

情報電気電子工学特別演習 EEE

題 目: 合宿免許 WAO!

WAO

発表者: 旧加勢大周

Taisyu Old Kase

所 属: XXXX 研究室 (XXXX 研)

XXXX 研究室 (XXXX 研)

Abstract

From 2014 onwards, mobile games will be ranked first in the consumer spending in the game market, excluding PC games and consumer games. Mobile games are highly growthable markets. Voodoo is located 2nd in the download number next to Google on the worldwide download count of the App Store, and is the number one download at the game publisher in the world. The category of games that Voodoo deals with is called a casual game. In the casual game, the balance of game difficulty (game level design) is adjusted by human hand. Evaluation of difficulty level is generally determined by human playing. Evaluation of this degree of difficulty has an important influence on game balance and it is an important process. This process is a costly task and therefore is a task in game creation. In this research, a stage is automatically generated for a casual game of a stage system, an automatically generated stage is played by an agent who has already learned using reinforcement learning, and evaluate difficulty level reinforcement learning compared with human evaluation. We used the Unity ml-agents platform to create and learn environment of learning.

1 はじめに

2014 年以降, ゲーム市場における消費支出でモバイルゲームが PC ゲーム, 家庭用ゲームを抜き 1 位になり¹, モバイルゲームは成長性が高い市場である. Voodoo 社²は App Store の全世界ダウンロード数で Google 社に次いでアプリダウンロード数 2 位に位置し, ゲームパブリッシャーではダウンロード数世界第 1 位である. Voodoo 社が手掛けるゲームのカテゴリは, カジュアルゲームと呼ばれる. カジュアルゲームの特徴として, まずグラフィックがシンプルで, 操作が非常に簡単である, マニュアルなどもほとんど無く, すぐにゲームが始められる内容である. 一方で, 非常に高度なゲーム要素があり, 難易度のバランスが絶妙でユーザーを飽きさせない工夫がされており, 中毒性の高いゲーム内容になっている. カジュアルゲームにおける, 難易度のバランス (ゲームレベルデザイン) は人間の手によって調整されている. 難易度の評価は一般的には, 人間がプレイすることによって決定される. この難易度の評価はゲームバランスに大きく影響を及ぼすため重要なプロセスになる. このプロセスはコストの高い作業となるためゲーム作成における課題である.

本論文では, ステージ制のカジュアルゲームを対象として, ステージを自動生成する. 自動生成されたステージを強化学習を用いて学習済みのエージェントにプレイさせ, ステージの難易度を自動的に設定する. 強化学習による難易度付けと人間の評価を比較して評価する. 学習

の環境の作成および学習に Unity ml-agents[5] プラットフォームを使用した.

本論文の構成は以下の通りである. 第 1 章では本論文の背景, 目的について, 第 2 章では本論文における事前知識について, 第 3 章では強化学習を用いたゲームレベルデザインの提案手法について, 第 4 章では現時点における実験結果について, 第 5 章では現時点での結果第 6 章で将来における課題, まとめについて述べる.

2 強化学習

強化学習とは試行錯誤によって学習を行う機械学習の 1 種である. 強化学習ではエージェントと呼ばれる行動の主体とエージェントの行動に対して反応を返す環境の相互作用によって学習を行う. 環境はエージェントの行動によって報酬と次状態を生成する. 報酬はエージェントの行動の評価である. 次状態はエージェントの行動によって環境に生じた変化を表す. マルコフ決定過程は, このエージェントと環境の相互作用を数理モデルである. 図. 1 にモデルを示す.

2.1 ML-Agents

Unity Machine Learning Agents (ML-Agents)[5] はゲーム開発環境である Unity において強化学習などの機械学習を使用できる環境を提供しているプラグインである. 機械学習のバックエンドとして tensorflow[1] が使用されている. 他の学習環境である ViZDoom[6] や Malmo[4] と比較して優れている点として Unity API の

¹<https://www.appannie.com/jp/insights/market-data/idc-mobile-gaming-report-2017/#download>

