

新しい迷路ゲームの開発とその面白さに関する研究

太田隆仁†¹ 小泉康一†² 大槻正伸†²

概要：本研究では新しい迷路ゲームを作成，そしてその迷路を解くための手順となる解を解析することでこのゲームを面白くするにはどうすべきか考察していく．このゲームの特徴としては，一般の迷路ゲームとは異なり，プレイヤー駒が1歩動くたびに周りの壁も動くことである．壁にさわらないようにプレイヤー駒を動かし，指定された歩数を歩くゲームと，ゴールへ移動するゲームの2種類を作成する．研究の方針としては，まずこれらのゲームをコンピュータ上で再現し，プレイヤー駒や壁の動作として可能な行動すべてを解析する．そして，解析したデータやテストプレイヤーの意見をもとに，最大の盤面の大きさ，または壁を何枚配置するといった難易度の設定やゲームの改良をする．盤面には，移動する壁とプレイヤー駒があり，駒を上下左右のいずれかに1歩進ませるとすべての壁が指定の方向に1歩動き，まわりの壁の状況が変化する．今回作成するゲームでは，2つの盤面作成法を考案した．まず1つ目は，初めにランダムに盤面を作成し，プレイヤー駒がクリアできるかどうかを解析し，1つ解が見つければその盤面を出力するというものである．2つ目は，あらかじめ壁の移動に関する完全データベースを作成し，それを参照することで必ず1回で目的の仕様に沿った盤面を作成するようにする．今回作成したゲームでは，プレイヤー駒に各ターンどの方向に移動するのがよいか考えてもらうために，適切な難易度設定の1つの方法として各ターンの移動する選択肢の平均を2もしくは3に近づけるのがよいと考えられる．

キーワード：ボードゲーム，コンピュータアルゴリズム，迷路

1. はじめに

近年，ボードゲームは人々に楽しまれており，自分でも作成したいと考えた．そこで本研究では新しい迷路ゲームを作成し，そしてその解を解析することでこのゲームを面白くするにはどうすべきか考察していく．

研究の方針としては，まずこのゲームをコンピュータ上で再現し，可能な行動すべてを解析する．そして，解析したデータやテストプレイヤーの意見をもとに，難易度の設定やゲームの改良をすることを目的とする．

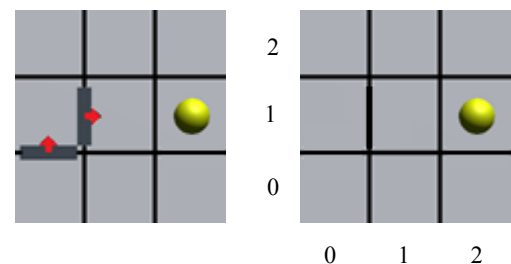
今回ゲームを作成するにあたって，Unity という開発環境を利用している^[1]．Unity は，2005 年に Mac OS X に対応したゲーム開発ツールとして誕生したゲームエンジンである．

2. ゲームの概要

今回作成したゲームは，図 1 (a) のように $N \times N$ の盤面に M 個の壁とプレイヤー駒が 1 個配置される．

プレイヤー駒と水平方向，垂直方向の壁は例えば図 1 (a) のように配置されている．プレイヤー駒，水平方向の壁，垂直方向の壁の位置は，それぞれ図 1 (b), (c), (d) のような座標で表現する．図 1 (b), (c), (d) より，プレイヤー駒，水平方向の壁，垂直方向の壁はそれぞれ $(x, y) = (2, 1)$ ， $(x, y) = (0, 1)$ ， $(x, y) = (1, 1)$ の座標にいることになる．全ての壁は次に移動する方向を示しているため，水平方向の壁と垂直方向の壁は図 1 (c), (d) よりそれぞれ上と右に移動することになる．本研究において，プレイヤー駒，水平方向の

壁，垂直方向の壁をそれぞれ $P(x, y)$ ， $H(x, y)$ (次の移動方向)， $V(x, y)$ (次の移動方向)と表すことにする．図 1 (b) から(d) をそれぞれ表してみると $P(2, 1)$ ， $H(0, 1, 上)$ ， $V(1, 1, 右)$ となる．

