

タートルグラフィックスを用いた イベント駆動型プログラミングの導入教材と例題の提案

清野 貴博^{1,a)}

概要: タートルグラフィックスはプログラミング初学者のための古典的な導入教材であり、現在でも手続き型プログラミングの基礎概念を教育するために用いられている。今日では、それ以外のプログラミングパラダイムも重要であり、とりわけ、JavaScript が用いられるウェブアプリケーションでは、イベント駆動型プログラミングが要求される。本研究では、タートルグラフィックスを JavaScript 版ライブラリとして開発し、さらにそれをイベントを発火できるように拡張した。この教材を文系大学での授業で使用し、複数のタートルをイベントによって連携させる課題を受講生に課したところ、高い教育効果が見られたので報告する。

An Educational Material and an Exercise for Introduction to Event-driven Programming with Turtle Graphics

Abstract: This paper proposes an educational material to learn fundamentals of event-driven programming with turtle graphics. Turtle graphics are the classic introductory teaching materials for a programming beginner. Turtle graphics use to teach concepts of procedural programming, but other programming paradigms are also important. Especially, many web applications are written in JavaScript and they are designed with the paradigm of event-driven programming. We have developed a library of turtle graphics in JavaScript and extended it to trigger events. We use this library in a programming class in a liberal arts college. In that class, we gave the assignment which two or more turtles cooperate with triggering and capturing events and many students submitted their great works.

1. はじめに

長い間、プログラミング教育の重要性が叫ばれているが、多くの学習者はその過程でプログラミングに対する興味を失ってしまう。興味を惹きつける教材が必要であり、ゲームを題材としたプログラミング教育 [11][14] など提案されているが、筆者の経験ではゲームに偏重しすぎると、逆に興味を失ってしまう学生も多いと感じている。とりわけ、昨今の大学生は仲間や社会の役に立てるかどうかに敏感であり、一人で黙々とゲームを作ったり遊んだりするというイメージを持たれると、逆効果のようである。筆者は、できる限り早期に彼らの欲求を満たす教育プログラムが必要であると痛感しており、HTML と JavaScript を使った

ウェブアプリケーションに教育資源を集中するのは、一つの方策であると考えている。

JavaScript は Scheme を Java 風の構文にし、Self からプロトタイプベースのオブジェクト指向を採り入れた言語である。Google 社の GMail などに代表されるウェブアプリケーションの成功により、JavaScript の価値が見直されて久しい。言語の核の部分は、ECMAScript[4] として標準化され、ウェブブラウザだけでなく、Microsoft Windows においては JScript, Adobe 社においては Action Script として使用されている。Node.js[9] の登場により、クライアントサイドだけでなく、サーバサイドのソフトウェアも JavaScript で記述可能となり、JavaScript だけでウェブアプリケーションを実装することもできる。

プログラミングの初学者に対し、手続き型プログラミングのパラダイムを教育することは広く行われてきたが、JavaScript を用いたウェブアプリケーションの開発を行う

¹ 共愛学園前橋国際大学
Maebashi Kyoai Gakuen College 1154-4 Koyahara, Maebashi, Gunma 379-2192, Japan

^{a)} seino@c.kyoai.ac.jp

ことを想定すると、それでは十分ではない。JavaScript ではイベント駆動型プログラミングを採用しており、ウェブブラウザを通したユーザの振る舞いはイベントとして扱われる。Node.jsにおいても、クライアントからサーバへの通信はイベントとして扱われており、サーバの動きはイベントに対する応答として記述される。このようにJavaScriptのプログラミングにおいてイベント処理が中核となっており、逆に言えば、学習者にとってイベント駆動型プログラミングのパラダイムを理解することがハードルであると言える。

本稿では、手続き型プログラミングにおける変数や制御構造を理解した学習者を想定し、タートルグラフィックスを用いてイベント駆動型プログラミングを導入する教材を提案する。タートルグラフィックス [1] は、タートルを用いて絵を描くための教材であるが、本稿では、学習者に反復学習を抵抗なく行わせる課題として、複数のタートルを用いた画面上でのパフォーマンスを提案する。このパフォーマンスは、ダンスと呼んでいる。この課題を、いわゆる“文系”の大学において学生に課したところ、多くの力作が提出され、高い教育効果があった。

本稿は、次のように構成している。2 節において、JavaScript におけるイベント駆動プログラミングについて述べる。3 節では、提案する課題を導入した共愛学園前橋国際大学のプログラミングに関するカリキュラムについて述べる。4 節では、実装したタートルグラフィックスのライブラリについて述べ、5 節では授業における適用状況と結果について述べる。6 節で教育効果について考察し、7 節で関連研究との比較を行い、8 節で結論を述べる。

