

QRコードを用いたスマートフォン端末向け プログラミング学習環境

古川賢士^{†1} 神林靖^{†1}

概要: 本稿は、スマートフォン上で動作するタンジブルなプログラミング学習環境を提案する。本研究は、スマートフォンのみを所持しているプログラミング学習者に、快適に学習を行うことのできる環境を構築することでプログラミング初学者の学習を支援することを目的としている。初学者を対象としたプログラミング学習では、ビジュアルプログラミング言語がよく利用されており、これらの学習ツールは、プログラミングを行う際に構文を意味するブロックの接合など細かな操作が要求される。しかし、スマートフォンは画面が小さいため細かな作業に向いておらず、他のデバイスと同様の学習環境を実装した場合、操作性の悪化や誤作動の頻度が高くなると考えられる。本研究では、画面を操作することでプログラムを作成するのではなく、実在のブロックを用いて画面外でプログラムの作成を行うタンジブルな学習環境を構築することで、このような問題に対処する。

Programming Learning Environment on a Smart Phone that uses QR Code

KENSHI FURUKAWA^{†1} YASUSHI KAMBAYASHI^{†1}

1. はじめに

昨今、プログラミングが小学生の習い事として注目され、子ども向けプログラミング教室が全国各地で開催されている。プログラミングは、コンピュータに意図した処理を行うよう指示する体験を通して、課題解決に向けて順序立てて考える思考力、つまり論理的思考力を高めるのに有効だとされている。平成28年6月に文科省より公表された「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」[1]では、「プログラミング的思考などの育成を目的として小学校でのプログラミングは既存教科内で実施する」と提言されている。このプログラミング的思考とは、論理的思考のことである。プログラミング教室は、論理的思考力を高めると同時にITエンジニアに必要な技術を体験することができるとして注目されている。

プログラミング開発環境の多くは、JavaやCのようなキーボードを使用したテキスト記述型の言語でのプログラミングを前提としているが、プログラミング教室では、ビジュアルプログラミング言語を使用するが多い。ビジュアルプログラミング言語は、図形やアニメーションを用いてプログラムを記述する言語であるため、テキスト記述型の言語で生じる文法的なエラーが生じにくく、マウス操作のみでプログラミングが行えるため、キーボードに慣れない初学者が扱いやすい言語となっている。現在では、様々なビジュアルプログラミング言語がパーソナルコンピュータ（以下、PCとする）やタブレット向けに無料で提供されており、Scratch[2]やBlockly[3]などを使用したプログ

ラミング教育の研究が報告されている[4]、[5]。

多機能携帯電話であるスマートフォンの普及は増加し続けており、最近では、PCを所持していない利用者也増加している。総務省が公表した「平成28年版情報通信白書」[6]では、スマートフォンの世帯普及率は72.0%であり、20代のスマートフォン利用率は87.0%を占めており、ほとんどの者が保有しているという現状である。しかし、普及率に対して、スマートフォンのみを用いたプログラミング学習アプリは少ない。そのうえ、現在スマートフォンでプログラミング学習を行えるアプリにThe Foos[7]やDigital Puppet[8]といったアプリがあるが、これらのアプリは正解するごとに問題が次々と難しくなるドリルやパズルのようなアプリであり、Scratchのような学習者がプログラムを用いて自由な作品を創造できるといったアプリではない。

Scratchは、プログラミングに必要な構文を表すブロックを接続することでプログラミングを行うが、これをスマートフォンで実装した場合、ブロックの縮小により操作性が悪化し、操作ミスが発生する頻度が高くなると考えられる。この問題は、スマートフォンの画面が小さいために発生しているものであり、スマートフォンよりも画面が大きく同様の特徴を持つタブレットならば、Pyonkee[9]やScratch Jr[10]などのブロックを用いたアプリが数多くあることから言えるであろう。この問題を解決し、スマートフォンで作品を制作しながら、快適にプログラミング学習を行える環境を構築することで、多くの人がプログラミングに触れる機会を増やすことができると考えられる。