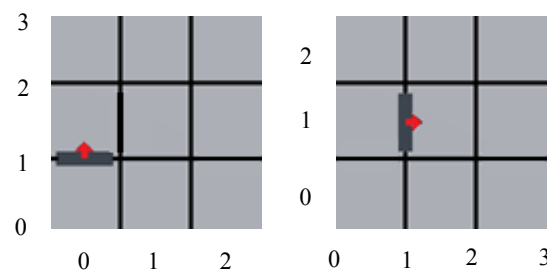


(a) 元の盤面

(b) プレイヤー駒の座標

(a) Original board face.

(b) Player piece coordinates.



(c) 水平方向の壁の座標

(d) 垂直方向の壁の座標

(c) Horizontal wall coordinates. (d) Vertical wall coordinates.

図 1 3×3 盤面，壁 2 個のときの座標例

Figure 1 An example of the 3×3 board surface, 2 walls coordinates.

†1 福島工業高等専門学校 電気工学科
Electrical Engineering,
National Institute of Technology, Fukushima College
†2 福島工業高等専門学校 電気電子システム工学科
Electrical and Electronic System Engineering,
National Institute of Technology, Fukushima College

このゲームの特徴としては、一般の迷路ゲームとは異なりプレイヤー駒を上下左右のいずれかに1歩進めると、1歩進んだ後全ての壁があらかじめ指定された方向に1歩動き、全体の壁の状況が変化する。図2左においてP(2,1), H(0,1,上), V(1,1,右), H(2,1,上)の状態ではプレイヤー駒を上へ1歩進めると、図2右のようにプレイヤー駒がP(2,2)の位置に移動した後H(0,2,上), V(2,1,右), H(2,2,上)となるように全ての壁が移動している。

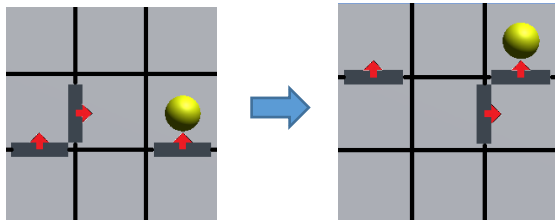


図2 3×3 盤面, 壁3個のゲームの例
Figure 2 An example of the 3×3 board surface, 3 walls game.

今回のゲームにおける壁の移動方向は6種類あり、上、下、左、右、cw(時計回り)、ccw(反時計回り)であり、cwとccwはどちらも90度ずつ回転する。また、壁が移動した際に盤面の端にいた場合は壁の移動方向を反射させる、つまり移動してきた方向と反対の移動方向にする。その様子を図3に示す。図3左において、P(0,1), H(1,1,下)の状態ではプレイヤーを下へ1歩移動させると、図3右のようにプレイヤーがP(0,0)の位置に移動した後H(1,0,下)に移動する。すると、移動後に盤面の端にいるため移動方向を下から上、反対方向に変更させH(1,0,上)となる。

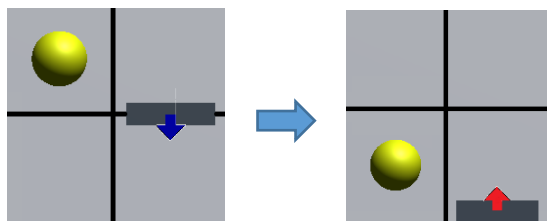


図3 壁の反射の例
Figure 3 Example of wall reflection.

