

JSLesson - 高校生向け JavaScript 学習環境

長島 和平^{†1} 長 慎也^{†1} 間辺 広樹^{†2}
並木 美太郎^{†3} 兼宗 進^{†4}

高校の科目「情報の科学」において、JavaScript によるプログラミングを扱うことがある。高校における JavaScript の授業はこれまで、テキストエディタなどを使って行われてきた。しかしこの方法では、開発環境と実行環境が別であるためデバッグしにくいことや、生徒の身近な題材としてよく扱われるゲーム制作の難易度が高いこと、生徒間で各自の作品を共有することが難しいことなどの問題がある。そこで、ブラウザ上でコーディングと実行ができる環境「JSLesson」を制作した。JSLesson は、生徒がゲームを制作するための助けとなるような API をいくつか用意した。また、サーバ上で動いているため、他の生徒と作品を共有することができる。本発表では、JSLesson を、高校で行われる「情報の科学」の授業で使った結果を報告する。

JSLesson – JavaScript Learning Environment for High School Students

KAZUHEI NAGASHIMA^{†1} SHINYA CHO^{†1} HIROKI MANABE^{†2}
MITAROU NAMIKI^{†3} SUSUMU KANEMUNE^{†4}

JavaScript is used at some classes of "Information Science" in high school. Most textbooks of the subject make students code JavaScript using simple text editor and run program by web browser. However, this method has some problems: coding and running environment are different one, it is difficult to make games which students are familiar with and students are hard to share their programs with others. We developed "JSLesson". JSLesson has some APIs to help students to coding easily. Because it is running on server, students can share their program. This article report how it used in high school.

1. はじめに

高校の情報の授業において、プログラミングを取り入れることがある[1]。その中には、生徒の意欲を向上させるためにゲーム制作などを扱った実践例もある[2]。教科書の中には、JavaScript を使用したものもある[3][4]。実際に使用されている教科書には、順次、反復、分岐といった概念を学習するためのサンプルプログラムが記載されているが、その多くは計算を行った結果の文字列が画面上に表示されるような単純なものである。これらのサンプルプログラムは、学習者である高校生にとって身近な内容ではない。また、開発環境も生徒にとっては扱いが難しい。

そこで、JavaScript を使って生徒にとって身近な作品を、簡単に制作できるよう配慮した環境 JSLesson[5]を開発した。JSLesson では、生徒にとって身近なアニメーションやゲームを題材とした作品作りを行うことができるため、生徒のモチベーションを高めることができると考えられる。また、コード記述と動作確認を一つのブラウザ上でできるため、生徒にとって扱いやすい。

JSLesson を、文部科学省委託事業のプログラミング教育実証授業[6]にて用いた。そこで得られた生徒の反応や授業

の進行状況などを元に、JSLesson を用いて JavaScript の学習を行う利点を述べる。

2. JavaScript を用いたプログラミング授業

2.1 高校における JavaScript によるプログラミング学習の現状

高校の科目「情報の科学」においてプログラミングを学習する場合、教科書に JavaScript が使われることがある[3][4]。JavaScript と HTML を学習することを通じて、普段何気なく使用している Web アプリなどが動く仕組みなどに直接触れることができ、情報学への興味を惹きつけることもできる。また、プログラミングの授業を通じて、順次、反復、分岐といったプログラミングの基本的な概念を学習させることが望ましい。

現在実際に使用されている教科書では、JavaScript の標準機能のみが使用されており、そのコードは計算問題や、並べ替えを扱ったものがほとんどである。また、実行結果も画面に表示させるだけのものとなっている。生徒の学習意欲向上のために、ゲーム制作を題材とした授業を展開したいが、高校で使われる教科書に JavaScript を用いてゲームを制作するサンプルはない。

2.2 言語仕様・ライブラリの問題

JavaScript と HTML を用いて簡単なゲームを制作するた

^{†1} 明星大学

^{†2} 神奈川県立柏陽高等学校

^{†3} 東京農工大学

^{†4} 大阪電気通信大学

めには、アニメーションをさせる必要がある。例えば、キャラクタを往復させようとする、図 1 のようなコードを記述する必要がある。

```
x=0;
y=0;
state=0;
setInterval(loop,30);

function loop() {
    var s=document.
        getElementById("hello").style;
    s.position="absolute";
    s.left=x+"px";
    s.top=y+"px";
    if (state==0) {
        x+=3;
        if (x>300) state=1;
    }
    if (state==1) {
        x-=3;
        if (x<0) state=0;
    }
}
```

