

# 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 3年伊藤理乃

令和4年1月11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, 継続的なプログラミング学習を行うことを目的としたゲーミフィケーション要素を取り入れたプログラミング教材の提案を行う。

# 目次

<b>第 1 章</b>	<b>はじめに</b>	<b>5</b>
1.1	背景	5
1.2	ジュニアドクター鳥海塾	6
1.2.1	受講生の学習の取り組み	6
1.2.2	受講生の活動時間と理解度	6
1.3	研究目的	7
<b>第 2 章</b>	<b>関連研究と類似サービス</b>	<b>9</b>
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
<b>第 3 章</b>	<b>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</b>	<b>13</b>
3.1	ゲーミフィケーション要素	13
3.2	使用する要素と手法の提案	14
<b>第 4 章</b>	<b>システム設計</b>	<b>17</b>
4.1	システム設計の前提	17
4.2	システム全体の設計	17
4.3	システム選択画面の設計	18
4.4	クイズ形式学習機能画面の設計	19
4.4.1	コード記入型学習機能画面の設計	20
4.4.2	データベースの設計	21
<b>第 5 章</b>	<b>システム開発</b>	<b>23</b>
5.1	開発言語	23
5.2	システム全体の開発	23
5.3	クイズ形式学習機能の開発	23
5.4	コード記入型学習機能の開発	25

---

第 6 章	ゲーミフィケーションの適用実験	27
6.1	研究対象 . . . . .	27
第 7 章	結論	29
7.1	考察 . . . . .	29
7.2	今後の展望 . . . . .	29
参考文献		31

# 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 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経験させながら、「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、講義時間外の自学習による反復学習が必要である。しかし講義時間外の自学習において、プログラミングを勉強し始めたばかりの学習者やプログラミングに苦手意識を持っている学習者はエラーやバグの解決に時間を取られてしまい、学習意欲が減少してしまう。エラーやバグの解決に時間を取られない手法として、Scratch<sup>\*1</sup>[3] 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てい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 [4] や SDGs[5] 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他の学習者との情報交換・共有を行うことができるクイズ形式学習機能を実装する。

---

<sup>\*1</sup> 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 ジュニアドクター鳥海塾

「ジュニアドクター鳥海塾」とは、地方では学習機会の少ない理数・情報分野に関心の高い児童に、「ジュニアドクター鳥海塾」を提供し、「地域社会」と「情報技術」について、それぞれの研究を深め、地域の魅力を自ら発信し、地域と世界の橋渡しができる人材育成を行う活動である。なお、この活動は、JST(国立研究開発法人科学技術振興機構)の「ジュニアドクター育成塾」事業に採択されているものである。活動の中で、塾生はプログラミングの基礎を学び情報擬実が他分野でどのように生かされているかを知る。そして身につけた技術を使って課題解決プログラムを作成し発表する。

### 1.2.1 受講生の学習の取り組み

令和4年度ジュニアドクター鳥海塾では、40名の塾生が受講している。学生一人一人に一台の計算機を用意し、塾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。塾生は講義の中で、変数や入出力処理、制御構造、配列、メソッド定義を学習する。また、地域文化、交通心理、天文学、宇宙科学の講義を受け、さまざまな分野の情報技術との関連を学ぶ。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、研究成果発表を行う。

### 1.2.2 受講生の活動時間と理解度

「ジュニアドクター鳥海塾」では、開塾式から閉塾式を含めた全22回の活動でプログラミング基礎から情報技術との関連を学ぶ。塾生は、第一回目の講義(PC操作の基本)から7回の講義内容を元に中間成果発表を行う。中間成果発表からは8回の講義を行った後に個人成果発表を行う。その後、2回の講義を経て最終成果発表を行う。講義の際の塾生の活動プロセスを図1.1に示した。2時間の講義を行った後、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、最終的に塾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塾生と講義内容を講義中に理解できた塾生の双方において自学習は必要不可欠である。講義時間は塾生全員平等なため、塾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、学習意欲が低い塾生は学習を諦めてしまうことが起こりうる。