2. JavaScript

JavaScript はウェブブラウザで動く唯一の言語である。HTML5 によって API が整備されたことにより、HTML と JavaScript だけでも、これまでにない実用的なアプリケーションが開発できるようになっている。言語の核は ECMAScript として標準化されており、ECMAScript によって制御する対象となる資源をホスト環境と呼ぶ。ホスト環境がウェブブラウザであるものや、Node.js を特に JavaScript と呼ぶことが多い。それ以外のホスト環境、例えば Microsoft Windows をホスト環境とする場合は JScript と異なる名前では呼ばれる。

jQuery Mobile などを利用することによって、スマートフォン用のアプリを簡単に作成することができる。また、それをラッピングすることにより、iOS 端末や Android 端末においてネイティブアプリとして動作させることも可能である。

2.1 JavaScript のオブジェクト指向プログラミング

JavaScript におけるオブジェクト指向プログラミング

は、現在でも流動的な部分があり、Crockford らによって、言語処理系のサポートに依らない独自の考え方 [3] が提案されており、ECMAScript 6 においても変更の可能性がある [6]。筆者は、オブジェクト指向プログラミングを初めて学ぶ学生を対象とした教育という点を考え、あまり凝った方法は避け、ECMAScript 5.0 において言語処理系のサポートがある最も標準的と考えられる方法を採用することにした。

JavaScript のオブジェクトは、キーと値のペアによる連想配列である。オブジェクトはプロパティやメソッドを持つが、それらは連想配列の中では明示的には区別されず、言語処理系としては、単に連想配列に格納されたキーによって識別されるプロパティでしかない。JavaScript では関数は第一級オブジェクトであるので、関数は値として扱うことができる。プロパティの中で、値として関数リテラルが格納されているものがメソッドである。

空の連想配列を変数 `o` に代入するには、次のようなプログラムを記述する。

```
var o = {};
```

変数 `o` に代入されたオブジェクト（インスタンス）に対して、プロパティ `p` やメソッド `m` を持たせるには、次のようなプログラムを記述する。

```
o.p = 123;  
o.m = function () {  
    /* something */  
};
```

‘.’ 演算子でキーを指定できる。`p` には数値 123 が、`m` には関数リテラルが代入される。`o` のメソッド `m` を呼び出すには、次のようなプログラムを記述する。

```
o.m();
```

通常関数呼び出しと同様に、‘(’ と ‘)’ の中には引数を書くことができる。

オブジェクトの初期化を行い、インスタンスを返す関数をコンストラクタと呼ぶ。JavaScript では、通常関数とコンストラクタの区別はなく、`new` 演算子を前置して呼び出されたとき、関数はコンストラクタとして動作する。今、関数 `Foo` にオブジェクトの初期化に関する処理が書かれているとき、次のように実行すると、`Foo` はコンストラクタとして動作する。

```
var o = new Foo();
```

これが実行されると、`Foo` が返した新しいインスタンスが変数 `o` に代入される。

JavaScript では、前述のように空の連想配列からオブジェクトを作り上げてもよいし、コンストラクタを使って初期化したインスタンスを生成してもよく、両者のオブジェクトを利用するプログラムから見たときに、プロパティの参照や更新、メソッド呼び出しにおいて、差異はない。

習慣として、コンストラクタである関数の名前は、英大文字から始める。

2.2 JavaScript のイベント駆動型プログラミング

JavaScript による開発が複雑化したことや、歴史的な事情によるウェブブラウザ間の仕様の差異も未だに残ることから、現在では JavaScript をそのまま用いるのではなく、jQuery[7] のようなブラウザの差異を吸収するライブラリを用いられることが多い。以下では、jQuery を使用したイベント駆動型プログラミングについて述べる。

JavaScript は、シンプルなシングルスレッドの実行モデルを採用している。JavaScript のホスト環境や JavaScript で書かれたプログラムで発生したイベントは、イベントキューに蓄えられる。各イベントはイベントオブジェクトで表現され、発生したイベントの種類が `type` プロパティに文字列として保持される。例えば、ユーザがクリックした場合、`type` プロパティには文字列 `"click"` が代入されている。イベントに対する処理は、関数として記述し、それをイベントハンドラとして登録する。

イベントディスパッチャは、イベントキューから処理すべきイベントを順に一つ取りだし、登録されているイベントハンドラを呼び出す。呼び出したイベントハンドラ（関数）が `return` 文に到達するなどして実行を終えるまで、イベントキューに残っているイベントの実行はブロックされる。画面の再描画についても、イベントハンドラの実行中は行われない。

jQuery はグローバル変数 `$` を通じて利用する。`$` は関数であり、その引数に HTML の要素を特定するための式（CSS セレクタ）を与えると、それによって抽出された HTML 要素を表すオブジェクトの配列（以下では、要素集合オブジェクトと書く）が返る。本稿では、HTML の要素を特定せず、HTML 文書全体を表す `document` のみを使用する。

