

ローグライクゲームの研究用ルール提案とモンテカルロ法の適用

高橋一幸^{†1} Temsirirkkul Sila^{†2} 池田心^{†3}

概要: 日本では「不思議のダンジョン」シリーズで名高いローグライクゲームは、マップや敵配置がランダムなこと、そのため臨機応変にアイテム等を使い分ける必要があることなどを特徴とする一人用ダンジョン探索ゲームの総称である。囲碁や将棋などよりは複雑で、Starcraft や Civilization などよりは単純なこのゲーム群は、短期的戦術と長期的スケジューリングの両方を要するなど、高度なゲーム AI を研究するうえでの良い課題をいくつも持つ。本論文では、学術用プラットフォームを共有できるようにするために、ローグライクゲームの最低限の要素を持たせたゲームのルールを提案した。そのうえで、ルールベースのコンピュータプレイヤー、短期的な戦術行動を改善するためのモンテカルロ法プレイヤー、長期的な視点での行動が取れるようにするための教師あり学習を提案した。

キーワード: ローグライク, 不思議のダンジョン, モンテカルロ法, 教師あり学習, プラットフォーム

Research Platform of Rogue-Like Games and Monte Carlo Players

Kazuyuki TAKAHASHI^{†1} Sila TEMSIRIRIRKKUL^{†2} Kokolo IKEDA^{†3}

Abstract: Roguelike Game, known as "Mystery Dungeon" in Japan, is a genre of one-player role-playing video games. In roguelike games, the map is randomly created and the monsters are randomly located every play, then the player needs to select his strategy properly according to the given situation. Roguelike games are more complex than the game of Go or Chess, and less complex than Starcraft or Civilization. They are less complex, still they have several typical problems for sophisticated game AIs, for example both of short-time strategy and long-time scheduling are necessary and the balancing is important. In this paper, we proposed a new roguelike game, for sharing an academic platform of such games among researchers. Also we presented a rule-based computer player, a Monte-Carlo based player for solving short-term combat, and a supervised learning for long-term scheduling.

Keywords: Roguelike Games, Mystery Dungeon, Monte-Carlo Search, Supervised Learning, Game Platform

1. はじめに

近年のコンピュータゲームプレイヤー（単純にゲーム AI と呼ぶ）の研究では、囲碁や将棋などのボードゲームのみならず、ビデオゲームも対象とし、様々なジャンルのゲームにおいて高性能な AI の研究が進められている。不完全情報性、多人数性、リアルタイム性など、囲碁や将棋とは違った困難さを持つゲームも多く、それぞれ新しい解決手段が必要なため、これからも研究の重要性は高い。

ゲームのジャンルによってはゲーム環境の構築が面倒であったり、ルールが煩雑であったりするため、学術研究では AI 部分のみを開発できるようにした統一のプラットフォームが用いられることが多い。実ゲームに近いものとしては Starcraft, Metastone, FightingICE, Infinite Mario など、ルールを単純化したものとしては TORCS, デジタルカーリング, 人狼, 我々の研究室で提供している崩珠, TUBSTAP なども挙げられる。

本研究では、人気の高いゲームジャンルの 1 つである“ローグライク(Rogue like)ゲーム”を取り上げる。これはある種の一人用ダンジョン探索ゲームの総称であり、国内では“不思議のダンジョン”の名前で知られ、1993 年の第一作発売以降、いくつかのタイトルは 100 万本以上を売り上げている。非常に知的で面白いゲームとして知られる一方で、一人用ゲームであるために囲碁などに比べるとゲーム AI の実益性は低く、学術研究も十分に行われていない。しかし我々は、このゲームには「長期的なスケジューリング」「短期的な戦術」「探索と収穫のジレンマ」「疎密のある行動選択」など AI が取り組むべき複数の課題があると考えた。これらの課題は、より複雑なゲーム例えば Starcraft にも登場するが、ゲームが複雑なほどその学習や探索、実験や分析は質的にも量的にも困難になることが多いため、中程度の複雑さのゲームも研究対象として有益であると考えた。

そこで本論文では、ローグライクゲームを対象として、これらが持つ要素を列挙・整理し、多くのタイトルに共通する基盤的要素を抽出する。この基盤的要素を基に、新しいプラットフォームを提案し、そのルールを提示する。その上で、ルールベースプレイヤーの作成、疎な意思決定を行うモンテカルロプ

†1 北陸先端科学技術大学院大学 s1610110@jaist.ac.jp

Japan Advanced Institute of Science and Technology

†2 temsirirkkul@jaist.ac.jp

†3 kokolo@jaist.ac.jp

レイヤの作成, 価値関数の機械学習, を試みる.

