

高等学校共通教科情報における 「ドリトル」を利用した授業の実践報告

佐々木 寛^{1,a)} 島袋 舞子^{2,b)} 兼宗 進^{2,c)}

概要：高等学校「情報科」の授業において、JavaScript を用いた実習をおこなってきた。しかし、JavaScript による実習は初学者にとって簡単なものではなく、関心・意欲・態度の個人差が大きい。そこで、教育用プログラミング言語「ドリトル」を使用し、実習をおこなった。今回実施した 12 回の授業の内容と生徒の制作課題を報告し、情報の授業におけるプログラミング教育の効果と今後の実施方法を検討したい。

キーワード：プログラミング教育, ドリトル, 情報教育

A report of using Dolittle in high school informatics classes.

SASAKI HIROSHI^{1,a)} SHIMABUKU MAIKO^{2,b)} KANEMUNE SUSUMU^{2,c)}

Abstract: We had used JavaScript language in informatics classes in high school. Because JavaScript was not easy for our students, some of them couldn't get interested in programming. So this year, we used Dolittle language. In this paper, we will report the contents of those classes and student's works.

Keywords: Programming education, Dolittle, Informatics education

1. はじめに

教科「情報」が開設されてから 10 年が経過し、情報教育は高校現場に根付いてきた。しかし、情報の教育内容は多岐にわたっており、その指導方法に先生方が頭を悩ませている現状に変わりはない。北海道の情報科教諭を対象としたアンケートにも多数の質問や、悩みの声が寄せられ、特に「他校ではどのように授業を行っているのか?」という声は多い。私自身も情報教育に携わるようになり 6 年目になるが、その間、「生徒の実態にあった効果的な情報教育とは何か」を考えながら試行錯誤を繰り返してきた。

昨年度までの授業では JavaScript を扱ってきたが、生徒にある程度の知的好奇心と理解力がないと実施できない特殊な内容であり、更にプログラミングを身近に感じてもらうためにはもっと生徒の心を捉え、印象に残る題材を用意しなければとの思いを感じていた。

そのような中で出会ったのがドリトル [1][2][3][4] である。ドリトルを授業に取り入れることによりプログラミングの授業を理想と考えていた形に近づけることができた。ドリトルは多様な生徒に受け入れられる魅力とわかりやすさがある。今年度は昨年度までは導入が難しかった定時制の授業にもプログラミング教育を取り入れることができ、その効果を感じている。

本校は一昨年に創立 110 周年を迎えた道内で 3 番目に古い歴史を持つ学校である。北海道教育委員会が実施する「地域医療を支える人づくりプロジェクト」の指定校（道内で指定校は 9 校）でもあり、9 割以上の生徒が大学へ進学する。2 年次から文系と理系・医進類型コース（医進類型選択者は理系クラスに含まれる）に分かれてクラス編成

¹ 北海道小樽潮陵高等学校
Otaru Choryo High School, Otaru, Hokkaido, 047-0002, Japan

² 大阪電気通信大学
Osaka Electro-Communication University, Shijonawate, Osaka 575-0063, Japan

a) hirochi@hokkaido-c.ed.jp

b) shimabuku.m@gmail.com

c) kanemune@acm.org

されるため、より各コースの生徒の特性に合わせた共通教科「情報」の授業を展開できるよう、文系クラスでは「社会と情報」、理系クラスでは「情報の科学」を2単位開設している。また、3年次には選択科目として学校設定科目で「情報活用」を2単位開設しており、そのすべての科目でドリトルによるプログラミング教育を取り入れることができた。

本稿では昨年までの JavaScript の授業の概要を報告した後、今年度実施したドリトルの授業を分析し、その学習効果と次年度以降の実施に向けた授業の改善案を報告する。

2. 生徒のプログラミングに関する実態

授業開始前に事前アンケートを実施し、プログラミングに対する生徒の現状を調査した。結果を表1に示す。Q1ではプログラミング経験の有無を聞いた所、経験があると答えた生徒は1,2度だけの経験を含めても272名中20名(7.3%)のみで、252名(92.6%)の生徒はプログラミングの経験がまったくなかった。また、数少ない経験者もほとんどが学習途中で挫折してしまっている。Q2ではプログラミングについて知っているかどうかを聞いたが、どういものかわからない生徒が186名おり、そのうち33名はプログラミングという言葉すら聞いたことがないと答えている。今回のアンケートで、普段自分たちが利用しているソフトウェアの仕組みをまったくイメージできないまま使用している様子が浮き彫りになり、現在の学校現場ではなかなかプログラミングに触れる機会を得られない現状がうかがえる。しかし、このような状況の中、Q3の質問では162名もの生徒が多少なりともプログラミングに興味があると答えている。

