

JavaScript 版ドリトルのタブレットでの利用可能性の提案

本多 佑希¹ 長 慎也³ 長島 和平⁴ 大村 基将^{1,2} 島袋 舞子¹ 並木 美太郎⁴ 兼宗 進¹

概要: JavaScript で記述したドリトル言語のプログラミング環境を開発した。ドリトルのプログラムを JavaScript に変換して実行する方式により、Web ブラウザ上での開発と実行が可能である。PC に加え、スマートフォンやタブレットでの動作も可能になった。また、端末に内蔵された各種のセンサから値を取得することも可能である。本稿ではその実装を報告するとともに、タブレットなどに搭載された各種センサをドリトルから利用する学習環境を提案する。

1. はじめに

近年、Web ブラウザ上で動作するオンラインプログラミング環境が増えつつある。さまざまな OS で動作することや、開発環境のインストールが不要といったメリットがある。

今回、教育用プログラミング言語であるドリトル [1] [2] をオンラインプログラミング学習環境である BitArrow 上で動作するように実装した。これまで、ドリトルはローカル環境で動作することが可能であったが、BitArrow 上で動作することにより、オンライン環境で動作することが可能になる。また、スマートフォンやタブレット上で実行させることが可能になり、それらのデバイスに内蔵されたセンサを扱うことも可能になる。

本稿では、JavaScript 版ドリトルの実装を報告するとともに、スマートフォンやタブレット端末に内蔵されたセンサをドリトルで扱えるようにし、それらの教育利用の可能性を検討する。

2. オンラインプログラミング環境

2.1 関連研究

Web ブラウザ上で動かすことができる、プログラミングを学ぶことができるオンライン学習サイトやプログラミング環境は多く存在する (表 1)。

ドットインストール [3] は、プログラミングを動画で学

表 1 Web ブラウザ上で動作する学習教材や環境

名前	e-learning	プログラムの実行
ドットインストール	○	×
Codecademy	○	○
jsdo.it	×	○
paiza.IO	×	○

ぶことができるオンライン学習 (e-learning) サイトである。サイト上でプログラムの実行はできないが、サンプルプログラムを自由に閲覧することができる。

Codecademy [4] は、課題に沿ってプログラミングを学習できるオンライン学習サイトである。課題は一部日本語に対応している。プログラムの記述部が用意されていて、実際にプログラミングを体験しながら学ぶことができる。

jsdo.it [5] は、JavaScript や HTML、CSS を用いた開発を Web ブラウザ上でできるオンラインプログラミング環境である。開発者向けのサイトであり、学習向けのコンテンツは用意されていない。開発したプログラムを他者に公開することができ、他者が開発したプログラムを自由に閲覧できる。

paiza.IO [6] は、JavaScript や Python を含め、24 種類のプログラミング言語を Web ブラウザ上で実行することができるオンラインプログラミング環境である。jsdo.it と同様、開発者向けのサイトであり、学習向けのコンテンツは用意されていない。また、開発したプログラムを他者に公開することができ、他者が開発したプログラムを自由に閲覧できる。

2.2 Bit Arrow

現在公開されているオンラインのプログラミング環境は、手軽にプログラミングを学ぶことが可能な反面、個人での利用を想定しており、授業での利用が想定されてい

¹ 大阪電気通信大学
Osaka Electro-Communication University

² 静岡大学
Shizuoka University

³ 明星大学
Meisei University

⁴ 東京農工大学
Tokyo University of Agriculture and Technology

いものが多かった。

そこで筆者らは、教育での利用を目的とし、オンラインプログラミング環境 BitArrow を開発した [7] [8]。JavaScript、C 言語に対応しており、次のような、特徴がある。

