

プログラミング教育用授業資料の研究および授業実践報告

愛媛県立南宇和高等学校

古田 賢司

1 はじめに

新学習指導要領では、年々高度化する情報社会に対応するため、教科「情報」においてプログラミング教育が必修化される見通しである。そういった現状を踏まえて研究委員会では、5か年計画でプログラミング教育を実践するための準備を進める中で、高等学校における現行の『社会と情報』と『情報の科学』との関連や、義務教育における学習内容と教科「情報」との関連について、研究を行っている。今年度の研究では、これまでの研究内容を踏まえた上で具体的にどんなプログラミングに関する授業が実施できるかということを中心に、授業内容の提案および実践結果の報告を行うこととした。

2 概要

昨年度までの研究を確認したうえで、高校1年生を対象とする授業案をいくつか考え、実際に授業を行って結果の考察を行う。

3 研究内容

(1) 昨年度までの研究内容と概略の確認

ア 『社会と情報』と『情報の科学』について

愛媛県下の多くの学校が『社会と情報』を履修している。そのうち実質的なプログラミング実習を取り入れているところは少ないが、表計算ソフトの実習を行う中で『情報の科学』の内容を取り入れている学校もあることがわかっていく。『情報の科学』を履修している学校では、座学でモデル化やアルゴリズムに関する授業を行い、具体的な実習はされていない場合もある。このように履修科目に限らず、プログラミングに関する実習を行うことはハードルが高いということがわかったが、エクセルによるマクロ実装を表計算実習の延長として行っている事例もある。

イ 義務教育における教科「技術・家庭」について

中学校の教科「技術・家庭」では、専用の学習キット等を用いて本格的なフローチャート構築やプログラミングに関する学習を行っている。特に、フローチャートに関しては、順次処理や条件分岐、反復などプログラミングの基礎となる考え方をしっかり学習していた。これらが既習であるということは、高校でプログラミングに関して学習をする際にとっても心強く感じた。

(2) 授業の実践内容の提案

ア スクラッチを用いた実習

スクラッチとは、マサチューセッツ工科大学（MIT）のメディアラボが開発したプログラミング学習用ソフトである。様々な言語に対応しており、世界中のプログラミング教育で最も使われているソフトと言われている。NHK教育テレビでスクラッチの番組が放送されたり、小学生がスクラッチを利用したゲームを制作して発表するイベントが開催されたりと、教育の場で使われることも多い。

特徴は「ビジュアルプログラミング」と呼ばれる、視覚的なオブジェクトで構成されたプログラミング言語であることで、命令の書かれたパーツをパズルのように組み合わせることで簡単に作品を作ることができる。わかりやすさや、命令によって可能な制御の幅広さ、エラーに対する融通の効き方は評価が高い。一覧にある命令を直接クリックすることで、オブジェクトがどんな動きをするか確認できるという点も、便利であると感じた。

