

IF-THEN ルールで制御された 2-スネークゲームの実装について

On a 2-Snakes Game Controlled with IF-THEN rules

コウ ルチン

指導教員 三堀 邦彦

拓殖大学大学院 工学研究科 電子情報工学専攻 三堀研究室

キーワード：2-スネークゲーム，IF-THEN ルール，マルチエージェントシステム

1. はじめに

この研究では、プログラムで制御されたスネークゲームについて考える。このゲームは比較的簡単なルールで構成されるが、その攻略に一定の戦略性が要求される。こうしたゲームは、最適化手法の性能を評価・改善するための試験問題として役立つ[1][2]。これらの報告の中では、しばしばエージェントの概念が導入される。エージェントは、プログラム上で周囲からの情報に基づき行動を決定し実行する。先ほど挙げた報告のほとんどで、エージェントの行動方針がとて複雑になる。我々は「シンプルな行動方針で、このゲームをどこまで攻略できるのか？」という問題にチャレンジする。

本研究で扱うスネークゲームの構成を図 1 に示す。このゲームはヘビ・エサ・ステージで構成される。このゲームは通常ヘビを 1 匹とし、それを人間が操作する。その目的は「ヘビを操作して、エサを食べ続けること」である。本研究では、これを「2 匹のヘビがエサ 1 個を取り合うゲーム」に拡張し、両方のヘビの制御方針を IF-THEN ルールで与えたシステムを構築して、そのプログラムを作成した。このとき、ゲーム全体はマルチエージェントシステム(MAS)になる[3]。MAS は、お互いに影響を及ぼし合う複数のエージェントで構成される。このゲームにも様々な最適化アルゴリズムが適用されている[2]。この発表では、このプログラムの構成と基本的な結果について報告する。

2. 本論

図 1 の構成を考える。その目的は、「画面上に表示されるヘビを制御して相手より早くエサを食べること」である。ステージは正方形であり、周りを壁に囲まれている。その 1 辺は 640 である。ゲームが開始されると、ヘビとエサが配置される。2 匹のヘビはステージの中心に点対称な与えた位置に、エサはランダムに重ならないように配置される。

ヘビとエサは、正方形のブロックを単位として表現され、その 1 辺は 10 である。エサは 1 個のブロックで表現される。ヘビは、連結された正方形のブロックで表現される。先頭のブロックがヘビの頭、後ろのブロックがヘビの胴である。ヘビの頭は

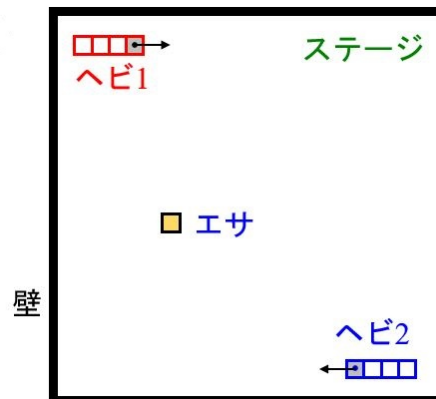


図 1 2-スネークゲームの構成

表 1 (d_x, d_y) の値と変数 u の値の対応

	+	2	3	4
	0	-1	0	1
d_y	-	-4	-3	-2
	u	-	0	+
		d_x		

上下左右に制御される。ヘビの 1 番目の胴は、頭を追って移動する。2 番目の胴は、1 番目の胴を追って移動する。3 番目以降も同様である。ヘビの頭がエサにぶつかれば「エサを食べた」とする。その直後に、ヘビの体の長さが 1 つ増え、エサが再配置される。ヘビの頭が、壁または自分の体にぶつかればゲーム終了になる。

以上の枠組みの下で、2 匹のヘビがエサを取り合う。相手よりも早くエサを食べた方が勝者になる。勝者が決まるとゲーム終了となる。

3. アルゴリズム

全体のアルゴリズムを以下に示す：

1. ゲーム開始後、両方のヘビとエサが配置される
2. (a)から(h)まで繰り返す
 - (a)エサがなければ、エサが配置される
 - (b) 両方のヘビの頭とエサの座標を取得する
 - (c) 両方のヘビとエサを描く
 - (d) どちらかのヘビの頭が壁または自分の体にぶ

