

# 協調 AI への学習系の導入に対する一考察

安武諒平

指導教員 白濱成希

数年来アメリカでは、ゲーム開発が研究分野に値すると認知されてきているが、日本国内においてはこの分野に関する研究はほとんどなされていない。

本研究では、複数人チーム同士による対戦ゲームにおいてプレイヤーと協調して動く AI モデルに対して学習能力を導入し、人間のような知性を有する NPC を開発することを最終的な目的とする。そのために協調 AI モデルに対し学習系を実装し、実際にゲームをプレイした感想をアンケート調査した。アンケート結果は学習、協調ともにあまり強く感じることはできない、という意見が過半を占めた。

これを承けて今後の研究においてはより強く学習の効果を感じるような系の構築や協調性を高めるためにプレイヤーに対応付けられた行動の実装が必要であることが分かった。

## 1. はじめに

近年、デジタルゲームの分野ではハードウェアのメモリ容量や演算能力といった、リソースに関する技術の発達に特に顕著である。これにより、グラフィックの描画能力や画面の処理能力は著しい向上を見せ、従来のゲームよりも格段にリアリティの高い仮想空間が構築できるようになった。

このようにゲームの「外観」がより精緻に現実感を伴うものへと発展していくことで、必然的にゲームの「内面」、つまり仮想空間内の知性についてもこれまで以上のリアリティが求められる状況が発生した。

この仮想空間内の知性がそのゲームの有する視覚的な現実感よりも遥かに劣る現実感しか有さない場合、そのゲームのプレイヤーは大きな違和感と共に多大なストレスを感じてしまう。

これらの背景からゲーム開発の分野においてデジタルゲーム AI が注目を集めている。デジタルゲーム AI とは、ゲームという仮想空間内で、キャラクターが活動するために必要な知性のことである。従来、研究が行われてきた、例えばボードゲームの AI のような、固定されたルールの中で推論を行うプレイヤーの代理としての知性であるところのゲーム AI とは全く異なり、ゲーム世界の中にゲームデザインの要求に沿って配置されている、プレイヤーを楽しませるゲームの一部としての AI を指している。

そして、このデジタルゲーム AI を用いることで、プレイヤーはより楽しくより長期的にゲームをプレイ出来るようになる。

このデジタルゲーム AI が搭載された主なゲームの実例として「Halo」シリーズ(PC, Xbox, Xbox360)や、「F.E.A.R.」(プレイステーション3, Xbox360), 「クロムハウズ」(Xbox360)などが挙げられる。

しかしながら、日本国内ではデジタルゲーム AI という分野はあまり知られておらず、研究もほとんど行われていない。

本研究では、複数人で構成されたチームによる対戦ゲームにおいて、プレイヤーと協

調して行動する AI モデルに学習能力を付与し、より人間らしい知性を実装した NPC の開発を行う事を目的とする。

## 2. 開発環境

本研究の開発環境については以下に箇条書きで示す。

### • Windows PC

- OS
  - ✧ Microsoft Windows XP Professional Version 2002 Service Pack 3
- CPU
  - ✧ AMD Athlon(tm) 64 Processor 3500+ 2.2GHz
- メモリ
  - ✧ 1.5GB RAM
- グラフィックボード
  - ✧ ASUS EAH2400PRO
- ソフトウェア
  - ✧ Visual C# 2008 Express Edition
  - ✧ XNA Game Studio 3.1

### 2.1. XNA Game Studio

XNA Game Studio はゲーム開発者向けの統合開発環境である。XNA Game Studio は無料でダウンロードでき、プログラマがより簡単にゲームの開発をできるようにすることを目的としている。プログラマは XNA Framework を使用して Windows で実行できるゲームを無料で作成できるが、それを Xbox360 で実行するには有料会員制の「XNA Creators Club」に加入しなければならない。

また、XNA Framework は .NET Framework 及び .NET Compact Framework をベースにしたクラスライブラリの集合体であり、異なるプラットフォーム間でのコードの再利用を促進するためのゲーム開発に特化した拡張クラスライブラリが追加設計されている。

これには、XNA ゲームの実行に最適化されたマネージド実行環境を提供するための共通言語ランタイムが含まれている。ラ

ンタイムは Windows XP, Windows Vista 及び Xbox 360 に用意されており, XNA を利用して作成したゲームはこれらのプラットフォームにおいても最小限の改変もしくは全くの改変なしに動作する.

