

ゲーミフィケーション要素を用いたプログラミング教材の提案

広瀬研究室 4年伊藤理乃

令和6年1月9日

概要

プログラミング言語の学習は情報教育において重要な役割を担っている。学生の時にプログラミングのスキルを身につけることで論理的思考力や想像力, 問題解決能力の向上を図ることが可能である。プログラミング言語を学習し理解する際には, 教員が行う講義だけではなく講義時間外に学生が自主的に行う自学習が必要である。しかし, 問題点としてプログラミング言語の自学習を行う上で学習意欲の維持・向上が難しいという点が挙げられる。そこで本研究では, プログラム学習者の自学習のモチベーションの維持・向上を推進させ, 継続的なプログラミング学習を行うことを目的としたプログラミング教材の提案を行う。本研究では, ゲーミングを行うことを目的としないシステムなどに「レベルアップ」や「スコア競争」などのゲームの要素を取り入れることでモチベーションの維持や向上を図ることが期待できるものであるゲーミフィケーション要素を取り入れたゲーミフィケーション手法を利用し, 自学習のモチベーションの維持・向上を図る。(434 字)

目次

第 1 章	はじめに	5
1.1	背景	5
1.2	基礎プログラミング I・II	6
1.2.1	基礎プログラミング I・II での取り組み	6
1.2.2	学生の活動時間と理解度	6
1.3	研究目的	7
第 2 章	関連研究と類似サービス	9
2.1	類似サービス	9
2.1.1	忍者 CODE	9
2.1.2	Progate	9
2.2	関連研究	10
2.2.1	ゲーミフィケーション要素を取り入れた小学生の学習効果	11
2.2.2	ゲーミフィケーション要素を活用したウォーキング支援	11
2.2.3	技術者のモチベーションに関する研究	11
2.2.4	ゲーミフィケーションにおける構成要素の有用性に関する研究	11
2.3	関連研究での課題点	12
第 3 章	ゲーミフィケーションを用いたプログラミング教材の提案	13
3.1	関連研究からの考察	13
3.2	ゲーミフィケーション要素	13
3.3	要素を使用する期間	14
3.4	使用する要素と手法の提案	15
3.5	システム提案	17
第 4 章	システム設計	19
4.1	本システム全体設計	19
4.2	システム選択画面の設計	19
4.2.1	システム選択画面使用手法	20
4.3	クイズ形式学習機能画面の設計	20
4.3.1	クイズ形式学習機能使用手法	21
4.4	コード記入型学習機能画面の設計	21

4.4.1	コード記入型学習機能使用要素	22
第 5 章	システム開発	23
5.1	システム全体の開発	23
5.2	開発環境	23
5.3	クイズ形式学習機能	23
5.3.1	クイズ形式学習機能の概要	23
5.3.2	クイズの内容	24
5.3.3	クイズ形式学習機能の開発	24
5.4	コード記入型学習機能	26
5.4.1	コード記入型学習機能の概要	26
5.4.2	学習問題の内容	26
5.4.3	コード記入型学習機能の開発	26
5.5	予備実験	27
5.5.1	予備実験結果	27
5.5.2	予備実験からのシステム改良	29
第 6 章	システムの実験・検証	31
6.1	実験・検証概要	31
6.2	ゲーミフィケーション要素の有無の実験	31
6.2.1	実験使用資料	32
6.2.2	実験結果	33
6.3	ゲーミフィケーション要素の効果の検証	34
6.3.1	検証結果	35
第 7 章	結論	37
	参考文献	41

第1章

はじめに

本章では、研究を行なっていく上での背景や目的について説明する。

1.1 背景

現代社会では、AI（アーティフィシャル・インテリジェンス）やビッグデータ、ロボットテクノロジーなどの最先端技術が高度化し、様々な知識や情報が共有され、今までにない新たな価値を生み出している。これにより、ロボットや自動走行車などの技術で、少子高齢化、地方の過疎化、貧富の格差などの課題や困難を克服する世界を「Society5.0」と呼称している [1]。文部科学省は、2020年度から小学校段階でのプログラミング教育の導入を決定した。プログラミング教育では、コンピュータに意図した処理を行うよう指示することができるということを体験させながら、「知識及び技能」「思考力、判断力、表現力」「学びに向かう力、人間性等」を身につける狙いがあるとしている [2]。プログラミング言語は講義から文法的要素やコンピューターにさせる処理を学ぶことによる論理的な学習だけでなく、学習者が実際にコードを記述することでプログラミング的思考が身につく、コードの書き方が身についていく。プログラミング的思考は演習の中でトライ&エラーに取り組み、感覚的にコードが書けるようになることで身につくが、数時間という講義内での学習のみではプログラミング的思考を理解するには難しく、学習者が個別に演習課題に取り組む必要がある。しかし、個別に演習課題に取り組むにあたって順調に演習課題を進めている学生と比較して遅延や停滞している学生が多く存在してしまう [3]。個別にプログラミング演習課題に取り組む際に、エラーやバグの解決に時間を取られない手法として、Scratch^{*1} [4] などを用いた感覚的学習も推進されているが、プログラミング言語の文法にあまり触れることがないため、実際のコードに触れることがあまりない。学習始めたばかりの学習者や学習に苦手意識を持っている学習者の自学習による反復学習を向上させる手法としてゲーミフィケーションが挙げられる。ゲーミフィケーションとは、ゲーミングを行うことを目的としないシステムなどに「レベルアップ」や「スコア競争」などのゲームの要素を取り入れることでモチベーションの維持や向上を図ることが期待できるものである。近年では、日本語教育 [5] や SDGs [6] のように幅広い場面で課題解決の手法として利用されている。以上を踏まえ、本システムでは学習意欲の維持・向上を目的としてゲーミフィケーション要素を取り入れる。学習内容として、実践的にソースコードに触れることができるコード記入型学習機能とプログラミング言語をクイズを用いてインプットしながら学ぶことができるクイズ形式学習機能を実装する。

^{*1} 非営利団体 Scratch 財団によって設計、開発、維持されているシンプルなビジュアルインターフェースを持った子供向けプログラミング言語。

1.2 基礎プログラミング I・II

本研究では、プログラミング学習の研究背景として「基礎プログラミング I・II」の学習環境を用いて考察を行っていく。「基礎プログラミング I・II」とは、東北公益文科大学が開講している講義である。主にプログラミング言語である Ruby 言語の学びを中心としている講義であり、一つの教室に受講者 40 名がいる状態で教師が教える講義を受講する形の講義である。この講義は、一般的に使われているアプリケーションプログラム制限されることを、プログラミングの知識をベースに、あらゆる制約を取り去った自由なプログラムを作成し、問題解決能力の向上をはかるものである。

1.2.1 基礎プログラミング I・II での取り組み

基礎プログラミング I・II では、約 40 名の受講生が受講している。学生一人一人に一台の計算機を用意し、受講生はプログラミングの基礎を学ぶ手段として、Ruby 言語を利用して学習している。基礎プログラミング I では、Unix の基礎知識、基礎的文法である変数や入出力処理、制御構造、配列を学習する。基礎プログラミング II では、メソッド定義、ハッシュ、基礎的文法の応用、CGI プログラムの書き方を学習する。最終的にこれまで学んだ内容を活かし、課題解決プログラムを受講生全員が作成し、課題成果発表を行う。

1.2.2 学生の活動時間と理解度

「基礎プログラミング I・II」では、全 27 回の講義でプログラミング基礎知識からプログラミングを用いた実践的なデータ処理と情報表現を学ぶ。「基礎プログラミング I」では、12 回の講義内容を元に課題成果発表を行う。「基礎プログラミング II」では、4 回の講義を元に中間試験を行い、その後 6 回の講義を経て課題成果発表を行う。課題成果発表が終わった後に筆記によるプログラミング的思考が身についているかを図るための期末試験を行う。講義の際の受講生の活動プロセスを図 1.1 に示した。105 分の講義を行った後、自身で講義内容を理解できた場合は自身で講義内容を利用したオリジナルプログラムの作成、できなかった場合は再度講義内容の復習を行い、最終的に受講生全員がオリジナルプログラムに着手し、研究発表を行う。図 1.1 では、「講義内容を理解できているか」という部分から講義外の自学習である。講義内容を講義中に理解できなかった受講生と講義内容を講義中に理解できた受講生の双方において自学習は必要不可欠である。講義時間は受講生全員平等なため、受講生の理解度の違いは自学習の学習量の違いによって生まれる。しかし、自学習は学習者の学習意欲によって学習量が変わるため、学習意欲が低い受講生は学習を諦めてしまうことが起こりうる。

