

平成 24 年度 公立はこだて未来大学卒業論文

物質プログラミング
— 物質によるプログラムの可視化と開発環境の制作 —

八城 朋仁

情報アーキテクチャ学科 b1009182

指導教員 迎山 和司

提出日 2013 年 1 月 31 日

Material Programming
— A visual programming development environment with
material —

by

Tomohito YASHIRO

BA Thesis at Future University Hakodate, 2013

Advisor: Kazushi MUKAIYAMA

Department of Media Architecture

Future University Hakodate

January 31, 2013

Abstract– In this study, I created a programming development environment that allows programming of the material by the tile to make it easier to use the program more. With the popularization of computers, as well as professional acts programming has to be carried out widely. In programming has been incorporated as part of the class of information education, there is a language called Scratch for programming education for low age. There is a point that is also easy to use and easy for beginners that can be programmed in one of the concepts of Scratch, as building blocks by hand in an intuitive way. So, I have to recreate blocks of Scratch by the material. I have created a programming development environment that can be a combination of tile and programming. Programming development environment was created using a marker for image processing and computer, the USB camera. I exhibited a programming development environment that produced. And I have observed the appearance of those who experience programming development environment. Then, it was seen how the people who experience while inserting or removing a tile to the shooting range of USB camera, do the trial and error of building a program. However, people who experience seemed unable to grasp the shooting range of the USB camera. And I know I will need to consider the shape and material because there is a difference in the reaction of the user in the shape and material of the tiles. Finally, with respect to the operation of the program, I get a good reaction against that change the behavior of the program by the tile. However, I could not get a reaction to the behavior program. I think there is a limit for operation that can be seen in the exhibits, there was no fun to operate. Therefore, it's important that there be a teaching materials and exercises for people to experience to be able to build their own program.

Keywords: Scratch, Visual programming, USB camera

概要: 本研究ではプログラムをより使いやすくするために物質のタイルによってプログラミングが可能なプログラミング環境を制作する。コンピュータの普及に伴い、プログラミングは一部の専門職の行為だけではなく、広く一般に行われるようになってきた。プログラミングが情報教育の授業の一環として取り入れられている中で、低年齢向けプログラミング教育のための Scratch という言語がある。Scratch のコンセプトのひとつとして、直感的に手でブロックを組み立てるようにプログラミングできるため初心者にも簡単に使いやすいという点がある。そこで、本研究では Scratch のブロックを現実の物質によってタイルとして再現し、タイルを組み合わせることでプログラミングができる環境の制作を行った。プログラミング環境は、USB カメラとコンピュータ、そして画像処理のためのマーカを用いて実装した。制作したプログラミング環境は展示を行い、体験者の様子を観察した。すると、体験者が USB カメラの撮影範囲にタイルを出し入れしながら、プログラムの構築の試行錯誤を行う様子が見られた。しかし、体験者が USB カメラの撮影範囲を把握できていない様子が見られた。さらに、タイルの形状や材質によってユーザの反応に違いがあり、形状と材質について十分考慮する必要があると分かった。最後に、構築されたプログラムの動作に関して、タイルによってプログラムの動作が変化することに対しては良い反応を得られたが、動作については反応が得られなかった。これは、展示で見せることのできる動作には限界があり、面白さを感じなかったためと考えられる。そのため、体験者が自らプログラムを構築できるようにするための演習や教材を用意することが、今後の展望として重要と考える。

キーワード: Scratch, ビジュアルプログラミング, USB カメラ

目次

第1章	序論	1
1.1	目的	1
1.2	背景	1
第2章	関連研究	2
2.1	LOGO	2
2.2	Scratch	2
2.3	Viscuit	3
第3章	システム	4
3.1	作品概要	4
3.2	仕様	6
3.3	作品動作概要	6
3.4	実装方法	7
3.4.1	USBカメラ入力と画像解析	7
3.4.2	物質による命令タイルの制作	8
3.4.3	入力されたプログラムの構築と実行	10
第4章	展示と評価	12
4.1	Maker Faire Tokyo 2012	12
4.1.1	展示内容	12
4.1.2	観察結果	12
4.2	公立はこだて未来大学オープンラボ	13
4.2.1	展示内容	13
4.2.2	観察結果	13
4.3	平成24年度公立はこだて未来大学情報デザインコース卒業研究展示会	14
4.3.1	展示内容	14
4.3.2	観察結果	14
第5章	考察	16
5.1	命令タイルによるプログラミング	16
5.2	実行結果	17
第6章	結論と今後の方針	18
6.1	結論	18
6.2	今後の方針	18

第1章 序論

本研究の目的と、それに至った背景を以下に述べる。

1.1 目的

プログラムをより使いやすくするために物質のタイルによってプログラミングが可能な環境を制作する。プログラミング言語として Scratch を対象とし、Scratch のブロックを物質のタイルとして再現する。プログラミング言語をデジタルデータではなく、現実の物質によって作成されたタイルを用いてプログラムを構築できるようにする事で、より使いやすいプログラミング環境の制作を行う。

