

連載



情報の授業をしよう！

本コーナー「情報の授業をしよう！」は、小学校や中学校で情報活用能力を育む内容を授業で教えている先生や、高校で情報科を教えている先生が、「自分はこの内容はこういう風に教えている」というノウハウを紹介するものです。情報のさまざまな内容について、他人にどうやって分かってもら

うか、という工夫やアイディアは、読者の皆様にもきっと役立つことと思います。そして「自分も教え方の工夫を紹介したい」と思われた場合は、こちらにご連絡ください。

(E-mail : editj@ipsj.or.jp)

プログラミングの授業 段階を踏んでやってみた

岡本弘之 | アサンプシヨン国際高等学校



授業のねらい

最初に反省を書くと、2002年から情報科が始まり情報A・C、社会と情報の科目を教えてきたが、その中でプログラミングの授業を教えたことはほとんどなかった。しかし2022年に始った情報Iではプログラミングが必須の内容となり、かつテキストプログラミングを行うことが求められている。

情報Iの教科目標から考えると、プログラミングの授業では問題解決まで体験させることが理想だが、テキストプログラミングで問題解決を行うことは、生徒にも教員にもハードルが高い。

本実践ではテキストプログラミングの基本も学習しつつ、制作や問題解決をScratchやロボット教材を使ったプログラミングにゆだね、教員も生徒もハードルが低い授業を目指してみた。本実践の工夫のポイントは次の4点である。

- ①プログラミングをScratchから始めてみた
プログラミングの授業の最初で、「できた」「楽しい」とポジティブな印象を持ってほしいと思い、生徒にも分かりやすいScratchから始めてみた。
- ②コードを使って短いプログラムで基本を実習した
テキストプログラミングは、演算、変数、配列、分

岐、繰り返し処理などを行う短いプログラムを用意し、実習を通じて学ぶ授業とした。

③プログラミングでの問題解決はモノを動かした画面で完結するプログラミングより、実際にモノを動かして試行錯誤を繰り返してほしいと考え、ロボットを使った問題解決の課題を行った。

④シミュレーションするためにコードを使った最後にテキストプログラミングによるシミュレーションの実習を行い、目的を明確にしてプログラミングをしてみた。

授業の展開と生徒の反応

本授業は、高校2年生情報I(2単位)の2学期の11時間を用いて、学習指導要領の「(3) コンピュータとプログラミング」の単元の「イ. アルゴリズムとプログラミング」「ウ. モデル化とシミュレーション」の部分までを授業した。

授業案の検討にあたっては目標として

- 体験や実習を通じて学ばせたい
- 書き写す「写経」だけでなく工夫させたい
- 成功失敗も含め試行錯誤し問題解決させたい
- 共通テスト対策にも役に立つ授業としたい

の4点を考え、具体的には次の流れで企画した(表-1)。

アルゴロジックで学ぶ(2時間)

アルゴリズムの説明では、体験的に理解できるように JEITA「アルゴロジック2」の実習を行った。

アルゴロジックは、図-1の画面のように、ロボットが旗をとるための動きの手順を考えて命令のブロックを並べ、ステージをクリアしていくゲームである。順次処理、繰り返し、条件分岐とステージが進むにつれ、複雑な手順を考える必要がある。また同じ課題で複数の手順が取れるものもある。授業では多くの生徒が順次処理、繰り返し、条件分岐までクリアできるように25分と時間を区切り、各自の1人1台端末を使って個人で取り組ませた。実習後の解説では、この処理手順がアルゴリズムであり、①この手順をコンピュータに分かるようにプログラム言語で書くのがプログラミングであること、②アルゴリズムでは効率の良い最適手順を考えることが大切なこと、③順次・分岐・反復の基本的な処理があることを説明した。その後身近な作業の手順を考える課題を行い、教科書の内容を実習を通じて体験的に学習した。