表1 事前アンケート (集計人数 272名)

Q1: プログラミングの経験がありますか?			
回答	人数	割合	
経験があり、得意である	0	0.0%	7.3%
挑戦したことがあるが、挫折した	8	2.9%	
1,2度程度なら経験したことがある	12	4.4%	
まったくない	252	92.6%	92.6%

Q2: プログラミングという言葉を知っていますか?			
回答	人数	割合	
知っている	9	3.3%	31.6%
なんとなく知っている	77	28.3%	
どういものかわからない	153	56.3%	68.4%
その言葉自体、聞いたことがない	33	12.1%	

Q3: プログラミングに興味がありますか?			
回答	人数	割合	
とても興味がある	21	7.7%	59.6%
結構興味あったが無理と思っていた	34	12.5%	
ちょっとだけ興味がある	107	39.3%	
まったく興味がない	110	40.4%	40.4%

3. 昨年度までのプログラミング指導

本校では実践的なプログラミングを体験させるために JavaScript による実習を取り入れている。JavaScript は、テキストエディタとブラウザさえあればプログラミングを体験させることが可能であるため、生徒が簡単に家庭でも同じ環境で学習できるという利点がある。JavaScript による実習内容を表2に示す。

JavaScript での実習は初学者にとって簡単なものではないが、生徒は初めてのプログラミング体験にそれなりの好反応は示していた。しかし、生徒にある程度の知的好奇心と理解力がないと実施できない特殊な内容であり、生徒の関心・意欲・態度にも個人差が大きかった。

また、最後に授業で学んだ基礎知識を応用して「じゃんけんゲーム作り」に取り組むことにより、プログラミングを学んだ成果としてある程度の形にすることはできたが、更にプログラミングを身近に感じてもらうためにはもっと生徒の心を捉え、印象に残る題材を用意する必要があった。

表2 JavaScript を用いた授業内容

回	テーマ	内容
1	出力、コメント	document.write, alert による出力
2	変数、演算子	変数の宣言、代入、計算
3	基本制御 (分岐)	if 文で条件による分岐
4	基本制御 (繰り返し)	for 文で繰り返し処理
5	乱数	Math.random で乱数を生成
6	配列、関数	配列、関数の基礎
7	じゃんけんゲーム 1	じゃんけんゲームを作成
8	じゃんけんゲーム 2	じゃんけんゲームの改良

4. ドリトルを導入する狙い

ドリトルを授業に導入する最大の利点は言語そのものの学習を短時間で終わらせることができるため、プログラミングを活用した応用的な学習までをスピーディーに展開することができることにある。ドリトルは日本語でプログラムすることができるために英語の壁がなく、一般の汎用言語に比べると初学者の負担はかなり軽い。そして、ボタンなどのオブジェクトにメソッドを定義する形で現在主流のオブジェクト指向やイベント型のプログラミング感覚を身につけることができる。言語処理系がインタプリタ方式で、1行ずつ動作を確認しながらプログラムを作っていくことも、生徒の反応がその都度よくわかり、エラーも見つけやすいため授業には適していると感じられた。

また、ドリトルを授業計画に組み込む際の柔軟性の高さも大きな魅力の一つである。ドリトルはその目的に応じて授業回数を調整しながら実施可能であり、時数に対する柔軟な効果が期待できる。プログラミングを体験することにより、ソフトウェアが動く仕組みを実感させ、興味を持た

せるという目的であれば、1回だけの授業でも十分実現でき、2、3回の時間があれば少しレベルの高いゲーム作りに取り組むことも可能である。体験させるだけではなく、プログラミングの基礎から応用までしっかり学習させたい場合でも5から15回（当然、それ以上も可能）の授業での実施に対応できるだけの多彩な題材が用意されており、各校の実情に合わせて自在に授業をデザインすることができる。今年度、本校ではプログラミングを体験することで、ソフトウェアの仕組みを学ぶとともに、最終的には生徒自身がオリジナルのプログラムを作成できる力を養成する目的があったので、相応の技術を習得するために、16回をかけてドリトルの基礎から応用までを段階的に指導した。具体的な内容は後述する。

5. 本年度の授業

今年度、本校では12回の授業でタートルグラフィックスを利用したアニメーションやゲーム作りの方法を指導し、4回の授業でオリジナルプログラムの課題制作を実施した。各回の授業テーマとその内容を表3に示す。