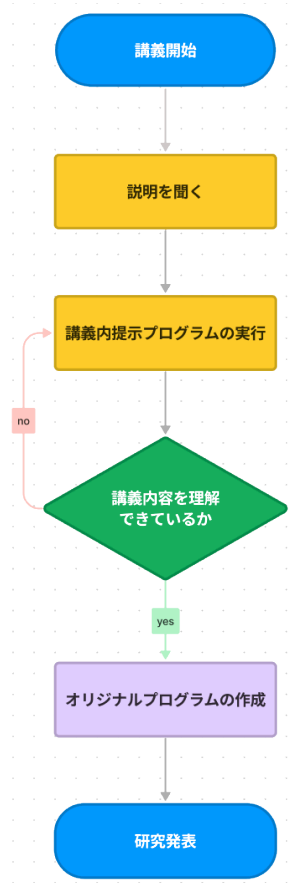


図 1.1 塾生活動プロセス

### 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 [6]。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 [7]。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

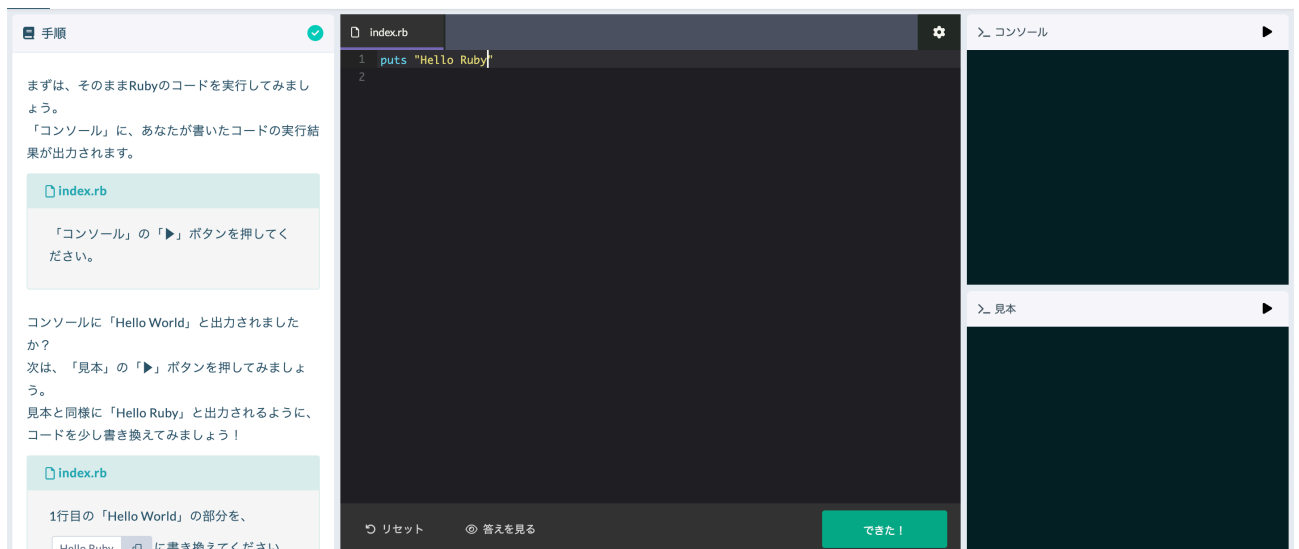


図 2.2 Progate 実行画面

## 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 [8] 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 [9]。

### 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 [10]。

\*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。



## 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、第2章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 ゲーミフィケーション要素

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を表3.1に示した。

表3.1 ゲーミ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	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
	個別化	個別に区分できるフィールドがある
	即時フィードバック	答えに対し即時に可視化されたフィードバックを獲得できる
	可視化されたステータス	各ステージの成績などが示される
	随時追加されるコンテンツ	コンテンツが随時追加される
	選択の自由	学ぶステージやステージ内の解答を選べる
	失敗の自由	間違えても何度もやり直すことができる
	ストーリーライン	ストーリー制を持つ
	時間制限	解答に対して時間制限を設ける
	チュートリアル	使い方を示す
	社会的参加	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	要素	要素の内容
	ポイント制	成績をポイントで示す
	バッジ制	成績やクリアでバッジを獲得できる
	レベル制	クリアするとレベルが上がる
	リーダーボード	ランキングでの成績の掲載
	アバター	自身とアバターを重ねる

## 3.2 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、ゲームフィケーション要素を取り入れた手法を提案する。本研究で利用するゲームフィケーション手法を以下に示す。また、3.1節で表記したゲームフィケーション要素と本システムで使用するゲームフィケーション手法の比較を表3.2に示した。3.1節で示したゲームフィケーション要素のうち、ストーリーライン・チュートリアル・社会的参加のみゲームフィケーション手法に取り入れていない。