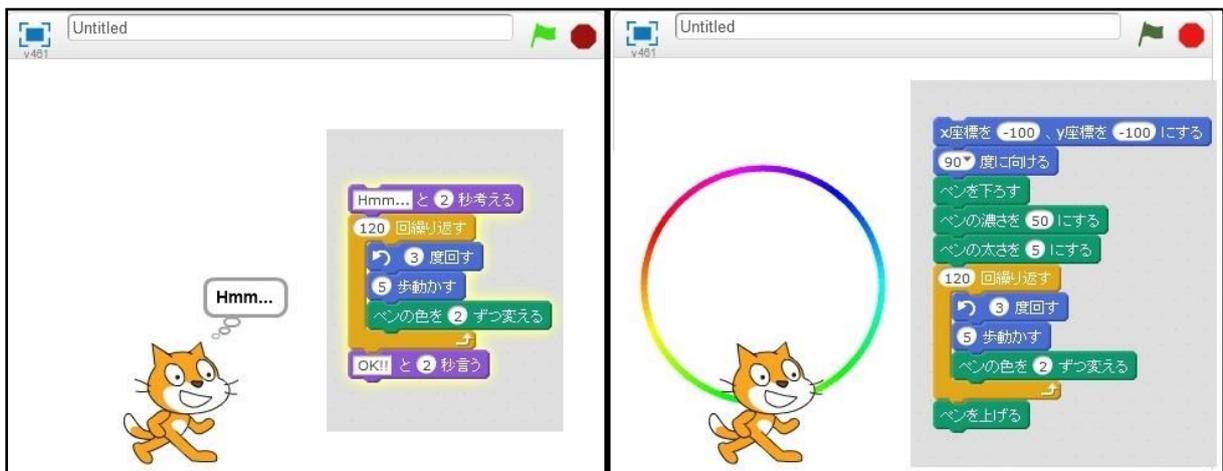
授業としては、ビジュアルプログラミングの導入として利用すると良いと思われる。

例えば（図1-a）は、オブジェクトの初期位置を指定した上で一回転させる命令文である。

（図1-b）はそれに加えてオブジェクトに吹き出しを表示させたり、軌跡をカラフルな線でなぞらせたりしたものである。



（図1-a）



（図1-b）

上から順番に命令を実行していくことや、ブロックで挟んだり繋げたりして一連の命令を作ることなど、ビジュアルプログラミングがどんなものかを直感的に理解することが出来る。

イ Google Blocklyを用いた実習

Google BlocklyとはGoogle社が提供するビジュアルプログラミング言語である。各種プログラミング言語に対応した命令が、シンプルなブロックで視覚化されており、それらを組み合わせることでプログラミングを行うことができる。Blockly Gamesのページには、Blocklyを用いたゲームがいくつか紹介されており、中でも「迷路」(図2-a)はパズルのような感覚で反復や条件分岐を、「タートル」(図2-b)は後述のドリトルで紹介するような、オブジェクト移動による軌跡の描写をおこなうことができる。どちらも全10ステージ用意されているが、すべてを解くためには繰り返しや条件分岐など様々な考え方を組み合わせる必要であるため、それらを考えさせるだけでも十分な論理的思考力の訓練になる。



(図2-a)

(図2-b)

注意事項として、Web上で動作する場合、学校の設定によっては上手く動作しないこともあるようだ。また公式ページは日本語化されていないが、平易な英語しか使われていないため高校生ならば問題ない。また、Blocklyは現在Android版とiOS版の開発アプリがダウンロード可能であるため、ICT教室にタブレットが備え付けられている学校ならばインストールして授業で利用することができるかもしれない。

授業実践の資料として、いくつかプログラムを作成してみた。次のページから授業プリントの例を掲載しているので、参考にいただくと幸いです。

1 カウントダウン

右図のようにブロックを組んでみましょう。実行すると、カウントが1から5までされます。最後にドーン！と表示され、処理終了です。

しかしこのままではカウント『ダウン』とは言えません。そこで、ブロックの中の数字や演算子を編集して、5から1へのカウント『ダウン』となるようにしてください。

```
set Count to 1
print "カウントダウン!"
repeat while Count ≤ 5
do
  print create text with "残りカウント"
  print Count
  set Count to Count + 1
print "ドーン!"
```

2 大小判定1

上側にあるブロックは、1から100までからランダムに1つの数字を作り、変数にセットするというものです。

そこで、図の下側にある『if』ブロックと『print』ブロックを用いて、

「作られた変数が50より小さければ、50より小さいと表示するプログラム」を作ってください。

```
set Count to random integer from 1 to 100
print create text with Count
print "を作りました。"
if do
  print "処理を終わります。"
  set Count to 1
print create text with Count
print "は50より小さいです"
```

3 大小判定2

②で作ったプログラムにある『if』ブロックを『test』ブロックに変えて、次のようなプログラムを作りました。

①と②に入る適切な文章を考えて下さい。

```
set Count to random integer from 1 to 100
print create text with Count
print "を作りました。"
print test Count < 50
if true
  print create text with Count
  print "は ① "
if false
  print create text with Count
  print "は ② "
print "処理を終わります。"
```