初回はプログラミング初体験の生徒に簡単なゲーム作りを体験させることにより、ドリトル（プログラミング）の魅力を感じてもらい、今後の学習意欲を高めるねらいで実施し、2回目から7回目までの6回の授業で基礎技術を指導、8回目から11回目までの授業で実践的なプログラミングを体験させるためにゲームプログラム作りに挑戦し、12回目の授業ではオリジナルプログラムを作る際のヒントをつかんでもらうねらいで実施した。各回の授業で生徒に伝えたいテーマを吟味し、扱う内容の関連性に配慮しながら授業を構成し、毎回生徒が新たな発見に驚き、感動する学習ができるよう題材の選定を工夫した。

授業はコンピュータ教室にて、一人1台のコンピュータと授業プリントを用いて行う（図1）。第7回の授業で用いた授業プリントを図2に示す。



図1 高校での授業風景

各回の授業の理解度を確認するため、次の授業の導入部で5分程度、筆記の小テスト（配点5点）を実施し、生徒の学習評価にも利用した。

表3 ドリトルを用いた授業内容

回	テーマ	内容
1	はじめてのプログラミング	「宝物拾いゲーム」を作成
2	繰り返し処理 図形オブジェクト タイマーオブジェクト	三角形等の図形を作成し、 タイマーオブジェクトで回転 させる
3	命令定義 パラメータの受け渡し	オブジェクトに命令を定義し パラメータを受け渡す
4	衝突定義 条件分岐 論理演算 オブジェクトの複製	衝突を定義し壁にぶつかった ときに方向転換する。条件分岐 で動作を場合分けする。
5	色オブジェクト	新しい色を作る
6	ラベルオブジェクト 変数 配列	文字を画面に表示する。 複数のオブジェクトを管理する
7	フィールドオブジェクト 乱数	経過時間を表示させる。 タートルをランダムに配置する
8	ピンポンゲーム 1	習った技術を総動員してゲーム を作る
9	ピンポンゲーム 2	
10	シューティングゲーム 1	
11	シューティングゲーム 2	
12	アニメーション	帆掛け船をタイマーで走らせる
13	課題制作 1	アニメーションやゲーム等、
14	課題制作 2	生徒が自分でテーマを決めて
15	課題制作 3	オリジナルプログラムを作成
16	課題制作 4、相互評価	

授業の題材は主にドリトルのテキストである「ドリトルで学ぶプログラミング」[5]から選定しているが、授業時間の関係で本校実施用に授業プリントにまとめなおして実施している。また、筑波大学久野靖先生の「5時間でプログラミングを学ぶ（ドリトル編）」[6]、静岡大学紅林秀治先生の「ドリトルテキスト（ドリトルを覚えよう）」[7]に魅力的な題材が多く、参考にさせていただいた。

6. 各回の授業内容と学習効果の分析

6.1 第1回「はじめてのプログラミング」

初回の授業では、プログラミングを体験することにより普段自分たちが利用しているコンピュータシステムやソフトウェアがどのような仕組みでできているのかを知ること。そして、プログラミングを学ぶ意義を理解し、今後の学習意欲を高める目的で実施した。題材として「1時間で学ぶソフトウェアの仕組み」[8]の宝物拾いゲームを選んだ。

授業後のアンケートでは、97.8%もの生徒が楽しかったと回答し、自由記述では初めてプログラムを書いた感動の声や自分の指示通りにプログラムが動いた喜びの声が大多数を占めた。また、プログラミングに興味を持ったと答えた生徒が94.0%、今後もプログラミングを学びたいと興味を示した生徒が93.3%おり、プログラミングの導入として、ドリトルの授業利用は効果的であった。