²<https://socialappsupport.com/column/voodoo/>

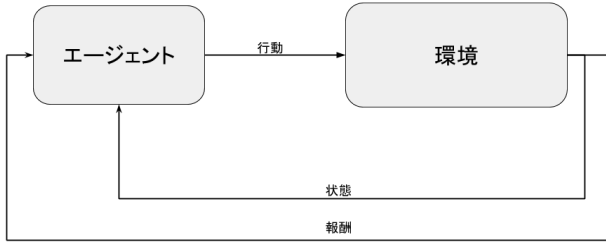


図 1: マクロ過程

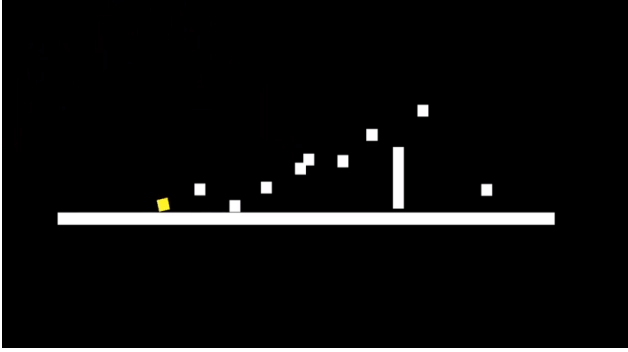


図 2: Shikaku プレイ画面

強力な支援を得られる点や、高精度な物理環境が整っている点、既存のゲームに依存しない点、さらに学習アルゴリズムが事前に実装されている点などがある。

2.2 Shikaku

Shikaku は今回作成したゲームである。図. 2 にゲームのプレイ画面を示す。黄色の立方体はプレイヤーである。ゲームクリアのためのタスクとしてプレイヤーはステージ上に存在する他のオブジェクト (箱) を集めることである。プレイヤーの立方体が他の立方体に接触するとその 2 つが結合し 1 つのオブジェクトとなる。また、プレイヤーとの接触時に結合しない壁が存在する。

3 強化学習を用いた難易度レベルデザイン

レベルデザインとしてこれまでに用いられた手法として Super Mario Brothers のステージを GAN を用いて生成する手法 [7] や Doom のステージを生成する GAN[3] などが登場している。これらの手法を用いて生成されたステージにおいてはステージの難易度について問われることはなかった。また, GAN を用いた手法においては学習にサンプルデータが必要となる。新しくゲーム作成する場合においては学習のための既存データが存在しないため GAN を用いた生成が困難である。

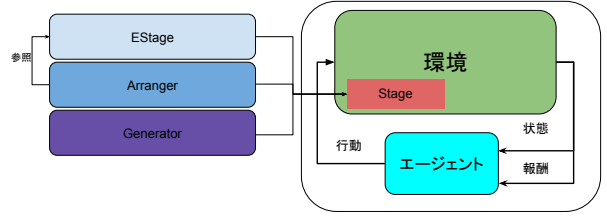


図 3: レベルデザインのための学習

強化学習は事前知識なしで、経験により学習を行うことができるため、サンプルデータを必要としない。

3.1 構成

図. 3 に難易度レベルデザインの構成を示す。Generator はステージの生成器である。Generator はプレイヤーの初期位置, 取得できる箱および壁の位置, 大きさをランダムに自動生成する。Generator は生成したステージが実際にタスクを達成可能かを保証しない。Generator が生成する箱および壁が生成する個数の最小値と最大値を設定できる。強化学習では達成不可能なタスクにおいてはタスク達成のための解法を学習することができないため、タスクの達成が可能なステージを今回 5 ステージのみ作成した。EStage は既存ステージの集合である。Arranger は EStage をパラメータを多少調節することで既存のステージを元に新しくステージを生成する。Agent が受け取る情報としてプレイヤーの速度, 回転速度, 回転角, 箱, 壁のプレイヤーとの相対位置, 大きさである。表. 1 に Agent が Environment から受け取る報酬とその条件を示す。

表 1: 報酬の割り当て

報酬条件	報酬値
プレイヤーがタスク未達成	-0.01
オブジェクトに結合	+0.3
タスクを達成	+1
タイムアウト時にタスク未達成	-1
ゲームオーバー	-1

3.2 フロー

図. 3 によって実際に学習していく際の手順について説明する。

1. EStage を用いてエージェントを学習
2. 1 で学習したエージェントを用いて Arranger で生成したステージをプレイし、プレイ可能なステージを選別する

3. Arranger で生成したステージでエージェントを学習
4. 3. で学習したエージェントを用いて Generator で生成したステージをプレイし, プレイ可能なステージを選別する
5. Generator で生成したステージでエージェントを学習
6. Generator で生成できる箱と壁の個数の最大値を増やし 4,5 を繰り返し行いエージェントの学習を行う.

Generator のエージェント学習の際に使用するステージは EStage で学習したエージェントや Arranger で学習したエージェントより学習に使用したステージと違い, 既存のステージに依存しないため, より多様性のあるステージを用いて学習ができる. 一般に, 学習時における環境の多様性はエージェントの汎化能力が獲得できる [2] ため, EStage, Arranger, Generator と生成した (保持する) ステージを用いて学習したエージェントはより段階が進むに従って汎化能力が得られると考えられる.

