

ボードゲーム戦略を題材とした Java プログラミング演習における 学生の取組み状況とアンケート結果の分析

花川直己^{†1} 岡崎敏和^{†1} 富永浩之^{†1}

問題解決型の応用プログラミングとして、ボードゲーム戦略を題材とする対戦形式での Java 演習を提案し、実際の授業で実践している。支援システム WinG を開発し、提出された戦略同士を対戦させる予備大会を運営して、順位や戦績を公開する。試行錯誤的なプログラミングを体験させ、持続的な戦略修正への動機付けを行う。2011 年度からの実践の総括として、最終大会の結果について、戦績や達成状況など、各側面から分析する。また、学生の演習レポートや自由記述のアンケートによる意見も整理し、今後の課題を検討する。

Analysis of Students' Efforts and Summary Reports in Java Programming Exercise with Board-Game Strategy

NAOKI HANAKAWA^{†1} TOSHIKAZU OKAZAKI^{†1}
HIROYUKI TOMINAGA^{†1}

We have proposed an applied Java programming exercise by board-game strategy for problem solving learning. During implementation of hand method of Gogo game, students learn realization of idea as algorithm and revision with trial and error by execution result. We also have developed support system WinG, which consists of the local review package and the contest support server. The server maintains a preliminary and the final league, which decide students' score by the result of round-robin matching. We performed an educational practice between 2011 and 2014. By introducing three kinds of dummy strategies as the standard of strength, the number of submission increased. We analyze the relation and tendency of the ranking in both leagues. We discuss analysis of students' efforts and summary reports in this exercise..

1. はじめに

本研究では、情報系学科の応用プログラミング演習として、ボードゲーム戦略を題材とする対戦形式での Java 演習を提案している[1]。ボードゲームとして採用している五五は、五目並べに石取りを加えた二抜き連珠のルールを整備したものである(図 1)。五五は、石を取ることで局面が大きく変化する。連と取という 2 つの勝利条件がある(図 2)。それぞれに攻撃と防御の優先度が考えられることから、初心者でも戦略の個性が出やすい。

問題設定として、Java 言語で作成したゲーム実行ライブラリを提供し、13×13 の盤面での五五の戦略を Java プログラミングで実装させる。学生は、Computer クラスを継承したサブクラスで、着手メソッド calc_hand() をオーバーライドする。calc_hand() は、局面と直前の相手の着手を引数とし、各枱の評価値を求め、次の着手を返却する。局面は、State クラスのインスタンスで、盤面の石の配置や取った石の個数を保持している。

戦略の作成手順は、図 3 のように行う。まず、戦略の方針に従って、各枱の評価値を求める。評価値の高い位置のうち、より効果的な位置を着手として決定する。評価値は、基本的には、局面のパターンを検出し、if-then 形式のプロダクションルールである。この個々の戦術の集成が戦略と

なる。評価値は、経験的に割り当てた値から、実戦を通して調整していく必要がある。また、局面パターンのより詳細な判別に基づいて精密化していく。学生には、プロトタイプソースコードを提示し、最低限必要な処理をコメントで指示しておく(図 4)。典型的な配置パターンの実装から始め、独自の局面分析に進んでいく。

対戦では、先手後手の 1 組で 1 試合とし、勝敗で勝点を付ける。1 勝 1 敗では、取った石の数で優勢を決め、同数は引分とする。戦略の評価として、総当たり戦での勝点の合計で順位を決める。ただし、全体の評価は、戦績だけでなく、戦略の自己評価を行った総括レポートも加味する。

2. 支援環境 WinG

このような戦略作成を支援するため、支援環境 WinG を開発している(図 5)。ローカル側 WinG-LA では、対戦の実行や、戦略のデバッグを効率的に行うモジュールを提供する。また、戦略検討に用いる各種のサンプルを用意する[2][3]。サーバ側 WinG-CS では、提出された戦略同士を対戦させる大会を運営し、ランキングや戦績を公開する。これにより、試行錯誤的なプログラミングを体験させ、持続的な戦略修正への動機付けを行う。