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。
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。
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。
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。
5. プロフィール設定が可能  
名前の設定ができる。
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。
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。
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。
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。
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。

表 3.2 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	可視化されたステータス・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・可視化されたステータス
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





## 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 システム設計の前提

本研究で作成するクイズ形式学習機能とコード記入型学習機能の2つを設計する際に使用する前提を以下に示す。

- 前提1 本システムは「ジュニアドクター鳥海塾」塾生40名を対象とする
- 前提2 使用する塾生と「ジュニアドクター鳥海塾」メンター・指導教員のみがアクセスできる
- 前提3 運用は講義以外の自学習のみとする

### 4.2 システム全体の設計

4.1節の前提を元にシステムを設計する。システム上では、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。図4.1に本システムのフローを表示した。学習の際に利用者は、コード記入型学習機能とクイズ形式学習機能の2種類から選択する。コード記入型学習機能を選択した場合、応用問題と基本問題の2種類から選び、基本問題を選んだ場合には基本問題の中からも分野別に選ぶことが可能である。また、クイズ形式学習機能を選択した場合、基本のクイズと他の学習者が作成したクイズの中から選ぶことができ、自身でクイズを作成することもできる。クイズ形式学習機能を選択した場合、基本のクイズと他の学習者が作成したクイズの中から選ぶことができ、自身でクイズを作成することもできる。自身で作成したクイズを他の学習者に提供することで、共に学んでいく学習をクイズ形式学習機能で行える。

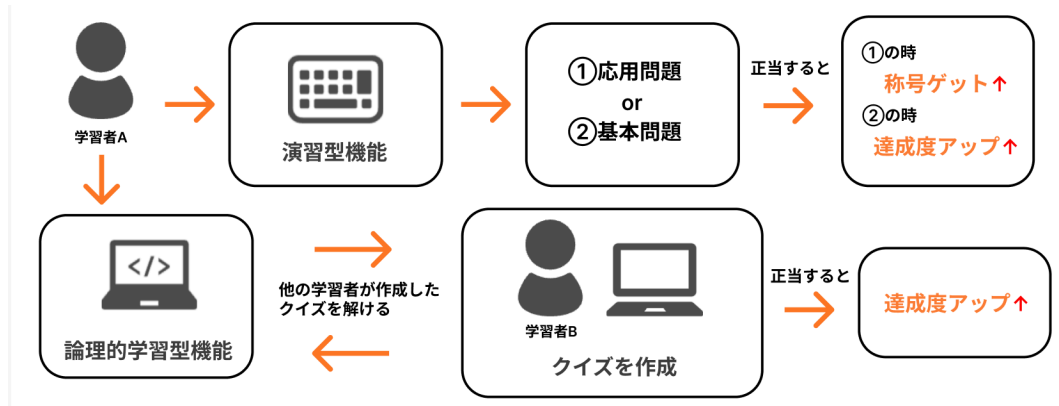


図 4.1 システム全体概要図

### 4.3 システム選択画面の設計

クイズ形式学習機能かコード記入型学習機能を選ぶ画面では、システムを作る上で利用者の使いやすさとゲーミフィケーション手法をより効果的に使う目的として、全ての機能にトップページから移動できる動線を意識し作成する。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

システム選択画面のデザイン設計は図 4.2 に示した。



図 4.2 システム選択画面

## 4.4 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。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
- ランキング制度

本機能のフローは図 4.3 に示した。クイズ形式学習機能にはクイズを正解すると学習者の達成の度合いの得点に追加されていく。クイズは間違えても何度も挑戦できるようにする。また、学習者自身も Ruby に関するクイズを作成できるようにし、学習者が作成したクイズを他の学習者が回答することも可能である。クイズを作成すると、論理的学習型機能の通常得点の 1.5 倍の得点が達成の度合いに加算させる。学習者が作成したクイズに回答し、正解しても達成の度合いの得点に追加されていく。

