

動的スクリプトにおけるルールの内部情報を考慮した 多様性の向上

澤田 海彦[†] 佐藤 晴彦[†] 小山 聡[†] 栗原 正仁[†]

北海道大学大学院 情報科学研究科[†]

1 研究目的

1.1 研究背景

近年、ビデオゲームはハードウェアの性能向上により、品質の高い作品が出続けている。

ゲーム AI とはゲーム内のエージェント(キャラクター)を操作する、またはプレイヤーと対峙する仮想のプレイヤーとしてゲームに組み込まれ、プレイヤーの目的達成を阻む、あるいは補助するものとして動作するものである。ゲーム AI の品質が高ければ、ゲームのプレイヤーは他のプレイヤーとゲームをプレイしているような感覚を覚える、あるいは目的達成時に達成感を得られることによってゲームに対し、高いエンターテインメント性を感じることができる。ゲーム AI の品質が高い状態とはプレイヤーがゲーム AI によって得られるゲームからの応答に対して高い知性を感じ、かつ、ゲーム AI が様々な応答を示すことによってゲームが単調になっていない状態である。

昨今のビデオゲームはより高い現実感を表現するものが増えてきており、それに伴いより知的で複雑な思考を行うゲーム AI が要求され、ゲーム AI 作成に必要な時間が増え、求められる技術力が高くなってきている。また、個々のプレイヤーによってゲームをプレイするための技術には大きな差があり、多くのプレイヤーに「面白い」と感じさせるためにはゲームの目的達成の難易度を幅広く設定する、つまりゲーム AI の強さを変動させることができるようにしなければならない。これらの要求を満たすことは困難であるため、より簡便かつ品質の高いゲーム AI を作成する手法が必要となっている。

1.2 研究目的

ゲーム AI の開発にはスクリプトを用いて行われることが多い。ここで、スクリプトとは簡易プログラム言語のことを指し、プログラマーでなくても容易に扱え、スクリプトのコーディングミスによってゲームのクラッシュが起りにくく設計されている。したがって、ゲームデザイナーが直接ゲーム AI を作成することができ、ゲーム AI 開発の効率向上といったメリットがある。しかし、スクリプトは大概、静的で、長く複雑になりやすくなりがちであり、このことがゲーム AI の行動の多様性や幅広い強さの設定を阻害している。この問題を解決するために提案された手法が動的スクリプトであり、動的スクリプトの多様性を向上するために考案された手法がマクロアクションである。本論ではマクロアクションにおける問題点を指摘し、さらに多様性を向上する方法を提案する。

2 先行研究

2.1 動的スクリプト

動的スクリプトとは Pieter Spronck によって提唱されたゲーム AI のための機械学習のひとつである[1]。動的スクリプトにおける学習の手順は以下に示すとおりである。

1. ゲームにおけるエージェントごとにルールベースを用意する。ルールベース中のルールにそれぞれ重みを設定する。
2. ゲーム開始時にルールベースからルールの重みを使用し、ルーレット選択によってルールを選び出し、エージェントを制御するためのスクリプトを生成する。
3. 生成したスクリプトを使用してゲームを行う。
4. ゲーム終了時にゲームの結果によってスクリプトの評価値を計算、評価値によってルールごとの重みに対して報酬あるいは罰を与えることによって値を更新し、2.に戻る。

成功を導いたルールの重みは大きくなり、逆に失敗を導いてしまったルールの重みは小さくなることによってスクリプトにルールが選択される確率が変化し、学習が行われる。評価値がゲームに圧勝すると高くなるような評価関数を設定した場合、出来る限り強くなるように、プレイヤーと同等の強さであるほど評価値が高くなるような評価関数を設定した場合はプレイヤーと同等の強さとなるようにルールが選択されるようになり、ゲームデザイナーの意図したゲーム AI を作成することができる。