ローカル支援ツール WinG-LA は、学生の躓きを減らし、全体的な戦略のレベルアップを図るための支援を行う。WinG-LA およびゲームの実行ライブラリは、大会運営サー

^{†1} 香川大学
Kagawa University

バから事前にダウンロードしておき、4段階にモデル化した演習の進行フェーズに沿って、各種サンプル素材と合わせて利用する。

WinG-LA は、4つのモジュールから構成される(図6)。従来は別個に起動していた各モジュールを統合し、1つの支援ツールへと改良した。各モジュールは、タブ形式のGUIで、自由に切り替えることができる。各モジュールが扱うデータも共有され、機能の有機的な連携を実現する。

1. 置けない着手(盤外、重置、連打)は反則負け
2. 打って(石取り前に)三々になったら、禁手で負け
3. 石を取っても、相手の五連を崩せなければ負け
4. 相手の長連から石を取り、五連ができれば負け
5. 自分で石を取り、五組に達すれば勝ち
6. 相手に崩されない五連を作ったら勝ち

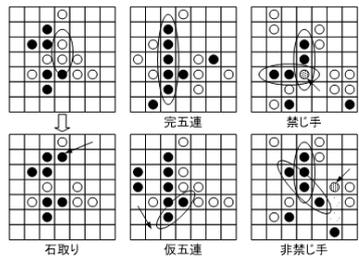


図1 五五のルールと局面

着手	先手が黒石、後手が白石を使用し、交互に打つ2個並んだ相手の石を両側から挟んで取れる1手で複数の方向の2連を同時に取ることも可能後から石間に置いて2連になったものは取れない
勝利条件	完全な五連を作るか、10個(5回)石を取ると勝ち 完全な五連とは、挟んで取られない五連のこと 長連は五と認められない
禁手	「三々」は、先手後手共に禁手である 石を取った後の「三々」は、禁手とならない

図2 勝敗条件

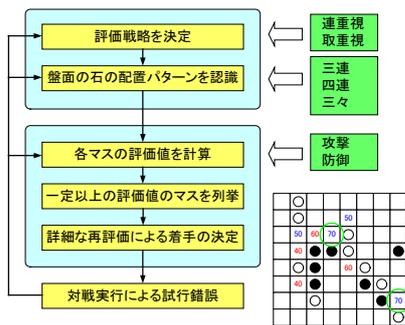


図3 戦略プログラムの組立て方

3. 大会運営

作成中の戦略にフィードバックをかけて、持続的に演習に取り組ませるため、最終大会の前に予備大会を設ける。予備大会中に提出された戦略は、サーバ上で他の戦略と対戦し、定期的に結果が更新され、順位が公開される。順位の推移を見て、自分の戦略を再検討し、状況に応じて戦略を修正していく。予備大会後に、提出した戦略の強さを総

合的に判断し、最終大会の戦略を選択する。これらの戦略同士で総当たり戦を行い、最終順位を決定し、成績に反映させる。このように、自分の戦略を常に評価する機会を設けることで、試行錯誤の繰返しを動機付ける(図7)。

過去の演習実践を踏まえ、学生の競争意欲の刺激を図り、目標を明確化するため、強さの基準となる3段階の指標戦略を用意している[4]。強さの段階(強・中・弱)ごとに、複数の指標戦略を用意する。予備大会の開始時から幾つか登録し、大会序盤の対戦相手とする。また、途中から指標戦略を追加して、中だるみを防ぐ目標としても用いる。これらは、バイナリのみで配布されるサンプル戦略でもある。

また、見かけ上の強さや相性によるバイアスを減らすため、対戦相手の勝点度で重み付けした重付勝点度(WWG)を導入している。WWGは、区間[0,1]内の実数値である。初期値として各自の勝点を区間内に正規化したものを与え、個々の対戦の勝点に相手の暫定値を重みとして乗じて計算する。これを繰り返す。値の変化が閾値以下になるまで、あるいは順位が収束するまで行う。実際には、4~5回程度でほぼ安定する。