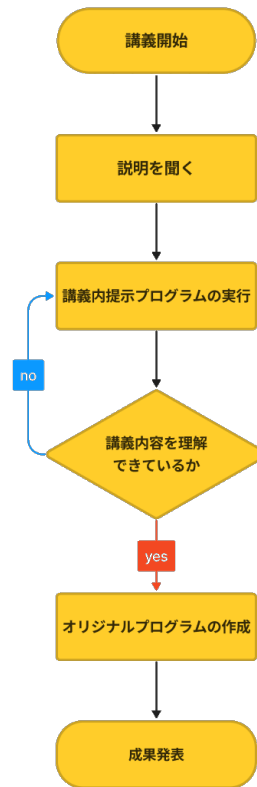


図 1.1 受講生活動プロセス

1.3 研究目的

本研究では現状の課題を解決するために、プログラム学習者の自学習のモチベーションの維持・向上を推進し、トライ&エラーを繰り返しながらプログラミング的思考を理解するための、ゲーミフィケーション要素を用いたクイズ形式学習機能・コード記入型学習機能から構成されるプログラミング教材の構築を目的とする。

第2章

関連研究と類似サービス

本章では、ゲーミフィケーションを用いて学習者のモチベーション維持・向上に活用するために既存のシステムや関連研究の内容を具体化する。

2.1 類似サービス

本研究にて構築するシステムであるクイズ形式学習機能・コード記入型学習機能と類似している機能を搭載しているシステムを説明する。

2.1.1 忍者 CODE

株式会社甲賀-KOGA-が開発したプログラミング学習サイトである [7]。さまざまなプログラミング言語をクイズ形式で学べる形になっている (図 2.1)。HTML, CSS と JavaScript, jQuery と PHP のように Web サイトを作成する際に使用する言語のクイズに特化している。HTML, CSS のクイズが全 91 問あるのに対し、Ruby 言語のクイズは全 15 問と少なくなっている。クイズの内容は、プログラミング言語の語句を選んでいくようなクイズではなく、実際に回答者の環境にあるテキストエディタで実行して回答を導き出し、回答することを想定して作られているクイズである。そのため、回答者は忍者 CODE サイトをは別にテキストエディタを使って実際にコードを実行することが必要である。

2.1.2 Progate

株式会社 Progate が開発したプログラミング学習サイトである [8]。指定したプログラミング言語の構文、メソッドについてスライド式で学習した後、画面上で実際にプログラミングを行う。プログラミングを行う画面では、実際にコードを入力していき、問題に正解すると次に演習へ進む。図 2.2 のように左側に回答して欲しいプログラムについての説明があり、中央にコードを記入する場所、右側に実行結果と実行結果の見本が表示される。回答者が答えに迷った場合は、スライドを何度も見ることができ、ヒントも表示できる。Ruby 言語についての学習も行うことができるが、基礎的な部分は無料で受講できるが、応用部分は有料である。



図 2.1 忍者 CODE 実行画面

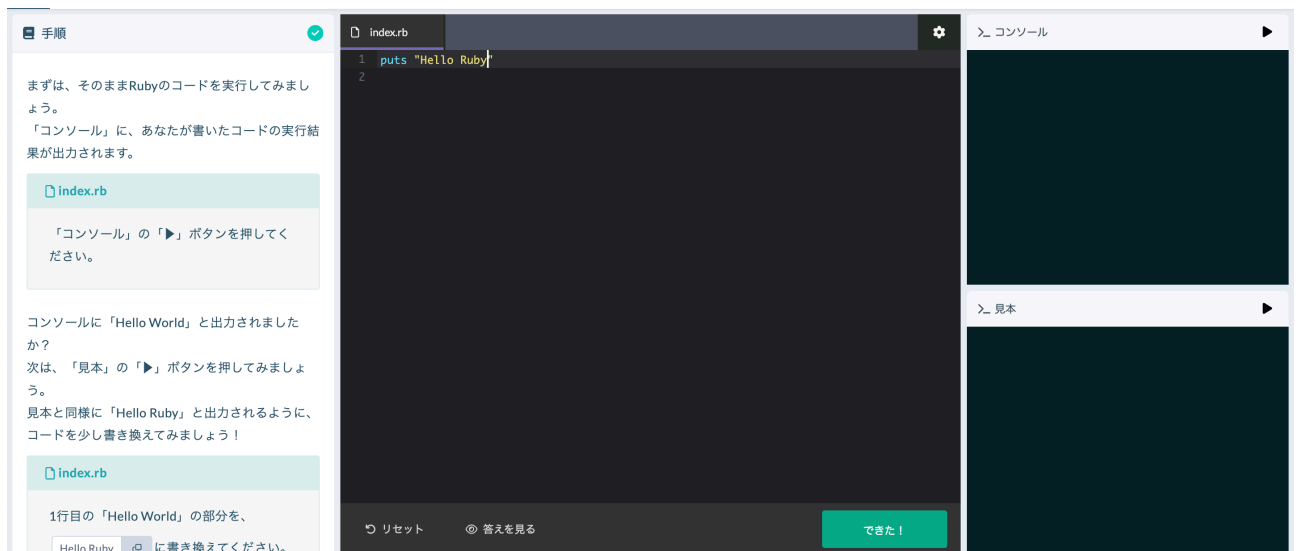


図 2.2 Progate 実行画面

2.2 関連研究

ゲーミフィケーションを利用した研究は今まで数多く行われている。

2.2.1 ゲーミフィケーション要素を取り入れた小学生の学習効果

福山佑樹らの研究では、ゲーミフィケーション教材を小学校に数週間導入した際の学習効果を検討すること、小学校教員がゲーミフィケーションにどのような印象を持ち、実際に導入してどのような影響を教材が与えたと認識したのかを検討することを目的とし、小学1年生用に開発されたゲーミフィケーション教材「アプリゼミ」*1を小学校において導入する実践を行った。使用するアプリのゲーミフィケーション要素の確認として、Dichevaら [9] の教育研究における先行研究をもとに「アプリゼミ」のゲーミフィケーション教材としての妥当性を判断した。ゲーミフィケーション教材の実践を2つの小学校において2週間の実践を行った結果、計算分野に関して30%以上点数が向上するなど大きな成果が見られた。ゲーム要素によって児童がドリル形式の学習を楽しんで行う事ができたことに由来する事が示唆された [10]。

2.2.2 ゲーミフィケーション要素を活用したウォーキング支援

藤田美幸らの研究では、ドコモ・ヘルスケアがリリースしたウォーキングを支援することで健康維持・増進を目的とした歩数計アプリケーション「歩いておトク」を利用し、ゲーミフィケーションによる外敵動機付けについて自己決定性の高低の種別により分析し考察した。分析では、同時に幾つかの動機付けを持つことを実証する理論的フレームワークとして自己決定理論 (Deci&Ryan, 1985) を利用し、自己決定性の高低の種類として1, 外発的動機づけの発動 2, 外敵調整 3, 取入的調整 4, 統合的調整 6, 内発的動機づけの観点から分析を行った。分析の結果、ゲーミフィケーション要素はユーザーの動機づけを目的とした行動変容を促すシステムに必要な動機づけのマネジメントの方向性を示してくれ、エンゲージメントを高め行動変容を促すものとした [11]。

2.2.3 技術者のモチベーションに関する研究

西康晴らの研究では、ソフトウェアプロジェクト管理にモチベーションという概念を利用し、マズローの欲求段階説に着目したモチベーションマップというツールを作成し、それらをプロジェクト管理に用いるための方策について議論した。ツール作成ではモチベーション向上のためのモデル作成のために、具体的なインセンティブの想起を行なった。29項目のインセンティブの列挙を行い、列挙されたインセンティブをマズローの欲求段階に従って、自己実現欲求に相当するインセンティブ・承認欲求に相当するインセンティブ・安全欲求や生理的欲求に相当するインセンティブの3つに整理した。整理したインセンティブから人間の心の中でどのように作用しているかを表すモデルを作成し、議論を行なった。外発的動機付けに関しては消極的なほど理性が感情を抑え込み、理性だけがモチベーションを向上させ、内発的動機付けに関しては具体的なほど外発的なインセンティブの要因が強く、概念的なほど理性や感情といった内発的インセンティブの要因が強くなることがまとめられた [12]。

