

大貧民 PC 対人間のストレスなき実現

谷聖一 研究室 長谷川 貴啓

Takahiro Hasegawa

概要

電気通信大学が主催する大貧民大会では、各参加者が戦略をクライアントプログラムに実装し、各クライアントプログラムがサーバーと相互的にデータのやりとりをすることによりゲームが進行する。本演習では、このクライアントプログラムを読み込み、人間と PC が対戦できるシステムを構築した。

1 はじめに

ふとここ 10 年を振り返る。電車の改札を通る時は切符を通す代わりにカードをかざす。テレビはより鮮明になり、実際にクイズ番組に参加することもできる。携帯電話はボタンではなく直接画面に触れて操作。そして傾きを検知する機能はもはやあって当たり前になった。今日、人間の生活において、情報科学の寄与は著しく、IT 社会と言われるほど、もはや人間には欠かせないパートナーとなった。しかし、ここまで発展している IT であっても、人間の方が優れている部分が存在する。人間ができる思考と対等以上の働きをするシステムを実現することは、情報科学の発展において大きな目標の 1 つである。しかし、実際に人間より優れた働きが出来るかどのように判定すればよいのだろうか。その方法の 1 つとして、勝敗の決まる状況下で AI と人間が対決して、勝つことができるかどうか調べることが考えられる。そうすれば、少なくともその状況下で、人間以上の働きをするかどうか判定することができる。

これまで、コンピュータプログラムはオセロ、チェス、将棋、囲碁などのゲームで人間と戦ってきた。今回、谷研究室の長谷川、佐藤、松藤の 3 人で電気通信大学で開催された大貧民大会 (略称 UECda) ([3]) に参加し、最強の大貧民のプレイヤー AI を作りあげるべく奮闘してきた。結果は敗退だったが、そこには今の段階で世界最強の AI が存在した。大貧民は、上記に挙げた 4 つのゲームとは異なり、不完全情報多人数ゲームである。この状況下で、果たして人間はこのプログラムに勝てるのだろうか。答えは、「わからない」。なぜなら、容易に大貧民プログラムと戦うことを可能にするシステムが現段階で筆者のまわりにはないからだ。そこで、私はそれを実現するシステムを開発することにした。以下そのシステムを「大貧民 VS」と呼ぶこととする。

UEC-da では、サーバークライアントシステムが用いられる。サーバーはカードの配布、及び受理などのゲーム

進行管理を行い、クライアントはそれに従いカードを受け取ったり、提出したりしてゲームは進行する。今回、私は Windows アプリケーションとして実装しようとしているので、まずサーバープログラムに当たるプログラムを実装し、さらにクライアントプログラムを自動で大貧民 VS で使用できるようなプログラムに変換するシステムを開発する必要がある。その為に、カード交換、カード提出などモジュール別に自動で関数化し、大貧民 VS で使用できるようなファイルとして保存するシステムを実装する。今回はサーバープログラムにあたるシステムの実装に VisualC++、変換処理の実装に HSP ([2]) という神言語を使用する。

本演習以外にも、大貧民大会のクライアントプログラムと人間が戦えるシステムを、サーバークライアントシステム上での実現を目標に、開発するチームが存在している。まだ未完成 (2012 年 12 月現在) のようだが、これならば、クライアントプログラムを変換する必要もなくなり、自分の大貧民 VS よりスムーズに対人間が可能になるように思える。しかしこれには、ネット環境、あるいはプレイヤーの PC にサーバーの動作環境がないと使えないなどのデメリットも存在している。今回私は、アプリとして開発し、なるべく変換をスムーズにすることで、よりデメリットの少ないシステムを作りあげるべく、開発を進めた。

