

# Unity による学習アプリの提案

広瀬研究室 3 年  
C1211088 佐藤逸樹

令和 6 年 1 月 11 日

## 概要

日本では 2020 年からは小学校, 2021 年からは中学校, 2022 年からは高校でのプログラミング教育の必修化が始まっている。小学校, 中学校の場合, 実際に行われるプログラミング教育とは文字を入力していくようなプログラミング言語を学ぶものではなく, ビジュアル型プログラミング言語などを用いてプログラミング的思考を育むものである。実際にテキスト型プログラミング言語を学ぶのは高校に進学してからである。しかし, 後に学ぶ言語に触れておくことは予習となり, 高校に進学してかの授業理解に役立つのではないだろうかと考えた。そこで, 小学生, 中学生が高校に進学したときに予習となるような学習アプリケーションの作成を提案する。本研究で作成するアプリケーションは, Unity を用いて作成し, パズル形式で学習できるようにすることで, 小学校, 中学校で扱うビジュアル型プログラミング言語を学ぶ際の形に近い状態で, テキスト型プログラム言語を学べるようにする。(412 文字)

# 目次

第 1 章	はじめに	5
1.1	背景 . . . . .	5
1.2	目的 . . . . .	5
第 2 章	先行事例	7
2.1	先行サービスの例 . . . . .	7
2.2	先行研究の例 . . . . .	7
第 3 章	Unity を用いた学習アプリケーションの提案	9
3.1	システムの提案 . . . . .	9
第 4 章	システム的设计	11
4.1	システム的前提 . . . . .	11
4.2	システムの全体の流れ . . . . .	11
4.3	システムの概要 . . . . .	12
4.4	学習内容について . . . . .	12
第 5 章	システムの開発	13
5.1	開発環境 . . . . .	13
5.2	システムの機能 . . . . .	13
5.3	システムの提供方法 . . . . .	14
第 6 章	実験	15
6.1	実験方法 . . . . .	15
第 7 章	結論	17
7.1	現在の状況 . . . . .	17
7.2	今後の展望 . . . . .	17
参考文献		19



# 第1章

## はじめに

### 1.1 背景

2017年及び2018年に行われた文部科学省による学習指導要領改訂 [1] によって、コンピュータを理解し上手に活用していくことは、これからの社会を生きていく子どもたちにとって、将来どのような職業に就くとしても、極めて重要なこととなっているという考えから、2020年には小学校、2021年には中学校、2022年には高校でのプログラミング教育の必修化が行われている [2]。平成28年に行われた文部科学省の有識者会議によると、小学校、中学校のプログラミング教育は実際に文字を書くようなテキスト型プログラミング言語を学ぶのではなく、自分の意図した一連の活動を実現させるためにどのような動きや記号が必要で、それらをどのように組み合わせたり改善していけばよいかを論理的に考えていく力であるプログラミング的思考を育むことが目的であると定義した [3]。テキスト型プログラミング言語は高校に進学してから学習する [4]。文部科学省はプログラミング的思考を育む方法のうちコンピュータを用いる方法でブロックを積み上げるようにしてプログラミングを行うビジュアル型プログラミング言語を用いて自分が考えている動きを実現させる方法を想定している。実際に岐阜県の小学校では第3学年からビジュアル型教材を扱っているという実施例がある [5]。ビジュアル型プログラミング言語は初学者にとって使いやすいプログラミング言語と考えられており、入門段階で成功を取めている。

しかし、ビジュアル型プログラミング言語が入門段階で成功を取めていると言っても、高校に進学後いきなりテキスト型プログラミング言語を学ぶことは、学習内容を理解することが難しい人も出てくるのではないかと考える。プログラミングに限らず学校で学ぶことは一度で覚えられるものばかりではないため、学校での学びで授業に遅れず理解を深めるためには予習を行うことが重要であると考えます。

そこで、このビジュアル型プログラミング言語を利用して後に学習するテキスト型プログラミング言語を学ぶことは、今後プログラミング言語を学習する際の予習となり、授業理解が捗るのではないかと考えた。

### 1.2 目的

本研究ではビジュアル型プログラミング言語の形でテキスト型プログラミング言語を学習できるアプリケーションを開発し、プログラミング学習を受けている小中学生が、高校に進学した後にプログラミング言語を学習する際の予習としてもらうことを目的とする。



