

HSPはプログラミング授業の 新しい選択肢

神奈川県立厚木西高等学校 梁取 新平 先生

科目：社会と情報（1年生 2単位）
内容：プログラミング
クラス：7クラス 各37名
時間：10時間
時期：1月～2月

■ 1 ねらい

■実践のねらい

本校は、神奈川の県央部に位置する全日制普通科の学校で、情報科では、1学年に必修修科目として「社会と情報」を2単位、3年生の必修選択科目に「情報の科学」を2単位設置している。小・中学校におけるプログラミング教育の影響もあり、「社会と情報」の授業でプログラミングを学びたいという意欲をもった生徒が多いのが近年の傾向である。2018（平成30）年3月告示の「高等学校学習指導要領情報科」において、情報科の必修科目が「情報I」となり、プログラミングが大きな単元として設定されることを踏まえ、現在さまざまなプログラミング環境を授業で導入し、試行を繰り返している。ここ2年ほどは、年間のまとめの時期である3学期に、HSP（おにたま氏制作・Hot Soup Processor）を利用させていただき、簡単なアプリ（ソフトウェア）作りにチャレンジしている。

「社会と情報」や「情報I」でプログラミングを学ばせる目的は、プログラマーを育てることではなく、プログラミングでアプリ作りを通して、コンピュータやスマートフォンが動作する仕組みに関心をもたせて生徒の創造性を広げ、かつ、自分以外の人を使うことを考えたユーザインタフェースをデザインすることの重要性に気づかせることだと考えている。生徒自身がコンピュータやスマートフォンでやりたいことや課題を見つけ、それを実現するための手段・ツールとして気軽にプログラミングに取り組めるようになることが将来的なねらいである。

数あるプログラミング言語の中で、高校段階の実習用言語としてHSPを選んだ理由は、HSPが以下のような特長をもっているからである。

特長① 導入が簡単である。インストールが不要で、レジストリを書き換えない。動作が軽い。

特長② インタプリタ型である。授業で作成するような数十行から数百行程度のアプリであれば処理速度は全く気にならない。

特長③ スクリプトエディタが付属している。小中学校ではScratchなどのブロック型やDolittleなどの日本語でのプログラミングが多いので、高校では、より本物のプログラミング作業に近づけるため、英語によるコーディングにチャレンジさせたい。付属のスクリプトエディタはデバッグがF5、ヘルプがF1を押すだけで簡単である。初学者向けなので、命令や使い方がかなりシンプルな環境である。

特長④ 画像や音声ファイルを扱える。HSPはBASICライクな言語であるが、画像や音声の読み込み、表示、再生などに強いので、ビジュアルを意識したアプリ制作が可能である。

特長⑤ 実行ファイルを出力できる。アプリに使用する画像素材や音声素材もまとめてパッケージングして1つの実行ファイル（EXEファイル）に書き出すことができるので、課題として集める時も扱いやすく、生徒同士の相互評価の活動などもさせやすい。

■全体のカリキュラムの中での位置づけ

本校の「社会と情報」では、1,2学期は教科書に沿って授業を行い、以下の①から⑦のようなスキルを実習の中で生徒に身につけさせている。このスキルが十分身についているかどうかで、プログラミングの授業の進めやすさが変わってくるので、できるだけ丁寧に指導している。

- ①英字、記号を含め、正確な文字入力ができる。
- ②全角文字と半角文字の違いが見分けられる。
- ③画像ファイルを保存したり、簡単な加工をしたりできる（大きさの変更、背景の透明化、適切なファイル形式での保存）。
- ④音声ファイルを取り扱うことができる。
- ⑤ファイル名の拡張子から、ファイル形式を判別することができる。
- ⑥自分のアイデア・イメージ・考えをわかりやすく他人に伝えられる。
- ⑦他人のアイデア・イメージ・考えに耳を傾けて理

解しようとする努力し、他人と共通認識を作ることができる。

⑥や⑦はコンピュータの操作スキルではないが、生徒がグループ単位でアプリの設計、コーディング、デバッグ作業、相互評価を行う上で欠かせないスキルとなるので、さまざまな単元の実習の中で生徒に意識させている。そのために、パソコン教室の生徒机には、サインペンやメモ用紙をあらかじめ準備しておき、いつでも利用できるようにしている。

その上で、1年間の集大成かつ課題解決学習の一環として、プログラミング体験・アプリ制作実習に全員取り組ませている。年度によって配当時間は変えているが、下記に10時間配当の例を挙げる。

■授業計画

①配当時間

3学期の10時間（1月～2月）

②時間ごとの学習目標と内容

- ・1時限：目標・ねらい・評価方法の説明，HSPとエディタの操作，デバッグ，文字の表示
- ・2時限：座標（X-Y座標による指定），色（光の三原色による指定），図形の作成
- ・3時限：画像ファイルの表示（背景とキャラクタ），音声ファイルの読み込みと再生
- ・4時限：3大構造（順次構造・反復構造・分岐構造）の理解と実際のコーディング，変数の理解，誕生月占いアプリの制作
- ・5時限：画像ファイルと反復構造，分岐構造を組み合わせて，キー入力に従いキャラクタが移動するアプリを制作
- ・6時限：乱数の理解，画像・乱数・分岐・反復を用いて10連ガチャのアプリを制作（ここでソーシャルゲームの注意にも触れる）
- ・7～9時限：オリジナルのアプリ（ゲーム）のプログラミング，コードと素材ファイルを1つの実行ファイルに変換して提出フォルダに提出
- ・10時限：クラス内で，生徒が作ったアプリ作品を実際に動かし，相互評価を行う