この XNA Framework を対象とするゲームは C# などマネージ言語を使ってコードを記述できる. XNA Framework には, ほかにもゲーム内容の充実のための援助となる XACT などのツールも含まれている. これらのツールはゲームの外観やサウンドを充実させるための支援となる.

また, 一般的な画像ファイルや 3D モデル, サウンド等を Windows, Xbox360 の両プラットフォームにおいて使用できる最適なデータ型への変換を行うシステムであるコンテンツ・パイプラインが存在し, 主にこの2つでゲーム開発の支援を行っている.

## 2.2. XNA Game Studio におけるゲーム開始から終了までの流れ

XNA Game Studio のゲーム起動からゲーム終了までの流れを示す.

### (1) Initialize メソッド

ゲームの開始時に呼び出され, ゲームで使用するデータの初期値を設定する.

### (2) LoadContent メソッド

ゲームで使用する画像や音楽データなどを呼び出す.

### (3) Update メソッド

キャラクターの移動や戦闘処理, キーボードやコントローラなど入力装置からの入力判定などを行う.

### (4) Draw メソッド

2D 画像や 3D モデルのレンダリングを行い, ゲーム画面の描画を行う.

### (5) UnloadContent メソッド

デバイスのリセット時に呼び出され, コンテンツの破棄を行う.

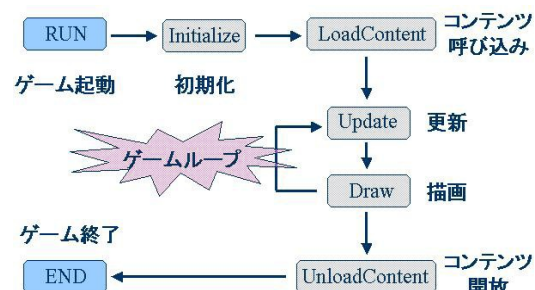


図1: ゲーム起動から終了までの流れ

## 3. ゲーム概要

本研究で学習系の導入を行った協調 AI

モデルを用いた対戦型ゲームの概要と実際のプレイ画面を示す.

- 1 チーム 3 人編成
- 2 チームによるチームバトル
- プレイヤ以外は全員 NPC
- 攻撃は射撃のみ
- 弾は自機の正面から放たれる
- 相手の攻撃を防ぐブロックはランダムに配置される
- 相手チームの全滅が勝利条件
- 勝利条件を満たすと画面は初期化される.

またプレイヤー, 及び NPC はそれぞれ以下の行動を取ることができる. 以下の行動は同時に取ることも出来る.

- 移動
- 旋回
- 射撃

射撃を再度行うには, 0.5 秒のディレイ時間が必要である.

また, 画面の左上と右上にはそれぞれのキャラクターのライフ残量が表示されており, このライフ表示が無くなったキャラクターはフィールド上から削除される.

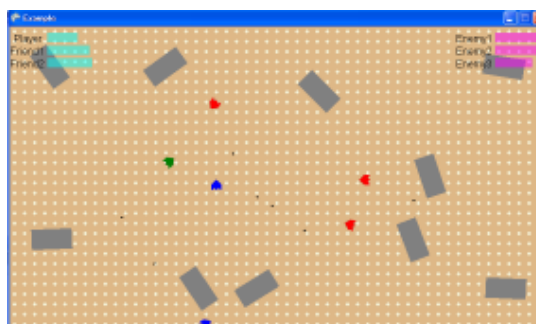


図2: 実際のプレイ画面

## 4. 協調 AI

本研究で用いた協調 AI モデルは, ウェイポイントを用いた移動を行い, 攻撃可能範囲内に相手陣営のキャラクターを感知した場合, その方向に回頭を行い射撃による攻撃をする.