## 第 2 章

# 先行事例

### 2.1 先行サービスの例

ビジュアル型プログラミング言語はすでにいくつも世の中に出回っている。ここでは先行サービスを 3 つ紹介していく。

#### 2.1.1 Scratch

Scratch<sup>\*1</sup>は世界最大の子供向けコーディングコミュニティでシンプルなビジュアルインターフェースを持ったコーディング言語であり、特に 8 歳から 16 歳向けにデザインされているがすべての年代の人々使われている。自宅、学校、博物館、図書館など幅広い場面で利用されている。

#### 2.1.2 Viscuit

Viscuit<sup>\*2</sup>はメガネという単純な仕組みたった一つだけで簡単なプログラムから組み合わせ次第で複雑なプログラムまで作ることができるプログラミング言語である。Viscuit ではアニメーション・ゲーム・絵本などを簡単に作ることができる。

#### 2.1.3 TILES

TILES<sup>\*3</sup>はシンプルなプログラミング言語であり使用する命令は 6 つのみであるため多くの命令を覚える必要がない。またプログラミングの基本を学ぶための学習テキストがついているため、学習を勧めていくのが容易である。この学習テキストの応用編では人工知能の基礎を学ぶこともできる。

### 2.2 先行研究の例

ここではビジュアル型プログラミング言語とテキスト型プログラミング言語の両方を使用した先行研究を紹介していく。

---

\*1 <https://scratch.mit.edu/>

\*2 <https://www.viscuit.com/>

\*3 <https://vpltiles.com/>

### 2.2.1 松澤芳昭・酒井三四郎による研究

松澤と酒井の研究 [6] は、ビジュアル型プログラミング言語は入門段階で成功を収めているが、将来テキスト型プログラミング言語を必要とする学習者がテキスト型言語への発展が見えにくいシステムを利用することは、入門学習への動機付けをそぐ理由となり、ビジュアル型プログラミング言語で学習した内容がその後の発展につながらないとしたら本末転倒であるとして、ビジュアル型プログラミング言語とテキスト型プログラミング言語を相互変換できる BlockEditor というシステムを用いて、ビジュアル型言語でプログラミングの基礎知識を学習し理解したあとテキスト型言語による実装へとシームレスな移行を目的としている。結果として、最終的に全体の 80 % の人の BlockEditor 使用率が低下している。また BlockEditor を使うことによる Java 習得への影響についての質問で BlockEditor を使うことで Java 構文の理解が深まると思うというものの支持率が半数程度得られた。このことから BlockEditor が Java 言語習得の足場かけとして機能しているという結論がなされている。

### 2.2.2 岡本恭介・安藤明伸による研究

岡本と安藤の研究 [7] は、ビジュアル型プログラミング言語として Blocky Games を、テキスト型プログラミング言語として Python を利用し、その学習順序がプログラミングへの知識・技能・情意面の変化として教育効果にどのような影響を与えるのか考察することを目的としている。結果として、どちらのグループも事前・事後アンケートの結果から情意面の意識に有意な変化はなかったが、事前・事後テストでは先にビジュアル型プログラミング言語を学習したグループの方がもう一方に比べ「全くわからない」という回答が少なくなっていたことから、先にビジュアル型プログラミング言語を学習することの優位性が示された。

### 2.2.3 先行事例の課題点

スクラッチやタイルズはもともとが子供向けに作られているためプログラミングを全く行ったことがない人であってもプログラムを組むことができるが、ビジュアル型言語であるためテキスト型言語を習得することができない。ビスケットはプログラムを書く際にメガネという独自のルールに則って作業をするため、テキスト型言語を使用する際の形と大きく異なることになる。松澤、酒井らは彼らの研究の限界として、BlockEditor の利用率の分析にとどまっており、BlockEditor が理解度の向上に寄与した直接的なデータが得られなかったことを上げている。岡本恭、安藤らの研究の今後の課題として、彼らが実験で行った画像処理に対する学習以外の学習内容に変えても効果が得られるかどうかというものを挙げている。