2.2 マクロアクション

動的スクリプトには学習が進むと特定のルールに重みが集中してしまい、ゲーム AI の行動が単調になる、したがってエンターテインメント性が低下するという問題があった。その問題を解決するために提案されたものがマクロアクションである[2]。マクロアクションはルールベースの中のルールを複数選び出し、一連の行動の流れ(以降、マクロと呼ぶ)としてひとつのルールに定義しなおすことで特定のルールに重みが集中してしまうことを回避でき、ゲーム AI が様々な行動を示すことができるようにするものである。マクロアクションがマクロを生成する手順を以下に示す。

1. ルールベースのルールそれぞれに対し確率パラメータを設定する。
2. 確率パラメータを使用し、ルーレット選択で複数のルールを選び出し、スクリプトを生成する。
3. ゲームを行い、スクリプトを評価、スクリプトの数が集団サイズになるまで 2. と 3. を繰り返す。集団サイズに達した場合は 4. へ。
4. スクリプトを評価値の高いものからエリートサンプルとしていくつか選び出し、エリートサンプルであるスクリプトに含まれるルールの確率パラメータを増やし、含まれないルールの確率パラメータを減らす。
5. 2.~4. を規定の回数繰り返したら終了。

3 提案内容

マクロアクションの評価値の計算には以下に示す評価関数が使われる。

Considering Internal Structures of Rules to Improve the Diversity in Dynamic Scripting

[†]Umihiko SAWADA

[†]Haruhiko SATO

[†]Satoshi OYAMA

[†]Masahito KURIHARA

[†]Graduate School of Information Science and Technology, Hokkaido University

$$f_{div} := \frac{1}{K} \sum_{k=1}^K \|v_k - v_0\|$$

$$v_{k,j} := \begin{cases} 1, & \text{if macro } k \text{ contains rule } j; \\ 0, & \text{otherwise} \end{cases}$$

$$f_{str} := \begin{cases} 0.55 + 0.35 \frac{h_T(A)}{h_0(A)} & h_T(A) > 0; \\ 0.1 \min\left(\frac{D}{D_{max}}, 1\right) + 0.1 \left(1 - \frac{h_0(S)}{h_T(S)}\right) & \text{otherwise} \end{cases}$$

$$f := f_{str} + c f_{div}$$

数式中の f_{div} は多様性の評価値であり、 K は生成するマクロの総数、 v_k は k 番目のマクロの特徴ベクトル、 f_{str} はスクリプトの強さの評価値、 A はマクロを学習するエージェント、 S はマクロを学習するエージェントと同じ行動の選択肢を持つ敵対するエージェント、 $h_T(A)$ はゲーム内時間 T におけるエージェントの体力、 D はマクロを学習するエージェントの体力が 0 になったときのゲーム内時間、 D_{max} はゲームを打ち切るゲーム内時間である。 c は定数であり、0.25 に設定される。 f_{div} によって他のマクロに同じルールが入っていないほど多様性の評価値が高くなるようになっている。だが、この評価値の計算方法ではルールの内部情報を考慮していないため、ルールベース中に似たルールが存在したときに多様性が損なわれてしまう可能性がある。言い換えると、ルールの違いをルールにつけられている ID で判断しているため、まったく同じルールであったとしても評価値が高くなってしまいう可能性があるということである。

そこで、本論ではルールが「行動」、「対象」、「条件」で構成されている点に注目し、ルールの構成要素それぞれが異なるほど評価値が高くなるような評価関数を提案し、ゲーム AI の多様性を高め、エンターテインメント性を向上する方法を提案する。

例

体力が 50 以下のとき	(条件)
最も近くにいる味方を	(対象)
回復する	(行動)

マクロ間での多様性を高めるために次の多様性の評価値を求めるための評価式を提案する。

$$f'_{div} := \frac{1}{K} \sum_{k=1}^K \|v'_k - v_0\|$$

$$v'_{k,j} := c_1 * cnd_{k,j} + c_2 * tar_{k,j} + c_3 * act_{k,j}$$

$$cnd_{k,j} := \begin{cases} 1, & \text{if macro } k \text{ contains condition } j \\ 0, & \text{otherwise} \end{cases}$$

$$tar_{k,j} := \begin{cases} 1, & \text{if macro } k \text{ contains target } j \\ 0, & \text{otherwise} \end{cases}$$

$$act_{k,j} := \begin{cases} 1, & \text{if macro } k \text{ contains action } j \\ 0, & \text{otherwise} \end{cases}$$