1 カウントダウン

```
set Count to 5
print "カウントダウン!"
repeat while Count >= 1
do
  print create text with "残りカウント"
  Count
  set Count to Count - 1
print "ドーン!"
```

注意点は、以下の3点です。

- 『set』ブロックの初期値を、5にする
- 二つ目の『set』ブロックの演算処理を、変数-1にする
- 『repeat』ブロックの反復条件を、1以上に設定する

2 大小判定1 (if関数)

```
set Count to random integer from 1 to 100
print create text with Count
"を作りました。"
if Count <= 50
do
  print create text with Count
  "は50より小さいです"
print "処理を終わります。"
```

注意点は、以下の2点です。

- 『if』ブロックの反復条件を、50以上に設定する
- 処理の終了を示す『print』ブロックは、『if』の外に設置する

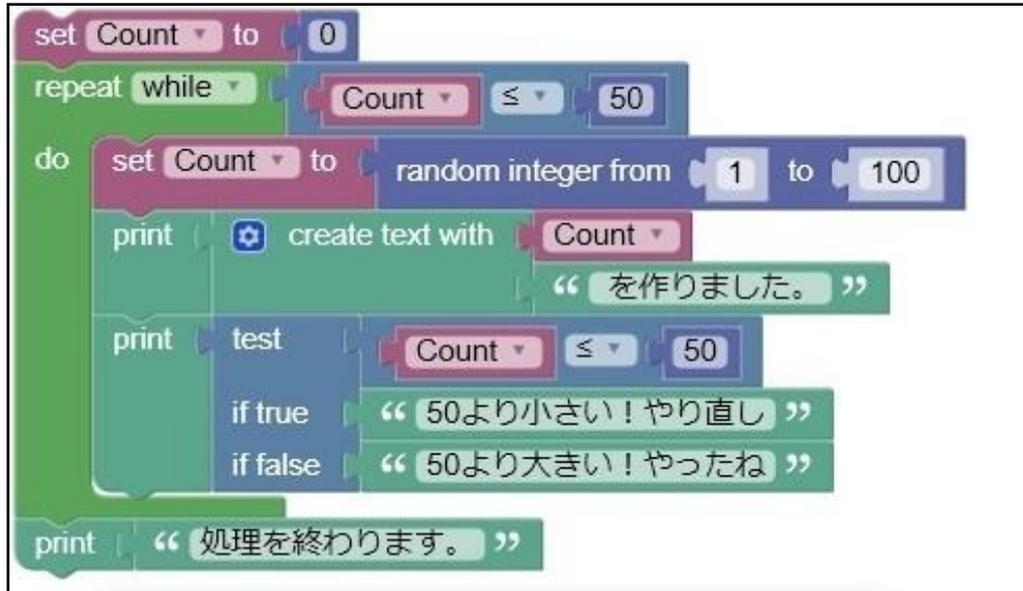
3 大小判定2 (test関数)

- ① 「～は、50より小さいです」
 - ② 「～は、50以上です」
- ※ 変数が50のときは『if false』に分岐することに注意させる

問 3で作ったプログラムに『repeat』ブロックを加えて、以下のルールに沿ったプログラムを作って下さい。

- ① 50以上の変数になるまで、変数作成および50以上かの判定を繰り返す
- ② 50以上の変数が出たら処理を終了する

答



注意点は、以下の3点です。

- 初めに変数へ0をセットする
- 『repeat』ブロックの反復条件を、50以下に設定する
- 『repeat』ブロックの反復処理させる範囲に気を付ける

ちなみにこの問題で作ったプログラムをJavaScriptでプログラムすると以下の通りであり、Blocklyがいかにかかりやすく視覚化されているかわかる。

```

var Count;
function mathRandomInt(a, b) {
  if (a > b) {
    var c = a; a = b; b = c;
  }
  return Math.floor(Math.random() * (b - a + 1) + a);
}
while (Count <= 50) {
  Count = mathRandomInt(1, 100);
  window.alert(String(Count) + String('を作りました。'));
  window.alert(Count<=50?'50より小さい!やり直し':'50より大きい!やったね');
}
window.alert('処理を終わります。');
play_arrow