イベントハンドラの設定は、要素集合オブジェクトのメソッド `bind` を使用する。第 1 引数はイベントの種類を表す文字列であり、第 2 引数はイベントハンドラを表す関数リテラルである。例えば、クリックされたときに関数 `foo` をイベントハンドラとして呼び出す場合には、次のように記述する。

```
$(document).bind("click", foo);
```

同様に、イベントの発火は、要素集合オブジェクトのメソッド `trigger` を使用する。第 1 引数はイベントの種類を表す文字列であり、第 2 引数以下は本稿では使用しないので割愛する。例えば、プログラムからイベント `done` を発火するには、次のように記述する。

```
$(document).trigger("done");
```

イベントは、ホスト環境であらかじめ定義されているもののほか、jQuery を使用することにより、任意の名前のイ

ベント（カスタムイベント）を発火させることができる。

時間によるイベントは、JavaScript 組み込みの関数 `setTimeout` により発火させることができる。第 1 引数にはイベントハンドラの関数を、第 2 引数にはイベントの発火時刻を、現在時刻からの相対時間で与える。時間の単位は 1000 分の 1 秒である。

```
setTimeout(foo, 1000);
```

このように記述すると、プログラムが実行された時点から 1 秒後に、関数 `foo` が実行される。

jQuery では、頻繁に使用されるイベントのイベントハンドラの設定は、専用のメソッドが用意されている。例えば、HTML の読み込みが終わったことを通知するイベントに対しては、`ready` メソッドを使ってイベントハンドラを設定できる。

3. カリキュラム

共愛学園前橋国際大学（以下、本学）は、1888 年に開学された共愛学園女子短期大学を母体とし、1999 年に開学した単科大学であり、約 1000 人の学生が学ぶ小規模大学である。いわゆる“文系”の学生の集まりではあるが、情報・経営を専攻する学生に対しては、情報通信技術についても手厚い教育をしている。JavaScript は、手続き型、オブジェクト指向、関数型、およびイベント駆動型プログラミングのパラダイムを採用しており、適宜組み合わせることができる。文系学部では、プログラミングの授業に十分な時間を割けないため、特定のパラダイムに特化した複数の言語を用いてプログラミングを学習すると、結局どの言語も、そしてどのパラダイムも習得できない結果に終わる危険性がある。JavaScript では、一つの言語で主要なパラダイムが学習でき、さらに学習成果をウェブアプリケーションとしてまとめ上げることができるという利点があると筆者は考えている。

本学では、1 年次前期にコンピュータリテラシーの授業を設置し、1 年次後期よりプログラミング教育を開始する。セメスター毎に 1 つずつのプログラミングの授業を履修し、3 年次後期までに独力でウェブアプリケーションを実装できるようになることを目標としている。具体的には、次のような授業を開講している。

アルゴリズム（1 年次後期）

プログラミングの入門科目である。テキストエディタやウェブブラウザに内蔵されているプログラミングのためのツールなど、処理系の使い方から解説し、手続き型プログラミングに基づく制御構造（連接、分岐、および反復）、関数宣言と呼び出し、探索や整列など初歩的なアルゴリズムについて学ぶ。この授業では、プログラムを読めるようになることを目標としている。