つかれば、ゲーム終了となる

(e) どちらかのヘビがエサにぶつかれば、それがエサを食べ体の長さを 1 増やす

(f) 各ヘビが頭の制御則に従い次の方向を決める

(g) ヘビが互いに接近したら、衝突回避則に従い両者の頭の方向を決める

(h)ヘビを移動させる

頭の制御則では、より短い経路を見つけさせる。ヘビの頭の中心の座標を (x_h, y_h) 、エサの中心の座標を (x_f, y_f) とし、それらの差を $d_x = x_h - x_f, d_y = y_h - y_f$ で定義する。これらの符号の組合せで、ヘビとエサの位置関係を分類する。そのために変数 u を導入し、その値を表 1 のように割り当てる。 u の各値で、頭の方向 d_h に基づき次の d_h を決める。 $u = -4$ の場合の制御則を以下に示す：

```

|d_x| > |d_y| ならば {
  d_h ≠ 左ならば {
    頭の右側に体がなければ { d_h = 右; break; }
  }
  d_h = 左ならば {
    頭の上側に体がなければ { d_h = 上; break; }
  }
}
|d_x| ≤ |d_y| ならば {
  d_h ≠ 下ならば {
    頭の上側に体がなければ { d_h = 上; break; }
  }
  d_h = 下ならば {
    頭の右側に体がなければ { d_h = 右; break; }
  }
}

```

衝突回避則を説明する。ヘビ 1、ヘビ 2 の頭の中心の座標を各々 $(x_{h1}, y_{h1}), (x_{h2}, y_{h2})$ 、それらの差を $e_x = x_{h1} - x_{h2}, e_y = y_{h1} - y_{h2}$ とし、 $|e_x| < 30$ かつ $|e_y| < 30$ を衝突と定義する。2 匹のヘビの衝突は形式的に (a) 並走衝突・(b) 横切り衝突・(c) 正面衝突の 3 種類に分類される[4]。このゲームでは両方のヘビが同じエサに向かうため、(b) は発生してもすぐに (a) になる。また (c) は必ず間にエサが入るので衝突しない。そこで以下の (a) の回避則を構築した：

```

|e_x| < 30 かつ |e_y| < 30 ならば {
  (d_{h1} = 上 かつ d_{h2} = 上) または (d_{h1} = 下 かつ d_{h2} = 下) ならば {
    e_x > 0 ならば { d_{h1} = 右; d_{h2} = 左; }
    e_x ≤ 0 ならば { d_{h1} = 左; d_{h2} = 右; }
  }
  (d_{h1} = 右 かつ d_{h2} = 右) または (d_{h1} = 左 かつ d_{h2} = 左) ならば {
    e_y < 0 ならば { d_{h1} = 下; d_{h2} = 上; }
    e_y ≥ 0 ならば { d_{h1} = 上; d_{h2} = 下; }
  }
}

```

以上に従い、C 言語でそのプログラムを作成した。図 2 に、その実行結果のスナップショットを示す。ゲーム開始時の配置は図 1 に従い、そのヘビの長さは 3 個にした。各ヘビがより短い経路を選びながら、エサにたどり着いている。



図 2 実行結果のスナップショット。

4. まとめ

IF-THEN ルールで制御された 2-スネークゲームのアルゴリズムを構築し、そのプログラムを作成して基本的な結果を確認した。

参考文献

- [1] S.Russell and P.Norvig, “Artificial intelligence: a modern approach”, Prentice Hall (2002).
- [2] J.Chung, A.Luo, X.Raffin, and S.Perry, “Battlesnake Challenge: A Multi-agent Reinforcement Learning Playground with Human-in-the-loop”, 2nd ICML 2020 Workshop on Human in the Loop Learning, PMLR 108, (2020)
- [3] G.Weiss, “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”, MIT Press (1999).
- [4] 福井淡・岩瀬潔：図説海上衝突予防法、海文堂(2006).