

卒業研究報告書

題目 ゲーミフィケーションを使用したC言語の文法や
アルゴリズムの学習支援アプリケーションの開発

指導教員 水野 修 教授

崔 恩瀨 助教

京都工芸繊維大学 工芸科学部 情報工学課程

学生番号 18122036

氏名 谷本 嵩晃

令和4年2月10日提出

ゲーミフィケーションを使用した C 言語の文法や
アルゴリズムの学習支援アプリケーションの開発

令和 4 年 2 月 10 日

18122036 谷本 嵩晃

概 要

プログラミング言語を学び始めることにおいて、その言語の文法や、アルゴリズムの実装方法を学ぶことは必須である。特に、プログラミング言語を初めて学ぶ初心者にとって、これらを学ぶのには、非常多くの時間を要する。これは初心者にとって、プログラミング学習に対して、常にモチベーションを維持し続けるのは非常に困難である。

Code Quiz ではユーザの学習モチベーションを向上させるためにゲーミフィケーションを用いて初心者だけでなく中級者、上級者のプログラム学習も支援する。また、Code Quiz を用いることでプログラミング学習に対するモチベーションを維持できたかを確認するため評価実験を行った結果、Code Quiz を用いることで学習のモチベーションを維持できることがわかった。さらに Code Quiz は、初心者が学ぶ単純な文法だけでなく、アルゴリズム実装方法など、中級者以降でも利用出来るように設計した。また、Code Quiz を使うことでユーザがプログラミング学習へのモチベーションを維持できたかを調査するために、実験最終日にアンケートを行った。その結果から、本アプリケーションは、モチベーションを向上させる事はできなかったが、改善点が多く得られ、モチベーションを向上させる可能性を多分に含んでいる事がわかった。

目 次

1. 緒言	1
2. 背景	2
2.1 pgtracer	2
2.2 ゲーミフィケーション	2
2.3 pgtracer の問題点	3
3. 研究概要	4
3.1 研究目的	4
3.2 アプリケーションの概要説明	4
4. Code Quiz	10
4.1 学習機能	10
4.2 レベルシステム	10
4.3 バッジシステム	11
4.4 カレンダー	11
4.5 実装	11
4.5.1 Flutter	11
4.5.2 Express.js	11
5. 実験	12
6. 実験結果	13
7. 考察	21
7.1 本実験から得られた考察	21
7.2 今後の課題	21
8. 結言	23
謝辞	23
参考文献	24

1. 緒言

情報工学において、プログラミング学習は非常に重要である。しかし、プログラミング学習が多くの初心者にとって困難である反面、アルゴリズムのみの勉強は初心者にとって難しくない事が報告されている [1]。つまり、初心者にとって最も困難なことは、各プログラミング言語の文法を勉強し、さらにアルゴリズムを実装することである。

この報告からプログラミング学習を支援するための様々なツールが存在する。その一例として pgtracer というツールがある [2]。このツールは、初心者を対象とし、C++およびJavaの学習支援ツールとして作成されたものである。しかし、このpgtracerには、一問あたりにかかる時間が多く、頻度やモチベーションに関する調査は行われていない。

本研究では、このモチベーションの問題を解決し、初心者のプログラミング学習を支援することを達成するために、Code Quizを開発した。

本稿の構成を次に示す。第2章で、本研究の背景として、プログラミング学習支援ツール pgtracer の紹介及び問題点、また、今回採用した手法の説明であるゲーミフィケーションについて述べる。第3章で、本研究の目的を述べる。第4章で、本研究で開発したアプリケーションの仕様と実装方法を述べる。第5章で、本研究で開発したアプリケーションに対しての評価実験の方法を述べる。第6章で実験結果を述べた後、第7章で実験結果から得られた考察と今後の課題を述べる。最後に、第8章で結論を述べる。

2. 背景

2.1 pgtracer

既存のプログラミング学習支援ツールの一つとして、掛下らが開発した pgtracer が挙げられる [2]. pgtracer は初心者を対象とした「補習, 学生の自習等で活用することを主目的とし, C++(構造化プログラミング) および Java(オブジェクト指向プログラミング) の初心者向けの学習支援ツールとして企画」されたものである. このツールは, 予め作成された, プログラムに対して, トークン, 文, 変数名等をマスクし, 穴抜きになった状態のプログラムをユーザに提示し, 正しいプログラムを記述させることでプログラミング学習を支援する.

この pgtracer は問題作成に対する汎用性が高く, 同じ元のプログラムであっても, 別の箇所を穴抜きにすることで, 別の問題として成立させることができる. そのため, ユーザがこういった箇所で間違えるのかを分析することができ, 効率的にユーザのプログラミング学習を支援できる [3], [4]. また, pgtracer を実際に学生に対し運用した結果によると, 実験を実施する前の試験の点数の割合が, 満点が7割近くなのに対し, 実施後は満点を獲得したユーザが, 9割を超えている.

以上を踏まえると, 非常に良い結果を示すことができていると考えられる.

2.2 ゲーミフィケーション

ゲーミフィケーション (Gamification) とはゲーム要素を, ゲームではない分野で活用することで, モチベーションの向上を図る手法 [5] である. ゲーミフィケーションの一例として, Piano Stairs が挙げられる [6]. これは, 2009年に Volkswagen 社がスウェーデンのストックホルムの地下鉄で, エスカレーターの隣にある階段をピアノに見立て, 階段を上るとその鍵盤の音がなるようにデザインしたものである. Piano Stairs を導入することで, 導入前に比べて, 階段を利用する人が66%増加した. ゲーミフィケーションはこういった行動経済学の分野以外でも, 様々な分野で新たな手法として着目されており, 特に教育分野において注目されている.

