

ドリトル言語における Leap Motion対応と教育的利用の可能性

島袋 舞子^{1,a)} 兼宗 進^{1,b)}

概要: 教育用プログラミング言語ドリトルの処理系を、Leap Motion 社の入力デバイスである Leap Motion の計測値を読み取れるように拡張した。現在は手のひらの座標、指の本数と4つジェスチャーを認識することができる。これらの実装とともに教育用言語からの利用可能性について報告する。

キーワード: ドリトル, プログラミング言語, Leap Motion

Possibility of Education of using Leap Motion from Dolittle.

SHIMABUKU MAIKO^{1,a)} KANEMUNE SUSUMU^{1,b)}

Abstract: We have developed Dolittle programming language environment to communicate with the Leap Motion. Now, Dolittle can communicate with the Leap Motion is coordinates of the palm, number of fingers and four gestures. We report the implementation and possibility of education of using Leap Motion from Dolittle.

Keywords: Dolittle, programming language, Leap Motion

1. はじめに

情報技術の発達により、キーボードとマウスが主流であったコンピュータへの入力装置が多様多様になってきており、携帯電話におけるフリック入力や音声認識入力、最近では人の動きを認識して入力をおこなうモーションセンサー入力といった様々な入力方式がある。このように日常生活で扱う入力装置は多様化しつつあるが、小中高での情報教育において、そのような入力装置の仕組みは扱われていない。生徒がこれら入力装置の仕組みを理解する一つの方法として、プログラミングを通してセンサーの動きを体験することがあげられる。しかし、これらの入力装置を利用したプログラムを作成しようとする、実用性が重視されたプログラミング言語を用いる必要があり、プログラミ

ング経験のない生徒にとってプログラムを作成することは困難である。

一方で、小中高で利用されているプログラミング言語としてドリトル [1] がある。ドリトルは、初等中等教育における実践から、プログラミング経験のない生徒がプログラムを作成できることが知られている [2][3][4]。

今回、モーションセンサー入力装置の1つである Leap Motion に着目し、ドリトルの処理系から、Leap Motion の値を読み取れるよう拡張をおこなった。Leap Motion は手と指の動きを赤外線によって読み取ることができる。本体の大きさが小さく、設置する場所をとらないため、新たな教育利用の可能性がある。

本稿では、ドリトルの実装と Leap Motion の教育的利用の可能性について述べる。

2. Leap Motion とその支援技術

2.1 Leap Motion

Leap Motion[5] は、キーボードやマウスに代わるデバイ

¹ 大阪電気通信大学
Osaka Electro-Communication University, Shijonawate, Osaka 575-0063, Japan

a) shimabuku.m@gmail.com

b) kanemune@acm.org



図 1 Leap Motion 本体

スを目指し、2012年にLeap Motion社から発売された入力デバイスである。Leap Motionは、手のひらにのるほどの大きさで、2基の赤外線カメラと赤外線照射LEDで構成されており、赤外線LEDに照らされた手や指を赤外線カメラで撮影し、それを画像解析することによって、3D空間での手や指の位置を検出している。赤外線センサーの認識の範囲は、本体を中心にX,Y,Z軸を取る逆三角形のドーム状になっており、25mm~600mmとなっている。Leap Motion本体の画像を図1に示す。

2.2 Leap Motion の検出する動作

Leap Motionでは、手のひらと指の向きと位置、手のひらの球体半径、指よりも細長い棒やペンといった棒状のツールの認識に加え、次の4つのジェスチャーを認識することができる。実際の手と指の動きを図2に示す。

- サークル
正面に向かって円を描いた動きを検出する(図2左上)。
- スワイプ
正面に向かって一直線方向に動かした動きを検出する(図2右上)。
- スクリーンタップ
正面に向かって突く動きを検出する。(図2左下)。
- タップ
下に向かって叩く動きを検出する(図2右下)。

Leap Motionでは、これらの機能を用いたコンピュータの操作やゲームのコントローラーの役割や、多方向から図形を観察したり、アルファベットの書き方を学ぶといった教育分野での活用など幅広い用途で使用されている。それらは公式サイトにてアプリケーションが提供されている。

2.3 従来の開発環境

Leap Motionでは開発者向けにLeap Motion Developer SDKが公開されており、開発言語としてJavaScript, Unity, C#, C++, Java, Python, Objective-Cなどを用いてアプリケーション開発をおこなうことが可能である。

