

ゲーミフィケーションによるコードリーディングの促進

岡 大貴^{1,a)} 西田健志^{1,b)}

概要: ソフトウェア開発において、他人の記述したコードを読解することは非常に重要である。プログラミング学習サイトや技術書の充実によりプログラミング学習自体のハードルは下がったものの、実践的なコードの読み書きへのステップアップを促す方法については課題が残されている。そこで本研究では、ゲーミフィケーションによって GitHub 上に存在する豊富なソースコードを読むことを促すツールを提案する。

1. はじめに

近年、IT 人材の不足や 2020 年度からの小学校教育におけるプログラミング教育の必修科などから分かるように、プログラミングの重要性が叫ばれている。Progate[1] やドットインストール [2] を代表とするプログラミングに関する Web 学習サイトなどが充実し、プログラミングを始めやすい環境が整ってきた。また技術書なども充実してきたことにより、コードを書いたり、ソフトウェアに関する知識を身につけるハードルは下がりがつつある。

しかしながら、プログラミングにおいては、それらのスキル・知識に加えて、コードを読解する技術も必要である。ソフトウェア開発において他人の記述したコードを読解することは非常に重要であり、その技術の向上に関しては課題が残されている。Web サイトや書籍でのプログラミング学習においては、学習用の平易なコードしか読む機会がなく、実践的なコードを読む機会は少ない。

プログラミング上級者は、GitHub[3] などに蓄積されているコードを読むことで実践経験を積むケースが多い。GitHub は複数人のソフトウェア開発者と協同開発を行うことができるオープンなプラットフォームであり、そこで公開されるソースコードを読むことは、初学者にとっても有益であると考えられる。しかし、GitHub 上のソースコードを読むことは入門直後の初学者にとって心理的なハードルが高い。

そこで本研究では、プログラミング学習者が実践的なソースコードに触れることを促進するシステムを提案する。本システムでは GitHub 上にあるソースコードを閲覧することにより、コードリーディングを促進する。またユーザ

がシステムを使用するハードルを下げるため、日常的に使用するツールである Slack からシステムを使用できるようにした。なお、ゲーミフィケーションの要素をシステムに取り入れることにより、能動的にユーザがシステムを利用するような工夫を施した。

2. 関連研究

GitHub のデータを利用した研究は今までも数多く行われている。永野らの研究では GitHub とコンピュータや情報技術に関する Q&A サイトである Stack Overflow[4] のデータを用いてその相関について調査している [5]。この研究では、ユーザが GitHub で作成したリポジトリと Stack Overflow への投稿のコンテンツの関連性について調べたもので、双方への投稿コンテンツに一定の相関があることを示している。また GitHub 自体の利用について、柴藤らは GitHub 上の断片データに関する情報を取得できるシステムを開発している [6]。このシステムではユーザが指定したソースコード中の一部の連続したコードに関するプルリクエストを取得できるというものであり、ソースコードを閲覧する際の利便性を高めている。

また、システムを構築する際に日常的に利用するツールにシステムを組み込んでいる例もある。樋川らの研究では、日常的に使用することの多いコミュニケーションツールである LINE を使い、bot を導入することにより実験の協力者を募りやすいようなシステムを提案している [8]。

なお、ゲーミフィケーションに関する研究も昨今では盛んである。一ノ瀬らの研究では、ソースコード上の技術的負債を可視化し、さらにゲーミフィケーションの要素を加えることによってその除去を促している [7]。この研究ではソースコードのファイル構造を街のように可視化し、技術的負債が存在するファイルを目立たせ、さらに技術的負債を取り除いたユーザをランキング形式で表示することに

¹ 神戸大学

^{a)} 198c125@stu.kobe-uac.jp

^{b)} tnishida@people.kobe-u.ac.jp

よって、ゲーミフィケーションの要素を元に生産的な行動を促している。

これに加え、コードリーディングに関してもいくつかの研究がなされている。石尾らや大村らはソースコードを読解する際のコードリーディング支援を行うツールを開発している [9][10]。しかし、これらの研究ではコードがある程度書けるプログラマがソースコードを読むという状況を想定しているため、コードに触れる機会を増やすための本研究とは目的が異なる。