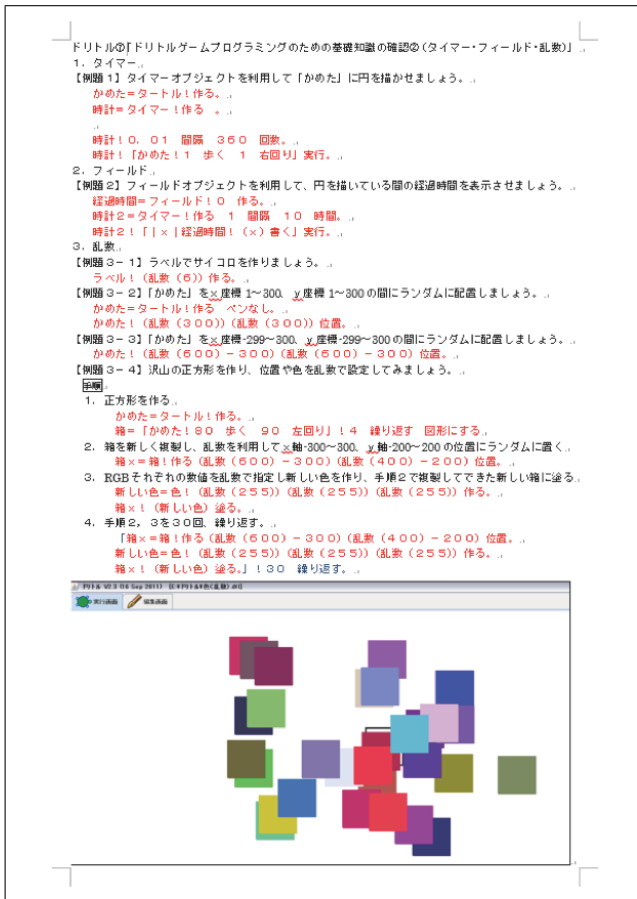


図 2 第 7 回の授業で用いたプリント

表 4 1 回目の授業後アンケート (集計人数 268 名)

Q1: プログラミング (ドリトル) を体験した感想				
回答	人数	割合		
とても楽しかった	175	65.3%	97.8%	
少し楽しかった	87	32.5%		
あまり楽しくなかった	6	2.2%		2.2%
まったく楽しくなかった	0	0.0%		

Q2: プログラミングに興味を持てたか				
回答	人数	割合		
とても興味を持った	87	32.5%	94.0%	
少し興味を持った	165	61.6%		
あまり興味を持てなかった	16	6.0%		6.0%
まったく興味を持てなかった	0	0.0%		

Q3: 今後もプログラミングを学びたいか				
回答	人数	割合		
もっと深く学びたい	108	40.3%	93.3%	
少し学びたい	142	53.0%		
あまり学びたくない	17	6.3%		6.7%
まったく学びたくない	1	0.4%		

6.2 第 2 回「繰り返し・図形・タイマー」

ドリトルプログラミングの基礎を学ぶことを目的とし、よいプログラムを作るための繰り返し処理、ドリトルプログラミングに欠かせない図形オブジェクトの扱い、オブジェクトに動きを加えるためのタイマーを重点的に学習

した。

小テストの平均点は 3.0 点で、ドリトルと小テストの出題形式に慣れていないこともあり、「」の使い方やオブジェクトの名前の付け方の理解が不十分でケアレスミスが多かった。タイマーに関する設問の正答率は低く、1 度の学習ではタイマーの技術を習得するまでには至らないことがわかった。繰り返し処理の設問は正答率は低くなかったが、課題制作の際には必要な場面で利用できていない生徒も多く、様々な場面に応用して利用できる機会を増やす必要性がある。

6.3 第 3 回「命令の定義」

ドリトルではオブジェクトごとにあらかじめ実行できる命令が用意されており、新たに自分で追加 (定義) することもできる。今回は、より高度なプログラムを作るために必要となる命令の定義やパラメータの受け渡し方法を学習した。

まず、タートルに正方形を作るための命令を定義する方法を学習した後、パラメータの受け渡しができるように改良し、大きさの違う正方形を 3 個描いた。次に、別の題材で長方形を作る命令を定義し最初の内容を復習した後、タイマーの実行間隔と実行回数を変更する方法を学習した。

小テストの平均点は 3.1 点で、出題形式に慣れ前回よりも平均点はアップしているものの、命令の定義とタイマーの利用は生徒にとって理解しにくい内容であることを表している。タイマーに関しては課題制作の際に、生徒からの質問が最も多かったものの一つであり、最初のうちは正確に記述し、場面に応じて自在に利用できる生徒が少なかったが実習を重ねるうちに習得していった。命令の定義は課題制作で利用できている生徒がほとんどおらず、テストの結果以上に生徒の理解不足が端的に表れており、初学者にとって応用が難しい内容であることがわかった。しかし、効率よいプログラムを作成するために必須の技術で習得すべき内容であり、次年度以降実施の際には効果的な応用例を数多く示す等、自分で活用できる力を養うための工夫が必要である。