WWGによって、ノイズとなる弱い戦略にいくら勝っても、重みが低いため勝点度はあまり上がらず、それらに対する勝敗の影響が軽減される。逆に、強い戦略に勝つと、金星のような意味を持つ。また、結果として、中位の戦略でもどれに勝ったかで勝点度への寄与が異なり、戦略の優劣がより明確になる。

```
public Hand calc_hand(State state, Hand hand) {
    :
    // 置石チェック
    // 各マスの評価値
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            // 埋まっているマスはスルー
            if (values[i][j] == -2) { continue; }
            // 相手の五連を崩す → 1000;
            // 勝利(五取) → 950;
            // 勝利(五連) → 900;
            if (check_run(cell, mycolor, i, j, 5)) {
                values[i][j] = 900;
                continue;
            }
            // 相手の石を取る → 300;
            :
            // 三々の禁じ手は打たない → -1
            // ランダム
            if (values[i][j] == 0) {
                int r = (int) Math.round(Math.random() * 15);
                if (values[i][j] < r) { values[i][j] = r; }
            }
            // 四々や四三の判定
            // 飛び三や飛び四の判定
            // 三をどちらで止めるか
            // 着手の決定
            return decide_hand();
        }
    }
}
```

図4 戦略のサンプルコード

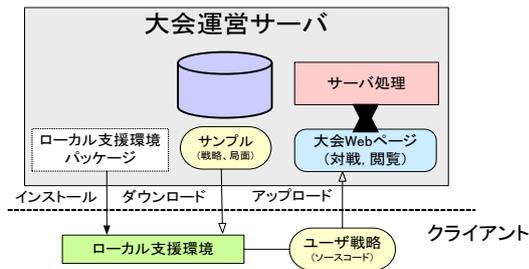


図5 支援環境 WinG のシステム構成

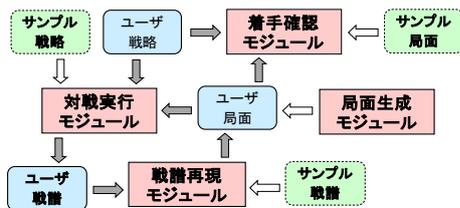


図6 WinG-LA のモジュール構成

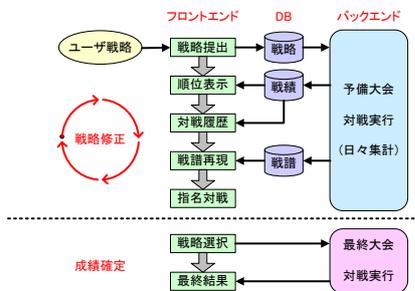


図7 予備大会と最終大会

4. 演習実践と戦績の分析と考察

本研究の演習実践は、2005年度から実施している。2011年度からは、カリキュラムの変更に伴い、必修の課題として実施している。システムも、旧版から新版に移行した。WinG-CSの主な改良点は、中間目標として教員側から3段階の強さの指標戦略を提示して対戦相手とした、戦略のレーティングに相当する重付勝点度 WWG を導入して予備大会での順位の妥当性を高めた、WWG に基づいた間引対戦に変更して順位の反映を迅速にした、などがある(図8)。2014年度の演習実践では、サーバトラブルが発生し、大会開始が、例年より2ヶ月ほど遅れた。しかし、開発環境は例年と同じ頃に提示しており、実質の開発期間は長い。

本論では、2011年度以降の最終大会について分析する。予備大会での提出状況については、図9の通りである。2012年度は、戦略数が全体で824個、1人当たり18個であった。2013年度は、戦略数が全体で942個、1人当たり約25個となり、前年の戦略数を大きく上回った。最大で150個近い戦略を提出する受講者もいた[5]。