図 1 キャラクタを往復させるプログラム

JavaScript を用いてゲーム制作することを題材として授業を進めるには、次のような言語仕様・ライブラリの問題点を解消しなければならない。

(1) 打ち込まなければいけない文字数が多い

図 1 に示したキャラクタを往復させるプログラムでは、"hello" という id を持つ要素を取得するために、document.getElementById("hello").style という長い文を打ち込む必要がある。書くプログラムの長さが長くなれば、ミスタイプをする確率も上がり、これだけの長い文章を打ち込む間にミスタイプを犯せば、生徒がエラーの起こった場所を確認することも難しくなる。

要素を取得するためのプログラムは、初学者が学習すべきプログラミングの基本的な概念ではない。そのため、必要最小限のプログラム以外は入力しなくて済むような API を用意し、概念の学習に専念できるような環境を整えることが望ましいと考えられる。

(2) フレームごとの繰り返し処理にイベント駆動が使われる

JavaScript でゲーム制作に必要なフレームごとの繰り返し処理を記述するためには、図 1 に示したように setInterval を使って一定間隔ごとに関数を呼び出すイベント駆動を扱う必要がある。この記述方法を使わなければならないことで、フレームごとの繰り返し処理に while 文や for 文といった繰り返し文を使用することができない。それだけでなく、現在動いている方向を保持するためにステータス変数を使用しなければならない[7]。

教科書中には、反復構造に関する記述もあり、フレームごとに繰り返し行う処理に while 文などを使用することが

反復の概念を学習する機会となると考えられる。しかし、JavaScript の標準機能のみではこれを実現することができないため、while 文や for 文を用いてフレームごとの繰り返し処理を記述することができるような拡張機能を使用することが望ましい。

以上のような点を踏まえると、JavaScript の標準機能のみを使用してプログラミング授業を行った場合、ゲームを制作することはかえって生徒の負担となってしまう可能性があり、教科書に書かれたようなメッセージの表示や簡単な計算といった単純な動作をするプログラムしか作ることができない。

2.3 学習環境の問題

これまで高校で JavaScript を扱う場合、テキストエディタにコードを入力し、ブラウザでコードの動作を確認する方法で授業を行っている。教科書中にも、プログラムの組み方の説明として、次のような手順が示されている。

1. テキストエディタでプログラムを入力する。
2. プログラムに名前をつけて保存する。拡張子は「.html」にする。
3. プログラムをブラウザに読み込み、動作を確認する。
4. 想定した動作をしない場合にはデバッグする。

この環境で授業を進めることで生じる問題がいくつか挙げられる。

(1) 複数のウィンドウを使わなければならない

テキストエディタでプログラムを打ち込み、それをブラウザで確認しなければならないことは、作業効率に悪影響を及ぼす。また、頻繁にウィンドウを切り替える必要があることが、生徒への混乱を招く可能性もある。

(2) エラーを容易に確認することができない

テキストエディタとブラウザを使用してプログラムを作成し、プログラム内にエラーがあった場合、そのエラーメッセージを確認するためにはブラウザ上でコンソールを開く必要がある。コンソールを開くという行為は日常生活でコンピュータを使用しているときには行うことがない行為で、エラーを確認するためにこの無駄な手順を踏まなければならないことは生徒にとって負担となる。また、ここで表示されるエラーメッセージも、ただで初心が者が問題の発生した場所をすぐに確認できるものではなく、起こったエラーを生徒が自力で解決することは困難であると考えられる。

(3) プログラムにインデントを付けることができない

テキストエディタでプログラムを入力していると、自分でスペースやタブを打ち込まなければインデントを付けることができない。しかし、初めてプログラミングに触れる生徒が自らインデントを挿入するとは考えにくい。プログ

ラム中にインデントが入っていることで、繰り返しや分岐などの範囲を把握しやすくなるが、インデントが入っていないプログラムは、プログラミング経験者にとっても読みにくいものである。

(4) サーバにアップロードするなどして、ネットワーク上で共有することができない

作成したプログラムをネットワーク上で公開し、生徒間で共有したりする場合には、動作を確認した後、アップローダを使用してネットワーク上にアップロードする必要がある。しかし、高校の授業環境ではファイルをアップロードするというところまで行うことなくカリキュラムを終えていることがほとんどである。

(5) ファイルの保存し忘れや、開く際にテキストエディタで開くか、ブラウザで開くかが分かりにくい

通常 HTML ファイルを開こうとすると、ブラウザを使ってファイルが開かれる。そのため、コードの編集を行いたい場合は、そのファイルを右クリックし、「プログラムから開く」→「メモ帳」などを選択して開かなければならない。そして開いたファイルにコードを記述したあと、実行前に保存を忘れることもあり、記述したコードにエラーがあると勘違いしてしまうこともある。

