

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

尾崎 浩和, 富永 浩之, 林 敏浩, 山崎 敏範

香川大学工学部 〒761-0396 香川県高松市林町 2217-20

E-mail: s06g457@stmail.eng.kagawa-u.ac.jp

あらまし 大学での知識情報処理の応用課題として、ボードゲーム戦略の Java プログラミング演習を提案する。題材として、五目並べに石取りの要素を加えた五五ゲームを採用する。コンテストでの対戦を目標に、局面のヒューリスティックな評価に基づいて、試行錯誤的なプログラミングを体験する。効果的な演習に向けて、戦略作成のためのローカルな実行環境と、公開対戦による大会運営サーバを開発する。昨年度には、実行環境の暫定版を学生に配布し、中間大会と最終大会を実施した。その結果を基に、サンプル局面を提示し、着手メソッドのデバッグを支援するように実行環境を改良する。また、予備戦期間を設け、順位を表示して、持続的な戦略修正を促進させる。

キーワード 戦略プログラミング, Java 演習, 五五ボードゲーム大会, 予備戦

Support System for Java Exercise with Board-Game Strategy Programming

Hirokazu OZAKI, Hiroyuki TOMINAGA, Toshihiro HAYASHI, Toshinori YAMASAKI

Faculty of Engineering, Kagawa University 2217-20 Hayashi, Takamatsu, Kagawa, 761-0396 Japan

E-mail: s06g457@stmail.eng.kagawa-u.ac.jp

Abstract As an educational practice to knowledge information processing in college, we propose a Java exercise about strategy programming of board game. We adopt Gogo, which is similar to Gomoku and has the other rule with stone removing. We develop a support system Gogo for the exercise. Through the try-and-error exercise, a student must consider pattern analysis of board situation and heuristic searching. And he must learn design and implementation of Java programming. We carried out practical experiment in last year. According to result, we revise local execution environment to support debugging with sample board situations. And we arrange preliminary league to keep motivation for programming. In the duration of preliminary league, a student uploads his strategy program and meets other strategy in the Web server. The server exhibits the score and the game log to promote his emulous exertion. To analyze the feature of the logs, he revises the strategy in several times, until the regular league for final score.

Keyword strategy programming, Java exercise, Gogo board-game contest, preliminary league

1. はじめに

知識情報処理の学習では、プログラミング演習を通して、探索手法などの具体的な応用を理解することが求められる。学習内容は、ヒュー

リスティックな評価に基づいて、試行錯誤的にアルゴリズムを改良することである。近年、このような題材として、ゲーム戦略を用いる事例が増えている[1][2]。例えば、広島市立大学や

東京工科大学では、オセロが用いられている [3]。特に、戦略プログラム同士を対戦させ、大会形式で評価を行う演習が目ざされている [4]。より教育効果を高めるには、多くの対戦過程を学生に提示し、対戦結果から戦略を修正させるなどの工夫が必要となる。本研究では、ボードゲームを題材とした、対戦形式の Java プログラミング演習を提案する。

プログラミング演習の題材にボードゲーム戦略を用いる利点として、表 1 の事項が挙げられる。香川大学でも、3 年次の演習に、複数のボードゲーム題材を取り入れている [5]。ボードゲームは、与えられた盤面の上で互いに自分の駒を置いたり動かしたりし、決められた勝利条件を満たすことを目的とする。互いの手は公開されており、偶然要素がなく、戦略の優劣だけで勝敗が決まる。

表 1 ボードゲーム戦略による演習の利点

● 身近な題材による関心と興味
・ 学生にとって取っつきやすい題材である
● 試行錯誤的な問題解決の訓練
・ 正解が 1 つに定まらないオープンプロブレムである
・ 試行錯誤を通して各自の自由な発想を引き出せる
● 明確な目的と競争による意欲向上
・ 勝つという目的が明確である
・ 対戦結果による得点比較が客観的な評価基準となる
・ 学生の競争意識を刺激し、演習意欲を向上させることが期待できる
● 安易な不正の防止
・ 他人にプログラムを写させると自分の戦績が不利になる
・ 安易なコピーを防げる