### 4.1. ウェイポイント

ウェイポイントとは図1において, フィールド上に等間隔で配置されている白い点を指す. このウェイポイントは各々が知識表現としてデータを持ち, NPC はそれを基にゲーム空間を理解, 把握する. 本研究では, ウェイポイントに攻撃適正值, 防御適正值をそれぞれ持たせ, NPC はその情報から攻撃に有利な地点, あるいは防御に有利

な地点を判断することができる。

攻撃適正值(WP\_a), 防御適正值(WP\_d)は以下の式によって求めた。

$$\begin{aligned} WP\_a = & 0.5 \times (\text{間に障害物がない}) \\ & + 0.3 \times (\text{適正攻撃範囲内}) \\ & + 0.2 \times (\text{見晴らしが良い}) \end{aligned}$$

$$\begin{aligned} WP\_d = & 0.5 \times (\text{障害物の陰}) \\ & + 0.3 \times (\text{敵の射線軸上でない}) \\ & + 0.2 \times (\text{味方の近く}) \end{aligned}$$

適正值の計算における情報に対してはブール理論を用いて求め、例えば「間に障害物がない」ならば1, そうでなければ0で計算を行う。そして各NPCは自身の周囲のウェイポイントの適正值に対して自身の攻撃性を用いて重みを付け評価値を定め、最適な移動地点を推論し行動する。

#### 4.2. 評価関数

以下の評価関数で評価値の導出を行う。

$$\begin{aligned} WP = & A \times WP\_a \times WP\_a\_st \\ & + D \times WP\_d \times WP\_d\_st \end{aligned}$$

まず、それぞれのウェイポイントが持つ攻撃適正值に、NPC自身のライフとチームリーダーのライフを入力とするファジィ推論の出力である重み係数A, 学習系によって求めた攻撃志向度(WP\_a\_st)を掛け合わせる。

次に、それぞれのウェイポイントが持つ防御適正值に、ファジィ推論の出力である重み係数Aを100%から引いた値Dと、学習系によって求めた防御志向度(WP\_d\_st)を掛け合わせる。

そして、重み係数と志向度を掛け合わされた攻撃の適正值と防御の適正值を足し合わせたものがそれぞれのウェイポイントの評価値となる。

#### 5. ファジィ理論

ファジィ理論とは、人間が問題を解決するのと同じような方法でコンピュータに問題を提示する手段として考案された理論である。つまり、ファジィ理論を用いることでコンピュータがある情報に対し多め、あるいは少なめと言ったような程度の問題であると表現、理解させる事ができるということを示している。

具体的に言えば、コンピュータが情報を判断するための境界をグレーゾーンとし、人が判断する時のようなあいまいさを持たせることが可能となっている。このような

境界があいまいな集合をファジィ集合という。また従来のブール理論では、判断するための境界となるポイントを定義する必要がある、このような境界がはっきりとした集合をクリस्प集合という。クリस्प集合の例を図3に、ファジィ集合の例を図4に示す。