```

ウ ドリトルを用いた実習

ドリトルとは単純な日本語の組み合わせで記述することができる教育用プログラミング言語である。Blocklyの「タートル」に似ているが、こちらは実際にプログラムを書く必要があるため、より実践的な内容となっている。特に、ルールから外れると構文エラーが発生すること、書き方によってはこちらの期待したことと別の動きをしてしまうことなど、実際のプログラミングにありがちなことを体験することができる。ドリトルはダウンロード版とWeb版があるため、Web上で動作しない場合はダウンロードしたファイルから実行すると良い。

(図3)は、昨年度に紹介したドリトルの画面であり、左が命令の編集画面、右が実行画面である。また(図4)は、授業後のアンケート結果である。



(上：図3 下：図4)

問1	「ドリトル」は、やってみてどう感じたか。	やり方を勉強すれば出来そう	簡単なので、すぐに出来そう	とても難しかった
		6	22	0
問2	次があるとしたら、やってみたいか。	ぜひやってみたい	時間があればやってみたい	どちらでもいい
		13	15	0
問3	プログラミングがどんなものか、なんとなくわかったか。	わかった	どちらかというかわかった	よくわからない
		24	4	0

(3) 授業実践の結果と考察

今年度は2時間の授業を行い、最後に授業に関するアンケートを行った。

ア フローチャートの復習とスクラッチの体験（1時間目）

初めに、中学校の既習事項を思い出させるためにフローチャートについて確認した。フローチャートに関しては、例えば『カレーをつくるために』等の身近な例を紹介し、料理を作る手順をフローチャートに変換させるなどして、反復操作や条件分岐の考え方について簡単に復習した。

続いてスクラッチのホームページにアクセスさせ、自由に命令文を作らせた。その際に（2）-アで示したような例を生徒に紹介した。生徒同士で相談をしながら、様々な命令を積極的に試している様子がみられた。

イ ドリトルによる実習（2時間目）

スクラッチで行ったプログラミングを、実際に記述してみようということで、ドリトルを紹介した。実習の際は、ドリトルを共有サーバにダウンロードし、そこに直接アクセスしてbatファイルを開かせた。今回の授業で利用したプログラム例は以下の通りである。なおこれらのプログラムはドリトルの開発元である大阪電気通信大学兼宗研究室のホームページで紹介されているものを参考にしている。

(例1) ボタンオブジェクトの作成

かめた=タートル！ ペンなし 作る。
加速ボタン=ボタン！”加速”つくる。
加速タン:動作=「かめた！10 歩く」。
左ボタン=ボタン！”左” 作る。
左ボタン:動作=「かめた！10 左回り」。
右ボタン=ボタン！”右” 作る。
右ボタン:動作=「かめた！10 右回り」。
トンボボタン=ボタン！”トンボ”つくる。
トンボボタン:動作=「かめた！”tonbo.gif” 変身する」。

(例2) 宝集めゲーム

※例1の続きに以下のプログラムを記述する。
宝=タートル！作る”tulip.png” 変身する ペンなし。
宝！（乱数(600)-300) (乱数(300)-150)位置。
宝！作る(乱数(600)-300) (乱数(300)-150)位置。
宝！作る(乱数(600)-300) (乱数(300)-150)位置。
かめた:衝突=「| 相手 | 相手！消える」。
時計=タイマー！作る 200 回数。
時計！「かめた！10 歩く」実行。