2. ゲーム題材としての五五

2.1. ボードゲーム戦略を題材とする演習の目的

ボードゲーム戦略によるプログラミング演習の教育目標は、知識情報処理の基本手法の設計技能を習得することである。例えば、以下のような項目がある。石の置き方などの適用条件を定式化し、ルール通りに着手させる。勝敗に関わる特徴的な局面のパターンを抽出する。ヒューリスティックな知識に基づいて分析する。

着手の優劣を評価し、次の一手を決める。また、これらを Java 言語で実装することで、オブジェクト指向によるデータ操作やメソッド定義を習得する。特に、ボードゲーム戦略では、アルゴリズムの優劣が勝敗を大きく決めるため、プログラミング演習の題材として適している。

2.2. 五五の概要とルール

ボードゲームには、オセロや将棋など、知名度の高いものがある。ただし、既存のゲームを題材にすると、学生の予備知識の有無で差が生じてしまう。そこで、新規な、あるいはあまり知られていないゲームを取り入れる [6]。そこで、本論では、五目並べに石取りを加えた、二抜き連珠というボードゲームに着目する [7][8][9]。二抜き連珠は、昭和初期に久保山機山(連珠九段・囲碁八段)が家元を名乗り、研究を進めていたが、現在では正確なルールは不明である [10]。また、似たゲームとして、以前に米国で商品化された Pente がある。Pente には、ドイツやポーランドに少数の愛好家がいるようである [11]。本研究では、表 2 のように勝利条件や禁じ手などのルールを整備し、五五と名付ける。禁じ手については、幾つかのルールの変種が考えられるが、ここでは戦略プログラミングに適した簡単なものを採用する。ただし、今後の戦略の研究によって、先手の有利性が確認されれば、変更する可能性がある。

2.3. 五五のボードゲームとしての特徴

五五のゲームの特徴としては、次のことが挙げられる。連と取という 2 つの勝利条件があり、それぞれについて攻撃と防御の優先度が考えられる。戦略のバリエーションが広がるため、戦略の個性が出やすい。これらの特徴から、五五は設計技能の学習に必要な要素を備えている。また、よく知られた五目並べの変種なので、初めてでも取っつきやすい。石を取ることで局面が大きく変化するため、奥が深い。資料が少

なく、必勝法が分かっていないため、自分で試行錯誤しながら解く必要がある。これらの特徴から、演習題材として魅力的であると考える。

2.4. 戦略プログラミングの問題設定

学生に提示する五五の Java 実行環境は、図 2 のような構成とする。コンピュータとユーザのどちらでも対戦できるようにする。将来的に、汎用的なボードゲームへの拡張とネットワークへの対応を念頭に置いている。Master クラスは、Player クラスから着手を受け取り、盤面に反映させたり、ゲームを進行させたりする。ゲームの勝敗判定は、表 3 のような手順で行う。学生の演習内容は、Computer クラスを継承したサブクラスで、着手決定のメソッドをオーバーライドすることである。着手メソッドは、局面を引数とし、次の 1 手を返す。局面には、盤面、手数、手番、石数などの情報が含まれる。着手メソッドに渡される局面は、実行環境が持つ情報のコピーとし、データの改竄によってゲームの進行に影響が出ないようにする。このような設計にすることで、局面や着手を抽象データとしてオブジェクト化すること、メソッド単位のモジュール化、オーバーライドによるメソッドの再利用と拡張といった、実装技能を学習させる。

2.5. 戦略の組立てと評価関数

学生が実際に戦略を検討する上では、図 3 のようなサイクルで考えさせる。連重視と取重視で戦略の方向性を決める。局面のパターンを認識する。攻撃と防御の観点から評価値を計算する。評価値の高いマスを一覧し、詳細に再評価を行い、着手を決定する。実際に対戦を行い、試行錯誤により、評価関数を修正する。評価値を使った戦略の例を、サンプルとして学生に配布する。少なくとも、サンプル戦略に勝つことを目標とさせる。サンプル戦略では、例えば表 4 のような基準で、各マスに評価値を付ける。い

ずれの状況にも当てはまらないマスは、ランダムに評価値を設定する。

表 2 五五のルール

<ul style="list-style-type: none"> ● 基本ルール <ul style="list-style-type: none"> ・ 先手が黒石、後手が白石を使用し、交互に石を打つ ・ 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
<ul style="list-style-type: none"> ● 勝利条件 <ul style="list-style-type: none"> ・ 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
<ul style="list-style-type: none"> ● 禁じ手 <ul style="list-style-type: none"> ・ 「三々」は、先手後手共に禁じ手である ・ 石を取った後の「三々」は、禁じ手とならない