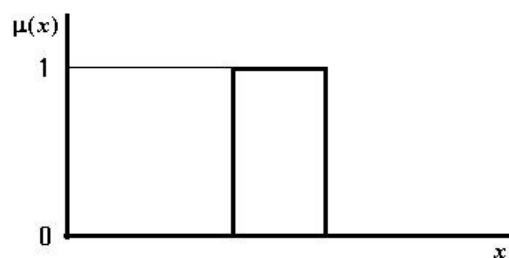


図3: クリस्प集合

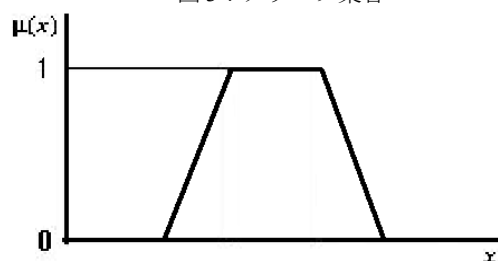


図4: ファジィ集合

ファジィ理論の推論プロセスは、以下の3つの基本的な手順で構成されている。

- (1) クリस्प入力 → ファジィ入力
- (2) ファジィルール → ファジィ出力
- (3) ファジィ出力 → クリस्प出力

(1)のプロセスはファジィ化と呼ばれる処理で、与えられたクリस्पデータをマッピングプロセスによってファジィデータに変換する。

このマッピングプロセスでは例えば図1のようなファジィ集合における、クリस्प入力のメンバーシップ度を割り出す。

システムへのすべての入力をファジィ集合のメンバーシップで表せるようになったら、ファジィルールに基づいてそれらを組み合わせ、各ルールの「真」の度合いを計算する(2)のプロセスを行う。これらのルールは論理演算によって、ファジィ入力変数を組み合わせ、該当する出力値を出力ファジィ集合のメンバーシップ度として割り出す。

最後にファジィ出力のメンバーシップをもとに、該当する数値のクリस्प出力を生成する(3)のプロセス、非ファジィ化を行う。

以上がファジィ推論の概略である。説明

に関しては[1]を参考にした。

### 5.1. ファジィ推論による重み係数の導出

前件部，つまりファジィ化されるクリस्प入力を NPC 自身のライフ  $sl$  とチームリーダーのライフ  $rl$  とし，後件部，非ファジィ化で生成されるクリस्प出力  $A$  を NPC の攻撃性とする．  $sl, rl$  は共に  $0 \sim 100$  までの値を取り，  $A$  は  $0 \sim 100\%$  までの値を取るものとする．また，  $A$  を求めるためのファジィ推論の IF-THEN 形式のルールを下記のように設定した。

```

IF(sl is SM)and(rl is SM)THEN(A is DE)
IF(sl is SM)and(rl is MD)THEN(A is SD)
IF(sl is SM)and(rl is BG)THEN(A is NO)
IF(sl is MD)and(rl is SM)THEN(A is SD)
IF(sl is MD)and(rl is MD)THEN(A is NO)
IF(sl is MD)and(rl is BG)THEN(A is SA)
IF(sl is BG)and(rl is SM)THEN(A is NO)
IF(sl is BG)and(rl is MD)THEN(A is SA)
IF(sl is BG)and(rl is BG)THEN(A is AT)

```

このファジィルールでは論理積演算子によって入力メンバーシップ度の中で最小の度合いをとり，それを用いて割り出した出力メンバーシップ度の加重平均によって出力クリस्प値の生成を行っている。

この協調 AI モデルにおいては NPC の個性を表現させるために，後件部  $A$  のファジィ集合に攻撃型と防御型の 2 つが用意されている．図 5 に前件部  $sl$  と  $rl$  のファジィ集合を，図 6 に攻撃型の後件部  $A$  のファジィ集合を，図 7 に防御型の後件部  $A$  のファジィ集合をそれぞれ示す．図 6，図 7 においてファジィ集合がそれぞれ左右に少しずつシフトしており，これにより出力メンバーシップ度が攻撃もしくは防御に偏りをみせる。