2.2.4 ゲーミフィケーションにおける構成要素の有用性に関する研究

安政駿らの研究では、時間・レベル・ポイントなど情報の可視化がゲーミフィケーションの重要な本質の一つであると捉え、任意の作業に関してユーザの行動に伴う各種指標を数値化しゲーム性を付け加えることを可

*1 DeNa 社が開発した算数・国語・英語を学習できるタブレット端末上で動作するアプリケーション。

能にする視野拡張トレーニングシステムを実装した。具体的には、ブラウザ上で行う作業と連動し、その結果や進行具合、所要時間の数値化を行うことでゲーム性を加えるためのモジュールを開発した。また、このような数値化を行ったデータをゲーム性を持つ手法に取り込んだ時、手法自体にどの程度の効果が出るのかを検証した。検証の中では、ゲーミフィケーションの効果をあげたい機関によってユーザーのタスクに関するモチベーションの維持に関わる要因が変化すると考え、ゲーミフィケーションを応用する際に目標とする期間を即時的・短期的・中期的・長期的の四つに分類して実験を行った。ゲーミフィケーションの構成要素の有効性を検証していく上で、ユーザの行動基準の可視化が作業効果の向上につながることを示唆された [13]。

2.3 関連研究での課題点

2.2.2 の関連研究より、ゲーミフィケーションは「ゲーム要素」を取り入れることで感情的エンゲージメント*2[14] に訴えかけることで、人々の動機づけとエンゲージメントを高め行動変容を促すものであることが報告された。しかし、人間の動機づけは多種多様でありゲーミフィケーションを利用する際には、ユーザの動機づけや行動特性について十分に理解し、目的ごとにフィットするものをデザインする必要があることが課題点として挙げられた。

*2 興味、退屈、不安、楽しさといった学習者の感情的反応に関する概念。ここでは怒りや悲しみではなく感動、喜び、楽しさを表す。

第3章

ゲーミフィケーションを用いたプログラミング教材の提案

本章では、2.2.1節で挙げたゲーミフィケーション要素を用いた関連研究を元に、課題点の解決を図る本研究のシステム内容を定義する。

3.1 関連研究からの考察

2.2節,2.3節より、ゲーミフィケーションを用いる目的達成を図る場合は目的達成に適したゲーミフィケーション要素を選択し、システム作成を行う必要がある。ゲーミフィケーション要素の一つ一つがどのような効果を持っているのかを知ることはいままでもゲーム性を持っていなかったコンテンツにゲーム性を組み込む際の一つの指標になると考えられる。また、各ゲーミフィケーション要素における効果性を把握することによって、パフォーマンスを向上させたいときや複数回同じコンテンツを行わせたいときなど、仕掛ける側が目的とする状況に応じてどのような要素を取り入れることが最適なものが把握できる。

3.2 ゲーミフィケーション要素

本研究では、システム作成の際に使用するゲーミフィケーション要素に関連研究の福山佑樹らの研究 [10] で用いられているゲーミフィケーション要素を使用する。福山佑樹らの研究で使用されているゲーミフィケーション要素は、教育学習における先行研究の Dicheva ら [9] のゲーミフィケーション要素を使用しており、本研究でもプログラミング学習のモチベーションの維持・向上を目的とした研究であるためこちらのゲーミフィケーション要素を使用する。関連研究の福山佑樹らの研究で用いられているゲーミフィケーション要素と要素の内容を表3.1に示した。本研究では表3.1に示したゲーミフィケーション要素を使用する。

表 3.1 ゲームフィケーション要素一覧

デザイン原則区分	要素	要素の内容
	ゴールと挑戦 個別化 即時フィードバック 可視化されたステータス 随時追加されるコンテンツ 選択の自由 失敗の自由 ストーリーライン 時間制限 チュートリアル 社会的参加	各ステージにおけるクリアすべき課題が設定されている 個別に区分できるフィールドがある 常時可視化されたフィードバックがある 各ステージの成績などが示される コンテンツが随時追加される 学ぶステージやステージ内の解答を選べる 間違えても何度もやり直すことができる ストーリー制を持つ 解答に対して時間制限を設ける 使い方を示す コンテンツ外の社会的要因に関する
ゲームメカニズム区分		
	ポイント制 バッジ制 レベル制 リーダーボード アバター	成績をポイントで示す 成績やクリアでバッジを獲得できる クリアするとレベルが上がる ランキングでの成績の掲載 自身とアバターを重ねる

3.3 要素を使用する期間

本研究では、安政駿らの研究 [13] で使用されているゲームフィケーション要素を応用する際に目標とする期間の分類を使用する。安政駿らは、ゲームフィケーション要素を応用する際に目標とする期間によって即時的・短期的・中期的・長期的の4つに分類した。また、4つの期間の分類がゲームフィケーション要素を適用するには効果を上げたい期間によってユーザのタスクに対するモチベーションの維持に関わる要因が変化すると考えられ、ゲームフィケーションを応用する際に目標とする期間によって、即時的・短期的・中期的・長期的の4つに分類することとした。4つの期間の具体的な目的は以下の通りである。3.2節で示した表3.1のゲームフィケーション要素と4つに分類された期間を表3.2に対応させた。対応させる際には、時間制限が即時的なゲームフィケーション要素として入っているが、プログラミング言語を学ぶ際に時間制限の存在はモチベーション向上の逆効果であると考えたため実験の項目からは外す。また、ストーリーライン・チュートリアル・社会的参加は4つの分類に分けられない項目であると考えたため今回は除外する。

- 即時的：作業中の効果の向上が目的
- 短期的：1タスク内での効果の向上が目的
- 中期的：1か月程度の継続的な来訪が目的
- 長期的：一年以上モチベーションを向上させることを目的

表 3.2 要素と期間の対応表

適応期間	ゲーミフィケーション要素
即時的	即時フィードバック・失敗の自由・時間制限
短期的	可視化されたステータス・選択の自由
中期的	ポイント制・バッジ制・レベル制・個別化・ゴールと挑戦
長期的	アバター・随時追加されるコンテンツ・リーダーボード

3.4 使用する要素と手法の提案

関連研究により得られた知見を活用し、3.2 節で記述したゲーミフィケーション要素を取り入れた手法を提案する。本研究では、実験の際にはゲーミフィケーションを応用する際に目標とする期間の即時的・短期的に当てはまるゲーミフィケーション要素を使用する。そのため、本研究ではゲーミフィケーション手法は、何度もやり直して解答が可能・問題選択の自由・時間表示・結果表示を使用する。以下にゲーミフィケーション手法を表示する。また、3.1 節で表記したゲーミフィケーション要素とゲーミフィケーション手法の比較を表 3.2 に示した。

1. 達成度の可視化

学習者が問題に正解するとその正当数が達成度に追加されていき、達成度は常に見ることができる。

これにより、学習者は自身の現状の学習量を確認することができ、自身の学習の不測量を認識できる。

2. 達成度による称号の獲得

達成度がある程度の数値に到達すると、称号が獲得できる。

これにより、学習者は学習を進める動機づけがしやすくなる。

3. 何度もやり直して解答が可能

解答を何度間違えてもやり直すことができる。

これにより、学習者は反復学習を行うことができ、より深い学習の理解につながる。

4. 他の学習者の名前と達成度を提示する

自身の達成度を誰でも見ることができる。

これにより、学習者は競争心が湧き、学習を進める動機づけにつながる。

5. プロフィール設定が可能

名前やコメントの設定ができる。

これにより、学習者は他の学習者との違いを出すことができる。

6. ランキング制度

達成度の違いからうまれるランキングを確認できる。

これにより、学習者は競争心が湧き、他の学習者よりもを行うようになるという動機づけにつながる。

7. 時間制限が含まれている課題の搭載
解答時間に制限を設けた問題を解くことができる。
これにより、学習者は一つの問題に集中して取り組むことができる。
8. 自身のプロフィールにアバターの付与
キャラクターを自由に選び、アバター指定できる。
これにより、学習者は自身のプロフィールの変化を楽しむことができる。
9. 講義後に講義内容の問題追加
新しい内容の問題が講義ごとに追加される。
これにより、学習者は学習に飽きることなく進めることができる。
10. 問題選択の自由
どの問題も自由に選択することができる。
これにより、学習者は特定の問題につまづいても別の問題を進めることができる。
11. 時間表示
作業における時間の把握ができる。
これにより、学習者は自身がどれくらいの時間の間学習したかわかり学習を続けやすくなる。
12. 結果表示
作業を行った結果をそのままの形で把握できる。
これにより、学習者は次の作業に対する行動基準・学習基準を持つことができる。