本研究では、スマートフォンの画面で行う工数を削減するには、スマートフォン以外の場所で作成した情報を取り

^{†1} 日本工業大学
Nippon Institute of Technology

込み、作品を制作する工程で使用する必要があると考えた。そこで、安価かつ認識精度の高い、QRコードとスマートフォンに搭載されているカメラを用いて、作品で使用する画像やプログラムを外部から取り込むことにした。

本稿では、QRコードを貼付した厚紙とスマートフォンのカメラを使用することで、画面で行う操作を最低限に抑えたシステムを提案する。

このシステムでは、プログラミングに実体のあるブロックを使用し、実際に手で組み合わせることで画面で発生する操作ミスを避けることができる。実体のあるブロックは、QRコードを貼付した厚紙を使用する。ブロックの認識にはQRコードを使用しており、読み取る際はスマートフォンのカメラを使用する。接続されたブロックのQRコードは、スマートフォンをスライドさせながら一つずつカメラに収めることで読み取ることができ、これらの読み取ったQRコードの文字列を元にプログラムを生成する。

また、作品で使用するキャラクターや背景の画像は、スマートフォンのカメラで撮影したものを使用することで、画像の選択や配置といった作業を省き、画面で行う工数を削減している。これらシステムを取り入れたプログラミング学習環境を制作した上で、スマートフォンで快適にプログラミングを行える環境について考察する。

本稿では以下の通りに記述する。第2章では研究背景と関連研究について述べる。第3章では提案するプログラミング学習環境の詳細を述べる。第4章では今後の課題について議論し、第5章ではまとめを述べる。

2. 研究背景と関連研究

本稿で提案している学習環境では、実体のあるブロックを使用してプログラミングを行う。実体のあるインタフェース、つまりタンジブル・ユーザ・インタフェース（以下、TUIとする）のプログラミング学習環境は、実体のあるブロックを使用してプログラミングを行うため、実際に手で組み合わせた各ブロックを認識し、PCに取り込む必要がある。

これまでに提案されたTUIの学習環境では、ブロックの認識に電気回路を用いた方法とカメラによる画像入力を用いた方法が使用されている。八城らは、ブロックの形状が学習者に与える影響を調べるため、ブロックの形状と認識方法が異なった学習環境を制作した[11]。八城らは、電気回路を用いた認識方法として、抵抗器が組み込まれたブロックと電氣的にブロックの判別を行うブロック判別器を用意した。ブロック判別器には電気回路が組み込まれており、判別器にブロックを配置することで、配置されたブロックの抵抗器から種類と位置の特定を行う。八城らは、電気回路を用いた認識方法の特徴として、即時性と認識精度が高いことを挙げている。電気回路を用いた認識方法は、実行命令を出してから実行結果を出力するまでの時間が短く、

学習者は実行結果を出力するまでの待ち時間を意識することなくプログラミングを行うことができる。また、ブロックの認識制度が非常に高いため、誤認識により間違った実行結果が表示され、学習者が混乱するといった事態を避けることができる。

しかし、本稿で提案している学習環境は、PCを所持していない学習者又は、PCの利用頻度が低い学習者に、現在所持しているデバイスのみで快適にプログラミング学習を行ってもらうために制作したものである。そのため、スマートフォン以外に必要なハードウェアが増えることは望ましくない。また、本稿で提案している学習環境は、学習者がプログラムを用いて自由な作品を制作しながらプログラムを学ぶものであるため、使用するブロックの個数が多くなることが想定される。そのため、ブロックはできる限り安価な物が望ましい。以上の理由から、本稿で提案している学習環境には、電気回路を用いた認識方法は適していないと判断し、カメラを用いた認識方法を採用した。