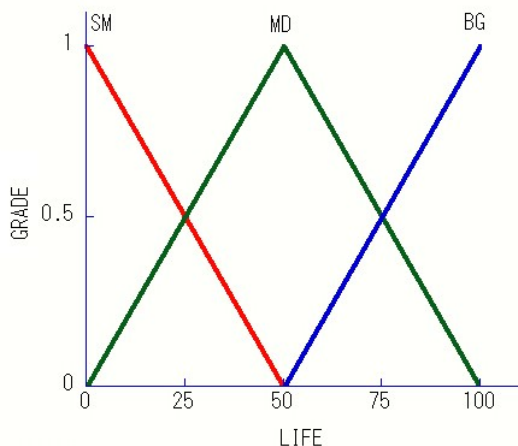


図 5 : NPC 自身のライフ  $sl$  とリーダーのライフ  $rl$  のファジィ集合

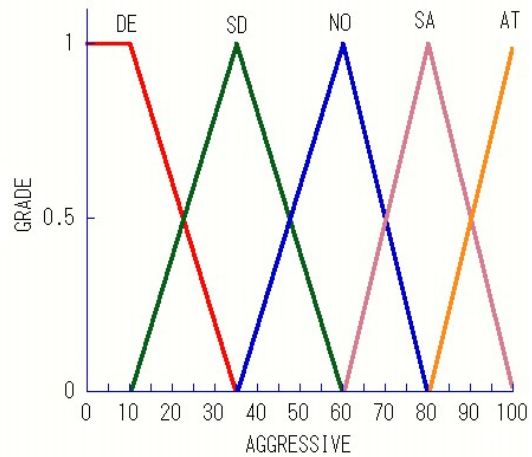


図 6 : 攻撃型 NPC の攻撃性  $A$  のファジィ集合

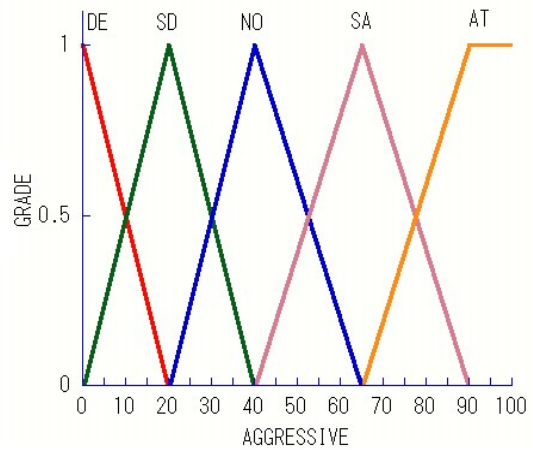


図 7 : 防御型 NPC の攻撃性  $A$  のファジィ集合

## 6. 学習系について

本研究で協調 AI モデルに対して導入を行った学習系は，勝敗が決した時点での味方及び，敵 NPC の生存数を記憶し，次ゲームの開始時に味方 NPC の攻撃志向度，防御志向度をそれぞれ以下のような計算式で増減させる事で NPC の行動を変化させるものである。

$$\begin{aligned}
 WP\_a\_st(r) &= WP\_a\_st(r-1) \\
 &\quad + (-0.2 \times friends(r-1) \\
 &\quad \quad + 0.15 \times enemies(r-1))
 \end{aligned}$$

$$\begin{aligned}
 WP\_d\_st(r) &= WP\_d\_st(r-1) \\
 &\quad + (0.2 \times friends(r-1) \\
 &\quad \quad - 0.15 \times enemies(r-1))
 \end{aligned}$$

ここで  $r$  は現在の戦闘，  $r-1$  は 1 つ前の

戦闘を表している。また friends は味方 NPC の生存数, enemies は敵 NPC の生存数である。

また, 敵 NPC の志向度は次式により導出する。

$$WP\_a\_st(r) = WP\_a\_st(r-1) + (-0.15 \times enemies(r-1) + 0.2 \times friends(r-1))$$

$$WP\_d\_st(r) = WP\_d\_st(r-1) + (0.15 \times enemies(r-1) - 0.2 \times friends(r-1))$$

