

研究ノート

楽しさで引っ張るプログラミング入門講座 — JavaScriptを使った 1日プログラミング教室用教材の開発と試用 —

室谷 心・矢野口 聡・浅見(林) 大輔

A Playful Learning Class for Programing:
Development and Trial Use of a Lesson Plan for a One-Day Programing Class with JavaScript

MUROYA Shin, YANOKUCHI Satoshi, and ASAMI Hayashi Daisuke

要 旨

まつもと広域ものづくりフェアにおいて、中高生向けの1日プログラミング教室を開催した。開催にあたって、中高生が興味を持ちさらに講座終了後持ち帰って続きができるような題材を、という要望があり、JavaScriptを使ったスマホゲーム作成の1日コース用の教材を開発した。ブロックプログラミングではない、言語のキー入力によるプログラミングであるが、コピー&ペーストを主体としたコーディング作業とすることで、1日でまとまった内容をこなすことが可能となった。受講者の興味を維持するために、教材の進みをスモールステップに分割し、ステップごとに動作確認を行いコーディング作業の結果を可視化した。ゲーム作成という楽しさに加えて、パラメータの変更などによってカスタマイズできる要素を多くすることによって、動作確認時の受講者の内容理解とプログラム作成の達成感が増加する傾向がうかがわれた。

キーワード

プログラミング教育 JavaScript 地域イベント

目 次

- I. はじめに
- II. 利用システム
- III. 授業計画と実施状況
- IV. 受講者の反応
- V. おわりに

文献

I. はじめに

次期小学校学習指導要領や高等学校の情報Iに取り入れるという中教審の答申以来、プログラミング教育に対する社会の注目度が増加している。本ノートでは、我々が行った中高校生向け1日プログラミング教室用教材の開発と、「まつもと広域ものづくりフェア」において実施した試用結果について報告する。

「まつもと広域ものづくりフェア」は、松本市を中心とした中信地区の市町村、広域連合、及び商工会議所などが主催する2日間のイベントであり、“ものづくり”をキーワードに地元企業や県内の大学、高専、高校などがブースや講座を開き、子供たちにもものづくりに親しんでもらおうというイベントである。われわれは2011年度から“ソフトウェアによるものづくり体験”をキーワードに“松本大学キッズプログラミング教室”を小学生向けに開催してきた¹⁾。

2016年には、①ものづくりフェアを製造業への就業意識を高める場としたいので、プログラミング教室を中高生向けにして欲しい、②受講者が帰宅後、家族や友達に見せて自慢でき、さらに続けて発展的な内容を自習できるような内容にして欲しい、という2つの要請を主催者側から受け、JavaScriptによる1日プログラミング教室を行うこととした(図1)。

ある程度まとまった内容のプログラム作成を目指し、定員10人で昼食をはさんで午前・午後トータル4時間の講座とした。講師は教員が務め、



図1. 受講者募集の案内

大学生の助手が3人つくことにした。

本講座は当日募集のオープンな講座であり、終了まで参加する義務のない催しものである。従って、興味が無くなれば受講者は容赦なく途中で帰ってしまう。最後まで参加してもらうためには、単調さを感じさせず、楽しさで引っ張る授業プランが必須である。

まず参加者を集めるために、中高生の興味を引く題材として、スマートフォン(以下スマホと略記)で動くゲームの作成を行うことにした。現状の普及率から考えて、スマホはiPhoneとAndroidのどちらでも作品を動かせる必要がある。そこでHTML5とJavaScriptを利用することにし、ゲームを簡単に作れるライブラリーである `enchant.js` を利用した²⁾。

受講者の興味を最後まで維持するために、以下のような工夫を行った。まずプログラムの作成過程を細かく分け(スモールステップ)、各段階でビジュアルな動作確認をとることにした(可視化)。これにより受講者は、コーディング作業の単調さを解消し、さらに直前の作業で追加修正した部分のコードの意味を明確に理解することが期待される。コーディング作業では、キー入力に習熟していない受講者を想定し、コピー&ペーストで進めていく形をとった。さらに、パラメータや引数を利用して“イージーオーダー形式”とでも呼ぶべき簡単にカスタマイズできる余地を各所に用意し、変更しながら動作確認を行い、受講者の好みでオリジナリティーを出せるようにした。ここでの狙いは作成コードの動作による理解とそれを検証するカスタマイズであり、“わかる”や“できる”という楽しさで学習の継続を期待し、楽しみながら学ぶ Playful Learning を目指した³⁾。