- (1) エラーを日本語で表示する
- (2) 生徒のプログラムがクラウドに保存される
- (3) 教員が生徒のプログラムを表示・実行できる

ドリトルを BitArrow 上で動作させることで、ユーザ管理機能、クラス管理機能といった BitArrow の利点を得ることができる。

3. JavaScript 版ドリトルの開発

3.1 開発方針

3.1.1 概要

ドリトルを主要 web ブラウザで実行可能な Web アプリケーションとして提供する。この時、学習者が作成したドリトルプログラムの実行方法については、サーバにプログラムを送信して実行結果を受け取る方式と、クライアント側 (Web ブラウザ内) にて実行する方法が考えられる。教育現場のネットワーク環境や回線の状況、授業を受ける学習者の人数、学習者がプログラム入力してから実行までのレスポンスなどを考慮すると、Web ブラウザ内でプログラムが実行されることが望ましい。よって、本研究では、Web ブラウザ内でのプログラム実行の方法として、JavaScript へのトランスパイル方式を採用した。また、授業利用での利便性を考慮し、様々な教育向けの管理機能を有する BitArrow の機能として上記を実現した。なお、言語仕様については Java 版ドリトルの設計 [9] [10] に準拠することを原則とする。オブジェクトと定義されたメソッド例を図 2 に示す。

このような Java 版に対する互換性をできるだけ確保しつつ、スマートフォンやタブレットでの開発や実行を可能にする、端末に内蔵された各種センサーの利用を可能にする、といった JavaScript の特徴を活かした拡張を行い、Bit Arrow のプラットフォームで動作させることでクラウドでのプログラム管理と授業での利用の利便性の実現を目指す。

3.1.2 オブジェクト

ドリトルの言語的な特徴として、プロトタイプ方式のオブジェクト指向言語であることが挙げられる。ドリトルでは、オブジェクトに対し「作る」のメソッドコールを行うことで、そのオブジェクトを親に持つ、新たなオブジェクトを生成することができる。この時、元になったオブジェクトを親オブジェクト、新たに作られたオブジェクトを子オブジェクトと呼ぶ。子オブジェクトは、親のメソッドやプロパティを引き継ぎ、自由に参照することができる。JavaScript においても、オブジェクトの親子関係を構築することができる。ただし、ドリトルと JavaScript ではメソッド実行時に変数を参照する場合において、仕様の差異

表 2 ドリトルのオブジェクト例 (一部抜粋)

オブジェクト	命令	意味
タートル	作る	新たなタートルオブジェクトを生成する
	歩く	引数で指定した距離だけ歩く
	戻る	引数で指定した距離だけ戻る
	右回り	引数で指定した角度だけ右に回転する
	左回り	引数で指定した角度だけ左に回転する
	移動する	引数で指定した相対位置に移動する
	位置	引数で指定した絶対位置に移動する
	向き	引数で指定した数値を角度に設定する
	ペンなし	歩いた時に足跡が残らないようにする
	ペンあり	歩いた時に足跡が残るようにする
	中心に戻る	画面の中心に移動する
	閉じる	足跡を書き始めた位置まで移動する
	ボタン	作る
読む		ボタンの文字列を返す
書く		ボタンに文字列を設定する
タイマー	作る	新たなタイマーオブジェクトを生成する
	間隔	実行間隔を引数で指定した秒数に設定する
	回数	実行回数を引数で指定した回数に設定する
	時間	実行時間を引数で指定した秒数に設定する
	実行	タイマーを実行する
	中断	タイマーの実行を停止する

が存在する。この点においては言語の記述方法の置換では実現ができないため、変数の参照先をドリトル仕様に合わせるための例外処理を実装することで対応することにした。

3.1.3 タイマーオブジェクト

タイマーオブジェクトは、定間隔でプログラムを実行するための仕組みを提供するオブジェクトであり、メインのプログラム (メインスレッド) とは非同期で動作する。授業実践においても、アニメーションやゲーム開発などで多用されるオブジェクトであり、ドリトル言語における主要オブジェクトの一つとなっている。Java 版ドリトルにおいては、タイマーオブジェクト毎にスレッドを生成することで非同期処理を実現した。同期処理の方法を見直し、「次に実行」、「最後に実行」の命令を追加した。これにより、タイマー動作の逐次処理を、一つのタイマーに整理することができると考えた。