今回の授業の目的はコーディングをさせることではなく、その意味を考えさせることであったため、前述にあるプログラムをメモ帳ファイルで生徒に送信し、コピーしてドリトルの編集画面に貼り付けさせ、生徒自身がプログラムを写して書く事による手間を省略させた。

そのようにして動作を確認させた後、次のような問いを生徒に与えた。

- 問1:「かめた! 10 歩く」「かめた! 10 左回り」について
 数字を変えたら亀の動きはどのように変わりましたか?
 問2:トンボボタンを押すと何が起こりますか?
 問3:トンボボタン:動作=「かめた! "tonbo.gif" 変身する 」について
 この命令の tonbo.gif 部分を変更すると、どうなりますか?
 問4:時計=タイマー! 作る 200 回数。時計! 「かめた! 10 歩く」。について
 数字を変えたら亀の動きはどのように変わりましたか?
 問5:このプログラムを応用すると、どんなことが出来そうですか?

各問いに対して生徒は以下のように答え、プログラミングについてある程度は理解してくれた様子だった。

- 答1:それぞれ、「歩く」は移動距離、「左回り」は回転の角度が変わった。
 答2:カメがトンボに変わった。
 答3:例えば fish.png にすると、亀が魚になった。
 (※ドリトルの初期設定で、fish.png は魚の絵と定義されている)
 答4:「タイマー! 作る」の部分は、数を増やせばその分カメが長く動いた。
 「10歩く」の部分で数を増やしたら、一步がその分大きくなった。
 答5:パソコン上で絵が描ける、ロボットに命令、ゲームが作れる
 探査機の遠隔操作、自動で動くカメを作る など

ウ アンケート結果と考察

昨年度は亀を動かして図形を書くだけだったが、今年は亀を自由に動かしながら、変数を変えることで挙動がどう変化するかを考察をさせ、さらに発展としてタイマー機能とオブジェクト変更機能を用いた宝探しゲームを実装させた。より複雑な内容を扱ったためか昨年度より難しいという印象をもった生徒が多いが、その分プログラミングへの実感を持てた生徒も多かった。

問1	「ドリトル」は、やってみてどう感じたか。	やり方を勉強すれば出来そう	簡単なので、すぐに出来そう	とても難しかった
		2 5	4	5
問2	次があるとしたら、やってみたいか。	ぜひやってみよう	時間があればやってみよう	どちらでもいい
		1 3	1 5	6
問3	プログラミングがどんなものか、実感できたか。	わかった	どちらかというかわかった	よくわからない
		3 0	4	0

4 まとめ

今年度の研究では、様々な学校で活用できそうな授業資料を提示し、また自分自身のプログラミングに関する勉強をすることができた。今回取り上げたものは、以前の研究で紹介したドリトルやスクラッチ、新規で紹介したものではG o o g l e B l o c k l y など、W e b 上で公開されているソフトウェアなどであったが、様々なソフトウェアやプログラミング言語について調べるうちに、色々なソフトウェアやW e b ページがあることもわかってきた。特にこれからのプログラミング教育に関して、ビジュアルプログラミングの分野には大きな可能性を感じている。

今後、高校で行われるプログラミング教育については逐次的に具体的な指針が示されていくと思われる。新しい情報にアンテナをはりつつ、私たち教員も研さんを積みながら対応していかないといけない。今回の研究内容が、不安をかかえる情報科教員にとって少しでも刺激や参考になるものであれば、幸いである。

5 参考文献

(1) スクラッチ

ア S c r a t c h 日本語版ページ

<https://scratch.mit.edu/>

イ N H K による紹介ページ

<https://www.nhk.or.jp/school/programming/start/index.html>

(2) G o o g l e B l o c k l y

ア G o o g l e 社によるB l o c k l y 紹介ページ

<https://developers.google.com/blockly/>

イ B l o c k l y G a m e s (日本語対応は一部のみ)

<https://blockly-games.appspot.com/>

(3) ドリトル

ア 大阪電気通信大学兼宗研究室による紹介ページ

<https://dolittle.eplang.jp/>