ここでは例としてProcessingで記述されたプログラムを図3に示す。

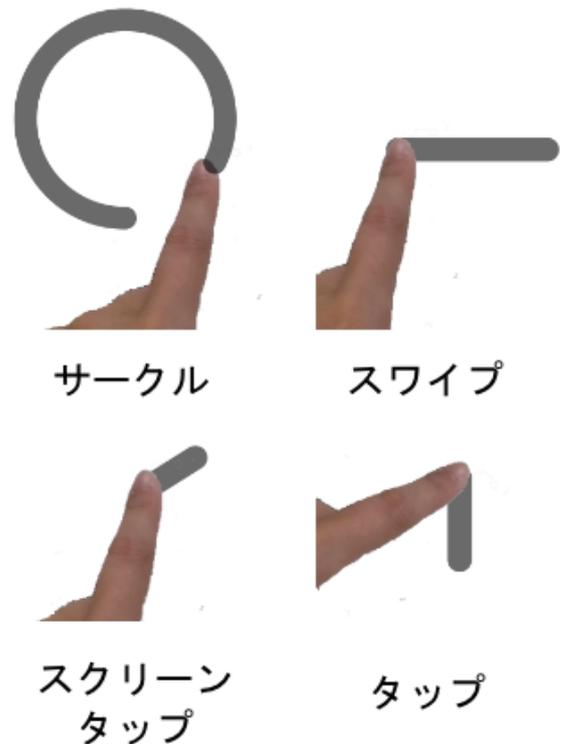


図 2 ジェスチャーの種類

```
import com.onformative.leap.LeapMotionP5;
import com.leapmotion.leap.Finger;
LeapMotionP5 leap;
public void setup() {
    size(500, 500);
    leap = new LeapMotionP5(this);
}
public void draw() {
    background(0);
    fill(255);
    for (Finger finger : leap.getFingerList()) {
        PVector fingerPos = leap.getTip(finger);
        ellipse(fingerPos.x, fingerPos.y, 10, 10);
    }
}
public void stop() {
    leap.stop();
}
```

図 3 Processing で記述したプログラム例

3. ドリトルによる Leap Motion 対応

3.1 ドリトルの Leap Motion オブジェクト

Leap Motionの開発には、これらの開発言語を利用する必要があり、プログラミング経験のない生徒が、プログラミングを通してセンサーの働きを体験することは難しい。

そこで我々はドリトルを用いて、Leap Motionから入力された値を得られるようにした。ドリトルを用いることに

表 1 対応する命令

命令	命令によって受け取る値
リープ！横の位置？	x 座標の値
リープ！縦の位置？	y 座標の値
リープ！手前の位置？	z 座標の値
リープ！指の数？	指の本数
リープ！回転？	サークルジェスチャーの値
リープ！はらった？	スワイプジェスチャーの値
リープ！突いた？	スクリーンタップジェスチャーの値
リープ！叩いた？	タップジェスチャーの値

```

リープ！接続。
表示＝フィールド！作る。
タイマー！作る 600 時間「
表示！（リープ！横の位置？）書く。
」実行。

```

図 4 Leap Motion と通信するドリトルのプログラム例

より、プログラミング経験のない生徒が Leap Motion を利用したプログラムを作成できる可能性がある。

Leap Motion が対応している機能の中から、ドリトルでは次の機能に対応した。

- 手の位置座標 (x,y,z) の値
- 指の本数の認識
- サークルジェスチャーの認識
- スワイプジェスチャーの認識
- スクリーンタップジェスチャーの認識
- タップジェスチャーの認識

Leap Motion では両手を認識することができるが、今回ドリトルではプログラム作成時の複雑化を避けるため片手のみの対応となっている。

Leap Motion からデータを取得するため、新たに Leap Motion オブジェクトを用意した。ドリトルと Leap Motion を接続するには、「接続」メソッドを用いる。

リープ！接続。

Leap Motion で認識した x,y,z 座標の値や指の本数、ジェスチャーを読むには、表 1 にある命令を実行することで、取り出すことができる。ジェスチャーの場合、値は true か false で返される。

たとえば、x の値を取得したい場合は、次の命令を実行すればよい。

リープ！横の位置？。

図 4 に x の値を表示するサンプルプログラムを示す。

4. Leap Motion の教育利用の可能性

ドリトルにより Leap Motion から値を得ることができるようになったことで、プログラミングの経験がない児童・生徒が Leap Motion から得た値を用いたプログラムを作成できる可能性がある。ここでは児童・生徒が作成するプログラムからの利用の可能性を検討する。

```

かめた＝タートル！作る。
リープ！接続。
タイマー！作る 600 時間「
「リープ！パー？」！なら「かめた！10 歩く」実行。
「リープ！左回転？」！なら「かめた！15 左回り」実行。
「リープ！右回転？」！なら「かめた！15 右回り」実行。
」実行。