授業を受ける生徒全員が日常的にコンピュータに触れているとは考えにくく、また、日常的にコンピュータに触れていたとしてもプログラミングを学習することは初めての生徒がほとんどである。にもかかわらず、これだけの問題点がある中でプログラミングを学習することは、プログラミングの本質や概念を学習することの妨げとなる可能性がある。学習者が、プログラミングを学習することに集中できる環境を用いて授業を進めることが重要であると考えられる。

3. Web ブラウザで JavaScript を学習できる既存の環境

前述した問題点のうち、既存の環境で解決が可能であるものは限られている。複数のウィンドウを遷移しなければいけない点と、保存忘れやファイルの開き方といった点は、Web ブラウザ上で JavaScript のコーディングができる環境があれば解消することができる。また、生徒同士がお互いのコードを共有する体験を通して、ネットワークの仕組みを学ぶことにつなげることができる。

Web ブラウザ上で JavaScript を使うことができるプログラミング環境は複数存在する。jsdo.it[8]、JS Bin[9]、JSFiddle[10]、Coding Tab[11]といった環境が、Web ブラウザ上でコーディングと実行ができる環境として挙げられる。これらの環境を扱うことでウィンドウ遷移、保存忘れ、ファイルの開き方の問題は解決される。しかし、jsdo.it、JS Bin、

JSFiddle はユーザ登録が必要である。ログインするためには生徒それぞれに google アカウントなどを発行し、生徒が各自そのアカウントを管理する必要がある、現実的ではない。Coding Tab は、ユーザ登録は必要としないが、コードを保存することができない。したがって、テキストファイルなどに保存することを忘れてディスプレイを閉じてしまうと記述していたコードが消えてしまう。

JavaScript の標準機能のみを使用していることから起こる問題は、適切なフレームワークを使用して解決できる可能性がある。しかし、現在使われる多くのゲーム制作用フレームワークでは、イベント駆動でフレームごとの動作を記述したり、関数を値として扱っていたりと、JavaScript そのものの書き方が多く残っている。これらのフレームワークは、ゲーム制作を補助するために開発されたものであり、教育に使用することは考えられていない。また、プログラミングに初めて触れる生徒にフレームワークのことを考えさせることも適切ではないと考えられる。

4. JSLesson

著者らは高校生向け JavaScript 学習環境として JSLesson を開発した。JSLesson では、Web ブラウザ上でコーディングと動作確認を行うことできる(図 2)。HTML タブ、または JavaScript タブを選択することで、コード編集領域でコーディングをすることができる。コードの実行を行うと、実行画面でコードの内容が実行されるため、この一つの画面でコーディングと動作確認を行うことができる。



図 2 JSLesson の画面

また、記述したコードにエラーがあった場合も、図 3 のように画面上にエラーの内容と、その場所が!マークで表示されるため、エラーを調べるためにコンソールを開くような特別な手段を用いる必要がなくなっている。

そして、実行の際にファイルの自動保存と自動インデントを行う。また、自動インデントについては、改行時にもインデント済みの場所にカーソルが合う。さらに、if 文や while 文などで「{」を入力し改行すると、「}」が自動的に入力されるため、括弧の閉じ忘れを防ぐことができる。



図 3 JSLesson のエラー表示(セミコロン忘れ)

高校において、ファイルのアップロードなどが行われなかった点についても、JSLesson をサーバ上で動かし、スマートフォンで動かせる QR コードの発行を行ったことで、生徒間でお互いの作品を共有することができるようになる。生徒が日常的に使用しているスマートフォン上で自分自身が作ったプログラムを動かせることは、生徒にとってもよいモチベーションになると考えられる。

以上の点の実装により、前述した開発環境の問題を解消することができる。

言語仕様の問題点として指摘した、フレームごとの繰り返し処理にイベント駆動を使用しなければいけないことの解決策として、JSLesson では、フレームの処理をその場で終了させる wait 命令を用意した。wait 命令を用いることで、フレームごとの繰り返し処理に while 文を使用することができる。図 4、図 5 は、図 1 で示した JavaScript でキャラクタを往復させるプログラムを JSLesson で用意した API を用いて記述したプログラムである。JavaScript の標準機能のみを用いて記述した図 1 のプログラムとは異なり、アニメーションを反復構造を用いて記述できる。また、これに伴い、ステート変数も使用する必要がなくなっている。

```
<html>
  <body>
    <span name="hello">Hello</span>
  </body>
</html>
```

