

Title	人間らしさを考慮したテストプレイヤを用いる迷路の自動生成と難易度評価
Author(s)	藤平, 啓汰; 池田, 心; シュエ, ジュウシュエン
Citation	ゲームプログラミングワークショップ2021論文集, 2021: 192-199
Issue Date	2021-11-06
Type	Conference Paper
Text version	publisher
URL	http://hdl.handle.net/10119/18235
Rights	<p>社団法人 情報処理学会, 藤平 啓汰, 池田 心, ゲームプログラミングワークショップ2021論文集, 2021, 192-199. ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.</p>
Description	第26回ゲームプログラミングワークショップ(GPW-21), 2021-11, オンライン開催

人間らしさを考慮したテストプレイヤーを用いる 迷路の自動生成と難易度評価

藤平 啓汰^{1,a)} シュエ ジュウシュエン^{1,b)} 池田 心^{1,c)}

概要: ロールプレイングゲーム (RPG) のマップデザインにおいて、プレイヤーの進行方向を適切に誘導することは重要な課題のひとつである。プレイヤーに対する誘導があまりに露骨なものだと、プレイヤーに不快感を与えゲームの面白さを損ねてしまう可能性があるからである。そのため、RPG では、プレイヤーの行動傾向に応じて自然に誘導するマップが求められることが多い。本稿では、自然に誘導する RPG マップの自動生成に向けた研究の初期段階としてまず、人間プレイヤーの分岐選択傾向に基づいた難易度の迷路を自動生成する。具体的には、分岐選択傾向を教師あり学習によって調査し、その傾向を模倣したテストプレイヤーを作成、そのテストプレイヤーを用いて「人間プレイヤーだからこそ難しい迷路」を生成する。被験者実験の結果、テストプレイヤーを用いて自動生成した難しい迷路は、実際の人間プレイヤーに対しても難しい迷路であることが示された。

キーワード: 自動生成, 迷路, 難易度, 誘導, RPG

Procedural Maze Generation with Test Player Considering Human-likeness

KEITA FUJIHIRA^{1,a)} CHU-HSUAN HSUEH^{1,b)} KOKOLO IKEDA^{1,c)}

Abstract: In the design of maps for role-playing games (RPG), guiding players properly is one of the big challenges. Too explicit guidance, such as simple single paths and flashing marks, may harm players' play experience. Therefore, it is better to have maps that guide players according to their behavioral tendencies. In this paper, as the first step of our research, we aim to know the difficulty of mazes from human players' tendencies and generate mazes accordingly. We investigate the tendencies by supervised learning and use the learned model to create a test player considering human-likeness for maze generation. In evaluation experiments, the results show that our method succeeded in generating mazes with difficulty matching human players' tendencies.

Keywords: Procedural content generation, maze, difficulty, guidance, RPG

1. はじめに

ゲーム分野ではこれまで、強いコンピュータプレイヤーの作成等に Artificial Intelligence (AI) が用いられてきた。さらに近年では、Procedural Content Generation (PCG) と呼ばれるゲームコンテンツの自動生成技術の研究にも

AI が応用され、学術・産業界を問わず注目を集めている。PCG の研究は、横スクロールアクション [1] や first-person shooter (FPS) [2] といった有名ジャンルを対象にすることが多い。一方、日本で親しまれているファイナルファンタジーやドラゴンクエストに代表されるロールプレイングゲーム (RPG) を対象とする研究は比較的少ない。

RPG においてプレイヤーは、最後の大ボスを倒すなどストーリー上の最終目的以外にも、ダンジョン探索やアイテム探しといった多様な目的を持つ。その目的を達成しようとするなかで、例えばダンジョン内で長時間迷ってしまったり、逆にレアアイテムが簡単に手に入ってしまったたりす

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology, Nomi,
Ishikawa 923-1211, Japan

a) keita.fujihira@jaist.ac.jp

b) hsuehch@jaist.ac.jp

c) kokolo@jaist.ac.jp

ると、ゲームの面白さが損なわれる可能性がある。そのため、ゲームデザイナーはプレイヤーが迷いすぎないように助けたり、アイテムからあえて遠ざけたりするために、マップ構造や敵・アイテムの配置方法によりプレイヤーを誘導することも多い。しかし、その誘導が一方的で露骨なものであれば、RPGの醍醐味である探索の高揚感を喪失させ、ゲームの面白さを損ねてしまう可能性がある。RPGでは、プレイヤーが「自分は能動的に行動している」と感じつつも、実はゲームデザイナーの意図に沿った行動をとる、つまりプレイヤーが自然に誘導されるマップがしばしば求められる。通常、そのような複雑な特性を持つマップの作成は、ゲームデザイナーによって手作業で行われるため、ゲーム開発における高コストな作業とされている。

本研究の主目標は、プレイヤーを自然に誘導するRPGマップを自動生成することである。RPGマップの複雑さを考慮し、研究の初期段階では敵やアイテムを含まず、壁と通路のみで構成される2次元迷路マップを生成対象とする。そのうえで、人間プレイヤーの分岐選択傾向に基づいた難易度の迷路を自動生成することを本稿の目的とする。まず、教師あり学習により分岐選択確率予測モデルを構築し、そのモデルから人間プレイヤーの傾向を調査する。次に、予測モデルをベースとする人間らしさを考慮したテストプレイヤーを作成し、迷路の自動生成に用いる。最後に、自動生成した迷路を評価するために被験者実験を実施する。