表 3.3 手法と要素の比較一覧

	ゲーミフィケーション手法	ゲーミフィケーション要素
1	達成度の可視化	ポイント制・レベル制
2	達成度による称号の獲得	ゴールと挑戦・バッジ制
3	何度もやり直して解答が可能	失敗の自由
4	他の学習者の名前と達成度を提示する	リーダーボード
5	プロフィール設定が可能	個別化・アバター
6	ランキング制度	リーダーボード・ゴールと挑戦
7	時間制限が含まれている課題の搭載	時間制限
8	自身のプロフィールにアバターの付与	アバター・個別化
9	講義後に講義内容の問題追加	随時追加されるコンテンツ
10	問題選択の自由	選択の自由
11	時間表示	即時フィードバック
12	結果表示	可視化されたステータス

3.5 システム提案

本研究で作成するシステムは、学習者が Web サイト上でクイズ形式学習機能とコード記入型学習機能を利用しプログラミングの自学習を行うものとする（図 3.1）。クイズ形式学習機能では、プログラミング基礎を Web サイト上でクイズ形式で学習することができ、自身でプログラミング基礎に関するクイズの作成も可能である。コード記入型学習機能では、学習者が Web サイト上でプログラムのコードを書いて問題を解くことが可能である。

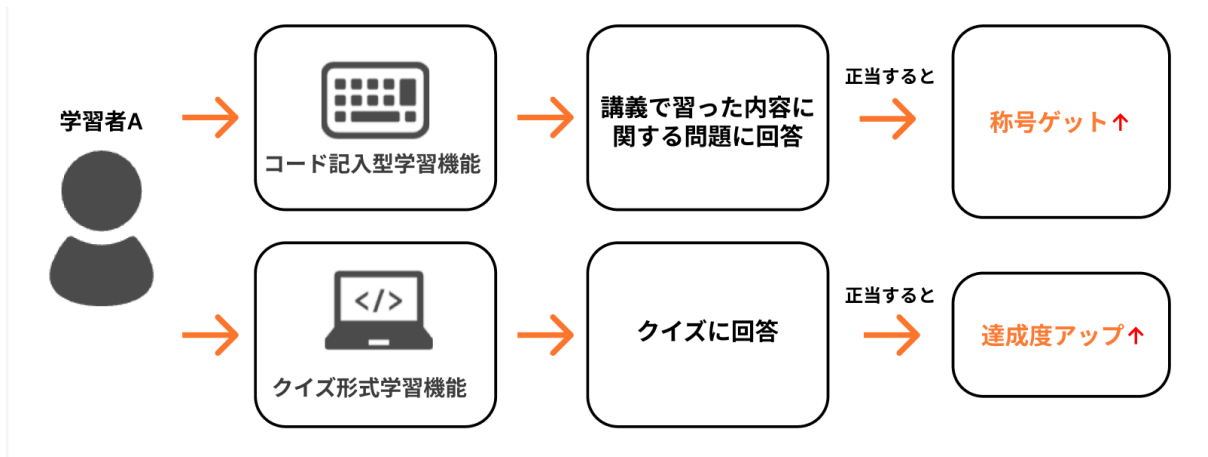


図 3.1 システム全体概要図

第4章

システム設計

本章では、第3章で挙げたゲーミフィケーション手法の説明に先立ち、システムをクイズ形式学習機能とコード記入型学習機能の2つに分類する。

4.1 本システム全体設計

本システムは、Webアプリケーション上で動かす。Webアプリケーション上にはクイズ形式学習機能の画面とコード記入型学習機能ごとに画面を用意する。また、2つの機能にはそれぞれの機能に合わせたWebアプリケーション作成ではRuby on Railsを利用して機能を増やしていく。データの保存では、sqlite3を利用する。3.4節より、本研究ではゲーミフィケーション手法の、何度もやり直して解答が可能・問題選択の自由・時間表示・結果表示を使用して実験を行う。しかし、システム上にはランキング設定やプロフィール設定など他のゲーミフィケーション手法も実装し、システムの実用化も行えるように設計する。システムの中で実装した本研究実験で使用しないゲーミフィケーション手法は、開発者側で実験の際には取り外せるようにする。

4.2 システム選択画面の設計

システム選択画面ではランキング制度、プロフィール設定、アバターの付与、達成度の可視化が行えるようにした。また、システム選択のUI画面は図4.1に示した。



図 4.1 システム選択 UI 画面

4.2.1 システム選択画面使用手法

本機能画面で利用したゲーミフィケーション手法を以下に示した。

- 達成の可視化
- 他の学習者の名前と達成度を提示する
- プロフィール設定が可能
- ランキング制度
- 自身のプロフィールにアバターの付与
- 問題選択の自由

4.3 クイズ形式学習機能画面の設計

クイズ形式学習機能は、プログラミング言語である Ruby を学習内容の主体としたクイズを学習者が解いていく機能である。クイズは全てで 30 問用意し、実験のためにクイズの選択を自由にできる機能とランダムでクイズが表示されていく機能を作成する。本機能のフローは図 4.2 に示した。

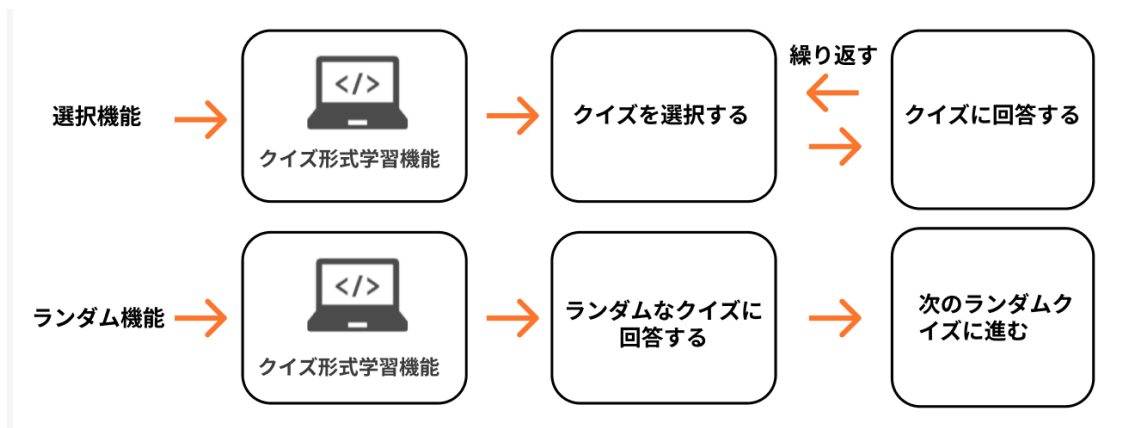


図 4.2 クイズ形式学習機能概要図

4.3.1 クイズ形式学習機能使用手法

本機能画面で利用したゲーミフィケーション手法を以下に示した。

- 達成の可視化
- 何度もやり直して回答が可能
- 問題選択の自由

4.4 コード記入型学習機能画面の設計

コード記入型学習機能は、プログラミング言語である Ruby を学習内容の主体として学習者自らがサイト上でプログラミングのコードを打ち、課題に取り組んでいく機能である。学習者の自学習の際にトライ&エラーを何度も実行する演習を行なっていくことでプログラミング言語の学習の深い理解を目的としている。実験のためにコードの記入の後に回答を修正できる機能と修正できない機能を作成する。本機能のフローは図 4.3 に表示した。

