

複数A. I. のプレイ結果に基づくFPSゲームのステージ自動評価手法 Automatic Game Stage Evaluation Method for the First Person Shooting Game with Multiple Player A.I. Programs

沼田 哲史†
Satoshi Numata

ヌニェス ロベルト†
Roberto Núñez

1. はじめに

シューティング・ゲームには様々な種類のものがあるが、目標は基本的に同じである。プレイヤーが敵を倒し、ステージや迷路の出口まで行くということである。文献[1]によると、ゲーム人工知能(A.I.)はゲームの成功においてますます重要な要素になっている。文献[2]によればゲーム会社の成功か失敗を決定する場合すらあり、A.I.の開発の目標はプレイヤーが勝つことができないような敵を作ることよりも、人間と近い行動をする敵を作ることである。そうすることにより、プレイヤーはゲームに対してより没入する事ができる。

ゲームの面白さには様々な要素があるが、シューティング・ゲームにおいては、ストーリーやグラフィックス以外でゲームが与えるチャレンジと関係のある要素がある。この要素はゲームに大きな影響を与える。例えば、敵のA.I.と敵とプレイヤーの属性である。属性に関してはヒットポイント(HP)や武器の力などがある。この要素の組み合わせの変化は、プレイヤーがよりゲームに興味を持つ、あるいは失うことにつながる。ゲームにおけるこの要素の影響を見分けることができれば、面白さを計ることが可能になる。

2. 研究内容

ゲームの面白さを分析する様々な研究がある。例えば心理学者のチクセントミハイによってフローという概念が提唱されており[3]、その概念に基づいてゲームの面白さを数値化する研究も行われている[4]。しかしその数値化は人間の主観に基づくものである。また、ゲームのプレイ結果の楽しさを評価する論文がある[5]。これはゲームの開発中に難易度を分析し、調整するという研究である。A.I.で制御するプレイヤーをRPGゲームの中で敵と100回戦わせ、その結果によって最も適当な敵のA.I.設定を選択するというものである。この研究は数学的な分析を行うが、本研究で提案する手法とは異なり、面白さを数値化することが目標ではなく、適切な難易度の設定のみを目標としている。

本研究では、シューティング・ゲームの中において、プレイヤーA.I.によってゲームの難易度と面白さを評価する方法を提案する。つまり、人間のプレイヤーの代わりにA.I.によってゲームをプレイするプレイヤーを使用し、そのプレイ結果を元にゲームの面白さを分析し評価する。複数の人間プレイヤーのレベルをシミュレートするために、複数のプレイヤーA.I.を用いる。また、複数の敵の行動と難易度を評価するために複数の敵のA.I.も用いる。なお、この評価をするためにシューティング・ゲームを作成した。このゲームは一人称視点シューティング・ゲーム(FPS)として作

成したが、上方からの視点でプレイすることも可能である。プレイヤーは異なる特徴を持つ4つのステージを通して、ロボットのキャラクターを制御する。ゲームの目標はロボットの敵を全て倒し、ステージの出口に到達することである。

敵とプレイヤーには、基本的に次の3つの属性が使われる。速度、HP、武器の力である。速度とはプレイヤーの動ける速さを指す。HPはプレイヤーが攻撃として与えるダメージを数値化したものである。武器の力とは、敵を倒すための武器の属性である。例えば、ショット率と与えるダメージである。これらの属性で強さを定め、これらの属性も評価時に考慮する。プレイヤーA.I.に関しては異なる能力を持つプレイヤーをシミュレートするために異なるアルゴリズムを使い、ゲームの流れの分析に使用する。この分析は適切な試行と行動の多様性を論じる文献[6]で使われている判断基準に基づいている。

3. ゲームのA. I.