3.2 実装

3.2.1 トランスパイル

トランスパイルとは、あるプログラム言語を抽象化レベルが同程度の別の言語に変換することである [11]。JavaScript は主要ブラウザが標準的に実装しているスクリプト言語であり、プロトタイプ方式であることや、記述法などにおいてドリトルと類似点が多い。そこで、ドリトルを JavaScript に変換するトランスパイラを構築した。トランスパイルの実行モデルを図 1 に示す。学習者の書いたドリトルプログラムは、ブラウザ内の用意したトランスパイラにより、JavaScript のコードに変換する。そして、変換した

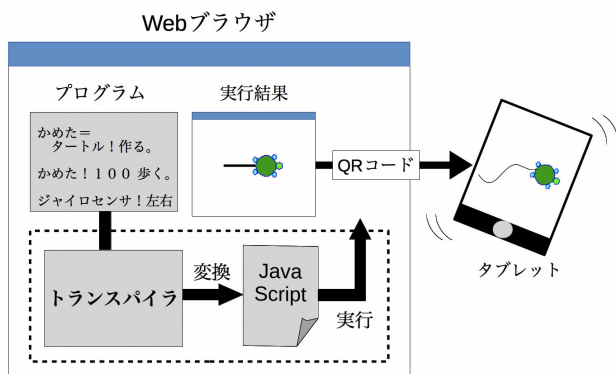


図1 JavaScript版ドリトルの実行モデル

JavaScriptのコードを改めて、ブラウザ内で実行する。トランスパイラ自体もJavaScriptで書くことで、Webブラウザへの特別な機能追加なしに、ドリトル言語のプログラムを実行することができる。

3.2.2 親子関係

ドリトルの「作る」メソッドは、「Object.create」関数を用いてメソッドが所属するオブジェクトをプロトタイプに持つ、新たなオブジェクトを生成することで実現した。これにより、JavaScriptのオブジェクトは、親オブジェクトのプロパティやメソッドを自由に参照することができる。ただし、メソッド実行時に変数を参照した場合、ドリトルとJavaScriptでは参照する先が異なる。この問題は、メソッドの実行中に変数を参照する時、ドリトルとJavaScriptで該当変数を探す際のスコープの範囲が異なることが原因である。変数の検索処理では、ドリトルとJavaScriptは共に、最初はメソッド内の引数の参照を試み、なければローカル変数の参照を試みる。しかしこの後の振る舞いが異なる。ドリトルでは、メソッドが所属するオブジェクトのプロパティを参照し、そこにもなければ、親であるオブジェクトのプロパティを参照する。一方、JavaScriptでは、そのメソッドが所属するオブジェクトのプロパティを飛ばして、「グローバル変数」を参照する。そこで、引数の参照、ローカル変数の参照に失敗した場合には、変数参照箇所ではthisを参照するように指定する処理をトランスパイルの段階で組み込むことで、従来のドリトルと同様の動作を実現した。

3.2.3 オブジェクト

ドリトルのオブジェクトの実装については、プロトタイプ拡張とオブジェクトのラップの手法を併用した。プロトタイプ拡張は、ライブラリ同士での名前の競合が発生する危険性がある。このため、JavaScript固有の機能を利用したり、プロトタイプ拡張の利用の方が実装の容易さが明らかに勝る、あるいは、直感的であると判断した部分については、プロトタイプ拡張を利用した。したがって、文字列・数値・ブロック・真偽値・配列の言語のコアとなるオブジェクトは、JavaScriptのオブジェクト（window.String、

window.Numberなど）をプロトタイプ拡張して実装を行った一方、GUI関連のオブジェクトなど、ライブラリを利用するオブジェクトなどについては、JavaScriptの機能をラップする形で実装した。タートルやボタン、ラベルなどのオブジェクトがこれに該当する。タートルオブジェクトでは、jQueryのSVGを操作する機能をラップして実装した。ボタンやラベルは、jQueryのHTMLタグを操作する機能や、CSSを操作する機能をラップして実装した。これら実装方法の併用によって、実装したオブジェクトは、従来のドリトルのオブジェクトとの互換性を高く維持できた。なお、Java版ドリトルの機能であるネットワーク・音楽演奏・外部機器の制御などは現在検討を進めている。