1.2 背景

コンピュータの普及に伴い、プログラミングは一部の専門職の行為だけではなく、広く一般に行われるようになってきた。そのため、専門知識の無い人々でも使いやすいプログラミング言語が必要となってきている。プログラミングは専門知識としてだけでなく、大学の情報教育の授業の一環として取り入れられており、授業の中では使いやすい教育用のプログラミング言語が使用されている [1]。使いやすいプログラミング言語として、低年齢向けプログラミング教育のための Scratch というものがある。Scratch はビジュアルプログラミング言語のひとつである。ビジュアルプログラミング言語とはプログラミングの要素を文字ではなく、コンピュータの画面上でグラフィックによって指定し、視覚表現によってプログラミングが可能なプログラミング言語である。Scratch のコンセプトのひとつとして、直感的に手でブロックを組み立てるようにプログラミングできるため、初心者にも簡単に使いやすいと言う点がある。Scratch はプログラムの要素ひとつひとつがブロックのようなグラフィックで表されている [2]。様々な教育での実践報告があり、使いやすさが実証されている [3]。しかし、あくまでもマウスによる入力を主としており、ブロックを直接手で組み合わせるわけではない。プログラミングはコンピュータによる行為だが、紙のノートやペンなどの現実の物質の方が使いやすい行為もある。長塚隆らの研究では授業におけるノートテイキングでは、電子テキストへのノートテイキングより、紙でのノートテイキングの方が利用しやすいと感じている者が多いという結果が出ている [4]。ノートテイキングとはノートをとることである。本研究では Scratch の中でプログラムの要素を表しているブロックのグラフィックを現実の物質で再現することにより、より使いやすいプログラミング環境の制作を目指す。

第2章 関連研究

本研究に関連する先行研究を紹介する。

2.1 LOGO

LOGO とはシーモア・パパートによって開発された画面上に映るタートルと呼ばれるものと交流するためのコンピュータ言語の総称である。タートルとはコンピュータの画面に表示される三角形の呼称であり、キーボードでコマンドを入力することで操作できる。タートルは移動した軌跡に線を描画することができる。線を描画する機能を用いて様々な図形や絵を描くことができる [5].

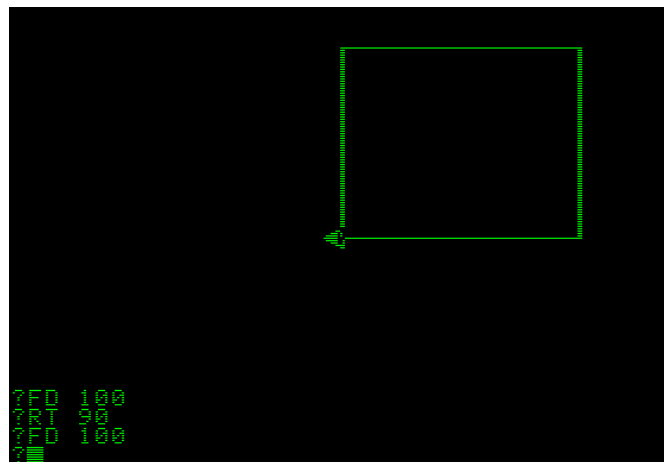


図 2.1: LOGO [6]

2.2 Scratch

Scratch とは MIT メディアラボが開発したプログラミング言語で、画面上で絵と絵に対する命令を表したブロックを組み合わせることでアニメーションなどを作成できる。子ども向けのワークショップが多く開催されており、その中で制作された作品はオフィシャルサイトで共有することができる。Scratch は画面上のスプライトと呼ばれる絵に対してプログラミングを行う。スプライトの動き、繰り返しの制御、条件分岐などのブロックが存在し、ブロックを組み合わせることでプログラミングすることができる [2].



図 2.2: Scratch

2.3 Viscuit

Viscuitとは原田康徳によって開発されたプログラミング言語であり、絵を描き、描いた絵を配置するだけでアニメーションなどを作ることができる。国内で多くのワークショップが開催されている。描かれた絵を2枚を配置することでアニメーションを作成できる。アニメーションの速さや移動方向は、2枚の絵の位置関係から決定される。絵の組み合わせを複数使用することで複雑な動きのアニメーションを作成できる。Viscuitは絵を描き、それを配置するだけなので必ずしもキーボードやマウスが必要なわけではない。ワークショップでは、タブレットPCのタッチパネルが利用されることもあり、より直感的な操作を実現している [7].



図 2.3: Viscuit [8]

第3章 システム

物質プログラミングの仕様を説明する。

3.1 作品概要

物質プログラミングは、物質のタイルの組み合わせによってプログラムを構築できるプログラミング環境である。物質のタイルは、Scratch の命令を表したブロックを物質によって再現したものである。以下、命令タイルとする。コンピュータの画面には車の絵が描画されており、実行ボタンを押すと、コンピュータは構築されたプログラム通りに車を移動させる。実行ボタンが押される前は、コンピュータの画面の背景に USB カメラによって撮影された画像が描画されている。コンピュータへの命令は4種類あり、それぞれ「車を進める」、「車の向きを回転する」、「繰り返す」、「変数」である。また、車の移動速度、回転速度、繰り返しの回数は「変数」の命令によって変化する。従って、「車を進める」、「車の向きを回転する」、「繰り返す」の命令は「変数」の命令と組み合わせることで車を移動させる命令となる。命令タイルの組み合わせは、USB カメラの撮影範囲に命令タイルを置くことによってコンピュータに読み込まれる。そして、USB カメラの撮影範囲に置かれた命令タイルからプログラムが構築される。構築されたプログラムは実行ボタンが押されるとコンピュータによって実行される。実行ボタンはコンピュータに付属しているポインティングデバイスを利用している。