3. 提案システム

3.1 目的と設計指針

本研究の目的は、GitHub にある実践的なコードを読むことを促進することで、初学者向け学習コンテンツからのステップアップを促進することである。この目的を達成するため以下の設計指針を提案する。

- (1) 遊びの要素を取り入れる
- (2) 日常的に使用するツールに組み込む
- (3) プログラムを読むことに専念させる

3.1.1 設計指針 1：遊びの要素を取り入れる

1つ目の設計指針に関しては、このシステムを使用する際、目的を達成するためにはユーザが継続的にシステムを使用することが必要である。そのためにはユーザがシステムを使いたいというモチベーションを起し、能動的に使用したくなるような工夫が必要であると考えた。そのため、ゲーミフィケーションなど遊びの要素を取り入れる設計指針を立てた。

3.1.2 設計指針 2：日常的に使用するツールに組み込む

2つ目の設計指針に関しては、ユーザにシステムを使用してもらうためには、使用するハードルが極力低くなるように設計すべきである。そのため、日常的に利用するツールに組み込むことが必要であると考え、設計指針 2 を導入した。

3.1.3 設計指針 3：プログラムを読むことに専念させる

プログラムの記述を促進するならば対象のソースコードを1つの言語に絞るべきという考え方もある。コードを書くことが目的であれば、複数の言語を同時進行させることは混乱を招き、デメリットが大きい。しかし、コードを読むことが目的であれば、上級者との会話のきっかけができる、将来的に適材適所で言語を使い分けることにつながる、飽きずにシステムを使用させ、コードの読解に興味を持たせることにつながるなどメリットが大きい。従って本システムでは、多様な言語に触れられる機能と1つの言語に読解を絞った機能を実装した。

3.2 システム概要

今回は日常的に使用されるツールとして、チームコミュニケーションツールである Slack を選び、システムを Slack から bot として使用できるように設計した (設計指針 2)。本システムでは、Slack からコマンドを入力することによって、コマンドに応じたソースコードのファイルを bot が表示し閲覧することができる。またシステムをユーザに能動的に使用させる仕組みとしてゲーミフィケーションの要素を組み込み、「占い」と「クイズ」の2つの機能を持たせた。

3.3 システムの機能

以下で実装した各機能の詳細について説明する。本システムの提供するコマンドは表 1 の通りである。

表 1 システムの機能

コマンド	説明
fortune	1日の運勢を占うソースコードを表示
quiz	記述言語を問うクイズを出題
hint	クイズに関するヒントを表示
answer	クイズの回答を表示
score	各ユーザの累計得点を表示

3.3.1 占い機能

この機能は GitHub から得られた1つのランダムなソースコードから1日の運勢を占うというものである。表示されたソースコードの意味を解釈することが必要であるため、必然的にコードの読解が必要となる。この機能は Slack 上で「fortune」というコマンドを実行することにより、GitHub gist 上のソースコードからランダムにその日の運勢を占うコードが表示される。この実行結果を図 1 に示す。



図 1 占い機能の様子

この機能は、カメの甲羅に熱を加えて生じたヒビの形状をみて占いをする「亀卜」のようにシステムによってランダムに取得されたソースコードを読解することによって、そのソースコードを占いとして利用できるようにしたものである (設計指針 1)。

3.3.2 クイズ機能

この機能は GitHub から得られた 1 つのランダムなソースコードがどの言語で記述されているかをクイズ形式で答えるというものである。この機能は Slack 上で「quiz」というコマンドを実行することにより、GitHub gist 上のランダムなソースコードが 1 つ得られる。そしてユーザはソースコードがどの言語で記述されているかを Slack 上で回答する。Slack 上で回答を送信し正解したユーザは 10 ポイントの得点が得られる。そしてこの得点の累計が週間で最も多かったユーザが週ごとに発表されるという仕組みである。この実行結果を図 2 に示す。