プログラミング支援環境は数多く存在し、これらは大きくGUIなものやTUIなものに分類される。

GUIのプログラミング支援環境として、MITの開発したScratchが挙げられる。Scratchでは、プロジェクトで使用するキャラクターなど操作対象のオブジェクトをスプライトと呼び、背景画像の上にスプライトを配置し、スプライトの動きをプログラミングすることで作品を制作していく。プログラムは、一覧からブロックを選択し、他のブロックと接合することで作成する。ブロックには、ブロックで表している構文と使用できる数値が表記されており、数値は必要に応じて変更が可能である。また、Scratchにはクローンと呼ばれる機能がある。クローンは、プログラム実行中に実行画面に描画されたスプライトを複製することができる機能である。複製されたスプライトは、複製元のスプライトと見た目は同じだが、異なったプログラムを設定することができる。

TUIのプログラミング支援環境として、前述した八城らの他に、鈴木らによって開発されたアルゴブロックが挙げられる[12]。アルゴブロックは、複数人での共同作業の支援を目的とした実体を持つプログラム言語である。より良い学習のためには複数人での共同作業が重要だとされているが、広く普及しているPCでのプログラミング学習は、ディスプレイやキーボードが複数人での使用に適した形をしておらず、共同作業に適した装置ではない。アルゴブロックは、実体のあるブロックにプログラム言語におけるコマンドが割り当てられており、これらを手で組み合わせることによってプログラムを作成することができる。鈴木らは被験者にアルゴブロックを使用した共同作業を行わせ、その様子の観察と分析から実体のあるブロックを用いたプログラミング学習の有用性を示した。

松崎らは、ARマーカーが貼付されたブロックを並べた

り、動かしたりすることでロボットの振舞いをプログラミングできる学習環境を提案した[13]。ブロックの認識に USB カメラを用いており、USB カメラの撮影範囲に並べられたブロックは画像として PC へ入力され、画像処理によってブロックの情報へと変換される。プログラミングに実体的なブロックを使用するため、コードを記述する必要がなく、子供でも容易に理解することが可能である。

3. プログラミング学習環境

本稿で提案している学習アプリの画面構成を図 1 に示す。このアプリは、Scratch のように自由な作品を制作しながらプログラミングを学ぶことができる。作品で使用する画像と画像の動きに関するプログラムを、カメラを用いて外部から取り込むことで、画面で行う操作を最低限に抑えたユーザ・インタフェースを実現している。



図 1 画面構成

このアプリでは、学習者が紙に描いたイラストを作品に使用することができる。描いたイラストは、スマートフォンのカメラで撮影することで本システムへと送られる(図 1, ②)。取り込まれたイラストに関するプログラムは、コマンドブロックと呼ばれる QR コードが貼付された厚紙を実際に手で並べることで作成する。イラストの撮影後、並べたコマンドブロックの QR コードを一つずつ撮影することで、QR コードに格納された命令内容が読み取った順番で配列に格納される(図 1, ③)。最終的に画面の再生ボタンを押すことで、配列に格納された命令内容に従ってイラストが動き、プログラムの結果を確認することができる(図 1, ④)。

3.1 作品で使用するイラスト

本環境では、イラストの動かしたい部分をスプライトと呼び、それ以外を背景と呼ぶ。スプライトと背景は、一枚の紙に描いてもらい、カメラを用いて画像データとして取り込み作品に使用する。スプライトを動かすためには、一枚のイラストのスプライト部分と背景部分を識別する必要があるため、撮影前にイラストのスプライト部分を識別マークで囲ってもらい、識別マークを元に画像認識でスプライト部分の識別を行う(図 2)。本環境では、配色が異なった識別マークが三種類用意されており、最大で三つのスプライトを登録することができる。

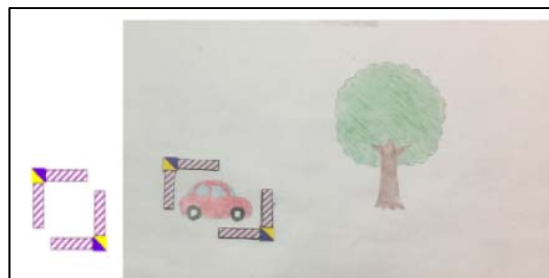


図 2 識別マークと使用例

3.2 コマンドブロック

コマンドブロックとは、スプライトに対する命令内容を表したブロックである。上部にはそのブロックが表している命令の説明が記述されており、下部には命令内容が格納された一辺 13mm の QR コードが貼付されている(図 3)。