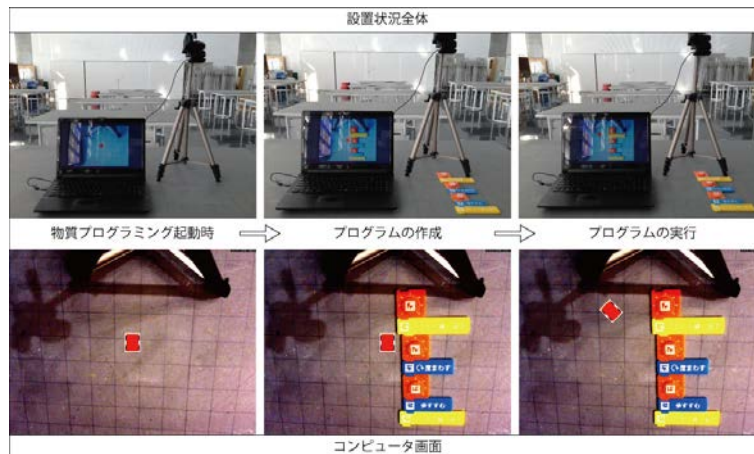


図 3.1: 作品動作風景



図 3.2: 命令タイル



図 3.3: 命令タイルの組み合わせ



図 3.4: プログラム例

3.2 仕様

本作品には、USB カメラと PC を使用した。USB カメラは Logicool HD webcam C525 を使用した [9]。USB カメラは命令タイルの置く位置から 47cm 上に設置した。USB カメラを設置する高さは、命令タイルを読み取りの精度が低くならず、かつ USB カメラの撮影範囲が大きくなるように調整した。USB カメラは周りの環境から光の影響を受けるため事前に環境に合わせたパラメータの設定を手動で行う。USB カメラの撮影サイズは横 1280 ピクセル、縦 960 ピクセルで撮影間隔は 60 フレーム毎秒とした。

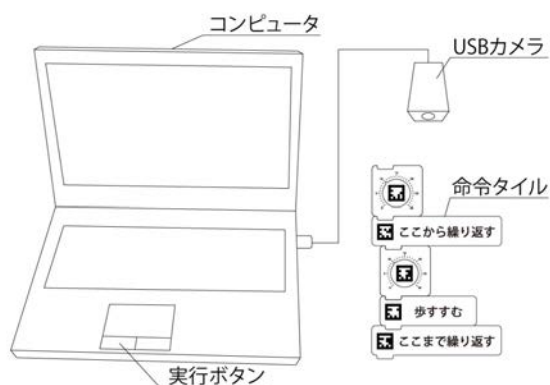


図 3.5: 物質プログラミング接続図

3.3 作品動作概要

USB カメラを用いて、命令タイルの組み合わせが写っている画像を入力すると、コンピュータによってプログラムの構築準備がされる。プログラムは実行ボタンを押すことで構築され、実行結果が PC の画面上に車の動きとして出力される。実行ボタンは PC のポインティングデバイスを使用する。コンピュータで行われる処理は Processing1.5 によって遂行される。USB カメラによって入力された画像を画像処理する。画像処理により、命令タイルの種類と位置情報が得られる。USB カメラからの入力と画像処理は毎秒 60 回行われる。実行ボタンが押されると、画像処理によって得られた情報から、車の動きの流れを決定する。決定された動きの流れは順番に実行されていく。車の動きは 60 分の 1 秒間隔で実行される。車の動きの実行が全て終了するまで USB カメラからの入力と画像処理は行われず、また実行ボタンも動作しない。USB カメラからの入力の停止に伴い、コンピュータの画面の背景は実行ボタンが押されたときのまま更新されなくなる。車の動きの実行が終了すると、USB カメラからの入力と画像処理が再開される。

3.4 実装方法

実装方法については、USB カメラ入力と画像解析、物質による命令タイルの制作、入力されたプログラムの構築と実行に分けて説明する。

3.4.1 USB カメラ入力と画像解析

命令タイルで構築されたプログラムの PC への入力は USB カメラによって行う。USB カメラでリアルタイムに撮影し、撮影された画像から画像処理によって命令タイルの情報を抽出する。命令タイルの情報を抽出するために、四角形のマーカを使用する。そのため、命令タイルには白色でマーカが付けられている (3.4.2 参照)。画像処理では、マーカによって命令タイルの認識を行う。マーカによる認識は精度の高さから選択した。認識された命令タイルから命令タイルの情報の抽出を行う。命令タイルの情報とは、命令タイルに付けられたマーカの ID と、USB カメラで撮影された画像内でのマーカの位置情報である。また、マーカの位置情報にはマーカの 4 角の位置情報が含まれる。画像処理には NyAR4psg というライブラリを利用している。NyAR4psg とは、加藤博一らが開発した ARToolkit を Processing に移植したものである [10] [11]。NyAR4psg を利用した画像処理ではマーカを認識させるために、マーカの一部を黒色にする必要がある。そのため、画像処理を行う前に USB カメラで撮影した画像の色を反転させる。そして、色を反転させた画像に対して、NyAR4psg を利用した画像処理を行う。マーカの形状と認識できるマーカの数には条件があり、これは NyAR4psg の仕様からきている。はじめに、マーカの形状の条件とはモノクロ画像であること、正方形であること、マーカは黒い太枠の中に描画すること、太枠とマーカが接しないこと、マーカは左右対称または上下対称にならないように工夫することである。この条件を満たせばオリジナルのマーカの作成が可能であるが、本研究ではマーカの使用が容易である NyID マーカを使用した。NyAR4psg では、マーカの認識のためにそのマーカの情報を事前にパターンファイルとして登録する必要があるが、NyID マーカを利用すればパターンファイルが必要ない。次に、認識できるマーカの数についてだが、画像処理を行う際に、USB カメラで撮影された画像の中に同じマーカが 2 つ以上ある場合、同時に認識できず作品の動作が停止する。これを解決するために、同じ種類の命令タイルでもそれぞれに違う NyID マーカを使用している。