この評価方法を実際にFPSで試すために、本研究ではプレイヤーと敵のA.I.を作成した。これらのA.I.の開発には有限オートマトンを使用する。有限オートマトンを用いた方法は、状態の追加によって複雑さを増し、ゲーム中での反応も上達させることができる。この有限オートマトンはSPAWN, IDLE, SEARCH, ATTACK, DIEの5つの状態をとる。敵のA.I.の有限オートマトンを図1に示す。

SPAWN状態においては、初期状態で有限オートマトンの初期値が設定される。IDLE状態は敵が方向なしで進む状態

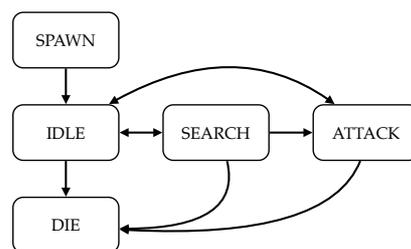


図1 敵の有限オートマトン

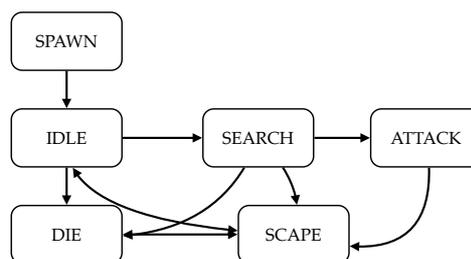


図2 プレイヤーの有限オートマトン

であり、道がある限り止まらずに進む状態である。SEARCH状態においては、プレーヤを探すため探索アルゴリズムが使われる。敵がプレーヤを発見するとATTACK状態に入り、進まずにプレーヤと戦う。この戦闘はプレーヤが敵の視界内にいる間に進行する。

プレーヤの場合は図2に示す有限オートマトンが使用される。プレーヤには、SCAPEステートが追加されている。プレーヤのHPが20パーセントになった場合、SCAPE状態に入る。この場合、プレーヤは戦いから逃げ、アイテムが現れたらそのアイテムがある方向に進む。敵とプレーヤのA.I.の複雑さを制御するため異なる状態の組み合わせを使用する。

敵とプレーヤのA.I.の複雑さを制御するため異なるステートの組み合わせが使用される。このような形で全ての状態を使用すれば最も複雑な行動ができる。この組み合わせをスクリプトと呼ぶ。敵の場合はWALL_TURN(WT), QUICK_SEARCH(QS), IDLE_SEARCH(IS), RANDOM_MOVE(RM)を用意し、プレーヤの場合はPLAYER_SCAPE(PS)というスクリプトを追加する。

QUICK_SEARCHスクリプト

それぞれのA.I.スクリプトは、異なる難易度と行動をシミュレートするために作成している。例えば難易度の高い敵を作るために、プレーヤの位置を知っているスクリプトを作成した。これがQUICK_SEARCH(QS)というスクリプトであり、一定の時間ごとにプレーヤを探し、その位置まで進み、プレーヤを見つければ停止してプレーヤを倒す、あるいはプレーヤが見えないところに行くまでプレーヤに向かって弾を撃つというスクリプトである。このスクリプトは SPAWN, SEARCH, DIE, 有限オートマトンのステートを用いる。

WALL_TURNスクリプト

一方で難易度の低い敵も作成した。この敵は WALL_TURN(WT)というスクリプトを使い、プレーヤを探さずに歩き、プレーヤと出会えば弾を撃つが、移動はやめない。このスクリプトが用いる有限オートマトンのステートは SPAWN, IDLE, DIEである。

IDLE_SEARCHスクリプト

中級の敵を作るために、前の2つのスクリプトを組み合わせたものを使用する。この新しいスクリプトは決定された時間によってQSスクリプトとWTスクリプトを交換する。このようにすることで敵の行動は分散する。このスクリプトは有限オートマトンのIDLEとSEARCHという状態を使うためIDLE_SEARCH(IS)と呼ぶこととする。このスクリプトは SPAWN, SEARCH, IDLE, DIE, 有限オートマトンのステートを用いる。

RANDOM_MOVEスクリプト

IDLE_SEARCH(IS)スクリプトにより分散した行動が可能であるが、さらに無作為に行動する敵を作るためにRANDOM_MOVE(RM)というスクリプトを用意する。このスクリプトは敵が現れたらISとQSとWTの間において1つを選び、敵を作成する。こうすることで、新しい敵の行動が予測しにくいものとなる。