2. 関連研究とプラットフォームの必要性

ローグライクゲームの詳細な定義は次章で述べるが, 概ね, 1人の冒険者がダンジョンを探検し, アイテムを集めそれを駆使しながら敵を倒し, 力を強め, 複数階層を突破してゴールを目指す形態を取る (図1). マップや敵配置・アイテム配置がランダムであることが特徴の一つであり, プレイヤは毎回異なる状況に合わせて作戦を立てる必要がある.

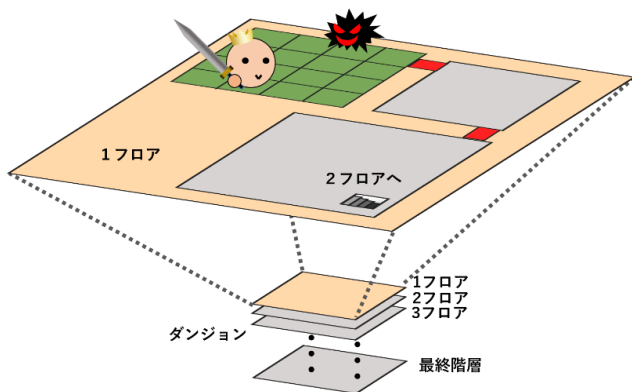


図1: ダンジョンの構造, 概略

ローグライクゲームを直接扱った研究としては, 田中らによる提案[1]や, Vojtechらによるものがある[2]. 田中らの提案は現在では追試・実装が困難であり, また Vojtechらは Desktop Dungeon というゲームを対象としているが, これには複数階層性がなく, 長期的なスケジューリングが必要ないという不満がある. これらから, 新規プラットフォームを作成公開することには一定の価値があると考えられる.

またそもそもローグライクゲームに絞ってAIを研究することに価値があるのかという議論もあり得るだろう. 近年は IEEE-CIG などの国際会議で General Game Playing (GGP) および General Video Game Playing (GVGP) といった汎用ゲーム AI の競技会が行われ, さまざまなモンテカルロ探索の拡張[3]や Deep Q-Network[4]などによりその性能も向上しているためである. しかし, いずれはそういった問題群の1つになるにせよ, 課題を明確にしてそれを解決していくことには価値があると考えられる.

もう一点, ローグライクゲームはプレイヤー対ルールベースの単純な敵キャラ複数という形を取り一人ゲームと称されることが多いが, 実は「AIが制御する味方キャラ」がいるゲームも多く, その動きは時としてプレイヤーに不満を感じさせるものになってし

まっている. 注意したい点は, これが必ずしも“仲間AIの行動が勝利のために最適でないから”ではなく, “人間プレイヤーの好みや意図に合った行動をしてくれないから”であることである. これは我々の価値観学習チームプレイ AI の研究[5]と同様の背景であり, この点を解消するためにも, AIプレイヤーの性能向上と, 将来的には味方キャラがいるプラットフォームの提案が必要だと考えている.

3. 必要な要素とルールの提案

ローグライクゲームは Rogue のほか Angband, Nethack, 不思議のダンジョンシリーズなど数多くの有名な作品がありその数はフリーゲームを除いても数十にのぼる. 当然ルールも様々であり, そもそも何をもってローグライクゲームと呼ぶのかすら明確ではない. 2008年の International Roguelike Development Conference では, Roguelike とは何かを定義する試みが行われ, Berlin Interpretation としてまとめられた[6]. その中では, 全てが必須ではないと断った上で, 以下のような要素が high value factors として提唱されている.