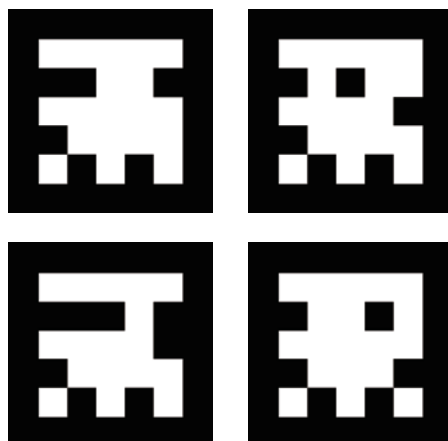


図 3.6: NyID マーカ

3.4.2 物質による命令タイルの制作

命令タイルを3色の不透明なカラーアクリル板を用いて制作した。命令タイルには「歩すすむ」、「度まわす（右まわり）」、「度まわす（左まわり）」、「ここから繰り返す」、「ここまで繰り返す」、「変数」の6種類がある。命令タイルの形状と色はScratchのプログラムの要素を表したブロックを参考にした。参考にしたブロックは「○歩動かす」、「○度回す（右回り）」、「○度回す（左回り）」、「○回繰り返す」、「変数」である。「○」には数字が入り、キーボードによる入力、もしくは変数のブロックを組み込むことで決定される。参考にしたブロックの色は、ブルー、クロームイエロー、マンダリンオレンジの3色のカラーアクリル板によって再現した。命令タイルの制作には、レーザー加工機を用いた。カラーアクリル板はレーザー加工機による加工のしやすさから選択した。命令タイルの形状は角丸の四角形の左下が凸部、左上は凹部になっている。凸部、凹部は同じ形状で、はまる様になっている。さらに、命令タイルの裏側に磁石を付け、はめたときに力を加えなければ離れない様にした。命令タイルの厚みは2 mm, 4 mm, 6mmの厚みのものを試作し、一番持ちやすいと感じた6 mmとした。6 mmの厚さを作るために、3 mmのアクリル板を2枚重ねて接着した。その際、透明なアクリル板を使用すると接着面が透けて見える。接着面は気泡が入りやすく、綺麗に見えない場合が多いため、不透明なカラーアクリル板を使用した。命令タイルにはNyIDマーカと種類を表す文字や記号を彫刻した。さらに、彫刻だけでは見えにくかったため、彫刻した溝に塗料を流し込み着色した。塗料の色もまたScratchのブロックに書かれている文字や記号の色を参考にし、白色とした。NyIDマーカの大きさは、命令タイルの大きさに合わせ、1辺を17mmとした。

6種類の命令タイルについて説明する。はじめに、全ての命令タイルに共通でNyIDマーカを彫刻した。ただしNyIDマーカは画像処理のために、同じ種類の命令タイルであっても異なるものを彫刻する(3.4.1参照)。次に、それぞれの命令タイルについて説明する。「歩すすむ」命令タイルは「車を進める」命令を表した命令タイルである(図3.7)。「歩すすむ」命令タイルには「歩すすむ」と文字を彫刻した。「度まわす（右まわり）」、「度まわす（左まわり）」命令タイルは「車の向きを回転する」命令を表した命令タイルである

(図 3.8)(図 3.9).「度まわす (右まわり)」、「度まわす (左まわり)」命令タイルには、「度まわす」という文字を彫刻した。さらに、左回りか右回りかを表すために矢印を丸めたような記号を文字の左隣に彫刻した。「ここから繰り返す」、「ここまで繰り返す」命令タイルは「繰り返す」命令を表した命令タイルである(図 3.10)(図 3.11)。「ここから繰り返す」命令タイルには「ここから繰り返す」という文字を、「ここまで繰り返す」命令タイルには「ここまで繰り返す」という文字を彫刻した。「ここから繰り返す」命令タイルの凸部と、「ここまで繰り返す」の凹部は他の命令タイルに比べ、右側にずれている。このずれはインデントを再現したものである。インデントとは、プログラムの構造を見やすくするために行われる字下げである。「変数」命令タイルは、「車を進める」、「車の向きを回転する」、「繰り返す」命令の度合を決める「変数」命令を表した命令タイルである(図 3.12)。「変数」命令タイルは2つのパーツによって構成されている。1つ目は、1箇所から三角形が突出している円形のパーツである。さらに、円形のパーツの中心にはNyID マーカを彫刻した。2つ目は他の命令タイルと同様の凸凹部を持つ角丸の四角形のパーツである。円形のパーツは四角形のパーツの中央に配置する。四角形のパーツには中央に配置した円形のパーツに沿って、放射状に目盛を彫刻した。目盛は配置した円形のパーツを中心として9度ごとに彫刻した。目盛の数は30個で、四角形パーツ下部の90度分の目盛は彫刻していない。さらに、左下の目盛を0として、0、15、20、25、30にあたる目盛付近にそれぞれの数字を彫刻した。中央に配置した円形のパーツは目盛が彫刻されている範囲を回転し、円形のパーツの突出部が目盛を指し示す様になっている。