3.2.4 タイマーオブジェクト

Java版ドリトルのタイマーオブジェクトでは、タイマーのスレッドはメインスレッドと並列に実行され、メインスレッドは必要に応じてタイマーの実行が終了するまで停止する必要があった。複数のスレッドを待ち合わせて統合する機能は便利だが、必ずしも初心者にはわかりやすい仕様ではなかったため、JavaScript版ではこの機能を実装せず、タイマーの終了後に同じタイマーで次の処理を行う「次に実行」を用意した。

3.3 得られた利点

トランスパイル方式によりドリトルがWebブラウザ上で動作するようになったことで次の利点が生まれた。

- (1) URLを伝えることで作品を公開できる
- (2) QRコードを読むことでスマートフォンやタブレットで実行できる
- (3) スマートフォンやタブレットでもBitArrowとドリトルが動作する
- (4) 実行端末に内蔵されたセンサ類をプログラムから扱える

(1)(2)は、URLを公開することで作成したアニメーションやゲームを他の人に使ってもらえるようになった。また、QRコードを読むことで、スマートフォンやタブレット端末からも作品を見ることができるようになった。作成したゲームをスマートフォンで実行した画面を図2に示す。(3)は、現代の携帯端末のブラウザはコンピュータで使用するブラウザと遜色なく使用できるため、プログラムを実行するだけでなく、文字入力を工夫できれば、BitArrowでドリトルのプログラム開発が可能になる。(4)は、実行する端末に内蔵された各種センサ類をプログラムから扱えるようになった。加速度センサ、ジャイロセンサ、地磁気センサ、GPSなどのセンサらしいセンサのほかに、画面のタッチ、マイク音声、カメラ画像を使用することができると考えられる。

そこで、(4)の利点を活かすために、実行端末に内蔵された各種センサ類をドリトルのプログラムからわかりやす

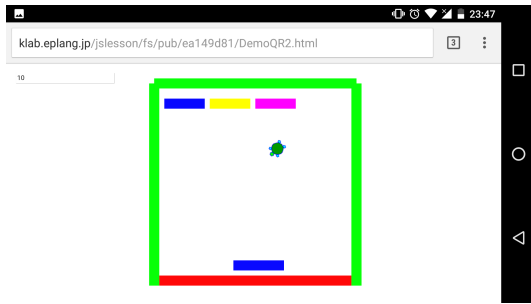


図 2 作成したゲームのスマートフォンでの実行

```

1 かめた=タートル!作る。
2 タイマー!作る 200 時間「
3   x =加速度センサ!左右の加速度?。
4   かめた!(x) 0 移動する。
5 」実行。
    
```

図 3 加速度センサのプログラム例

```

1 かめた=タートル!作る。
2 タイマー!作る 200 時間「
3   x =ジャイロセンサ!左右の傾き?。
4   y =ジャイロセンサ!前後の傾き?。
5   かめた!(x) (y) 移動する。
6 」実行。
    
```

図 4 ジャイロセンサのプログラム例

く使えるようにし、オブジェクトとして用意した。

4. 端末内蔵センサへの対応

実行端末に内蔵された各種センサ類をドリトルから扱えるようにした。今回は、加速度センサ、ジャイロセンサ、磁気センサ、タッチセンサに対応する。実装はタッチセンサをイベントドリブン方式、その他のセンサはポーリング方式でおこなった。表 3 に、現在ドリトルで扱えるセンサオブジェクトと各命令を示す。

加速度センサは、端末の x, y, z 軸の加速度を取得することができる。サンプルプログラムを図 3 に示す。これは、端末の x 軸の加速度に応じて、画面上のタートルが左右に移動するプログラムである。3 行目で加速度センサの x 軸の値を取得している。

ジャイロセンサは、x 軸、y 軸、z 軸の回転の角度を取得することができる。サンプルプログラムを図 4 に示す。このプログラムは、端末の傾きに合わせて、画面上のタートルが移動するプログラムである。3 行目でジャイロセンサの x 軸回転の角度を取得している。また、4 行目でジャイロセンサの y 軸回転の角度を取得している。