6.4 第 4 回「衝突定義・条件分岐」

衝突の定義と条件分岐の技術を習得することを目的として実施した。

前回作った長方形のオブジェクトを壁に見立てて、タートルを壁に向かって歩かせ、タートルが壁に衝突した時の動作を定義し下方向や来た方向に方向転換させた。更に壁を複製して色を変え、赤、青、黄の 3 つの壁を用意し、「青色の壁に衝突した時だけ方向転換」、「黄色の壁か青色の壁に衝突した場合、方向転換」等、条件分岐と論理演算を組み合わせて条件に応じてタートルを壁の間で行き来させる命令に発展させた。

小テストの平均点は2.5点で、ドリトルの授業範囲では最も低く、初学者である生徒にとって、条件分岐の理解と活用は非常に難しく大変苦戦している様子だった。説明を聞いている分にはある程度理解できているつもりでも、いざ活用するとなると必要な場面で正確に使いこなすことができる生徒は少なく、課題制作の際、生徒がつまずき、作業が滞る最も大きな要因となった。今回は衝突定義と条件分岐を一つの命令の中で説明したため、より難易度をあげてしまい、更に論理演算との組み合わせまでの過程を急ぎ過ぎてしまった。今回の内容はゲームプログラムを作る際、一番の鍵となる大変重要な技術なので、もっと時間をかけて段階的に指導する必要がある。

6.5 第5回「色オブジェクト」

ドリトルにはあらかじめ基本8色（黒・赤・緑・青・黄色・紫・水色白）の色オブジェクトが用意されている。これ以外の色を利用する場合はこれらの色を混ぜたり、赤緑青（RGB）の数値を指定して新しい色オブジェクトを作る必要がある。今回はアニメーション等を作成する際に必要な色オブジェクトを自分で用意できる技術を身につけるとともに、教科書で取り上げられている光の三原色等の理解を深めることを目的に実施した。

生徒は色オブジェクトに関する内容をよく理解し、課題制作の際には各場面に合わせた効果的な色を作り出していた。

6.6 第6回「ラベル・変数・配列」

第6回と第7回の2回の授業でゲーム等の応用プログラムを作る際に必要でまだ説明できていない技術をまとめて実習した。今回はプログラミングに必須の知識である画面出力、変数、配列の理解を目的として実施した。

配列はシューティングゲームやアクションゲームで敵を管理する際に必須の技術であるが、ゲームで利用する際には命令が複雑で量も多くなり、場面に応じて自在に使いこなすためには相当な力量が必要となる。課題制作の際には理解不十分な状態で利用しようとしてエラーで前へ進めなくなり質問を受ける場面が多かった。

6.7 第7回「フィールド・乱数」

ゲーム作成時、得点や経過時間等を表示するのに利用できるフィールドオブジェクトと敵やキャラクター等をランダムに配置することができる乱数を効果的に活用できる力を身につけることを目的に実施した。

まずは、タイマーの理解を深めるためにタートルが1歩、歩いては、1度回転する動作を0.01秒間隔で360回繰り返して実行するタイマー命令で円を描いた。これを応用して生徒は、課題制作でアニメーションを作る際に、ドラえもんや雪だるまの絵を描いたり、車のタイヤを作る場面等に活

用できていた。次にフィールドオブジェクトの命令を加えて円を描いている間の経過時間を表示させた。シューティングゲーム作成の際にはプレイ時間をカウントダウン表示するのに応用したが、難易度が上がり、課題制作時に自力で活用できる生徒は少なかった。最後に乱数でサイコロを作ったり、タートルをランダムに配置する練習をした後、箱（正方形）を繰り返し処理でたくさん作り、それぞれの色と位置を乱数で指定して実行するたびに色と配置が変わるイラストを作った。

当初の予定では6回目の授業からゲーム作りの実習に入る予定だったがこの2回の授業内容の説明をゲームプログラム作成と並行して進めるのは難しいと考え、授業計画を変更して追加した。それぞれが応用プログラムを作成する際、重要な役割を果たす機能で、事前に説明できたことにより、スムーズにゲーム作りの授業を進めることができた。

6.8 第8・9回「ピンポンゲーム」

第8回と第9回の2回の授業で、これまでの技術を総動員してピンポンゲーム作りに挑戦した。ゲームプログラムの作成を通して今までの技術を状況に合わせて応用する力と実戦力を身につけるとともに、学んできた技術が結実して総合的なプログラムになる感動を体験させる目的で実施した。また、ゲームの難易度を上げるためにはどうすればよいかを生徒に考えさせた上で、ボールの動くスピードやパドルの大きさを変更したり、フィールドで得点を表示する等の改良を加えた。