- **Random environment generation.** マップ, アイテムの配置や出現, モンスターの配置 (種類は固定) が毎回ランダムに決定されること.
- **Permadeath.** ゲームキャラクタは, アイテムや経験を溜めても, 死ねばそれらを失って最初からプレイすること.
- **Turn-based.** リアルタイムの判断や操作を要求せず, 熟考できること. 通常, 自分のキャラクタが1つ移動または行動をすれば, その後にモンスターも1つ移動または行動する (注: 倍速などの敵もいる).
- **Grid-based.** マップは正方形のマスキュで構成され, 敵味方のキャラクタは1つのマスキュをふさぐこと. ピクセル単位ではない.
- **Non-modal.** 移動と戦闘のシーンは別個でなく, 全ての行動はゲームのどの時点でも利用できること. ただしいくつかのゲームでの“店”はこれにあてはまらない.
- **Complexity.** ゲームの目標は共通でも, それを解決する方法や道筋は複数あること. 敵とアイテムの相性, アイテム同士のシナジーなどがこれを生みだすこと.
- **Resource management.** 食料や傷薬といったアイテムは有限で, 利用が制限されること.
- **Hack'n'slash.** モンスターを倒す必要とメリットがあること.

- ・ Exploration and discovery. プレイヤーはマップやアイテムの位置を知らされず、毎回慎重に探索してアイテムを探し、時には未鑑定アイテムの使用法を考える必要があること。

例えば Starcraft とは Turn-based や Grid-based の点で異なり、ドラゴンクエストなどの JRPG とは Permadeath, Non-modal などの点で異なる。

我々は、この high value factors をベースに、多くのローグライクゲームに共通しているかどうか、また良い AI を作ることが「面倒」ではなく「挑戦的」になるかどうか、という基準に基づき、第一段のプラットフォームのルールを策定し、研究環境を実装した。詳細は以下の通りである。

- ・ マップは四角マスで、50×30 とする。
- ・ マップは 3~4 部屋で構成され、1つの階段があり、階段を進むと新しい階層となる。4階層から構成され、最後の階層で階段に辿りつければ勝ちである。
- ・ 部屋のサイズや通路は 5 パターンのみを用意した (図 2)。

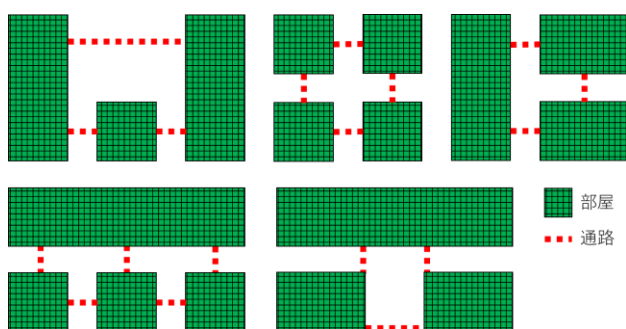


図 2：部屋と通路の構成

- ・ プレイヤーは、9×9 視野を持つ。さらに、部屋の内部にいる場合はその内部の全ての敵やアイテムの位置が分かる。
- ・ ターン制で、自分の移動または攻撃後、敵の移動または攻撃が処理される。
- ・ 移動や攻撃は上下左右斜めの 8 方向 1 マスである。ただし、上または右に壁がある場合、右上には移動や攻撃ができない。他の場合も同じ。(図 3)
- ・ プレイヤーキャラクタにはレベル・経験値・体力・満腹度・攻撃力のパラメータがある。
 - レベルは 1 を初期値とし、経験値は 0 を初期値とする。経験値は敵を倒すと入手でき、 $10 \times 1.1^{(\text{レベル}-1)}$ 溜まるとレベルが 1 上昇し、経験値はその分減る。

- 攻撃力は 35 を初期値とし、レベルごとに 1 上昇する。
- 体力は 100 を初期値とし、レベルごとに上限が 1 上昇する。また、毎ターン、上限の 1/200 だけ回復する。
- 満腹度は 100 を初期値および上限とし、10 ターンに 1 減り、0 になると体力が毎ターン 1 減る。

