

3次元空間を用いた初学者向けプログラミング学習環境

A Programming Environment for Beginners in a 3-Dimensional Space

渡辺 雅也[†], 松浦 敏雄[†]

Masaya WATANABE[†], Toshio MATSUURA[†]

概要 : 2020年から始まる新しい学習指導要領によって小中高校でプログラミングが必修となる。そこで初学者を対象とした様々なプログラミングの学習環境が用意されていることが望ましい。Scratchは、2次元のアニメーションを簡単に作成できることから、初学者に向いていることが知られている。このように2次元モデルをベースとした環境の有用性から、3次元モデルを用いることでさらに初学者の興味を引くことができるのではという考えに至った。

そこで、本研究では初学者でも容易に使用可能な3次元空間を用いたプログラミング学習環境を設計・実装を行いその有効性を検証することを目的とする。

実装した学習環境は二つの使い方を提供しており、一つは3次元空間に物体をインタラクティブに操作し、積み木のように自由に物体を組み合わせたりの使い方である。もう一つは物体をプログラムによって作成・移動などの指示を出す使い方である。ここで使用するプログラミング言語は既存の言語ではなく、物体に指示を出すための単純な言語を実装した。本プログラミング学習環境は基本的な動作を実装することができた。

キーワード : 学習指導要領, プログラミング, 初学者, 3次元空間, 積み木

Keywords: Course of Study, Programming, Beginner, 3-Dimensional Space, Combining Blocks

1 はじめに

2020年から始まる新しい学習指導要領^[1]によって小中学校および高等学校でプログラミングが必修となる。小中学生、高校生の多くはプログラミングの初学者であることが想定されるため、初学者に向けた多様なプログラミングの学習環境が望まれる。初学者向けのプログラミング学習環境の一つとしてMITで開発されたScratch^{[2][3]}が注目されている。Scratchは実行結果が図形などで表示される動きのあるプログラミングで、学習者のモチベーションが維持しやすく文字ベースのプログラミングよりも初学者の学習に向いていることが知られている。このようにScratchは初学者に対して2次元モデルの有用性を示しているが、3次元モデルならさらに興味を引くことができるのではないかという考えに至った。

しかし、一方で2次元の画面を通して3次元モデルを見ることになるのでかえって学習に不向きという可能性も否定できない。3次元モデルの一例としてMinecraft^[4]があるが、環境の設定が必ずしも容易でないこと、有償であること、さらに高機能すぎて初学者には扱いにくいことなどのため初学者向けの学習環境としては使いづらい。

そこで本研究では初学者でも容易に使用することができるような3次元空間を用いたプログラミングの学習環境を設計・実装することでその有用性を確かめる。実装した学習環境は二つの使い方を提供しており、一つは3次元空間に物体をインタラクティブに操作し、自由に物体を組み合わせたりの方法である。もう一つは物体をプログラムによって作成・移動などの指示を出す方法である。ここで使用するプログラミング言語は既存の言語ではなく、物体に指示を出すための単純な言語を実装した。

以下、2章では関連研究について述べる。3章では作成したソフトウェアの概要およびソフトウェア内で用いる言語の仕様について述べ、4章でまとめる。

[†]大阪市立大学大学院 創造都市研究科

[†]Graduate School for Creative Cities,
Osaka City University

2 関連研究

2.1 Scratch

Scratch は MIT メディアラボが無償で公開している初学者向けプログラミング学習環境である。Scratch は作成したプログラムの実行結果が 2 次元の物体のアニメーションとして表示される。また、作成したプログラムを公開・閲覧など共有することが可能で、学習者が学習のモチベーションを維持させるための工夫がなされている。

2.2 MakeCode for Minecraft

MakeCode for Minecraft(図 1) は Minecraft を使用した Microsoft が提供するプログラミングツールである。Minecraft は Mojang が開発したサンドボックスゲームである。x, y, z 座標が存在する 3 次元空間上に様々な種類が存在するボックスを空間に配置することで建造物を作ったりなどが可能で、その他にも多彩な機能が存在する。MakeCode for Minecraft で作成したプログラムを用いて Minecraft 内で命令を呼び出すことで対応した処理が行われる。このように MakeCode for Minecraft は 3 次元モデルを用いたプログラミングが可能である。しかし、Minecraft の環境設定が必ずしも容易でないこと、有償であること、さらに高機能すぎて初学者には扱いづらいことが想定されるため初学者向けとしては使いづらい。