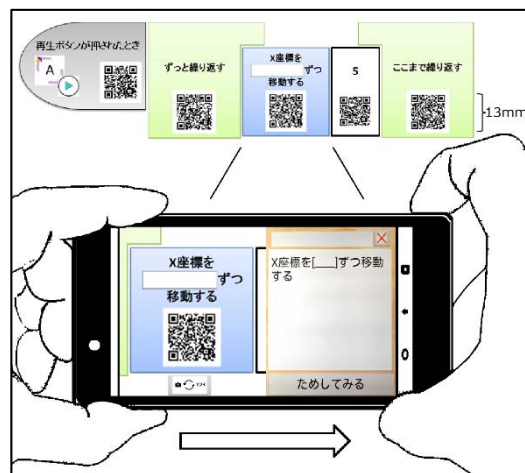


図 3 コマンドブロック

コマンドブロックの認識には、貼付された QR コードを使用する。QR コードの情報は、英字や漢字を含めた 1817 文字が格納可能である。また、最大 30% を損失してもデコードが可能であるため、認識精度が高い。図 4 に QR コードに格納された文字列を示す。図 4 で示すように本環境で扱われる QR コードには、次の三つの要素が必ず格納されている。

Program	{001001,X座標を[]ずつ移動する}[MoovX]
①ID	①
②説明文	②
③命令文	③

図 4 QR コードに格納された文字列

● ID

本環境では、複数の QR コードを読み込むため、一度読み込んだ QR コードと新しい QR コードを判別する値が必要となる。ID は、各コマンドブロックに割り振られた六桁の数字である。読み込んだコマンドブロックの ID を記録し、新しい QR コードが読み込まれた際に一致するものがないか調べることで、コマンドブロックが重複して読み込まれることを避けている。

● 説明文

本環境では、QR コードを一つずつ読み取るため、現在カメラを向けている QR コードが読み込めたかを確認する方法が必要となる。説明文は、自身が貼付されているコマンドブロックの働きを記述したものであり、この内容を画面に表示することによって、学習者が、QR コードが読み込めたかを確認できるようにした。

● 命令文

命令文は、実行時にスプライトやプログラムをどのように動かすかを記述した文字列である。この文字列を読み取った順番で配列に格納し、実行が行われる際には、配列の先頭から順番に命令文に沿った処理が実行される。命令文の文字列は、行う処理の種類を表した単語と処理で 사용되는式や数値の二種類がある。式や数値といった文字列は、行う処理の数値を細かく指定する場合に使用する。

3.3 コマンドブロックの詳細

3.3.1 スタートブロック

本環境では、一つの作品に最大三つのスプライトを使用することができ、それぞれのスプライトに異なったプログラムを設定することができる。また、スプライトを複製するクローンという機能があり、複製されたスプライトは複製元と異なったプログラムを設定することができる。つまり、実行画面の再生ボタンが押された際に実行されるプログラムと、スプライトが複製された際に実行されるプログラムの二つを各スプライトに設定することができる。クローンについては、3.3.8 で説明する。

コマンドブロックを並べる場合は、最初にスタートブロックと呼ばれるブロックを置き、その右隣りに実行したい順番でコマンドブロックを並べる(図 5)。スタートブロックは、自身の右隣りに並んだコマンドブロックが、どのスプライトの動きを表したプログラムであるのかとどのようなタイミングで実行されるのかを指定する役割を持っている。並べたブロックを読み取った後に実行画面の再生ボタンを押すことで、各スプライトのプログラムが並行処理で実行される。