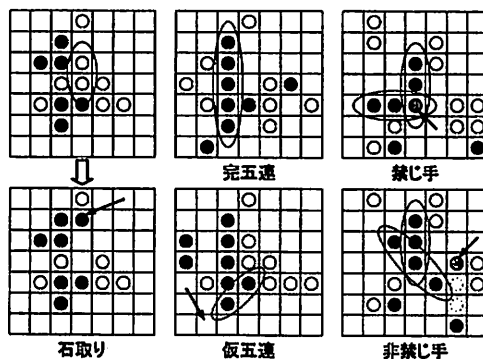


図 1 五五のルールと局面

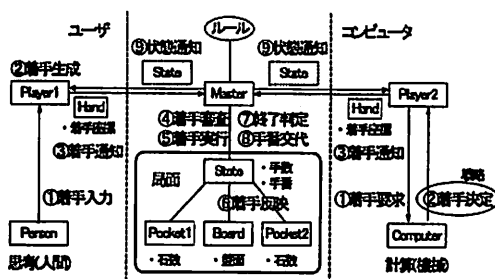


図 2 ゲーム戦略プログラミングの実行環境

表 3 勝敗判定の手順

<ol style="list-style-type: none"> 1. 置けない打手(盤外, 重打, 連打)は反則負け 2. 打って(石を取る前に)三々になったら、禁じ手で負け 3. 石を取っても、相手の全ての五連を崩せなければ負け 4. 相手の長連から石を取り、相手の五連を作ったら負け 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 7. 全ての枡が埋まっていたら引分 8. 時間内に打たないときは投了で負け
--

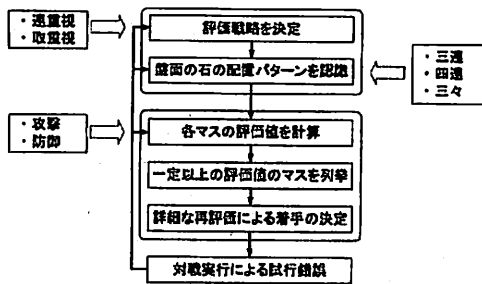


図3 戦略プログラムの組立て方

表4 サンプル戦略の評価値

相手の五連を崩す	100	相手の三連を阻止	50
自分の五取を実現	95	自分の三連を実現	40
自分の五連を実現	90	自分の石取を実現	30
相手の五取を阻止	85	相手の石取を阻止	20
相手の五連を阻止	80	禁止手	-1
相手の四連を阻止	70	その他	乱数
自分の四連を実現	60		

3. 戦略プログラミング演習の実施例

3.1. 演習の実施形態

五五によるゲーム大会のルールを以下のよう設定する。盤面は13×13とする。総当たり対戦を行い、勝ち点の合計で順位を決める。先手後手の1組で1ゲームとし、表5のように、勝ち点を付ける。1勝1敗では取った石の数で勝敗を決め、同数ならば引分とする。このため、五連を作るより、石を取って勝つ方がやや有利である。戦略に乱数が用いられる場合、対戦するたびに結果が変わることが考えられるが、対戦は1回限りとする。実行中にエラーが出たり、時間切れになると、そのゲームは負けとする。

演習は、表6のように5週間ほどの期間で行う。中間大会は、最終大会の前に、最終大会と同じ形式で総当たり対戦を行う大会である。結果は、公開されるが、成績には考慮されない。人数は、20人程度で1リーグとし、受講者が多ければ、複数のリーグに分ける。

3.2. 戦略プログラミング演習の概要

以上のような実施形態で、2005年度の香川大学工学部信頼性情報システム工学科3年次

対象の「システムソフトウェア演習2」において、本演習を実施した。受講者は47人で、これを2つのリーグに分けた。中間大会の後、評価関数を使った戦略のサンプルを提示した。

最終大会の結果は、どちらのリーグも、上位は70～80点台、中位は40～50点台、下位は20～30点台が多い。最上位付近の戦略は、ほとんど全ての相手に完勝していた。逆に、ほとんど全ての相手に完敗し、得点が1桁台の者もいた。全体的に、完勝/完敗が多く、1勝1敗となった対戦はやや少ないようである。