```

図 5 ドリトルプログラム

4.1 小学校での活用案

小学校では、理科や総合的な学習の時間での活用が考えられる。児童にとって自動ドアやエアコンなどの機器が、センサーで値を読み取り、制御されているという実感は薄い。そこで、センサーで制御されている機器が、搭載されたセンサーにより、何らかの動きや状態に反応して動作していることを伝えた上で、その制御はプログラムによっておこなわれていることを伝える。

何らかの状態に反応して、定義した動作を行うという基本的な仕組みの説明をした上で、生徒が Leap Motion に搭載されたジェスチャーを利用し、何らかの動作するプログラムを作成する。

たとえば、Leap Motion に手をかざすと、画面上のタートルが動き、サークルジェスチャーをすることで方向転換することを定義したプログラムを作成することによって、何らかの動きに反応して、定義された動作をするということを体験することができる。

図 5 にプログラム例を示す。このプログラムを実行し、手を広げて Leap Motion にかざすと、画面上のタートルが前に進み、サークルジェスチャーの回転方向によって進む方向を変更することができる。プログラムの 4,5,6 行目で Leap Motion の値を読み取り、ジェスチャーが行われたとき、それぞれの処理をおこなっている。

「…」！なら「…」実行。

という命令は、ある条件に当てはまったら、プログラムを実行するという意味である。**！なら**の前にある「…」の中に条件文を入れて、**実行。**の前にある「…」に、条件文が当てはまったとき実行させたいプログラムを記述する。

また、「リープ！パー？」で手を広げているかどうか、「リープ！左回転？」で左まわりのサークルジェスチャーが起きたかどうか、「リープ！右回転？」で右まわりのサークルジェスチャーが起きたかどうかを Leap Motion から読み取ることができる。ジェスチャーが行われたなら true、そうでなければ false が返される。

たとえば、図 5 の 4 行目のプログラムでは、「リープ！パー？」で手を広げているかを読み取り、もし広げているなら、「かめた！10 歩く」が実行され、画面上のタートルを歩かせることができる。

```

システム！” Arduino” 使う。
a = Arduino ! 作る。
a ! (システム！シリアルポート選択) ひらけごま。
リープ！接続。
タイマー！作る 600 時間「
  led = a ! 13 デジタル出力。
  「リープ！回転？」！なら「led ! 1 書く。a ! 1 待つ。
led ! 0 書く。a ! 1 待つ。」！2 繰り返す。
  」実行。
」実行。
とじごま。

```

図 6 サークルジェスチャーで LED を制御するプログラム

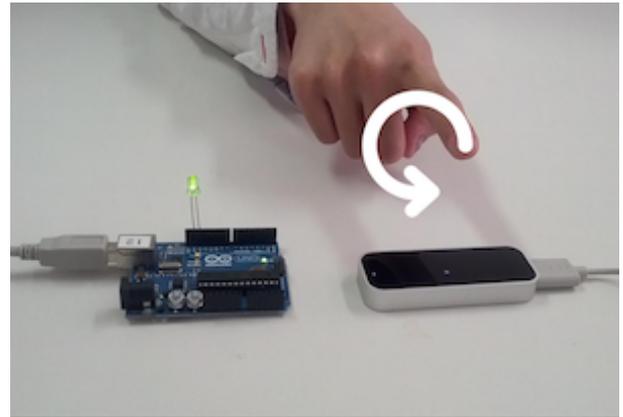


図 7 サークルジェスチャーにより LED が点灯する様子

4.2 中学校での活用案

中学校の技術・家庭（技術分野）ではプログラムによる計測・制御が2012年から必修化されており [6], Leap Motion で計測された手や指の値を利用し、LED やロボットの動きを制御することで計測・制御の教材に活用できる可能性がある。

まず、ドアの前に立つと自動でドアが開く自動ドアや人を感じるとライトが点灯する照明機器など、センサーで制御されている家電などを例にあげ、どういう仕組みになっているのかを考えさせる。それらは、搭載されたセンサーが何らかの動きや状態に反応して動作するが、それはプログラムによって制御されていることを伝える。世の中におけるセンサーの利用例と、何らかの状態に反応して、定義した動作を行うというセンサーの基本的な仕組みの説明をした上で、生徒が Leap Motion に搭載されたジェスチャーを利用し、何らかの動作するプログラムを作成する。