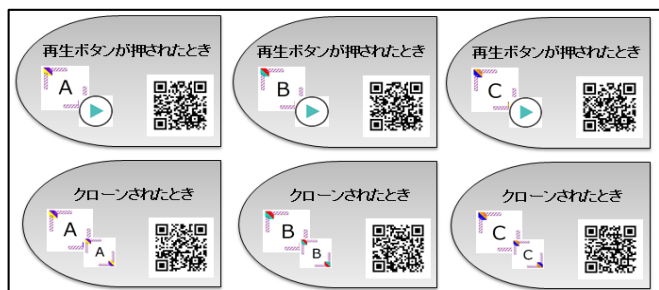


図 5 スタートブロック

3.3.2 命令ブロック

各スプライトは、X 座標、Y 座標、回転率、拡大率といった値を持っており、作品ではこの値を変更することで、動きや見た目の変化を表現している。命令ブロックはスプ

ライトの位置や見た目を変更する際に使用するブロックである(図 6)。具体例として、スプライトの X 座標を 50 ほど右に移動し、Y 座標を 0 にするプログラムを図 7 に示す。命令内容を細かく指定したい場合は、数値ブロックを使用する。数値ブロックの詳細については、3.3.3 で説明する。本環境で扱う命令ブロックは、次の 10 種類である。

● X 座標指定

スプライトの X 座標を指定した値に変更する。

● Y 座標指定

スプライトの Y 座標を指定した値に変更する。

● X 座標移動

スプライトの X 座標を相対的に移動する。

● Y 座標移動

スプライトの Y 座標を相対的に移動する。

● 方向移動

スプライトの回転率から進む方向を算出し、移動する。

● 回転

スプライトを時計回りに回転させる。

● 跳ね返る

画面の端についたとき、スプライトを 180 度回転する。

● 拡大、縮小

スプライトのサイズを変更する。

● 透明化

スプライトを透明にして見えなくする。

● 表示

スプライトの透明化を解除する。



図 6 命令ブロック



図 7 X 座標を 50 ほど右に移動し、Y 座標を 0 にする

3.3.3 数値ブロック

コマンドブロックの内容を細かく指定する方法として、数値ブロックを並べる方法とスマートフォンの画面から入力する方法が用意されている。数値ブロックは、0 から 9 の数字を表したブロックの他にも、演算子、乱数、スプライトが持つ値などを表したブロックがあり、26 種類のブロックがある(図 8)。数値を指定したいコマンドブロックを並べた後に数値ブロックを並べることで命令内容を細かく指定することができる。具体例として、スプライトの回転

率を現在の値から 90 度増やすプログラムを図 9 に示す。

スマートフォンの画面から数値の入力を行う方法では、数値の指定を行いたいコマンドブロックを読み込んだ後に画面左下の「きりかえボタン」を押すことで、数値入力用の画面に切り替えることができる (図 10)。



図 8 数値ブロック



図 9 回転率を現在の値から 90 度増やすプログラム

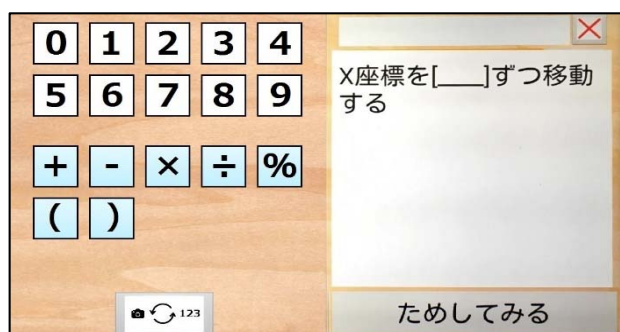


図 10 数値入力画面

3.3.4 変数ブロック

本環境では、変数を使用して数値を記憶し必要な時に利用することができる。実行開始直後は、各変数には 0 が格納されており、学習者は変数代入ブロックを使用することで自由な値を変数に持たせることができる (図 11)。

