3ZF-08

ボードゲーム戦略を題材とした Java 演習における提出コードのメトリクス分析

花川直己† 山田航平† 富永浩之†
香川大学† 香川大学† 香川大学†

1. はじめに

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 演習を提案している[1]．ゲームとして，五目並べに石取りを加えた五五を採用する．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 つの勝利条件があり，それぞれに攻撃と防御の優先度が考えられ，初心者でも戦略の個性が出やすい．

戦略の作成手順は，戦略方針に従って，各枡の評価値を求め，最高点の位置を着手とする(図 1)．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，プロトタイプの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する．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．

2. 予備大会と最終大会

作成された戦略同士を対戦させる大会を運営するために，大会運営サーバ WinG-CS を開発している．WinG-CS は，提出された戦略プログラム同士を対戦させ，戦績や順位を公開する．演習期間中は，予備大会として，対戦結果を参考に，何度でも戦略を提出できる．締切時に各自の最強戦略で総当りの最終大会を実施する．これにより，試行錯誤的なプログラミングを通して，持続的な戦略修正を動機付ける．

2011 年度から，強さの指針として，3 段階の指標戦略を導入し，具体的な目標を提示した．2012 年度からは，レーティングとして，重付勝点度(WWG)を採用し，対戦相手の強弱を戦績に反映させた[2].

3. 戦略プログラムの内部評価

本研究で提案している演習では，戦略同士の対戦による戦績をプログラムの評価として用いている．しかし，勝利数が多く，上位に位置する戦略が，必ずしも質の良いソースコードであるとは言えない面がある．そのため，実行結果という外面的な評価だけでなく，モジュール設計やソースコードの書法など，内面的な評価も考慮しなければならない．現在，このような評価については，最終大会が終了した後，設計，ソースコード解説，実験状況などをまとめた課題レポートを提出させ，その評価を加味することで補っている．しかし，現状の方法では，事後の分析となるため，大会得点のように，即時のフィードバックを行うことが困難である．

そこで，質の良いコードを定量的かつ，自動的に評価する手法を導入する必要がある[3]．その手法として，ソフトウェアメトリクス(計量)に着目する．特に，コードの量，冗長性，構造化，複雑度などを計測する．ただし，戦略プログラムは，if-then 形式のプロダクションルールの実現に近く，一般的なプログラムとは異なった特徴を有している．また，オブジェクト指向の特性や実行ライブラリの影響もある．

本論では，2013 年度の最終大会の戦略コードを成績群に分けて分析する．成績群は，WWG による順位によって，大きく 4 群に分け，上位から第 1〜4 群とした．対象とする戦略コードは，事前処理として，コメントや空行を削除し，実質的なコードに対して計測を行う[4].

4. 戦略コードの分析と考察

コードの量については，戦略コードの行数を用いる(図 2)．下位陣は，200 行前後と行数が少なく，明らかな努力不足といえる．一方，上位陣は，200 行から 500 行の行数を記述している．中には，1000 行を超えているものもあった．しかし，下位陣に存在しながらも，400 行を超えるコードも存在している．これは，努力はしているものの，効果が薄いか，冗長な記述をしているのではないかと考えられる．

冗長性については，ZIP によるファイル圧縮率を用いる(図 3)．ZIP の中身は，元の戦略コードと提示したサンプルコードを連結したものである．冗長なコードには，類似の処理が多いため，圧縮率が高い．提出コードの多くが 15〜21%のサイズ比に収まるが，ややサイズ比が小さいコードも存在する．上位陣と下位陣のサイズ比が小さい．要因として，コードの行数が影響して

Code Metrics Analysis in the Final Contest of Java Programming Exercise with Board-Game Strategy
†Naoki HANAKAWA, Kagawa University
†Kohei YAMADA, Kagawa University
†Hiroyuki TOMINAGA, Kagawa University

いると考えられる．

構造化については，メソッド定義の個数を用いる(図 4)．ここで指すメソッドとは，学生が独自に定義したものである．本演習では，実行ライブラリを提供しているため，メソッド定義のピークは 5〜10 個と少なめである．15 個以上定義されているコードも少なくない．成績群による，顕著な傾向の違いは見られない．上位陣は，個数が少ないものも存在する．学生は，まず，機能を追加し，その後，リファクタリングを行うというサイクルで実装していると推測できる．WWG が上がるにつれ，そのサイクルが顕著だと考えられる．