たとえば、Leap Motion と汎用入出力デバイスである Arduino を用いることで、Leap Motion の値を計測し、その値に応じて Arduino に接続した LED を制御することが可能である。また、Arduino に LED 以外の出力デバイスを接続することにより、教材の幅が広がると考える。

図 6 にサークルジェスチャーで LED を制御するプログラム例を示す。7行目の「リープ！回転？」でサークルジェスチャーが行われたかどうかを読み取り、ジェスチャーが行われたときだけ LED を点灯させる処理をおこなっている。

プログラムを実行すると Leap Motion でサークルジェスチャーを認識するたび、Arduino につなげた LED が2回点滅する。実際に実行している様子を図 7 に示す。矢印で示したように円を描くことで、LED が点灯する。

このように、Leap Motion でジェスチャーを入力し、Arduino に接続した LED を点灯させるプログラムを作成することにより、センサーによって動作する照明の仕組み、その計測と制御の方法を学ぶことができると考える。

4.3 高等学校での活用案

高等学校では、ハードウェアの単位における活用が考えられる。高校情報のハードウェアの単位において、学習指導要領では「コンピュータの周辺機器の種類の特性や役割及びその接続に使われる様々なインタフェースの種類、特性や役割などについて理解させる。周辺機器については、データの流が理解できるように相互に関連付けて取り扱う。」とある [7]。

コンピュータにおける入力装置は、キーボードとマウスが主流であるが、最近では、モーションセンサー入力など新しい技術もあることを紹介した上で、Leap Motion を利用したプログラムを作成させる。作成したプログラムを実行した後に、仕組みについて説明をおこなう。

たとえば、Leap Motion がジェスチャーを認識したとき、そのジェスチャーに応じて動作を変化させるプログラムや値を用いて動作を変化させるプログラムを作成することにより、コンピュータの周辺機器の1つである入力装置の仕組みを体験できる。このようなプログラミング体験を通して、入力装置の仕組みや特性、データの流が理解できると考える。

サンプルプログラムを図 8 に示す。このプログラムでは、スワイプジェスチャーによって画面上のタートルの動きを制御している。

プログラムの 6,7 行目にある「リープ！右はらい？」で、右方向へのスワイプジェスチャーが起きたか、「リープ！左はらい？」で左方向へのスワイプジェスチャーが起きたかを読み取っている。

5. おわりに

教育用プログラミング言語ドリトルの処理系を、Leap Motion 社の入力デバイスである Leap Motion の計測値を読み取れるよう拡張をおこなった。ドリトルを使用することで、汎用的に利用されているプログラミング言語に比べて容易に Leap Motion から値を得ることが可能になり、生

```
かめた=タートル!作る ペンなし。
時計=タイマー!作る 100 時間。
時計!「かめた!5 歩く」実行。
リープ!接続。
タイマー!作る 600 時間「
  「リープ!右はらい?」!なら「かめた!30 左回り。」実行。
  「リープ!左はらい?」!なら「かめた!30 右回り。」実行。
」実行。
りんご=タートル!作る "apple.png" 変身する ペンなし。
りんご!(乱数(600)-300)(乱数(300)-150)位置。
りんご!作る(乱数(600)-300)(乱数(300)-150)位置。
りんご!作る(乱数(600)-300)(乱数(300)-150)位置。
かめた:衝突=「|相手|相手!消える」。
```

図 8 スワイプジェスチャーを使ったプログラム

徒が作成するプログラムから利用することが可能となった。

謝辞 本研究は、科学研究費補助金（基盤研究（C）25350214）の補助を受けています。

参考文献

- [1] 教育用プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp>
- [2] 兼宗進, 中谷多哉子, 御手洗理恵, 福井眞吾, 久野靖: 初中
等教育におけるオブジェクト指向プログラミングの実践と
評価. 情報処理学会論文誌プログラミング Vol.44, No.13,
pp.58-71 (2003).
- [3] 紅林秀治, 兼宗進: 制御と計測を取り入れた情報教育の提
案. 情報処理学会研究報告. コンピュータと教育研究会報
告 Vol.2004, No.100, pp.41-48 (2004).
- [4] 佐藤和浩, 紅林秀治, 兼宗進: 小学校におけるプログラミン
グ活用の現状と課題. 情報処理学会研究報告. コンピュ
ータと教育研究会報告 Vol.2005, No.15, pp.57-63 (2005).
- [5] Leap Motion.
<https://www.leapmotion.com/>
- [6] 文部科学省: 中学校学習指導要領 (2008).
- [7] 文部科学省: 高等学校学習指導要領解説情報編 (2010).