図 4 キャラクタを往復させるプログラム
(JSLesson : HTML)

この実装により、フレームごとの繰り返し処理に while 文を使って記述することができ、反復構造の学習が可能になる。

記述する文字数が多いという点について、生徒が記述しなければならない文字が少なくなるよう用意した JSLesson で使用できる命令一覧を表 1 に示す。

```
// Javascript
x=0;
y=0;
while(true){
  while(x<300) {
    move("hello",x,y);
    x+=3;
    wait(30);
  }
  while(x>0) {
    move("hello",x,y);
    x-=3;
    wait(30);
  }
}
```

図 5 キャラクタを往復させるプログラム
(JSLesson : JavaScript)

表 1 JSLesson の API

命令	説明
move (要素名, x, y)	要素を x, y 座標に移動させる。
rotate (要素名, 角度)	要素を指定の角度回転させる。
resize (要素名, 横, 縦)	要素を指定の比率に拡大/縮小する。
changeImage (要素名, 画像アドレス)	要素の画像をアドレスのものに変更する。
setText (要素名, 文字)	要素に文字を設定する。
addText (要素名, 文字)	要素に文字を書き足す。
onClick (要素名, 命令)	要素がクリックされたときに行う動作を設定する。
onTouch (命令)	画面がタッチされたときの動作を設定する。実行される命令ではタッチの情報を受け取ることができる。
transform (要素名, 角度, 横, 縦)	要素の rotate と resize を同時に行う。
wait (時間)	指定された時間 (ms) 処理を中断。
rnd (値)	0～値未満の整数を返す
setBGColor (色)	指定の色に背景色を変える

JSLesson において、HTML で画面上に表示した要素を操作するときは、name 属性の値を指定する。図 5 で用いられている move 命令は、図 4 で "hello" という name 属性を持たせた要素を引数で指定した座標へ動かす動作を行う。

要素を取得することや、イベントの設定といった、プログラミングの本質でない機能を使用することに対する苦労を極力抑え、プログラミングの本質を学習することができるよう API を用意した。この API を使うことで、命令の中

で要素の name 属性の値を指定するだけで要素を操作することができる。この API はゲーム制作を通じてプログラミングを学習するため、簡単なゲーム制作をする際に使われる動作を補助できるものを選択した。

5. JSLesson を用いた授業

JSLesson を、2015 年度文部科学省委託事業のプログラミング教育実証授業にて用いた[6]。実証授業の対象となったのは神奈川県立柏陽高等学校の 1 年生のクラスである。授業は、はじめにドリトルを用いてプログラミングへの導入の授業が 3 コマ、HTML と JavaScript を用いて Web ページの作り方、ゲーム制作を通じたプログラミング学習を扱う授業が 5 コマ、PHP を用いて Web の仕組みとサーバサイド技術を学習する授業が 2 コマ、合計で 10 コマ行われた。JSLesson はこのうち、HTML と JavaScript を使用した 5 コマ目から 8 コマ目で使用された[12]。

5.1 教材の作成

JSLesson の API を用いて JavaScript を学習するための教材を作成した。用意する教材では、はじめに HTML について学習し、続いて JavaScript で HTML の要素を操作する方法を学習する。

HTML では、文字を画面に出す、背景色や文字の色を変更する、画像を画面に出すという内容を順に学習する。HTML を教材通りに制作すると、図 6 のようなプログラムが完成する。

```
<html>
<body bgcolor="green">
  <h1>こんにちは</h1>
  <span><font color="white">HTML</font>
  の勉強をしています。</span>
  <span>次の文章です。</span>
  <div>改行して次の文章です。</div>
  ねこ<br/>
  ボール<br/>
</body>
</html>
```

図 6 教材に沿って制作した HTML プログラム

HTML で静的なコンテンツの作り方に触れた後、JavaScript のプログラミングに移行する。JavaScript を使って、HTML 上の要素を動かすプログラムを制作する。HTML には要素を一つだけ置き、name 属性をつけておき、JSLesson の move 命令でその要素を移動させる。次に wait 命令も使用し、少しずつ要素を移動させ、移動する座標を保持させるために変数の概念を学習させる。そして、移動をさせ続けるために繰り返しの概念を使用し、毎フレーム要素を操作させる。wait 命令があることで、図 7 のように while 文を使ってフレームごとの繰り返しを記述させることができる。

```
x=100;
while(true){
  move("moji",x,100);
  wait(1000);
  x+=10;
}
```