図 1: MakeCode for Minecraft

3 本ソフトウェアの概要

本ソフトウェアは二通りの使い方があり、一つは積み木のように立方体や球等の**ブロック**と呼ばれる物体を利用者の対話的な指示によって 3 次元の空間に配置したり、移動させたりする使い方である。もう一つは、プログラムを記述し、それを実行することで、ブロックに対して同様の操作を行うものである。前者を**積み木モード**、後者を**プログラミングモード**と呼ぶ。積み木モードは空間認識能力を高めるための学習に利用でき、プログラミングモードは 3 次元空間を利用したプログラミング学習環境として利用できる。

3.1 画面構成

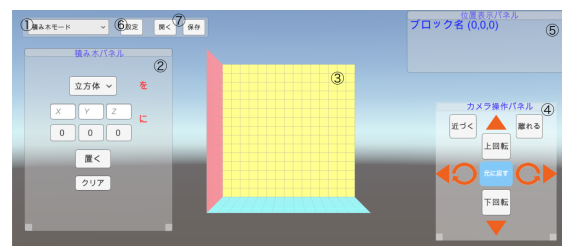


図 2: 積み木モードの画面

ここでは画面を構成するパネル、および、ブロックを配置する 3 次元空間について説明する。図 2 は積み木モードの画面表示例である。①の**モードメニュー**では積み木モードとプログラミングモードを切り替えることが可能である。②の**積み木パネル**は積み木モードで表示されるパネルで、どんなブロックを③の**ブロックエリア**内のどこに置くかを指定する。④の**カメラ操作パネル**で視点の変更を行う。⑤の**位置表示パネル**は選択されているブロックの座標と名前を表示する。⑥の**設定ボタン**を押すと**設定メニュー**(図 8)が表示される。設定メニューは重力の有無の切り替え、ブロックエリアのグリッド数の変更などを行う。⑦の**開くボタン**はファイルの読み込みを行う。**保存ボタン**はファイルの保存を行う。

図 3 はプログラミングモードの画面である。③～⑦は積み木モードと同様であるが、②は**プログラム**

パネルと呼び、プログラムの記述，実行を行う。

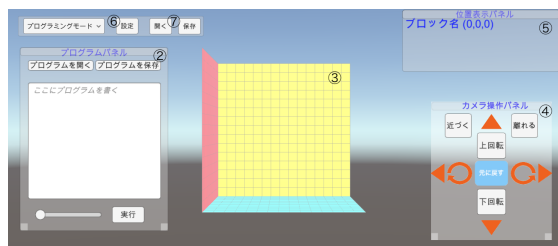


図 3: プログラミングモードの画面

3.1.1 ブロック

まず，このソフトウェアの操作対象となるブロックについて説明する。ブロックは空間上に置いたり，移動させたりすることが可能な物体であり，立方体，球および円柱の 3 種類が基本の形として存在する。これらを図 4 (a)~(c) に示す。また，基本の形を組み合わせることで新しいブロックを定義することが可能である。(図 4 (d))

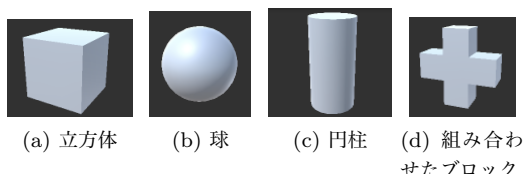


図 4: 使用可能なブロック

各ブロックは名前を有している。積み木モードでは積み木パネルで指定したブロックの種類によって自動的にブロックの名前が設定される。例えば，立方体なら $c1, c2, \dots$ 球なら $s1, s2, \dots$ というように順に名前がつけられる。プログラミングモードではユーザーが自由に名前を決めることができる。

3.1.2 積み木パネル

積み木パネルは積み木モードで表示されるパネルである。図 5 はその画像表示例である。その操作方は①のプルダウンメニューでブロックの種類を指定し，②でブロックを置きたい (x, y, z) 座標をそれぞれ入力する。そして④の置くボタンを押すことでブロックを置くことができる。これを繰り返すこ

とでブロックを配置できる。⑤のクリアボタンを押すと置いたブロックが全て消去される。



図 5: 積み木パネル 図 6: 連続の設定

また，③を用いることでブロックを連続で置くことができる。例えば図 6 のように $+1$ を選択するとブロックを置いた直後に x 座標が自動的に $+1$ 加えられ，次に置くボタンを押した際にその座標にブロックが置かれる。置くボタンを連続して押すだけで， x 座標が 1 ずつ増えた位置に次々とブロックを置くことができる。連続する座標の値は図 6 のように $+2, +1, 0, -1, -2$ から設定できる。 y 座標， z 座標も同様の操作が可能である。