教育分野でのゲーミフィケーションの例として, Quest to Learn が挙げられる [7]. これは, ニューヨーク市の公立のミドルスクールとハイスクールで採られている教

育方針である。コンセプトとして「ゲームの根本的な設計原理を用いて、没入感の高い、ゲームのような学習体験を実現する学校」を挙げている。2012年には、英語と数学の州テストで生徒の成績を測定する Student Performance で A 評価を得ており、1年間で、パーセンタイルランキングが15パーセント上昇させることができた。

しかし、ゲーミフィケーションは依然発達段階であり、様々な分野での研究も必要とされている。本論文では、初心者がプログラミング学習において、モチベーションを向上させる事を目標としているため、この手法を有効であると判断した。

2.3 pgtracer の問題点

2.1 節で説明した pgtracer はユーザからのフィードバックの結果が肯定的で、有用性が高い結果を残したが、しかし、一問当たり平均して4分～5分の解答時間がかかるため、ユーザが授業と授業の間などの短時間で学習したい場合は様々なプログラムやアルゴリズムを学習することは困難である。

また、pgtracer の運用実験では、操作性についての調査は行っていたが、ユーザのモチベーションや、起動回数、頻度などの調査は行われていなかった。

以上のことから、ユーザにモチベーションを損なわせないような、プログラム学習を支援するアプリケーションの開発が必要であると思いついた。

3. 研究概要

3.1 研究目的

本研究の目的は、昨今注目視されているプログラミング学習を支援するアプリケーションの開発である。特に本研究においては、初心者が学習した文法を演習問題で学習することで、プログラミングに対するモチベーション維持を目標としている。

3.2 アプリケーションの概要説明

3.1 節で説明した本研究の目的を達成するために、本研究では、プログラミング学習を支援するアプリケーション Code Quiz(以下 CQ)(図 3.2) をネイティブアプリケーションとして開発した。CQ を Web アプリケーションではなく、ネイティブアプリケーションでの開発を行った理由は、Web アプリケーションの場合、ブラウザを起動して、URL にアクセスし、ログインをすることで、アプリケーションを起動する必要がある、アプリケーションの起動までに手間がかかり、ユーザに多段階の操作を要する。その反面、ネイティブアプリケーションの場合、Android など自分の機器に一度ダウンロードを済ませておけば、ホーム画面に設置することができ、ユーザがアプリの存在を認識しやすくなるだけでなく、アプリを起動するまでのプロセスが一回のタップのみになるため、ユーザに必要な最低限の操作のみ要求することができるからである。

また、CQ では、ユーザのモチベーションを向上させるために、2.2 節で説明したゲーミフィケーションを用いて、初心者だけでなく、中級者、上級者のプログラム学習を支援している。

CQ の支援対象は主に、本学の学生であるため、学習対象言語は C 言語とした。CQ は、初回起動時と 1 回目以降の起動時に表示される画面が異なる。CQ の初回起動時に、ユーザ名とパスワードを設定画面が最初に表示され、ユーザ名とパスワードを設定することができる(図 3.3)。次に、C 言語の知識レベルを設定できる画面が表示され、ユーザは C 言語に対する知識レベルを初心者、中級者、上級者の 3 つのレベルから選択できる(図 3.4)。その後、CQ の学習機能(4.1 節参照)で、ユーザが選択したレベルに合わせた適切な 7 つの問題が選択され、ユーザは提示された問題を

解く事で学習が出来る (図 3.5) と (図 3.6). 7つの問題が終了した後, 解いた問題の結果が画面に表示され, 全ての機能を扱うことができるようになる (図 3.7).

初回起動時以降の場合, ユーザはホーム画面 (図 3.8) で Question ボタンをタップし, レベルを選択し (図 3.10), 学習機能を利用し, C 言語の文法やアルゴリズムの実装方法についての問題を解くことができる (図 3.5 と (図 3.6)). また, ユーザはホーム画面の Score ボタンをタップし, レベルを選択することで, 現在解くことができる問題を確認することができる (図 3.9).

アプリケーションの画面遷移の仕方を図 3.1 に示す.