2. 関連研究

本章では、「迷路の自動生成」「テストプレイヤーを用いるコンテンツの自動生成」に関する研究をいくつか紹介する。

2.1 迷路の自動生成

一般的な迷路生成アルゴリズムとして、穴掘り法 [3] や壁伸ばし法 [4]、棒倒し法 [5] などが挙げられる。難易度に焦点を当てた場合には、Genetic Algorithm (GA) などのメタヒューリスティックが適用されることも多い [6][7]。Adamsら [6]は、Cellular Automaton (CA) を用いて迷路を自動生成し、CAのルールを進化させるために正解路長や行き止まりの数を評価値とするGAを適用した。また、Kwieceń [7]は、正解路長や方向転換数、分岐点数などを評価値とするCockroach Swarm Optimizationを用いて、通路構造が複雑な難しい迷路を自動生成した。

正解路長や分岐点数などの特徴が迷路の難易度に影響していると報告する研究は少なくない [8]。例えば、分岐点が多いほどプレイヤーの迷いが発生し得る地点が増えるため、難易度に関連すると報告されることは合理的であると考えられる。しかし、人間だからこそ迷ってしまう、あるいは迷うはずのない分岐点も迷路には存在するはずであり、その点を踏まえて難易度を調整すべきだと考える。本稿では、そのような人間らしさを考慮した難易度の迷路を生成する。

2.2 テストプレイヤーを用いるコンテンツの自動生成

PCGには様々なアプローチが存在するが、一般的なアプローチとして「生成したコンテンツを何らかの指標で評価し、その中から評価値の高いものを選んだり、より評価値が高くなるよう変化を加えたりする手法」がよく知られている。このアプローチの評価部分では、2.1節で紹介した研究のようにコンテンツの特徴から評価する方法のほか、テストプレイヤーによるシミュレーション結果から評価する方法がある。テストプレイヤーの実装方法は、ヒューリスティック [9] やモンテカルロ法 [10]、A*アルゴリズム [11]、強化学習 [12] など様々である。

誤解を恐れずに言えば、PCGの主目標は「人間プレイヤーが楽しめるコンテンツを生成することであり、テストプレイヤーが人間プレイヤーからかけ離れた行動をとると、適切な評価結果を得られない可能性が高くなる。そのため、例えば、吉田ら [9]は「人間プレイヤーは、正確で素早い操作が苦手である代わりに、大局的な判断によって余裕を持った回避を行うことが多い」という観点からいくつかの工夫を施し、シューティングゲームのテストプレイヤーに人間らしさを取り入れステージの自動生成に用いた。ただし、このように人間らしさを考慮したテストプレイヤーを用いるPCGの研究は、我々の知る限り比較的少ない。その理由として、(1)対象ゲームによっては単純なテストプレイヤーでさえも作成が困難、(2)対象ゲームにおける人間らしさの抽出が困難であることが挙げられる。本稿の場合、詳細は3章で述べるが、対象ゲームがシンプルな迷路であるため(1)については容易である。(2)については、人間プレイヤーの分岐選択に着目し、教師あり学習によって人間らしさを抽出する。

3. 人間プレイヤーの分岐選択傾向の調査

迷路の種類は非常に多く、次元数や正解路の一意性、外観など様々な観点で分類できる。本稿では、 $N \times N$ マスで構成される2次元格子状の全域木型迷路を対象とする。本章では、対象迷路の設定や教師あり学習用のデータを収集するため実施した被験者実験の概要について述べた後、学習内容とその結果について報告する。

3.1 学習データ収集のための被験者実験

本稿が対象とする迷路の特徴を以下にまとめる。図1はその例である。

- 迷路はディスプレイ上に映り、プレイヤーはキーボード入力による操作を行う。
- プレイヤーが辿ってきた道のりは、ディスプレイ上に可視化されない。
- 迷路サイズは、 31×31 (S)、 41×41 (M)、 51×51 (L)のいずれかである。
- 制限時間があり、制限時間を過ぎてもゴールに辿り着

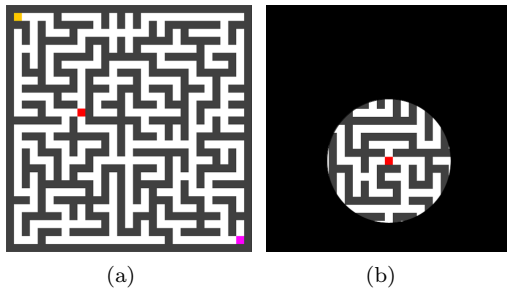


図 1 Sサイズの迷路の例：(a) Wide 迷路, (b) Narrow 迷路. 左上隅の黄色マスはスタート, 右下隅のピンク色マスはゴール, 赤色マスはプレイヤーの現在地をそれぞれ示す.

いていない場合, プレイヤはゲームオーバーとなる. 制限時間は迷路サイズによって異なる. Sサイズは80秒, Mは100秒, Lは150秒である.

- 通行可能なマスと通行不可能なマスのみで構成され, それぞれを通路マス, 壁マスと呼ぶ.
- 迷路の外周部分は, 壁マスで構成される.
- スタートは迷路の左上隅, ゴールは右下隅に設置する.
- プレイヤの目的は, スタートからゴールに辿り着くことである.
- プレイヤは1回の行動につき1マス分のみ, 現在地の上下左右に隣接するいずれかの通路マスに移動できる.
- 迷路左上の頂点座標を(1, 1)とすると, 通路マスの配置候補となるのは, x 座標, y 座標, あるいはその両方が偶数となるマスである. 言い換えれば, x, y 座標がどちらも奇数の場合, そこは必ず壁マスとなる.
- ループ(閉路)は存在しない. 到達できない通路マスも存在しない. 言い換えれば, プレイヤが同じ通路マスを2回以上訪れない限り, スタートとゴールを結ぶ経路(正解路)は必ず一意に定まる.
- プレイヤは, 以下のいずれかの視界状態を持つ.
 - Wide: 図1(a)のように迷路全体が見える状態.
 - Narrow: 図1(b)のようにプレイヤーを中心とする円の内部のみ見える状態. 円の直径は, 迷路の一边の長さの半分である. 例えば, Sサイズの場合, 円の直径は $31/2 = 15.5$ となる.