この学習系では, もし味方が多く生き残っていたのならば, 敵よりも攻撃性において勝っていたと判断し, 防御志向度を上昇させ, 攻撃志向度を減少させる。また敵が多く生き残っていたならば, 攻撃性が足りなかったと判断し攻撃志向度を上昇させ, 防御志向度を減少させている。但し, この志向度による修正があまりに大きく影響しすぎないように, 蓄積は 0.1 から 1.5 までの間で行うものとした。

以上のようにして志向度の増減を蓄積させていくことで学習を行い, NPC の行動を調整する事を図っている。

また, この AI の記憶方法として, 必要なデータをシリアル化して外部に保存しておき, ゲームを再開する際に, そのデータをロードしてくる, という手法を用いた。

ゲーム内部にシリアル化可能属性を持つゲームデータ構造体を用意し, ゲームオーバーした際に志向度, 及び敵味方の生存数を構造体に格納し XML シリアル化して保存し, ゲームのリトライ時にオートロードを行った。

## 7. 実験

### 7.1. 実験方法

今回, 学習系の導入を行った協調 AI を組み込んだゲームを実際に遊んでもらう。

その後アンケート調査を行い, 協調性や学習についての評価を行う。アンケートで調査するのは以下の 2 項目とした。

- NPC に協調性を感じたか
- NPC が学習していると感じたか

上記の設問に対し「はい」、「まあまあ」、「あまり」、「いいえ」、「わからない」の 5 段階評価をしてもらい, また自由記述欄において何か意見があれば述べてもらう。

### 7.2. 実験結果

本校の学生 23 名に対しアンケートを行

った。アンケート調査結果を以下に示す。

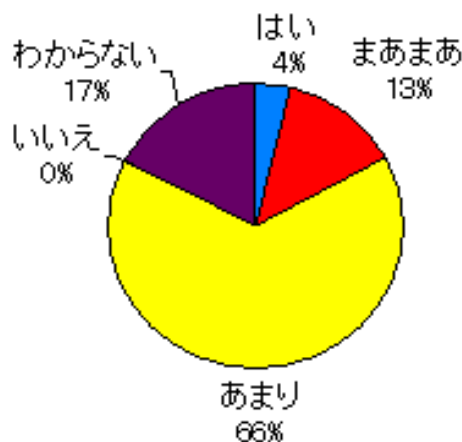


図 8 : アンケート調査結果 1  
NPC に協調性を感じたか

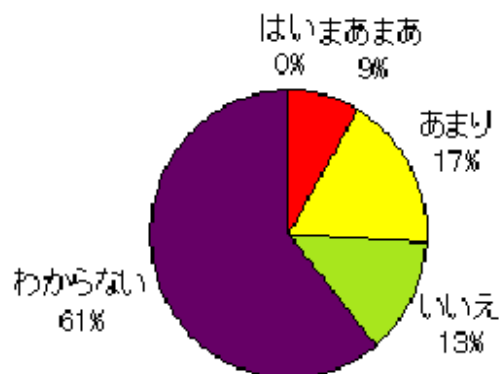


図 9 : アンケート調査結果 2  
NPC が学習していると感じるか

また, 自由意見を抜粋し, 箇条書きで以下に示す。

- 味方 NPC がプレイヤーの動きに対応してくれないように思う
- 敵 NPC 間には連携のとれた挙動が見られ, 協調性を感じた
- NPC に対し個性を感じる時もあるがそうでない場合もあった

## 8. 考察

アンケート調査結果 1 より, 味方 NPC に対して協調性を感じたのはおおよそ 2 割程度であり, 過半数があまり協調性を感じないと回答している。

全く協調性を感じない事はないようだが, これは自由記述でも指摘されているようにプレイヤーの意志が NPC の挙動に反映されない事が大きな原因となっているのではないかと考えられる。

また、これに関して敵 NPC 間に対しては協調性を感じたという意見も、敵はほぼ同一の移動アルゴリズムによって行動する為にきちんと統制のとれた動きを執っているように感じたのではないかと考える事ができるだろう。