今回は2種類の1人用ゲームと1種類の対戦ゲームを作成する。1人用ゲームの1つ目は、壁に触らないようにプレイヤーを動かす、指定された歩数を歩くゲームで、2つ目は盤面に用意されたゴールの座標へとプレイヤー駒を移動させるゲームである。対戦ゲームは、1人用のゲームの1つ目と同じルールで、プレイヤー駒側と壁側の2人に分かれて操作をする。ただし使用する壁の移動方向は上下左右のみとしており、壁側の人間は上下左右の方向を1つ選び次にその方向に移動するような壁全てを移動させる。本研

究では、それぞれのゲームをゲームA、ゲームB、ゲームCと呼ぶことにする。ゲームAにおいて、ゲーム開始時に指定された歩数をプレイヤー駒が負けることなく、すなわち壁に接触されたり壁に囲まれて動けなくなることなく、歩くことができればプレイヤー駒の勝ちとなる。ゲームBはゲーム開始時に用意されたゴールの座標にプレイヤー駒が負けることなく移動できればプレイヤー駒の勝ちとなる。ゲームCはゲーム開始時に指定された歩数を負けることなく歩ければプレイヤー駒の勝ち、プレイヤー駒が壁に触れたら壁側の勝ちとなる。ゲームA、Bの両方において、プレイヤー駒が壁に接触してしまうか、指定時間経過してもクリアできなければプレイヤー駒の負けとなる。ゲームCにおいてプレイヤー駒側、壁側にそれぞれ指定時間が設定されており、指定時間経過した場合は、経過させたプレイヤーの対戦相手側の勝ちとなる。

ゲームの流れを説明するためにまずゲームAの一例をそれぞれ図4(a)から(f)に示す。

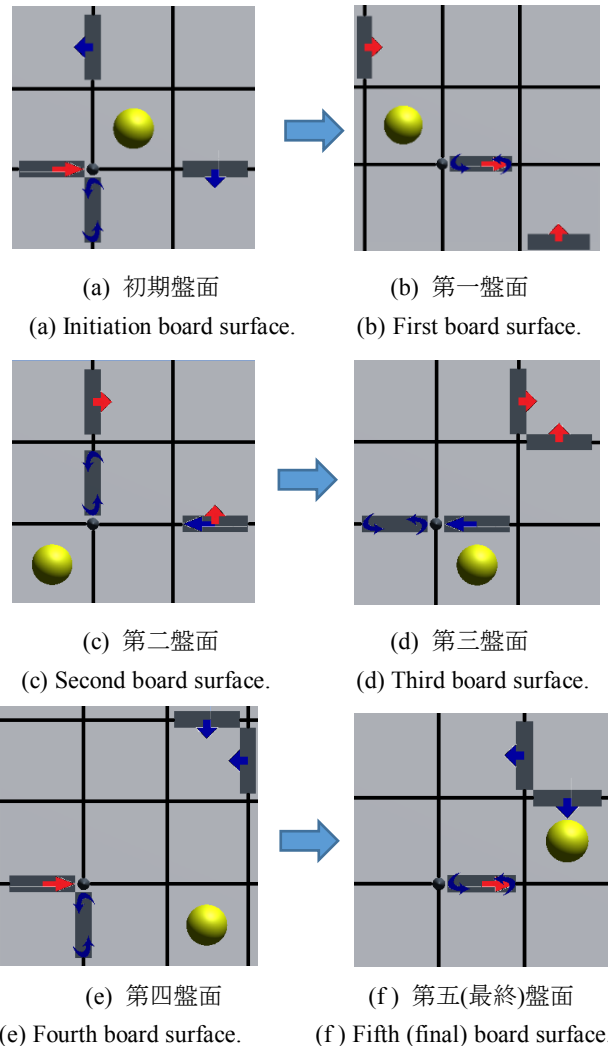


図4 指定歩数5のゲームAの例
Figure 4 Example of game A with 5 steps.

ゲーム開始時の盤面は図 4 (a) になっている。図 4 (a) の盤面において、P (1, 1)を下に移動させると V (1, 0, ccw)に触れてしまうため、下以外の方向として 1 歩目は左に移動させる。そうするとプレイヤー駒の壁がそれぞれの方向を動き図 4 (b) となる。以下同様にしてゲームを進める。このように、壁に触れないように指定された歩数歩くゲームがゲーム A である。