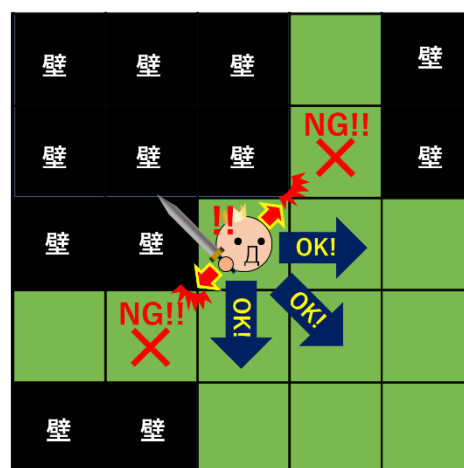


図 3：移動・攻撃できないマップの例

- ・ 飢餓または敵の攻撃で体力が 0 になると負けである。
- ・ 通常攻撃は周囲 8 マスのいずれかに対して行え、攻撃力分のダメージを与える。
- ・ アイテムは食料、回復薬、矢、杖がある。
 - アイテムは各階層に合計 3~5 つ、数も種類も位置もランダムに配置される。通路には配置されない。アイテム数は 50% の確率で 4 つ配置され、3 つまたは 5 つの場合が 25% ずつである。
 - 食料は満腹度を 20 回復する。
 - 回復薬は体力を 50 回復する。
 - 杖と矢は消耗品で、上下左右斜めの 8 方向、プレイヤーから 5 マス以内に見えている敵に遠隔で利用することができる。
 - 杖は敵を別の部屋のどこかに飛ばす(一時的のぎ) ことができる。
 - 矢は敵に 15 ダメージ与えることができる。矢は 3 本まとめて拾える。
 - アイテムは 10 個まで所持できる。
 - アイテムを置いたり投げたりはできない。
- ・ 敵キャラクタは各階層 1 種類である。
 - 敵の体力は階層ごとに 60,80,90,95、攻撃力は 20,25,30,35、得られる経験値は

15,20,25,30 とした。

- 敵キャラクターは階段を下りた時点で 4 匹ランダムな位置に配置され、さらに HP が 0 になってから 64 ターン後にランダムな位置に再配置される。
- 敵はプレイヤーキャラクターを見つけると最短距離で接近して攻撃し、そうでなければランダムに移動する。

プレイヤーは短期的には目の敵を倒したり、強い敵や複数の敵にはアイテムを消費して凌ぐ、階段から次階層に逃げるなどの選択を行わなければならない。一方で長期的には将来を見据えアイテムの収集と節約、食料に余裕があれば経験値稼ぎなどを考慮する必要がある。「突発死を恐れるかジリ貧を恐れるか」「満腹度と経験値とアイテムのバランス」などはローグライクゲームの中心的な課題であり、簡単に判断することはできない。本ルールは第一段階のテーマとして多くの要素を排除したが、それでも十分に困難で面白い課題になっていると考える。

例えば、

- ・ お金パラメータと、店要素、泥棒要素。
- ・ モンスターハウス。敵が大量出現する部屋。
- ・ 畏。踏むとペナルティのある隠れタイル。
- ・ 特殊状態。暗闇、睡眠、混乱など。
- ・ 自分と敵以外のキャラクター。味方など。
- ・ 様々な個性のある敵やアイテム。
- ・ 使うまで効果の分からない未鑑定アイテム。

などの要素は今回敢えて含めていない。これらは多くのゲームに登場し、面白い要素ではあるが、AI 作成をむやみに煩雑にする可能性が高いと判断し、今後その導入を検討する予定である。

4. アプローチ

ゲーム AI に用いられる手法は教師あり学習、強化学習、木探索、Direct Policy Search など様々であるが、我々は今のところモンテカルロ法が有望であると考えている。囲碁ではモンテカルロ法登場初期から「形勢が悪い場合は勝負手を、良ければ安全な手を」といった人間的判断が得意であることが知られており、それは前述のローグライクゲームの中心的課題とも良く似ているからである。直近の死亡リスク回避と、中長期的なジリ貧の回避、またたまたま階段が近くにあるクリアできるような幸運、これらを自然にバランスした意思決定ができる可能性が高いと期待している。