情報演習・プログラミング（2 年次前期）

HTML と JavaScript を用いて、GUI プログラミングを

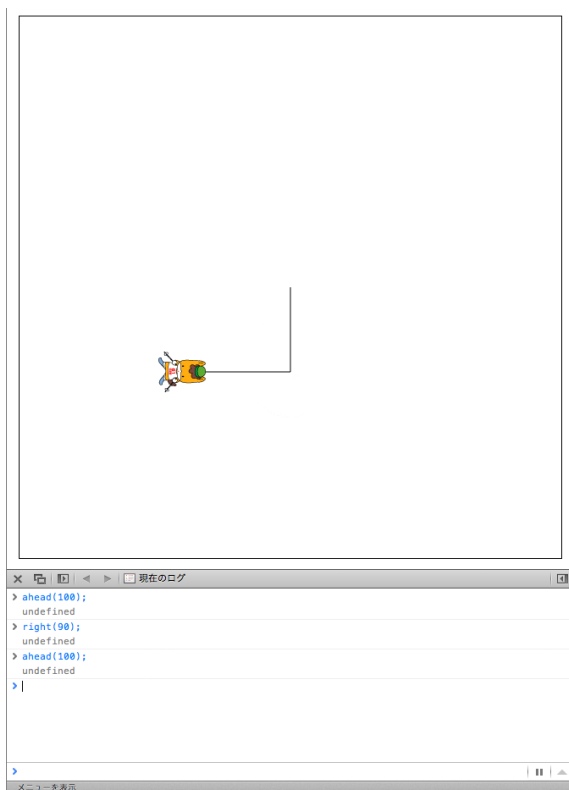


図 1 開発したタートルグラフィックスライブラリ

Fig. 1 screen shot of using our turtle graphics library

学ぶ。本学では 2010 年から iPod touch を全学生に配布していることもあり、スマートフォンに準拠した性能を持つ端末が全学生共通のプラットフォームになっている。この授業では、学生が本学のプラットフォームを活用できるように、スマートフォン向けのクライアント側アプリケーションの開発を想定して教育している。開発を容易にするため、かつ将来のプラットフォーム変更に対応するため、jQuery Mobile[8] を用いている。

オブジェクト指向プログラミング (2 年次後期)

オブジェクト指向プログラミングを学び、より大きな規模のソフトウェアの開発ができるスキルの獲得を目指す授業である。前段の授業でもオブジェクトは利用するが、この授業では単にオブジェクトを利用するだけではなく、自らオブジェクトを記述することを目標としている。前述のように、JavaScript のオブジェクト指向プログラミングにはまだ混乱が見られるため、2.1 節で示した考え方に基いた授業を展開している。

C 言語 (3 年次前期)

JavaScript とは対極に位置する C 言語を通して、コンピュータの仕組みを学ぶ授業である。

システム開発 II (3 年次後期)

Node.js を用いて、JavaScript によるサーバサイドプログラミングを学ぶ。アプリケーションフレームワークとして、Express[5] を使用している。旧来型のフォームとクエリによるアプリケーションの他に、WebSocket[10] による

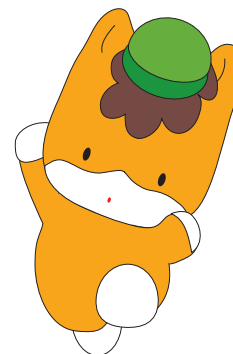


図 2 群馬県のマスコット「ぐんまちゃん」

Fig. 2 Gunma-chan, the mascot of Gunma Prefecture

リアルタイム通信を用いたアプリケーションについても学ぶ。いずれも、JavaScript のイベントモデルによって作られている。この授業で、クライアントからサーバサイドまで、一通りのウェブアプリケーションの開発が可能になる。

4. タートルグラフィックスライブラリ

タートルグラフィックスの機能は、JavaScript の言語としての特性、ユーザの利便性、および教育目的を考慮し、次のように設計した。

開発したタートルグラフィックスライブラリは、HTML 5.0 に対応するウェブブラウザ上で動作する。実行画面を図 1 に示す。プログラミングの習得を目的とするため、ライブラリ自身はユーザインターフェースを持たない。学習者は、初期においてはウェブブラウザのコンソール機能を利用し、ライブラリに宣言されている関数を呼び出して対話的に実行することを想定している。図の上部がタートルで描画するキャンバスであり、下の部分がコンソール*1である。タートルには亀ではなく、本学の立地から学生に親しみの深い群馬県のマスコットキャラクターであるぐんまちゃん (図 2) [12] を使用した。

タートルを操作する関数は、次の 6 つである。

- `ahead(n)` タートルが n ピクセル前進する。
- `back(n)` タートルが n ピクセル後退する。
- `right(n)` タートルが n 度右を向く。
- `left(n)` タートルが n 度左を向く。
- `penup()` 前進または後退時の描画を停止する。
- `pendown()` 前進または後退時の描画を再開する。

いずれも、 n は整数である。 n に負数が与えられた場合、逆の動作 (例えば、`ahead` ならば後退、`right` ならば左を向く) となる。

タートルは初期状態では、タートルは画面中央に位置し、前進または後退時に描画を行う状態にある。タートルは描画コマンドを格納するキューを持ち、これらの関数はキューに描画コマンドを追加するだけで、すぐに制御を返

*1 Safari はエラーコンソール、Google Chrome はコンソールなど、ブラウザによって呼び方に差異がある。

```
var i;  
for(i = 0; i < 3; i = i + 1) {  
    ahead(100);  
    right(180 - 60);  
}
```

図 3 正三角形を描くプログラム

Fig. 3 A program to draw a triangle.