生徒の振り返りからは、「コンピュータに手順を教えるには労力がかかる」「小さな工夫でも手順に差が

出るといった」「人間のよう経験や勘で動けない分、コンピュータにはすべての動きを指示しないと行けない」「AIは賢いが、そのプログラムを作る人間はもっと賢い」などがあった。コンピュータを動かすには、いきなりプログラミングをするのではなく、細かな手順に分解し、より効率の良いアルゴリズムを考えることが大切ということを書いた振り返りが多くみられた。

授業の様子としては、生徒はアルゴロジックをゲーム感覚で取り組み、友人にもヒントをもらいながら1つずつ次のステップに進めていった。その後の説明では、アルゴロジックでの経験と重ねて説明したことで、必要な知識についての生徒の理解はスムーズだった。

Scratchでプログラミング(2時間)

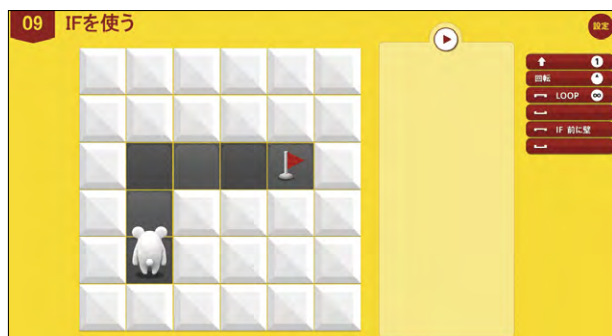
最初はプログラミングに慣れてもらうため、Scratchを使って簡単なゲームを作る実習を行った。日本文教出版教科書『情報I 図解と実習』に掲載されている「あっち向いてホイ」(図-2)、「暗算ゲーム」の2つを使って、見本どおりそのままプログラムを入力するいわゆる「写経」の授業を行った。

見本どおり入力したあと、音を鳴らしたりとプログラムを各自で改善すること、さらに発展課題として「じゃんけんゲーム」を自分でプログラムも考えて作る実習も行った。

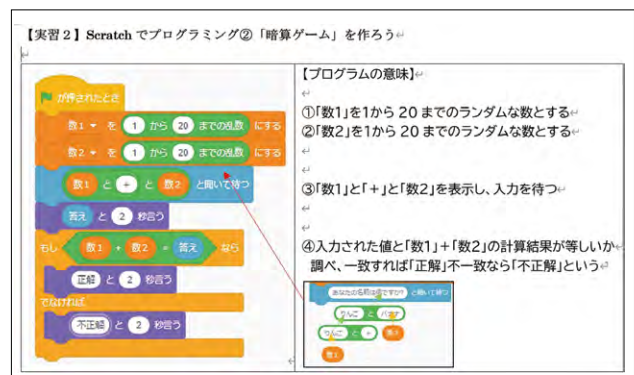
生徒の振り返りからは「思ったより簡単に自分でもゲームが作れてうれしい」「乱数を利用しただけで、いろいろなゲームが作れそう」、「プログラミン

■表-1 授業の流れ(11時間)

単元名	授業(配当時間)
アルゴリズム	アルゴロジックで学ぶ(2時間)
プログラミング①	Scratchでプログラミング(2時間)
プログラミング②	Pythonの基本を実習で学ぶ(2時間)
プログラミング③	ロボットを動かそう(3時間)
シミュレーション	Pythonでシミュレーション(2時間)



■図-1 アルゴロジックの画面
(<https://algo.jeita.or.jp/prm/2/index.html>)



■図-2 Scratchの課題「じゃんけんゲーム」
(日本文教出版教科書『情報I 図解と実習』)

グは難しいと思っていたが思ったより簡単」,「プログラミングができれば、ほしいアプリを作れそう」,「アプリを使うときに、条件分岐・乱数と考えるようになった」,「ゲームのガチャの乱数調整の意味が分かった」などがあった。「できた」という達成感,「おもしろい」という興味関心,「プログラミングで何かできそう」という期待が多くみられた。