3.3. 戦略プログラミング演習の分析と評価

対戦結果の分析結果について述べる。下位が中位に、中位が上位に勝つことは稀にあったが、上位が下位に負けていることは無かった。上位戦略の戦譜を見ると、三々四をほぼ確実に止めており、禁止手を打つこともほとんどなかった。また、連と認識されにくいように、1つ置きに打つ戦略が多かった。下位戦略は、連などを正しく判定できていない、ランダムに打つ、というものが多かった。なお、先読みをするに至った学生は2名程度であった。中間大会の段階では、大半の学生はランダムに打っていた。評価関数を使ったサンプル戦略を提示したことで、どのように戦略を組み立てればよいかを理解できた学生が多かったようである。また、学生の感想として、表7のような意見が挙げられた。学生の願きについては、戦略のアイデアを見つけること、アイデアをアルゴリズムで表すこと、アルゴリズムを正しく実装すること、実装したプログラムの仕様を検証することなどの箇所に整理できる。それぞれに合わせた支援を検討する必要がある。

表5 勝ち点の付け方

2勝	完勝	+4
1勝1敗	判定勝ち(取った石が多い)	+3
1勝1敗	引分(取った石が同じ)	+2
1勝1敗	判定負け(取った石が少ない)	+1
2敗	完敗	0

表6 演習スケジュール

1週目	ルール説明, 実行環境配布	実行環境で試遊
2週目	実行環境説明 戦略説明	中間大会用に 戦略を作成
3週目	中間大会実施 サンプル戦略の説明と配布	最終大会用に 戦略を修正
4週目	最終大会実施	レポート作成
5週目	レポート提出	

表7 学生の感想

<p>● ゲーム題材について</p> <ul style="list-style-type: none"> ・ ルールを理解するのは簡単だった ・ ゲームとしての戦略の幅の広がりを感じた ・ 様々な攻撃パターンを、もう少し研究すればよかった ・ 対戦相手との駆け引きがあり、難しかった ・ 進と取のパターンが状況によって変わるので、難しい
<p>● プログラミングについて</p> <ul style="list-style-type: none"> ・ 戦略を考える力と、それを構築する力を養えた ・ 自分で評価値を加えていく作業が楽しかった ・ 人間がやることをプログラムで表現するのは難しい
<p>● 実行環境について</p> <ul style="list-style-type: none"> ・ デバッグ環境がなく、プログラムの評価が難しかった ・ 人間からランダム着手のプログラムだけでは改良が難しい ・ もう少し別のプログラムを参考にしたかった

4. サンプルの提示と実行環境の改良

4.1. 改良の基本方針

以上の結果や学生の意見から、次年度に向けて改良を検討する。基本方針としては、ゲームのルールを変更し、バリエーションを増やす。戦略改良の支援として、サンプル戦略に加えて様々なサンプルを提示する。また、デバッグをしやすくするために実行環境を改良する。

ゲームのルールを変更し、与えられた初期局面からスタートするようにする。初期局面とは、ゲーム開始時の石の配置や取石数の状態のことである。これまでは、盤面上に石が無く、取石数も0の状態が初期局面であった。初期局面としては、通常の対戦では起こり得ない状態を考えてもよい。様々な初期局面をサンプル局面として学生に提示する。

戦略アイデアが浮かばない学生に対しては、次の一手を考える練習問題を、サンプル局面として提示する。様々な局面での次の一手を検証させ、戦略を考えさせる。実装の仕方が分からない学生に対しては、戦略プログラムの基本パ

ターンを、サンプル戦略として提示する。実装はしたが、それが仕様通りに動かない学生に対しては、サンプル局面としてテスト用の局面を提示する。試作した戦略のテストケースとして、適切な局面を選ばせる。実装はできるが、効果的な戦略が思い浮かばない学生に対しては、強い戦略との対戦をサンプル戦譜として提示し、参考にさせる。他人の対戦過程を分析することで、アルゴリズムを推測させる。

4.2. サンプル局面の意義

サンプル局面の導入の意義としては、ゲームの変種、戦略のデバッグ、練習問題がある。初期配置を年度やクラスごとに変えることで、戦略のコピーを防ぐ。また、先手と後手の有利さに差を付けることで、戦略を使い分けさせる。