図10は、最終戦略の WWG の頻度分布である。2011年度は、WWG が 0.3~0.4 前後の戦略が最も多く、中位から上位の戦略は少ない。2012年度は、下位から上位にかけて緩やかに人数が減少しており、全体のレベルが均されて

きたと言える。2013年度は、下位の戦略が減少し、上位の戦略が増加している。一方で、WWG が 0.5 前後の戦略は減少しており、全体的に、強い戦略と弱い戦略の二極化が起こっている。2013年度における予備大会での提出数と WWG の相関性について図11に示す。WWG と提出数には、中程度の正の相関があることがわかる。しかし、予備大会の提出数が少ないが、上位に位置する学生も存在している。これは、ローカル環境での試行錯誤を熱心に行っているのではないかと考えられる[6]。

次に、勝敗状況を見る(図12)。先手の黒と後手の白では、ほぼ互角であるが、わずかに先手の黒が優位である。勝因については、全体的には五連が多い。2012年度以降は、設計のミスや実装のバグによる反則が減っている。2012年度は、勝因に五連が目立ち、完勝(2勝0敗)も多い。2013年度に均衡しているのは、非常に弱い戦略が減り、好勝負が増えたためと思われる。2014年度は、2013年度よりも僅勝が増えており、より好勝負が増えたと考えられる。

演習の最初には、プロトタイプとしてのサンプルコードを提示している。そこに戦術として示した典型的な配置パターンの実装状況は、図13のようになっている。2012年度から、禁じ手の判定、五取の実現と阻止、飛びの判定の実装が大きく向上している。仮五連崩しと取の回避(相手を取られる枡に置かない)の実装は、2013年度に向上した。乱数での重み付けは、どちらでもよいときは、できるだけ中央に置くというものであり、戦略が均衡しているときに効果がある。しかし2014年度は、全体的に実装状況が悪化している。先述したが、2014年度は、サーバのトラブル等によって、予備大会の開始が2ヶ月ほど遅れた。そこで学生のモチベーションが下がったのではないかと考えられる。勝敗状況と実装状況から、2014年度は、2013年度に比べ、低レベルな試合展開だったと考えられる。上位戦略同士の対戦は、好勝負だったが、中下位戦略同士の対戦は、おもしろみのない勝負だったと思われる。

	項目	2011	2012	2013	2014
演習実施	授業	必修	必修	必修	必修
	期間(週)	5	8	5	6
	人数	35	44	37	36
	リーグ	1	1	1	1
作成支援	サンプル戦略	○	○	○	○
	対戦実行	○	○	○	○
	局面生成	○	○	○	○
	戦譜再現	○	○	○	○
大会運営	着手確認	◎	○	○	○
	中間大会	—	—	—	—
	予備大会	○	○	○	○
	指標戦略	◎	○	○	○
	重付勝点度		◎	○	○
	間引対戦			◎	○