変数は、数値ブロックと同様に変数ブロックをコマンドブロックの後に並べることで、コマンドブロックの内容を細かく指定することができる (図 12)。



図 11 変数代入ブロックと使用例



図 12 変数ブロック

3.3.5 プログラムの一時停止と終了

本環境では、スプライトに対する命令の他にプログラムの一時停止、繰り返し、条件分岐、終了といったプログラムに対する処理が割り当てられたコマンドブロックがある。図 13 にプログラムの一時停止と終了を行うブロックを示し、その使用例を示す。

プログラムの一時停止を行うブロックは、指定された秒数だけプログラムの実行を停止し、その後にプログラムを再開するブロックである。

プログラムを終了するブロックは、実行されているすべてのスプライトのプログラムを終了して再生ボタンを押す前の状態へと戻すブロックである。



図 13 プログラムの停止と終了を行うブロックと使用例

3.3.6 繰り返しブロック

指定した範囲のプログラムを繰り返したい場合は、繰り返しブロックを使用する (図 14)。スプライトの移動を 10 回繰り返すプログラムを図 15 に示す。プログラムの繰り返しは、「____回繰り返す」と記述されたブロックと「ここまで繰り返す」と記述されたブロックの二枚のブロックを使用して表現する。繰り返したい範囲のプログラムをこの 2つのブロックで挟むことにより、挟まれた範囲のプログラムが指定回数繰り返し実行される。繰り返しの回数は、繰り返し開始ブロックの後に数値ブロックを並べることで指定することができる。



図 14 繰り返しブロック



図 15 スプライトの移動を 10 回繰り返すプログラム

3.3.7 条件分岐ブロック

条件に応じて実行するプログラムを分けたい場合は、条件分岐ブロックと条件記号ブロックを使用する（図 16）。条件分岐ブロックを使用した具体例を図 17 に示す。図 17 は、X 座標が 50 より大きい場合は、X 座標を 0 に設定し、それ以外の場合は X 座標を 3 ずつ移動するプログラムである。

「もし____なら」と記述されたブロックは、条件分岐の始まりを意味しており、その後に続く、数値ブロック、条件記号ブロック、変数ブロックが条件式を表している。条件が満たされている場合は、条件式を除いた「もし____なら」と記述されたブロックから「でなければ」と記述されたブロックまでのプログラムが実行される（図 17, ①）。条件が満たされていない場合は、「でなければ」と記述されたブロックから「ここまで」と記述されたブロックまでのプログラムが実行される（図 17, ②）。また、「もし____なら」と記述されたブロックの代わりに、接触分岐ブロックを用いることで、スプライト同士が接触したかを判定し、その結果で実行するプログラムを分岐させることができる（図 18）。

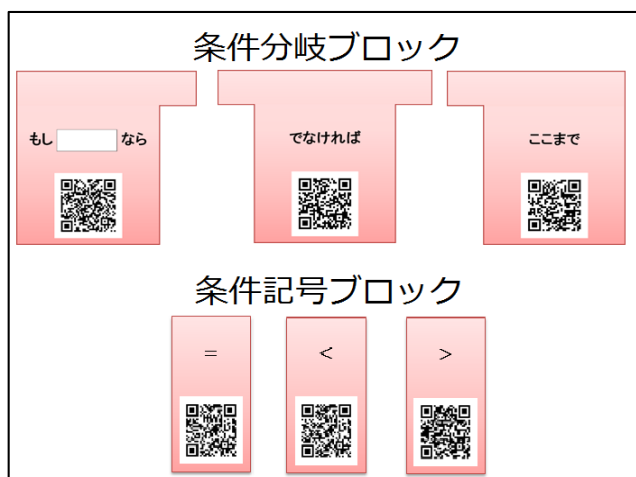


図 16 条件分岐ブロックと条件記号ブロック

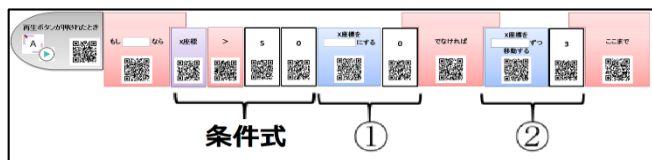


図 17 条件分岐ブロックの使用例



図 18 接触分岐ブロック

3.3.8 クローンブロック

実行中にスプライトを複製したい場合は、クローンブロックを使用する（図 19, 左）。複製されたスプライトは元のスプライトと同じ見た目だが、異なるプログラムを設定することができる。複製されたスプライトのプログラミングは、クローン用のスタートブロックの後にコマンドブロックを並べることで行う（図 20）。また自身を画面から削除したい場合は、削除ブロックを使用する。（図 20, 右）