3.1.3 ブロックエリア

ブロックエリアとは縦，横，高さが予め決められた空間で，この空間にブロックを置くことができる。この空間の外にはブロックを置くことはできない。各座標軸やその向きは図 7 のとおりである。

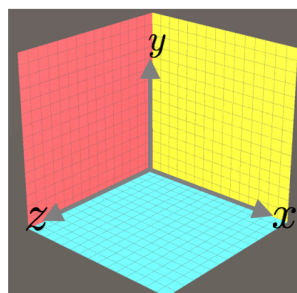


図 7: ブロックエリア

3.1.4 設定

図 2 または図 3 の設定ボタン (図中の⑥) を押すことで設定メニューを表示する (図 8)。設定メニュー

では重力の有無の切り替え、グリッド数の変更ができる。画面のレイアウトを初期位置に戻すことが可能である。

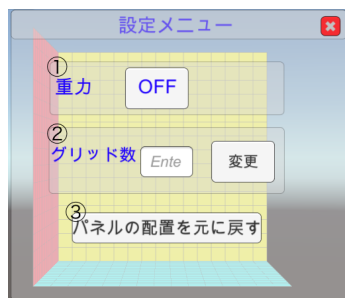


図 8: 設定メニュー

(1) 重力

ブロックエリアの重力の有無を切り替えることができる。初期状態では重力無しに設定されている。ブロックを配置した後で「重力あり」に切り換えるとブロックが落下する。ただし床面 ($y = 0$) より下には落下しない。

(2) グリッド

ブロックエリアのグリッド数を変更できる。初期状態ではグリッド数 15 に設定されている。

(3) パネルの配置を元に戻す

設定メニュー③の「パネルの配置を元に戻す」ボタンを押すことでパネルのレイアウトを初期状態に戻すことができる。

3.1.5 位置表示パネル

座標パネルは現在選択しているブロックの名前と座標を表示する。図 9 は一つのブロックを選択している状態の一例である。

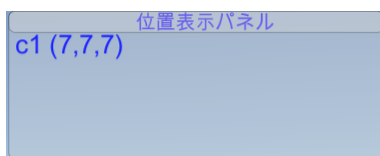


図 9: 位置表示パネル

3.1.6 カメラ操作パネル

初期設定の視点はブロックエリアの中心に向けられており、カメラ操作パネルを使うことで上下左右

に回転、移動、前後にズームすることができる。移動した視点を元の位置に戻すことも可能である。カメラ操作パネルを図 10 に示す。

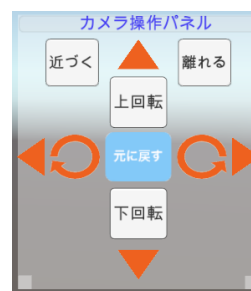


図 10: カメラ操作パネル

3.1.7 保存パネル

図 2 の保存ボタンを押すと保存パネル (図 11) が表示される。ここで保存したいファイル名を入力し、保存パネル内の保存ボタンを押すとブロックエリアにあるブロック群を指定された名前のファイルとして保存できる。ブロック群が保存されたファイルを特に**ブロックファイル**と呼ぶ。

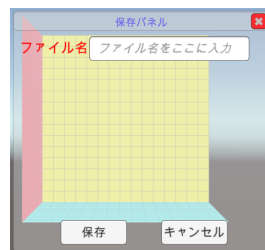


図 11: 保存パネル



図 12: 開くパネル

3.1.8 開くパネル

図 2 の開くボタンを押すと開くパネル (図 12) が表示される。ロードパネルには保存されているブロックファイルの一覧が表示されており、ここから読み込みたいファイルを指定し、開くパネル内の開くボタンを押すことでブロックファイルが読み込まれる。

3.1.9 プログラムパネル

プログラムパネル (図 13) はプログラミングモードで表示され、①のテキストエリアにプログラムを入力する。②の実行ボタンを押すことで①で記述されたプログラムが実行される。③のスライダーによってプログラムの実行速度を変更できる。④のプログラムを**プログラムを開く**ボタンで作成済みのプログラムの読み込みを行う。④の**プログラムを保存**ボタンでは①に記述しているプログラムを保存する。また、プログラムを保存しているファイルを**プログラムファイル**と呼ぶ。