磁気センサは、端末が向いている方向を取得することができる。サンプルプログラムを図 5 に示す。このプログラムは、端末の向きに合わせて、方位を画面上に示すプログラムである。4 行目で磁気センサから、端末の向きを取得

```

1 かめた=タートル!作る。
2 方位表示=フィールド!作る。
3 タイマー!作る 200 時間「
4   x =磁気センサ!向き?。
5   方位表示!(x) 書く。
6 」実行。
    
```

図 5 磁気センサのプログラム例

```

1 タッチ表示=フィールド!作る。
2 タイマー!作る 200 時間「
3   x =タッチセンサ!タッチしている?。
4   タッチ表示!(x) 書く。
5 」実行。
    
```

図 6 タッチセンサのプログラム例

表 3 センサ用オブジェクトの命令一覧

オブジェクト名	命令名	説明
ジャイロセンサ	左右の傾き?/ロール?	x 軸回転の角度を得る
	前後の傾き?/ピッチ?	y 軸回転の角度を得る
	水平の傾き?/ヨー?	z 軸回転の角度を得る
	動作設定	イベントの動作を設定
磁気センサ	向き?	端末の向きを得る
	動作設定	イベントの動作を設定
タッチセンサ	タッチした?	一度でも触れたかを得る
	タッチしてる?	触れているかを得る
	横の位置?	触れた位置の x 座標を得る
	縦の位置?	触れた位置の y 座標を得る
	動作設定	イベントの動作を設定
加速度センサ	左右の加速度?	x 軸方向の加速度を得る
	前後の加速度?	y 軸方向の加速度を得る
	上下の加速度?	z 軸方向の加速度を得る
	動作設定	イベントの動作を設定

している。

タッチセンサは、端末の画面に触れたか、その位置に触れたかなどを取得することができる。サンプルプログラムを図 6 に示す。このプログラムは、端末の画面に触れている場合は true、触れていない場合は false を出力する。3 行目で値を取得している。

スマートフォンやタブレットの機種により、内蔵されているセンサが異なることや、加速度センサなどで取得できる値が異なる場合がある。使用する Web ブラウザによっても同様の問題が起こることがある。また、一部の PC では、加速度センサのセンサが搭載されている場合があり、スマートフォンやタブレット端末と同様に内蔵のセンサを使用することができる。

このように端末に搭載されたセンサ類をドリトルから扱えるようになったことで、プログラムの体験を通して、コンピュータの原理やアプリケーションのしくみを学ぶことが期待できる。

```

1 センサ=タッチセンサ！作る。
2 tap=いいえ。
3 タイマー！作る 180 時間 0.05 間隔「
4 X 座標=センサ！横の位置？。
5 Y 座標=センサ！縦の位置？。
6 「センサ！タッチしてる？」！なら「
7 「tap==はい」！なら「かめた！ペンあり」実行。
8 かめた！（X 座標）（Y 座標）位置。
9 tap=はい。
10 」そうでなければ「
11 「tap==はい」！なら「かめた！図形を作る」実行。
12 かめた！ペンなし。
13 tap=いいえ。
14 」実行。
15 」実行。
    
```

図 7 プログラムの一部

5. タブレット端末の学習利用の可能性

端末に搭載されたセンサ類をドリトルから扱えるようになったことで、プログラムの体験を通して、コンピュータの原理やアプリケーションのしくみを学ぶことが期待できる。

例えば、端末のタッチセンサを利用し、お絵かきアプリを作成する。色を追加したり、図形を描くボタンを追加するなど、作品に個性を出すことが可能である。お絵かきアプリの作成を通して、スマートフォンのアプリケーションの仕組みやタッチパネルに文字を書く仕組みを体験することができる。プログラムの一部を図 7 に示す。このプログラムはタッチセンサの値を読み、タッチの有無、タッチされた画面の座標に応じて線を描いている部分である。実行画面は図 8 に示す。

また、同じプログラムで使用するセンサを変えて実行することで、各種センサの特性を知ることができる。例えば、図形をセンサから取得した値によって操作することで、センサの違いによる動きの変化や特性、特徴を体験し、学ぶことができる。例として、加速度センサの値で、図形の位置を変更するプログラムを図 9 に示す。