PLAYER_SCAPEスクリプト

前の4つのスクリプトはプレーヤにも適用できるが、プレーヤにしか使えないものも用意した。これはPLAYER_SCAPE(PS)といい、ISスクリプトに基づいたものである。一定の時間によってQSスクリプトとWTスクリプトを交換するが、プレーヤのHPが20%以下になるとプレーヤは戦闘から逃げ、アイテムが現れたらその方向に進む。アイテムが現れていない場合には、敵のいない方向に進むというものである。

4. シミュレーション

人間のプレーヤと敵のA.I.が戦うことをシミュレートするため人間のプレーヤの代わりにA.I.におけるプレーヤを使用し、それぞれのステージにおいて、敵1体のA.I.につき50回のシミュレーションを行う。最初に敵の人数と属性を次のように設定する。

- ステージあたりの敵の数は6体である。
- 同時に現れる敵の数は2つである。
- 敵1体のHPは10点である。
- 敵の弾1つが与えるダメージは1点である。
- 敵の速さは0.8pixel/frameである。
- 発射速度は1秒につき弾3発である。

プレーヤには、敵の場合と同じように始めに任意の属性が決められる、その後プレーヤの3つのレベルの能力をシミュレートするため、この属性とA.I.のアルゴリズムを変更する。プレーヤの初期値は次のように設定する。

- HPは30点である。
- 弾1つが与えるダメージは1点である。
- プレーヤの速さは1 pixel/frameである。
- 発射速度は1秒につき弾4発である。

ゲームをプレイしている期間中にこれらの情報を取得し、ゲームの終了時にテキストファイルに書き出す。

人間の場合には、プレイ結果と比較するためにアンケートを実施する。上述した情報をプレーヤにアンケートにおける評価と比較する。このアンケートではプレーヤに対してゲームの難易度を10段階で数値化させる。1はプレーヤにはゲームが簡単すぎることを意味し、10はゲームは難しくすぎることを示す。

人間のプレーヤが出した結果を、類似したA.I.プレーヤに基づいて数値化する。例えば結果がA.I.のプレーヤの中級のデータと似ていれば、人間のプレーヤのレベルは中級であると言える。あるいは上級のデータと似ていれば人間のプレーヤのレベルは上級であると言える。しかし、人間のプレーヤが出した結果に明確な傾向がなければ、2つのレベルの間で補間して数値化する。

その次に、前に計算した人間のプレイ・レベルと使用した敵のA.I.を用いて理論上の面白さを算出することによって、面白さを数値化する。この評価をアンケートの結果と比較する。

5. 評価

文献 [6] では3つの判断基準が使われている。この判断基準は、適切な難易度、行動の分散と空間の多様性である。本研究では異なる構成のステージが用いており、また敵が1つのステージの全てを巡るようには設計していない

め、空間の多様性の判断基準は用いないことにした。そこで他の2つの判断基準に関して次に述べる。

適当な難易度

プレーヤが早く倒される、あるいは倒されるのに多くの時間がかかるならばゲームはつまらなくなる。そのため、ゲームの難易度がプレーヤの能力にどれほど近づいているかを評価するために適当な難易度という判断基準を使う。

この判断基準を評価するためゲームの難易度と関係ある要素を分析する。例えば、プレーヤ自身のダメージやプレーヤが倒された回数など。それぞれの現在のゲームは異なるルールを持ち、異なるプレイの方法があるのでゲームの難易度を評価するため異なる判断基準が使われる。しかし、本研究のゲームにおいて使われるのは、プレーヤ自身のダメージ（ゲーム全体）、敵の正確度、ステージ終わるまでの時間、敵が現れるから倒されるまでの時間である。