図8 演習実践の概要

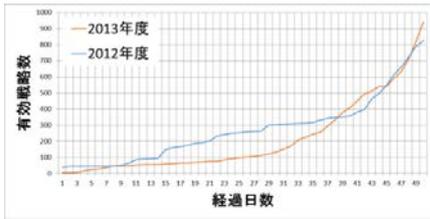


図9 提出状況

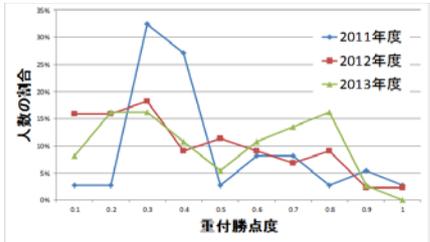


図10 得点分布

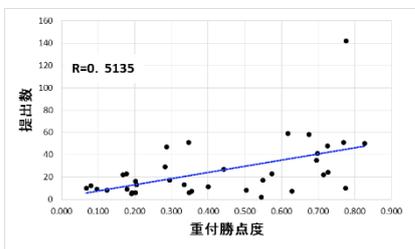


図11 2013年度の提出数とWWGの相関性

分類	項目	2011	2012	2013	2014
実施	受講者	35	44	37	35
	対戦数	595	946	630	595
	試合数	1190	1892	1260	1190
勝者	黒対白	52-48	51-49	51-49	48-52
勝因	五連	55%	74%	68%	69%
	五取	36%	21%	27%	27%
	反則	9%	5%	5%	4%
手数	平均	37	34	33	36
戦況	完勝	66%	73%	66%	64%
	僅勝	26%	20%	23%	27%
	引分	8%	8%	11%	10%

図12 各年度の最終大会の勝敗状況

分類	項目	2011	2012	2013	2014
例示した機能の実装	禁止手判定	51%	70%	78%	58%
	仮五連崩し	34%	48%	70%	36%
	五取の実現	69%	80%	86%	72%
	五取の阻止	63%	84%	78%	69%
	飛びの判定	46%	80%	84%	44%
学生の工夫	序盤の定石	23%	20%	17%	22%
	乱数での重み付け	51%	18%	46%	17%
	取の回避	20%	20%	40%	17%
不真面目な学生	サンプルコード通り	9%	0%	0%	3%
	完全ランダム	0%	0%	0%	0%

図13 各年度の最終大会の達成状況

5. 総括レポートの分析と考察

本演習では、演習終了後に総括レポートとして、紙媒体での演習総括を提出してもらう。学生は、問題分析、基本戦略、全体設計、詳細設計、実行結果、考察の6つの観点から、本演習における行動についてまとめる。評価は全て、0~5の6段階である。各観点について、評価をつけ、それらの合計からレポート点を算出する。

2013年度と2014年度の2年間のレポートを採点し、最終大会の順位との相関を調べた。

2013年度を図14に示す。大会順位の高い学生は、レポートもしっかりと書けていることが多く、順位の低い学生は、レポートへの取組みが不十分であることが多い。このように、順位とレポート点数には正の相関が認められた。

一方で、2014年度のレポートでは、あまり順位とWWGの相関は見られない(図15)。

6. 後記の分析と考察

総括レポートに、自由記述として、演習についての意見や感想を書いている。これらの記述を、言及している項目それぞれについて、肯定的意見と否定的意見に整理した[7]。

大会運営について、図16に示す。肯定的な意見では、予備大会を設けることでモチベーションが上がったという意見が多かった。対して、大会終盤でのサーバトラブルの回避や、対戦結果の表示の高速化を求める意見が多かった。受講生には、コンテスト期間中に順位の指標であるWWGについて説明を行ったが、周知が足りなかったようである。

課題題材について、図17に示す。直前に行うポーカー演習は個人で実行でき、結果から作戦を練り直すことのできる一人ゲームを用いた演習である。対して、五五は対戦形式であり、相手との勝敗によって順位が決定する。そのため、意欲がわいたという意見が多かった。一方で、ポーカー演習は、友人と相談しながら実装できるが、五五は対戦形式であるため、友人に作戦がばれないよう、一人で黙々と進めるのが寂しいという意見もあった。

戦略実装について、図18に示す。複数の学生がWinG-LAで、着手確認や対戦再現などを行っていることがわかった。また、戦略実装において、ソフトウェア工学的な開発手順を体験できたと、肯定的な意見が集まっている。しかし、戦略アルゴリズムの発想が難しく、追加で講義を行ってほしいという意見も出た。

授業形態について、図19に示す。概ね好意的な意見が集まった。大会形式での演習は、学生の意欲を刺激に効果的であるといえる。また、戦略を題材とするプログラミング演習は、実装を進めると勝率や得点上がり、やりがいがあるという意見が多かった。学生の中には、一種の育成ゲームをする感覚で、戦略の実装を進める者もいるようだ。システム機能について、図20に示す。