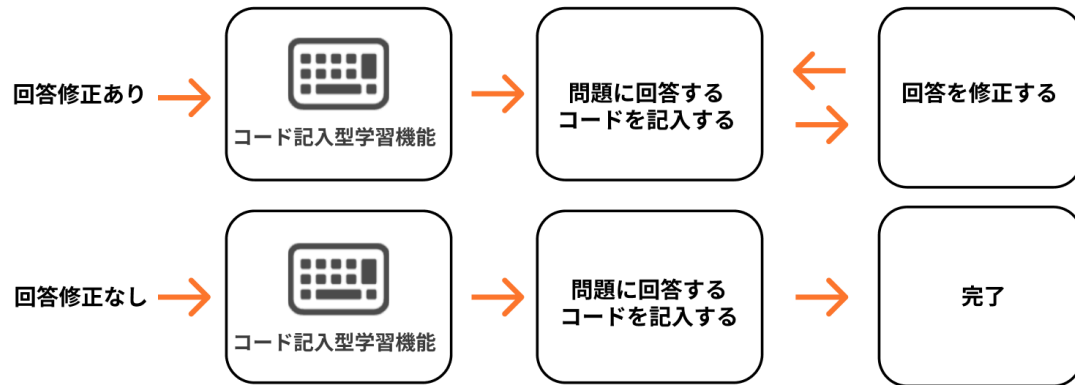


図 4.3 コード記入型学習機能概要図

4.4.1 コード記入型学習機能使用要素

本機能画面で利用したゲーミフィケーション手法を以下に示した。

- 達成の可視化
- 何度もやり直して回答が可能
- 問題選択の自由

第5章

システム開発

本章では、クイズ形式学習機能とコード記入型学習機能のシステムの開発を行う。

5.1 システム全体の開発

本システムはを Rbuy on Rails を使用して機能の作成を行なっていく。

5.2 開発環境

開発環境を以下に示す。

- サーバーサイド
 - Ruby(version 3.1.2)
- クライアントサイド
 - HTML(HyperText Markup Language)
 - CSS3
 - JavaScript(ES2022)
- フレームワーク
 - Ruby on Rails(version 7.0.3)
- データベース
 - sqlite3 - 3.38.2

5.3 クイズ形式学習機能

本節では、クイズ形式学習機能の画面とそれに含まれる機能を説明する。

5.3.1 クイズ形式学習機能の概要

クイズ形式学習機能は、プログラミング言語である Ruby 言語についてのクイズを 30 問用意し、そのクイズに学習者自身が本研究システム上で回答していくという機能である。クイズ形式学習機能には、実験のために

クイズの選択を自由にできる機能とランダムでクイズが表示されていく機能の二つを作成した。また、クイズ回答後の成績発表も付けるか付けないか選べるように作成した。

5.3.2 クイズの内容

クイズの問題は30問用意した。問題ないようには「基礎プログラミングI・II」で使用している東北公益文化大学教授が作成したWebページ参考書から抜粋したものを使用する。以下は実際のクイズ形式学習機能から抜粋した問題である。

- print() の処理方法は？
 - － () 内の文字列を出力できる
 - － () 内の文字列を出力して、最後に改行が追加される
 - － キーボードで打ち込んだ文字を取り出す
- gets の処理方法は？
 - － キーボードで打ち込んだ文字を取り出す
 - － コメント文にする
 - － 対応する値を整数の文字列に置き換える
- redo の処理内容は？
 - － 処理を中止して、ループを終わらせる
 - － 条件を判断せず、処理を最初からやり直す
 - － 処理をスキップして、次のループに移る
- for の処理方法は？
 - － 配列や範囲などの複数の間を順に変数に代入する
 - － 配列の要素を小さい順に並べ替えた結果を返す
 - － 配列の末尾に要素を追加する

5.3.3 クイズ形式学習機能の開発

クイズ形式学習機能の開発には、Ruby on Rails と JavaScript を使用した。本研究システムでは JavaScript でローカルストレージを利用して実験に使用するページ間共有ができるタイマーなどの機能を掲載したためクイズシステムも JavaScript での開発としている。



図 5.1 クイズ出題画面



図 5.2 クイズ終了画面

5.4 コード記入型学習機能

本節では、コード記入型学習機能の画面とそれに含まれる機能を説明する。

5.4.1 コード記入型学習機能の概要

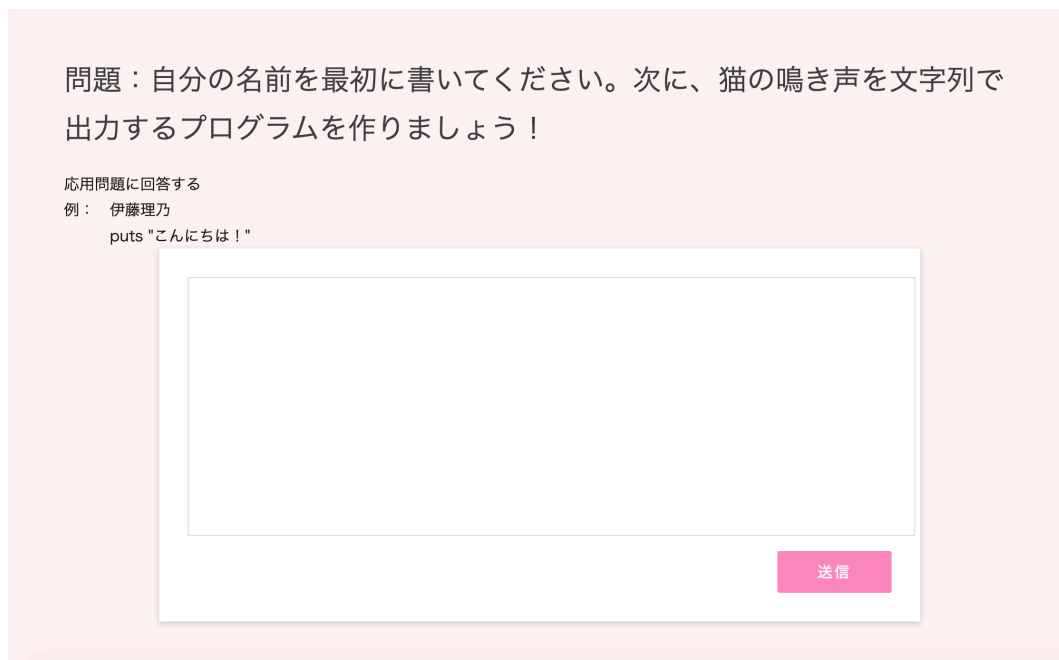
コード記入型学習機能は、プログラミング言語である Ruby 言語に関するコード記述問題を解いていく機能である。本機能では、実験のためにコードの記入の後に回答を修正できる機能と修正できない機能を作成する。

5.4.2 学習問題の内容

問題内容は、東北公益文化大学が「基礎プログラミング I・II」で使用している教授が作成した Web ページ参考書の中にある問題を少し変えて出題する。実際のコード記入型学習機能から抜粋した問題の例には「猫の鳴き声を文字列で出力するプログラムを作ってください。」「配列 like の中に自分の好きなものを値として3つ入れて、文章が printf で表示されるようにプログラムを作ってください。」がある。

5.4.3 コード記入型学習機能の開発

クイズ形式学習機能の開発には、Ruby on Rails を使用した。クイズ形式学習機能に関する情報は全てデータベースに保存する。問題に対する回答に関しては、1つの問題に対して複数の人が回答するため、回答を保存して置けるようにテーブルを作った。また、クイズ形式学習機能ではコードの記入の後に回答を修正できる機能と修正できない機能の2種類を作るため、回答を保存・修正・削除できるようにした。



問題：自分の名前を最初に書いてください。次に、猫の鳴き声を文字列で出力するプログラムを作りましょう！

応用問題に回答する
例：伊藤理乃
puts "こんにちは！"