4 実験

現時点では評価結果などのデータが全て収集できていないため, 評価方法について述べる.

4.1 ステージ難易度設定の評価手法

エージェントはステージの難易度を決定する手法として, 以下のフローで行う

1. エージェントは評価するステージを規定回数プレイし, ステージをクリアできた回数を計測する
2. クリア回数に応じて 5 段階で評価を行う.

同様に人間のプレイヤーに同じステージでプレイしてもらい評価してもらう.

評価するステージは Generator で生成し, プレイヤがタスク実行が可能であるようプレイが可能であることを確かめた上で使用する.

図. 4 に評価する 10 ステージを示す.

エージェントの評価の比較対象として人間による評価との類似度によってエージェントの評価能力を評価する. エージェントのプレイ能力 (ゲームのプレイのうまさ) によって人間の評価との類似度は変化するため, 人間の評価を基準としてエージェント能力を比較する.

人間とエージェントのプレイの類似式を式. 1 に示す.

$$E = \frac{\sum_{i=0}^n (eh_i - ea_i)}{n} \quad (1)$$

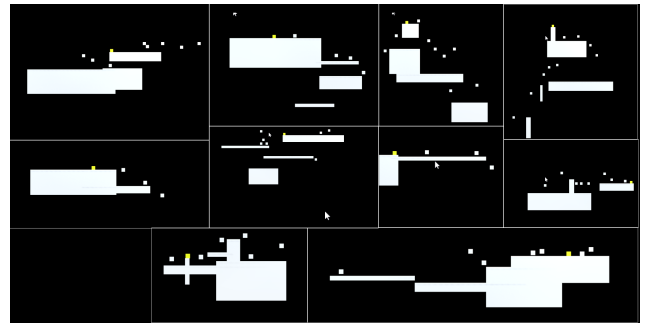


図 4: 評価に使用する 10 ステージ

n は評価したステージ数. eh_i は i 番目のステージにおける人間の難易度評価である. ea_i は i 番目のステージにおけるエージェントの難易度評価である.

E が正の場合エージェントは人間のプレイと比較して, よりプレイ能力が高い. 反対に値が負の場合エージェントは人間よりプレイ能力が低い. この値がより小さいほどエージェントと人間のプレイが同程度の能力であることを示す.

4.2 実験結果および考察

現時点では実験結果が取りきれいでないので結果の評価および考察は省略する.

5 まとめ

本論文では, カジュアルゲームの難易度レベルデザインに対して強化学習を用いて評価を行い, 既存データが少ない環境において自動生成と強化学習で汎化能力を手に入れる方法を提案した. 実験結果については, 現時点では評価結果の収集が終了していないため, 評価方法のみを提案した.

5.1 課題

現在この手法の評価において, 少人数の人間の評価を元に行なっているが, さらに多くの人数から評価を集めることで難易度をより細分化できるようにし, プレイヤに沿った難易度を提案できるようにする必要がある.

強化学習を用いてステージの評価はできるが, 現段階においてステージ生成のジェネレータはヒューリスティックに作成している. このため, ジェネレータを作成するというコストに対して, DoomGAN[3] などと同様に生成することでより自動化が提案できる. また, ステージのタスク可能判定について, 強化学習のエージェントを用いて判定を行なっているが, これをステージデータおよびそのタスクの可否をエージェント用いてステージのデー

タセットを生成する。これによって、前提知識を学習することが可能になると考えられる。

参考文献

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zhang. 2016. TensorFlow: A system for large-scale machine learning. In *OSDI*.
- [2] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 2018. Quantifying generalization in reinforcement learning.
- [3] Edoardo Giamello, Pier Luca Lanzi, and Daniele Loiacono. 2018. DOOM level generation using generative adversarial networks.
- [4] Hutton T. Johnson M. Hofmann K. 2016. *The malmo platform for artificial intelligence experimentation. Proc. 25th international joint conference on artificial intelligence*. AAAI Press, California USA. Retrieved from <http://www.ijcai.org/Proceedings/16/Papers/643.pdf>
- [5] Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. 2018. Unity: A general platform for intelligent agents.
- [6] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. 2016. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE conference on computational intelligence and games*, 341–348. Retrieved from <http://arxiv.org/abs/1605.02097>
- [7] Vanessa Volz, Jacob Schrum, Jialin Liu, Simon M. Lucas, Adam Smith, and Sebastian Risi. 2018. Evolving mario levels in the latent space of a deep convolutional generative adversarial network.