

内容梗概

プログラミングが重要視されている昨今では、プログラミングの授業が小学校に導入されるなど、プログラミングの一般教養化が推し進められている。しかしプログラミングを楽しむためにはある程度の習熟を必要とする。プログラミング初学者でもプログラミングの楽しさを感じられるように学習コンテンツをデザインできれば、初学者の学習モチベーションを損ねることなく、プログラミングをさせることができると考えられる。

そこで本研究では初学者の継続的なプログラミング学習を促すため、2つのアプリケーションを開発した。1つ目はエンタテインメントによりコードリーディングを促進するシステムである。このシステムではクイズ、占いといったエンタテインメントを交えてGitHubにあるソースコードを表示し、ユーザに読解させることにより、楽しみながらコードリーディングを促進することを目指した。実装したアプリケーションを使用し、筆者の所属する研究室のメンバーを対象に運用を行い、その後アンケート調査を行った結果、クイズに関しては楽しかったという回答が得られたが、ややプログラミング初学者にとっては難解であるという意見が得られた。また日常的な使用を促すために更なる工夫が必要であり、これらの問題点を今後改善していく予定である。2つ目はライブコーディングの要素を取り入れた対人形式のプログラミングゲームである。このシステムでは習熟度の高いプログラマがそのプログラミングスキルによって力量を競うことができ、プログラミング初学者でも観戦して楽しむことができるゲームの作成を通して、初学者のプログラミングに対する興味関心を高めることを目指した。評価実験では実際に2人のプログラマの対戦を初学者に観戦させ、プレイヤであるプログラマと観戦した初学者の双方にアンケート調査を行った。実験結果としてシステムのプレイ・観戦は楽しく、プログラミングに対する興味が高まったなどシステムに対する肯定的な意見が多く得られたが、ゲームシステム・デザインの改善点に関するコメントも多く得られた。この結果を受け、今後は提案システムを改善すると共に、より多くのプログラミング初学者を含むプログラマにシステムを使用させ、より良いシステムの構築を目指す。

目次

1	はじめに	1
2	コードリーディング支援システムに関する研究	2
2.1	関連研究	2
2.2	設計指針	2
2.3	提案システム	2
2.4	ケーススタディ	2
2.5	課題と考察	2
3	競技性・観戦性を拡張したプログラミングゲームの開発	3
3.1	関連研究・関連システム	3
3.1.1	プログラミングを用いたエンタテインメントシステム	3
3.2	設計指針	3
3.3	提案システム	4
3.3.1	自機プログラミングフェイズ	4
3.3.2	行動プログラミングフェイズ	5
3.3.3	ゲームフェイズ	5
3.4	評価実験	5
3.4.1	実験参加者	5
3.4.2	実験内容	6
3.4.3	実験結果	6
3.4.4	考察	7
3.5	課題	7
4	まとめ	8
	謝辞	9
	参考文献	10

1 はじめに

昨今、自然言語での読み書きや健康のために行う運動のようにプログラミングを一般教養にしようとする動きがある。小学校におけるプログラミング教育の必修化がその最たる例で、将来プログラミングに関わる職業につかない人でもプログラミングに関する知識やその考え方をある程度身につける方が望ましいと考えられている。

しかし最初から理解が容易で楽しさを感じやすい読み書きや運動とは異なり、プログラミングは楽しさを感じるまでにある程度の習熟を要する。よく書かれた小説などの文章やプロスポーツでのファインプレーには初心者でも心動かされるもので、そうした小説家やスポーツ選手への憧れが自ら書くことや体を動かすへの感心を高めるが、プログラミングの場合にはそのような憧れを持つ機会に乏しいのが現状である。プログラミング初学者にプログラミングの楽しさを実感させるために設計された学習コンテンツは多数存在するが、その多くは小・中学生など若い年代をターゲットに設計されているため、高校生や大学生などには向かない。

本研究ではプログラミング初学者のプログラミングに対する興味関心を高めるため、ゲーミフィケーションやゲームといったエンタテインメントの要素を組み込んだコンテンツを開発し、初学者を対象に運用を行うことでその効果を検証した。

本論文では以降、2章でゲーミフィケーションを用いたコードリーディング支援システムについて述べ、3章でプログラミング初学者の興味喚起を目的としたプログラミングゲームについて述べる。4章で本論文をまとめる。