図 2 クイズ機能の様子

また、コードの言語が難解な場合には「hint」というコマンドを実行すると、正解の言語の文字数が表示され、正解が分からない場合には「answer」と入力すると正解の言語が表示され、クイズが終了する。また全ユーザの累計得点を「score」というコマンドを実行することによって取得することができる。1 週間ごとに累計得点の最も多かった者が Slack 上で通知され、また累計得点は 1 週間ごとにリセットされる。

この機能は、クイズ形式のため、ゲーム感覚で利用できる (設計指針 1)。またランダムなソースコードが取得されるため、多様な言語に触れることができ (設計指針 3)、それぞれの言語に対し理解を深めることができる。また各言語の特徴に着目しコードを読解する必要が必然的に生まれるため、言語の構文の理解につながると考えられる。

3.4 ソースコードの選定および表示方法

システムを導入するツールに Slack を使用した理由として、Slack 上ではソースコードをスニペットとして言語ごとにハイライトして表示することができるという点がある。また表示するソースコードのファイルは断片的なものでは

読解しづらく、1 つのファイルに内容が集約されているものが望ましい。そのため今回は、GitHub gist からソースコードを取得するようにした。

4. ケーススタディ

提案システムの効果および課題を調査するために、著者が所属する研究室のメンバーにシステムを 1 週間使用してもらいケーススタディを行った。研究室のメンバーは 3 名のプログラミング初学者であり、自由に機能を使用してもらい、各コマンドの使用回数を調べた。またインタビューによって所感を調査した。

なお、占い機能に関しては、メンバー内に JavaScript を使用する者が多かったため、表示するコードを JavaScript に固定した。

行ったインタビューの内容は以下である。

- どういう状況で機能を使用したか
- 占い機能について、楽しかったか (5 段階評価, 1: 楽しくなかった, 5: 楽しかった)
- 占い機能について、どういう点が楽しかった (楽しくなかった) か
- クイズ機能について、楽しかったか (5 段階評価, 1: 楽しくなかった, 5: 楽しかった)
- クイズ機能について、どういう点が楽しかった (楽しくなかった) か
- 機能を使用したことによって学びがあったか
- どういう学びがあったか
- コードを読むことへの抵抗は減ったと感じるか (5 段階評価, 1: 減らなかった, 5: 減った)
- 今後も機能を使いたいのか
- 感想, 意見, 改善点など

4.1 使用結果

結果、各コマンドの実行回数は表 2 のようになり、quiz 機能に関連したコマンドの使用回数が多い結果となった。またケーススタディを行った最初の 3 日間ほどは頻繁に機能が使用されていたが、後半にはあまり使用されていなかった。またどの機能も個人で使用するより、研究室にメンバーが集まる際に多く使用される結果となった。

表 2 各コマンドの実行回数

コマンド	使用回数
fortune	9
quiz	43
hint	27
answer	3
score	33

4.2 インタビュー

インタビューの結果は表3のようになった。

表3 インタビュー結果

インタビュー内容	被験者 A	被験者 B	被験者 C
占い機能は楽しかったか	1	2	3
クイズ機能は楽しかったか	4	4	5
コードへの抵抗感は減ったか	3	5	5

使用した状況に関しては、空き時間に使用する場合や、研究室のメンバーで集まる場合に使用するという回答が得られた。

また占い機能に関しては、あまり楽しかったという意見は得られず、表示されたソースコードをどう解釈すればいいのか分からず、占いという感覚が得られないという意見があった。クイズ機能に関しては、楽しかったという意見が多く、スコアを他人と競えるため楽しめたという意見が多かった。

さらに、機能を利用した際の学びに関しては、全員があったと答え、「言語ごとの文法の特徴が何となく分かってきた」「その言語を知らない意識して見る機会がないので、いろんな言語の存在を知ることができた」「知らないプログラミング言語の名前と文法を知り、親近感を持った」というポジティブな意見が得られた。