次に、ゲーム B の一例を図 5 (a) から(e) に示す。

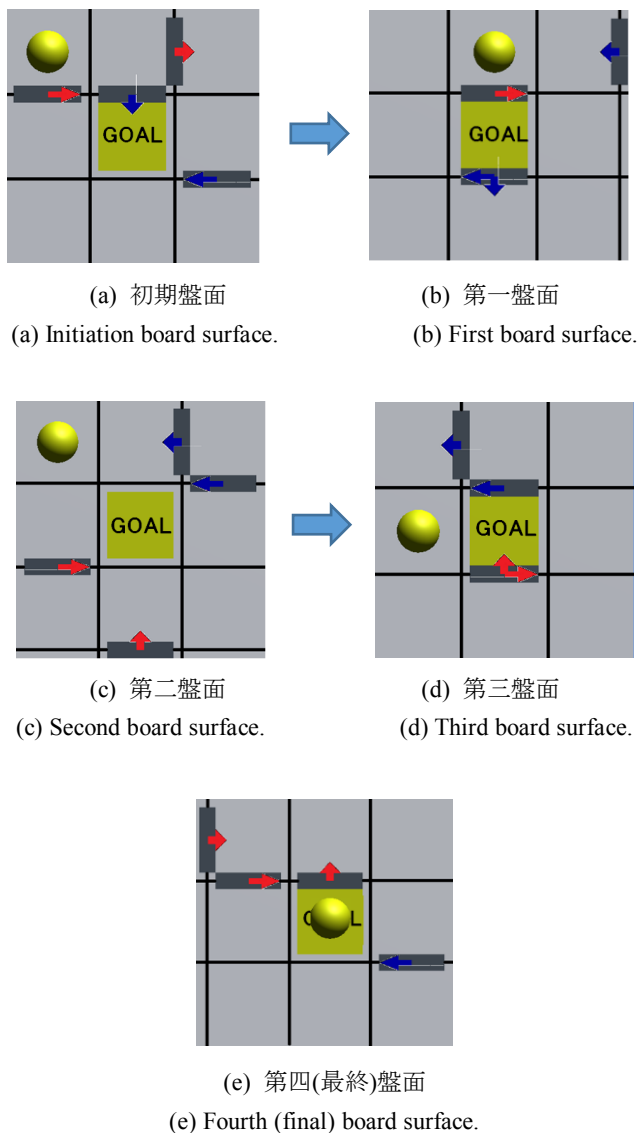


図 5 ゲーム B の例

Figure 5 Example of the game B.

ゲーム開始時の盤面は図 5 (a) になっている。ゲーム B も、ゲーム A と同様に壁に触れないように移動させ、ゴールを目指すゲームである。ゴールの座標は、図 1 (b) のプレイヤー駒座標と同様のルールに従って決定する。しかし、このゲームではプレイヤー駒がゴールした後ならば壁に触れてもよいことにしている。そのため、図 5 (d) の盤面において、P (0, 1) を右方向に移動させることで図 5 (e) のよ

うにプレイヤー駒がゴールの座標(1,1)に移動できる。その後 H(1, 1, 上)がゴールマスの上を通過してもプレイヤー駒はゴールした後のため、プレイヤー駒は負けることはない。