今回はこれまでのドリトル学習の中で最大のプログラムであり、総合的なプログラム作りを経験することにより大きな成果を得られた。生徒もこれまでの学習の集大成としてゲームプログラムを作り上げることができ、達成感を得ることができ、場合によってはこれを一つのゴールとすることも考えられる。課題制作の際にはブロック崩しゲーム等に活用する生徒もおり、大きな影響を受けていた。

6.9 第10・11回「シューティングゲーム」

第10回と第11回の2回の授業で、ピンポンゲームより難易度の高いシューティングゲームに挑戦した。よりレベルの高いゲーム作りに取り組むことにより更に実践力を高めるとともに、生徒にも馴染の深いシューティングゲームを自分自身の手で作ることができるプログラミングの魅力と可能性を感じてもらう目的で実施した。

まず、ゲームの主役となるタートルに必要な設定をする。上から迫ってくる敵をミサイルで迎え打つため、タートルに上を向かせて、左右に移動するボタンとミサイルを発射するボタンを用意する。次に敵となるタートルを5匹作り、配列に入れて同時に右へ6回、1段下がつて左へ6回移動するようタイマーを設定する。最後にミサイルが敵に当たった時、敵が画面と配列から消えるよう衝突を定義

し、制限時間内に敵が全滅し配列の要素数がゼロになった場合ゲームクリアとなるようタイマーと条件分岐で設定した。またフィールドオブジェクトを利用して制限時間をカウントダウン表示した。

シューティングゲームはドリトルテキストを利用した学習者がプログラミングを楽しむための題材であり、学校での教育用に用意されたものではないため、授業で扱うには少し不向きな所もあり、次年度以降は題材を再度選定し直す必要がある。しかし、今回に関してはシューティングゲームが完成し、自分で作ったゲームを試行した時の感動体験ができ、ここまでの技術を習得すればこのレベルのプログラムを作ることができるという、目標になったという効用もあった。

6.10 第12回「アニメーション」

ここまで三角形や四角形等、繰り返し処理を利用した図形しか描いていなかったため、課題制作の際に作成できる図形オブジェクトの幅を広げる目的で、歩数と回転角度を細かく指示して帆掛け船のオブジェクトを作り、タイマーで川を走らせるイメージのアニメーションを作成した。課題制作の際には、生徒は城や車等、様々な図形オブジェクトに応用しており、アニメーションを作るイメージも膨らんだようだった。

6.11 第13～16回「課題制作」

12回のドリトル指導を終えた後、4回の授業で生徒はオリジナルプログラムの作成に取り組んだ。

12回の指導中、毎回生徒は新しい発見に出会い、驚きながら楽しく授業を受けていたが、それでもやはり指導されているうちは受け身の態度になっていた。それが自分でオリジナルプログラムを作らなければならない段階になって、初めて自分が作品を作るだけの技術を習得できていないことを痛感し、課題製作のために試行錯誤を繰り返すことで、初めて技術を自分のものとすることができた。そして、自分のアイデアを自分のプログラムで実現しようと取り組むことによって、プログラミングの楽しさが倍増した。コンピュータの操作に自信がなく、授業を受けている時は苦痛でしかなかった生徒も、自分でプログラムを作り上げる過程になって、急におもしろくなった生徒も少なくないようだった。そして、少しでもミスがあると全く動いてくれないプログラムの本質を自ら体験し、世の中にあふれるソフトウェアの凄さを実感し、敬意を持つようになっていった。また、自らの技術の未熟さを痛感し、もっともっと腕を磨いて、もっとレベルの高いプログラミングをしてみたいという意欲が芽生え、格段に成長することができた。生徒の作品例を図3に示す。

作品1は画面を下へとスクロールさせ、空から海、砂浜へと場面が移り変わる。砂浜の場面に移り変わると、緑の

カメと赤いカメが近づきハートマークが出てくるラブストーリー仕立てのアニメーション作品である。この作品では、「繰り返し処理」、「図形オブジェクト」、「衝突定義」、「分岐条件」、「論理演算」、「タイマーオブジェクト」、「色オブジェクト」が使用されている。プログラムの行数が78行(2077字)と少なく「繰り返し処理」が頻繁に用いられていることから、効率的にアニメーションが作成されている。

作品2は大きな魚と小さな魚が画面右へ、ヨットが左側へとフェードアウトしていき、最後に「Thank you for watching!」と文字が表示されるアニメーション作品である。この作品では、「図形オブジェクト」、「タイマーオブジェクト」、「色オブジェクト」、「ラベルオブジェクト」を使用し、作成されている。プログラムの行数が175行(9611字)と多くなっているのは繰り返し処理を使わずに逐次処理としてプログラムを書いているためである。

