



## 君の実力を試してみよう!

—「ラーニング・スルー・コンテスト」のすすめ—

基  
般



鷲崎弘宜（早稲田大学／国立情報学研究所／（株）システム情報）

### 「コンテストは上級者向け」は誤り!

これまでの連載記事を通じてプログラミングの基本を学んだ皆さんが次にすべきこと、それはプログラミング・コンテストへの挑戦である。コンテストは、上級者が技を競うだけの場ではない。初学者が実力を客観的に把握しながら、楽しみつつ中級者、上級者へと自ら駆け上がっていくための最適なプラットフォームである。

本稿では、コンテストがプログラミング学習に優れていることを説明した上で、国内外で高校生が参加できるコンテストやその出題内容、若年層の取り組みの様子を紹介する。具体的には、筆者が創設した本会 SamurAI Coding や、情報オリンピックなどを紹介する。ぜひ自身の挑戦、あるいは指導者の方は指導対象者への呼びかけなど検討されたい。

### なぜコンテストで学ぶのか?

コンテストへの挑戦を通じて学ぶことは「ラーニング・スルー・コンテスト (Learning through Contest/Competition)」や「コンテスト・ベース・ラーニング (Contest-based Learning)」などと呼ばれる。上級者が優勝を目指して技を究め競うことよりも、あらゆる参加者が強く動機づけられながら継続的に学びを深めることを主眼とする。本稿では特に、公開された（講義内のようなクローズドではない）コンテストの活用を紹介する。

コンテストには、段階的チャレンジと達成、短いフィードバックサイクル、試行錯誤、報酬、スコアと競争、協調といったゲーム（ゲーミフィケーション）要素<sup>1)</sup>が多く含まれる。これらによって、楽し

みながら目的を達成することにつながり、特にプログラミング学習に適している。プログラミングの特徴がゲーム要素を強く支持することを図-1に示す。

- プログラムはコンピュータに対する計算手順の指示であるから、実行してその正否を自動評価しやすく、インターネット上で自動判定するオンラインジャッジと呼ばれるサービスも多く登場している。たとえば AIZU ONLINE JUDGE (AOJ), PKU JudgeOnline (POJ), AtCoder などがある。この特徴は、実力の客観的な把握につながるとともに、フィードバックサイクルを短くし試行錯誤およびスコアリングを通じた競争を促進する。初学者向けのオンラインジャッジを用いた入門書もある<sup>2)</sup>。
- プログラミングは、創造的活動である。物理的制約を受けずに自由に新たなサービスや仕組みを創造できる。同一の問題であっても、無数の異なる解決策を創造し得る。この特徴は、試行錯誤につながる。
- デジタル化によりプログラムや入出力を容易に共有できる。これは他者との協調や競争を促進する。高校球児にとっての甲子園がそうであるように、公開コンテストが初学者から上級者

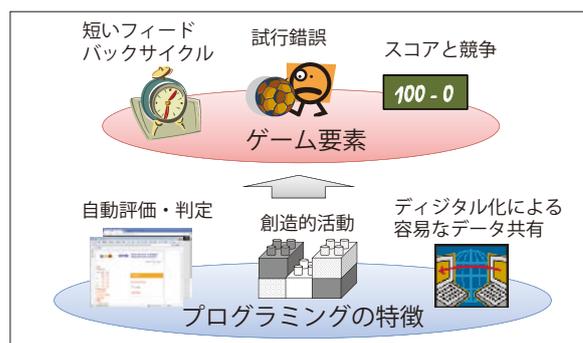


図-1 コンテストのゲーム要素とプログラミング

までの共通の目標、プラットフォーム、言語となる。

## コンテストを選ぼう!

プログラミング・コンテストは、問題解決・競技系、人工知能・ゲーム系、自由制作系の3種に大別できる。特徴比較を表-1に示す。

競技系は、いわばドリルとして、アルゴリズムとデータ構造を中心に反復的に自習することに優れている。逆に言うと地味な印象は否めない。

ゲーム系は、プログラミング学習をゲーム化する(ゲーミフィケーション)のみならず、直接的にゲームを題材とする(ゲームベース学習)。したがって多くの人にとって魅力的であるが、総合的な取り組みとなり、個々の要素やアルゴリズム単独の影響を把握しにくい可能性がある。

自由制作系は、参加者それぞれのアイデアをプログラムやシステムとして形にすることに主眼を置き、競う側面は限られている。評価においてもしばしばアイデアに重きが置かれる。

このようにそれぞれ長短あり、学びの段階や興味に応じて使い分けると良い。たとえばゲーム系への挑戦により楽しく学びを動機づけた上で、競技系により個々のアルゴリズムを深め、再びゲーム系で総合力を競いつつ確認し、最後に自由制作系でアイデアを形にするという具合である。

以降では、各種別で代表的なものを紹介する。

## サムライを操ろう!

Samurai Coding<sup>☆1</sup>は、対戦型ゲームを題材に人工知能のプログラミングスキルを個人またはチームで競う国際コンテストである。プログラミングや人工知能のスキルにより世界市場で活躍できる人材を育成することを目的とする。