送信

図 5.3 コード記入型問題出題画面



図 5.4 みんなの回答表示画面

5.5 予備実験

本研究では、予備実験として東北公益文科大学が実施している小学 5 年生から中学 3 年生を対象としたプログラミング学習を目的とした塾である「ジュニアドクター鳥海塾」の塾生 38 名に本システムを使用してもらいその効果を考察してもらい、その意見をシステム改良に取り入れた。「ジュニアドクター鳥海塾」では Ruby 言語を学習の主体としており、毎週土曜日の本講義と週 3 回 19:00-21:00 の補講でプログラミングを教えている。本システムはプログラミングを学んでいる学習者に向けたシステムであるため、実験時にプログラミングの学習頻度のアンケートも取り入れ被験者のプログラミング学習頻度も調べる。実験では、システム 5 分間を利用してもらった後にアンケート記入をしてもらい 38 名のシステム利用者のうち 32 名がアンケート記入を行った。

5.5.1 予備実験結果

アンケートの結果、1 日に家でプログラミングの勉強を 30 分以上するのが 75%、1 週間に家でプログラミング勉強をしている人が 65.6% いた。そのため、今回の実験被験者として相応しいという結論になった。またアンケートの結果、システムの実証が確認された。システムの感想として肯定的な意見では「自分も負けないように諦めないで取り組めたから。」「応用問題皆の回答一覧で皆がどんな答えをしているのかが見えることが良い」「自分のスコアが表示されてあって毎日アプリを開きたくなるから良い。」という意見があった。改善点を挙げた意見には、「なんとなく楽しさが足りないと思う。」「回答するときを選択するやり方が分かりづかった。」「回答の選択肢が同じ並びだったからそこもランダムにしたらいいと思う。」という意見があった。アンケート項目と結果は以下の通りである。

1日に家でプログラミングの勉強をする時間を教えてください。
32件の回答

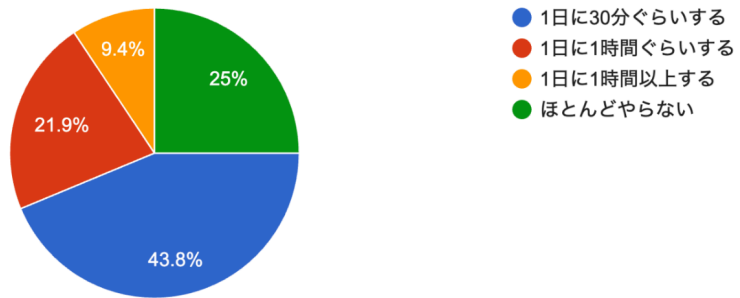


図 5.5 1日のプログラミング勉強時間グラフ

1週間に家でどれくらいプログラミングの勉強をしますか？
32件の回答

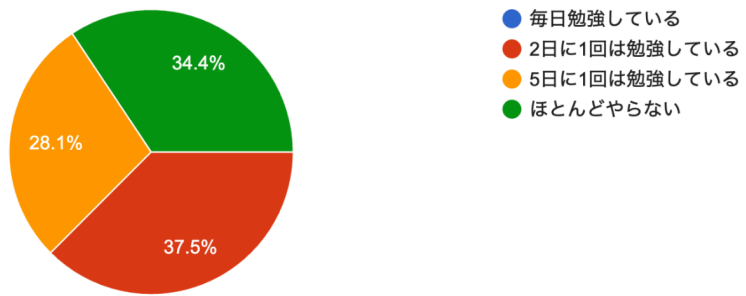


図 5.6 1週間の家でのプログラミング勉強時間グラフ

このシステムで任意のプログラミングの宿題が出たらやりますか？
32件の回答

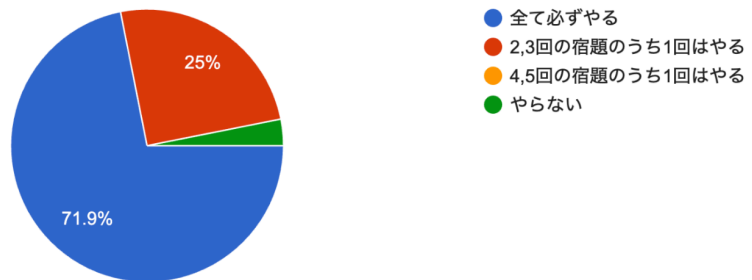


図 5.7 宿題提出グラフ

システムの感想を教えてください。
32件の回答

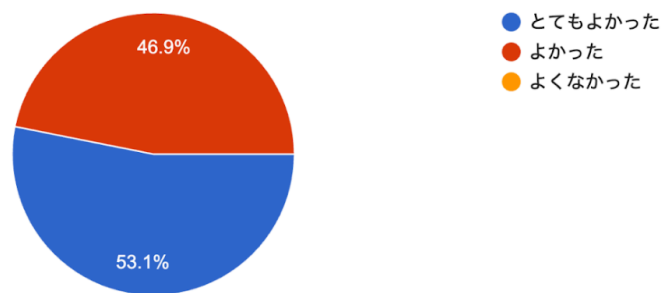


図 5.8 システム感想

5.5.2 予備実験からのシステム改良

「ジュニアドクター鳥海塾」での予備実験を経て、考察とシステム改良を行った。改善点の意見には、「なんとなく楽しさが足りないと思う。」「回答の選択肢が同じ並びだったからそこもランダムにしたらいいと思う。」という意見があったがそれらの意見を踏まえ、クイズの問題数を増やしランダムにした。また、「回答するときを選択するやり方が分かりづらかった。」という意見を反映し、画面のデザインをわかりやすくした。

第6章

システムの実験・検証

本章では、本システムを利用し実験・検証を行う。

6.1 実験・検証概要

本研究では、二つの検証・実験を行った。一つ目は「基礎プログラミングⅠ・Ⅱ」で本システムとゲーム要素のない教材の両方を使用してもらいその効果を図る実験、二つ目は自身が所属している研究室の大学生9名に本システムを使用してもらいゲーミフィケーション要素の効果を図る実験である。

6.2 ゲーミフィケーション要素の有無の実験

本実験では、東北公益文科大学在学の学生10名を対象に実験を行った。実験の際には、実験参加者を二つのグループに分け、1つのグループにはゲーミフィケーション要素が組み込まれたプログラミング問題を提供し、もう1つのグループにはゲーミフィケーション要素が組み込まれていないプログラミング問題を提供する。今回の実験では、ゲーミフィケーション要素が組み込まれたプログラミング問題には本研究システムを使用し、ゲーミフィケーション要素が組み込まれていないプログラミング問題には自身で作った紙のクイズ問題と紙のコード記入問題を用意した。実験の際に4つのグループは必ず一度はシステムと紙の教材を使用してもらう。その後、アンケート調査をもとにモチベーションの評価にはどう影響するかを調べた。

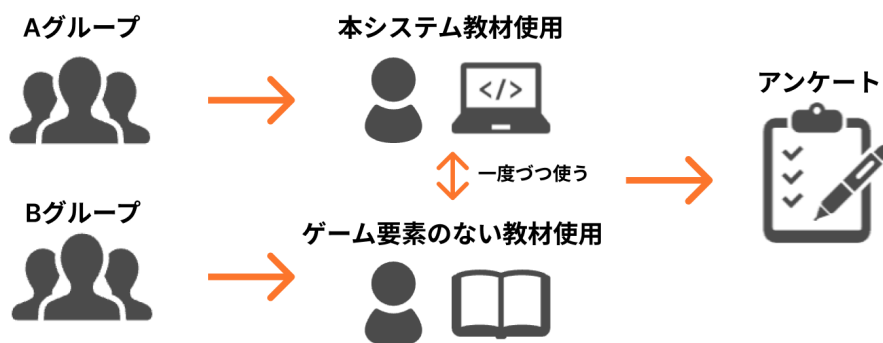


図 6.1 実験内容概要図

6.2.1 実験使用資料

本実験で使用したゲーミフィケーション要素のない教材は図 6.2 と図 6.3 である。本システムとゲーミフィケーション要素のない教材でコードを実際を書く問題内容とクイズの問題内容は同じにしている。

図 6.2 手法と要素の比較一覧

これはRubyに関する問題です。
本クイズは、広瀬ゼミ4年伊藤理乃が卒業研究で利用するものです。
本クイズで回答いただいた個人情報は利用しません。学籍番号
また、回答内容は基礎プログラミングの成績とは全く関係ありません。名前