図 3.7: 「歩すすむ」命令タイル

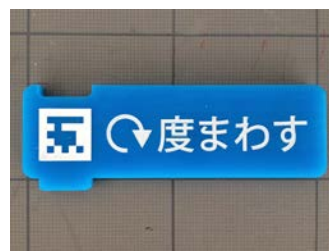


図 3.8: 「度まわす (右まわり)」命令タイル



図 3.9: 「度まわす (左まわり)」命令タイル



図 3.10: 「ここから繰り返す」命令タイル



図 3.11: 「ここまで繰り返す」命令タイル



図 3.12: 「変数」命令タイル

3.4.3 入力されたプログラムの構築と実行

実行ボタンが押されたとき、コンピュータは USB カメラによる入力と画像処理によって抽出された命令タイルの情報からプログラムの構築と実行を行う。抽出された命令タイルの情報とは命令タイルに付けられた NyID マーカの ID と USB カメラで撮影された画像内での NyID マーカの位置情報である (3.4.1 参照)。命令タイルの情報は ID 順に抽出し、保存される。はじめに、プログラムの構築の準備として、命令タイルの情報をソートする。ソートは NyID マーカの USB カメラで撮影された画像内の位置情報から行われる。命令タイルの情報は USB カメラで撮影された画像内のより上に近い順にソートされる。次に、ソートされた命令タイルの情報から、プログラムの構築を行う。NyID マーカの ID から命令タイルの種類を判別し、位置情報によってソートされた順に命令を保存していく。以上が、プログラムの構築の流れとなる。次に、構築したプログラムの実行準備として、「変数」命令タイルの情報の変換と繰り返す部分の展開を行う。「変数」命令タイルの情報の変換は、「変数」命令タイルの情報がある場合に行われる。NyID マーカの角度から、その「変数」命令タイルが表している数値を計算し、計算した数値を保存する。角度は NyID マーカの左上と右上の位置情報を利用して計算する。そして、計算した角度から「変数」命令タイルの円形のパーツが指し示す目盛の数値に変換する。変換した数字はさらに小数点以下を切り捨て、整数として保存される。次に、「ここから繰り返す」命令タイルと「ここまで繰り返す」命令タイルを検索し、繰り返しによって省略されている命令を展開する。展開は、「ここから繰り返す」命令タイルの情報の前に保存されている「変数」命令タイルの情報から変換された数値を繰り返す回数と見なして行う。「ここから繰り返す」命令タイルと「ここまで繰り返す」命令タイルは展開が終わったときに命令タイルの情報の並びから削除する。展開は「ここから繰り返す」命令タイルが無くなるまで行う。最後に、プログラムを順番に実行する。実行される命令は、それぞれの命令タイルの情報の前に並んでいる「変数」命令タイルの情報から変換された数値によって度合が変化する。「歩すすむ」命令タイルならば、車の移動距離が変化する。「度まわす (右まわり)」, 「度まわす (左まわり)」命令タイルならば、車の向きの回転速度が変化する。プログラムの実行が行われると、実行された命令に合わせて車の座標と角度を再計算し、再描画する。そして、全てのプログラムが実行されると、実行ボタンからの入力を待つ状態に戻る。