盤面の評価関数の仕様通りに着手が行われているかの確認(テスト/デバッグ)に、初期局面を用いさせる。必勝局面における最適な攻撃の手(五連、五組など)や必敗局面における最適な防御の手(五連崩し、五組阻止など)を打っているかの確認、連重視と取重視、攻撃と防御の優先度を適切に評価するための支援として用いる。サンプル局面だけでなく、学生の任意の局面でテストできるよう、初期局面を生成するような機能を提供する。

戦略研究の支援として、詰将棋に当たる練習問題として学生に与え、定石らしきものを見つけさせる。また、戦績が良くない学生に対して、これを解くことで得点を与える、最低限の課題としても用いる。

4.3. 実行環境の改良

以上のような支援に対応させるため、実行環境の改良を行う。実行環境の機能を分析し、モジュールを整理した。人間またはプログラムでゲームの対戦を行う対戦実行モジュール、対戦で使用する初期局面を生成する局面生成モジュール、対戦結果を保存した戦譜から対戦を再

現する戦譜再現モジュールの3つとする(図4)。局面生成モジュールで作成、もしくはサンプルとして提示された初期局面を選択し、対戦実行モジュールで試作した戦略を実行する。対戦過程は戦譜として保存され、戦譜再現モジュールで閲覧することができる。再現途中の局面を、初期局面として生成できるようにする。また、大量の初期局面に対し、次の一手のみを実行する、着手確認モジュールの導入も検討している。

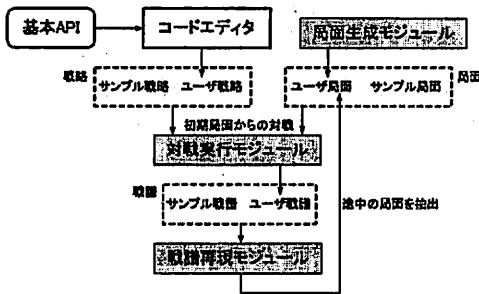


図4 ゲーム実行環境とサンプル

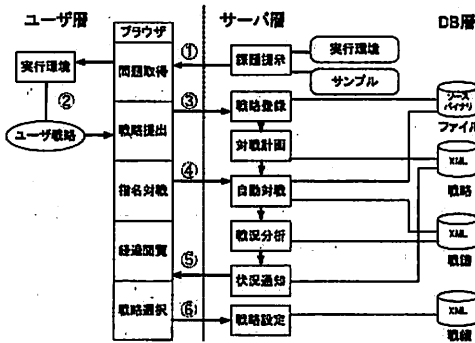


図5 Gogoのシステム構成

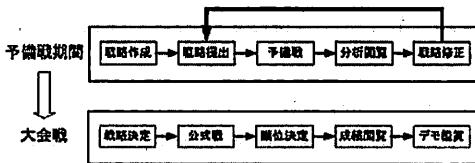


図6 予備戦と大会戦の流れ

5. 大会運営支援の検討

5.1. 大会運営システム Gogo

実行環境の改良とサンプルの提示だけでな

く、大会自体の運営の支援についても検討する。2005年度の演習では、中間大会や最終大会の直前になるまで、プログラミングに取り組まない傾向が一部に見られた。そこで、1回限りの中間大会の代わりに、その教育効果を持続的なものにするため、最終大会までの間を予備戦期間と定める。予備戦期間中は、運営サーバ上で他人の戦略と対戦できる。対戦結果は全て公開される。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。

5.2. 大会運営システムの概要

実行環境の配布から大会戦の実施までを支援する大会運営システム Gogo を開発する。Gogo の全体構成は、図5のようになる。Webサーバ側で、戦略の管理と対戦の実行、対戦結果の分析を行う。学生側には、ローカルな実行環境と各種サンプルを提供する。学生は、試作したユーザ戦略をこの実行環境で試行する。このシステムを用いた演習の流れを図6に示す。予備戦期間中に、学生は様々な戦略を試作する。試作した戦略をアップロードし、予備戦に挑む。対戦結果は、常時更新され、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。予備戦後に、提出した戦略の強さを総合的に判断し、大会戦用の戦略を選択する。選択された戦略同士で大会戦として総当たり対戦を行う。この結果から最終順位を決定し、成績に反映させる。