次にゲーム C の一例を図 6 (a) から(g) に示す。

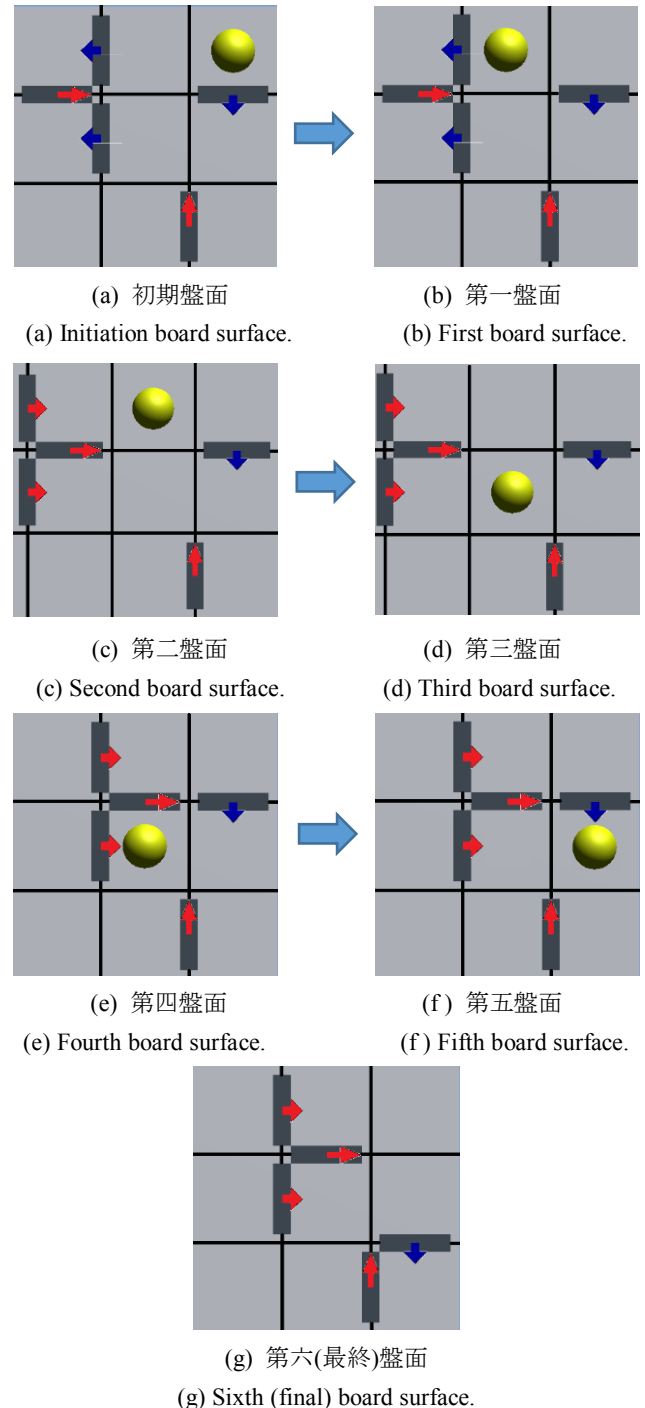


図 6 指定歩数 5 のゲーム C の例

Figure 6 Example of the game C.

ゲーム開始時の盤面は図 6 (a) になっている。ゲーム C はゲーム A と同じ勝利条件となっているが、前述のとおり壁側の移動方法が異なり、壁側は上下左右のいずれかの方

向を選択し、次に選択された方向に移動するような壁のみを全て移動させるようにしている。図 6 において、図 6 (b), (d), (f) はプレイヤー駒の移動後の盤面を、図 6 (c), (e), (g) は壁側の移動後の盤面を表している。このゲームでは、壁側はどの壁の移動でプレイヤー駒を追い詰めるかが重要になってくる。図 6 (b) において、壁側は左方向を選択することで $V(1, 1, 左)$, $V(1, 2, 左)$ を、それぞれ $V(0, 1, 右)$, $V(0, 2, 右)$ に移動させ図 6 (c) の盤面にしている。図 6 (c) において、 $P(1, 2)$ を下以外の方向に移動させると負けが確定してしまうため、下に移動せざるを得なくなる。このように壁側はプレイヤー駒を誘導し、どのように詰ませるかが重要になる。この例では、図 6 (f) において、 $P(2, 1)$ が間違っただけで右に移動したことで $H(2, 2, 下)$ を図 6 (g) のように下方向を選択し、 $H(2, 1, 下)$ にすることでプレイヤー駒に触れることができたためゲームが終了し、プレイヤー駒が負けたため盤面から消滅している。図 6 より、ゲーム C ではプレイヤー駒及び壁側は各ターンで 1 方向しか移動していないことが確認できる。

今回作成したゲームのプログラム上での各ターンの流れとしては、以下ようになる。

- ① プレイヤー駒の勝敗を確認する。もし勝敗が決していればゲームを終了し、勝敗結果を表示する。
- ② 上下左右のいずれかの方向を入力し、プレイヤー駒を入力された方向に移動させる。
- ③ 全ての壁を指定された方向に移動させる。その後、必要ならば全ての壁において移動方向を変えるかどうか確認する。