複雑度については，循環的複雑度を用いる(図 5)．循環的複雑度とは，モジュール構造度上の複雑さの測定値であり，分岐構文や反復構文の数に比例する．複雑度が高ければ，修正や拡張が困難になる．上位戦略は，複雑度が低く，保守性の高いコードが多い．最下位の戦略は，上位戦略とほぼ同じ程度の複雑度である．これは，未熟な戦略で条件式がまだまだ少ないのが原因である．

5. おわりに

ボードゲーム戦略を題材とする Java プログラミング演習において，最終大会で提出された戦略コードの内部評価を検討した． 2013 年度の成績群ごとの特徴を，幾つかのソフトウェアの計量を用いて分析した．今後の課題として，WWG の点差による得点群に分割し，特徴を分析する．複数のソフトウェアメトリクスによる組合せによる分析を行う．これまでの演習実践で提出された戦略に対して分析を行う．提出時に戦略の傾向をフィードバックする．有効な指標を制定し，戦略コードの診断に用いる．

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) 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 ー指標戦略の導入と重み付き勝点度による結果分析ー, 教育システム情報学会 研究報告, Vol.28, No.2, pp.127-134 (2013)

3) 玄馬史也, 吉田亜未, 大川昌寛, 山田航平, 富永浩之: カードゲーム戦略を題材としたプログラミング演習支援 ー最終大会の提出コードの特徴分析ー, 信学技報, Vol.114, No.121, pp.17-22 (2014)

4) 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 ー最終大会の提出コードの特徴分析ー, 信学技報, Vol.114, No.121, pp.13-16 (2014)

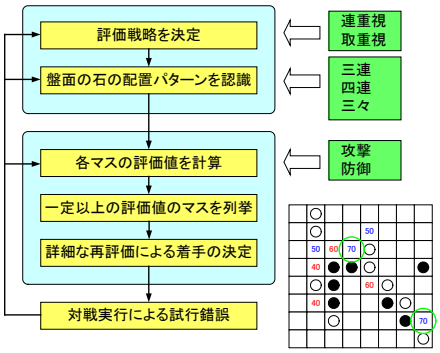

図 1 戦略プログラムの組立て方

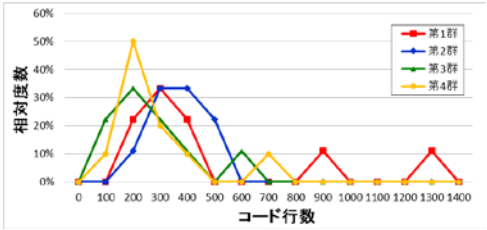

図 2 コードの行数

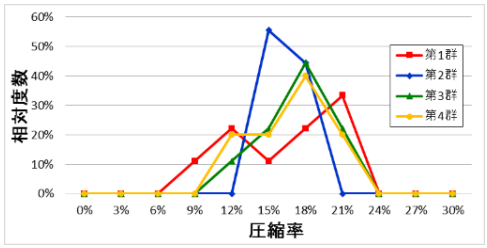

図 3 ファイルの圧縮率

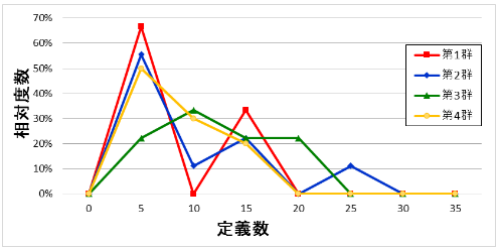

図 4 メソッド定義の個数

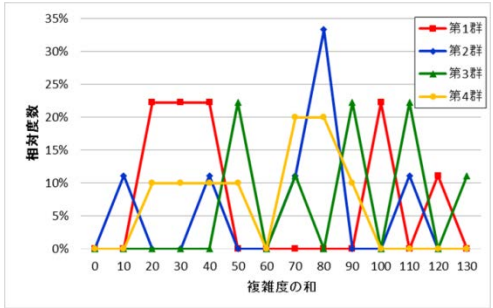

図 5 循環的複雑度の和