す。実際の描画は、一定時間ごとにキューから取り出したコマンドを実行することで行われる。タートルの前後方向の移動速度については、1000 分の 15 秒ごとに 2 ピクセル、左右の回転については、1000 分の 15 秒ごとに 2 度である。例えば、`ahead(100);` を実行した場合は、100 ピクセル分の直線を描画するのに 1000 分の 750 秒を要する。

タートルグラフィックスの教材は、`turtle.html` と `turtle.js` の 2 つのファイルから成る。`turtle.html` は HTML や CSS で書かれた画面、JavaScript で書かれたライブラリ、および Base64 エンコードされた画像データを含む。極力、不要なファイルやフォルダを作らないように配慮した。学習者は、`turtle.html` をブラウザで開き、学習を進める。`turtle.html` では、外部の JavaScript コードとして同じフォルダ上にある `turtle.js` を読み込む。配布した時点の `turtle.js` は空のファイルであり、受講生はこのファイルを編集し、上記の関数や、通常の JavaScript の制御構文などを用いてプログラムを記述することができる。アルゴリズムの授業での使用を想定したため、HTML は記述させず、純粋に JavaScript だけを記述するようにしている。また、JavaScript で通常問題になる、HTML の読み込み終了タイミングに関する制御については、手続き型プログラミングの範疇を超えるため、ライブラリで吸収して学習者には見せていない。

辺の長さが 100 ピクセルの正三角形を描くプログラム例を図 3 に示す。

4.1 ライブラリの拡張

開発したタートルグラフィックスライブラリを、次のようにオブジェクト指向プログラミングへ対応させた。

タートルは `Gunmachan` オブジェクトとして記述した。インスタンスを作成するコンストラクタ `Gunmachan()` を宣言した。`Gunmachan` オブジェクトには、前述の 6 つの関数をインスタンスメソッドとして宣言した。各インスタンスは固有のコマンドキューを持ち、一定時間ごとにキューからコマンドを取り出して、インスタンス毎に割り当てられた固有の HTML キャンバスに対して描画を行う。タートルの初期状態の位置は、画面中央付近で固定である。

さらに、イベントの発火を行うために `exec` メソッドを追加した。`exec` メソッドは、引数に関数リテラルを取る。他の描画コマンドと同様に、タートルのキューに蓄積され、順に実行される。`exec` メソッドで追加したコマンドが

```
var g1, i;  
  
$(document).ready(function () {  
    g1 = new Gunmachan();  
    for(i = 0; i < 3; i = i + 1) {  
        g1.ahead(100);  
        g1.right(180 - 60);  
    }  
    g1.exec(function () {  
        $(document).trigger("done");  
    });  
});
```

図 4 正三角形を描くプログラム (オブジェクト指向版)

Fig. 4 A program to draw a triangle (Object-Oriented version)

実行されると、引数に渡した関数リテラルが実行される。関数リテラルには何の制限も加えていないので、任意の JavaScript のコードが実行可能である。イベントを発火させるときは、要素集合オブジェクトの `trigger` メソッドを用いて発火させる。

図 4 は、拡張したライブラリを用いて正三角形を描くプログラムの例である。正三角形を描き終わった後、イベント `done` を発火させる。

オブジェクト指向プログラミングを導入した点以外で、手続き型プログラミング版 (図 3) との差異は、HTML の読み込みが完了 (`ready` イベントの発火) するまで待つためのコードが追加されている点である。手続き型プログラミング版では、HTML キャンバス要素を静的に配置することができるが、オブジェクト指向プログラミング版では、タートルの生成時に動的に HTML キャンバス要素を生成する必要がある。HTML の読み込みが完了するまでは、キャンバス要素の追加ができないため、このような処理を追加する必要がある。従って、この版では、`turtle.html` を読み込んだだけでは、タートルは表示されておらず、明示的にインスタンスを作成する必要がある。

5. 授業における利用

2 年次後期のオブジェクト指向プログラミングの授業において、次のように導入した。履修者は 12 名と聴講生 3 名の小規模クラスであるが、情報・経営を専攻する学生は 1 学年 50~60 名であるため、2~3 割の学生が受講している。男女比は、男性が 7 名、女性が 8 名である。受講生は 1 名を除き、昨年度開講のアルゴリズムの授業において、初歩的なタートルグラフィックスを体験している。

授業の冒頭で、開発したオブジェクト指向プログラミング版のタートルグラフィックスのライブラリを配布した。

最初に、コンストラクタとインスタンスの間の関係を説明し、コンストラクタに対して `new` 演算子を適用することで、インスタンスが生成されることを説明した。次に、インスタンスを変数に代入し、そのインスタンスのメソッドを実行する方法を説明した。続けて、複数のタートルを作