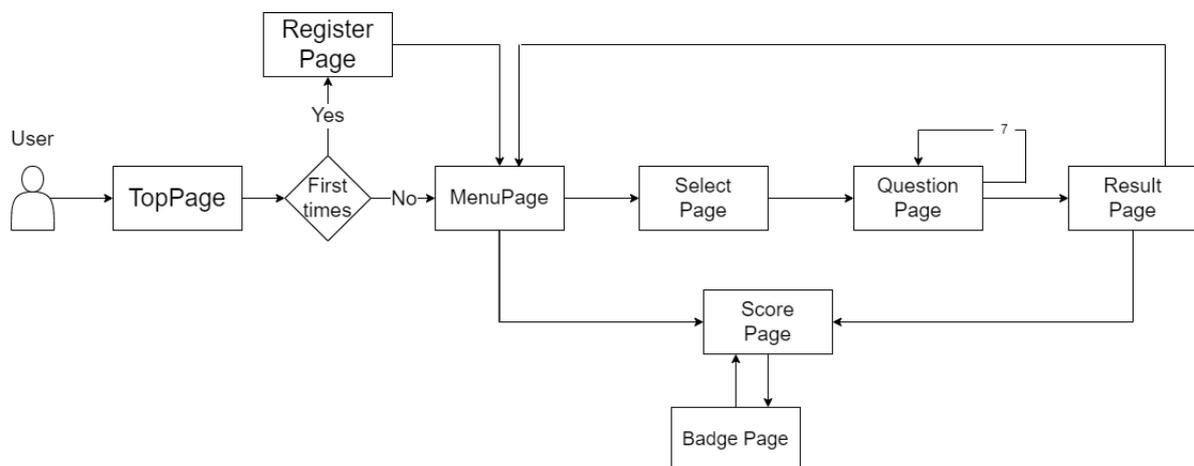


図 3.1 画面遷移

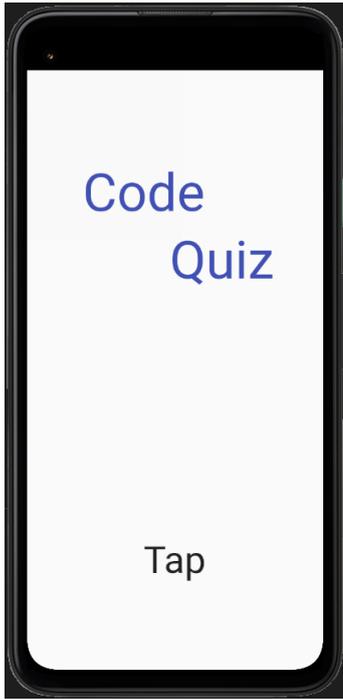


図 3.2 トップ画面

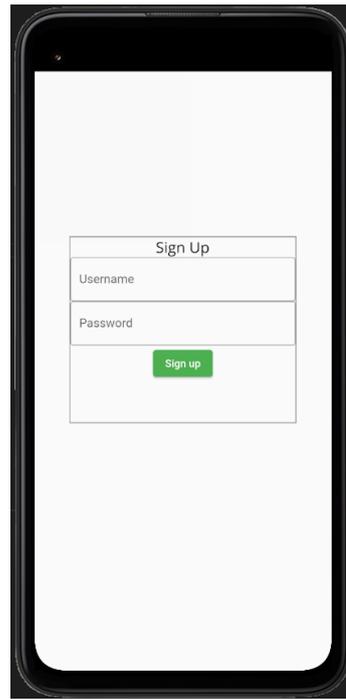


図 3.3 設定画面



図 3.4 レベル画面

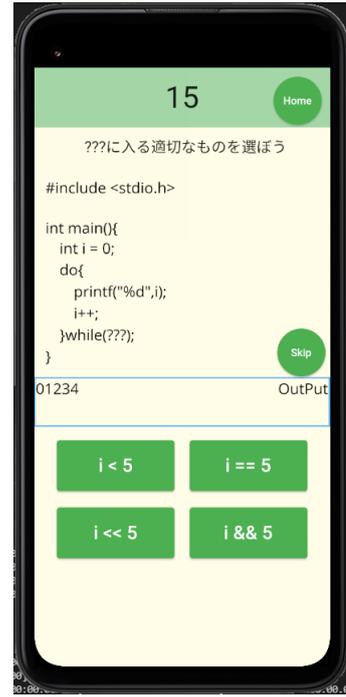


図 3.5 問題画面 (初心者)

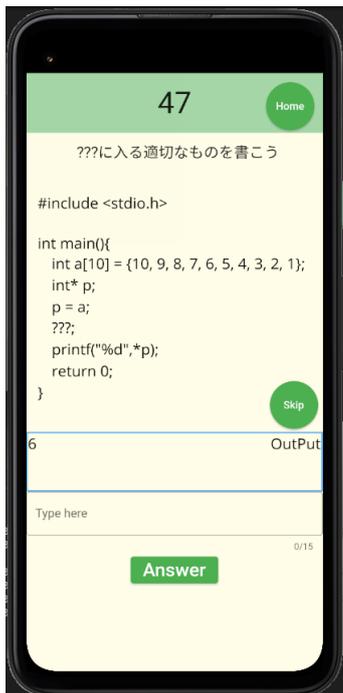


図 3.6 問題画面 (中級, 上級)