この設定のうえで, 実験に使用する迷路は穴掘り法 [3] と壁伸ばし法 [4], 棒倒し法 [5] のいずれかを用いて生成した. 実験に参加した被験者は20人で, 各被験者はサイズや視界状態を変更しながら計21個の迷路をプレイした.

3.2 教師あり学習による分岐選択確率の予測

本節では, 分岐点における人間プレイヤーの分岐選択確率を予測する教師あり学習について述べる. 学習内容を述べる前に, 本稿における「未確定」と「分岐点」を定義する. この定義はWide迷路とNarrow迷路に共通である. どちらの場合も, Narrow迷路の視界状態を用いて定義する.

プレイヤーの現在地からある方向に注目したとき, 「その先

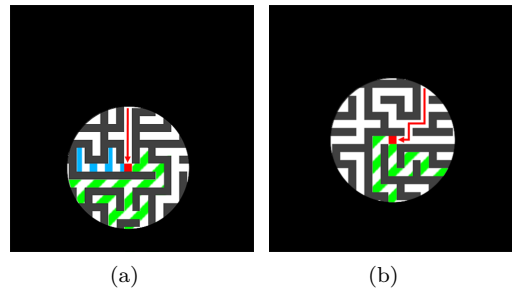


図 2 分岐点となる通路マスとそうでない通路マスの例：(a) 分岐点ではない, (b) 分岐点

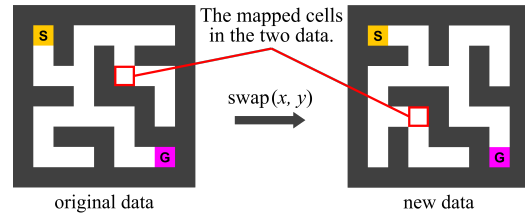


図 3 学習データの水増し方法

に到達可能な通路マスが, 1つでも Narrow 視界範囲外にある」ならば, その通路は「未確定」とする. 例えば, 図2(a)でプレイヤーの右方向に注目すると, その先(緑斜線の部分)が行き止まりなのか, ゴールへ続くのか判断できないため, 右方向に続く通路は未確定となる. 一方, 左方向の先(青縦線の部分)は, その通路を構成する通路マスが全て見えるため未確定ではない. なお, この「未確定」の判定は, 現在の視界で得られる情報のみで行われることに注意されたい. つまり, これまでのプレイヤーの移動によって得られた情報を覚えていれば行き止まりと分かる場合でも, 上記条件を満たしていれば未確定として扱う.

これを踏まえたうえで, 「プレイヤーが進行してきた方向の通路を除き, 未確定な通路が2つ以上接続されている通路マス」を「分岐点」と定義する. 図2(b)において, プレイヤの現在地は, 2つの未確定な通路と接続されているため分岐点となる.

3.2.1 学習内容

この教師あり学習では, 人間プレイヤーの分岐選択について予測する. 目的変数となるのは, 3.1節の実験データから算出した被験者の分岐選択割合である. 例えば, 右と下方向に進行可能な分岐点において, 20人中16人のプレイヤーが右方向を選んだ場合, 右方向の選択割合は $16/20 = 0.8$ となる. この選択割合を学習するため, 著者はこれまで15個の特徴量を用いていた [13]. 本稿では, 新たに特徴量を追加することで予測精度の向上を図る. 学習に用いる特徴量の詳細については付録 A.1 を参照されたい.

実験データより, Wide迷路は147個, Narrow迷路は197個の学習データをそれぞれ得ることができた. これらの学習データは, プレイヤが分岐点を初めて訪れたときのデータのみを収集対象とした. さらに, 図3のように x

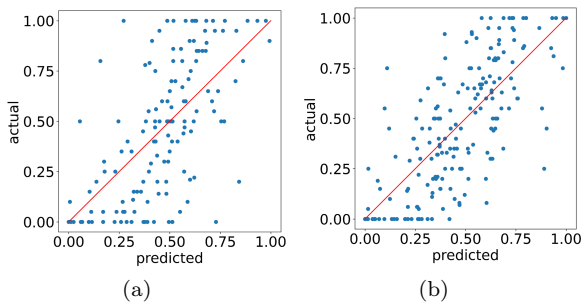


図 4 予測モデルによる分岐選択確率の予測結果（横軸：正規化後の予測値，縦軸：実測値）：(a) Wide 迷路，(b) Narrow 迷路

座標と y 座標を入れ替えたデータを作成することで学習データ数を 2 倍にした。学習には決定木ベースの勾配ブースティングモデルである LightGBM[14] を、精度検証には leave-one-out 交差検証を用いた。

