ＡＩプログラム大会向けのコード作成

谷聖一　研究室　　並木 彰浩
Akihiro Namki

概要

AI のゲームの応用について理解を深める事を目的とし，以下の２つを行った．1) 情報処理学会が主催する SamurAI Coding に参加．2) コペンハーゲン IT 大学の准教授 Julian Togelius が主催する Mario AI Championship で MarioAI を作成した．1) の samurAICoding では，Agent 間の距離を最短歩数で距離を定義した距離 1 と x 座標の差と y 座標の差の和で距離を定義した距離２に基づくプログラムを実装した．2) の Mario AI Championship では，敵と障害物をジャンプする MarioAI を実装した．

1　はじめに

背景: AI(Artificial Intelligence) は，人間と同等の知能を実現する事を目標とし，人間と同様な作業をコンピューターで模倣させるシステム ([1]) などと定義されている．人間と同等の知能とは判断，理解，推理，計算，学習など挙げれば切りがない．知能という言葉を踏まえた上で AI について考察すると，車を製造する過程で使用されているロボットアームなどの産業用ロボットや，アラーム機能，デジタルカメラなどは果たして AI と呼べるのだろうか. 自律性からの視点でみればこれらは AI ではない．しかし，ロボットアームは巧みに腕を動かし車を製造し，アラーム機能は時間の計算を行い，カメラは画像処理を行い風景を写す．人間の一部を模倣していると言え，AI と表現しても差し支えないのではないだろうか．つまり AI の厳密な定義は主観的となり，個人だけの定義が存在する. 著者の考える定義は，人間と同等の知能を実現する事を目標とし，人間と同様の判断を部分的にコンピューターで模倣させるシステムである．AI の最終目標の一つは人間と同等の知能を実装する事にある. 理想としては,「鉄腕アトム」や「ドラえもん」が挙げられる．近い将来，一人に一台付き添いロボットを所有する時代がやってくるかもしれない．ロボットを開発するという点で AI の歴史の出発点を考慮すれば，1923 年に K.Kapek が“R.U.R. (Rossum's Universal Robots)”がロンドンで上演，初めてロボットという言葉が用いられた ([2])．また,1956 年にダートマス会議が開催され，AI の先駆者であるマッカシー，ミンスキー，サイモン，ニューウェルなどが参加した ([3])．AI 研究は意外にも新しい技術であり，約半世紀の歴史しか有さない．しかし．現代では数学，人類学，哲学，遺伝子学などの様々な分野で AI は応用され，現代社会の技術開発に寄与している．例えば，Honda の ASIMO は歩行し，走り，ジャンプし，挨拶を交わす ([4]).

AI 開発の一環としてゲーム分野でも AI は応用され研究が進められている．１９９７年にはコンピューターチェスであるディープ ブルーが世界チャンピヨンであるガリル カスパロフと対戦し，互角に戦っている ([5])．もちろんチェスだけではなく，様々なゲームに AI は組み込まれている．それによって表現できる人間らしい動作は，ゲームの娯楽性を高めてくれる．かく言う私も日々その娯楽を楽しむ一員であり，更に情報学科の学生であることがゲーム AI に強く興味を抱かせた.

そこで，AI について理解を深める為に SamurAI Coding に参加と MarioAI に挑戦することにした．本演習では，２つのゲームの応用をする 1)SamurAI Coding については，六角形のセルで敷き詰められた盤上に配置される Agent 間の距離を異なる２つの定義 (最短歩数で定義した距離１，x 座標の差と y 座標の差の和で距離を定義した距離２) に基づくプログラムを実装した．２つのプログラムをコスト (ソースコードを重さで表す概念) と距離計算の効果で比較して優劣を判定した．2)MarioAI については大会の概要と，現段階での Mario の動作について記述する．第２節に SamurAI Coding の概要，第３節に犬の戦略，第４節に実装，第５節に結果と考察，第６節では Mario AI Championship を述べる．

2　SamurAI Coding

2.1　大会の概要

SamurAI Coding は 2011 年度から開催されている世界の学生を対象にしたプログラミング大会である．グリー株式会社がイベントスポンサーをし，2011 年度は早稲田大学鷲崎研究室が主催した.2012 年度には情報処理学会が継承して主催した.