図 7 繰り返し要素を動かすプログラム

ここから、ゲーム制作を通じてプログラミング学習をさせる。ゲームを制作するためには、画面に出したキャラクターを移動させる必要がある。画面にボタンを設置し、ボタンがクリックされたときに行う処理を関数に書かせる。onClick 命令でボタンがクリックされたときの処理を記述させる。入力を受け付けることは、毎フレーム繰り返す処理ではないため、ここではイベント駆動を使用する。ボタンでキャラクターを動かせるようになったあとは、ボール型の敵を画面に出す。敵の動きをまとめた関数を作らせ、フレームごとの繰り返しから呼ぶ。その中で、敵は画面下に落ちたらまた画面の上に出てくるよう記述する。その後衝突判定を計算し、ぶつかったらキャラクターの y を -100 にすることで画面から消す。敵を避けた数だけ加算されるスコア表示をできるようにする。ここまでの内容を記述したゲームのコードを図 8 に示す。また、ここまでで完成するゲームの画面を図 9 に示す。

その後、発展的な内容として、敵を複数体画面に出現させ、それらを操作するために配列を扱った。配列で複数の敵を扱うため、敵の name 属性の値は"ball10", "ball11", "ball12" ...とした。

```
x=100;y=300;bx=100;by=10;score=0;
onClick("right",right);
onClick("left",left);
while(true){
  move("neko",x,y);
  moveBall();
  wait(50);
}
function moveBall(){
  move("ball",bx,by);
  by+=10;
  if(by>400){
    by=0;bx=rnd(400);
    score+=10;
    setText("score",score);
  }
  if(bx>x-30 && bx<x+30 &&
  by>y-30 && by<y+30){
    y=-100;
  }
}
function left(){
  x-=10;
}
function right(){
  x+=10;
}
```

図 8 教材に沿って制作したゲームのプログラム

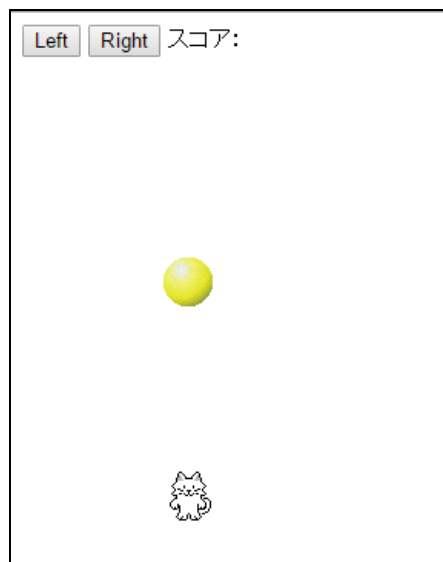


図 9 教材に沿って制作したゲームのキャプチャ

5.2 生徒へのアンケート

全ての授業を終えた生徒へのアンケートを行った。なお、授業を受けた生徒は約 310 人いたが、その中から本稿執筆時においてアンケート調査が完了している 76 人分の回答を元に報告する。

もう一度学びたいかという質問に対し、「少し思う」、または「強く思う」を選んだ生徒の数は、JSLesson を用いた 5 回目から 8 回目の授業で毎回 8 割ほどに上った。中でも、「強く思う」の割合が大きかった授業は、PHP を扱った 9 回目の授業(51.3%)、JavaScript で HTML 上の要素を動かした 5 回目の授業(50.0%)、JavaScript のゲーム制作を行った 7、8 回目の授業(46.1%、48.7%)の 4 つの授業で、それぞれ半数近い生徒の学習意欲を大きく刺激することができた。

PHP の授業では、スマートフォンアプリの「LINE」に似た画面を扱っていた。普段触れているアプリや、ゲームの仕組みを見られることが、生徒のモチベーションに大きな影響を与えたと考えられる[12]。

JSLesson の環境で使いやすかった点を自由記述で回答してもらった。表 2 には、その自由記述の回答から、特に多く寄せられた回答についてまとめた。

表 2 JSLesson の使いやすかった点として得られた回答

エラー表示(ポップアップ、×マーク)	16
対応する記号が自動で出てくる("など)	15
実行画面がすぐ見られる	8
HTML と JS の切り替え	8
コメントや文字列などの色付けがされている	7
その他レイアウトが見やすい	6
教材、命令リストの冊子があったこと	5

表 2 から、「エラーの表示があることで間違えた場所が分かりやすいこと」が使いやすかった点として最も多く挙げられた。初めてプログラミングを行う生徒にとって、エラーは避けることができないことであり、エラーが起きた時に分かりやすく表示をさせたことがプログラミング学習のサポートに役立っていたと考えられる。