他にも、磁気センサを使用した図 5 のプログラムを改良し、簡易的な方位磁石アプリを作成する。作成を通じて、小学校の社会科においては、方位記号や方位の位置や読み方などの学習に応用することができる。また、これを総合的な学習の時間で地域を散策する時に地図と併用し方角を確認するなどにも使用できる。児童自らが作ったアプリを使用することで、児童の学習意欲をひきだすことができる。と考える。

6. おわりに

オンラインプログラミング環境 BitArrow 上で動作する JavaScript 版ドリトルを開発した。言語のコアとなるオブ

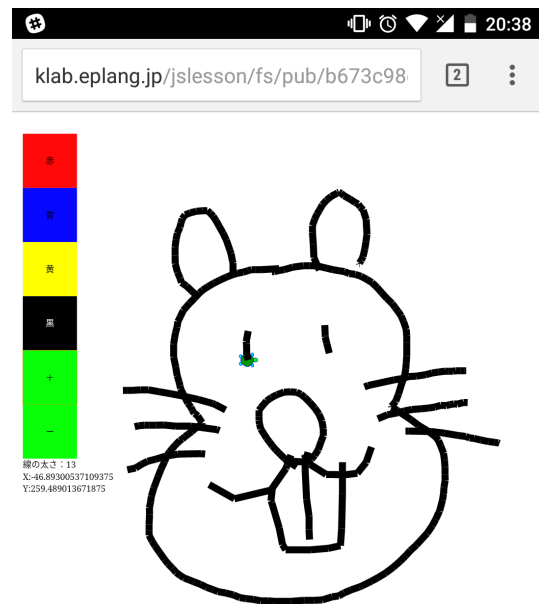


図 8 お絵かきアプリの実行画面

```

1 かめた=タートル！作る。
2 かめた！（青）線の色 20 線の太さ。
3 星=「かめた！100 歩く 144 左回り」！4 繰り返す（青）
  図形を作る。
4 かめた！消える。
5 加速度=加速度センサ！作る。
6 タイマー！作る 60 時間「
7 横=加速度！左右の加速度？。
8 星！（横）0 移動する。
9 」実行。
    
```

図 9 プログラム例

ジェクト、グラフィックのオブジェクトは実装を行った。ネットワークや音楽演奏、制御の機能は、検討を進めている。

JavaScript 版ドリトルでは、スマートフォンやタブレットでの実行に加え、それらに内蔵されたセンサを扱うことができるようになった。その結果、プログラムの体験を通して、コンピュータやアプリケーションの仕組みを学ぶといった教育利用が可能になった。

参考文献

- [1] プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp/>
- [2] 兼宗進, 久野靖: ドリトルで学ぶプログラミング [第2版]
(2011).
- [3] ドットインストール .
<http://dotinstall.com/>
- [4] Codecademy .
<https://www.codecademy.com/learn>
- [5] jsdo.it .
<http://jsdo.it/>
- [6] paiza.IO .
<https://paiza.io/>
- [7] オンラインプログラミング環境ビットアロー (BitArrow).
<http://bitarrow.eplang.jp/>
- [8] 長島和平, 本多佑希, 長愼也, 間辺広樹, 兼宗進, 並木美太郎: オンラインで複数言語を扱うことができるプログラミング授業支援環境. 情報教育シンポジウム (SSS2016), pp.137-140, 2016.
- [9] 兼宗進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野靖: 学校教育用オブジェクト指向言語「ドリトル」の設計と実装. 情報処理学会論文誌, Vol.42, No.SIG11, pp.78-90, 2001.
- [10] 兼宗進, 久野靖: プロトタイプ階層を持つ教育用オブジェクト指向言語「ドリトル」. コンピュータソフトウェア, Vol.28, No.1, pp.43-48, 2011.
- [11] 本多佑希, 大村基将, 長愼也, 久野靖, 並木美太郎, 兼宗進: Dolittle のオンラインプログラミング環境の開発. 情報処理学会 コンピュータと教育研究会, Vol.CE(134), No.25, pp.1-5, 2016.