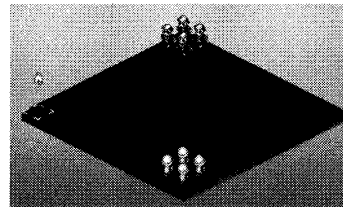
式中の c_1 、 c_2 、 c_3 は定数であり、全て 0.33 に設定した。

cnd_k は k 番目のマクロの「条件」の特徴ベクトル、 tar_k は k

番目のマクロの「対象」の特徴ベクトル、 act_k は k 番目のマクロの「行動」の特徴ベクトルである。マクロで使用されているルールの「条件」、「対象」、「行動」が他マクロと異なれば、評価値が高くなり、より多様性の高いマクロを生成できると考える。

4 実験

4.1 実験と結果



提案した評価関数により実際に多様性の高いマクロを生成することができていることを確認するために左図に示すような簡易ゲームで学習を行った。合計 4 体のエージェントで構成されるチーム同士

で戦い、どちらかのチームに所属するエージェントの体力が全て 0 になるまで戦闘を継続する。各チームには「戦士」として「直接攻撃」と「投射攻撃」を行うことができるエージェントが 2 体、「魔法使い」として「直接攻撃」、「投射攻撃」、「魔法による 2 種類の攻撃と 1 種類の回復」を行うことができるエージェントが 2 体存在するものとする。マクロを学習する「魔法使い」のエージェントを 1 体決め、「条件」に 7 つ、「対象」に 7 つ、「行動」に 5 つの選択肢を設けて実験を行った。生成されたマクロの例を以下に示す。

```

DIRECT_ATTACK(CLOSEST_ENEMY)
IF GET_HP(CLOSEST_FRIEND) > 50 MAGIC_HEAL(CLOSEST_FRIEND)
IF GET_HP(WEAKEST_WIZARD) > 50 MAGIC_ATTACK(WEAKEST_WIZARD)

DIRECT_ATTACK(CLOSEST_ENEMY)
STRONG_MAGIC_ATTACK(CLOSEST_ENEMY)
IF GET_HP(WEAKEST_FRIEND) > 50 MAGIC_HEAL(WEAKEST_FRIEND)

MAGIC_HEAL(CLOSEST_FRIEND)
IF GET_HP(CLOSEST_FRIEND) > 50 MAGIC_ATTACK(CLOSEST_FRIEND)
IF GET_HP(WEAKEST_WIZARD) < 50 BOW_ATTACK(WEAKEST_WIZARD)
    
```

4.2 考察とまとめ

「条件」は「無条件」、「対象の体力が 50 以上」、「対象の体力が 50 未満」、「対象」は「最も近い敵」、「最も近い味方」、「最も体力の低い敵魔法使い」が多く選ばれ、他の「条件」、「対象」はほとんど選ばれなかった。スクリプトの強さの評価値が高くなるよう、これらの選択肢が選ばれるようになったと考えられる。だが、「行動」に関しては 5 つの選択肢が均等に選ばれ、かつ、マクロによって適用される「条件」、「対象」、「行動」が全く同じになってしまうものはなかったことより、多様性を高めることができたと思う。今後の課題としては多様性を求める計算式 f'_{div} を改良し、「条件」、「対象」に選ばれるものを増やし、より高い多様性を実現していきたい。

参考文献

[1] Pieter Spronck, Marc Ponsen, Ida Sprinkhuizen-Kuyper, Eric Postma. Adaptive game ai with dynamic scripting. *Machine Learning*, pp.217-248, 2005

[2] Istv'an Szita, Marc Ponsen, and Pieter Spronck. Effective and Diverse Adaptive Game AI. *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES*, VOL. 1, NO. 1, 2009