次に多かった回答は、「対応する記号が自動的に出てくること」であった。初心者が起こしやすいエラーの一つに、対応する括弧の閉じ忘れがある。例えば、while 文の「{」を記述し、改行すると、自動的に「}」が入力される仕組みになっていたことで、少なからずエラーを未然に防ぐことができたと考えられる。

続いて多く回答が得られたのは、「実行画面がすぐに見られる」点と、「HTML と JavaScript を簡単に切り替えることができる」点であった。コーディングと実行を一つの画面でできることで、少し書き足してこまめに実行することができ、使いやすと感じたようである。また、HTML と JavaScript をすぐに切り替えることができ、そのタブに色付けもされていることから、どちらを記述しているかが分かりやすかったという感想や、画面の静的な見た目とその振る舞いを記述する場所が違っていたので使いやすかったという感想も得ることができた。

表 3 には、JSLesson の使いにくかった点を自由記述してもらった回答から、特に多かったものを抜粋して示す。

表 3 JSLesson の使いにくかった点として得られた回答

特になし	21
プログラミングが難しい・打ち込む量が多い	10
ミスがあると動かない	9
画面の切り替え	8
エラーがどこかわかりにくい	5
バックアップがされない	5
画面サイズ	4

表 3 の回答から、最も多く得られたのは使いにくい点は特になかったという回答である。また、次に多かった点は、プログラミングが難しく、書く量が多いと感じたことや、少しでもタイプミスやセミコロン忘れをすると動かなかったことである。なるべく生徒が記述しなければいけないプログラム量を減らすために API を準備したが、これでも多く感じた生徒がいたということは、この API がなければさらに多くの生徒がプログラム量に苦しんでいたことが想像できる。一方、タイプミスなどにより、プログラムが動かなかったことを挙げた生徒もいたが、ミスなく記述しなければ動作しないことを知ることもプログラミングを学ぶうえで重要なことであり、この苦労を経験することも必要であったと考えられる。

画面の切り替えに関する点が次に多く挙げられた。JSLesson ではコード記述後、プログラムを実行させると、フォーカスが実行画面に移動する。そのため、実行させた後にコードを記述するためには、コード編集エリアを一度クリックしなければならない。このことが生徒にとっては少し面倒に感じたようである。しかし、本授業で制作したゲームでは扱わなかったが、キーボードからのキー入力を受け付けるゲームを制作しようとした場合、フォーカスがコード編集エリアに残っていると、ゲーム中に入力した内容がコード内にも反映されてしまい、コードの中身を破壊してしまうため、この仕様を採用している。また、このクリックを忘れたために、バックスペースキーを押してしまい、コード編集画面から一つ前の画面に戻ってしまうこともあった。この点は、バックスペースキーが押された際にアラートを表示し、前の画面に戻ってもよいか確認を取ることで対策をとったため、本授業の途中から改善された。

また、使いやすかった点で多く挙げられたエラー表示に関して、使いにくかった点としても回答があった。これは、一部のエラーは正確にエラーが発生されている場所を指し示すことができないことや、エラーメッセージもすべて日本語に対応しているわけではないため、読めなかったということであった。

バックアップができなかったという問題も、使いにくかった点として報告された。これは、サーバと同期をとるタイミングが、ホーム画面に戻った時であったため、コード編集画面にいる状態からウィンドウを閉じてしまったとき、変更した内容がサーバに反映されていなかった。このため、次に開いたときに前回分のコードをサーバから取得することができなくなっていた。

ゲーム制作を教材のサンプル通りに制作することができたか、という問いには、76人中67人の生徒ができたと答えた。また、ゲーム制作において苦労した点を自由記述で回答してもらった。その結果を表4に示す。

表4 ゲーム制作において苦労した点(自由記述)の回答

エラーに苦労した	40
特になし	8
文の意味の理解	8
打ち込むプログラム量	6
時間不足	6
数値設定	3
バランス調整	1

半数以上の生徒が、「エラーの場所を探し出すこと」や、「たくさんエラーを出してしまったこと」を苦労した点として挙げた。プログラミングにおいてエラーは避けて通ることはできないことで、それを経験することもプログラミ

ング学習の一部である。

また、命令の意味を理解することに苦労したと答えた生徒もあり、教材上でより詳細な説明を記載する必要があると思われる。

さらに、「プログラムの量が多かったこと」、「時間が足りなかったこと」などを挙げた生徒もあり、ゲーム制作を基本部分と発展的な部分とで分けた方がよい可能性が示唆された。

サンプルから拡張してゲームを制作したか、という質問には、25人の生徒がはいと回答した。表4の結果から、時間不足や、記述するプログラム量の問題、またそれぞれの文の意味を理解することに苦労した生徒なども多かったことがわかっており、授業時間内では教材の内容を制作することで精いっぱいであった生徒が多かったが、従来使用していた環境における問題点は解決することができた。