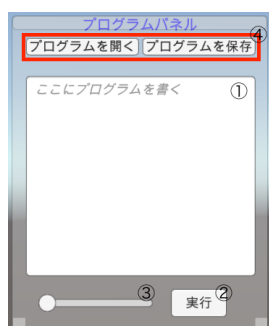


図 13: プログラムパネル

3.2 プログラミング言語の仕様

ここではプログラミングモードで用いるプログラムの形式、命令文の一覧を記載する。

3.2.1 言語の形式

言語の形式は初学者が複雑な記述があると理解しづらいと考えられるので、

命令名 対象となるブロック名 引数といった単純な形式で作成した。

3.2.2 命令一覧

作成した言語の命令一覧を以下に示す。

ブロックの定義

ブロックの定義では定義するブロック名と種類(引数)を記述することで定義することができる。定義されたブロックは非アクティブ状態であるためこの命令だけではブロックエリアにブロックを置くこ

とができない。

```
define A c(1,1,1)
```

ブロックを置く

この命令を記述することで非アクティブ状態のブロックをブロックエリアに置くことができる。引数の座標は絶対座標である。

```
put A (1,1,1)
```

ブロックの移動

ブロックエリアに置かれているブロックに対して指示を出すことができる命令で定義されただけのブロック、定義されていないブロックには指示を出すことができない。引数の座標は相対座標である。

```
move A (10,15,5)
```

ブロックの削除

ブロックエリアに置かれているブロックを削除することができる。

```
remove A
```

ブロックのグループ化

ブロックエリアに置かれているブロックをグループ化することができる。グループ化されたブロックはグループに命令を出すことで一度に移動させたり、色を変えることが可能になる。

```
group G (A,B,C)
```

グループから外す

グループに含まれているブロックをグループから外すことができる。

```
extract G (A)
```

ブロックの色を変更

ブロックエリアに置かれている指定したブロックの色を変更することができる。引数は R, G, B の三色の値である。

```
color A (255,0,0)
```

並列処理

命令は上から順に逐次処理で行われるが、命令 para を用いることで並列処理を行うことができる。並列処理を行う範囲は para から end までに書かれている命令である。

```
para  
...  
end
```

繰り返し

命令の繰り返しは for を用いることで可能になる。引数は繰り返す回数で、繰り返しを行う範囲は for から } までに書かれている命令である。

```
for{ (3)
...
}
```

4 おわりに

本研究では 3 次元空間を用いた初学者向けプログラミング学習環境の設計・実装を行った。本プログラムでは積み木モードとプログラミングモードの 2 つを備えており、積み木モードは物体をインタラクティブに操作することで 3 次元空間上に配置したり、移動させることができる。プログラミングモードではプログラムによって物体を操作する機能を備えており、初学者に適したプログラミング学習環境を提供できるようになった。本プログラミング学習環境は概ね動作しており、小中学校で試用し、評価することが今後の課題である。

参考文献

- [1] 文部科学省：“小学校学習指導要領解説 総則編”，http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2017/07/12/1387017_1_1.pdf p.83-87 (2018 年 1 月 25 日確認).
- [2] MIT メディアラボ：“Scratch について”，<https://scratch.mit.edu/about> (2018 年 1 月 22 日確認).
- [3] “Scrach wiki”，<https://jp.scratch-wiki.info/wiki/Scratch> (2018 年 1 月 22 日確認).
- [4] Microsoft：“MakeCode for Minecraft”，<https://minecraft.makecode.com/> (2018 年 1 月 22 日確認).
- [5] Unity Technologies：“Unity スクリプトリファレンス GameObject”，<https://docs.unity3d.com/ja/560/ScriptReference/GameObject.html> (2018 年 1 月 17 日確認).
- [6] Unity Technologies：“Unity スクリプトリファレンス MonoBehaviour”，<https://docs.unity3d.com/ja/560/ScriptReference/MonoBehaviour.html> (2018 年 1 月 17 日確認).
- [7] 森 秀樹, 杉澤 学, 張 海, 前迫孝憲：“Scratch を用いた小学校プログラミング授業の実践 ～小学生を対象としたプログラミング教育の再考～”，https://www.jstage.jst.go.jp/article/jjet/34/4/34_KJ00007142887/_pdf/-char/ja/ (2018 年 1 月 23 日確認).
- [8] 渡辺 雅也：“3 次元空間を用いた初学者向けプログラミング学習環境の設計と実装”，大阪市立大学大学院創造都市研究科修士学位論文, 2018.