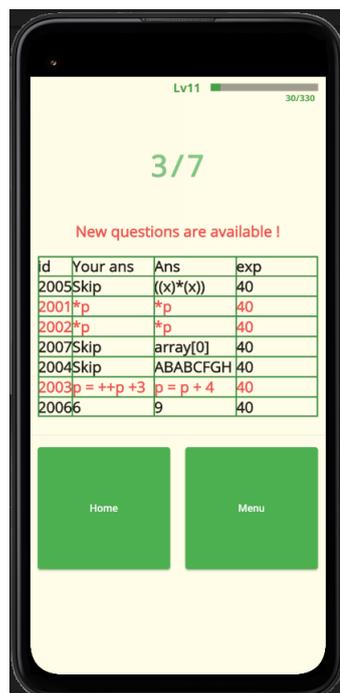


図 3.7 リザルト画面



図 3.8 ホーム画面



図 3.9 問題リスト画面

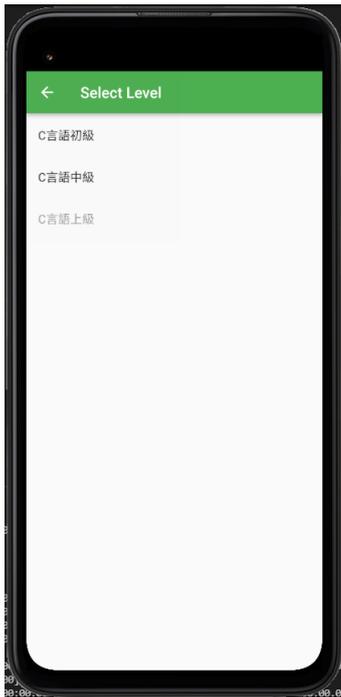


図 3.10 難易度選択画面

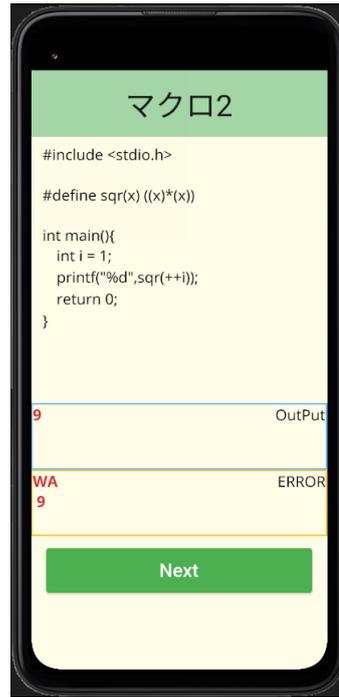


図 3.11 誤答画面

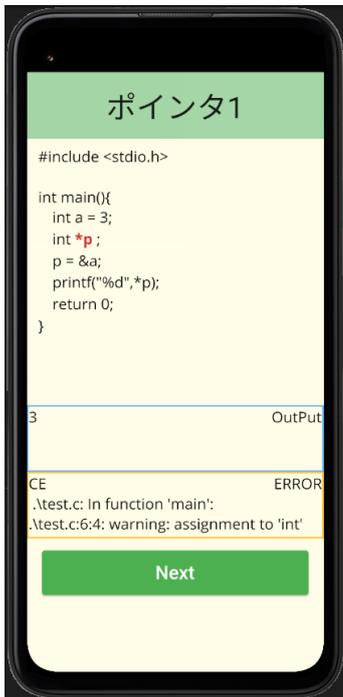


図 3.12 誤答画面 (コンパイルエラー)



図 3.13 バッジ画面



図 3.14 バッジ詳細画面

4. Code Quiz

次に、以下に CQ の主な機能と、実装方法を記載する。

4.1 学習機能

CQ の最も主な機能である。ユーザはこの機能を使って、C 言語の文法やアルゴリズムの実装方法を復習することができる。難易度は初級、中級、上級の三段階に分かれており、レベルによって、選択可能な難易度が解放される (図 3.10)。問題形式は穴埋め式であり、一回あたりに解く問題数は 7 問で、初級は 4 択形式、中級以降は記述式で回答する。7 問解き切った後に、結果が発表され、経験値の計算やバッジ判定も同時に行われる。問題は現在ユーザが解放されている問題の中からランダムに選択される。ユーザの回答は回答したタイミングで正誤判定が行われ、正解していた場合は次の問題に進み、誤答の場合は、正解を埋めたコードを表示する (図 3.11)。尚、中級以降の問題は、コンパイルした際のエラー文も表示される (図 3.12)。

また、ユーザが誤タップした時のための Home ボタンと、わからなかった場合の Skip ボタンを設置している。

4.2 レベルシステム

各ユーザにはレベルの属性を付与している。また、各問題にも経験値のパラメータを付与しており、ユーザが正解した問題から与える経験値を計算し、ユーザに与える。この時、ある一定値の経験値を超えていた時、ユーザはレベルアップする。レベルアップの主な報酬は、新たな問題が追加されることである。この新しく追加される問題は基本的に、経験値のパラメータが高く、ユーザのレベルアップを加速させる事が可能である。この機能を追加した理由としては、新しく追加された問題を解き、さらにレベルアップし、また新しく追加された問題を解くという学習において良いループが発生すると考えた。

4.3 バッジシステム

ホーム画面の Score ボタンをタップした後の画面の右下のボタンをタップすると確認することができる (図 3.13). ある一定の条件を満たすと, バッジが解放される (図 3.14). バッジの獲得判定は, 7 問解き切った後の結果発表の画面で行われる. どういった問題を正解したかを, 実績として管理することができ, ユーザのモチベーションを維持することができると思った.

4.4 カレンダー

練習を行った日にチェックマークを付け, 練習を行った日を管理することができる (図 3.8). これは, 毎日チェックをつけることで, モチベーションを維持することができると思った.

4.5 実装

4.5.1 Flutter

Flutter は, Google によって開発されたフリーかつオープンソースの UI の SDK である [8]. 単一のコードベースから, Android, iOS, Linux, macOS, Windows 等のクロスプラットフォームアプリケーションを開発するために利用される.

言語は Dart が採用され, Widget と呼ばれるコンポーネント単位で管理される.

4.5.2 Express.js

Express.js は, Node.js の Web アプリケーションフレームワークである [9]. 今回は, サーバサイドで動作させる REST API の作成に採用した.

5. 実験

当研究室に配布用のサーバを設置し，android版を公開した [10]．同時に，当アプリ (CQ) の説明資料も配布し，被験者 (5人) がアプリを使用する前に読んでもらい，主な機能の説明を行った．その後各自で使用してもらい，CQを使うことでユーザがプログラミング学習へのモチベーションを維持できたかを調査するために，最終日にアンケートを行った．アンケート項目を以下に示す．

表 5.1 アンケート項目

アンケート項目	
Q1.	C言語の経験年数
Q2.	プログラミングの経験年数
Q3.	当アプリ (CQ) を使うことで，プログラミング学習に対するモチベーションは向上した
Q4.	問題の難易度は適切だった
Q5.	そのように選択した理由を書いてください．
Q6.	制限時間は適切だった．もしくは，問題を解くのに十分な時間があった．
Q7.	そのように選択した理由を書いてください．
Q8.	当アプリ (CQ) の良かった点を書いてください．
Q9.	当アプリ (CQ) の改善したほうが良い点を書いてください．
Q10.	今回の実験・アプリについて自由に記述してください．

6. 実験結果

以下にアンケート結果を記載する.