一方でモンテカルロ法をこのようなゲームに用いるには、囲碁などとは違う点に注意する必要がある。

一つには行動に疎密がある点である。敵に囲まれているような状況では 1 ターン 1 ターンに細心の意思決定をしなければならない一方で、単なる移動や経験値稼ぎの待機状態は、数十ターン程度をひとかたまりにした意思決定が求められる。毎ターンモンテカルロ法で行動選択をしていては、時間もかかるうえに行動の一貫性が損なわれる可能性がある。もう一つはゲーム終了までに多くのターンが必要な点である。囲碁でも数百手で終わるのに対し、これらゲームは 1 マップに数百ターン、それが数十階層あることも珍しくない。

これらの課題を解決する手段は単純なものから複雑なものまでさまざまに考えられる。残念ながら我々は現在までに十分に洗練された手法を実装できていないが、本稿では、通常はルールベースで動作し、敵が存在する場合のみモンテカルロ法を用いることで行動の疎密に対処する手法を紹介する。ゲーム終了までに多くのターンが必要な点については、ランダムシミュレーションをゲームの終了まで行わず、途中で打ち切り状態評価関数を用いて勝率を推定する Depth Limited なアプローチ[7]を試みる。さらに、その状態評価関数を手作りする場合と、教師あり学習を用いて学習を行った場合について比較する。

5. ルールベースプレイヤー

本章では、if-then の決定木からなるルールベースプレイヤーを紹介する。これは著者らがこのゲームをプレイする場合に考えている内容や優先順の一部のみを書き起こしたものであり、賢いとは言いがたい。より複雑な分岐を加えることでこれを強くすることはできるだろうが手間の割に性能は伸びにくいと思うため、これをベースに多少なり汎用的な手法を構築していく。

【敵が隣接している場合】

- 1) 体力が隣接敵の攻撃力（の合計）よりも高いならば、HP の低い敵を攻撃する。プレイヤーの攻撃により倒すことのできる敵がいるならば、その敵を攻撃する。
- 2) （そうでなければ：以下同じ）回復薬を持っているならば、使う。
- 3) 左下、下、右下、左、右、左上、上、右上の順に見ていき、敵からの攻撃を一旦避けられるマスがあるならば、移動する。
- 4) 杖があるならば、使用して一時しのぎする。
- 5) 敵を攻撃する。倒せなければ死ぬことになる。

【敵が2マス離れたところにいる場合】

- 1) 最下層で敵よりも階段（ゲームクリア）に近いならば、階段に向かう。
- 2) 敵の攻撃をくらうと体力が上限の25%以下になり、回復薬を持っているならば、使う。
- 3) 矢を持っていて当たるなら、使う。
- 4) 待機する。敵は隣接してくるので、プレイヤーキャラクターが先制できる。

【敵が3マス以上離れたところにいる場合】

- 1) 2マスの場合の1)と同じ。
- 2) 2マスの場合の3)と同じ。
- 3) 敵に近づく。

【敵が視野内にいない場合】

- 1) 最下層で階段が見えていればそれに近づく。
- 2) アイテムが落ちていて、アイテム数が上限に達していないなら、それに近づく。
- 3) 体力が上限の70%未満で、満腹度は40以上の場合、待機して体力を回復させる。
- 4) 満腹度が70%未満で、食料を持っているならば食べる。
- 5) 満腹度が70%未満で食料を持っていないならば、階段が発見されていればそれに近づき、そうでなければ未知の部屋を探す。
- 6) 全ての部屋を探索済み、またはアイテムを4つ以上拾って階段が発見されていれば、階段に近づく。
- 7) 未知の部屋を探す。