5.3 生徒の制作した作品

授業を通じて、ほとんどの生徒がサンプル通りにプログラムを制作することができた。何人かの生徒は学んだことを活かして自らコードを書き換え、独自にゲームを発展させた。また、授業時間外にもコード更新の形跡などが見られた。

生徒の制作した作品の中には、独自に改変をしていたものがいくつかあった。それらを次の手順で抽出した。

- すべての学生が作成した HTML/JavaScript のソースコード（これら1つ1つを「文書」とする）から識別子と予約語を「単語」として抜き出す。
- それぞれの単語について重みを $\text{tfidf}[13]$ を用いて算出し、文書ごとに単語の出現回数×その単語の重みからなるベクトルを作成する。
- 各文書のベクトルのなかで、互いに同一である（すなわち、サンプルを写しただけで改変がない）ものが複数検出されるので、それらのうちから同一の個数が多い順からいくつかをとりだし、これらを「サンプルベクトル群」とする。
- それぞれの文書のベクトルについて、サンプルベクトル群内の各ベクトルからのユークリッド距離を算出し、その最小値が大きい（すなわち、サンプルからの改変の度合いが大きい）順に並べる

この並び替えで上位に来た作品の一部を示す。これらの作品は、生徒が振る舞いを追加するため、独自の変数名や関数名を使用している。このような単語は全体として出現頻度が少ないため重みがあり、他のプログラムのベクトルとの距離が大きくなる傾向がある。

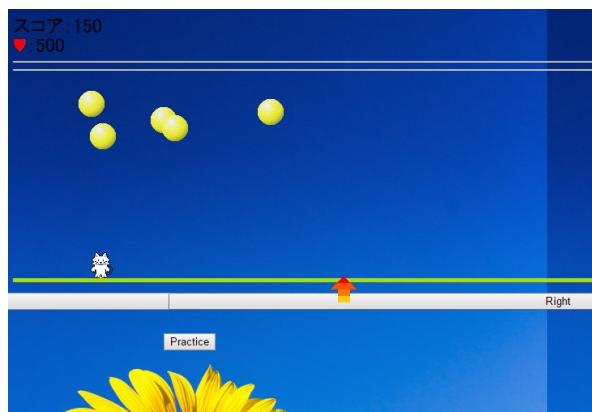


図 10 生徒が独自の改変を加えたゲームその 1

図 10 の作品は、背景に画像を貼られており、キャラクターにライフがつけられている。また、練習モードがあることや、しばらくプレーしているとボールが落ちてくるスピードが速くなることなど多くの点を改変している。



図 11 生徒が独自の改変を加えたゲームその 2

図 11 の作品は、ベストスコアが保存されるようになっている。

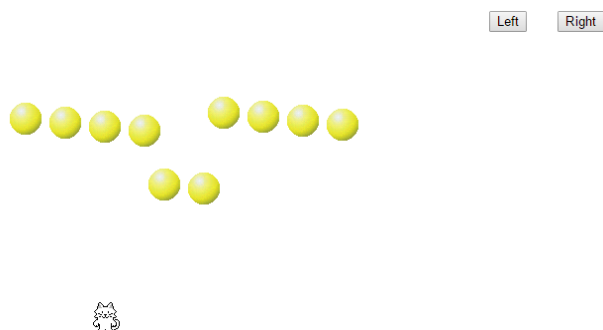


図 12 生徒が独自の改変を加えたゲームその 3

図 12 の作品は、ボールが真下だけでなく斜め方向にも落ちてくるようになっている。

6. 考察

JSLesson を授業で使うことで、一つの画面でコーディングと実行を行うことができ、API を使って簡単なゲームを作ることができた。生徒からの意見では、エラーが発生したとき、ポップアップでエラーの場所や内容が表示されることや、自動的に対応した記号が入力されることが特に使いやすかったと感じたことが分かった。

また、使いにくかった点は特になかったとした生徒が多く、それ以外に多い意見には、少しでも間違えると動かなくなることや、プログラムの量が多かったという点が挙げられた。少しでも間違えると動かないことは、ほぼすべてのプログラミング言語に共通することであり、これを経験することも、プログラミング学習の一つとして考えられる。プログラムの量が多いという意見からは、記述するプログラムをなるべく少なくするために用意した API を使用しても、初めて触れた生徒たちにとっては記述量が多いと感じたことが分かる。この点において、JavaScript の標準機能のみを使用することは、今回の授業よりも記述量が増えてしまうことになる。ゲーム制作で苦労した点においても、時間が足りなくなってしまったことや、プログラムの量が多かったこと、プログラムの文の意味が理解できなかったことなどが挙げられた。