また、予測モデルは 1 つの分岐点に関する複数の選択割合を別々に予測するため、予測値の和が 1.0 になるとは限らない点に注意されたい。本節と後述する 4.1 節では、1 つの分岐点に関する予測対象が n 個あるとき、予測値の和が 1.0 になるように次式を用いて各予測値 p_i を正規化し、確率分布へと変換している。

$$f(p_i) = \frac{p_i}{\sum_{k=1}^n p_k} \quad (1)$$

3.2.2 学習結果

図 4 は、横軸に予測モデルによる予測値を正規化した値、縦軸に実験データから得た実測値をとる散布図である。表 1 は、以前の結果 [13] も含めた、予測値と実測値の二乗平均平方根誤差 (RMSE)、相関係数をまとめたものである。RMSE は小さいほど、相関係数は大きいほど予測精度が高いことを意味する。したがって、Wide, Narrow 迷路ともに、特徴量を追加したことによる改善、および予測値の正規化を行ったことによる改善が表 1 より確認できる。

人間プレイヤーの分岐選択傾向を調査するため、予測モデルから特徴量重要度を算出した。特徴量重要度とは、学習に用いた特徴量が予測にどの程度貢献しているのか数値化したものである。

Wide 迷路すなわち視界制限のない迷路において、最も重要度が高い特徴量は area (重要度 0.5) であった。特徴量の詳細は付録 A.1 で述べているため省略するが、area は「視界範囲に関係なく、予測方向の先にある到達可能な全ての通路マスの個数」を示す特徴量である。area の値が小さければ、続く通路が行き止まりであるか、ゴールに繋がっているのか判断しやすくなる。一方、area の値が大きいと、特に制限時間がある中では、その判断が比較的難しくなるはずである。そのため、area の値が小さい不正解路に続く方向は選択確率が低く、値が大きい方向は正解・不正解にかかわらず選択確率が高くなるといった理由により、area が重要視されたと推測する。

Narrow 迷路について結論から述べると、人間プレイヤー

表 1 旧特徴量 [13]，新特徴量を用いた各予測モデルの精度比較

視界状態	予測モデル	RMSE	相関係数
Wide	旧特徴量・正規化なし	0.29	0.55
	新特徴量・正規化なし	0.28	0.58
	新特徴量・正規化あり	0.26	0.66
Narrow	旧特徴量・正規化なし	0.27	0.53
	新特徴量・正規化なし	0.23	0.70
	新特徴量・正規化あり	0.21	0.74

には (1) 上や左方向よりも下や右方向を選ぶ傾向と (2) 下と右の両方向が選択可能な場合、直進方向を選ぶ傾向があると考えられる。図 5 は、形状の異なる分岐点ごとにまとめた、被験者たちの各方向の選択割合である。読みやすさの観点から割合が 0.5 未満のデータを、信頼性の観点からサンプル数が 20 未満のデータをそれぞれ除外している。

図 5 より、下や右方向の選択割合が高いことが分かるが、これはゴールが迷路の右下隅に固定されているためだと推測できる。また、Narrow 迷路において最も重要度が高い \cos_goal (重要度 0.35) は、予測方向が右や下であれば値が大きくなり、そうでなければ値が小さくなる特徴量である。つまり、 \cos_goal が最も重要であるとする結果は、予測モデルが人間プレイヤーの傾向 (1) を十分に反映した結果であると言える。

さらに、図 5 において、下と右の両方向が選択可能な場合 (赤矢印) に注目すると、いずれも直進方向の選択割合が高いことが分かる。Narrow 迷路で 2 番目に重要度が高い特徴量は $is_straight$ (重要度 0.15) であり、これも予測モデルが傾向 (2) を十分に反映した結果であると言える。

4. テストプレイヤーによる迷路の難易度評価

迷路の生成・難易度評価には、人間らしさを考慮したテストプレイヤーを用いる。本章では、テストプレイヤーの作成方法と難易度の定義、難易度の評価方法について述べる。また、テストプレイヤーの作成について、視界状態を Narrow と仮定していることに注意されたい。

4.1 人間らしさを考慮したテストプレイヤー

テストプレイヤーの行動は、その現在地となる通路マスが分岐点 (3.2 節で定義) であるかどうかで異なる。

通路マスが分岐点ではない場合の行動をアルゴリズム 1 に示す。アルゴリズム 1 では迷路を、スタートを根とする木に見立てたとき、テストプレイヤーの現在地の親ノードを「前のマス」、子ノードを「次のマス」と呼ぶ。また、テストプレイヤーは、図 6(a) のようにゴールが見えている場合を除き、図 2(a) の青縦線部分のような未確定でない通路に含まれる通路マスには移動しない。アルゴリズム 1 において特に重要なのが 8・9 行目であり、条件 (1) と (2) に該当する状況の例を図 6(b) と 6(c) にそれぞれ示す。これは人間プレイヤーの行動により近づけるため導入したヒューリス