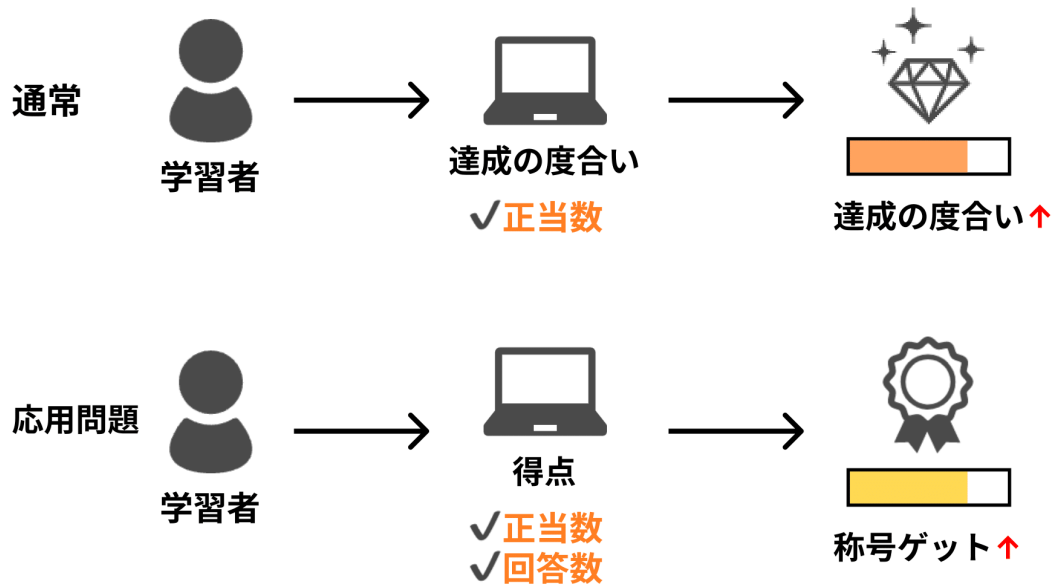


図 4.3 クイズ形式学習機能概要図 (仮)

#### 4.5 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。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
- 達成による称号の獲得
- 何度もやり直して回答が可能
- 時間制限が含まれている課題の搭載
- 講義後に講義内容の問題追加
- 問題選択の自由
- ランキング制度

本機能のフローは図 4.4 に表示した。正解するごとに得点が集計され、達成の度合いの得点に追加される。また本機能では、研究対象である「ジュニアドクター鳥海塾」の講義後に講義で学習した内容の応用問題を出題する。応用問題には 1 時間の時間制限を設け、学習者の理解度によって差が出る問題を選択する。応用問題の評価項目では、プログラムコードの正確さによって得点をつける。応用問題の得点は、通常の達成の度合いの得点には追加されず、応用問題単体の回答数・正当数に追加される。応用問題では、集計された正当数が 5 増えていくごとに学習者が称号をもらえる。

