

# 2007 IEEE Student Branch Leadership Training Workshop in Japan Council 学生企画

平成 19 年 10 月 20 日

## 1 概要

本学生企画では小規模なプロジェクトを通じて、技術者として必要なスキルの情報交換を行うことを目的とする。このスキルとは具体的には、プログラムの書き方や、複数人数での作業の進め方、納期までに仕事を完成させる気合などのことを指す。

今回、五目並べのアルゴリズムの検討とその実装を、このためのプロジェクトとした。五目並べは 15x15 マス上に白と黒の石を互いに配置して、先に 5 つ連続して並べた方が勝ちという単純なゲームである。このゲームに勝つためのアルゴリズムを検討し、それを実際にプログラミング言語で記述する。

4, 5 人ずつのチームに分かれ、それぞれのチーム毎にプロジェクトを独自に進めることとする。本企画の最後には、それぞれのチームにおいて作成したプログラム同士を実際に戦わせ、どのプログラムが最も強いかを決定する。また、各チーム毎におけるプロジェクトの進め方について発表し、本企画を終了とする。

## 2 五目並べ

今回、あらかじめ五目並べの基本的なルーチンについては実装しておいた。これは学生企画に割り当てられた時間が短いことと、チーム同士での対戦が必要だったことによる。学生企画では各チームが五目並べのアルゴリズムの開発に注力できるように、五目並べを大きく 2 つに分割した。1 つ目は盤面の管理、プレイヤーの勝利判定を担当するプログラムであり、これを五目並べサーバと呼ぶ。2 つ目はある盤面が与えられたときに、次の手を考えるプログラムであり、これを五目並べクライアントと呼ぶ。五目並べサーバとクライアントは INET ドメインのストリーム型ソケット

(いわゆる TCP/IP) を介して互いに通信を行う。そのためサーバとクライアントを異なるプラットフォームで開発することが可能である。五目並べサーバはあらかじめ準備してあるので、各チームは五目並べクライアントの作成をすることとする。

### 2.1 ルール

基本的には 5 つ石を並べた方が勝ちである。両プレイヤー共に石を置く場所が無くなった場合は引き分けとする。また、後攻が不利とならないように以下に示すルールを適用する。

- 三三となるように石を置くことはできない。例えば以下の図において自分の石が  であったときに、x の位置に石を置くことである。

x

### 2.2 五目並べサーバ

五目並べサーバは C 言語で記述されたプログラムであり、Linux 上で動作する。本節では五目並べサーバについて説明する。

五目並べサーバはおおまかに以下のように動作する。ここでプレイヤーとは五目並べクライアントのことを指す。

1. インターネットソケットの作成
2. 10000, 10001 番ポートにおいてそれぞれプレイヤー 0 とプレイヤー 1 の接続待ち



5. 4. の候補からランダムに石を置く場所を決め、サーバに送信  
6. 2.~5. をサーバとの接続が切断されるまで繰り返す
- 定すると、その座標を起点として右方向に同じ色の石がいくつ並んでいるかを数える関数およびメソッド (match\_right) が実装されている。アルゴリズムの検討を行う時、これを参考にしてほしい。

1. において、C 言語で実装したクライアントプログラム (以下 C クライアント) はソケット通信により、Ruby で実装したクライアントプログラム (以下 Ruby クライアント) は telnet によりサーバに接続する。接続の際、コマンド実行時の引数により、接続先および先手が後手かを指定する。実行コマンドのフォーマットを、以下に示す。

```
$ ./renju_client address port [-b,-w]
$ ./renju_sample.rb address port [-b,-w]
```

第一引数で、五目並べサーバが動作しているマシンの IP アドレスを指定する。第二引数で、ポート番号を指定する。第三引数で、先手 (黒) か後手 (白) かを指定する。-b が先手、-w が後手である。

2. において、リクエストの際、C クライアントでは "get bin" を、Ruby クライアントでは "get text" をサーバに送信する。これは、バイナリの処理が C に、テキストの処理が Ruby に向いていることを考慮してのものである。

4. において、石を置く場所の候補を既に置いてある石の隣に限定しているのは、探索の範囲を限定することにより、計算時間を短縮することが目的である。

これらの実装は、あくまでもサンプルとして提示したものであり、必要であれば書き換えても、あるいは利用しなくても構わない。

5. において、4. で選んだ候補からランダムに石を置く場所を決めている。盤面の情報をもとにして最適な場所に石を置けるように、この部分のアルゴリズムを検討し実装するのが、今回の課題である。

なお、クライアントプログラムは、デバッグが容易に行えるよう、サーバに接続せずローカルで動作させることができる。この際、プログラムは同色の石が 5 つ並んだことによる勝利の判定は行うが、反則と規定されている三三の判定は行わない。クライアントプログラムをローカルで動作させるには、それぞれ

```
$ ./renju_local
$ ./renju_sample.rb
```

を実行する。

また、クライアントプログラムには、ある座標を指