ルールベースプレイヤーを1000試行したところ、クリア率は39.0%であった。人間プレイヤー3人に練習後合計100ゲームしてもらった場合のクリア率は51%であったので、やや低い程度である。挙動を見てみると、広い部屋の中で複数の敵を相手にしてしまっていることが、直接の死因や回復薬の無駄遣いに繋がっていることが特徴的であった。多くの場合、部屋の入口に戻って1対1の状況を作ることが有効であるが、そのようなルールは含まれていない。

6. モンテカルロプレイヤー

本ゲームを知的にプレイするためには、短期的長期的さまざまな種類の能力と判断が求められる。対峙した敵にどう対処するか、マップをどのように巡回するか、経験値稼ぎを行うかどうか、アイテムの良い保持バランス、など考えるべきことは多い。前章最後の観測から、まず我々はルールベースプレイヤーの戦術面の弱さに着目した。

モンテカルロ法を用いる場合、直近行動の長期的な評価を行うことは短期的な評価を行うことに比べて一般に難しい。本ゲームにおいても、「ゲームをクリアできたかどうか」で評価を行うとすると、シミュレーション部分は非常に長くなる。これでエージェントをランダムに動かしすぎると、一手目の良し悪しが結果に反映されにくくなってしまう。そこで一旦、“一般的で長期的な”意思決定を改善することは諦め、“特定の短期的な”意思決定についてのみ改善を図った。

具体的には、提案AIは、敵が視野内にいる場合のみモンテカルロ法を行う。また、ゲーム終了までではなく、「10ターン経過、自分の死亡、階段への到達、または敵が視野内にいなくなった時点で」シミュレーションを打ち切り、結果の評価を行う。その評価関数は、細かな調整を経たものではないが、(敵のダメージ) - (自分のダメージ) + (敵を倒したら5) + (自分が生きていたら50) + (階段を降りたら100, クリアなら1000)とした。

モンテカルロ法にはUCTなど様々な実装法があるが、本論文の実験では深さ2の全幅均等探索を行った。すなわち、現在の状況で取れる全ての行動を列挙し、敵行動を予測し、さらにそこから自分が取れる行動を列挙する。なお本ゲームでは視野内の状態遷移についてランダム性はないので(敵の行動パターンは既知で、攻撃が外れるようなこともない)、min-max探索等は不要である。その上で、各遷移先状態について同じ回数ランダムシミュレーションを行い、前述の評価関数値の平均値が最も高い行動対の初手を選択する。

シミュレーション回数を100とした場合、実行時間は手元のPCで1ゲームあたり平均54秒であった。これはルールベースプレイヤーの場合の平均1秒に比べれば長いですが、想定しているような用途では十分な長さである。

これを1000試行したところ、クリア率は52.2%となった。ゲーム中敵が見えている状況は割合としては1割以下だが、そのときの行動を改善しただけで13ポイントのクリア率向上が見られた(ただし1000試行のみなので、統計的有意性が言えるのは9ポイント程度である)。具体的には、部屋に入って敵を近くに発見した場合、通路側に戻り敵を単体で迎え撃つことができるようになったことが大きい(図4)。また、複数の敵を相手にする場合、杖を使うことでダメージを減らすことができるようになっており、それは平均0.36回から1.16回という杖の使用回数変化にも表れている。その結果、きつい戦闘の発生しやすいつい2階層3階層での死亡数が、計395回から計