ほとんどコンピュータに触れたことがない高校生にとっては、簡単なゲームとはいえ記述するコード量が多く、またエラーにも苦労したことがアンケートから読み取れた。したがって、教材を基本編と応用編に分ける工夫や、エラー内容をより詳細に教えることができるよう改良していくことが必要である。

JSLesson を使用した生徒の中には、ネット上などからコードを見つけてきて、それを試した生徒もいたようである。しかし、JSLesson では、HTML と JavaScript を分けて記述することを基本的な考えとしたため、HTML 内に JavaScript のコードを記述しても想定通りの動作をしないことがある。また、現段階では、CSS を設定させることができない。JSLesson でプログラミングに興味を持った生徒が、実際の JavaScript や CSS を動かすことができるよう考慮すべきであった。また、本研究では、柏陽高等学校のみで実践授業を行ったため、ユーザ ID のみでログインすることができた。しかし、今後多くの高校で使用されることを考え、アカウントの管理も求められると考えられる。

7. まとめ

高校の科目「情報の科学」で使用されることを想定した JSLesson を開発し、実際の授業で使用した。

高校生には、高校の科目「情報の科学」において、プログラミングに触れる機会がある。しかし、これまで使われていた教科書に載っているプログラムは、計算や、値の並

べ替えを行い、その結果の文字列をただ画面に表示するだけのものではなかった。このようなプログラムを扱うことで、プログラミングの基礎を学習することはできるかもしれないが、生徒が日常的に触れているものを扱った内容ではないことから、生徒の興味をそそる内容ではない。JSLessonでは、静的なページを表示させるHTMLと、HTML上の要素を操作するJavaScriptをそれぞれ教えることで、生徒が日常的に触れているWebアプリの仕組みを理解することができる。また、JavaScriptのプログラムでHTML上の要素を動かすために必要なAPIを用意することで、生徒が記述する必要のあるプログラムの量を減らす工夫をした。HTML上の要素を動かすことで、ゲーム制作ができる教材を準備し、多くの生徒がもう一度学びたいと思う授業を支援することができた。

参考文献

- 1) 中西渉, 辰己丈夫, 西田知博: PenFlowchart を用いた、フローチャートによるプログラミング学習の効果に対する評価, 情報処理学会論文誌教育とコンピュータ (TCE) 2015-1 巻 4 号, 75-82(2015).
- 2) Mohammed Al-Bow, Debra Austin, Jeffrey Edgington, Rafael Fajardo, Joshua Fishburn, Carlos Lara, Scott Leutenegger, and Susan Meyer: Using game creation for teaching computer programming to high school students and teachers, ACM SIGCSE Bull. 41, 3, 104-108(2009).
- 3) 水越敏行, 村井純, 生田孝至ら: 情報の科学, 日本文教, (2012).
- 4) 赤堀侃司, 永野和男, 東原義訓ら: 情報の科学, 東京書籍, (2012).
- 5) JSLesson. <http://eplang.jp/cho/jslesson/> (2016/02/05 アクセス).
- 6) 三菱総合研究所: 「平成 27 年度情報教育指導力向上支援事業」に係るプログラミング教育に関する実証校及び協力団体の募集のご案内について.
http://www.mri.co.jp/news/press/public_offering/recruit/018737.html (2016/01/20 アクセス).
- 7) 長島和平, 長慎也: Tonyu System 2 ～ ゲーム制作を通じたプログラミング学習に適したフレームワーク, 情報処理学会研究報告コンピュータと教育 2015-CE129 巻 2 号 (2015).
- 8) jsdo.it. <http://jsdo.it/> (2016/01/20 アクセス).
- 9) JSBin. <https://gratipay.com/jsbin/> (2016/01/20 アクセス).
- 10) JSFiddle. <https://jsfiddle.net/> (2016/01/20 アクセス).
- 11) Coding Tab. <http://codingtab.net/> (2016/01/20 アクセス).
- 12) 間辺広樹, 長島和平, 長慎也, 並木美太郎, 兼宗進: 高等学校における複数言語によるプログラミング教育の提案～情報システムの理解を目標としたドリトル, JavaScript, PHP の連携～, 情報処理学会研究報告コンピュータと教育 2016-CE133 巻 3 号(2016, 予定).
- 13) Ramos, Juan: Using tf-idf to determine word relevance in document queries., Proceedings of the first instructional conference on machine learning. (2003).