問題: 配列likeの中に自分の好きなものを値として入れて、`printf`で配列の中身を表示させてください。

回答場所

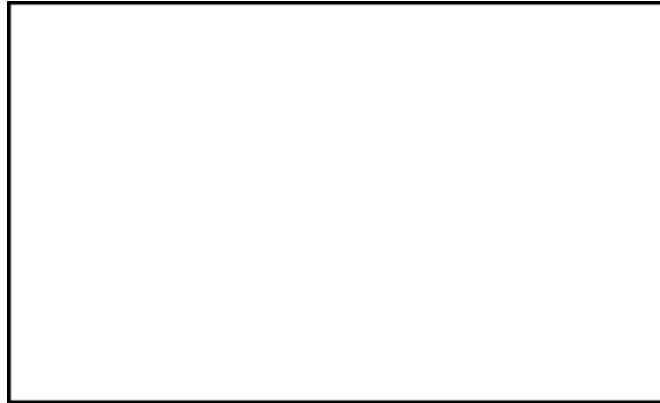


図 6.3 手法と要素の比較一覧

これはRubyに関するクイズです。
 本クイズは、広瀬ゼミ4年伊藤理乃が卒業研究で利用するものです。
 本クイズで回答いただいた個人情報は利用しません。
 また、クイズ内容は基礎プログラミングの成績とは全く関係ありません。

学籍番号
名前

問1 print()の処理方法は？
 ①()内の文字列を出力できる
 ②()内の文字列を出力して、最後に改行が追加される
 ③キーボードで打ち込んだ文字を取り出す

問2 getsの処理方法は？
 ①キーボードで打ち込んだ文字を取り出す
 ②コメント文にする
 ③対応する値を整数の文字列に置き換える

問3 %d、%f、%sが使える出力形式は？
 ①print
 ②gets
 ③printf

問4 対応する値を文字列に置き換えるのはどれ？
 ①%d
 ②%s
 ③%f

問5「#」を使った時はどのように処理が行われる？
 ①改行される
 ②掛け算される
 ③コメントアウトされる

問6 プログラムの一部を実行させなかったり、同じ処理を何回も繰り返したりすることができるものは？
 ①制御構造
 ②破壊的操作
 ③乱数

問7 redoの処理内容は？
 ①処理を中止して、ループを終わらせる
 ②条件を判断せず、処理を最初からやり直す
 ③処理をスキップして、次のループに移る

問8 forの処理方法は？
 ①配列や範囲などの複数の間を順に変数に代入する
 ②配列の要素を小さい順に並べ替えた結果を返す
 ③配列の末尾に要素を追加する

問9 変数とは何か？
 ①値につける名前
 ②複数の値をひとまとめにできるもの
 ③数字の中からランダムに発生させる数

6.2.2 実験結果

実験のデータ分析はカイ二乗検定^{*1}を使用した。本実験では、帰無仮説を「システム問題と紙問題の違いはやる気が出る・出ないに関係する」として対立仮説を「システム問題と紙問題の違いはやる気が出る・出ないに関係しない」として有意水準は5%として実験を行っていく。クイズ問題とコード問題はどちらも自由度 = 1 となり、カイ二乗分布は「3.84」である。棄却ルールとしてカイ二乗値が「3.84」を上回れば、帰無仮説は棄却されることとする。クイズ問題の方はカイ二乗値 = 3.8095, p 値 = 0.05096 であり、コード記述問題の方はカイ二乗値 = 10.769, p 値 = 0.001032 となった。そのため、クイズ問題ではカイ二乗値 (=3.8095) がカイ二

^{*1} カテゴリカルなデータの頻度分布を比較し、グループ間の統計的な関連性を評価するための統計

乗分布 (=3.84) を下回っているため帰無仮説は棄却されず、「システム問題と紙問題の違いはやる気が出る・出ないに関係しない」と判定された。コード記述問題ではカイ二乗値 (=10.769) がカイ二乗分布 (=3.84) を上回っているため帰無仮説は棄却され、「システム問題と紙問題の違いはやる気が出る・出ないに関係する」と判定された。

表 6.1 クイズ問題

	やる気が出た	やる気が出なかった	合計 (人)
システム問題	9	1	10
紙問題	5	5	10

表 6.2 コード記述問題

	やる気が出た	やる気が出なかった	合計 (人)
システム問題	10	0	10
紙問題	3	7	10

6.3 ゲーミフィケーション要素の効果の検証

本実験は、東北公益文科大学在学の学生 9 名を対象に実験を行った。本実験では、本研究で作成したシステムを使用して「1 タスク内の作業回数」と「作業時間」の二つの指標を作成することで検証を行う。また、アンケートの結果から各ゲーミフィケーション要素の有無によるプログラミング学習への取り組み方の違いを明らかにすることによって、有効性の検証を行う。本実験で評価するゲーミフィケーション要素は即時的・短期的・中期的・長期的の 4 つの分類から即時的・短期的の二つの分類に当てはまるゲーミフィケーション要素を使用する。実験で使用するゲーミフィケーション要素は表 6.3 に示しており、即時的・短期的の二つの分類に当てはまるゲーミフィケーション要素は太文字で表示されている。実験では、今回使用するゲーミフィケーション要素を一つずつ組み合わせたものを使用して実験していくため 4 回使用してもらう。その後、毎回「効果があるか」「続けていきたいか」「興味を持ったか」についてのアンケートに記入してもらう。アンケートは 4 個のゲーミフィケーション要素の実験に合った内容のものを 4 個作成した。アンケート例は以下の通りである。

表 6.3 要素と期間の対応表

	ゲーミフィケーション要素	運営側の構成要素
即時的	即時フィードバック (継続時間の表示)・失敗の自由	
短期的	可視化されたステータス・選択の自由	ゴールと挑戦 (課題設定)
中期的	ポイント制・バッジ制・レベル制・個別化	
長期的	アバター・随時追加されるコンテンツ	

6.3.1 検証結果

時間はかかった作業時間の平均値、継続回数はタスク内の作業数の平均値を示す。作業時間では小数点以下の秒数は切り捨てて平均値を出す。これらの結果をグラフとして示したものが図 6.4 と図 6.5 である。図 6.4 より、作業中に時間を表示すること・ステータスを表示することによって作業時間に変化が起きることがわかる。しかし、図 6.5 より作業中に時間を表示すると作業時間は伸びるが継続回数は 3.44 回と他より低いことがわかった。これは時間表示によりコード記入型学習機能のプログラム作成に時間をかけるようになりタスク数が減少したことが要因として考えられる。タスク数としては、問題に失敗しても何度でも挑戦できること・問題がランダムでなく自由に選択できることによって継続率が 4.33 回、4.22 回と高い継続率が出ている。また、図 6.6 は実験後に行ったアンケートの結果である。全体的に高い数値の結果だが、どのアンケート内容でも時間表示は評価が低くステータス表示は評価が高いことがわかる。