WinG-LAについては、機能が充実しており、実装中に躓

くことがなかったという意見が多かった。一方で、起動方法や、戦譜の読込方法が煩雑であり、使いづらいという意見があった。また、利用できるメソッドなどをまとめたAPI仕様書に不足があるという意見もあった。また、WinG-LAだけで、提出や大会における対戦履歴の確認などが完結するようにしてほしいという意見もあった。

WinG-CSについては、対戦履歴の読込が遅い、エラーの状況が出力されないなど、機能不足やエラーに関する意見が多く書かれていた。

ID	順位	問題分析	基本戦略	全体設計	詳細設計	実行結果	考察
T001	28	4	3	4	4	3	3
T002	4	4	3	4	3	3	3
T003	30	3	2	3	3	2	3
T004	25	4	3	3	3	3	4
T005	28	3	3	3	3	2	3
T006	34	2	3	2	2	1	4
T007	2	3	3	4	4	2	3
T008	32	4	3	3	2	3	4
T009	13	3	4	5	5	3	3
T010	6	4	5	5	5	4	4
T011	9	4	3	4	4	2	3
T012	10	4	3	4	4	2	3
T013	10	4	4	3	4	3	4
T014	8	5	4	5	5	5	5
T015	33	4	2	2	3	3	3
T016	1	5	5	4	5	4	4
T017	14	3	5	3	2	2	3
T018	27	4	4	3	2	2	3
T019	6	5	4	3	5	4	4
T020	3	4	5	4	5	3	5
T021	15	3	2	0	3	2	1
T022	18	4	3	3	4	4	4
T023	17	3	3	3	3	3	3
T024	12	4	3	3	3	2	3
T025	19	4	3	2	3	2	3
T026	26	4	2	2	3	3	2
T027	31	4	3	3	3	3	3
T028	15	3	3	4	4	2	3
T029	24	3	3	3	4	3	3
T030	35	4	4	5	5	3	3
T031	22	3	3	3	3	2	4
T032	23	4	5	5	3	4	4
T033	35	3	3	3	3	2	3
T034	5	4	3	3	3	4	3
T035	37	3	3	3	3	2	3
T036	19	3	3	4	4	3	3
T037	21	3	2	3	4	3	3

図 14 2013 年度のレポート評価

ID	順位	問題分析	基本戦略	全体設計	詳細設計	実行結果	考察
S001	35	3	2	2	1	1	0
S002	3	4	3	4	4	3	3
S003	36	3	3	3	3	4	3
S004	27	4	3	3	4	4	3
S005	4	2	3	2	3	3	3
S006	28	4	4	3	4	2	3
S007	31	3	3	4	3	4	3
S008	15	4	4	3	5	3	4
S009	32	3	2	2	2	2	3
S010	33	3	3	3	3	2	3
S011	12	3	3	3	3	3	3
S012	19	5	4	4	5	3	5
S013	21	4	4	3	4	3	4
S014	8	2	2	3	4	3	4
S015	20	3	2	2	5	2	3
S016	6	5	5	5	5	4	4
S017	17	3	3	3	4	2	4
S018	23	4	3	4	5	3	4
S019	13	3	4	3	3	4	3
S020	11	3	3	3	4	3	3
S021	5	2	4	3	4	3	4
S022	14	2	3	3	4	2	2
S023	7	2	3	4	3	4	4
S024	9	3	3	3	5	3	4
S025	2	2	3	2	3	2	3
S026	15	3	4	3	5	5	4
S027	1	3	4	4	5	4	4
S028	25	3	3	3	4	2	3
S029	30	3	4	3	4	4	4
S030	21	3	3	4	4	4	3
S031	29	4	4	3	2	3	3
S032	26	2	3	2	4	4	3
S033	18	3	4	3	3	4	3
S034	34	3	4	4	5	4	4
S035	24	5	5	4	5	4	4