II. 利用システム

学内LANと机に電源のある教室を利用し、パ

ソコン(Sony VAIO core i5 4200 1.60GHz 4GB メモリー、タッチパネル画面)を受講者人数分用意した。OSは64Bit Windows8.1であった。USBスティックメモリーに必要なライブラリーや画像ファイル、音データ、テンプレートファイルに加えて、完成プログラム例のファイルを用意して配布した。エディターはTeraPadを使い、JavaScriptの実行ブラウザはGoogle Chromeを利用した。Google Chromeにはデベロッパーツールと呼ばれる開発者モードがあり、JavaScriptの基本的な文法エラーに関しては、メッセージを出することができる。受講者が家に帰って続きができるためには、フリーソフトだけで作業が閉じることが肝要である。

JavaScriptなので、作ったプログラムをスマホで動かすにはHTMLサーバに上げる必要がある。講座内で使うだけならば、学内限定の臨時サーバで十分であるが、帰宅後家族や友達に見せるためには、公開サーバが必要となるDropboxなどのクラウドの利用やレンタルwebサーバの利用を検討したが、html公開の可否やライブラリー利用の自由度、アップローダの有無などの点で適当なものが見つからなかった。また、同一ドメインでのアダルトサイトとの共存も、中高生向けの教育講座としては好ましいものではない。そこで今回は自前のサーバを利用することとした。

Linux + apacheで動作するHTMLサーバに専用のディレクトリーを用意し、受講者ごとのサブディレクトリーにライブラリーや必要な画像を用意し、データ配布用USBスティックメモリーと同じファイル構成にした。パソコンからサーバへのファイル転送には、Perl CGIによるアップローダを用意した。

受講者が作ったプログラムにスマホからアクセスするURLは、無料のサイトを利用してQRコードを作成し、帰宅後家族や友達に見せられるように各自で写真に撮ることにした。

Ⅲ. 授業計画と実施状況

プログラム作成とデバッグ作業はエディターTeraPadとブラウザGoogle Chromeを使って、基本的にローカル環境で行った(図2)。

Google Chromeのデベロッパーツールにはスマホ画面のエミュレーションモードがあり、タッチパネル対応パソコンを使ったので、スマホのタッチ画面での操作性も確認することができた。

プログラム作成のスタートファイルとして、webページの基本設定に加えenchant.js関連ライブラリーの読み込み設定、加速度センサーの準備、game.rootScene設定のしてあるファイルをテンプレートとして、利用することにした(図3、4、5)。

ここで単にJavaScriptの文法を解説するのではなく、図5の末尾の「試してみよう」にあるように、コメントアウトしたり引数を変えてみたりして実行し、「エディターでのエディット⇒保存⇒ブラウザでの再読み込み」という一連の作業に慣れるとともに、コードの内容理解と受講者ごとのカスタマイズを行った(イージーオーダー形式)。受講者の反応を見ていると、キャラクターや背景色など簡単なものではあってもカスタマイズできることは、プログラム作成作業の充足感を高める重要なポイントであった。

講座のこれ以降の課程では、コードの意味ごとのブロックを挿入して、htmlファイルを順次更新していくように教材をデザインした。タイトルプログラムではなく、エディターを使うスタイルをとったのは、もともとは“言語の学習”の教育効果を狙って、受講者のキー入力を想定したものであった。しかし1日目のクラスの冒頭で、試験的に受講者にキー入力をさせてみたところ、キー入力が遅いばかりでなく、英語理解の不足からか、語単位でなく文字単位での入力確認作業となって時間がかかり、学習意欲の明らかな低下が見られた。そこで、今回はサンプルファ

イルの該当部分をコピー&ペーストで挿入して進めることにした。

サンプルファイルでは、解説用に更新差分部分をまとめてマークアップしてあり(図6)、エディター上で行番号を指示することによって、受講者は混乱なく該当部分を見つけることができ、