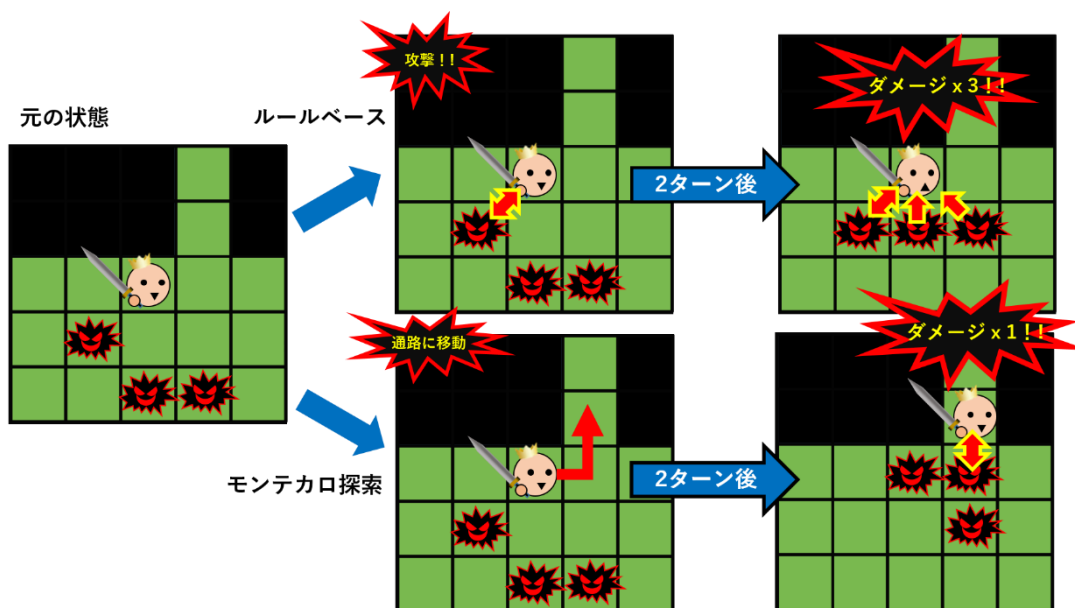


図4：ルールベースプレイヤーとモンテカルロプレイヤーによる行動の違いの例

276回に減少している。

一方で、長期的な観点では、アイテムを無駄遣いしてしまっている状況が見受けられる。使わなくとも良い状況でアイテムを使えば、それ以降で本当にアイテムが必要になったときに窮することになりかねない。具体的には、例えば今のプレイヤーは1対1の状況でも杖を使って戦闘を回避してしまうことがある。これは評価関数の設定に杖を使ったペナルティが含まれていないことや、ランダムシミュレーション中に自分が悪い行動をして死ぬリスクを避けていることが理由であると考えている。

7. 長期的な状態評価

キャラクターは体力・満腹度・レベルといったパラメータと、戦闘用のアイテムを持つ。これらは全て多いほうが良いが、どんな場合に何がどの程度有効なのかは自明ではない。前章では短期的な戦闘を乗り切るためのモンテカルロ法を提示したが、より強いプレイヤーを作成するためには、戦闘をただ乗り切るだけでなく、どうやって乗り切るかを判断させなければならない。すなわち、複数の状態（体力やアイテム数）を評価して最善の状態を目指せるようにならなければならない。

この目的のために行えることはさまざまにあり、例えばAlphaGoで用いられたような、盤面と勝敗のペアを学習データとして集めて教師あり学習し、勝率の推測を行うアプローチなどが有望と考える。この試みについて洗練されたものは本論文執筆時点で完成されていないが、途上の試みを紹介する。

本来、任意の状態からその状態の勝率を推定でき

れば、探索等にとっては最も都合がよい。しかし、未探索部分があったり、敵がいたりするような状況を正しく学習させるのは、入力次元数が高すぎて非常に多くの学習データや複雑な入出力モデルが必要になると考える。そこで本研究ではまず、「階段を降りたタイミング」のみを学習させることにした。複数階層のあるログライクゲームでは、階層を移ると新しいマップ（敵・アイテムを含む）がランダムに与えられるため、その移ったタイミングでの「それ以降の勝率」を推測するためには、その時点でのパラメータや所持アイテムのみを考慮すればよい。すなわち、マップ情報、自分や敵の位置情報などを無視することができる。

そこで我々は、ルールベースプレイヤーで約5万試行を行い、ある階層で階段を降りた時点での（体力、レベル、満腹度+食料×20、回復薬数、矢数、杖数）を6次元の入力、その試行でクリアできたかどうかをバイナリ出力とするような学習データを作成した。1階層については44214、2階層については33372、3階層については23444のデータが得られた。