### 2.2.4 先行研究からの考察

先行研究では、先にもしくはテキスト型言語と併用してビジュアル型言語を学習することによる優位性やテキスト型言語習得への足がけとなることが示された。しかし、理解度の向上や他の学習内容に対して効果があるのかは直接的に示されていない。本研究では、ビジュアル型言語を用いて直接テキスト型言語を学ぶため理解度の向上及び他の学習内容に対して効果があるのかどうかを示すことができると考えられる。これはこれからプログラミング教材を作成する人にとって意義のあることである。本研究で開発するシステムは先行研究に残された課題を解決できるよう提案を行った。



## 第3章

# Unity を用いた学習アプリケーションの提案

ここでは、先述した先行事例の課題点を解決できるシステムを提案する。

### 3.1 システムの提案

本研究では、語群欄に表示させたビジュアル型言語のような文字の書かれたブロックをパズルゲームのように解答欄に並べ、問題に回答することでテキスト型言語を学習できるシステムを作成する。(図 3.1) 問題の前にはその問題で扱われる単元やメソッドに対する説明を行う。(図 3.2) このシステムを利用することでプログラミングを初めて学習する人やビジュアル型言語しか学習したことがない人であっても、簡単にテキスト型言語を学習することができる。

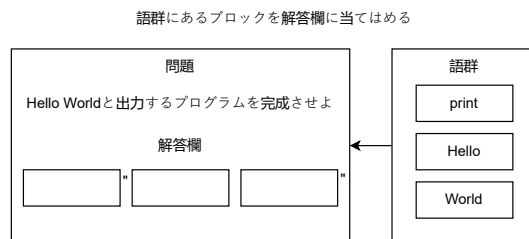


図 3.1 クイズの出題例

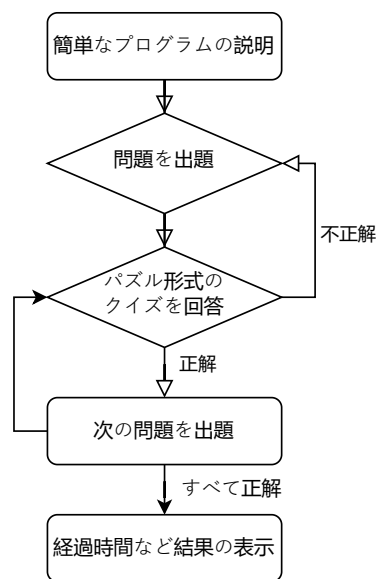


図 3.2 システムの流れ

## 第4章

# システム的设计

### 4.1 システムの前提

- 前提1 システムの利用者はテキスト型プログラミング言語を学習したことのない学生とする。
- 前提2 システムは Web 上に投稿し，アクセスした人は誰でも利用できるようにする。
- 前提3 アクセス及びシステムを用いた学習のタイミングはテキスト型プログラミング言語を学習するまでの間で任意とする。
- 前提4 システムの利用は，テキスト型プログラミング言語学習のための予習とするために行う。

### 4.2 システムの全体の流れ

1. 利用者はシステムが投稿された Web サイトにアクセスする。
2. 学習したい単元を選択し，問題のあるページに移動する。
3. はじめに問題に即した説明文を読む。
4. 問題文を表示し，解答に使うブロックとブロックを当てはめるための解答欄を表示する。
5. 解答欄に正しくブロックが配置されたら正解を表示する。
6. 1 から 3 のを別の問題で繰り返す。

システム概念図は図 4.1，システムの流れは図 4.2 の通りである。

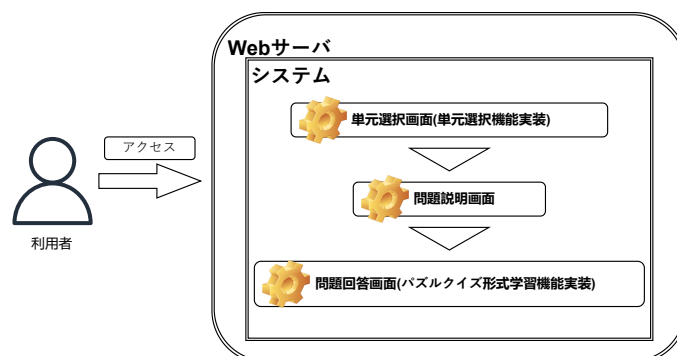


図 4.1 システムの概念図



図 4.2 システムの流れ

### 4.3 システムの概要

本システムの機能は、プログラミング言語である Ruby 言語についての問題を学習する单元ごとに用意し、その問題に利用者が本システム上でパズル形式で回答するという機能である。单元は問題回答前に利用者自身が選べるようにする。