スムーズにコピー&ペーストを行うことができた。マウスを使ったコピー&ペースト作業には、受講者は良く慣れているようであった。

講座でのプログラム作成のステップは、以下のようなものであった。

1. テンプレートの利用

Webページの基本設定とJavaScriptの利用
enchant.jsの読み込み部分

加速度センサーの準備

キャラクターの画像の読み込みと生成

ステージとアニメーションの設定

各オブジェクトのレイヤーの準備

エディターによる修正と保存

ブラウザによる実行や再読み込み

2. キャラクターの動作

タッチパネルによる動き

x, yの座標と移動

3. 発射弾の作成

Spriteクラスから弾クラスの生成

図2. プログラム作成作業の解説スライド

図3. 配布したテンプレートの先頭部分

図4. 配布したテンプレートの中盤部分

図5. 配布したテンプレートの終り部分

図6. サンプルファイルの差分部分の例

- 弾の属性(絵や動き)、発射タイミングの設定
弾を弾レイヤーの子プロセスとして生成
4. 敵登場
Sprite クラスから敵クラスの生成
敵の属性や機能の定義
敵を敵レイヤーの子プロセスとして生成
ローカル変数とグローバル変数の違い
 5. 敵の攻撃
Sprite クラスから敵弾クラスの生成
敵弾の動きや発射タイミングの設定
敵弾を敵弾レイヤーの子プロセスとして生成
子プロセスの消去の仕方
 6. 敵に命中
弾と敵の接触判定
親プロセスに対して子プロセスの排除命令
 7. 効果音
敵の爆発音を入れる。サウンドデータの扱い
 8. 新しいクラスを定義して使う
弾と敵弾の定義から共通部分を括りだし、
Sprite クラスから新しいクラスを定義する
新しく定義したクラスを使って弾と敵弾の
定義を簡単に書き直す
 9. 画面タップに反応
一定間隔で出ていた発射弾を、画面をタップ
した時に発射するように変更する
 10. ライフを有効にする
自キャラと敵弾の接触判定
ライフ機能を起動し、ゲームの終了を付ける
 11. 加速度センサーによるコントロール
自キャラクターを加速度センサーでコント
ロールできるようにする
 12. 発射音
弾発射時に発射音が出るようにする
 13. 新しい敵キャラの作成
別な動きをする新しい敵キャラ敵2を定義する
 14. 新しい敵クラスの定義
新旧の敵の定義を比べ、新しい敵クラスを定
義して、敵と敵2の定義を簡単にする

15. 新しい敵クラスの利用

新しい敵クラスをつかって、敵3を定義する

16. 被弾効果の追加

s 自キャラが被弾した際の効果(短時間の爆
発の絵と音)を追加する。

各ステップとも10分から15分程度の作業で、
コピー&ペースト作業に続けて、動作確認とパ
ラメータの変更や読み込みファイルの差替えな
どによる、受講者ごとのカスタマイズを行った。
各プロセスをスモールステップにとどめ⁴⁾、コー
ディングの効果を可視化して確認した。プログ
ラムの作成ステップとしては冗長で、全体とし
て完成形を目指した一本道になっていないのは、
これも受講者を飽きさせないために意図したも
のである^{5), 6)}。

実際の講座では、5か6の作業終了後にお昼休
みをとった。また、11の後で、ファイルのアップ
ロードとURLのQRコード作成を行い、各自の
スマホでの動作確認を行った。14、15のステッ
プと16のステップはどちらか一方をこなすのが
時間的に精一杯であった。

enchant.js はもともとゲーム作成のためのラ
イブラリーであり、Game クラスの属性として、
fps やスタート画面、得点、ライフなどが用意し
てあり、ゲームプログラムに特有の細々した設
定をいちいち自分で定義する必要はない。

JavaScript はモダンな言語であり、クラスか
らインスタンスを作ったり、クラスを継承する
といったオブジェクト指向プログラミングを行う。
プログラミングの教科書では色々な属性や機能
を持ったオブジェクトの説明に、動物や機械な
ど実世界にある物体を例に使うことが多いが^{7), 8)}、
ゲームに出てくるロケットや敵UFOといった
オブジェクトはまさしく実世界の物体のシンボ
ルであり、クラスの定義としていろいろな属性
や機能を持つことは改めて強調するまでもない
自然な話である。また、プログラムを発展させ

ていく際にも、新しい属性や機能の追加という形でのオブジェクトの発展もコードの自然な改良として導入でき、ゲーム作成はオブジェクト指向プログラミングの自然な導入となる。