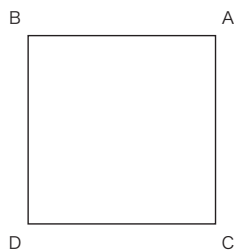


図 5 正方形 ABCD
Fig. 5 a square ABCD

```
var g1, g2;

$(document).ready(function () {
  g1 = new Gunmakan();
  g2 = new Gunmakan();
  g1.ahead(100); // 辺 AC
  g1.right(90);
  g1.ahead(100); // 辺 CD
  g2.right(90);
  g2.ahead(100); // 辺 AB
  g2.left(90);
  g2.ahead(100); // 辺 BD
})
```

図 6 2つのタートルで正方形を描くプログラム
Fig. 6 A program to draw a square with two turtles.

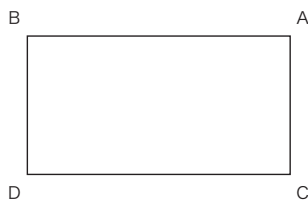


図 7 長方形 ABCD
Fig. 7 a rectangle ABCD

成することができ、それらを別々に動かすことができることを解説した。これらを使って、2つのタートルを作成し、図5に示す正方形 ABCD を、2つのタートルを用いて描けることを図6のプログラムを紹介して例示した。図5のプログラムでは、変数 g1 のタートルが辺 AC, CD の順に描き、g2 のタートルが辺 AB, BD の順に描く。

次に、図7に示す、長方形 ABCD を描くという例題を用いて、タートルの動きに対して、処理を同期させる方法について説明した。長方形では、辺の長さが異なるため、辺 AC と AB を描くタートルの移動時間が異なる。このため、より時間のかかる辺 AB のタートルが描画終了時にイベントを発火させ、辺 AC と AB の両方が描き終わった後、残りの辺 CD と BD を描き始める方法を説明した。図8では、変数 g2 のタートルが辺 AB を描き終わったあとにイベント step1 を発火させ、そのイベントハンドラの中で、辺 CD と BD を描くコマンドを追加している。

最後に、長方形の例を用いて、setTimeout を用いた時間による待ち合わせについても説明した。

```
var g1, g2;

$(document).ready(function () {
  g1 = new Gunmakan();
  g2 = new Gunmakan();

  g1.ahead(100); // 辺 AC

  g2.right(90);
  g2.ahead(150); // 辺 AB
  g2.exec(function () {
    $(document).trigger("step1");
  });
});

$(document).bind("step1", function () {
  g1.right(90);
  g1.ahead(150); // 辺 CD
  g2.left(90);
  g2.ahead(100); // 辺 BD
});
```

図 8 2つのタートルで長方形を描くプログラム
Fig. 8 A program to draw a rectangle with two turtles.

5.1 ぐんまちゃんダンス

オブジェクト指向プログラミングの授業において、イベントの発火とイベントハンドラに関する理解と、これらの知識の定着を目的とし、受講生に対して次のような課題を課した。

ぐんまちゃんダンスとは、画面に複数のタートル（ぐんまちゃん）を表示させ、それらを音楽に合わせて動かすプログラムを書くという課題である。作成の条件として、複数のタートルを表示させること、イベントまたはタイマーにより複数のタートルを同期させることの2点を付けた。音楽に合わせるという点は、課題の狙いからは必ずしも必要ではないので、必須の条件とはしなかった。

筆者が作成した音楽付きのサンプルを受講生に見せて完成のイメージをつかませた。作成の期間は次回の授業まで（一週間）とした。ちなみに、ダンスとしたのは、中学校でのダンス必修化から着想したものである。

課題を出してすぐ、ほとんどの受講生から数多くの質問を受けた。その内容は、「初期位置を変更することはできるか」や「前を向いたまま横に移動させることはできるか」など、「このような表現をすることはできるか」や「どのようにしたら表現できるか」といったものであり、中には「ライブラリを改造して新しい動きを追加してもよいか」といったものもあった。ほとんどの受講生は、ダンスという題材から自ら表現したいイメージを明確にしていると考えられる。

提出された課題の作例を図9に示す。ダンスは動きであり、紙面で示すことが難しいため、本稿では、従来のタートルグラフィックスの機能を使って画面に描画を行いながらダンスを実装した作品から2例を示す。それぞれ、別の