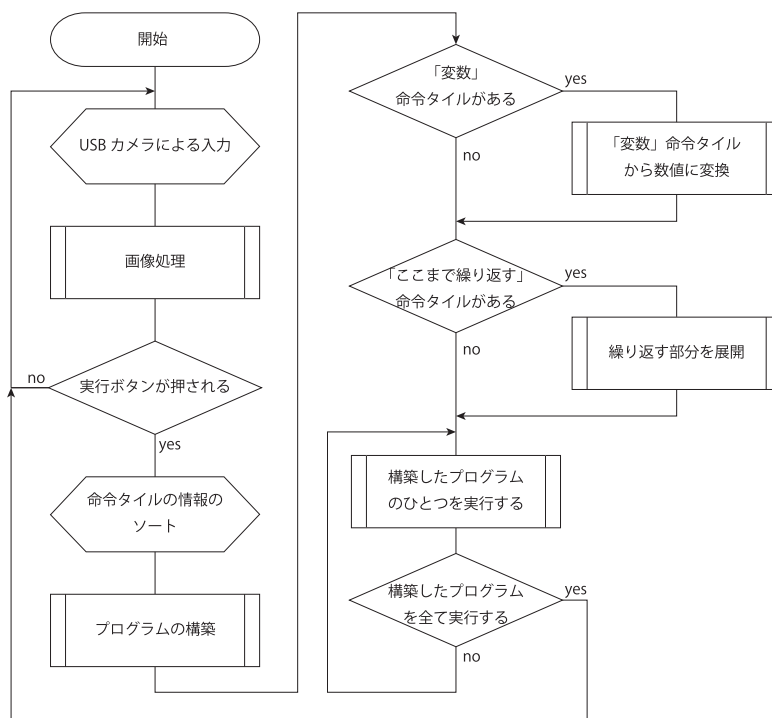


図 3.13: プログラムの構築と実行までのフローチャート

第4章 展示と評価

物質プログラミングを次に挙げたイベントに展示し、体験者から物質プログラミングの評価を得た。

4.1 Maker Faire Tokyo 2012

2012年12月1日から2012年12月2日にかけて行われたMaker Faire Tokyo 2012にて物質プログラミングを展示した。

4.1.1 展示内容

机にPCとUSBカメラを取り付けた三脚を置き、体験者に物質プログラミングの説明と実演を行い、体験させた。物質プログラミングの説明では、NyIDマーカによって命令タイルを認識していることは言わなかった。展示した物質プログラミングでは、コンピュータへの命令の種類は「車が進む」、「車の向きを回転する」のみだった。命令タイルは厚さ5mmのスチレンボードで制作したものを使用した。「車が進む」命令を表した命令タイルは「ゆっくり進む」、「進む」、「はやく進む」の3種類だった。さらに、「車の向きを回転する」命令を表した命令タイルも「小さく回る」、「回る」、「大きく回る」の3種類であり、「変数」命令と一体化したものになっていた。また、「小さく回る」、「回る」、「大きく回る」命令タイルは全て右まわりの動きを表している。プログラムの実行は実行ボタンに関係なく常に行われ、命令タイルをUSBカメラの撮影範囲に入れると、車が動き続ける状態だった。展示会場の環境に合わせてUSBカメラのパラメータを調整することができず、机に命令タイルを置いても認識ができなかった。命令タイルをUSBカメラに近づけると認識できた。そのため、体験者は命令タイルを手を持ち、USBカメラに近づけなければならなかった。

4.1.2 観察結果

物質プログラミングの実演を行うと、命令タイルをUSBカメラによって入力したときに、車が動いたことに対して驚いたような反応が見られた。実演が終わると、体験者が自ら命令タイルを手に取りUSBカメラによって入力を行う様子が見られた。車の動きを目で追いながら、命令タイルをUSBカメラの撮影範囲に出し入れする行為が見られた。さらに、体験者はコンピュータの画面に表示されているUSBカメラによって撮影された画像によって命令タイルがUSBカメラの撮影範囲に入っているかの確認を行っていた。体験者は命令タイルをUSBカメラの撮影範囲の中心に入れる傾向があった。体験者が複数で体

験を行ったとき、自分が持っている命令タイルで別の人が持っている命令タイルの NyID マーカを隠す行為が見られた。さらに、NyID マーカが手で隠れてしまったときに、命令タイルを持ち替え、NyID マーカ全体が USB カメラの撮影範囲に入る様になっていた。「変数に、物質ならではのひとひねりが欲しい。例えば、量や重さでの変数の入力はどうか。」、「バックグラウンドで動くプログラムの様子を見せると格好良い。」という意見を得られた。

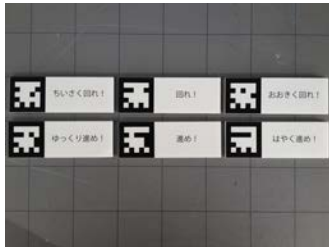


図 4.1: スチレンボードによる命令タイル

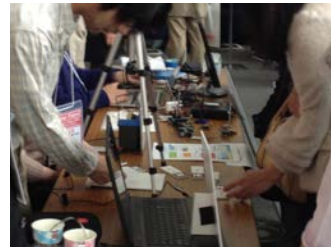


図 4.2: Maker Faire Tokyo 2012

4.2 公立はこだて未来大学オープンラボ

2012年12月17日、公立はこだて未来大学にて展示した。

4.2.1 展示内容

机に PC と USB カメラを取り付けた三脚を置き、体験者に物質プログラミングの説明と実演を行い、体験させた。プログラミングの実演では、「車を進める」命令で動かした例と、「繰り返し」命令を用いて動かした例を行った。命令タイルは制作中であったため、紙の NyID マーカに命令の種類を表す文字を書いた付箋を貼って代用した。「車を進める」命令は「すすむ」、「車の向きを回転する」命令は「回す」という文字で表した。また、「変数」命令タイルによる命令の度合も、数字ではなく文字で表した。「車を進める」命令の度合は「はやく」「おそく」、「車の向きを回転する」命令の度合は「ちいさく」「おおきく」という文字で表した。「繰り返し」命令の度合もまた「5回」、「10回」という文字で表した。

4.2.2 観察結果

プログラミングの実演を行ったとき、「車を進める」命令で動かした例では動きに気がつかない人が大半だった。「繰り返し」命令を用いて動かした例では驚いた様子が見られた。しかし、紙に印刷された NyID マーカを手にとる人はいなかった。「USB カメラの撮影範囲はどこまでか、どこにおいても入力できるのか」という質問があった。「面白い。」、「楽しそう。」という意見や、「PC の画面上だけでなく、ロボットなどの動きも制御できれば良い。」という意見を得られた。さらに、「はやく、おそく、ちいさく、おおきくという言葉ではなく、数字で変数を扱いたい。」という意見を得られた。

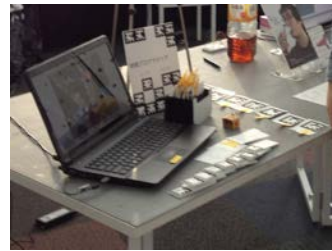
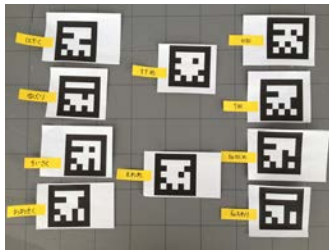