この点を分析するために次の式を使う。

$$C = \left(1 - \frac{ave}{max}\right)$$

aveはプレイ結果のデータの平均である。
maxはプレイ結果のデータの中の最大値である。

この方程式において0と1の間の数を得る。低い数値が得られた場合は偏差が少ないということを示している。つまり、プレーヤが常に勝っている、もしくは常に負けているということを表示する。

行動の分散

敵対者の行動が多様になればなる程ゲームが面白くなる。ゲームの場合は敵の生きている時間とプレーヤがステージを完全することを評価する。

この点を分析するためシミュレーションで得たプレイ結果のデータを次の式に合てはめる。

$$D = \left(\frac{\sigma}{\sigma_{max}}\right)$$

この式においてσは標準偏差であり次の式で計算する。

$$\sigma = \sqrt{\frac{\sum(ave-dat)^2}{n}}$$

averageはプレイ結果のデータの平均である。
dataはプレイ結果のデータの各値を表示する。
nはプレイ結果のデータの値の数である。

σ_{max}は次の式で計算される。

$$\sigma_{max} = \frac{1}{2} \sqrt{\frac{n}{n-1}} (max - min)$$

nはプレイ結果のデータの値の数である。

maxはプレイ結果のデータの最大値である。

minはプレイ結果のデータの最小値である。

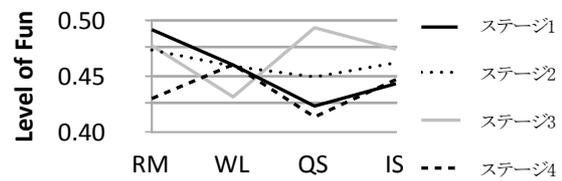
この式においても0と1の間の数値が得られる。この値は多様性の程度を示すものである。標準偏差を用いる理由は、平均と最大値の差違を計算するのではなく、価値の多様性を計算するためである。

このシミュレーションの結果に判断基準を当てはめた値を、次の図3から図6に示す。

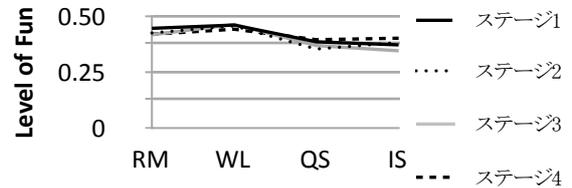
シミュレーションの結果から、各スクリプトの平均を計算すると表1に示す値が得られる。

2番目の実験は、ゲームを人間のプレーヤにプレイさせるというものである。このプレイ結果を記録し、プレーヤにゲームの面白さと難易度を調査するアンケートを行う。記録したデータとアンケートの答えをプレーヤのA.Iと比較して分析する。

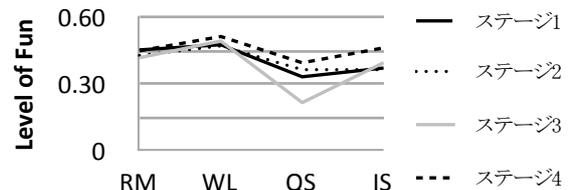
アンケートの質問項目は、敵の難易度とゲームの面白さの評価である。また、各ステージについての難易度と面白さについても問う。この質問は10項目あり、敵の各スクリプトについて回答させる。この実験は今回11人の被験者に