図 15 2014 年度のレポート評価

- ・予備大会は、自分の進捗を確認でき、今後の実装計画を立てることができた
- ・予備大会を設けることで早めから着手できた

- ・大会終盤にサーバが停止するのは管理不足だ
- ・提出してから結果提示までが長い。もっと高速化を
- ・勝率が低いが高順位戦略があり、判定基準が謎
- ・予備大会中も最良戦略同士の対戦結果が見たい

図 16 大会運営に関する意見

- ・相手と勝敗を競うゲームだったので、意欲がわいた
- ・勝率が良くななくても、上位戦略に勝つ逆転現象が、おもしろかった。
- ・やりこみ度が高く、指標戦略を倒すべき目標と考えられる
- ・対戦回数が増えると攻守の方法を変えるのが面白い

- ・五五というゲームに馴染むのに時間が掛かる
- ・相手がいないと結果がわからず不便だ
- ・ポーカーのように明確な戦略があまりなく、戦略のイメージをつかむのに時間がかかった
- ・友人と相談しながら実装できず、寂しい

図 17 課題題材に関する意見

- ・着手確認モジュールを利用することで、実行テストを効率的に行うことができた
- ・サンプル戦略との対戦から定石を参考にできる
- ・設計、実装、機能追加のスパイラルモデルを体験できた
- ・関数に分けることで再利用しやすく実装した

- ・インスタンスの使い方がわからない
- ・勝つためのアルゴリズムの発想が難しい。なにが解説を

図 18 戦略実装に関する意見

- ・順位付によって、意欲がわいた
- ・対戦結果によって成績が決まるのは、努力が認められるような感覚だ
- ・戦略の実装を進めるうちに、勝率が増え、やりがいが出た

図 19 授業形態に関する意見

- WinG-LA
 - ・着手確認モジュールは戦略が盤面を認識しているかの確認に有用だった
 - ・作成支援ツールが充実して、躓くことが少なかった

- WinG-CS
 - ・間引対戦による効率化で、やる気があまり低下しなかった

- WinG-LA
 - ・モジュール間の行来でエラーメッセージが出ている
 - ・システムの起動方法が煩雑でわかりづらい
 - ・API仕様書をもっと充実して欲しい
 - ・着手確認で、戦略からの標準出力を表示して欲しい

- WinG-CS
 - ・大会期間後半は、提出戦略の結果を見るのに時間がかかる
 - ・重付勝点度が不明瞭だった
 - ・対戦結果がエラーだったとき、原因が表示されない
 - ・対戦結果をダウンロードし、手元でも再現したい

図 20 システム機能に関する意見

7. システムへの追加機能の検討

レポートとアンケートから、WinGの新機能を検討する。まず、WinG-LAについては、戦略作成で利用するメソッドなどをまとめた、API仕様書を整備する。戦略からの標準出力を表示する機能を追加する。また、戦略のつけた評価値を盤面に表示し、デバッグを容易に行えるようにする。

WinG-CSについて考える。現在は、全てのデータベースをXML形式にて管理している。そのため、提出戦略数が増えるにつれ、対戦履歴の閲覧や対戦の再現を行う際、読み込み時間がかかってしまう(図21)。受講生からも、不満の声が多く上がっている。そこで、データベースをXML形式から、関係データベースSQLで管理する部分と、NoSQLを用いて管理する部分に分割することを検討している。ユーザ情報や、戦略の勝敗などについては前者で、局面や戦譜については後者で扱う。

現在、対戦再現ページなどのアプレットには、署名が付けられておらず、セキュリティの観点から実行がブロックされてしまう(図22)。現在は、暫定的な処置として、学生が運営サーバのページのURLをホワイトリストに登録している。今後、アプレットに署名を入れるなど、現在のJavaのバージョンに合わせた修正が必要となる。