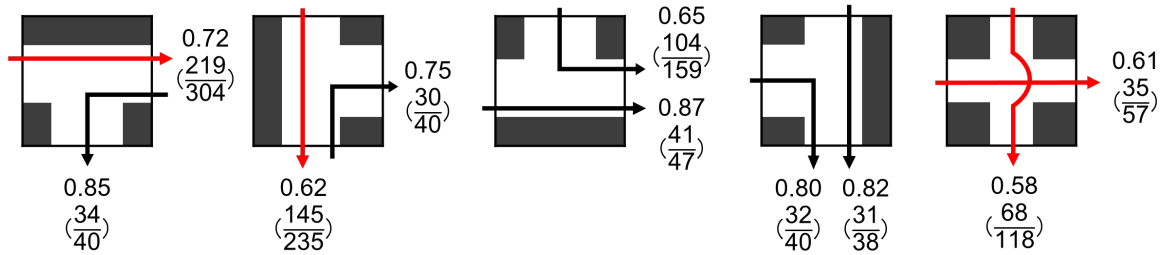


図 5 Narrow 迷路を対象とし、形状の異なる分岐点ごとにまとめた被験者の分岐選択割合

アルゴリズム 1 分岐点ではない通路マスにおける行動

```

Input: state : テストプレイヤーの状態
1: if state = 通路を進む then
2:   次のマスに移動する
3: else if state = 最後に通過した分岐点まで戻る then
4:   前のマスに移動する
5: end if
6: if (未確定でない通路を除いたうえで) 分岐点に着いた then
7:   state ← 分岐点にいる
8: else if (1) 外周に隣接するマスゴールとは逆方向に進んだ
or (2) 続く通路の先が全て行き止まりであることが、視界範囲
内 で判明した then
9:   state ← 最後に通過した分岐点まで戻る
10: else
11:   state ← 通路を進む
12: end if
    
```

ティックである。例えば、条件 (1) について、対象迷路はゴールが迷路の右下隅に設置されループも含まないため、外周に隣接する通路で上や左方向に進むことはその通路が不正解路であることを意味する。被験者実験でも、そのような通路を突き進む被験者はあまり見られなかった。

分岐点におけるテストプレイヤーの行動は、その分岐点を初めて訪れたかどうかで異なる。ある分岐点を初めて訪れた場合、3.2 節の予測モデルを用いて全ての未確定な通路の選択確率を予測し^{*1*}、その確率分布に従い進む先をランダムに選択する。その分岐点を訪れるのが 2 回目以降の場合には、未訪問かつ未確定の通路のみを予測対象としたうえで選択確率を予測し進む。

ただし、不正解路上^{*3}のある分岐点で 2 回目以降の予測を行う場合に限り、「3.2.1 項で述べた正規化処理前の予測値が 0.5 未満ならば、その予測値を 0.0 に変換する」という条件を加える。この理由について、例えば人間プレイヤーは、もし不正解の方向に進んだとしても、その探索結果から迷路構造の一部を把握することができる。そのため、次回以降の分岐選択時、人間プレイヤーなら「こちらは不正解路っぽい」など見当をつけ、未訪問の通路であっても無視

*1 図 6(a) のようにゴールが見えている場合には、予測ステップを省きゴールへ続く通路を進む。

*2 分岐点進入前にテストプレイヤーがいた方向は予測対象から除く。

*3 正解路か不正解路であるかは本来プレイヤーには区別できないが、本稿で用いたテストプレイヤーにはこの情報を与えている。

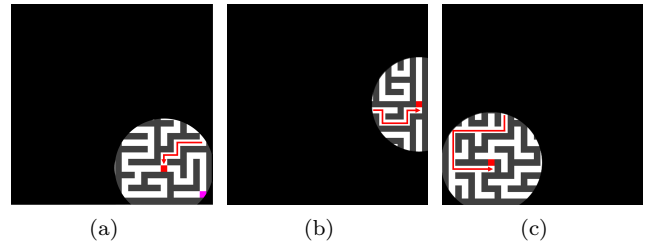


図 6 テストプレイヤーの行動と状況の例：(a) ゴールが見えている状況、(b) 右側の外周に隣接する通路を上方向に動いた状況、(c) 続く通路の先にある全ての行き止まりが見えた状況

することがあるのではないかと推測した結果、この条件を適用することとした。また、この条件を正解路上の分岐点に適用すると、テストプレイヤーがゴールに辿り着かない可能性があるため、不正解路上の分岐点のみに適用した。

4.2 迷路の難易度とその評価方法

迷路の難易度は、4.1 節のテストプレイヤーが迷路を解いたときの歩数をもとに評価する。ある迷路のスタートからゴールに到達するまでに必要な最短歩数を $n_{shortest}$ 、テストプレイヤーがその迷路を解いたときの歩数を n_{total} 、その差を $n_{extra} = n_{total} - n_{shortest}$ としたとき、 n_{extra} が大きいほど難しい迷路であると定義する。また、テストプレイヤーの行動にはランダム性があるため、1 試行ではなく複数回試行したときの n_{extra} の平均値と標準偏差をもとに難易度を評価する。本稿では、以下の 5 つの難易度を用意した。

- $easy_{lowSD}$ (n_{extra} の平均値：小、標準偏差：小)
- $moderate_{lowSD}$ (n_{extra} の平均値：中、標準偏差：小)
- $moderate_{highSD}$ (n_{extra} の平均値：中、標準偏差：大)
- $difficult_{lowSD}$ (n_{extra} の平均値：大、標準偏差：小)
- $difficult_{highSD}$ (n_{extra} の平均値：大、標準偏差：大)

5. 迷路評価の被験者実験

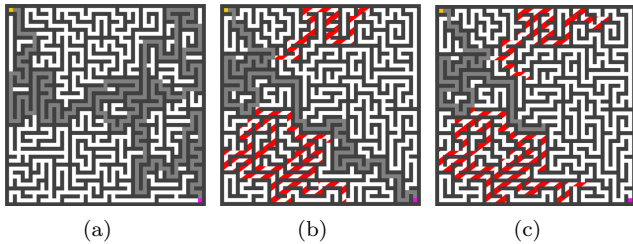
テストプレイヤーの評価した難易度が、人間プレイヤーが実際に解いたときの結果に適合するのかが検証するため被験者実験を実施した。本章では、その実験結果を報告する。