2.2 ゲームの詳細

2.2.1 開始（初期設定）

プレイ人数

4 チーム

盤面

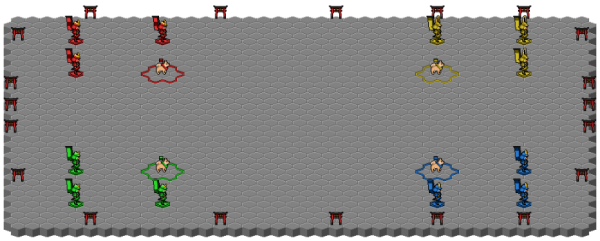
行列３９ ×(４１ or ４１) の六角形のセルが敷き詰めらている (if 奇数列４０ else ４１)

門 (Gate)

上下対称左右左右対称に五個づつランダムに配置 (合計２０個)

Agent

各チームに三人の侍と犬が用意される．初期配置はチームに対して上下左右対称 (以後, 侍と犬と表記)



図：初期配置

2.2.2 進行

ラウンド数

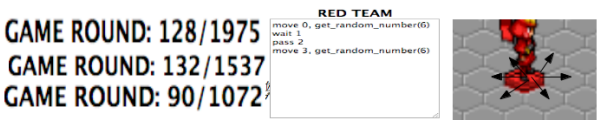
試合毎に回数変化 (１０００〜２０００)

プログラムの読み込み

ラウンド毎にソースが読まれる

Agent の行動

ラウンド毎に全ての Agent は 6 方向に 1 マス移動, または待機が可能



図：ラウンド プログラムの読み込み Agent の行動

侍の移動 (役割)

足跡を残す

犬の移動 (役割)

6 マス以内に侍を入れない (足跡は残らない)

図：侍の役割

図：犬の役割

Agent のフリーズ状態：行動不可 (待機で解除)

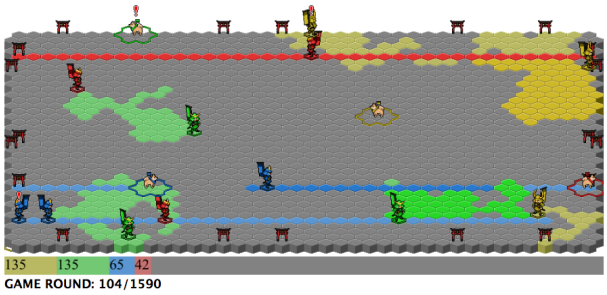
- 他の Agent との衝突
- 壁に衝突
- 犬の脅威側から侍を捕縛

図：フリーズ状態

2.2.3 終了（勝利条件）

大陸横断

ステージの上辺下辺，または右辺左辺に足跡を繋げた場合に即座に勝利 (赤チームが勝利)

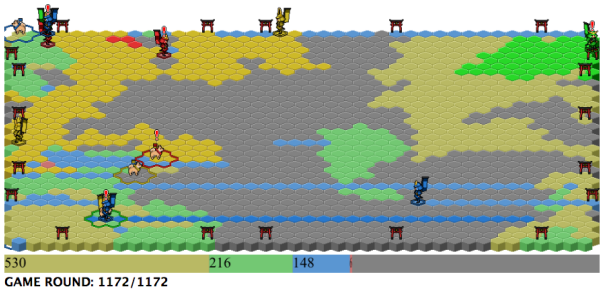


図：大陸横断

図：包囲

大陸占領

ラウンド終了時，最も足跡で占領したチームが勝利 (黄色チームが勝利)



図：大陸占領

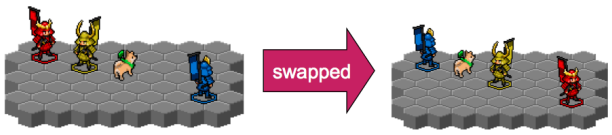
2.3 特別ルール

cyzygy

斜線に 4 以上の Agent が並んだ場合，対応した位置交換が発生 (if 奇数列：中心にいる Agent は変化無し)

cyzygy 発生不能期間

最初の１０ R or cyzygy が発生してから３ R



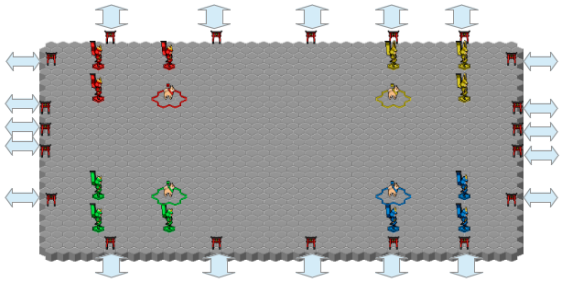
図：syzygy

包囲

足跡がセルを囲んだ場合、全て占領

門 (Gate)

対称の門に Agent がワープ可能



図：門 (Gate) の役割

コスト制約

全体で使えるコスト：２０００ (Agent ４人で２０００コスト共有)