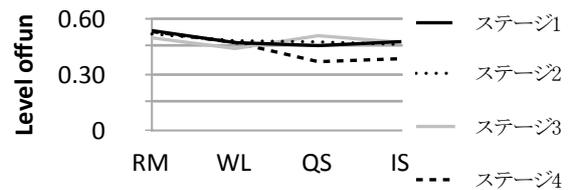
Enemy AI
図3 PSスクリプト結果



Enemy AI
図4 ISスクリプト結果



Enemy AI
図5 QSスクリプト結果



Enemy AI
図6 WTスクリプト結果

表1. シミュレーションから計算した面白さの平均

敵のスクリプト	面白さ
RM	0.4631
WL	0.4605
QS	0.3972
IS	0.4172

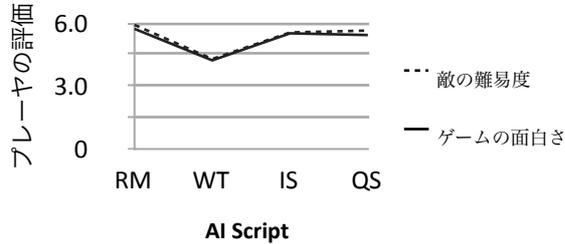


図7 アンケート結果

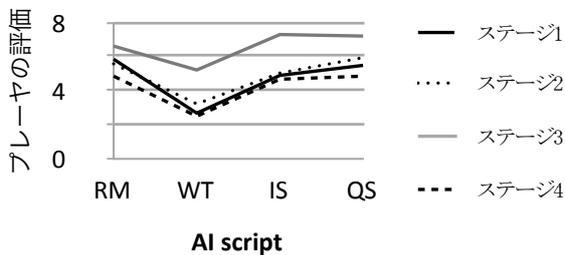


図8 各ステージの難易度のプレイヤーの評価

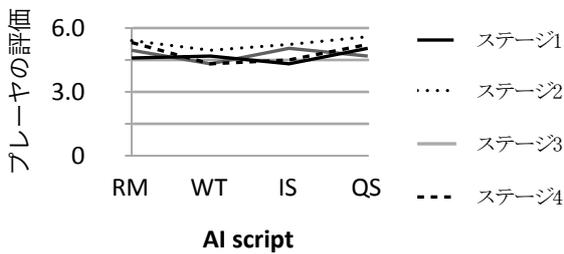


図9 各ステージの面白さのプレイヤーの評価

対して行った。ゲームの面白さと敵の難易度の質問に関する答えを図7に示す。

それぞれのステージの評価について図8、図9に示す。

図8と図9により、敵の難易度とゲームの面白さの関係が伺える。まず、ほとんど全員はRMスクリプトの方が面白いと答えており、敵の行動が無作為になるほどゲームが面白くなるのが分かる。またWTスクリプトは最も低い評価を得ており、ゲームをプレイした人間のプレイヤーのゲームプレイ能力が高いことが分かる。

6. 結論

本研究の実験で得られたシミュレーションとプレイヤーのプレイ結果により、ゲームにおいては敵の無作為な行動があることの重要性が分かる。このような実験で取得した情報でプレイヤーの評価に基づいてシステムを調整し、人間のプレイヤーの評価をA.I.のプレイヤーの評価に合わせる事が可能になる。一度調整を行うと、敵とプレイヤーのA.I.の実

験数が増え、データの傾向がより明確になるため、さらなる調整が行えるようになる。

ただし、今回の実験は1種類のゲームについて少数の人数の被験者に対してのみ行っているため、まだあらゆる種類のゲームに対して適用できる結果は得られていない。今後、より多様なゲームに対して、より多くの被験者に対する実験を行うことにより、正確な面白さの評価が自動的に行えるようになることを期待したい。

参考文献

- [1] Paul Tozour, The Evolution of Game AI, AI Game programming Wisdom, 2002, p.3.
- [2] Armand Prieditis and Mukesh Dalal, Applying Model-Based Decision-Making Methods to Games, Game Programming Gems 6, 2007, p.195.
- [3] Penelope Sweetser and Peta Wyth, GameFlow: A Model for Evaluating Player Enjoyment in Games, ACM Computers in Entertainment (CIE), Vol. 3, No. 3, July 2005.
- [4] Csikszentmihalyi Mihaly, Flow: The Psychology of Optimal Experience, Harper Perennial Modern Classics, 2008.
- [5] Pieter Spronck, Ida Sprinkhuizen-Kuyper and Eric Postma, Difficulty Scaling of Game AI, 5th International Conference on Intelligent Games and Simulation, 2004.
- [6] Georgios N. Yannakakis and John Hallam, Capturing Player Enjoyment in Computer Games, Advanced Intelligent Paradigms in Computer Games 2007, Springer, 2007.