筆者の研究室がグリー(株)と2011年度に実施したコンテストに端を発し、以降は本会プログラミングコンテスト委員会(現委員長は筆者)が企画開発し、IEEE Computer Societyなどから国際的な協力を得ながら毎年開催している。

インターネット予選と決勝戦の2段階方式をとり、2014年度からは決勝戦を本会全国大会の企画として位置付けて広く一般の観覧を集めている。2014年度には予選へ130超のプログラム投稿があり、決勝戦の参加者・視聴者はニコニコ生放送を含め2,000名強あった。また2015年度には、100程度の予選時投稿から、ポーランド、韓国、香港、台湾を含む18チームが決勝戦に選抜され、国際コンテストとして認知を広げつつある。

毎回、サムライをモチーフとした独自の複数名対戦型ゲームを題材としている。2012年度以降のゲーム実行環境はWeb<sup>☆1</sup>に掲載しているため、ぜひ挑戦されたい。画面例を図-2に示す。

- 2011年度：サムライと犬を操り、敵のサムライを倒しつつ、フィールド上の小判などの獲得を競う。ゲーム進行途中でフィールド上の小判

☆1 <http://samuraicoding.info/>

種別	内容	問う能力	長所	短所	例
問題解決・競技系	短い制限時間内で個々の問題を解決するプログラムの作成	問題の読解力 数学的・論理的解決力 正確さとスピード (チームワーク)	オンラインジャッジで自動判定可 個々のアルゴリズム・データ構造の反復自習 チームで役割分担可	万人に魅力的とはいえない アルゴリズムの特訓になりがち	情報オリンピック, ACM-ICPC, TopCoder, Codeforces
人工知能・ゲーム系	比較的時間をかけて(独自・対戦)ゲームを攻略する人工知能プログラムの作成	ゲームルールの読解力 総合的な解決力 人工知能アプローチ 一定期間内の設計と改善力	題材が魅力的で挑戦を動機づけやすい アルゴリズム+創造的	要素の影響を把握しにくい 対戦相手次第 題材次第では運の要素あり	Samurai Coding, CODE VS
自由制作系	比較的時間をかけて自由あるいは特定テーマのプログラム作成	アイデア・創造力 総合的な解決力 チームワーク	題材の自由度が高く革新を追求可 学際的な取り組み可 チームで役割分担可	審査員による主観的評価 客観的には能力を把握困難	U-22 プログラミング・コンテスト, Imagine Cup, 各種ハッカソン

表-1 プログラミング・コンテストの種別ごとの特徴



図-2 Samurai Coding のゲーム画面（左上から時計回りに 2011, 2012/2013, 2014, 2015 年度）

などを復活させるという「仕切り直し」といったさまざまなゲームデザイン上の工夫を施して、参加者および観戦者の興味を引き付けるようにしている<sup>3)</sup>。運営側の工夫として、プログラムを対戦させる共通の仕組みを実装した再利用可能なフレームワーク<sup>4)</sup>を用意し、その上に効率よく構築している。

- 2012/2013 年度：独自のスクリプト言語 Gunbai Script を用いてサムライを移動させ囲い込んで獲得した領地の広さを競う<sup>5)</sup>。基本的なルールや言語を 2 年間繰り返し用いた結果、参加者におけるゲームの攻略および環境構築が進展した。たとえば 2013 年度の優勝チームは、独自設計の言語および Gunbai Script への生成系を用意していた。
- 2014 年度：交渉を通じて多くの領主との親密度を上げて味方につけることを競う。優勝チームを含む決勝戦出場チームの多くは、ボードゲームの人工知能にしばしば用いられるモンテカルロ木探索 (Monte-Carlo Tree Search) を応用していた。これは、乱数によりランダムにゲームを進めた場合の勝敗回数等を記録して勝てそうな手を探索するにあたり、良さそうな手をより深く探索するよう工夫した考え方である。
- 2015 年度：異なる武器を使う 3 人のサムライが軍団を組み、相手軍団と獲得領地の広さを競う。ただし、1 人の参加者 (参加チーム) は 1 人のサ

ムライのみをプログラムで制御する。したがって、前年度までは参加チームから見てほかはすべて敵であったのに対し、味方と協力してゲームを進める点が特徴的である。サムライをランダムに動作させる Java 言語で記述したサンプルを図-3 に示す。実際にはランダムな行動選択では勝利は難しく、適切な状況判断が必要となる。具体的には、次に取り得るすべての行動について、敵からの位置や獲得済み領地面積といったさまざまな評価項目を合算して評価値を計算する評価関数を用意し、最も評価値の高い行動を選択する考え方が決勝戦出場チームに多く見られた。さらにその評価関数を、数多くの予備的な対戦結果から学習して改善する取り組みも見られた。

ゲームのシステム面では、2012/2013 年度以外はすべて、コンテスト委員会が多プログラミング言語対応のゲームプラットフォームを提供し、C, Java を含む主要言語を利用可能とした。

これまでの開催において、情報関係企業に勤めるプログラマやエンジニアの優勝が多いものの、高校生や大学生の健闘も光っている。たとえば 2014 年度において、プログラミングを始めて間もない高校生が予選を勝ち抜き表彰されている。