5.3. 予備戦の支援

予備戦の処理の流れを図7に示す。学生は予備戦期間中に、Webブラウザから戦略のソースコードを提出する。提出された戦略は、戦略登録部によって、サーバ上でコンパイルされ、バイナリとともに保存される。対戦計画部は、提出された戦略に対し、その時点での暫定順位を決めるのに必要十分な対戦相手をリストアップする。これに基づいて自動対戦が行われ、勝

敗と戦績が記録される。この結果を全体の勝敗表に組み込み、新たな順位が求まる。提出された戦略の戦績や順位は、経過表示ページで学生に通知される。

5.4. 予備戦における対戦計画

予備戦期間中の暫定順位を決めるには、提出された全ての戦略同士の総当たり戦を行うことが考えられる。しかし、20人程度のリーグで、各自が複数の戦略を提出する状況では、対戦数が多くなりすぎる。戦略を提出した際、戦績の表示までに時間がかかりすぎると、学生が意欲を失いやすい。そこで、対戦計画部が、適度な対戦数に抑えるよう、対戦相手を限定する。まず、他人の最新の戦略を対戦相手としてリストアップする。さらに、過去の戦略でも順位の高い戦略に限り、対戦相手に加える。なお、規定試合数を満たさないものは、順位表から外す。これにより、初期の対戦でブロック的に上位になった戦略は、時間の経過と共にふるい落とされていく。

5.5. 予備戦における指名対戦

これまでに提出された全ての戦略に関する情報は、図8の戦略管理ページで閲覧することができる。過去の自分の戦略、他人の戦略、勝敗や順位が表示される。他人の戦略については、対戦を再現することはできるが、ソースコードを閲覧することはできない。対戦の再現については、実行環境の戦譜再現モジュールをアプレットとして用いる(図9)。

また、図10のように、提出時に自動対戦されなかった戦略でも、順位表から指名対戦できるようにする。順位表では、最新の戦略および古くても強い戦略のみが表示される。上位にあれば、対戦の指名を受ける可能性が高くなる。真に強ければ上位が維持されるが、弱ければ順位が下がり、指名回数が少なくなるので、真に強い戦略がランキングに残るようになる。これ

らの戦略と自分の戦略との対戦を再現することにより、自分の戦略の欠点を見つけたり、戦略の流行を掴んだり、対抗策を考えて戦略を練り直す。予備戦期間を設け、その活動状況も評価することで、多くの学生を持続的に演習に取り組ませることができる。

5.6. 大会戦の支援

予備戦期間が終了すると、大会戦を行う。学生は、予備戦で提出した戦略から、締切日まで大会戦用の戦略を選択する。これにより、様々なタイプの戦略を提出するなど、予備戦期間を積極的に使った学生が有利になる。弱い戦略だけを提出し、直前に強い戦略を提出するなどの出し惜しみが減り、予備戦が活発になることが期待できる。選択された戦略同士で自動的に総当たり戦を行い、戦績と戦譜を保存する。戦績表示ページで、最終結果として勝敗表を公開する。

5.7. 最終レポートの構成

演習の評価は、戦績だけでなく、戦略の自己評価を行った最終レポートで行う。最終レポートには、まず、問題分析と基本戦略(要件定義)について書く。次に、プログラムの全体設計と詳細設計について書く。全体設計では、どのようなメソッドを定義するかを書く。詳細設計では、連の判定などで、どのようなアルゴリズムを用いるかを書く。次に、実行結果と考察について書く。アルゴリズムが正しいかなど、プログラム単体での視点と、戦略が有効であるかなど、対戦での視点の、それぞれについて書く。最後に、参考文献と、感想などの後記を書く。このようなレポートのテンプレートを予め学生に提示し、演習の最初からドキュメントによる総括を意識させる。

6. おわりに

知識情報処理の応用として、ボードゲームの

戦略プログラミングを題材とした Java 演習を提案した。五五を例題とし、二つの勝利条件の組合せによる多様な戦略を考えさせる。予備戦期間中の公開対戦を通して、競争意欲を促進し、戦略の修正を動機付ける。このような演習により、戦略のアイデアを実装すること、実装した戦略を検証すること、戦譜から他人の戦略を推定すること、相手に合わせて戦略を修正すること、などを学習する。今後の課題として、実行環境の各モジュールと大会運営システムを実装し、実際の演習に利用する。