学習に関しても 6割が学習の効果が「わからない」と回答している。本実験で導入を行った学習系は AI の行動指針に対して作用する、些か迂遠なきらいのあるものだった為か、実感に乏しくこのような結果になったように思う。

さらに自由意見の味方 NPC の挙動が同じように感じ、個性が失われているような状況があったという意見から、学習の内容如何では個性が均一化されてしまう恐れがあることに気がついた。今後の改良では、学習の際にも各 NPC の個性を活かせるシステムにすべきであると考えられる。

## 9. 今後の課題

前段の考察を承け、今後この AI モデルを改良するにあたり特に留意すべき課題について以下に述べる。

### 9.1. プレイヤの行動に対応付けられた味方 NPC の構築

考察においても述べた通りプレイヤに協調して動いていると感じてもらうためには、プレイヤの意志を理解しそのサポートを行っているような挙動を NPC がとることが重要であると考えられる。

具体的には、プレイヤが狙っている敵を優先的に追いかけるようなアルゴリズムを構築する。あるいはプレイヤが孤立した際に救援に来てくれるような挙動を実装することなどが考えられる。

このためウェイポイント間に繋がりを持たせ、移動テーブルを作成することが最重要の課題であると考えられる。移動テーブルが作成できたのならば、任意の点に対する最適経路探索などを用いることで、前述のアルゴリズムは構築できるものであると考えられる。また移動テーブルが実装されれば、障害物の回避が可能になり、移動のバリエーションも増やすことが可能になるなど多様な用途が見込まれ、AI がより知性的に振舞えるようになることが期待できる。

### 9.2. 学習系の再検討

考察で述べた個性を活かした学習系の構築は、例えば今回の学習系でいえば志向度を増減させるための係数を攻撃型か防御型かで異なるようにすることで実現できる。

しかしながら今回構築し、導入した学習系は効果が体感として分かりづらい事が問題であった。よって、よりプレイヤが学習の効果を感じ易いような学習系を構築することが必要である。

一定時間内に規定以上の攻撃を受けた場合に、一時的に防御を重視するようになるアルゴリズムを導入する。プレイヤの動きを学習し、その動きに応じて行動を大きく変えるアルゴリズムを導入する事などが考えられるが、何をもってして学習における最適とするかという議論を含め、さらに深く検討すべきである。

### 9.3. 人工知能分野の技術の導入

また、今回は導入できなかったが、評価関数の決定や学習系に関してニューラルネットワークを導入することでより人間に近い知性の実現が可能となる。

それ以外にも A\*アルゴリズムなど現在に到るまでに様々に研究の重ねられてきた、人工知能分野の技術を組み合わせる事で、より複雑で精緻な人間のように感じられる知性の構築が可能である。

## 謝辞

本研究を行うにあたり、また本稿の執筆においても様々なご指導、ご鞭撻を下さった白濱教員、並びに同研究室内の専攻科生の皆様、そしてアンケート調査に協力して頂いた電子制御工学科の方々に、最後に厚く御礼を申し上げます。

## 参考文献

- [1] David M. Bourg, Glenn Seemann, ゲーム開発者のための AI 入門, 株式会社オライリー・ジャパン, 2005 年
- [2] Mat Buckland, 事例で学ぶゲーム AI プログラミング, 株式会社オライリー・ジャパン, 2007 年
- [3] Stephen Cawood, Pat McGee, Microsoft XNA ガイドブック, 株式会社ボーンデジタル, 2008 年
- [4] Chad Carter, Microsoft XNA UNLEASHED, 株式会社ボーンデジタル, 2008 年
- [5] 有限会社オフィス加減, XNA Game Studio で始めるゲームプログラミング, 株式会社アスキー・メディアワークス, 2008 年
- [6] SoftComputing lab.  
<http://scl.m-kb.net/>
- [7] y\_miyake のゲーム AI 千夜一夜  
<http://blogai.igda.jp/>