実行されると 1 コストが加算される文:条件式 (if)，繰り返し文（while)，変数，配列，定数，演算子

例)

a = []...(3)

b = 0...(3)

wihle b != 3...(4)

if b != 1...(4)

a[b] = b...(5)

endif b = b+1...(5)

endwhile

実行コスト（５５コスト）

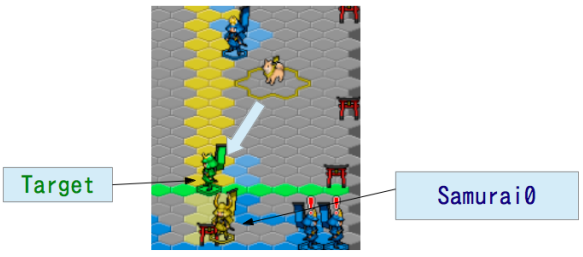
関数

組み込み関数：情報を取得する為に既存で用意されている関数

独自関数：独自で定義する関数 (中身のコストは２倍)

3　犬の戦略

大陸横断を狙う Samurai0 から，一番近い敵の Agent に突撃させる．



図：犬の戦略

3.1　大陸横断を狙う理由

Gate の存在

反対側にワープする門は大陸横断を達成するのに効果的

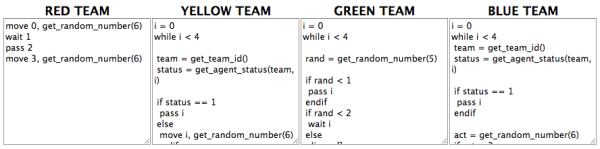
コスト制限

２０００コストを４人で共有するとなると時間がかる長期戦になる大陸占領よりも，短期決戦である大陸横断を採用

4　実装

4.1　実装環境

ブラウザ Google Chrome を使用し Web 上で動作を確認



図：ソースコードを読み込む Web ページ

4.2　実装方法

一番近い敵を探索するのに，二つの距離を定義

Target の座標：(tx,ty)

Samurai0 の座標：(sx,sy)

D(Distance) = 0

距離 1 ：最短歩数で定義

$$K1 = \begin{cases} K2(sy) & (|ty-sy|\%2=1) \\ K5(tx,sx,ty,sy) & (otherwise) \end{cases}$$

$$K2(sy) = \begin{cases} K3 & (sy\%2=0) \\ K4 & (otherwise) \end{cases}$$

$$K3(tx,sx,ty,sy) = \begin{cases} K5(tx,sx,ty,sy-1),D+1 & (tx<sx\&\&ty<sy) \\ K5(tx,sx,ty,sy+1),D+1 & (tx<sx\&\&ty>sy) \\ K5(tx,sx+1,ty,sy-1),D+1 & (tx>sx\&\&ty<sy) \\ K5(tx,sx+1,ty,sy+1),D+1 & (tx>tx\&\&ty>sy) \end{cases}$$

$$K4(tx,sx,ty,sy) = \begin{cases} K5(tx,sx-1,ty,sy-1),D+1 & (tx<sx\&\&ty<sy) \\ K5(tx,sx-1,ty,sy+1),D+1 & (tx<sx\&\&ty>sy) \\ K5(tx,sx,ty,sy-1),D+1 & (tx>sx\&\&ty<sy) \\ K5(tx,sx,ty,sy+1),D+1 & (tx>tx\&\&ty>sy) \end{cases}$$

$$K5(tx,sx,ty,sy) = \begin{cases} K7(tx,sx,ty,sy) & (tx=ty||sx=sy) \\ K6(tx,sx,ty,sy) & (otherwise) \end{cases}$$

$$K6(tx,sx,ty,sy)=\begin{cases}K5(tx,sx-1,ty,sy-2),D+2 & (tx<sx\&\&ty<sy)\\ K5(tx,sx-1,ty,sy+2),D+2 & (tx<sx\&\&ty>sy)\\ K5(tx,sx+1,ty,sy-2),D+2 & (tx>sx\&\&ty<sy)\\ K5(tx,sx+1,ty,sy+2),D+2 & (tx>sx\&\&ty>sy)\end{cases}$$

$$K7(tx,sx,ty,sy)=\begin{cases}K11(D) & (tx=ty\&\&sx=sy)\\ K8(tx,sx,ty,sy) & (otherwise)\end{cases}$$

$$K8(tx,sx,ty,sy)=\begin{cases}K9(tx,sx,ty,sy) & (sy\%2=1)\\ K10(tx,sx,ty,sy) & (otherwise)\end{cases}$$