図 19 クローンブロックと削除ブロック

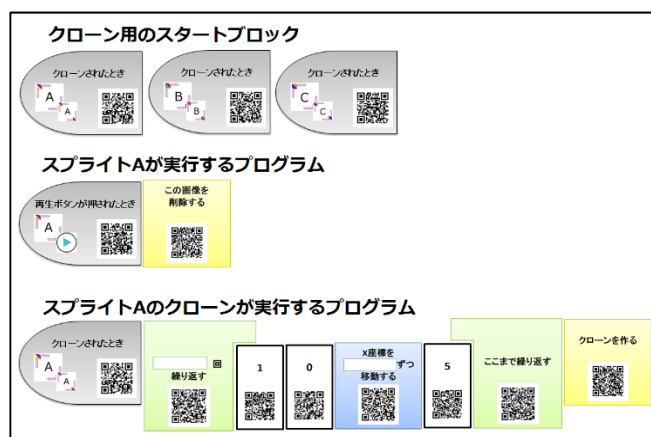


図 20 クローン用のスタートブロック

4. 議論

4.1 カメラを用いた認識方法について

QR コードを貼付したブロックとスマートフォンのカメラを使用することで、画面で行う操作を最低限に抑えたプログラミング学習環境を示した。ブロックには QR コードが貼付されており、スマートフォンに標準で搭載されているカメラによって、一つずつ撮影することで命令の認識を行っているが、全てのブロックを読み取るためには時間と手間がかかってしまう。特にプログラムを作り直すたびに、一から読み直す必要があるため、学習者が快適にプログラミングを行うことができない。

修正したブロックのみを読み取るシステムを追加することで、時間を大幅に短縮することができるが、ブロックの追加、削除、変更などの画面操作を増やす必要があるため、本システムの操作を最低限に抑えるというコンセプトに反することになる。

複数のブロックを一度に読み取ることができ、認識制度が高くなる認識方法について検討を今後行う必要がある。

4.2 ブロックの使用数について

本学習アプリは、Scratchのように自由な作品を制作しながらプログラミングを学ぶことができるが、作品が複雑なものになるほど使用するブロックが多くなる問題があり、ブロックが足りなくなるのを防ぐには、大量のブロックを用意する必要がある。

ブロックを読み取った際の QR コードの回転度合を検出するシステムを追加することで、ブロックの向きによって違った命令内容を持たせることが可能となり、不必要なブロックを減らすことができるが、学習者のスマートフォンが回転した状態で読み込んだ場合、学習者の意図していない結果が表示される事態が発生すると考えられる。

少ないブロックで豊かな表現を可能にする言語デザインを実現するか。回転度合の検出を行った際に誤認識が発生しにくいシステムを考える必要がある。

5. まとめ

本稿において、QR コードを貼付したブロックとスマートフォンのカメラを使用することで、画面で行う操作を最低限に抑えたプログラミング学習環境を提案し、その実装を行った。実装したシステムは、ブロックを並べることによってプログラミングが可能な環境である。今後の課題として、ブロックを読み取る場合の問題点や少ないブロックで豊かな表現が可能な言語デザインについての考察などが挙げられた。その後、評価実験を行い有効性について検討を進め、改善を行う予定である。