本講座の中では、弾と敵キャラに関して、類似オブジェクトの定義をわざと別々に行ってから(ステップ3、4)、その冗長性を解消する形で、クラスを定義することの意義を説明した(ステップ8)。ステップ13での新しい敵キャラの定義でも、新たな定義をまじめに行っていくが、作業を行っているうちにすでに在る敵の定義と同じ命令がいくつも出てくることに気づく受講者が出てくる。新しい敵の定義の際に、弾の時にステップ8で行ったのと同様に、すでに存在する敵とまとめて“敵クラス”を作りそれを継承することによって、効率良く定義できることに気づく受講者もいた。

IV. 受講者の反応

今回のヤングプログラミング教室は中高生を対象とした当日募集のオープンな講座であり、参加者についてはこちらではコントロールできない。結果として2日間とも高校生の参加者はなく、全員が中学生で、さらに約半分が1年生であった。男女比は、ほぼ1/3が女子であった。この参加者の学年構成のために、英語に対する素養が不十分であったのか、プログラム入力の際には語単位ではなく文字単位での作業をしていたようで、キー入力によるコード作成を試みた際にはコード入力とデバッグ作業に予想外に時間をとられた。実際にキー入力を試みたのは、ステップ2の冒頭で図7中の四角で囲まれた4行であったが、1日目のクラス全員がこの4行を正しく入れ終わるにはかなり(筆者の体感で20分程度)の時間がかかり、また、何よりも受講者がうんざりしている雰囲気がありありと伝わってきた。

キー入力方式からコピー&ペースト式の作業

に切り替えて以降は、特に問題なく計画通り授業を進めることができ、講座開始後2時間で § III で説明した授業計画のステップ5まで終了した。接触判定(ステップ6)の前で1時間の昼食をとり、午後の2時間で、受講者全員がステップ13までゲームを完成させてサーバにアップロードし、ダウンロード用URLのQRコードの作成まで終了することができた。追加課題としては、1日目はステップ14、15の新しい敵の作成を行い、2日目は被弾効果の追加を行った。

§ III の授業計画でも述べたように、5分程度の作業でのスモールステップでのプログラム作成とその直後での動作確認を心掛けたので、受講者は“自分がプログラムを作成している”という実感をしっかりと持つことができ、緊張感を失うことなく1日の講座を続けることができたようである。とくに、パラメータを自分で変更し人と違う見かけや動作を実現できることは重要で、ほとんどの受講者が、ゲームとしての統一感や完成度への気配りを忘れて、背景色やキャラクターの選択や動き、効果音などを自由に変えることに夢中になって作業していた。論理的な表現であるプログラミング言語を入力した結果が、画面上の見かけや動作としてわかりやすい形で直ちに表れることは、受講者の学習意欲の維持に大きな効果があることを実感させられた。

終了時にとったアンケートの結果が表1である。回収数は2日間の総計17枚で、うちの1枚は女子生徒の隣の席で授業参加していた父親のものであった。アンケートの結果は総じて好評であり、難易度も分量も良い評価であった。難易度につ

```

57  }); //player.addEventListener終了!
58
59
60  game.rootScene.addEventListener(Event.TOUCH_START, function(e) { //タッチした時!
61    player.x += 0.1*(e.x-player.x); //画面タッチに反応する時
62    player.y += 0.1*(e.y-player.y);
63  }); //game.rootScene.addEventListener終了!
64
65  game.rootScene.addChild(enemyBulletLayer);
66  game.rootScene.addChild(enemyLayer);
67  game.rootScene.addChild(itemLayer);
68  game.rootScene.addChild(player);
69

```

図7. 試験的にキー入力を行った部分 (図中四角で囲んだ4行)

いて「難しすぎる」という唯一の回答は女子生徒の父親のものであった。

「家で続きをやるような題材を」というのが主催者側からの要望であったが、アンケートを見る限り、受講者の意欲は十分に見られる。ただ、実際に家でプログラム作成の続きを行ったかどうかは、今回は追跡調査を行ってはいない。

V. おわりに

中高生を対象に、JavaScriptでenchant.jsを使ってスマホで動くゲームを作る1日プログラミング教室用教材の開発を行い、実際に開催した教室で試用を行った。受講者全員がゲームを完成し自分のスマホで動かすところまで完了したことや、アンケート結果から見て、難易度も分量も1日イベント(4時間程度)の講座として過不足ないものであったと考えられる。学内でのスマホの扱いはいろいろであろうが、本講座の教材は中学の技術家庭や高校の情報の授業でも4時間程度を使ったプログラミング教育に使えるのではないかと考えている。