また、WinG-CSとWinG-LAの連携を強化し、WinG-LA上で、戦略実装や提出、予備大会の結果表示や対局再現などを行うよう整備することが考えられる。

8. おわりに

問題解決型の応用プログラミングとして、ボードゲーム戦略を題材とする対戦形式でのJava演習を提案し、実際の授業で実践している。支援システムWinGを開発し、提出された戦略同士を対戦させる予備大会を運営して、順位や戦績を公開する。試行錯誤的なプログラミングを体験させ、持続的な戦略修正への動機付けを行う。2011年度からの実践の総括として、最終大会の結果について、戦績や達成状況など、各側面から分析した。また、学生の演習レポートや自由記述のアンケートによる意見も整理した。

今後の課題として、2014年度の演習実践の詳細な分析を行う。提出された戦略に対し、コードメトリクスを用いた品質の評価を行う[8]。各指標とWWGとの相関性を調べる。それらを基に、学生へのフィードバックを検討する[9]。

また、学生の意見を参考にしながら、システムへの追加機能や大会運営の改善を検討する。WinG-LAでは、ドキュメントを整理する。デバッグを支援するツールを実装する。WinG-CSでは、データベースをXML形式からSQLとNoSQLに切り替え、高速化を図る。Javaのバージョンアップによるセキュリティ的な課題を解決する。

参考文献

- (1) 尾崎浩和, 富永浩之, 林敏浩, 山崎敏範: ボードゲームの戦略プログラミングを題材としたJava演習の支援システムの開発, 情処研報, Vol.2006, No.108, pp.1-8 (2006)
- (2) 尾崎宏和, 富永浩之, 林敏浩, 垂水浩幸: ボードゲーム戦略を題材とする問題解決型プログラミング演習支援—試行錯誤的な戦略作成の支援環境とサンプル提示—, JSiSE 研究報告, Vol.22, No.4, pp.69-74 (2007)
- (3) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材としたJava演習支援—ローカル支援ツールの統合とサンプル素材の利用法—, 情処研報, Vol.2014-CE-118, No.19, pp.1-6 (2013)
- (4) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材としたJava演習支援—指標戦略の導入と重み付き勝点度による結果分析—, JSiSE 研究報告, Vol.28, No.2, pp.127-134 (2013)
- (5) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材としたJava演習支援—間引き対戦の導入と提出戦略の詳細分析—, 情処研報, Vol.2013-CE-124, No.10, pp.1-6, (2014)
- (6) 花川直己, 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材としたJavaプログラミング演習における学生の取り組み状況の分析, JSiSE 学生研究発表会 四国会場, No.3, pp.1-2 (2014)
- (7) 劉世博, 花川直己, 山田航平, 富永浩之: ボードゲーム戦略としてのプログラミング演習支援—最終大会の戦略と特徴分析とアンケート評価—, JSiSE 研究報告, Vol.29, No.3, pp.15-20 (2014)
- (8) 花川直己, 山田航平, 富永浩之: ボードゲーム戦略を題材としたプログラミング演習支援—最終大会の提出コードの特徴分析—, 信学技報, Vol.114, No.121, pp.13-16 (2014)
- (9) 玄馬史也, 富永浩之: ポーカー戦略を題材とする応用Cプログラミング演習の支援と実践—大会運営サーバWinTの提出状況とコード比較の機能の追加—, 情処研報, 2015-CE-128,1-6 (2015)

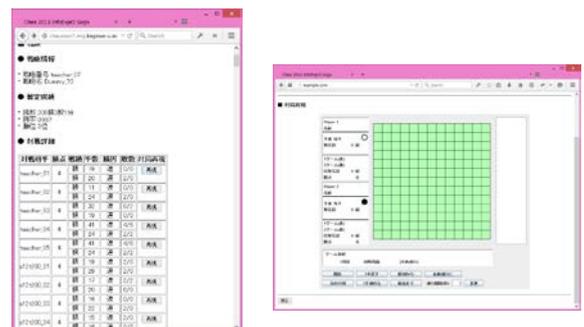


図21 対戦履歴ページと対戦再現ページ



図22 アプレットの警告画面