表 6.4 ゲーミフィケーション要素ごとの評価結果

	作業時間 (分)	継続回数 (回)
継続時間の表示	3.9	3.44
失敗の自由	2.9	4.33
ステータスの表示	3.68	4
選択の自由	3	4.22

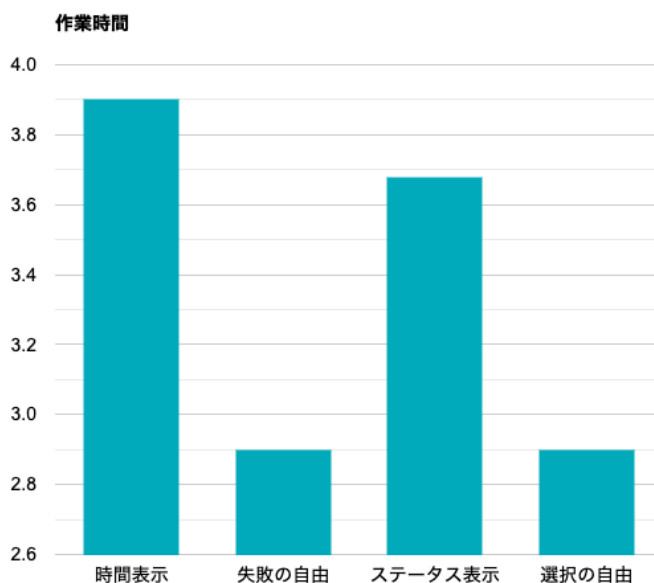


図 6.4 作業時間のグラフ

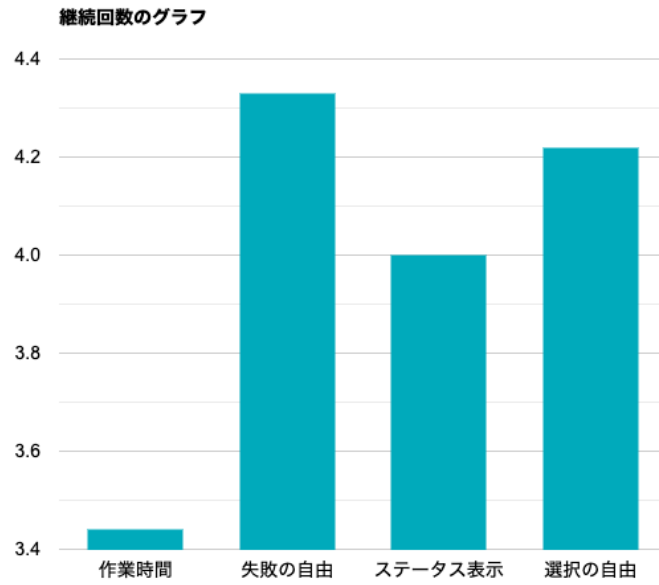
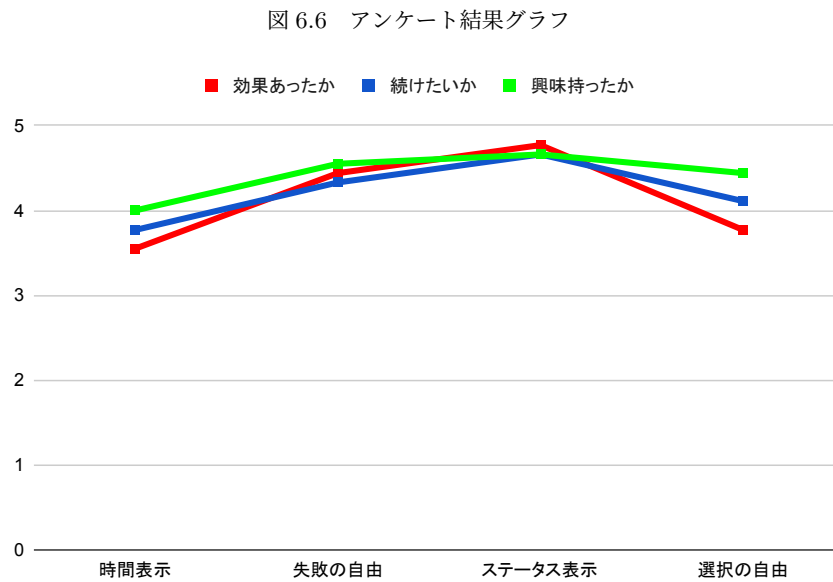


図 6.5 継続回数のグラフ



第7章

結論

本研究で開発したシステムによって、プログラム学習者のプログラミング学習のモチベーションの維持・向上を推進していくことを可能とした。しかし、システムに掲載したコード記入型学習機能とクイズ形式学習機能の二つにはプログラミング学習のモチベーションの維持・向上に違いが生まれた。コード記入型学習機能はゲーミフィケーション要素がないプログラミング問題学習よりも大きく効果が出たが、クイズ形式学習機能は違いがあまりないと言う結果だった。また、本システムに搭載した各ゲーミフィケーション要素の効果については時間表示とステータス表示が作業中の時間に関わる。また、問題に失敗しても何度でも挑戦できること・問題がランダムでなく自由に選択できることが継続回数に関わる。この結果から、システム内でコード記入型学習機能はプログラミング学習のモチベーションの維持・向上を推進していけるがクイズ形式学習機能はゲーミフィケーション要素がないものとモチベーションの維持・向上を推進については差がないことがわかる。しかし、本システム全体の評価としては「プログラム学習者のプログラミング学習のモチベーションの維持・向上を推進していける」を達成するに値すると評価する。

謝辞

本研究の遂行にあたり、指導教官として終始多大なご指導を賜った、広瀬雄二教授には、本研究の技術指導や論文の執筆にあたり多大なご協力を賜りました。ここに深謝の意を表します。並びにジュニアドクター鳥海塾の塾生の皆様と広瀬研究室の皆様には、本研究の遂行にあたり多大なご助言、ご協力頂きました。ここに感謝の意を表します。

参考文献

- [1] 内閣府.”Society 5.0”.https://www8.cao.go.jp/cstp/society5_0/,(参照 2022-7-05) .
- [2] 文部科学省.”小学校プログラミング教育の手引 (第三版)”.https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf,(参照 2022-7-05) .
- [3] 堀口悟史, 井垣宏, 井上亮文, 山田誠, 星徹, 岡田謙一. ”講義資料閲覧ログを用いたプログラミング講義進捗管理手法の提案”. 情報処理学会論文誌, 2012, No.1, p.61-71, <http://id.nii.ac.jp/1001/00079945/>.
- [4] MIT.”SCRATCH”.<https://scratch.mit.edu/>,(参照 2022-11-20) .
- [5] 岩本穰志. “日本語授業における、ゲーミフィケーションを用いた学習意欲向上の試み”. 日本語教育方法研究会誌, 2017, No.1, p.8-9, https://doi.org/10.19022/jlem.24.1_8.
- [6] 藪内一葉, 平本督太郎. “SDGs ゲーミフィケーション教材を用いたプロソーシャルモチベーション向上のための手法に関する研究”. Beyond SDGs イノベーション研究, 2022, No.3, p.14-28, https://doi.org/10.51097/bsir.3.3_14.
- [7] 忍者 CODE.”あなたのコードが未来を灯す”.<https://ninjacode.work/>,(参照 2022-12-5) .
- [8] Progate.”Progate | プログラミングの入門なら基礎から学べる Progate[プロゲート]”.<https://prog-8.com/dashboard>,(参照 2022-12-5) .
- [9] *Dariana Dicheva, Christo Dichev, Gennady Agreč Galia Angelova.* “Gamification in Education:A Systematic Mapping Study”. https://www.researchgate.net/publication/270273830_Gamification_in_Education_A_Systematic_Mapping_Study, (参照 2022-11-15) .
- [10] 福山佑樹, 床鍋佳枝, 森田祐介. “ゲーミフィケーション要素を取り入れた小学校1年生向け電子教材の実践と評価”. デジタルゲーム学研究, 2017, No.9, p.31-40, https://doi.org/10.9762/digraj.9.2_31.
- [11] 藤田美幸, 塚本麻紀. “ゲーミフィケーションを活用したモバイル・ヘルスケアサービス：ドコモ・ヘルスケア「歩いておトク」を事例として”. 日本情報経営学会誌, 2018, No.3, p.74-82, https://doi.org/10.20627/jsim.38.3_74.
- [12] 西康晴, 片山豊, 八木広行, 高橋宗雄, 菅田直美, 中條貴幸. “ソフトウェアプロジェクト管理における技術者のモチベーションに関する研究”. プロジェクトマネジメント学会誌, 2001, No.3, p.33-39, https://doi.org/10.14914/spmj.3.3_33.
- [13] 安政駿, 手塚太郎. “ゲーミフィケーションにおける構成要素の有用性の検証”. プロジェクトマネジメント学会誌, 2001, No.3, p.33-39, https://doi.org/10.14914/spmj.3.3_33.
- [14] 梅本貴豊, 伊藤崇達, 田中健史朗. “調整方略, 感情的および行動的エンゲージメント, 学業成果の関連”. 心理学研究, 2016, No.87, p.334-342, <https://doi.org/10.4992/jjpsy.87.15020>.
- [15] 鈴木啓真, 兼子正勝. “WebSocket を用いたリアルタイムな Web デスクトップ共有”. 情報処理学会第 77 回全国大会講演論文集, 2015, No.1, p.153-154, <http://id.nii.ac.jp/1001/00165177/>.