図 4.3: 紙と付箋による命令タイル 図 4.4: 公立はこだて未来大学オープンラボ

4.3 平成 24 年度公立はこだて未来大学情報デザインコース卒業研究展示会

2012 年 2 月 11 日，五稜郭タワーアトリウムにて展示した。

4.3.1 展示内容

机に PC と USB カメラを取り付けた三脚を置き，体験者に物質プログラミングの説明と実演を行い，体験させた。命令タイルはカラーアクリル板で制作したものを使用した。展示会場は日の光が強かったため，NyID マーカを USB カメラによって認識できない場合があった。その場合は，ダンボールを使って影を作り，NyID マーカが読み取れるように調整した。

4.3.2 観察結果

命令タイルによって画面上の車の動きが変わることを確認した体験者から「面白い」，「楽しい」という意見を得た。体験者が自ら「変数」命令タイルが示す数を変化させたり，命令タイルを取り替えてプログラムを再度実行する様子が見られた。命令タイルに彫刻されている文字に関して，「ここから繰り返す」，「ここまで繰り返す」命令タイルの違いを区別できない体験者がいた。さらに，「子どもと遊びたい，だから文字は無い方が良い。」という意見を得た。命令タイルの大きさに対しては，「この大きさなら小さい子とも遊べる。これ以上小さいともう少し年齢が高くないと使えない。」という意見を得た。体験者が命令タイルによってプログラミングしたとき，USB カメラの撮影範囲に余計な命令タイルが写り込んだり，NyID マーカが認識できずプログラムが正常に動作しない場合が多かった。



図 4.5: 平成 24 年度公立はこだて未来大学情報デザインコース卒業研究展示会

第5章 考察

物質プログラミングの命令タイルによるプログラミングと、プログラムされたものの実行結果について考察を行った。

5.1 命令タイルによるプログラミング

Maker Faire Tokyo 2012 での展示では、命令タイルを USB カメラの撮影範囲に出し入れする行為が見られた。また、平成 24 年度公立ほこだて未来大学情報デザインコース卒業研究展示会では、命令タイルを取り替えプログラムを再度実行したことから、命令タイルによって試行錯誤が引き出されたと考えられる。3つの展示を比較すると、命令タイルを手にする人の数に差があった。Maker Faire Tokyo 2012 ではスチレンボードで命令タイルを制作し、公立ほこだて未来大学オープンラボでは紙に印刷された NyID マーカを命令タイルの代わりに使用した。また、平成 24 年度公立ほこだて未来大学情報デザインコース卒業研究展示会ではカラーアクリル板で制作した。Maker Faire Tokyo 2012, 平成 24 年度公立ほこだて未来大学情報デザインコース卒業研究展示会での展示では、体験者が自ら命令タイルを手にとった。公立ほこだて未来大学オープンラボでは自ら紙の NyID マーカを手にとる人はいなかった。ここから命令タイルは紙で制作するより、厚みのある立体の方が手にとりやすいと考えられる。また、紙という材質の脆さに対して、体験者は理解していたと予想できる。従って、命令タイルの材質は一般の人が丈夫だと認識しているものが良いと考えられる。しかし、丈夫な材質であれば何でも良いとは言えない。材質の重さや価値、静電気の発生率など様々な要因について考えなければならない。平成 24 年度公立ほこだて未来大学情報デザインコース卒業研究展示会で使用した命令タイルの大きさに関しては、体験者が自ら手にとる様子が見られた事と、「この大きさなら小さい子とも遊べる。これ以上小さいともう少し年齢が高くないと使えない。」という意見を得た事から、適切だったと考えられる。

3つの展示会では全てで USB カメラの撮影範囲を気にする様子が見られた。また、命令タイルによってプログラミングをしたときに、余計な命令タイルが写りこんでしまう場合が多かった。コンピュータの画面に写っている USB カメラが撮影した画像を見れば、USB カメラの撮影範囲を把握できると考えていた。しかし、実際は USB カメラの撮影範囲を常に気にしながら、撮影範囲の一部しか使用しないことが分かった。体験者が USB カメラの撮影範囲を有効活用できる工夫が必要だと分かった。今後は命令タイルやコンピュータの画面に表示させるものだけではなく、机などの周りの環境についても考える必要がある。また、画像処理で、命令タイルの並び方から余計な命令タイルを認識しない様にする方法も考えられる。「変数」命令に対して、Maker Faire Tokyo 2012 と公立ほこだて未来大学オープンラボでは「物質ならではの入力」や「文字ではなく数字で扱いたい」という

意見があった。この意見を反映したものが制作した「変数」命令タイルである。これを平成 24 年度公立はこだて未来大学情報デザインコース卒業研究展示会で展示に使用したところ、「変数」命令タイルを手にする体験者がいた。「変数」命令タイルを手にとった体験者は、命令タイルが示す数を変化させる様子が見られた。これは、「変数」命令タイルの示す数 w 変化させる方法が、手で回転部を回す事だったためと考えられる。手で何かを回し、示される数を変化させるという行為は、スピーカの音量や、オーブントースターの時間、そしてガスコンロの火力など日常の中で多く見られる。そのため、「変数」命令タイルの示す数の変化の方法は体験者にとって馴染みやすく、手に取ったと考えられる。