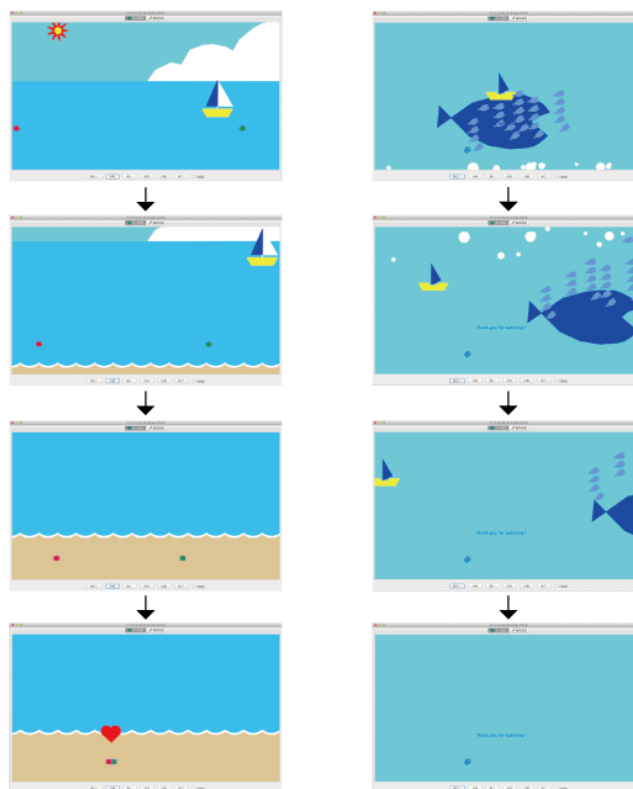


図3 生徒のアニメーション作品例(作品1、作品2)

7. 授業の振り返り

一通りドリトルの学習を終えた後、アンケート調査をおこなった。結果を表5に示す。

楽しかったと答えた生徒が96.9%おり、12回の授業の間、飽きることなく興味・関心を持続させることができた。初回のプログラム体験の授業とは違い、条件分岐や配列等、プログラミングの専門的な内容まで学習したので、序盤の

頃のように、ただただ楽しかったというわけにはいかないが、それでも「難しすぎてまったく理解できなかった」と答えた生徒は14名(5.3%)のみで、汎用言語を学ぶ場合に比べてドリトルがプログラミング初学者にとっていかに学びやすい言語であるかを示している。

自由記述では「ドリトルを学ぶことによってプログラミングがどういうものかを知り、プログラミングの楽しさを知った」という内容の回答が多かった。自分でアニメーションやゲーム作りを体験したことで、世の中にあふれているソフトウェアの凄さを実感し、それを作っているプログラマーの方々に、敬意を表す回答が多かったのも印象的であった。

以上のアンケート結果から、普段利用しているソフトウェアのすべてを形作っている身近であるはずのプログラミングを現状では多くの生徒が経験することなく大人になっている。早い時期からプログラミングの経験を持ち、体験的にソフトウェアの仕組みや性質を実感として理解するために「情報」の授業にプログラミングを取り入れることの重要性を改めて強く認識した。

8. 次年度以降の改善点

今年度は、初回の授業でまずは一気に簡単なゲーム(宝物拾いゲーム)作りまで体験させ、感動を味わわせてから、2回目以降の授業で基礎技術を指導していくという観点で授業を計画した。しかし、実際に授業を展開してみると、地味だと思われた基礎の実習でも毎回生徒は驚きの声をあげており、初回で生徒の心をつかむことに執着する必要はなく、次年度は初回から段階的に基礎を学ばせた方がより効果的であると考えている。

生徒は自分のアイデアでオリジナルプログラムを制作することにより、初めて自分の知識や技術がまだまだ未熟であることを痛感し、自力でプログラムが作れるよう腕を磨きたいという欲求が強くなっていった。また、最後の相互評価で皆の作品を鑑賞することにより、他の生徒のレベルの高さに驚き、かなりの刺激を受けていた。次年度は技術の指導内容を更に厳選し、指導回数を削減することにより2回の課題制作を取り入れ、「基礎技術の習得→課題制作(1)→相互評価・反省→応用技術の習得→課題制作(2)」というサイクルを組み込みたい。ただし、今年度の課題制作は当初2時間で実施の予定だったが、生徒の構想が自分の技術以上に膨らみすぎてとても間に合わなくなってしまったため4時間に延長した。次年度以降実施する際は、計画段階で生徒の状況をしっかり把握し、必要に応じてある程度の方向性を示しつつ、2時間で完成できるよう配慮する。

また、オリジナルプログラムを作成することにより、自分自身が著作者になることを体験し著作権に対する意識が変わるので、このタイミングで知的財産権に関する授業を実施したいと考えている。

表 5 事後アンケート(集計人数 262 名)

Q1: プログラミングを体験してみてどうでしたか?

