

AI プログラム大会向けのコード作成

谷聖一 研究室 荒金 誠二
Tomokazu Aragane

概要

人工知能 (AI) とは, 人間の計算, 学習, 思考, 判断などの機能を分析してメカニズムを知り, これらの機能をコンピュータに実現させようという試みである. 現在ではロボットや広告, ゲームに使われており, 情報技術にかかせない存在である. ゲーム技術者の発掘・育成を目的とした AI プログラミングコンテストが開催されている. 本演習では, そのようなコンテストの 1 つである, SamurAI Coding に参加した. 又, MarioAI Championship 向けのプログラムを制作した.

1 はじめに

人工知能 (AI) とは, 人間の計算, 学習, 思考, 判断などの機能を分析してメカニズムを知り, これらの機能をコンピュータに実現させようという試みである. 人工知能は 1950 年代から活発に研究され始めた ([1]). チューリングが 1950 年に知的活動のテストを可能にする方法としてチューリングテストを考案 ([1]). 1956 年にはダートマス会議で J.McCarthy により人工知能と呼ばれるようになった ([1]). 現在ではロボットや広告, ゲームに使われており, 情報技術にかかせない存在である. 人工知能には 2 種類あり情報から役に立つ知識を見つける学習型と, 知識をもとに新しい結論を得る推論型に分けられる. 学習型はショッピングサイト, 広告などに利用されている. 例えば, ショッピングサイトで買い物をするを買った品物と同じジャンルの商品をおすすめとしてメール等で知らせてくれる. 推論型は主にロボットやゲームがある. ロボットでは歩く, 走る, 段差があれば登る動作をさせる等である. ゲームではプレイヤーの動きに合わせて敵を登場させ, 攻撃させる等がある. AI はゲーム分野でも商品価値を高めるため, プレイヤーに面白さを提供するため応用, 研究されている. ゲーム制作に興味があるため, 本演習では, ゲーム AI について理解を深めることを目的に AI プログラミングコンテストに挑戦した. 現在 AI プログラミングにゲーム技術者の発掘・育成を目的とした様々なコンテストが開催されている. そこで, そのようなコンテストの 1 つである, SamurAI Coding に参加した. 又, MarioAI Championship 向けのプログラムを制作した. SamurAI Coding は 2011 年度から始まった新しいコンテストであり, 年に 1 回開催されている. 大会のゲームは年度によって異なるものになる. 今回は 2 回目にあたる 2012 年度に谷研究室の荒金, 並木の 2 人で参加した. SamurAI Coding には各チームにエージェン

トと呼ばれる 3 人の侍と犬が用意されている. 並木は犬の動作を, 私は侍の動作を担当した. SamurAI Coding は 1 月に終了し, 残りの期間は Mario AI Championship 向けのプログラムをできる限り制作した.

2 SamurAI Coding

2.1 概要

SamurAI Coding ([2]) は 2011 年度から開催されている, 世界中の若手エンジニア, 学生をターゲットにした大規模な国際的なコンテストである. GREE 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 年度は情報処理学会が継承して主催した.

使用する言語は samurai 言語というオリジナルの言語でプログラムを作成する. 作成したプログラムをテストするには公式ページにある Trial Game で実行できる.

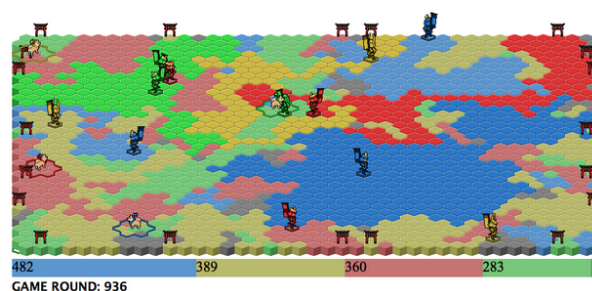


図 スクリーンショット

2.2 ルール

2.2.1 初期設定

プレイ人数は 4 人. 盤面は 39 行, 奇数行の列は 41 列, 偶数行の列は 40 列に六角形のセルが敷き詰められている. 各チームにはエージェントと呼ばれる 3 人の侍と 1 匹の犬が用意されている. 上下左右対称に 5 個ずつランダムにゲートが配置されており, 侍及び犬がゲート

の先に行こうとすると反対側のゲートにワープする。

2.2.2 進行

試合毎にラウンド数がランダムに設定されており、ラウンド数 (10002000) に上限に達すると試合が終了する。ラウンド毎にプログラムが読み込まれる。エージェントは1ラウンド毎に6方向の内1マス移動、または待機ができる。侍の役割は移動の際に足跡を残し、陣地を獲得する。犬には犬の脅威というものがあり、これにより犬の周囲6マスに敵の侍は入ることができない。また犬は足跡を残すことができない。エージェントにはフリーズ状態があり、他のエージェントと衝突、壁に衝突、犬の脅威の内側に入ることにより発生する。この状態を解除するには待機行動する必要がある。

2.2.3 勝利条件

勝利条件には大陸横断と大陸占領の2つがある。大陸横断は盤面の上辺下辺、または右辺左辺に足跡が繋がった場合、ラウンド数にかかわらず試合は終了し、勝利する。大陸占領は全てのラウンドが終了した時、最も占領したチームが勝利する条件である。