5.1 実験内容

本実験に参加した被験者は 10 人である。公正な評価の

表 2 迷路を解いた順番と被験者による n_{extra} の平均値と標準偏差 (SD)

easy _{lowSD}			moderate _{lowSD}			moderate _{highSD}			difficult _{lowSD}			difficult _{highSD}		
順番	mean	SD	順番	mean	SD	順番	mean	SD	順番	mean	SD	順番	mean	SD
5	200	194	1	71	43	3	53	113	4	299	173	2	257	180
9	72	131	10	241	197	6	243	206	7	364	123	8	231	162
14	69	44	12	285	114	11	195	209	13	371	121	17	447	162
18	77	116	19	220	137	15	135	214	16	406	150	20	368	200
21	175	151	23	228	140	24	232	201	22	262	133	25	265	234
26	48	47	29	178	152	27	228	180	28	517	159	30	104	173
33	27	25	32	235	42	31	314	291	34	374	77	35	140	184
Avg.	95	101	Avg.	208	118	Avg.	200	202	Avg.	307	134	Avg.	259	185

図 7 自動生成した迷路の例: (a) easy_{lowSD}, (b) difficult_{lowSD}, (c) ある被験者が (b) の迷路を解いたときの軌跡

ため、3.1 節の被験者とは異なる被験者を採用した。本実験で使用する迷路は、サイズを L, 視界状態を Narrow に設定し、穴掘り法 [3] を用いて自動生成した。実験の準備段階で 3 万個の迷路を生成し、各迷路を 10 回ずつテストプレイヤーが解いた。その結果、 n_{extra} の平均値が $[0, 50)$ であれば難易度を easy, $[150, 250)$ であれば moderate, $[350,)$ であれば difficult とした。さらに、それらの中で標準偏差が上位 7 個を highSD, 下位 7 個を lowSD とした。

実験で使用した easy_{lowSD} と difficult_{lowSD} の迷路を図 7(a), 7(b) にそれぞれ示す。色付きの通路マスはテストプレイヤーの軌跡を示しており、灰色マスは正解路中の軌跡、赤斜線マスは不正解路中の軌跡である。被験者は計 35 個の迷路を表 2 に記す順番で解いた。図 7(c) は、図 7(b) の迷路をある被験者が解いたときの軌跡である。これは実験結果の中でも、テストプレイヤーと人間プレイヤーの軌跡が特に一致した例である。

5.2 実験結果

被験者による n_{extra} の平均値と標準偏差を表 2 の mean, SD の列にそれぞれ示す。まず平均値について、おおそ easy < moderate < difficult となり、テストプレイヤーの評価と人間プレイヤーが解いた結果は適合することが示唆された。また、lowSD 迷路について Student の t 検定 ($p < 0.05$) を行った結果、moderate_{lowSD} は easy_{lowSD} よりも平均値が大きく ($p = 0.0016$), difficult_{lowSD} は moderate_{lowSD} よりも平均値が大きく ($p = 0.0083$), いずれも統計的に有意差があることが示された。ただし、これらの平均値の中には、テストプレイヤーによる評価区間から外れているもの

もあり、easy_{lowSD} の平均値は区間 $[0, 50)$ より 45 程度大きく、difficult_{lowSD} と difficult_{highSD} の平均値は区間 $[350,)$ より 43, 91 程度それぞれ小さくなった。この理由は 5.3 節にて考察する。

次に、lowSD と highSD 迷路の n_{extra} の標準偏差の差に注目する。lowSD と highSD 迷路の標準偏差の平均値について Student の t 検定 ($p < 0.05$) を行った結果、moderate_{highSD} は moderate_{lowSD} よりも標準偏差の平均値が大きく ($p = 0.0142$), difficult_{highSD} は difficult_{lowSD} よりも標準偏差の平均値が大きく ($p = 0.0057$), いずれも統計的に有意差があることが示された。

5.3 実験結果の考察

5.2 節の結果より、テストプレイヤーの行動は人間プレイヤーの傾向を十分に反映し、その傾向に基づく難易度の迷路を生成できていることが分かる。また、標準偏差が大きい迷路を複数の人間プレイヤーが解いた場合、同じ迷路であっても各プレイヤーは全く異なる難易度を体験してしまう可能性が高くなる。そのため、多くの場合において、標準偏差は小さいほうが望ましいと考える。テストプレイヤーによる標準偏差の評価と人間プレイヤーが解いた結果が大まかに適合したことは良い結果と言える。

しかし、人間プレイヤーの結果に反して、テストプレイヤーが難しさを (1) 過大評価していた、あるいは (2) 過小評価していた迷路もいくつか見られた。まず、(1) と (2) に共通する理由として、予測モデルに改善の余地があることが挙げられる。次に (1) について、これは制限時間に起因していると考えられる。被験者実験では、制限時間を設けていたため、特に難しい迷路の場合、ゴールまで辿り着けずゲームオーバーになる被験者も少なからず見られた。一方、テストプレイヤーが迷路を解く際には制限時間を設けず、必ずゴールまで辿り着くという条件のもと n_{extra} を計算していた。被験者とは異なり、テストプレイヤーは際限なく迷路内を探索できてしまったことで、(1) が生じたと推測する。

(2) については、人間プレイヤーの「思い込み」「勘違い」といった部分がテストプレイヤーに十分反映されていないことが理由であると考えられる。例えば、非常に長い正解路を持

つ迷路を人間プレイヤーが解いているとする。プレイヤーは正解路を進んでいる最中、長時間ゴールに辿り着けなければ「これは不正解路なのだろうか」と考え通路を引き返し、その分 n_{extra} が増加する可能性がある。一方、テストプレイヤーにはそういった迷いがないため、(2) のような食い違いが生じたと推測する。分岐選択のほかにも、「思い込み」「勘違い」といった人間らしさを取り入れることが (2) の解消に繋がると期待できる。