本会プログラミングコンテスト委員会では現在、2016 年度についても 2015 年度のゲームをおおむね踏襲した題材を検討している。まずは、図-3 に示すようなサンプルや簡単なコードにより、サムライを操ることを楽しんでほしい。慣れてきたら、上述のような評価関数を検討するとよい。

```
while (power >= 2){
    action = this.rnd.nextInt(10) + 1;
    if (cost[action] <= power
        && info.isValid(action)){
        power -= cost[action];
        info.doAction(action);
    }
}
```

図-3 ランダム動作の Java コード例<sup>☆1</sup> (乱数で行動を選択し、当該状況で残りの行動力以内で実施可能であれば実施することの繰り返し)

## オリンピックで学ぼう!

情報オリンピックは、高校生以下を対象とした歴史ある競技系のコンテストである。参加者は個人で複数の問題に取り組み、アルゴリズムの良さやプログラミングの速さを競う。1994年度からは日本情報オリンピック<sup>☆2</sup>が開催され、上位入賞者は国際情報オリンピックに派遣される。

このような競技系では特に、実行時間や使用メモリ量に厳しい制約が課され、単純な考え方では解けないことが多い。

たとえば、XY座標平面上にN個（ただし $N \geq 2$ ）の点があるときの最も近い2点間のユークリッド距離を求められたとする。まずは図-4のように、すべての点対の距離を調べることを思いつくであろう。しかしこれでは、明らかに計算が不要であるような極端に離れた点対も含めてすべて調べることになってしまい、必要以上に実行時間がかかってしまう。

そこでもし実行時間の制約を満たせない場合、工夫が必要となる。たとえば、最初にすべての点をソートした上で、平面を左右に分割して左右それぞれにおける最短距離を再帰的に求めて短い方をdとし、続いて左側と右側をまたぐ点対の中で、距離がdを下回る可能性のある限られた点対について距離を調べる、という具合である<sup>6)</sup>。この考え方は「分割統治」の一種であり、ゲーム系のコンテストにおいても用いられる。

まずは前述のオンラインジャッジや、情報オリンピックの過去問に挑戦することから始めるとよい。その中で、単純なアルゴリズムを検討し、続いてより良いものを検討、さらには応用へと進む、という具合に自ずと段階的に学びを深められる。

```
int i, j;
double d = DBL_MAX;
for(i = 0; i < N - 1; i++) {
    for(j = i + 1; j < N; j++) {
        double dx = px[i] - px[j];
        double dy = py[i] - py[j];
        d = fmin(d, sqrt(dx * dx + dy * dy));
    }
}
```

図-4 最近点対問題の素朴なCコード例

## 実力を把握し上げていこう!

本稿では、プログラミングのラーニング・スルー・コンテストを解説し、コンテストの種類および具体例としてSamurai Codingと情報オリンピックを紹介した。掲載例から分かるように、最初の一步はそう難しくなく、初学者が飛び込んでいける。そしてコンテストが持つゲーム要素により、楽しみながらゲーム感覚で客観的に実力を把握し高められる。2016年10月ごろに募集開始予定のSamurai Coding 2016-2017を始めとして、ぜひコンテストに挑戦されたい。

### 参考文献

- 1) Walz, S. P. and Deterding, S. : The Gameful World: Approaches, Issues, Applications, MIT Press (2015).
- 2) 渡部有隆: オンラインジャッジではじめるC/C++プログラミング入門, マイナビ (2014).
- 3) Sakamoto, K., Ohashi, A., Shimizu, M., Takahashi, S., Murakami, S., Washizaki, H. and Fukazawa, Y. : A Pattern Language for Programming Contests with Game AI Competition, AsianPLOP (2011).
- 4) 坂本一憲, 大橋 昭, 鷺崎弘宜, 深澤良彰: コンピュータプレイヤーのプログラム作成を通して競い合うゲームプラットフォームの開発を支援するフレームワーク, 電子情報通信学会論文誌, Vol.J95-D, No.3, pp.412-424 (2012).
- 5) 近山 隆: Samurai Coding 2012を終えて, 情報処理, Vol.54, No.4, pp.400-401 (Apr. 2013).
- 6) 秋葉拓哉, 岩田陽一, 北川宜稔: プログラミングコンテストチャレンジブック, マイナビ (2012).

(2016年8月2日受付)

### 鷺崎弘宜 (正会員) ■ washizaki@waseda.jp

博士 (情報科学)。早稲田大学教授, 同大グローバルソフトウェアエンジニアリング研究所所長, 国立情報学研究所客員教授, (株)システム情報社外取締役。情報システム, ソフトウェアおよびプログラムの研究, 教育, 社会実装に従事。ISO/IEC/JTC 1/SC 7/WG 20 Convenor, IEEE Computer Society Japan Chapter Chair, SEMAT Japan Chapter Chair, IPSJ Samurai Coding Director, ACM-ICPC 2014 Asia Tokyo Regional Contest Director, IEEE ICST 2017 PC Co-Chair.

☆2 <https://www.ioi-jp.org/>