これに加え、感想、意見、改善点などの項目では、クイズに関して、「答えと一緒に言語の特徴が表示されれば覚えられそう」「間違えた際に、惜しいなどのコメントをしてくれたら嬉しい」、占いに関して「占いの結果に沿って議論ができると楽しいかも」などの意見が得られた。

4.3 観察

システム使用時のユーザの様子を観察したところ、インタビューの結果にもあったように、研究室のメンバー同士が集まっている際に使用が活発になっていた。特にクイズ機能において、早押しクイズのように使用されている様子が見られた。

また分からない言語で記述されたクイズが出題された際に、hint コマンドによって得られた文字数を元に当てずっぽうに回答している様子も見られ、hint コマンドが上手く機能していないケースが伺えた。

なお GitHub gist 内に C# で記述されたプログラムが多く、クイズにおいて言語の偏りが見られた。

5. 今後の展望

今回のケーススタディの結果、システムに対しポジティブな反応が得られたものの、多くの改善点が見られた。

まず、今回用意したシステムはユーザの使用状況を考慮していなかった。そのため、インタビューで得られた、空き時間や人と集まった際に頻繁に使用されるという状況を

考慮し、早押しクイズのようなシステムにするなど改善を加えたい。

また占い機能に関して、占いという感覚を増強するよう bot から選ばれたソースコードについてコメントするなどの機能を追加したい。さらに、クイズ機能に関して、言語の偏りがある点を改善し、言語についての豆知識を表示するなど、よりプログラミング言語に関する学びが得られ、それを楽しめるような機能を追加したい。

これに加え、システム全体としても恒常的に利用されるようゲーミフィケーションの要素についても追加していきたい。

6. まとめ

本研究では、コードリーディングを促進するために、「占い」「クイズ」といったゲーミフィケーションを用いて GitHub 上に存在するソースコードを読むことを促すツールを提案した。

ケーススタディとして筆者の所属する研究室のメンバーに実際に利用してもらい、機能の利用回数の測定・インタビュー調査を行った。結果、システムはミーディングなどで集まる機会の前後に主に利用され、日常的に利用されたとはまでは言えなかった。またインタビューの結果から多くの改善点が見られた。

この結果を踏まえて、日常的に利用される有益なシステムを構築し、評価実験を行いたい。

参考文献

- [1] Progate, <https://prog-8.com/>.
- [2] ドットインストール, <https://dotinstall.com/>.
- [3] GitHub, <https://github.com/>.
- [4] Stack Overflow, <https://ja.stackoverflow.com/>.
- [5] 永野真知, 早瀬康裕, 駒水孝裕, 北川博之: GitHub と Stack Overflow におけるユーザ行動の統一的な分析, 情報処理学会第 79 回全国大会, pp. 363-364(Mar. 2017).
- [6] 柴藤大介, 有菌拓也, 宮崎章太, 矢谷浩司: CodeGlass: GitHub のプルリクエストを活用したコード断片のインタラクティブな調査支援システム, 情報処理学会 インタラクション, Vol.2019, pp.159-168(Mar. 2019).
- [7] 一ノ瀬智浩, 畑 秀明, 松本健一: ソースコード上の技術的負債除去を活性化させるゲーミフィケーション環境の開発, 情報処理学会関西支部 支部大会 講演論文集, Vol.2016, (Sep. 2016).
- [8] 樋川一幸, 松田滉平, 中村聡史: コミュニケーションチャンネルに入り込む研究室実験 BOT の提案と運用, 情報処理学会研究報告グループウェアとネットワークサービス, Vol.3, pp. 1-7(Mar. 2019).
- [9] 石尾 隆, 田中昌弘, 井上克郎: ソースコード上での情報タグ伝播によるコードリーディング支援, ウィンターワークショップ論文集, Vol.2008, pp.31-32, (Jan. 2008).
- [10] 大村 裕, 渡部卓雄: プログラム理解のためのコードリーディング支援ツールの提案と実装, 日本ソフトウェア科学会 講演論文集, Vol.31, pp.443-446, (Sep. 2014).