これを、平均化パーセプトロンを用いて学習した。2階層の階段を降りた場合の得られた学習重みは以下の通りである。

- ・ バイアス項 -1737.9
- ・ 体力 38.8
- ・ レベル 116.6
- ・ 満腹度 -44.7
- ・ 回復薬 202.4
- ・ 矢 134.1
- ・ 杖 83.2

満腹度が勝率に寄与していないのは、十分な経験値稼ぎをルールベースプレイヤーが行っていないためであろう。このパーセプトロンの判定精度は 54.3% に過ぎないが、これはモデルの貧弱さによるものよりは、ゲームのランダム性によるものが大きい。実際、Logistic や SVM など多少良い学習器を用いても判定精度は 55% 程度にしかならなかった。一方で、内積値が 2500 以上の場合は勝率 62.8%、-3000 以下の場合は 34.8% となっているなど、内積の値が勝率予測に役立つことは確認できた。

6章で説明したモンテカルロ法の評価関数を、単純にこのパーセプトロンの出力（勝敗、内積、勝率など）に置き換える試みはいまのところ成功していない。その理由の一つは、シミュレーションがランダムであるために、保持アイテム数の違いなどがかき消されてしまうことであると考えている。二つ目に、我々のモンテカルロ法では 10 ターン後を評価する必要があるのに、学習したのは階段を下りたタイミングのみであり、「残りの部屋数」「拾ったアイテム数」などが考慮されていないことで精度が悪くなっているのだろうと考えている。三つ目に、我々の方法では敵が見えているときのみモンテカルロ法を用いるが、実際には長期的な戦略は「満腹度と照らし合わせて経験値稼ぎを続けるかどうか」「階段が見つかりアイテムもいくつか拾っているが未探索の部屋に行くかどうか」などの敵がいない場合の判断にも重要であるためと考えている。

8. おわりに

本論文では、ローグライクゲームというジャンルを新たなゲーム AI の研究対象として取り上げ、何が面白くまた難しいのかをまとめた後、重要な課題を失わない程度には複雑さを保ちつつ、煩雑でない程度に単純化したルールを提案した。その上で、高度な手法の基盤ないし比較対象とすべく、ルールベースプレイヤーと、モンテカルロプレイヤーを提案し、その性能と挙動を紹介した。さらに、長期的な視野を持ったプレイヤーを作成するための一歩として、状態評価のための教師あり学習を試みた。

本論文のゲームのルールや設定パラメータ、提案した手法、さらにはソースコードなどは、必ずしも現時点では洗練されたものとは言えない。コンピュータプレイヤーを強くするための手法を考案することとは別に、ルールやコードを改良したうえで、多くの人に使ってもらえるようにプラットフォームの公開運営も行っていきたい。

参考文献

- [1] 田中 成俊, 橋山 智訓, 市野 順子, 田野 俊一, ローグライクの AI コンペティション, 第 29 回フェジシステムシンポジウム, 2013, pp.435-440.
- [2] Vojtech Cerny and Filip Dechterenko, Rogue-Like Games as a Playground for Artificial Intelligence – Evolutionary Approach, International Conference on Entertainment Computing (ICEC), 2015, pp.261-271.
- [3] D. J. N. J. Soemers, C. F. Sironi, T. Schuster and M. H. M. Winands, Enhancements for real-time Monte-Carlo Tree Search in General Video Game Playing, IEEE Conference on Computational Intelligence and Games (CIG), 2016, pp.1-8.
- [4] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. Nature 518, 2015, pp.529-533.
- [5] Naoyuki Sato, Kokolo Ikeda and Takayuki Wada, Estimation of Player's Preference for Cooperative RPGs Using Multi-Strategy Monte-Carlo Method, IEEE Conference on Computational Intelligence and Games (CIG), 2015, pp.51-59.
- [6] "Berlin Interpretation - RogueBasin"
http://www.roguebasin.com/index.php?title=Berlin_Interpretation,s
(参照 2017-07-24)
- [7] 藤木 翼, 村山公志朗, 池田 心, ターン制ストラテジーにおける状態評価関数を用いた深き限定モンテカルロの適用, エンターテイメントと認知科学研究ステーション (E&C) 第 8 回シンポジウム, 2014.