2 システムの仕組み

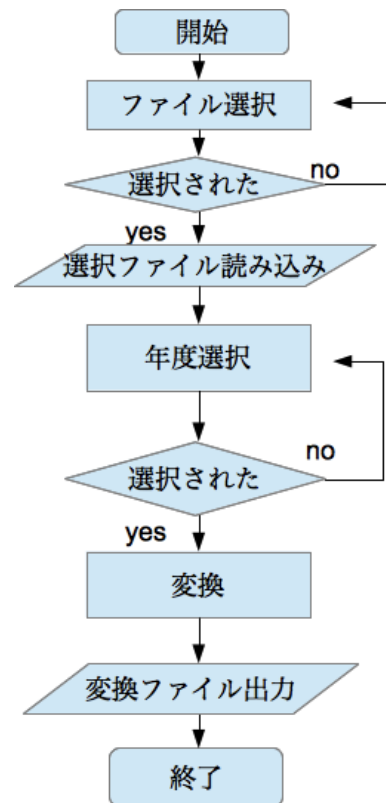
大貧民 VS は、大きく分けて 2 つに分けることができる。クライアントプログラムを変換する「変換」と、実際ゲームを遊べる「ゲーム」である。

2.1 変換

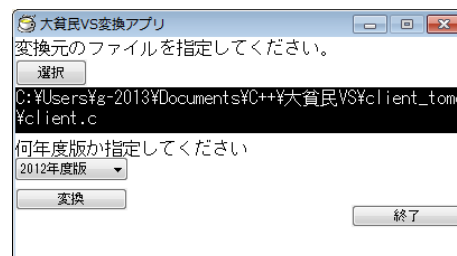
変換の説明の前に、大会参加者が、どのように大会用にクライアントプログラムを作るかを簡単に説明する。まず始めに参加者は UEC-da からデフォルトプログラム (年度で内容が異なる) をダウンロード、その後プログラ

ムのカード交換部分や、カードを提出する部分などを書き換えることによりオリジナルの戦略を持ったクライアントプログラムを開発していく。ちなみに大会側は、必要条件として上記のようにデフォルトプログラムから開発することを求めているが、特別な事情がないかぎり、筆者の知る限りでは、上記の方法以外で開発されることはない。また java 言語を使用して開発することも認められているが、今回は時間の関係で、c 言語のデフォルトクライアントプログラムから作成したクライアントプログラムのみを対象としている。

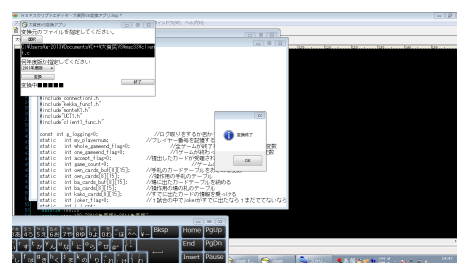
今回、上記の事を踏まえ、入力として「ファイル名」、使用したデフォルトクライアントの年度、出力として「変換後ファイル」を出力するプログラムを開発した。変換の大まかな流れとして、まず「クライアントプログラムの c ソースファイル」、及び「クライアントプログラムが使用する c, h ファイル」を読み込み、モジュール別(カードを出す処理、交換する処理など)に関数化して別ファイルとして保存する。それを「ゲーム」で include する事によって実際にゲームができる仕組みだ。本来クライアントプログラムは make を使用して動作するプログラムなので、必要なファイルが必ずしも include されていない。そこで、クライアントプログラムと同じ階層にある c, 及び h ファイル名を抽出し、必要なファイルを include に追加する仕組みにした。またデフォルトで配布されているライブラリ(中に入っている関数「カードテーブルの初期化」、「カードテーブルのコピー」、また「カード交換」、「カード提出」など)は、「ゲーム」の方で include して、プレイヤー全員で共用しようと考えていたのだが、このライブラリを直接いじるユーザーも存在した。なので、各プレイヤーが自分のライブラリを使用出来るよう、各プレイヤーの使用するライブラリ内の関数名の語尾に「1~4」をそれぞれ付加し、各ユーザーが自分の関数を使用することが出来るようにした。また大貧民 VS で使用している関数名が使われる事のないよう、大貧民 VS で使用する関数には、語尾に「_hasetaka」がついている。