3. アルゴリズムの概要

新しい迷路ゲームを作成するにあたって、最初に考案した盤面作成法としては、まず $N \times N$ 盤面を決め、壁の総数 M を何個にするか設定し、使う壁とそれらの移動する方向をランダムに設定し配置する。その後、その盤面にはプレイヤー駒がクリアできる手順があるかどうか解析し、手順が見つからなければ N と M の値をそのままにして新たにランダムに盤面作成、解析をプレイヤー駒がクリアできる手順が 1 つ以上になるまで繰り返し行う。解析アルゴリズムは、一般の迷路ゲーム探索の手法^[2]を参考にして本ゲームに対応させて構築している。

今回作成したゲームの解析プログラムの流れとしては、再帰呼び出しを用いて以下ようになる。

- ① 現在の壁とプレイヤー駒の状態の情報を全て作業用配列にコピーする。
- ② コピーした配列情報を元に、関数 Search を呼び出す。
- ③ 関数 Search の結果を表示する。

関数 Search

- ① プレイヤー駒の勝敗を確認する。もし勝敗が決していれば勝利、または敗北のカウンタを増加させ、処理を終了する。
- ② プレイヤー駒を上下左右それぞれに移動させる。
- ③ 全ての壁を指定された方向に移動させる。その後、全ての壁において移動方向を変えるかどうか確認する。
- ④ 現在の壁とプレイヤー駒の情報を元に、関数 Search の再帰呼び出しを行う。

この方法でゲームを作成した際は、プレイヤー駒の勝利数 W に対して難易度 EASY, NORMAL, HARD の 3 種類を用意し、 W が 40 以上のとき EASY, W が 11 以上 40 未満のとき NORMAL, W が 10 以下のとき HARD と区分し壁とプレイヤー駒のデータをテキストにして保存する。実際に人間にプレイしてもらったときは 3 つの難易度から 1 つを選択してもらい保存した盤面データからランダムに 1 つを選択し、盤面を再現する。

上で示した盤面作成法では、うまくプレイヤー駒のクリア手順が見つからないときに何度もランダム生成、解析を繰り返してしまい、1 つの盤面を作成するのに時間がかかることもある。そのため、適切な N , M の値を入力すれば必ず一定時間で作成できる新しい効率的な盤面作成法を考案した。作成アルゴリズムは、一般の迷路ゲームの手法^[3]を参考にして本ゲームに対応させて構築している。

まずゲーム A を新しい盤面作成法を用いて作成することにした。盤面を 6×6 、歩数を 10 歩とし、 6×6 盤面での壁の最大配置数 60 個における 10 歩分の壁の移動データを全てデータベースとしてテキストに保存した。作成したデータベースを用いた盤面作成法は以下ようになる。

- ① ゲームで使用する壁の個数を入力し設定する。
- ② プレイヤー駒の初期座標から、10 歩分の移動方向をランダムに指定する。
- ③ 全ての壁の各移動において、壁が移動することでプレイヤー駒に接触するかどうか確認し、10 歩中 1 回もプレイヤー駒に接触しない壁のみを記憶しておく。
- ④ 記憶させた壁の中から、①で設定した壁の個数分だけランダムに壁を選出する。

新しい盤面作成法では必ず 1 回の定数時間で盤面が作成できるため、ゲームを作る際に要する時間が少なくなるというメリットがある。

次にゲーム B に対する定数時間作成法を作成する。盤面を 6×6 とし、このゲームの場合は、プレイヤー駒がゴールへ移動するまで何歩でも歩いてよい場合、全ての壁 60 個における壁の移動の一周分分のデータベースを作成しテキストに保存した。作成したデータベースを用いた盤面作成法は以下ようになる。

- ① ゲームで使用する壁の個数を入力し設定する.
- ② プレイヤー駒の初期座標, ゴールの座標をランダムに設定し, プレイヤー駒がゴールに移動するまでの移動方向を保存しておく.
- ③ 全ての壁の各移動において, 壁が移動することでプレイヤー駒に接触するかどうか確認し, プレイヤー駒が歩いた歩数の内 1 回もプレイヤー駒に接触しない壁のみを記憶しておく.
- ④ 記憶させた壁の中から, ①で設定した壁の個数分だけランダムに壁を選出する.