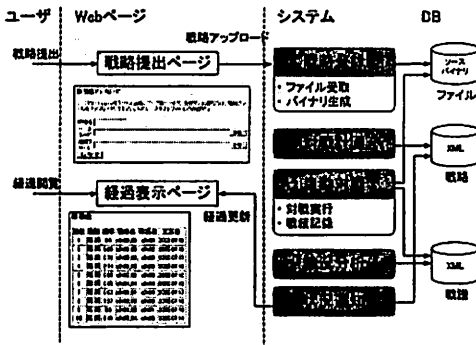


図7 予備戦の実行処理

■ 予備戦

自分の戦略をランゴボタンで選び、対戦したい相手の戦略のボタンを押す。

● 自分の戦略

戦略ファイル	更新日	更新時刻	閲覧	閲覧数	勝敗	対戦履歴
0001_01.java	2005.07.19	21:28:03	10	0	0	1回対戦済
0001_02.java	2005.07.17	15:24:44	3	1	0	ランゴ済
0001_03.java	2005.07.15	22:54:21	7	0	0	ランゴ済

● 相手の戦略

学籍番号	ハンドル	順位	勝敗	対戦履歴	更新日
0002	たけ	1	0	0	2005.07.19
0002	たけ	2	0	0	2005.07.19
0002	たけ	3	0	0	2005.07.17
0003	たけ	4	0	0	2005.07.14
0003	たけ	5	0	0	2005.07.14

図8 戦略管理ページ

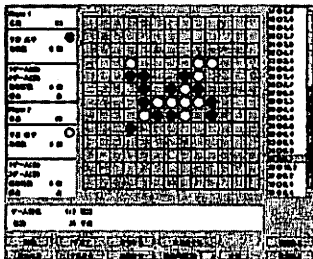


図9 デモ対戦の鑑賞アプレット

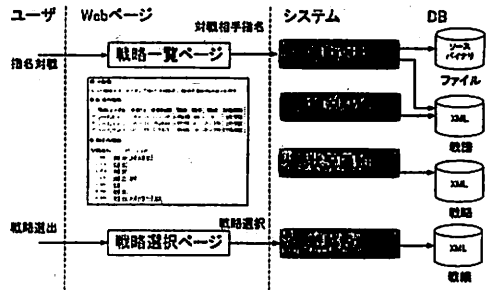


図10 指名対戦と戦略選出

謝辞 本研究は、日本学術振興会科学研究費補助金基盤研究(B)(課題番号 17300269)の助成による。

文献

- [1] 富永浩之, “プログラミング実習の題材としてのゲームと支援環境”, ゲーム学会 第1回合同研究会研究報告, pp.39-42, (2003).
- [2] 大西洋平, 富永浩之, 林敏浩, 山崎敏範, “作業過程を重視した LEGO プログラミング演習支援システム GoalPost の提案”, 信学技報, Vol.105, No.124, pp.23-28, (2005).
- [3] 広島市立大学情報科学部 シラバス, <http://www.easo.hiroshima-cu.ac.jp/OpenSllabus/>
- [4] 久保侯明, 香川考司, “ゲーム対戦による Web ベースプログラミング学習支援環境の構想”, 教育システム情報学会, 第30回全国大会, pp.279-280, (2005).
- [5] 香川大学工学部 システムソフトウェア演習II, <http://chausson.eng.kagawa-u.ac.jp/Class/SysExrc2/>
- [6] 松田道弘, “世界のゲーム事典”, 東京堂出版, (1989).
- [7] 尾崎浩和, 富永浩之, “二抜き連珠の戦略プログラミングを題材とした Java 演習”, ゲーム学会 第4回全国大会, pp.39-42, (2005).
- [8] 尾崎浩和, 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-五五ゲームの実行環境と大会運営方法-”, 信学技報, Vol.106, No.35, pp.55-60, (2006).
- [9] 尾崎浩和, 富永浩之, 林敏浩, 山崎敏範, “ボードゲーム五五の戦略を題材とした大会形式の Java プログラミング演習”, 教育システム情報学会 第31回全国大会, pp.531-532, (2006).
- [10] わくわく連珠ワールド, <http://i-alpha.com/renju/>
- [11] Pente Net, <http://www.pente.net/>