2 コードリーディング支援システムに関する研究

2.1 関連研究

2.2 設計指針

2.3 提案システム

2.4 ケーススタディ

2.5 課題と考察

3 競技性・観戦性を拡張したプログラミングゲームの開発

本稿では、プログラミング初学者の興味喚起を目的としたプログラミングゲームのプロトタイプについて紹介し、設計指針に基づくデザインの詳細や行った評価実験と見つかった課題、実験に関する考察について議論する。

3.1 関連研究・関連システム

3.1.1 プログラミングを用いたエンタテインメントシステム

プログラミングとエンタテインメントを掛け合わせたコンテンツはいくつか存在する。TopCoder[1]などの競技プログラミングやコードゴルフ[2]、SECCON[3]などのハッキングコンテストが有名であり、プログラマの間でも根強い人気がある。またプログラミングゲームとしてはRobocode[4]が有名である。しかしこれらはある程度プログラミングに習熟したプログラマ向けのコンテンツであるため、数学やセキュリティ、コンピュータサイエンス等の知識を要するためプログラミング初学者が参加するにはハードルが高い。

3.1.2 エンタテインメントを活用したプログラミング学習支援システム

エンタテインメントの要素を盛り込むことでプログラミング学習を促進しようとした研究は多くある。Joshaらはプログラミングゲームを用いてプログラミング初学者の問題解決能力を向上させるためのシステムを作成している[5]。また水口の研究ではプログラミングの講義における成績評価にロボットバトルシミュレーション型のプログラミングゲームを活用している[6]。三谷らの研究ではキャラクタをプログラムで制御するプログラミングゲームでプログラミングスキルの向上を図っている[7]。なお増谷らの開発したVLogic[8]ではVR空間上にブロックベースのプログラミングゲームを実装することで手足を使ってプログラミングを体験することができ、プログラミングに対する興味喚起を行っている。これらはプログラミングの習熟度が高くなくても使用できるが、従来のプログラミングゲーム同様静的なゲーム展開であり、プログラマ同士のリアルタイムな駆け引きやアドリブといった観戦を楽しむ設計は成されていない。

3.2 設計指針

提案システムの実装にあたり、初学者の興味関心を高めるために3つの設計指針を設けた。

1. 観戦するにあたり，高度な専門知識を必要としない

初学者が提案システムでの対戦を観戦するにあたり高度な専門知識を必要としてしまつては，利用の心理的ハードルを上げかねない．極力前提知識なしに理解し，楽しめるようにデザインする必要がある．

2. 手間がかからない

初学者にとってプログラミング学習をする際に環境構築や普段使用しない独自ソフトウェアのインストールはモチベーションを下げかねない．今回は提案システムを JavaScript によって制御可能な Web アプリケーションとして実装し，初学者が実際にプログラミングを行わずとも見るだけで利用できるようにした．

3. リアルタイムな駆け引き・アドリブを取り入れる

従来のプログラミング教育コンテンツとは異なり，スポーツなどの観戦する競技で見られるようなリアルタイムな駆け引き・アドリブの要素を提案システムに組み込むことで観戦して楽しめるようなデザインをする．

3.3 提案システム

本研究で提案するシステムでは，プログラミング初学者の興味喚起をするためにいくつかの工夫を施した．今回実装したシステムは，プログラマ同士がリアルタイムにプログラミングを行うことでキャラクタを制御し，対戦するという対人形式のプログラミングゲームである．この対戦の様子を初学者に観戦させることで，初学者のプログラミングに対する興味を高める．プログラミングゲームとして実装した理由としては，キャラクタがプログラムによって動作するというゲームの視覚的な出力が，初学者にとってプログラムの出力を理解する助けになると考えたためである．ゲームジャンルとしては，シューティングゲームの体裁をとった．これはシューティングゲームが「敵の攻撃を避けて，敵を攻撃する」というプリミティブなゲームシステムであり，見ていて展開を理解しやすいと考えたためである．またゲーム UI は LivecodeLab や Hydra などのビジュアルライブコーディング環境を踏襲し，ゲーム画面上にエディタを重畳することで，プログラムとその出力の双方を同時に見ることが可能なように設計した．またゲームは以下の3つのフェイズに分かれている．