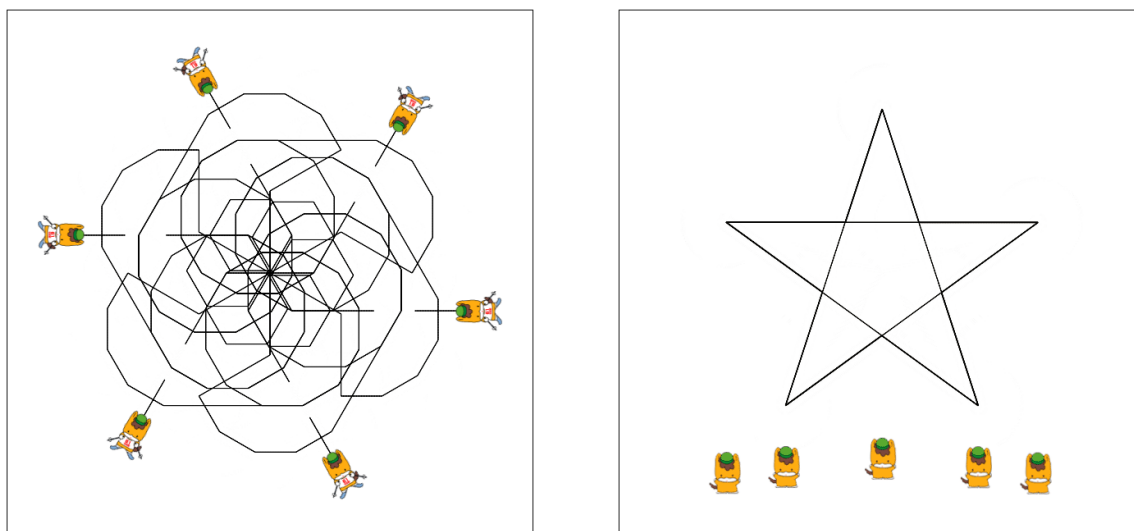


図 9 2人の学生の作品例

Fig. 9 two dances which are submitted by two students

学生による作品である。図の左の作品は、中央から次第に花のような幾何学的な模様をタイミングをずらしながら描いていくものである。図の右の作品は、しばらく何も描画せずにリズムカルにぐんまちゃんが動きまわった後、5匹のぐんまちゃんが所定の位置に散らばり、タイミングを合わせて星の模様を描いた後、図の位置まで移動するものである。これ以外にも、ユニークな動きをする作品が多いが、紙面の都合上割愛する。

履修者12名のうち、9名の学生から課題の提出があった。ただし、9名のうち1名は聴講の学生である。提出された課題について、学生ごとに作成したコードの概要を表1に示す。bindの数はイベントハンドラを設定した箇所の数を、triggerの数はイベントの発火を記述した箇所の数を示している。個別に測定したのは、イベントの発火による分岐や反復が行われているかどうかを知るためであったが、そのような記述は見られなかった。setTimeoutの数は、時間による制御を行った箇所の数である。コードの総行数は、学生が記述したコードの行数（コメントや空行を含む）である。

音楽との同期は必須としなかったが、表1に示すように、3人の学生が音楽との同期を行うプログラムを作成した。音楽を付けた学生は、いずれも女性であった。学生Hは、実際にライブラリの改造も行った。

6. 議論

タートルグラフィックスを拡張したライブラリを用いたダンスの課題は、現代の学生にとって、性別に関係なく楽しく学べる内容の一つと言える。

魅力あるダンスを実現するには、かなりのプログラミングを要するものであり、必然的に反復学習を行うことになるが、出来上がった作品を見る限り、反復学習に拒否反応

を示している受講生はほとんど見られない。提出作品のほとんどが100行を超え、半数が200行を超えており、半数以上の学生が複数回のイベントの発火を使用している。このため、イベント駆動型プログラミングの概念を定着させるための初期の課題として有効であると考えられる。

イベントの発火を利用した反復などを着想した学生はいなかった。初めて学んだ概念を説明した例を超えて使いこなすのは難しく、繰り返しを行う例を紹介するべきであったと考えている。

ダンスの課題は、多くの知識を必要としないので、オブジェクト指向を学ぶ初期の段階でも十分に導入可能である。

タートルにぐんまちゃんを採用した点についても有効であったと考えている。開発したライブラリでは、ペンを上げている状態と下げている状態では、ぐんまちゃんは違うポーズをとる。このことを利用したダンスも数多く見られた。受講生が他のポーズのぐんまちゃんを追加できるように改良することを計画している。

別の授業を担当している教員から、「受講生が見せてくれた」という話を聞いた。このことも、ぐんまちゃんダンスが受講生にとって印象深い課題となったことの現れであると考えている。

イベントハンドラの設定やイベントの発火は、本来はCSSセレクタによって絞り込んだ要素に対してだけ行う。これによってイベント処理を絞り込むことや、複数の要素を伝播するイベントなどを作成できる。本教材では、これらの機能については扱っていない。