授業は実際の入力画面を提示しながら一緒に進めたが、多くの生徒は十分理解できていた。発展課題や制作課題においても、Scratchは視覚的に意味が理解できるので、少しのヒントを与えるだけで、自分で考え取り組めた生徒がほとんどであった。

Python の基本を実習で学ぶ (2 時間)

次にコードによるテキストプログラミングでは、ブラウザ上のプログラム環境「paiza.IO」でPythonを使った。演算・変数・条件分岐・繰り返し・配列についてそれぞれ6行程度の短いプログラムを見本どおりに入力し、実行して結果を確認する授業を行った。

ワークシートにはプログラムだけでなく、図-3のようにそのコードが何を意味しているかの説明も記載した。入力の段階で何をしようとしているのかを理解するために記載したが、共通テストでも問題文の手順に対応するプログラムを考える問題が出題されることから、横に意味を書いたことは生徒の理解を深めた。

授業ではワークシートの見本のどおりに入力をして、実行し結果を確認する「写経」の授業の後、自分で計算式や数値を変更するなど、一部を変えて入力して、結果を確認することまでは取り組めた。

生徒の振り返りには「プログラミングは複雑で難しい」,「少しでも間違えるとうまくいかず、頭が混乱した」,「エラーが出るともやもやした」といった

「難しい」という意見が多い一方,「プログラミングコードの意味が分かった」,「ゲームでランダムに武器が出てくる仕組みが分かった」,「Scratch よりもっと複雑なプログラムができる」,「プログラミングは難しいが、案外できた」と前向きな意見もあった。

授業では、初めてのコード入力でありがちな、半角と全角の誤り、スペースの入力などによるエラーを半数以上の生徒が経験し、授業者はその対応に追われる結果となった。コードをそのまま入力するだけで手いっぱいな生徒も多く、この段階では短いプログラムを「そのまま入力する」,「一部を変える」段階までの指導が妥当と感じた。短いプログラムであれば修正しやすく指導もしやすいからである。

ロボットを動かそう (3 時間)

ここまでのプログラミングでは問題解決のレベルまで取り組むことはできなかったので、LEGO マインドストームをプログラムで動かして、グループで与えられた課題を解決する実習を行った。

課題としては「目標の位置でぴったりと止める」,「センサーを使って障害物をよけてゴールへ到達する」(図-4)という2つの課題を与えた。

ここで体験させたかったのは「プログラミング→実行→評価→改善」という過程を繰り返し、試行錯誤しながら問題解決することであった。また相談して課題をクリアできるようグループで挑戦させた。この問題解決に向けた試行錯誤の過程が分かるようにワークシート(図-5)に記録し、考察・振り返りとともに評価材料とした。

生徒の振り返りには,「課題を達成できたうれしい」,「課題達成のためにはほかのグループの観察も重要だ」,「失敗から試行錯誤を繰り返し、成功には時間がかかる」,「少し数値を変えるだけで動きが変わり成功につながる」,「プログラミングはトライ＆エラーの繰り返しが重要だ」,「物を動かすには、いろいろな要素が増え難しくなる」とあった。問題解決のために、何度も観察・修正や試行錯誤を繰り返し、解決した様子が伝わる記述であった。

授業の様子では、プログラミングで実際にモノを

(3) 条件分岐

TRY3 次の実行結果を予想し、実際に入力してみよう

例 60 点以上なら合格、60 点未満なら不合格と表示するプログラム

```
1 # coding: utf-8
2 # Your code here!
3 tensu=80
4 if tensu>=60:
5     print("合格")
6 else:
7     print("不合格")
```

←変数 `tensu` に 80 を代入
←もし `tensu` が 60 以上ならば
「合格」と表示
←4 行目の結果が異なれば
「不合格」と表示

応用: 変数 `tensu` の値を変えて正しく表示されるか確認しよう

図-3 ワークシートの抜粋

動かすことで、失敗や成功といった結果が分かりやすく、そのことが試行錯誤して問題解決に向かうモチベーションになっていた。情報Iでは「問題解決」がキーワードであり、プログラミングの授業も問題解決の手段として位置づけられており、今回問題解決まで実習できたことは教科の目標とも一致する。