図「変換」のフローチャート



図「変換」スクリーンショット 1

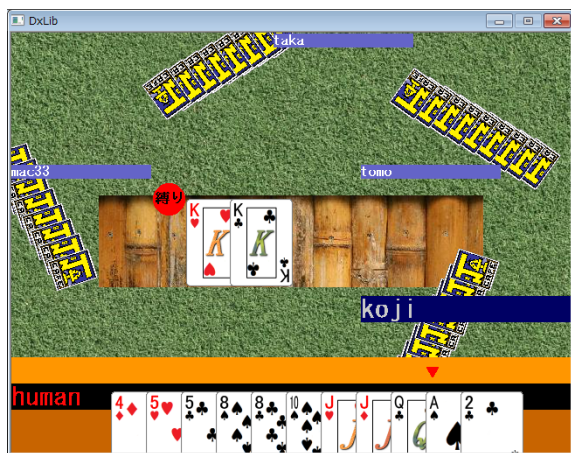


図「変換」スクリーンショット 2

2.2 ゲーム

今回、より効率的に開発する為ゲームライブラリである「DX ライブラリ ([3])」を使用した。UEC-da では 2

次元配列で表現される「カードテーブル」という、カード情報を示す2次元テーブルによりカードをやりとりする。大貧民大会でも同様に、2次元配列のカードテーブルを用いてカードのやりとりを行う事にした。主に「ゲーム」はカードの配布、受理といった、ゲーム進行をするプログラムとなっており、2013年2月現在では、4つのプログラムと1人の人間で遊ぶことが出来る用になっている。まずゲームが始まると、大貧民 VS 用に変換したクライアントプログラムを4つ読み込む。そしてカードを配り、ゲームを開始するのだが、カードは53枚あるので、プレイヤーによってカードの枚数が異なってくる。配るカードとカード枚数は、どちらも rnd 関数を用いてランダムに決められる。ゲーム中の進行は、ターンがきたプレイヤーがグローバルで見れる情報（前のカード情報、しぼりの有無、全プレイヤーの階級など）を駆使し、自分の出すカードを選択。カードテーブルに出すカードのフラグをたてて提出する。「ゲーム」プログラムはこれを受理し、グローバルで見れる情報を更新する。パスをするには、何もフラグがたっていないカードテーブルを提出してもらうことにより成立する。人間のターンでは、提出したいカードを全て選択し、決定ボタンを押すことによりカードを提出出来る。ちなみに出せないカードを出そうとすると出さないようになっている。本来何もカードを選択しないまま決定を押せば（空のカードテーブルを提出した事となり）パスとなるはずなのだが、誤操作を避けるため、パスは別に「パスボタン」が用意されている。



図「ゲーム」スクリーンショット 1



図「ゲーム」スクリーンショット 2

3 考察

プログラムの開発には時間を取られる。今回もやりたかったものが全て実装出来たとは、到底言えるものではなかった。少しでも開発時間を効率化し、より成果のあるものにするためには、「車輪の再発見」を出来る限り避けることが重要だ。今回時間の関係で、「ゲーム」で実装することのできなかったものとして、「交換」、「複数の人間が遊ぶ」、「変換」で実装出来なかったものとして「2011年度以前のクライアントプログラムへの対応」などが挙げられる。また、日本大学の北原教授から、より直感的に操作出来るよう、マウスのみで遊べるようなインターフェイスにすべきではというご指摘も頂いた。卒業演習としてはここまでだが、これからも個人でキリのいいところまで開発していきたいと考えている。

参考文献

- [1] 電気通信大学 UEC コンピュータ大貧民大会 <http://uecda.nishino-lab.jp/>
(アクセス日付:2013/2/22)
- [2] ONION software HSPTV! <http://hsp.tv/>
(アクセス日付:2013/2/22)
- [3] 山田 巧 DX ライブラリ置き場 <http://homepage2.nifty.com/natupaji/DxLib/>
(アクセス日付:2013/2/22)