7. 関連研究

イベント駆動型プログラミングの教育効果については、Bishop-Clarkの研究[2]がある。イベント駆動型プログラミングは、学生が普段触れているソフトウェアに近いアブ

学生	bind の数	trigger の数	setTimeout の数	コードの総行数	備考
A	16	16	9	205	
B	5	5	0	296	
C	1	1	0	108	
D	4	4	4	121	
E	1	1	1	117	
F	3	3	0	224	
G	6	6	3	225	音楽あり
H	15	15	12	207	音楽あり
I	1	1	0	64	音楽あり

表 1 学生が作成したコードの概要
Table 1 a summary of code written by students

リケーションを手軽に作れるため、学生の興味を惹きやすい。本研究で提案した教材は、そうしたアプリケーションとはかけ離れているが、学生の興味を惹いた点が新しい。

辰己等は、文献 [13] の中で、ドリトルの音楽演奏機能を使用したプログラミングの教育法を提案している。タートルの動きを音の演奏に置き換えて、手続き型言語の入門に利用している。本研究は、イベント駆動プログラミングの理解に焦点を当てているが、ダンスパフォーマンスの一環として、効果音の追加や音楽演奏が終了したタイミングでイベントを発生させるなど、本研究を改良するにあたり参考にすべき点が多い。特に本研究では、音楽との同期は時間だけで行っており、必ずしも十分とは言えない。

8. 結論

手続き型プログラミングにおける変数や制御構造を理解した学習者を想定し、タートルグラフィックスを用いてイベント駆動型プログラミングを導入する教材を提案した。さらに、複数のタートルを用いた画面上でのパフォーマンスをさせる“ぐんまちゃんダンス”の課題を提案した。この課題を、いわゆる“文系”の大学において学生に課したところ、学生の創作意欲を刺激し、多くの力作が提出され、高い教育効果があることを確認した。

タートルグラフィックスは、現在でも有効な教材であり、味付けの工夫も重要である。筆者は、システム開発 II の授業において、複数の受講生の PC を WebSocket によるリアルタイム通信で接続し、受講生 1 人 1 人が 1 つのタートルを操作し、1 枚のキャンバスを共有しながら共同で 1 枚の絵を描くという課題を課したところ、これも非常に好評であった。この点についても機会を得て発表したい。

受講生の中で要望が強かった円弧を描く機能については、メソッド `arc` として既に実装した。

受講生の中には、`right(10000);` のように入力して、ぐんまちゃんをグルグル回して楽しんでいる者もいるため、一定回数以上回ったらぐんまちゃんが目を回す機能の導入を検討している。この機能がダンスの中にどのように使用

されるかも楽しみである。

参考文献

- [1] Abelson H. and diSessa, A.: *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*, Artificial Intelligence Series, The MIT Press (1986).
- [2] Bishop-Clark, C.: Comparing Understanding of Programming Design Concepts Using Visual Basic and Traditional Basic, *Journal of Educational Computing Research*, Vol. 18, No. 1, 1998. pp.37-47.
- [3] Crockford, D.: *JavaScript: The Good Parts*, O'Reilly & Associates Inc., 2008.
- [4] ECMAScript Language Specification. 2011. available from <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf> (2013.02.13)
- [5] Express - node.js web application framework available from <http://expressjs.com> (2013.02.13).
- [6] Flanagan, D.: *JavaScript: The Definitive Guide 6th Edition*, O'Reilly & Associates Inc, 2011.
- [7] jQuery, available from <http://jquery.com> (2013.02.13).
- [8] jQuery Mobile, available from <http://jquerymobile.com> (2013.02.13).
- [9] Node.js, available from <http://nodejs.org> (2013.02.13).
- [10] Socket.IO: the cross-browser WebSocket for realtime apps. available from <http://socket.io> (2013.02.13)
- [11] 栗山裕, 橋下友茂, 山下利之: ゲームプログラミングによる情報教育の評価方法. 日本教育工学会論文誌 28, pp.181-184, 2005.
- [12] 群馬県のマスコット「ぐんまちゃん」. 入手先 <http://www.pref.gunma.jp/01/b0110211.html> (2013.02.13)
- [13] 辰己丈夫, 兼宗進, 久野靖.: ドリトルと「情報教育の音楽化」. 情報処理学会 コンピュータと教育研究会 第 82 回研究会論文集, pp. 77-84.
- [14] 野口孝文: ゲーム作成を課題にしたプログラミング教育とその分析方法の開発. 電子情報通信学会技術研究報告 ET, 教育工学, Vol. 104 No. 222, pp. 1-6, 2004.