6. おわりに

本稿では、プレイヤーを自然に誘導する RPG マップの自動生成に向けた研究の初期段階としてまず、人間プレイヤーの分岐選択傾向に基づいた難易度の迷路を自動生成した。人間プレイヤーの傾向は教師あり学習を用いて調査し、その傾向を取り入れたテストプレイヤーを作成、迷路の自動生成に用いた。迷路評価の被験者実験では、テストプレイヤーが評価した難易度と実際に人間プレイヤーが解いた結果は適合することが示唆された。

提案手法は、特定の難易度の迷路が生成されるまでに多くの時間を要してしまう可能性があり、その点が課題として残る。今後は特定の難易度の迷路を適時生成できるよう、生成効率の改善を目指す。また、既存の難しい迷路生成手法との比較も行う予定である。さらに、今回得た知見をもとに、敵やアイテムなどを含むより複雑なゲームに提案手法を適用していきたい。

謝辞

本研究は JSPS 科研費 JP18H03347, JP20K12121 の助成を受けたものです。

参考文献

- [1] Volz, V., Schrum, J., Liu, J., Lucas, S. M., Smith, A., and Risi, S.: Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. In 2018 Genetic and Evolutionary Computation Conference (GECCO 2018), pp. 221-228 (2018).
- [2] Cardamone, L., Yannakakis, G. N., Togelius, J., and Lanzi, P. L.: Evolving Interesting Maps for a First Person Shooter. In 2011 European Conference on the Applications of Evolutionary Computation (EvoApplications 2011), pp. 63-72 (2011).
- [3] Algoful, 迷路生成法 (穴掘り法), <https://algoful.com/Archive/Algorithm/MazeDig> (accessed 2021-10-01).
- [4] Algoful, 迷路生成法 (壁伸ばし法), <https://algoful.com/Archive/Algorithm/MazeExtend> (accessed 2021-10-01).
- [5] Algoful, 迷路生成法 (棒倒し法), <https://algoful.com/Archive/Algorithm/MazeBar> (accessed 2021-10-01).
- [6] Adams, C., and Louis, S.: Procedural Maze Level Generation with Evolutionary Cellular Automata. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI 2017), pp. 1-8 (2017).
- [7] Kwiecień, J.: A Swarm-Based Approach to Generate Challenging Mazes. *Entropy*, 20(10), 762 (2018).
- [8] 横田和幸, 船瀬新王, 藤原清悦, 内匠逸: ヒトを対象にした迷路課題の難易度を定量的に決定するパラメータの検討. *生体医工学*, 57(2-3), pp. 58-67 (2019).
- [9] 吉田友太, 池田心: 遺伝的アルゴリズムと人間らしい AI プレイヤーを用いた 2D シューティングゲームのステージ生成. *研究報告ゲーム情報学 (GI)*, 2020-GI-43(23), pp. 1-8 (2020).
- [10] 森永剛志, 池田心: 上海ゲームにおける面白いインスタンス生成. *研究報告ゲーム情報学 (GI)*, 2020-GI-43(21), pp. 1-8 (2020).
- [11] Shu, T., Liu, J., and Yannakakis, G. N.: Experience-Driven PCG via Reinforcement Learning: A Super Mario Bros Study. In 2021 IEEE Conference on Games (CoG 2021), (2021).
- [12] Gisslen, L., Eakins, Andy., Gordillo, C., Bergdahl, J., and Tollmar, K.: Adversarial Reinforcement Learning for Procedural Content Generation. In 2021 IEEE Conference on Games (CoG 2021), (2021).
- [13] 藤平啓汰, 池田心: 2D 迷路における行動選択の予測と自然なプレイヤー誘導への応用. *研究報告ゲーム情報学 (GI)*, 2021-GI-46(4), pp. 1-6 (2021).
- [14] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pp. 3147-3155 (2017).

付 録

A.1 教師あり学習に用いた特徴量

教師あり学習に用いた特徴量を以下に記す。著者の以前の論文 [13] から新たに追加された特徴量には下線を引く。方向を表す特徴量は、右なら 0, 下なら 1, 左なら 2, 上なら 3 に変換し、数値化している。

- **maze_size**: 迷路のサイズ (S, M, または L)。S なら 0, M なら 1, L なら 2 に変換し、数値化している。
- **x, y**: 分岐点の x, y 座標。
- **ent**: 分岐点へ進入する前にプレイヤーがいた通路マスの方向。
- **exit**: 予測対象となる通路マスの方向。
- **exit_up, exit_down, exit_left, exit_right**: 分岐点から見たとき、ent を除く各方向に未確定な通路があるかどうか。ある方向が True で、その方向を 1 回以上訪れた場合でも、その方向は True のままとする。
- **dist_edge_up, dist_edge_down, dist_edge_left, dist_edge_right**: 分岐点と各方向にある外周との直線距離。ただし、dist_edge_up と dist_edge_left は、特徴量 x, y と同じ値をそれぞれ持つため、あまり意味のない特徴量である。
- **is_straight**: ent と exit が直進方向にあるかどうか。
- **straight_depth**: is_straight が True の場合、視界範囲に関係なく*4, exit の先で直進できなくなるまでの

*4 Narrow 迷路では、本来プレイヤーは視界範囲外の情報を取得できないが、straight_depth と area, area_sum を計算する際には視界範囲外の情報も用いている。

通路マスの個数. `is_straight` が False なら 0 となる.