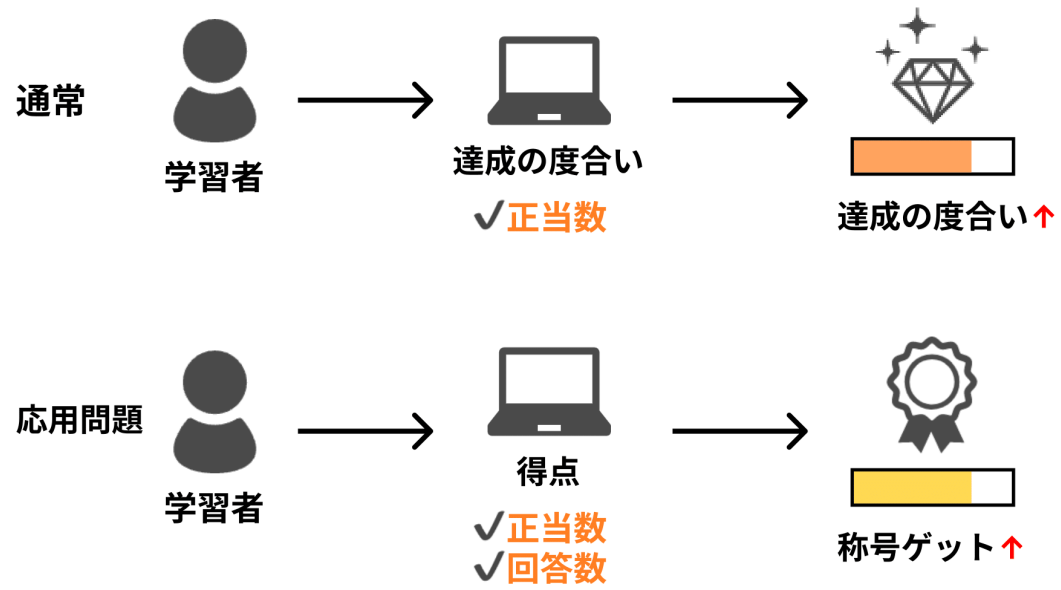


図 4.4 コード記入型学習機能概要図

#### 4.5.1 データベースの設計



## 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 開発言語

開発環境を以下に示す。

- 開発言語
  - サーバーサイド
    - \* Node.js(version 18.0.0)
  - クライアントサイド
    - \* HTML(HyperText Markup Language)
    - \* CSS3
    - \* JavaScript(ES2022)
- プラットフォーム
  - Docker(version 19.03.13)

### 5.2 システム全体の開発

本節では、システム全体に通じる画面とそれに含まれる機能を説明する。システムを作る上で利用者の使いやすさとゲーミフィケーション手法をより効果的に使う目的として、全ての機能にトップページから移動できる動線を意識してデザイン図 5.2 を作成した。システム全体を通して、利用者の学習の選択肢を増やすことを意識して作成した。

### 5.3 クイズ形式学習機能の開発

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。



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の開発

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。



## 第6章

# ゲーミフィケーションの適用実験

本章では, 本システムを利用し, 「ジュニアドクター鳥海塾」でアンケートを行い適用実験を行う。

### 6.1 研究対象

本研究では, 東北公益文科大学が実施している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 40 名の学習者を対象に, 事前アンケート調査を行う。アンケート内容は以下の通りである。事前調査から塾生の現状の確認を行い, 現状の改善を目標としたシステムを開発し, モチベーションの維持・向上を目指していく。

1. 自学習の頻度や時間
2. 交流活動に関するもの
3. 競争活動に関するもの



## 第7章

# 結論

本章では, 本システムを

### 7.1 考察

これからかく。

### 7.2 今後の展望

演習型機能のページの基盤は作ることができた。しかし, いまだ機能は不足しており, 利用者のユーザー情報を作成し, ログインする機能やユーザー情報と結びつけて達成の度合いに反映する機能を作成できていない。また, 論理的学習型機能の機能も不足している。まずは, 学習者が演習型機能を利用し達成の度合いに反映する機能, 学習者がクイズを作成し, 他の学習者との競争活動を行う機能の実装を進め, システムとしての完成度を高めていく。また, 「ジュニアドクター鳥海塾」の塾生の学習状況の調査を行い, 本研究のシステムを使う前と使った後の自学習に関する結果をまとめる予定である。調査の結果から, 本システムのゲーミフィケーション要素によって自学習の頻度や時間は増えたのかという結果を得る予定である。



## 参考文献

- [1] 内閣府.”Society 5.0”.[https://www8.cao.go.jp/cstp/society5\\_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](https://www.mext.go.jp/content/20200218-mxt_jogai02-100003171_002.pdf),(参照 2022-7-05) .
- [3] MIT.”SCRATCH”.<https://scratch.mit.edu/>,(参照 2022-11-20) .
- [4] 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”.[https://www.jstage.jst.go.jp/article/jlem/24/1/24\\_8/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/jlem/24/1/24_8/_pdf/-char/ja),(参照 2022-11-15) .
- [5] 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”.[https://www.jstage.jst.go.jp/article/jjet/37/1/37\\_KJ00008721376/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/jjet/37/1/37_KJ00008721376/_pdf/-char/ja),(参照 2022-7-05) .
- [6] 忍者 CODE.”あなたのコードが未来を灯す”.<https://ninjacode.work/>,(参照 2022-12-5) .
- [7] 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) .
- [8] Dariana Dicheva,Christo Dichev,Gennady Agre&Galia Angelova.”Gamification in Education: A Systematic Mapping Study”.[https://www.researchgate.net/publication/270273830\\_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) .
- [9] 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」”.[https://www.jstage.jst.go.jp/article/digraj/9/2/9\\_31/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/digraj/9/2/9_31/_pdf/-char/ja),(参照 2022-11-15) .
- [10] 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”.[https://www.jstage.jst.go.jp/article/jsim/38/3/38\\_74/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/jsim/38/3/38_74/_pdf/-char/ja),(参照 2022-11-15) .