タイトルではなくコード入力を行う形式をとったが、キー入力を断念してコピー&ペーストでプログラムの作成を進めたので、文法や予約語を覚えるような言語教育という面では教育効果は期待できないと考えている。キー入力による

プログラム作成に問題があったのは、“中高生”ということで受講者を募集したが、実際には中学生、それも1年生が多く、キー入力に慣れていなかったためだと考えられる。実際、著者の一人(浅見)は同様の内容を本務校商業科3年生対象の選択科目で授業に使ってみたが、キー入力による問題は特にみられなかった⁹⁾。これは中学入学から高校3年までの6年間に身に付けたキー入力や英単語のリテラシーの成果ではないかと我々は考えている。

ゲームという画像の動きや音を中心とした題材を使うことによって、JavaScriptでのオブジェクト指向プログラミングにおける基本的概念の自然な導入が可能であった。また、Scratchのような子ども用のプログラミング環境との接続もシームレスにできる可能性が高い⁹⁾。

コピー&ペーストでのコーディング作業ではあったが、パラメータや引数など簡単に変更できる部分を多く用意し、自由にカスタマイズできるように(イージーオーダー形式)することによって、動作確認時の受講者の反応が活発になり、達成感が高まった様子が見られた。カスタマイズ作業によって、作成コードに対する理解も深まったのではないかと期待している。中学や高校のプログラミングの授業でも、キー入力に躓く可能性は大きいと考えられるので、本教材同様にコピー&ペーストで作業を進め、イージー

表1 アンケート結果

Q1. 楽しかったですか?	5. 楽しかった	4. まあまあ楽しかった	3. まあまあだった	2. あんまし	1. 期待はずれ
	17	0	0	0	0
Q2. 家で続きを?	5. 絶対やる	4. できればやる	3. わからない	2. あんまし	1. やらない
	5	12	0	0	0
Q3. 説明は	5. グッド	4. まあまあグッド	3. こんなもん	2. あんまし	1. 判らない
	13	3	1	0	0
Q4. 内容は?	5. ちょうど良い	4. まあまあ	3. 少ない	2. 多すぎ	1. 物足りない
	6	6	4	0	0
Q5. 難易度は?	5. ちょうど良い	4. まあまあ良い	3. こんなもん	2. 難しすぎ	1. 物足りない
	6	6	4	1	0

オーダー形式のカスタマイズで動作確認を多くする(スモールステップと可視化)ことによって、キー入力作業の省略を補完し内容理解を深めるという方法は、学校でも授業中に生徒を飽きさせないために重要な工夫になるのではないかと期待している。

謝辞

本研究は松本大学COC「地(知)の拠点」活動の補助を受けた。キッズプログラミング教室(ヤングプログラミング教室)の開催にあたって便宜をはかってくださった「まつもと広域ものづくりフェア」実行委員会、並びに松本大学情報センターに感謝する。

文献

- 1) 室谷心: 地域イベントで子供に教える機会を利用した情報科教職課程学生の教育, 日本情報科教育学会誌, Vol.8, No.1, pp.61-69 (2015).
- 2) 布留川英一, 伏見遼平, 田中諒: “ゼロからはじめるenchant.js入門”ASCII, 東京 (2012).
- 3) Michel Resnick: “Edutainment? No Thanks. I Prefer Playful Learning”, <https://llk.media.mit.edu/papers/edutainment.pdf> (2017年6月13日確認).
- 4) 向後千春: “上手な教え方の教科書”, 技術評論社 (2015).
- 5) ポール・クロチア: “パソコン・インストラクター完全マニュアル”, 海文堂 (1997).
- 6) 島宗理: “インストラクショナルデザイン”, 米田出版 (2004).
- 7) 青木淳, 浅岡浩子, 滝本依里: “Smalltalkで学ぶオブジェクト指向プログラミングの本質”. 日経BP (2008).
- 8) 小森裕介: “なぜあなたはJavaでオブジェクト指向開発ができないのか”, 技術評論社 (2005).
- 9) 浅見大輔, 室谷心: “ScratchとJavaScriptによるプログラミング教育の実践”, 情報科教育学会全国大会講演論文集 (2017), pp113-114 (2017).