- **area**: 視界範囲に関係なく, `exit` の先にある到達可能な全ての通路マスの個数. 例えば, 図 A.1 中の B を分岐点とし, 下方向の選択確率を予測する場合, 緑斜線で塗りつぶされた通路マス (ゴールを含む) の個数が `area` の値となる.
- **area_sum**: 分岐点から見たとき, `ent` を除く各方向の先に続く通路マスの個数. 例えば, 図 A.1 中の B を分岐点とし, それに関連する選択確率を予測する場合, 緑斜線と青縦線で塗りつぶされた通路マス (ゴールを含む) の個数が `area_sum` の値となる.
- **num_branch**: 視界状態を Narrow と仮定し, 視界範囲内にある分岐点の個数 (現在いる分岐点は除く).
- **dist_start**: スタートから分岐点までの最短歩数.
- **avg_step**: 全被験者の分岐点到達時点の歩数の平均値. テストプレイヤーが迷路をプレイする場合, 分岐点到達時点の合計歩数を `avg_step` の値とする. 迷路や分岐点の情報から作成される他の特徴量とは異なり, 本特徴量はプレイヤーの歩数をもとに作成される. 歩数が多いほど迷路内を探索し, その構造を把握している可能性が高くなる. それが分岐選択に影響するだろうという観点から, 本特徴量を用いることとした.
- **cos_goal**: 分岐点から見たゴールの方向と予測対象通路マスの方向のなす角度に対する $\cos\theta$. 図 A.2 に例を示す. 図 A.2 中の B を分岐点, G をゴールとし, `exit` が右方向なら `cos_goal` の値は $\cos 30^\circ \approx 0.86$, 下方向なら $\cos 300^\circ = 0.50$, 左方向なら $\cos 210^\circ \approx -0.86$, 上方向なら $\cos 120^\circ = -0.50$ となる.
- **cos_start**: 分岐点から見たスタートの方向と予測対象通路マスの方向のなす角度に対する $\cos\theta$.
- **general_direction**: 視界状態を Narrow と仮定し, `exit` の先に続く通路の大きな方向. この特徴量を計算するうえで, 分岐点を根, `exit` の先に続く視界範囲内の通路マスを節や葉とする迷路の部分木を考える. また, 特徴量の候補値を決めるための変数 D_{LR} と D_{UD} を導入し, アルゴリズム 2 を用いて D_{LR} と D_{UD} を計算する. D_{LR} が正数であれば右方向を, 負数であれば左方向を候補値とする. また, D_{UD} が正数であれば下方向を, 負数であれば上方向を候補値に追加する. D_{LR} と D_{UD} の絶対値同士を比較し, 値の大きい変数に由来する候補値を `general_direction` の値とする.
 例えば, 図 A.3 中の B を分岐点とし, 下方向の選択確率を予測する場合, $D_{LR} = -1 - 1 = -2$, $D_{UD} = 1 + 1 + 1 + 1 = 4$ となり, $|-2| < |4|$ となるため, `general_direction` は下方向となる. また, 右方向の選択確率を予測する場合, $D_{LR} = 1 + 1 + \frac{1}{2}(1 + 1) = 3$, $D_{UD} = \frac{1}{2}(1 + 1 - 1 - 1) = 0$ となり, $|3| > |0|$ となるため, `general_direction` は右方向となる.

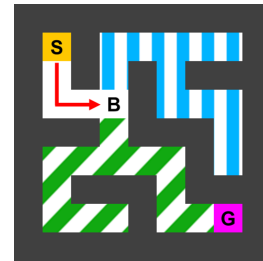


図 A.1 特徴量 `area` と `area_sum` の例

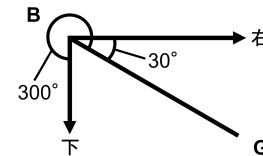


図 A.2 特徴量 `cos_goal` の例

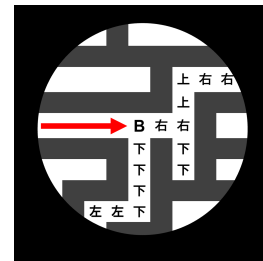


図 A.3 特徴量 `general_direction` の例

アルゴリズム 2 D_{LR} と D_{UD} の計算

```

1:  $D_{UD} \leftarrow 0$ 
2:  $D_{LR} \leftarrow 0$ 
3:  $S \leftarrow \{\}$  // 空のスタック
4:  $v \leftarrow$  ルートノード // 分岐点
5:  $r \leftarrow 1$ 
6:  $S.push((v, r))$ 
7: while  $S$  が空ではない do
8:    $(node, r) \leftarrow S.pop()$ 
9:   for all  $c \leftarrow node$  の子ノード do
10:     $d \leftarrow$  迷路上で  $node$  から  $c$  を見たときの方向
11:    if  $d =$  上 then
12:       $D_{UD} \leftarrow D_{UD} - 1 \times r$ 
13:    else if  $d =$  下 then
14:       $D_{UD} \leftarrow D_{UD} + 1 \times r$ 
15:    else if  $d =$  左 then
16:       $D_{LR} \leftarrow D_{LR} - 1 \times r$ 
17:    else if  $d =$  右 then
18:       $D_{LR} \leftarrow D_{LR} + 1 \times r$ 
19:    end if
20:    if  $c =$  十字路または丁字路の中心地点 then
21:       $S.push((c, r/2))$ 
22:    else
23:       $S.push((c, r))$ 
24:    end if
25:  end for
26: end while

```