4. 成果

今回の研究では, ゲーム A, B において, ゲームの作成, 解析, 新しい盤面作成法を完成させた. ゲーム C においてはゲームの作成, 解析を完成させた. 解析結果の例として, ゲーム A における 6×6 盤面, 壁 30 個, 10 歩のゲームの解析結果を図 7 に示す.

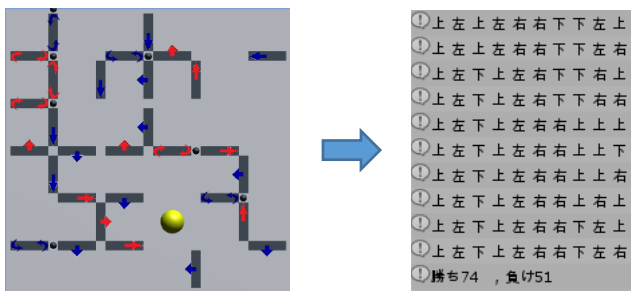


図 7 ゲーム A の解析結果の例
Figure 7 Example of analysis result of game A.

図 7 左の盤面で解析プログラムを実行すると, 図 7 右のように結果が表示される. 解析結果は, 左端から右端の順にプレイヤー駒の勝つことのできる移動方向が示されている. そのため, プレイヤー駒が各ターンの結果に従うことで必ず勝つことができる. また, 解析プログラムは, 現在のゲームの進行状況から好きなタイミングで実行することができ, プレイヤー駒の何歩歩いたかによって残りの歩くべき歩数における勝ちルートを表示するようにしている. 解析した結果, プレイヤー駒が勝てるルートが無い場合は, 負け, と表示するようにしている.

福島高専のオープンキャンパスで 20 人以上の来場者にテストプレイをしていただいた. その結果を表 1 に示す.

テストプレイでは, 難易度 EASY, NORMAL, HARD のうち 1 つを選び, あらかじめ保存しておいた盤面データの中からその難易度に応じた盤面データをランダムに 1 つ選出し, ゲームの盤面を再現している. テストプレイの際, まずはゲーム A の EASY を 1, 2 回プレイしてもらい, 余

裕があればゲーム B の EASY, ゲーム A の HARD 等をプレイしてもらった. また, テストプレイをしてもらう際, 勝つことのできるルートの数で設定していた難易度の他に, 追加で難易度 EASY, NORMAL では「移動すると壁に触れる」方向には移動できないように設定し, EASY のみそれを可視化するように設定している.

結果からは EASY はクリアしやすく 50%近い勝率となり, HARD は少なくとも 1 手先の盤面を考えなければクリアできない難しさとなる事が分かった. また, ゲーム中のテストプレイヤー駒を見ていると, 上下左右の選択肢が 1 つだけだと考える余地が無く簡単で, 選択肢が 4 つある場合は周囲に壁がない極端な配置であることが多いのでやはり簡単すぎるようであった.

表 1 各ゲーム, 各難易度におけるテストプレイヤーの勝率

Table 1 Each game, the winning percentage of the test player at each degree of difficulty.

	試合数 [回]	勝ち数 [回]	勝率 [%]
EASY	80	38	47.5
NORMAL	4	2	50
HARD	59	9	15.25
生存	81	41	50.62
脱出	43	6	13.95
対戦	19	2	10.53

単純な盤面作成プログラムと効率的な盤面作成プログラムについて, どちらもゲーム A, 6×6 盤面, 壁 40 個, 10 歩の条件で 100 回作成した際の実行時間を計測した. 結果を表 2 に示す. 結果からわかったこととしては, 単純な盤面作成プログラムより効率的な盤面作成プログラムの方が約 55 分の 1 の時間で作成できることがわかった.

