できない可能性がある。この問題は、1 つの学習ではなく行動に対してそれぞれ別に学習すれば解決できる可能性がある。例えば、ゲームのキャラクターを動かす場面と、敵を倒す場面で切り替えることができればより思い通りな学習になるといったものが挙げられる。さらに、敵の動きなどを考慮する場合、前の時間軸を考慮する必要があるため、Recurrent Network や Transformer を使用する学習の方が良くなる可能性がある。

また、動的難易度調整を行うため学習を区切る基準についても課題が残る。学習段階作成時パフォーマンスを分析し、漸進的にならないパフォーマンス向上のポイントを省くなどして難易度の停滞を防ぐことが必要だろう。例えば、学習の初期段階では学習段階の数を多くし、学習が停滞し始めたら学習段階の数を少なくするといったものが挙げられる。そして、難易度調整の初期難易度が手動設定であるため、自動である程度の難易度を選択できるような手法を検討する必要がある。

今回使用した環境は VizDoom であるが、現代のゲームは数多く存在するため、他のゲームでも適応できるかどうかは検討の余地が残る。また、今回は 1 対 1 のデスマッチルールのみを考慮したものであったが、5 対 5 のチーム制のあるルールなど、相手が一人でないゲームの方が多い。敵が一人ではない場合はさらに考慮するべき点が増えるため、さらなる検討が必要となる。

参考文献

[1] 山下利之, 清水孝昭, 栗山裕, 橋下友茂. コンピュータゲームの特性と楽しさの分析. 日本教育工学会論文誌, Vol. 28, No. 4, pp. 349–355, 2005.

[2] M. Csikszentmihalyi. *Flow: The Psychology of Happiness*. Ebury Publishing, 2013.

[3] Penelope Sweetser and Peta Wyeth. Gameflow: a model for evaluating player enjoyment in games. *Comput. Entertain.*, Vol. 3, No. 3, pp. 3–3, 2005.

[4] Robin Hunicke and Vernell Chapman. Ai for dynamic difficulty adjustment in games. *Challenges in game artificial intelligence AAAI workshop*, Vol. 2, , 01 2004.

[5] Frank Glavin and Michael Madden. Skilled experience catalogue: A skill-balancing mechanism for non-player characters using reinforcement learning. pp. 1–8, 08 2018.

[6] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[7] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning, 2016.

[8] Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. ViZDoom Competitions: Playing Doom from Pixels. *IEEE Transactions on Games*, Vol. 11, No. 3, pp. 248–259, 2019. The 2022 IEEE Transactions on Games Outstanding Paper Award.

[9] Pedro Almeida, Vitor Carvalho, and Alberto Simões. Reinforcement learning applied to ai bots in first-person shooters: A systematic review. *Algorithms*, Vol. 16, No. 7, 2023.

[10] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2020.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, pp. 529–533, 2015.

[12] Christopher Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, Vol. 8, pp. 279–292, 05 1992.

[13] Marek Wydmuch, Michal Kempka, and Wojciech Jaskowski. Vizdoom competitions: Playing doom from pixels. *CoRR*, Vol. abs/1809.03470, , 2018.

[14] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning, 2018.