回答	人数	割合	
とても楽しかった	152	58.0%	96.9%
まあまあ楽しかった	8	2.9%	
あまり楽しくなかった	6	2.3%	3.1%
全然楽しくなかった	2	0.8%	

Q2: プログラミングに興味が湧きましたか?

回答	人数	割合	
とても興味が湧き、自分でもやってみたいと思った	78	29.8%	81.7%
興味は湧いたが、自分には無理だと思った	136	51.9%	
ちょっとは興味は持ったが、自分ではやらないと思う	47	17.9%	18.3%
まったく興味が湧かなかった	1	0.4%	

Q3: プログラミングとはどういうものかわかりましたか?

回答	人数	割合	
とてもよくわかった	40	29.8%	81.7%
まあまあわかった	196	74.8%	
あまりわからなかった	26	9.9%	9.9%
まったくわからなかった	0	0.0%	

Q4: プログラミングの難易度はどうでしたか?

回答	人数	割合	
とても簡単だった	8	3.1%	35.5%
思っていたよりは簡単だった	85	32.4%	
さすがにちょっと難しかったがなんとかあった	155	59.2%	64.5%
難しすぎてまったく理解できなかった	14	5.3%	

Q5: 今後もプログラミングを学んでみたいですか?

回答	人数	割合	
ぜひ、もっと深く学んでみたい	65	24.8%	86.6%
ちょっと学んでみたい	162	61.8%	
それほど学びたいとは思わない	34	13.0%	13.4%
まったくやりたくない	1	0.4%	

今年度の授業実践で、初めてプログラミングを学ぶ際の言語はドリトルが最適であると感じている。その上で、従来実施していた JavaScript の授業を組み合わせることで更なる効果が期待できる。様々な言語を多角的に学ぶことで、どの言語を学ぶ際にも共通する必須の知識を理解し、その上で言語ごとの特徴を知り、変化に対応できる柔軟性と場面に応じて手段と方法を使い分ける応用力を身につけることができると期待している。

9. まとめ

これまで、筆者は高等学校「情報科」の授業において、JavaScript を用いた実習をおこなってきた。しかし、JavaScript での実習は初学者にとって簡単なものではなく、

生徒にある程度の知的好奇心と理解力がないと実施できない特殊な内容もあり、生徒の関心・意欲・態度にも個人差が大きかった。

そこで、現在主流のオブジェクト指向の考え方を基礎とした教育用プログラミング言語である「ドリトル」に着目し、授業をおこなった。

事前と事後のアンケート結果から、12回の授業の間、飽きることも無く興味・関心を持続させることができたということがわかった。また、生徒が作成したアニメーション作品から、プログラミングの基本的な概念を生徒が学ぶことができたと考えられる。

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] プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp/>
- [2] 兼宗進. 工学系学科でのプログラミング入門教育-ドリトルを利用して-. 情報処理, Vol.52, No.6, pp.736-739, 2011.
- [3] 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, 日本ソフトウェア科学会, Vol.28, No.1, pp43-48, 2011.
- [4] 兼宗進, 阿部和広, 原田康徳. プログラミングが好きになる言語環境. 情報処理, 特集未来のコンピュータ好きを育てる, Vol.50, No.10, pp.986-995, 2009.
- [5] 兼宗進, 久野靖: プログラミング言語「ドリトル」. イーテキスト研究所, 2007.
- [6] 久野靖: 5時間でプログラミングを学ぶ（ドリトル編）.
<http://www2.gssm.otsuka.tsukuba.ac.jp/staff/kuno/lectures/GEN/2009-09-Dolbook09.pdf>
- [7] 紅林秀治: ドリトルテキスト（ドリトルを覚えよう）.
<http://dolittle.eplang.jp/data/pdf/dolittleKurebayashi.pdf>
- [8] 兼宗進: 1時間で学ぶソフトウェアの仕組み, 2009.
<http://dolittle.eplang.jp/?1h>