表 2 盤面作成の実行時間の比較

Table 2 Comparison of execution time of board making.

作成法	時間 [s]
単純	834.5
効率的	14.94

また, ゲーム A, ゲーム B それぞれにおいて, 単純な盤面作成法と効率的な盤面作成法でのゲームを電気工学科の学生に各 10 回テストプレイしていただいた. 単純な盤面作成法でプレイしていただいた際はオープンキャンパスのときと同じ HARD, あらかじめ保存しておいた難易度 HARD の盤面データの中からランダムに 1 つ選出した. 効率的な盤面作成法でプレイしていただいた際はゲームを開始する

際に毎回3節に示した通りの方法で盤面を作成した。テストプレイからわかったこととしては、ゲームA、ゲームBともに単純な盤面作成法であらかじめ盤面を作成しておいた方が、難易度に応じた盤面を安定して生成できていた。効率的な盤面作成法でその場で盤面を作成する方は作成される盤面が、1歩でゴールに移動できるゲームB、プレイヤー駒の移動方向が4つ全てである等、極端になることがあり難易度に応じて安定した盤面を作成はできておらず、調整不足であった。

5. 考察

オープンキャンパスでのテストプレイではプレイヤー駒の勝利できるルートの数 W の数によって区分され、それに追加して操作性の違いを持って難易度とした。電気工学科の学生にしてもらったテストプレイではゲームの難しさを求めるなら単純な盤面作成法、ゲーム作成時間の短さを求めるなら効率的な盤面作成法を用いるのがよいことがわかった。そこで、今回作成したゲームの難易度を人間の視点から考えてみると、ゲームのプレイヤーは次の盤面を考えながら駒をどの方向に移動させるか決定させる必要があるため、駒の移動する選択肢は多すぎず少なすぎない2, 3個のみになるような盤面にすべきだと考えられる。具体的な方法としては、効率的な盤面作成法で盤面を作成した後、盤面の解析を行い、各ターンにおける、プレイヤー駒の移動しても壁に触れない方向を記録し、解析終了後にプレイヤー駒の歩数に対する移動方向の平均をとる。そして、移動方向の平均が2もしくは3に近いものをHARD, 1もしくは4に近いものをEASY, NORMALとすればよいのではないかと考えられる。

今回作成したゲームはプレイヤー駒を操作することで他のキャラクターを自動的に操作している。このような考え方でターンを進行させるゲームは種類として多く存在するため、今回の研究でこのようなゲームを作成することで他のゲームを作成する際の手助けとなると考えられる。

6. まとめ

新しいゲームの作成と、その解析は実施できた。そして、保存領域をほとんど使用しないであらゆる盤面を自動的に生成できるアルゴリズムの開発もできたが、難易度区分の点においてはまだ調整できていない。これについてはテストプレイの結果から改善すべき点が発見できた。しかし、今回のゲームにはテストプレイの回数がまだ足りないと考えている。そのため、今後はより多くのテストプレイをしてもらい難易度調整のために細かな修正、追加などをしていきたいと考えている。また、新しい盤面作成法について、壁の一周期分のデータベースを用いればゲームBだけでな

くゲームAの歩数や壁の個数 M を自由に指定できるようになるため、まずこれを修正し、一般的なゲーム設定ができるようにしたいと考えている。

今回のゲーム作成では、ターン毎に異なるキャラクターを操作するプログラムの考え方を理解できた。この考え方を利用すれば様々なターン制ゲームの作成の手助けになると思われる。

参考文献

- [1] Unityとは?特徴から参考サイトまで徹底解説!
<https://blog.blog.codecamp.jp/unity-explanation>
2019年2月7日 確認。
- [2] 迷路は幅優先探索?
<https://qiita.com/nati-ueno/items/a789095aff0aacc10d5d0>
2019年2月7日 確認。
- [3] 迷路自動生成アルゴリズム
<https://qiita.com/wgureimonn/items/99f2542acc6d400de0a2>
2019年2月7日 確認。