Python でシミュレーション (2 時間)

モデル化とシミュレーションの単元では、シミュレーションをするためにプログラミングを使うという目的を明確にしたプログラミングを行った。

前時と同様に「paiza.IO」上でPythonを使い、「当選確率1%のガチャを100回ひいてレアキャラをゲットできる確率」、「模擬店で必要な釣金を調べる」という2つのシミュレーションを行った。

プログラムの入力に際しては、最初に基本となるプログラム(図-6)を入力し、それを改善していく形式(図-7)をとった。授業者の意図としてはエラーが生じにくいように確認しながら進めるためにこの方法をとったが、共通テストの大問3の出題でも、同様に最初に短いプログラムを作り、それを改

善していく問題形式であり、結果的には共通テスト対策にもなるといえる。

基本となるコードはそのまま入力する形だが、数値を自分で変えたり、条件を変化させる課題にも取り組んだ。2回目のコードによるプログラミングで生徒も教員も慣れてきて、今回は思った以上にスムーズに進むことができた。

生徒の振り返りからは「紙やペンで計算するよりも早くにシミュレーションできる」、「プログラムを使えば、条件を変えることも簡単にできる」、「ゲーム好きなのでガチャの確率を求めるのは楽しかった」、「コツをつかんだらコードも早く打てるようになった」、「乱数を使ったシミュレーションは人によって結果が違った」、「活用すれば危険を回避し出費を抑制できる」、「シミュレーションは防災で役に立つ」とあった。プログラミングでシミュレーションを行うことの利点・可能性、身近なテーマ設定で生徒が興味を惹かれたこと、2回目のテキストプログ

センサー・条件分岐を使おう

・目標: 壁が近づくときUターンして戻るプログラム

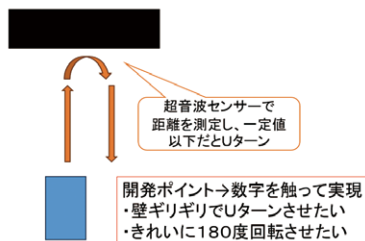


図-4 生徒に提示した課題

◇ワークシート①◇
①自分で考えたプログラミングの内容◇ ②他の実行結果を見ての修正・やってみた結果◇

プログラミングの内容 (設定した数値など)	他の実行結果を見ての修正・やってみた結果
置いたパーツ設定した数値 (例)タンク 50, 50, 50	(例)手前で止まったので もう少し数値を増やす

◇グループでの改善の経過の記録◇

プログラミングの内容	実行結果	振り返り・改善の方向性
1回 タンク 50, 50, 50	手前で止まる	数値を増やす
2回 タンク 50, 50, 60	手前で止まる 曲がった	数値を増やす 置き場所を変える
3回		
4回		

グループの他の人の結果も記録して改善案を考える

図-5 ワークシートの説明

STEP1 確率1%のガチャを作る

```
1 # coding: utf-8
2 # Your code here!
3 import random
4 a=random.randint(1,100)
5 print(a)
6 if a==1:
7     print("あたり")
8 else:
9     print("はずれ")
```

←乱数のモジュールを組み込む
←変数aに1~100の乱数を代入する
←変数aを表示する
←もし変数aの値が1と一致したら
←"あたり"を表示する
←そうでなければ
←"はずれ"を表示する

図-6 シミュレーションのプログラム①

STEP2 100回繰り返しあたりの数を数える

```
1 # coding: utf-8
2 # Your code here!
3 import random
4 b=0
5 for i in range(100):
6     a=random.randint(1,100)
7     print(a)
8     if a==1:
9         print("あたり")
10        b=b+1
11    else:
12        print("はずれ")
13    print("当たりの回数は",b)
```