5.2 実行結果

車が動いた時に、驚きや「面白い」という意見が得られたことから、物質によってコンピュータの実行結果が変化することに意外性があったと考えられる。車が動くことには反応を得られたが、車の動きには反応を得ることができなかった。これは展示で実演できる車の動きに限界があり、面白さを感じなかったためと考えられる。3D 映像やコンピュータゲームと比べると物質プログラミングで作ることができる車の動きは単純である。しかし、あらかじめ面白い車の動きを簡単に実現できるようにしても、それは自分でプログラムしたものではなくなってしまう。従って、これ以上複雑な車の動きのための命令は作らないほうが良いと考える。しかし、面白い動きができなければ飽きてしまい、すぐに手に取らなくなるだろう。自分が面白いと思う動きを自分で作れるようになるまでの演習方法や教材、ワークショップについて考える必要がある。そのために、物質プログラミングを用いた試験的なワークショップなどを行い、プログラミングが行われる様子を観察する必要がある。

第6章 結論と今後の方針

本研究の成果と、今後について述べる。

6.1 結論

本研究では、Scratch をより使いやすくするために命令タイルによってプログラミングが可能な環境を制作し、展示を行った。現実の物質を用いることによってプログラミングを行う上で必要であるコンピュータの動作を確認しながらプログラムの構築を行う試行錯誤を引き出した。命令タイルは手に取りやすくするために、形状や材質を十分に検討する必要があると分かった。また、体験者にとって、USB カメラの撮影範囲の把握は難しいことが分かった。USB カメラの撮影範囲が分かりやすいプログラミング環境が必要だと分かった。また、画像処理で、命令タイルの並び方から余計な命令タイルを認識しない様にする方法も考えられた。実行結果が物質によって変化することに対しては反応を得ることができた。しかし、車の動き自体には反応を得ることができなかった。従って、体験者が自分で面白い動きを実現できるようになるための演習方法や教材が必要な事が分かった。

6.2 今後の方針

命令タイルによって、プログラムの構築を行う際の試行錯誤をどれくらい引き出せるか検証を行う。Scratch を使用した時と物質プログラミングを使用した時を比較し、試行錯誤の数に違いが出るのかを検証する。そのために、物質プログラミングの機能を増やし、より Scratch に近づけていく。最後に、車の動き自体には反応を得ることができなかったことから、体験者が自ら面白い動きを作れるようになるための演習と教材が、今後の展望として重要と考える。そのために、物質プログラミングを用いた試験的なワークショップを行い、体験者の様子を観察し分析を行っていく。

謝辞

本研究の機会を与えていただき、またご指導をいただいた迎山和司准教授（公立はこだて未来大学システム情報科学部情報アーキテクチャ学科准教授）に深く感謝いたします。また、多くの助言を頂いた迎山研究室卒業研究生の井上慶彦さん、大木笙平、北山史朗さん、相内祥平さん、佐藤優大さん、成田桃子さんに感謝申し上げます。最後になりましたが、Maker Faire Tokyo2012、公立はこだて未来大学オープンラボ、平成24年度公立はこだて未来大学情報デザインコース卒業研究展示会にて、多くの貴重な意見を頂いた皆様に感謝致します。

参考文献

- [1] 大西建輔. プログラミング, 何をどう教えているか 連載開始にあたって. 情報処理, Vol. 51, No. 10, p. 1341, 2010.
- [2] RESNICK M, et al. Programming for All. *Communications of the ACM*, Vol. 52, No. 11, pp. 60–67, 2009.
- [3] 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. 日本教育工学会論文誌, Vol. 34, No. 4, pp. 387–394, 2011.
- [4] 長塚隆, 山川茜. 授業におけるノートテイキングの実態. 情報知識学会誌, Vol. 22, No. 2, pp. 57–64, 2012.
- [5] シーモア・パパート. マインドストーム—子ども, コンピューター, そして強力なアイデア. 未来社, 1982.
- [6] Syd Lexia. The terrapin logo language. <http://www.sydneylexia.com/logo.htm>.
- [7] 原田康徳. 子供向けビジュアル言語 viscuit とそのインタフェース. 情報処理学会研究報告-ヒューマンインタフェース, Vol. 2005, No. 116, pp. 41–48, 2005.
- [8] COMMUNITY DESIGN COUNCIL. cdc blog. http://www.cdc.jp/cdcblog/yuzukosho/post_199.php.
- [9] Logicool. Logicool hd webcam c525. <http://www.logicool.co.jp/ja-jp/product/7794?crid=34>, 1 2013.
- [10] 谷尻豊寿. 拡張現実を実現する ARToolkit プログラミングテクニック. カットシステム, 2008.
- [11] 橋本直. AR プログラミング —Processing でつくる拡張現実感のレシピ—. オーム社, 2012.

目 次

2.1	LOGO [6]	2
2.2	Scratch	3
2.3	Viscuit [8]	3
3.1	作品動作風景	4
3.2	命令タイル	5
3.3	命令タイルの組み合わせ	5
3.4	プログラム例	5
3.5	物質プログラミング接続図	6
3.6	NyID マーカ	8
3.7	「歩すすむ」命令タイル	9
3.8	「度まわす (右まわり)」命令タイル	9
3.9	「度まわす (左まわり)」命令タイル	9
3.10	「ここから繰り返す」命令タイル	9
3.11	ここまで繰り返す」命令タイル	10
3.12	「変数」命令タイル	10
3.13	プログラムの構築と実行までのフローチャート	11
4.1	スチレンボードによる命令タイル	13
4.2	Maker Faire Tokyo 2012	13
4.3	紙と付箋による命令タイル	14
4.4	公立はこだて未来大学オープンラボ	14
4.5	平成 24 年度公立はこだて未来大学情報デザインコース卒業研究展示会	15