参考文献

- [1] “小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議 議論の取りまとめ”。
http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm, (参照 2017-01-25)。
- [2] “Scratch ホームページ”. <https://scratch.mit.edu/>, (参照 2017-01-25)。
- [3] “プログラミン ホームページ”。
<http://www.mext.go.jp/programin/>, (参照 2017-01-25)。
- [4] 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. 日本教育工学会論文誌, 2010, Vol. 34 no. 4 p. 387-394
- [5] 深谷和義, 宮地晶子. 小学生向けプログラミング授業のための「プログラミン」利用の検討. 日本教育工学会論文誌, 2012, Vol. 36 no. Suppl. p. 9-12
- [6] “平成 28 年版情報通信白書”。
<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/pdf/n3200000.pdf>, (参照 2017-01-25)。
- [7] “The Foos ホームページ”. <http://thefoos.com/>, (参照 2017-01-25)。
- [8] “Digital Puppet アプリダウンロードページ”。
<https://play.google.com/store/apps/details?id=com.takoyaking.digitpuppet&hl=ja>, (参照 2017-01-25)。
- [9] “Pyonkee ホームページ”。
<http://www.softumeya.com/pyonkee/ja/>, (参照 2017-01-25)。
- [10] “ScratchJr ホームページ”. <https://www.scratchjr.org/>, (参照 2017-01-25)。

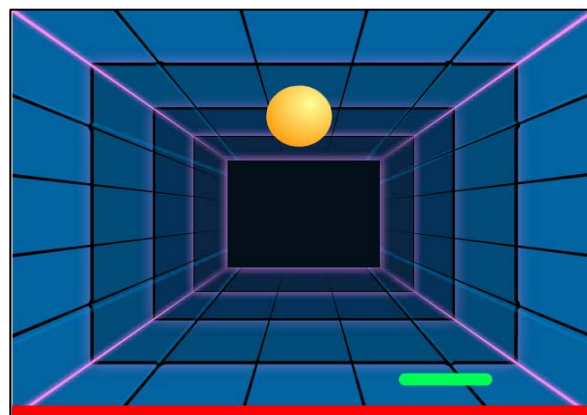
- [11] 八城朋仁. タンジブルなプログラミング学習環境の要件の検討. 情報教育シンポジウム 2015 論文集, 2015, Vol. 2015 p. 207-213
- [12] 鈴木栄幸, 加藤浩, 伊東祐司. 実体を持つプログラム言語「アルゴブロック」: 思考と共同作業の道具として. 日本教育工学会論文誌, 1993, vol. 46 p. 9-10
- [13] Matsuzaki, S., Takimoto, M., Kambayashi, Y., Design of Tangible Procedural Programming of Robots Based on Augmented Reality, Proceedings of the 10th International Conference on Computer Graphics Theory and Applications, 2015, Vol. 1 p. 492-497

付録

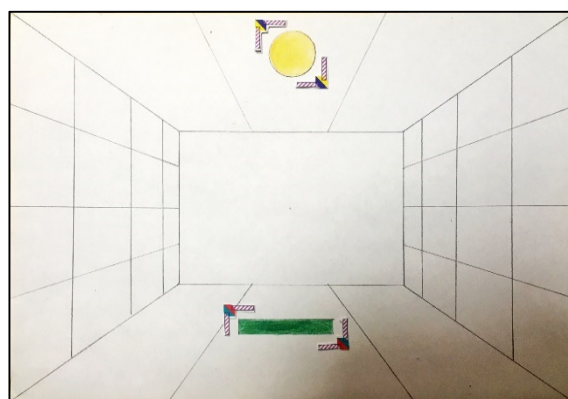
付録 A.1 使用例

本環境の実行例として、公開されている Scratch の作品を本環境で再現した場合のイラストとプログラムを示す。本環境で実装できていないブロックについては、その他のブロックを用いて再現可能ならば別のブロックで代用し、再現不可能な場合は、再現する対象に含めないものとする。

再現した作品の情報	
作品名	Pong Starter
作者	Scratchteam
URL	https://scratch.mit.edu/projects/10128515/
詳細	Scratch が初心者向けに公開している作品例の一つ。画面上を反射しながら移動するボールを、画面下部に落ちないように、バーを左右に操作して打ち返すゲーム。



Scratch のイラスト



本環境のイラスト