Q1.C 言語の経験年数

C言語の経験年数

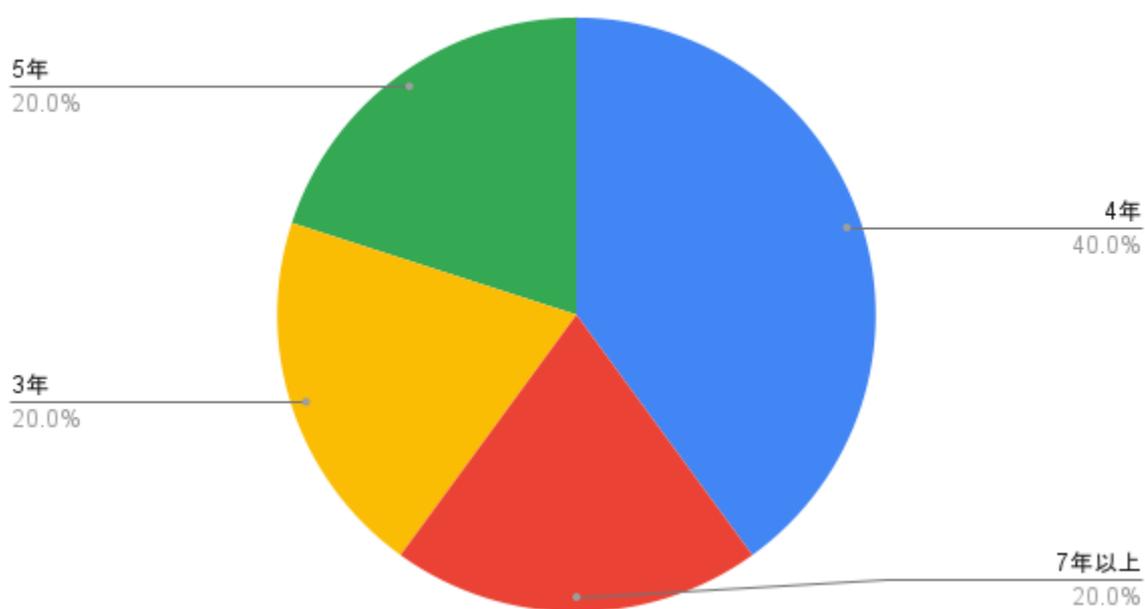


図 6.1 C 言語の経験年数

Q2. プログラミングの経験年数

プログラミングの経験年数

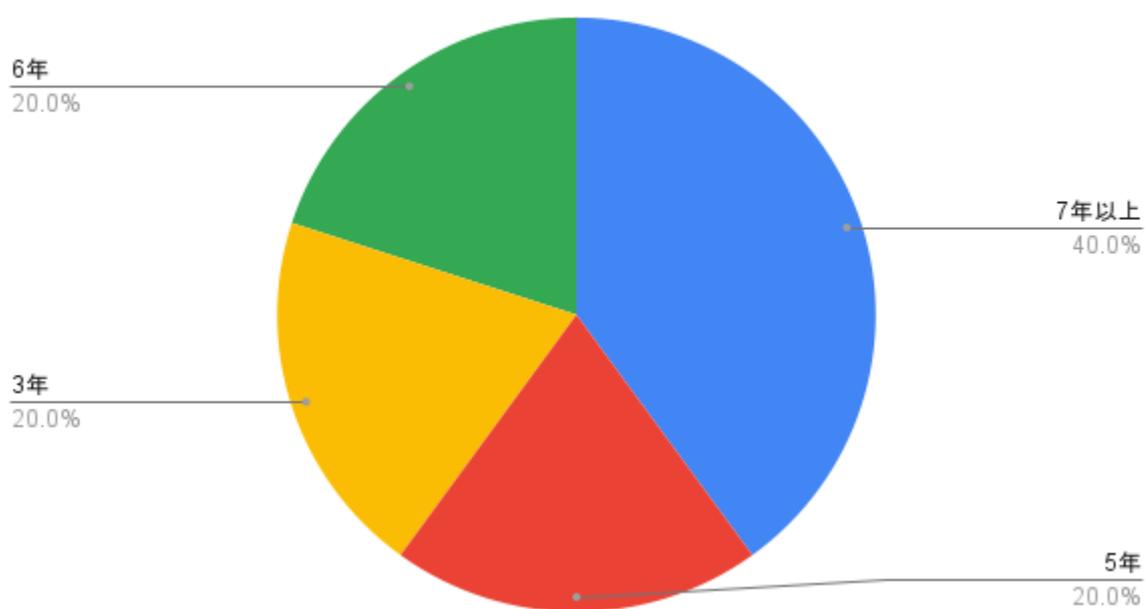


図 6.2 プログラミングの経験年数

Q3. 当アプリ (CQ) を使うことで、プログラミング学習に対するモチベーションは向上した

当アプリ(Code Quiz)を使うことで、
プログラミング学習に対するモチベーションは向上した

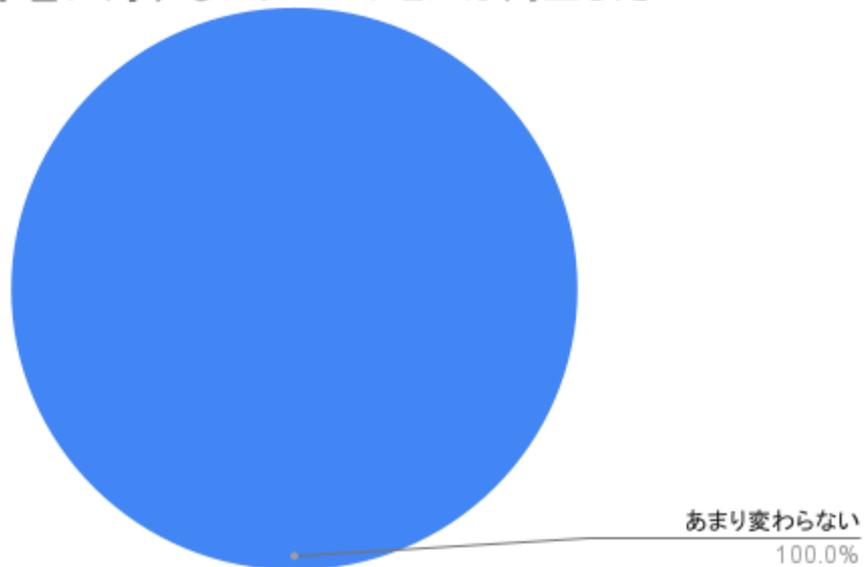


図 6.3 当アプリ (CQ) を使うことで、プログラミング学習に対するモチベーションは向上した

Q4. 問題の難易度は適切だった

問題の難易度は適切だった

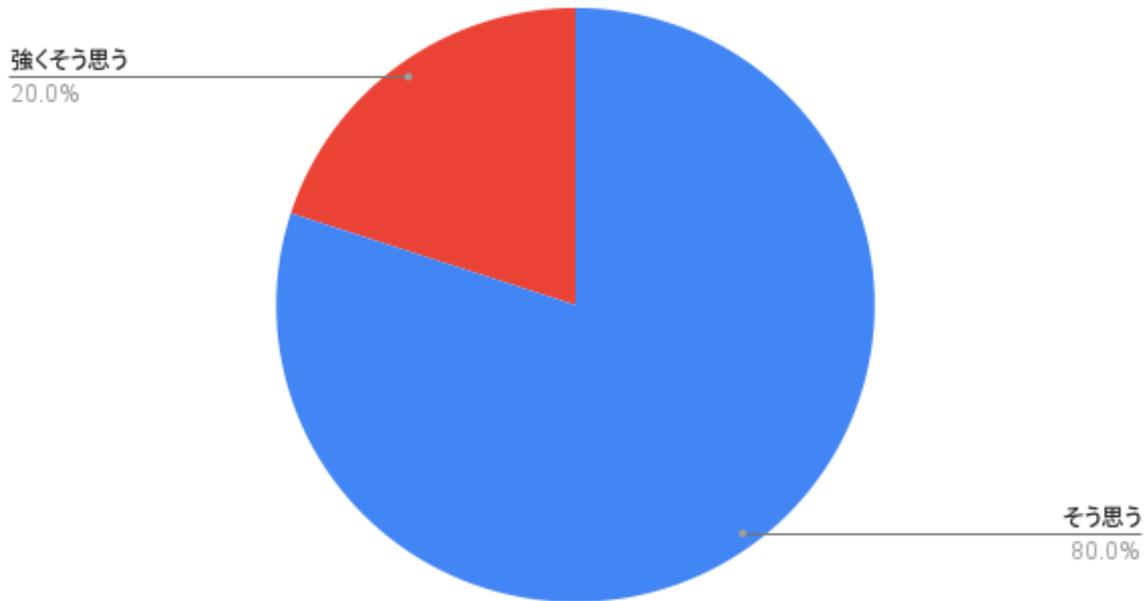


図 6.4 問題の難易度は適切だった

Q5. そのように選択した理由を書いてください。

1. 解いたのは初級だったけど、基礎的な文法を中心に問題が揃っていたように感じる。選択肢もよくミスりがちなものが並んでいた。
2. 完全な初心者になったときに、この初級難易度が適切かどうかは分からないので1つ下げて「強くそう思う」を選んだ。(初心者は main 文や include すら分からないはずなので)
3. 選択肢は、似た内容の項目が並んでいたため、しっかりとコードを読まないどれが正しいか判断することが難しかった。ここが良かったところだと思う。ひっかけ、少し意地悪な問題もあるように思ったが、逆に意欲を掻き立てているように感じた。プログラミング経験は長いですが、今でも正解を見てから、あ〜なるほどという知見があったり、素で間違える問題があった。
4. 初級と中級を解いたが、初学者がつまづきそうな型の扱いや、ポインタ等の問