### 4.4 学習内容について

プログラミング言語学習で学習する单元の一覧は図 4.1 のとおりである。本研究で作成したシステムを利用者が Web 上で利用する際に学習するプログラミング言語は東北公益文科大学の基礎プログラミングの授業でも扱われている Ruby とし、内容は上記の单元の中でもプログラミングをする際によく使われる変数、制御構造、メソッドの一部の範囲とする。

表 4.1 プログラミング学習における单元の一覧

单元	主な学習内容
変数	値と変数の種類、演算子、代入演算子、論理演算子 など
制御構造	if, while など
メソッド	print, printf, gets, chomp, to.i など
配列	配列について、配列の作り方、配列の値を取り出す方法 など
正規表現	正規表現とは、正規表現の使い方 など
ハッシュ	ハッシュとは、ハッシュの利用方法 など
など	

## 第 5 章

# システムの開発

### 5.1 開発環境

- Unity

Unity とは主に 2D, 3D ゲームの制作に用いられるゲーム制作エンジンで簡単なゲームから複雑なゲームまで様々なゲーム開発に用いられている

- C #

C #とはマイクロソフトが開発したプログラミング言語でオブジェクト指向型の言語であり, GUI アプリ開発や, Webt アプリ開発, ゲーム開発などに使われる。今回は Unity にプログラムを入力する際に使用。

### 5.2 システムの機能

ここではシステムに含まれる機能を説明する。

#### 5.2.1 単元選択機能

単元選択機能は, 単元選択画面に実装される機能である。システムの利用者がシステムを利用するときに自身が学習したい単元を選択するための機能である。作成途中の単元選択機能の画面を図 5.1 に示す。この図の unit1 と書かれているようなところに単元名を記入し, その場所が押されたら問題説明画面\*<sup>1</sup>に移動した後, パズルクイズ形式学習機能が実装された問題回答画面へ移動する。選択できるようにする単元の内容は 5.4 節で示したとおりである。

#### 5.2.2 パズルクイズ形式学習機能

パズルクイズ形式学習機能は, システムの利用者が単元選択画面で選択した単元についての問題が出題され, それに対して利用者はパズル形式で回答する機能である。作成途中のパズルクイズ形式学習機能の画面を図 5.2 に示す。この図の quiz と書かれているところに問題文が表示される。右にある world” と書かれたブロックがおいてある欄は語群であり, そこから解答欄である緑色の箱にブロックを移動して問題に回答する。

---

\*<sup>1</sup> 選択した単元やその問題で扱われるメソッドに対する説明を行う画面。その画面のときに画面を押すと問題回答画面に移動する。

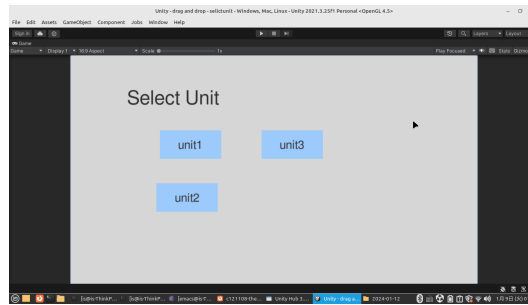


図 5.1 単元選択画面の図 (仮)

このような構図にしたのはシステムの利用者が Web サイトにアクセスする媒体が計算機であってもスマートフォンであっても同じ画面で利用できるようにするためである。

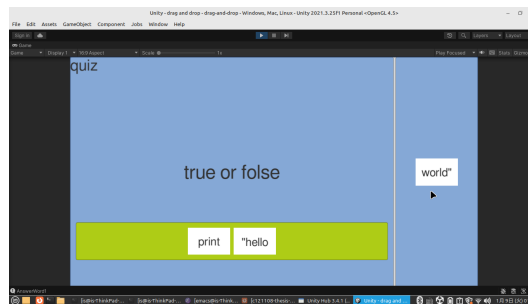


図 5.2 問題回答画面の図 (仮)