2.2.4 コスト

samurai 言語にはコストが設けられている。条件式、繰り返し文、変数、配列、定数、演算子を使うごとに1つづつコストがたまり、関数を作ると使うたびに中身の2倍コストがかかってしまう。合計2000コストが上限になっており、それを超えた場合プログラムは動かない。

2.3 侍の戦略

2.3.1 戦略

侍たちを動かすにあたって、まず考えられた戦略が序盤で一気に大陸横断を狙うか、もしくは相手の邪魔をしつつ自分の領地を広げて大陸占領を狙うかの2つになった。大陸横断を狙う戦略は相手の情報を取らず、侍2人を上下に行かせ、1人を横に行かせたところ、20ラウンド前後という速さで大陸横断を達成し、コストもかからず簡単に製作できた。一方、大陸占領をねらう戦略は相手の位置、あるマスはどのチームが占領しているのか、などの情報を取得し、それを頼りに侍たちに領地を獲得させるようなプログラムを製作した。しかし、コストが大幅にかかってしまい、コストオーバーで動作できなかった。機能を限定し、コストを削減して動かしたものの、満足のいく動作をさせることができなかった。これにより、大陸横断を狙う戦略を採用した。

2.3.2 動作

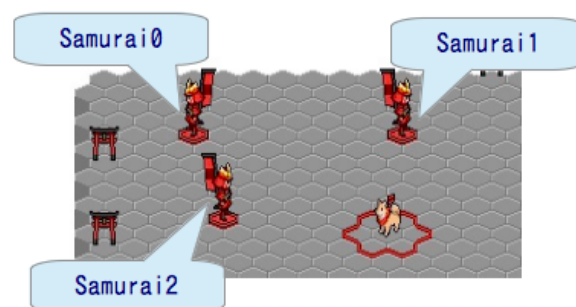


図 初期配置

上図の Samurai0 と Samurai2 の x 座標が近いいため、この2人の侍を大陸横断に専念させる。Samurai0 を上下にある近くのゲートと同じ x 座標まで真横に移動させ、下へ移動させた。Samurai2 も同様に真横に移動させてから上へ移動させた。これにより交差することになり、ラウンド数の短縮ができた。Samurai1 は左右にある近くのゲートと同じ y 座標まで上下に移動させ、真横へ移動させた。これは Samurai0,2 が失敗したときのためのリカバリーである。

Samurai0 にのみ周りの特定の3マスに誰かがいると横へ1マス移動するプログラムを実装した。これはコスト上限に達していたので3マスにしか実装できなかった。



図 特定の3マス

2.4 結果

予選は試合を40回行い、順位に応じて得点を割り振り、その合計点数で競い合われた。1位6点、2位4点、3位2点、4位0点が与えられた。結果は112点で残念ながら20チーム中18位に終わった。大会後、提出プログラムを再レビューしたところ、バグを発見した。samurai 言語はチーム毎に別々にプログラムを作成する必要がある。横へ1マス移動するプログラムが赤、黄色チームにのみ正常に動き、緑、青チームは赤、黄色チーム専用のプログラムが適用されていた。そして、味方同士で衝突する事態が発生してしまった。



図 バグ

3 Mario AI Championship

3.1 概要

コペンハーゲン IT 大学准教授 Julian Togelius 主催の大会 ([3]).Java 言語で開発を行う, 任天堂の有名なキャラクター, マリオを使った AI 大会である. 公式ページで Mario AI Benchmark source package をパッケージをダウンロードすることでプレイ可能になる. ステージはランダムで生成されゴールに辿り着くまでのモンスターを倒した数, コインを獲得した数, 残りタイムをもとにスコアを算出して競う.



図 スクリーンショット

3.2 競技

競技は4つに分けられている. 作成したプログラムを用いて課題ステージをプレイさせ, その結果を競う Game

Play Track. 作成したプログラムを指定回数学習させ, 課題ステージをプレイさせる Learning Track. ランダム生成されるステージをデザインする Level Generation Track. プログラムに課題ステージをプレイさせ, いかに人間がプレイしているかのように見せるのを競う Turing Test Track. 今回私たちは Game Play Track 向けのプログラムを製作した.

3.3 ルール

Game Play Track は敵を倒した数, コインを取得した数, ゴールまでの時間を元にスコアを算出して競い合う.

3.4 現状

壁と敵をジャンプで避けるプログラムを作成した.

4 考察

プログラムは最後に見直し, バグがないのを確認して初めてプログラムが完成したと言えることを今回反省した.

Mario AI Championship の今後の課題として穴を越えられないので越えるようにすること. スコアを伸ばすため敵を避けるのではなく倒すこと. 高所からの着地とジャンプをして着地したときに, 一瞬の硬直がある. そのときに敵を避けられず当たってしまうことがあった. マリオの前方の状況の情報を取得して, ジャンプの強弱と距離を調節できるようにしたい. これもスコアのためコインを取得するようにすること. 以上4つの課題を実装できれば, AI として第一歩を踏み出せるのではないだろうか.

参考文献

- [1] 人工知能学会. "人工知能の歴史". What's AI <http://www.ai-gakkai.or.jp/whatsai/AIhistory.html>, (参照 2013-02-21)
- [2] SamurAI Coding. <https://samuraicoding.org/>, (参照 2013-01-05)
- [3] Mario AI Championship. <http://www.marioai.org/>, (参照 2013-02-21)