$$K9(tx,sx,ty,sy)=\begin{cases}K7(tx,sx-1,ty,sy),D+1 & (tx<sx\&\&ty=sy)\\ K7(tx,sx,ty,sy-1),D+1 & (tx>sx\&\&ty<sy)\\ K7(tx,sx+1,ty,sy),D+1 & (tx>sx\&\&ty=sy)\\ K7(tx,sx,ty,sy+1),D+1 & (tx>sx\&\&ty>sy)\end{cases}$$

$$K10(tx,sx,ty,sy)=\begin{cases}K7(tx,sx-1,ty,sy),D+1 & (tx<sx\&\&ty=sy)\\ K7(tx,sx,ty,sy-1),D+1 & (tx<sx\&\&ty<sy)\\ K7(tx,sx+1,ty,sy),D+1 & (tx>sx\&\&ty=sy)\\ K7(tx,sx,ty,sy+1),D+1 & (tx<sx\&\&ty>sy)\end{cases}$$

$$K11(D)=\Big\{D$$

メリット：正確に距離を計算可能
デメリット：コストが膨大

距離２： x 座標の差と y 座標の差の和で定義

D=｜(tx-sx)｜+｜(ty-sy)｜
メリット：コストが安価
デメリット：最短ステップ数との誤差

4.3 検証

4.3.1 距離１で探索と距離２で探索とのコスト差

１ Agent 分のコスト差

距離２よりも距離１が３０コスト多い

全体の Agent 分のコスト差

(１６) － 自軍 (４)=１２
３０ × １２= ３６０

4.3.2 犬の追跡プログラムのコストを足した場合の平均

犬の追跡プログラムコスト

１００コスト

距離１での探索の場合

１００＋７６０＝８６０コスト

距離２で探索の場合

１００＋４００＝５００コスト

4.3.3 採用した距離

距離１を採用

5 結果と考察

考察

追跡は予想通りに動作したと思われるが，距離２でもコストがかかりすぎてしまった.（４００〜５００コスト）

結果

２０チーム中１７位

今後の課題

コスト削減を目指して、他にコストを回せるようにする

6 marioAI

6.1 大会の概要

Mario AI Championship はペンハーゲン IT 大学の准教授 Julian Togelius が主催する大会である．公式ページからダウンロードが可能である Mario AI Benchmark source package でゲームを動かし，Java 言語で Mario をプログラムから操作させる，大会は 4 つの競技で設けられ，参加者を得点で競わせる．

図：Mario AI Championship

6.2 競技

GamePlayTrack(推論型)

MarioAI を一度動かして, 得点を競う．

LearningTrack(学習型)]

MarioAI を何度か動かして学習させ，規定回数内で得点を競う．

LevelGenerationTrack(ステージ生成)

ステージ生成に関して得点を競う．

TuringTestTrack(人間味ある AI)]

どれだけ人間の動作に近い MarioAI かを競う．

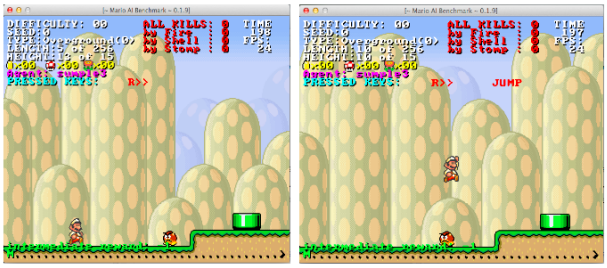
6.3 挑戦した推論型のルール

- 敵を倒した数
- コインを取得した数
- ゴールまでの時間

以上の 3 つでスコアを算出

6.4 現在の実装した MarioAI

壁と敵をジャンプで避けるプログラムを作成



図：MarioJUMP!

6.5 問題点

- 穴を検知する事ができない
- 上から降ってくる敵，または着地硬直後の敵と接触

6.6 今後の課題

- 穴を越える
- 敵を避けるから，倒すへ
- 目の前ではなく，その先を見る
- コインを取得

参考文献

[1] http://e-words.jp/w/AI.htm，(参照 2012-01-27)

[2] http://www.ai-gakkai.or.jp/whatsai/AIhistory.html，(参照 2012-01-27)

[3] 著者 荒屋真二 タイトル 人工知能概論 ，(参照 2012-02-11)

[4] http://www.honda.co.jp/ASIMO，(参照 2012-02-03)

[5] http://www.ne.jp/asahi/box/kuro/report/deepblue.htm，(参照 2012-02-21)