### 5.2.3 機能の開発

単元選択機能及びクイズ学習機能の開発には Unity と C # を使用する。

## 5.3 システムの提供方法

Unity ではシステムを web サイトに投稿することができるため web 上に上げたシステムを使用者に使ってもらうという形で提供する。

## 第6章

# 実験

### 6.1 実験方法

本研究のシステム評価の方法として東北公益文科大学の野球部一年生を利用者として、システムを利用してもらった後、アンケートに回答してもらいその結果によって評価を行う。(図.6.1)



図 6.1 実験の流れ

アンケートの内容は表 6.1 のとおりである。

- 実際にコードを書くときに自分でかけそうか  
この質問の回答結果からシステムによる学習で利用者が学習内容に対する予習となったかを考察する。
- これからもプログラミングを続けられそうか  
この質問の回答結果からシステムによる学習で利用者の学習意欲に対する影響を考察する。
- 自由記述  
利用者のシステムに対する意見を書いてもらい、システムの改善に繋がられるようにする。

表 6.1 アンケートの内容

質問	できない←→できそう
実際にコードを書くときに自分でかけそうか	1 - 2 - 3 - 4
これからもプログラミングを続けられそうか	1 - 2 - 3 - 4
自由記述	





## 第7章

# 結論

### 7.1 現在の状況

現在システムは単元選択機能・パズルクイズ形式学習機能を実装するための UI 画面はおおむね完成といったところまでできており、機能そのもののプログラムを書き進めているところである。課題点として利用者が使用する端末によって画面や動作にどのような変化が現れるかを確認できていないことが挙げられる。

### 7.2 今後の展望

今後の展望として機能ごとのプログラムを完成させ各機能を連結させることを考えている。また、システムが完成したときにもっと利用しやすい UI にするためにや使用する端末によって画面や動作にどのような変化が出るのかを確認し、どの端末であってもほぼ同じ動作と画面になるように改良を行っていく。



## 参考文献

- [1] 文部科学省. “小学校プログラミング教育の手引（第三版）”. 2020-02. [https://www.sejuku.net/blog/wp-content/uploads/2021/07/20200218-mxt\\_jogai02-100003171\\_002.pdf](https://www.sejuku.net/blog/wp-content/uploads/2021/07/20200218-mxt_jogai02-100003171_002.pdf), (参照 2023-08-15).
- [2] 文部科学省. “学習指導要領改訂に関するスケジュール”. [https://www.mext.go.jp/content/20220907-mxt\\_kyoiku02\\_100002604\\_001.pdf](https://www.mext.go.jp/content/20220907-mxt_kyoiku02_100002604_001.pdf), (参照 2023-08-15).
- [3] 文部科学省. “小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）”. 2016-06-16. [https://www.mext.go.jp/b\\_menu/shingi/chousa/shotou/122/attach/1372525.htm](https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm), (参照 2023-11-20).
- [4] 文部科学省. “高等学校 学習指導要領（平成 30 年公示）”. 2018-03. [https://www.mext.go.jp/content/20230120-mxt\\_kyoiku02-100002604\\_03.pdf](https://www.mext.go.jp/content/20230120-mxt_kyoiku02-100002604_03.pdf), (参照 2023-08-15).
- [5] 岐阜県教育委員会 学校支援課. “小学校プログラミング教育 実践事例集”. 2020-02. <https://www.pref.gifu.lg.jp/uploaded/attachment/238416.pdf>, (参照 2023-08-15).
- [6] 松澤 芳昭, 酒井 三四郎. “ビジュアル型言語とテキスト記述型言語の併用によるプログラミング入門教育の試みと成果”. 研究報告コンピュータと教育. vol.2013-CE-119, no.2.
- [7] 岡本 恭介, 安藤 明伸. “ビジュアル型とテキスト型プログラミングにおける学習順序が教育的効果に与える影響”. 2020. [https://www.jstage.jst.go.jp/article/jjet/44/Suppl./44\\_S44052/\\_pdf/-char/ja](https://www.jstage.jst.go.jp/article/jjet/44/Suppl./44_S44052/_pdf/-char/ja), (参照 2023-09-25).