3.3.1 自機プログラミングフェイズ

対戦が開始するとこのフェイズに移行する．ここでは予めエディタにプレイヤーが操作するキャラクタのコンストラクタが記述されており，プレイヤーはパラメータを書き換える

ことができる。具体的なパラメータには appearance, life, clock, power がある。appearance はキャラクタの外見であり、文字列を指定できるため、絵文字などを使ってプレイヤーの好きな見た目を選ぶことが可能である。life はキャラクタの体力であり、いわゆる HP(ヒットポイント)を表している。非負の整数を指定でき、この値が 0 以下になるとプレイヤーはゲームに敗北する。clock はキャラクタが行動できる回数の多さを表しており、非負の整数を指定できる。プレイヤーは後述する行動プログラミングフェイズにおいて自分のキャラクタを制御するプログラムを記述し対戦するが、その際に記述したプログラムは 10 秒間ループして実行される。このループのインターバルを決めるのが clock であり、値が大きいほどインターバルは短くなる。power はキャラクタの攻撃力を表しており、これも非負の整数を指定できる。この値が大きいほど、自分が操作するキャラクタの攻撃が相手キャラクタに命中した際に削る life の値が大きくなる。双方のプレイヤーが各パラメータを記述し終わると次の行動プログラミングフェイズに移行する。

3.3.2 行動プログラミングフェイズ

このフェイズに進むと、自機プログラミングフェイズで記述したコンストラクタを元に各プレイヤーが操作するキャラクタのインスタンスが作成され、ゲーム画面が表示される。各プレイヤーはプログラムをエディタに記述し、作成したキャラクタを操作する。プレイヤーは条件分岐や繰り返しなど従来の JavaScript の文法の他に独自に用意されたプロトタイプメソッドを使うことができる。用意したメソッドにはキャラクタを移動するメソッド (moveUp(), moveDown(), randomMove()) とキャラクタが攻撃を行うメソッド (shot()) などがある。またプログラム内で各キャラクタのパラメータを参照することもできる。両プレイヤーがプログラムを記述し終わると次のゲームフェイズに移行する。なおこのフェイズでは相手プレイヤーがどのようなプログラムを記述しているかは見ることはできない。

3.3.3 ゲームフェイズ

このフェイズでは行動プログラミングフェイズで記述したプログラムが 10 秒間ループして実行され、ゲームが進行する。この段階で両プレイヤーは相手プレイヤーが記述したプログラムを閲覧することができる。このフェイズにおいて相手キャラクタを攻撃し、life の値を 0 以下にしたプレイヤーの勝利となる。勝敗が決まらない場合はプログラム終了時の各パラメータを引き継いだまま行動プログラミングフェイズに戻り、再度プログラミングしゲームフェイズに移行するという過程を勝敗が決まるまで繰り返す。

3.4 評価実験

提案システムの使用・観戦に関する感想や影響、用法を調査するために評価実験を実施した。システムをプレイするプログラマと観戦するプログラミング初学者を集め、システムでの対戦と観戦を実施し、アンケート調査と実際に対戦で使用されたプログラムのログを分析することでシステムを評価した。

3.4.1 実験参加者

実際にゲームをプレイするプログラマとしては、プログラミング(主にオブジェクト指向言語)の経験が3年以上ある大学院生2名(男性)に声をかけた。2名とも日常的にアクションやシューティング等のジャンルのゲームをプレイするため、システムをプレイする際にゲームに不慣れなためハンデが生まれることはないと思われる。また両者とも3年以上JavaScriptを使用した経験がある。またプレイを観戦するプログラミング初学者は、国際人間科学部にて開講されていた講義「プログラミング基礎演習1」の受講者をとした。受講者のうち、実験に参加した者は77名であり、うち37名が男性、40名が女性であった。またこの講義ではJavaScriptにおける変数、条件分岐、繰り返しなどの基礎的な文法を教えており、実験参加者は実験を行う時点でこれらを学習済であった。なお、うち23名は授業以前にプログラミングを学習した経験があったが、ソフトウェア開発を行ったことのある者はいなかった。またプログラマ含む実験参加者の全員が、競技プログラミングやプログラミングゲームなどのプログラミングを題材としたエンタテインメントを使用した経験がなかった。