←当たりの回数を数える変数bを初期化する
←カウンタ変数iが100になるまで繰り返す
←あたりが出たら変数bに1を加える
←変数b=あたりの回数を表示する

図-7 シミュレーションのプログラム②

ラミングで慣れたことなどが書かれていた。

実際に授業でも、1回目のテキストプログラミングのときのような入力ミスによるエラーが格段に減少し、この背景には授業者も慣れて生徒がミスしそうな個所に気を配りながら説明したことや、生徒自身が慣れたことが大きいといえる。

授業の振り返り

本実践では、プログラミングの授業を段階を踏んで、実習を中心に構成して実践した。ここまで述べてきた、それぞれの段階での生徒の振り返りコメントからのキーワードをまとめると図-8のようになる。

生徒はプログラミングをScratchから始めたことで、プログラミングを「楽しい」と感じ、「自分でもできそう」と可能性を感じてくれた。

次にPythonを使ってプログラムの基本を実習したが、やはり入力ミスによるエラーが続出し、生徒のモチベーションは下がった。しかし短い課題を繰り返すうちに成功する生徒は増え、成功すると達成感を感じていた。

ロボットを使ったプログラミングでは、実際にモノを動かすことで再び生徒のモチベーションが上がり、プログラミングによる問題解決の段階まで実習することができた。

最後のシミュレーションでは再度Pythonを使ってシミュレーションを行ったが、2回目ということでありスムーズに取り組むことができ、シミュレーションをするという目的も達成できた。

今回の実践をふまえて、自分が感じたことをまとめると、以下ようになる。

生徒の振り返りのキーワード

2.2 Scratchでプログラミング

- ・楽しい、思ったより簡単、自分でもできた
- ・がんばれば複雑なプログラミングもできそう

2.3 Pythonでプログラミング

- ・難しい、エラーが出るとややもやする
- ・うまくいったら楽しい、意味が分かった

2.4 ロボットでプログラミング

- ・難しいが楽しい、
- ・試行錯誤、動かしてみても修正

2.5 Pythonでシミュレーション

- ・条件を簡単に変えられる、素早くできる
- ・重ねることにプログラミングが楽しくなった

■図-8 生徒の振り返りのキーワード

1つ目はプログラミングに苦手意識を持たせないために、Scratchから始めてもいいと考える。いきなりテキストプログラミングに取り組みエラーが頻発して苦手意識を持つよりも、Scratchで「プログラミングって楽しい」、「自分でも何かできそう」という気持ちを持ってもらう方がいいと考える。Scratchであれば、課題そのまま入力する「写経」で終わるのではなく、「一部を変えてみる」、「自分で考えて制作する」段階まで取り組みやすいからである。

2つ目はテキストプログラミングの最初は、短い「写経」からでもいいと考える。最初はどうしても入力ミスがつきものであり、短いプログラムなら生徒も教員も修正しやすい。生徒はそのまま入力する「写経」であっても、うまくいけば達成感を感じていたので、最初はこの程度にとどめておき、慣れてからシミュレーションのところで長めのプログラミングに取り組みさせる方がスムーズである。

3つ目はモノを動かすプログラミングは面白いということである。画面上で何か動くよりも、ロボットなどモノが動く方が、生徒のモチベーションが上がり、自然に試行錯誤するようになり、問題解決に取り組ませやすいからである。

情報Iは2単位であり、その中でプログラミングのところに配当できる時間は1/4程度、実質的には12時間前後ではないかと考え、今回の授業を実践してみた。全国高等学校情報教育研究会で分科会発表をしたところ、いろいろなアドバイスもいただいたので、引き続き改善を重ね、今後も生徒にとって興味深く役に立つ授業を実践していきたい。

最後に本授業のスライド・ワークシートは、以下の私のWebサイトにて公開している。

「情報科の授業アイデア」

(<https://www.okamon.jp>)



(2025年8月25日受付)



岡本弘之
okamoto@assumption.ed.jp

アサンプション国際中学校高等学校情報科・社会科・探究科教諭、大阪私学教育情報化研究会事務局、情報科の教科書・参考書・教材執筆など。