題が多くあったので、簡単すぎず難しすぎないような難易度だと感じた。

5. C言語の仕様を学習しないと解くことができない問題が多く、C言語学習用のゲームとして良いものであると感じたため。

Q6. 制限時間は適切だった。もしくは、問題を解くのに十分な時間があった。

制限時間は適切だった。もしくは、問題を解くのに十分な時間があった。

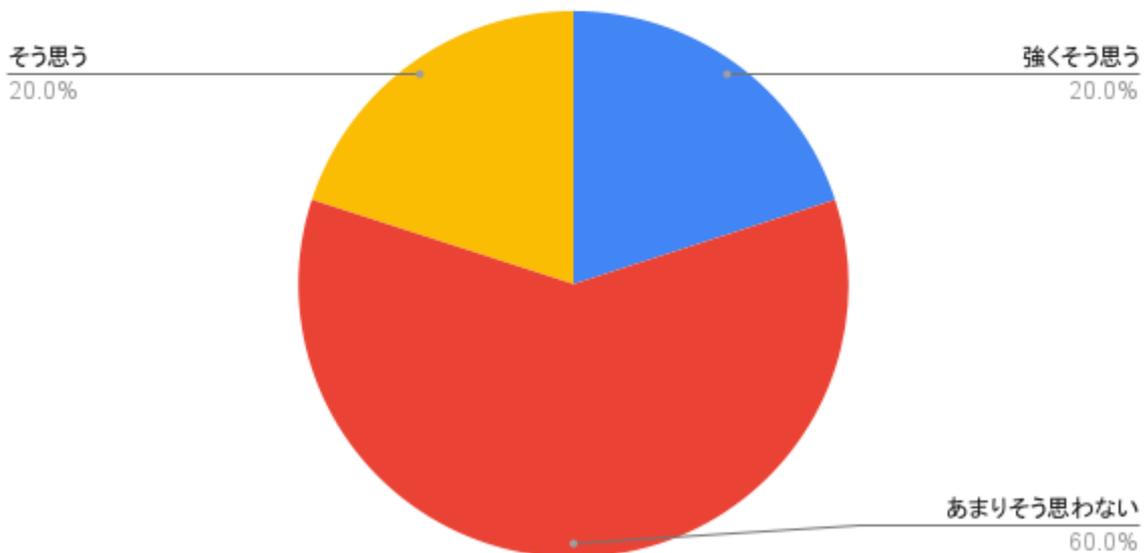


図 6.5 制限時間は適切だった。もしくは、問題を解くのに十分な時間があった。

Q7. そのように選択した理由を書いてください。

1. 基本的には知っていれば解ける時間が設定されていた。いきなり基数変換を 30 秒でさせるのはちょっとギョツとしたが、2 回目以降は普通に解ける時間配分だと思う。
2. 初見でプログラムを読んで、答えるには少し厳しい時間だと思いました。間違えるよりも、タイムアウトした問題の方がネガティブに感じてしまった（せつ

かく読んだのに答えられなかったし、もう次の問題か…という感じ) スマホの画面が小さくて、プログラムの全体を読むのに思った以上に時間がかかってしまったのかもしれない。縦幅が短く、全体像をつかみにくく、頻繁にスクロールする手間があったこと。特に横幅が足りなくて、折返しされてしまっていたところ、シンタックスハイライトが無いところが効いたと思う。一度見た事があるプログラムは、既に動作が頭の中に入っているのでは時間は足りたと思う。

3. 初級はコードの行数もあまり長くない、選択式の問題なので 30 秒は程よいと感じた。中級もコード行数はそれほど長く無いが、記述式かつスマートフォンの IME で入力したため記号などが若干打ちにくく 60 秒は短い場合もあると思った。
4. 焦ってじっくり考えられない問題もあったから
5. 思考する時間よりスマートフォンでのコード入力に時間がかかったため。

Q8. 当アプリ (CQ) の良かった点を書いてください。

1. 問題数も丁度良いし、1日に解ける問題のバリエーションも限られているので気軽に取り組みやすかった。
2. 自分に合った難易度の問題を解くことが出来ること。問題を解いた日が分かること。Levelがあることで、レベルアップするまで問題解いてみようと思ったこと。
3. 繰り返し問題を解くことになるので定着しやすいと思った、また中級以降は記述式で答えるため、正しく理解を求められる点が良いと思った。
4. 軽い気持ちでC言語の勉強ができる点
5. UIがシンプルでわかりやすかった。

Q9. 当アプリ (CQ) の改善したほうが良い点を書いてください。

1. レベルが画面上に表示されているのは分かったけど、それが具体的に何をしているのか、レベルを上げるインセンティブ等が伝わってこなかった
2. 隠されている Badge の最初の数個は獲得条件を書いておいたほうがユーザーのモチベーションが上がりやすいのでは？

3. 一言解説か正解は欲しいかも. パッと試せるのが良いところなので, 知識を得るところまでアプリで完結できるとユーザが離れにくくなる.
4. プログラムの読みやすさを改善したほうが良いと思います. 縦幅が短く, 全体像をつかみにくく, 頻繁にスクロールする手間があったこと. 横幅が足りなくて, 折返しされてしまっていたところ, シンタックスハイライトが無いところが効いたと思う. 新しく問題が解放されたようだけど, どこからその問題を解くことが出来るようになったのかよく分からなかった. 問題のレベル分けの下に更に, 問題セット 1, 2, 3...があれば分かりやすかったのかなと思います.
5. 経験値や問題が増えるシステムが分かりにくかったのでヘルプや説明を付けてほしい. (例えば中級以降は??Lv から挑戦できる, とか)
6. 問題に正解するとすぐに次の問題に移行するが, 正解した時も一旦その問題を振り返る時間があれば良いと思った. (勘で選択肢を選んで正解してしまう可能性もあるので)
7. バッジを解放する条件のヒントがあれば良いと思った. (個人的にはヒントがあった方がモチベーションになると思う)
8. ???の位置を見つけにくかったので, もっと強調して表示したほうがよいと思った.

Q10. 今回の実験・アプリについて自由に記述してください.

1. 復習としてたまに遊ぶくらいなら良いコンセプトだと思いました. 全く知らない言語に関してこのアプリで学ぶかと言われると, 初心者はこういった一問一答形式の細かい知識をはたして得たいのだろうか?という疑問は残ります.
2. クイズを解くシステムの部分は問題無さそうな気がします. 後は, 継続してもらうための仕組みを充実させていくことだと思います. 特に, 1人だと寂しかったので, 他に問題を解いている人がいるのが見えると良かったな, と思います. それだとランキングとか, フレンドの機能とかになるのかな.
3. アプリは一目で機能が分かるのでとても使いやすく感じた. その反面スマートフォンでしか利用できないので, 問題をスクロールする必要があったり, 文字が打ちにくかったりするので, ブラウザでもプレイできるような仕組みがあ

れば良いなと思った。初級から基本的な知識が求められるため、勉強の振り返りになるツールだと思うが、全くの初心者であれば取っ付きにくいかもしれない。(これは想定している対象者の範疇を超えているかもしれないので気にしないでください)