3.4.2 実験内容

初めにプログラマ2人に簡単な事前アンケートを行った後、提案システムにある程度慣れ、用法を理解してもらう必要があるため、システムの練習をするための期間を設けた。システムの使用方法、ゲームシステム、独自に用意したメソッドなどについて説明した後、11/13 11/19の1週間システムを自由に使用させた。また、ただシステムを使用させただけではゲームに対する理解度が上がらない可能性があるため、期間中に2つのタスクをこなさせた。1つは1人以上とシステムを使った対人戦を行うことであり、もう1つは相手がランダムな戦略を実行してくる対CPU戦において、勝率が高いと考えられるプログラムを作成することである。なお期間中はシステムに関する意見・疑問を逐次報告させ、システム使用における問題を改善した。練習期間が終わった翌日に、プログラマ同士の対戦を行った。対戦はシステムに関するプログラマ同士のコミュニケーション等の所作を観察するため、感染症対策を徹底した環境で対面に

て行った。両者の対戦時の画面を録画し、対戦後にシステムに関する事後アンケートを行った。なお練習期間中・対戦中にプログラマが記述した全てのコードのログを収集した。そして「プログラミング基礎演習1」の最終講義で受講者に事前アンケートを行った後、初学者が観戦しやすいように対戦動画を編集したものを zoom を介して閲覧させた。またその後にシステムに関する事後アンケートを行った。

3.4.3 実験結果

1. アンケート結果
2. コードログ分析結果

3.4.4 考察

評価実験に関する考察について述べる。プログラマ同士の対戦においては、行動プログラミングフェイズで設定した appearance について「かわいい」などとコメントしていた。また対戦後にお互いのプログラムの内容や今までのプログラミング経験に関するコミュニケーションをとっており、システムを使用することでプログラマ同士のコミュニケーションを促進できていたと考えられる。初学者の対戦の観戦においては、ライブでプログラマ同士が対戦している状況を用意することが困難であったため今回は動画を閲覧するという状況を用意したが、動画を見ただけでは実際に人が対戦しているという感覚が希薄であり、「プログラマがプログラミングしている」場面を見せるためには更なる工夫が必要であると感じた。また今回の評価実験において、システムに対する肯定的な評価が得られたものの、どのような要素が初学者のプログラミングに対する興味に影響を与えていたのか、特に提案システムにおいて独特な要素であるリアルタイムな駆け引き・アドリブを誘発する要素の影響について調査する必要があると考えられる。

3.5 課題

4 まとめ

謝辞

本研究を行うにあたり，日頃より御指導，御激励を賜り，数々の御教示を頂きました西田健志准教授に深甚なる謝恩の意を表します．また神戸大学大学院国際文化学研究科に在学中，御教示，御激励頂いた神戸大学大学院国際文化学研究科の諸先生方に感謝すると共に，諸職員の方々に感謝いたします．日頃より数々の御助言を下さいました諸先輩方，快適な環境を作って頂いた研究室の皆様方，実験に協力頂いた実験参加者の方々に深く感謝いたします．特に研究活動に対する多くのアドバイスとサポートを頂いた工学研究科の清水友順氏，国際文化学研究科の三嶋哲也氏に深く感謝いたします．

参考文献

- [1] Topcoder, <https://www.topcoder.com/>.
- [2] 浜地慎一郎: Code Golf, http://shinh.skr.jp/dat_dir/golf_prosym.pdf.
- [3] SECCON, <https://www.seccon.jp>.
- [4] Robocode, <https://robocode.sourceforge.io/>.
- [5] J.Shi et al: Pyrus: Designing A Collaborative Programming Game to Promote Problem Solving Behaviors, Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, No. 656, pp. 1–12(May.2019).
- [6] 水口充: 成績評価のためのプログラミングゲームの設計と実践, 研究報告エンタテインメントコンピューティング (EC), vol. 2016, pp. 1–7(July.2016).
- [7] 三谷将大, 寺田実: Web アプリケーションによるゲーミフィケーションを用いたプログラミング上達支援システム, 第 27 回インタラクティブシステムとソフトウェアに関するワークショップ, (Sept.2018).
- [8] 増谷海人, 赤澤紀子: 仮想現実を用いた初学者向けプログラミング学習システムの提案, 2018 年度情報処理学会関西支部支部大会講演論文集, vol. 2018, (Sept.2018).