4. なぜスマートフォンのアプリケーションとして開発したのかわかりませんでした。このアプリケーションの対象である(はずの)プログラミング学習者は、自分のパソコンを持っている(または自由に使えるパソコンがある)ケースが多いはずなので、パソコン上で動くアプリケーションとして開発された方がコードの打ち込みがしやすいと思われます。そんな中であえてスマートフォンのアプリケーションとして開発した理由があるならば、その理由を明確にした上で報告書に書いたほうが良いと思います。

また、以下にログから抽出したデータを示す。

表 6.1 ログから得られたデータ

延べログイン回数	41 回
延べ問題回答数	約 450 回以上 (初級問題含まず)

7. 考察

7.1 本実験から得られた考察

Q3の結果(図6.3)を見る通り、今回の結果では、CQを使うことで、プログラミングに対してモチベーションを維持することはできたが、上げることができなかった。これには様々な要因が考えられる。

初めに、今回の被験者が全員C言語の経験年数が4年以上であったこと挙げられる。これは、被験者が既にC言語の知識レベルが成熟しており、新しく学びそれを試す段階を既に終えてしまっていた。

次に、ユーザが回答する際のユーザビリティが良くなったことが挙げられる。Q4の結果(図6.4)やQ5の結果を見ると、問題の難易度に対する評価は高いと言えるが、Q6の結果(図6.5)やQ7の結果やQ9の結果を見ると、コードの掲載箇所が小さく、シンタックスハイライトもないため、コードを読むのにも時間がかかってしまうという問題と、スマートフォンでコードを記述するのは、非常に手間がかかるという問題が発生してしまった。

最後に、バッジシステムを有効活用できていなかった点が挙げられる。事前の説明資料でシステムは説明できていたが、バッジの獲得条件を隠していたことや、獲得報酬を設定していなかったため、バッジの獲得を目的としたときに、何を頑張れば良いのか不明瞭になってしまった。

これらの点から、モチベーションは上がらなかったと考えられる。

7.2 今後の課題

一番の課題は入力方法の改善である。C言語において頻繁に使われる記号(*,=,+等)が一般に搭載されているIMEでは、入力がしづらい。なので、入力の際に、キーボードだけでなく、頻繁に使われる記号はキーボードの上部にショートカットのボタンを設置するなどして、スマートフォンでのプログラミングをしやすいように改善する。

次の課題は、シンタックスハイライトの機能を追加することである。昨今のテキストエディタには標準で備わっている機能なので、シンタックスハイライトの無い

ソースコードは非常に可読性が悪い。

最後の課題は、ゲーミフィケーション要素をさらに有効活用することである。7.1節でも述べたように、バッジシステムを有効活用でき無かった点が挙げられる。さらに、Q10の2番のように、ランキングボードや、フレンド機能といった、他のユーザーと何らかの関わりを持たせられるシステムを構築すべきであった。またレベルアップした際に、どの問題が解放されたかを明瞭にする点も有効活用が出来ていなかったと言える。

8. 結言

本研究では、ゲーミフィケーションの手法を採用したプログラミング学習支援アプリケーション Code Quiz を開発した。Code Quiz を用いることでプログラミング学習に対するモチベーションを維持、モチベーションを向上させることは出来なかったが、維持には貢献が出来た。Code Quiz にはユーザビリティの面で多くの改善点が見られたが、手軽にプログラミング学習を行えるという印象を与えることが出来た。この点から、Code Quiz は、プログラミングに対するモチベーションを向上させる可能性を含んでいると言える。今回の実験では、C 言語の経験年数が4年以上の被験者のみだったので、今後は、初心者に対しての実験が必要である。

謝辞

本研究を行うにあたり、研究テーマの決定や研究に対する姿勢、本報告書の作成に至るまで、全ての面で丁寧なご指導を頂きました。本学情報工学・人間科学系 水野修教授、崔恩瀨助教に厚く御礼申し上げます。また、本報告書執筆にあたり研究生活を共にし、研究室を温かく迎えてくれたあくあたん、貴重な助言を多数頂いたソフトウェア工学研究室の皆さん、学生生活を通じて著者の支えとなった家族や友人に深く感謝致します。

参考文献

- [1] A. Oram, Making Software –What Really Works, and Why We Believe It, G. Wilson, ed., o’reilly, 2011.
- [2] 掛下哲郎, 柳田峻, 太田康介他, “穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発と評価,” 情報処理学会論文誌教育とコンピュータ, vol.2, no.2, pp.22–36, 2016.
- [3] 中山陽平, 掛下哲郎他, “プログラミング穴埋め問題における穴抜きの難易度と学生の解答過程のクラスタ分析,” 情報教育シンポジウム論文集, vol.2018, no.23, pp.166–173, 2018.
- [4] 村田美友紀, 嘉藤直子, 掛下哲郎他, “プログラミング学習支援ツール pgtracer を自学習に活用した授業実践と学習行動の分析,” 情報処理学会論文誌教育とコンピュータ, vol.6, no.2, pp.25–37, 2020.
- [5] 井上明人, ゲーミフィケーション, NHK 出版, 2012.
- [6] M. Peeters, C. Megens, E. Van Den Hoven, C. Hummels, and A. Brombacher, “Social stairs: taking the piano staircase towards long-term behavioral change,” Proceeding of International Conference on Persuasive Technology, pp.174–179, 2013.
- [7] “Quest to learn (q2l) middle school and high school,” <https://www.q2l.org/>, 2022.
- [8] “Flutter - build apps for any screen,” <https://flutter.dev/>, 2022.
- [9] “Express - node.js web アプリケーション・フレームワーク,” <https://expressjs.com/ja/>, 2022.
- [10] “配布用のサーバ,” <https://se.is.kit.ac.jp/beakfish/>, 2022.