③指導上の留意点

実習を行うにあたり，以下のような点を工夫した。

- ・プログラミングの基本の学習はサンプルに手を入れながら学ぶ形にした。タイピングが苦手な生徒が多いので，コードをゼロから入力するのではなく，重要な箇所を■に置き換えたサンプルを配布

して書き直させたり，コメントアウトを解除させながら説明したりするなどして授業進行をそろえる工夫を行った。

- ・オリジナルアプリの制作は，単独または2人組のどちらかで行うことを生徒に選択させた。多様な生徒の希望に配慮する形となった。
- ・オリジナルのアプリは，横320ピクセル×縦480ピクセルで統一することで，スマートフォンでの使用をイメージさせた。生徒にとって，パソコンのソフトよりスマートフォンのアプリのほうがよりイメージしやすいと考えたためである。なお，HSPは，付属の3Dishというプラグインを利用することで，作成したソフトをスマートフォン（Android）用に変換することも可能である（授業では紹介のみ）。
- ・オリジナルのアプリの種類は，
 - ①日常生活に役立つアプリ
 - ②他の生徒を楽しませるアプリのどちらかを選択させた。
- ・オリジナルのアプリの制作方法は，ゼロからコーディングするか，サンプルを改造するかを生徒に選択させた。生徒自身が各機能の理解度やスキル，イメージした目標に応じて選べるようにした。
- ・授業で説明していない発展的な機能を用いるのは，生徒自身が調べて扱える範囲内に限定した（教員側も全ての機能を説明できないため）。

■ 2 準備

■ 必要なハードウェア

- WindowsOS のパソコン（スペックが低いパソコンでも十分に実習ができる）

■ 必要なソフトウェア

- HSP スクリプトエディタ（執筆時点での最新安定版は 3.5）。インストール不要でフリーで使用できる。エディタ本体はフォルダ内の「hsed3.exe」
- 画像ファイルを加工する場合はペイントソフトやフォトタッチソフト。画像のトリミングやリサイズ、背景の透明化に対応したもの。
- ファイル名の拡張子をあらかじめ表示しておくようにする。

■ 必要な素材、材料

- 素材用画像ファイルとして学校のマスコットキャラの画像や学校近隣の自治体のゆるキャラなどの画像、アプリの背景用の画像などを PNG ファイルで準備した（透明色が利用できるため）。
- 効果音や BGM 用の音声ファイルを WAV ファイルでいくつか準備した。MP3 や MID も使用可能。

■ 3 実践内容

■ 1～3 時限

この時間の目標は、生徒がコーディングとデバッグを繰り返しながら、アプリの土台にさまざまな形式の情報を表示できるようになることである。

アプリの土台の設定なら

```
screen 0,320,480 (ウィンドウサイズ)
title "タイトル" (アプリのタイトル)
```

で設定できる。

文字列の表示は、

```
mes "Hello World!"
```

で行える。日本語の表示も可能である。

処理を一定時間中断するには、

```
wait 100 (単位は 10 ミリ秒)
```

で行えるので、この 3 つの命令だけでもノベルゲームが作成できる。

図形の作成は、

```
pset 10,10 (点)
line 120,80,300,400 (直線)
boxf 200,100,250,200 (円)
circle 220,220,300,400 (矩形)
```

と、始点や終点を座標で指定することで、好きな大きさの図形を作成できる。

画像の表示は、

```
picload "haikai.jpg" (初期表示)
picload "chara.gif",1 (重ねて表示)
```

などの命令で、用途に応じて使い分ける。

■ 4～6 時限

プログラミングの 3 大構造や変数、乱数などの実践的な機能を紹介する。

キャラクターが動くアプリの場合、HSP では、セルアニメーションと似た原理で、キャラクターが動くたびに背景画像を新たに貼って、少しずれた場所にキャラクターを貼ることを繰り返すような処理をしている。これには反復処理が必要で、

```
repeat 回数
    背景を貼る処理
    キャラクターを貼る処理
    wait
loop
```

と記述することで実現できる。反復回数を設定しない場合、キー入力などの何らかのアクションを待ち続ける処理になる。

変数や乱数は、

```
randomize
saikoro=rnd(5)+1
mes "サイコロの目は " + saikoro
```

のような電子サイコロのサンプルで紹介した。

分岐処理と乱数を組み合わせると、スマートフォンのソーシャルゲームで多く見られる「ガチャ」を作成できる。

<簡単なガチャシステム>

```
title "ガチャシステム"  
kane=0  
mes "ガチャ1回120円."  
button "引く",*kekka  
stop  
  
*kekka  
  randomize  
  a=rnd(10)  
  if a==0{  
    color 255,0,0  
    mes "レアです!"  
  }  
  else{  
    color 0,0,0  
    mes "ノーマルです"  
  }  
  kane=kane+120  
  mes "使った金額は"+kane+"円"  
  stop
```

このガチャシステムを紹介すると、多くの生徒が画像や音声を出すように改良したり、確率を操作して当たりが出ないようにしたりと、アプリを自分で作っている実感を持ち始めるようになる。併せて、教員からスマートフォンの適正な利用についての話をし、ソーシャルゲーム依存にならないよう注意も行うと効果がある。

■ 7～9 時限

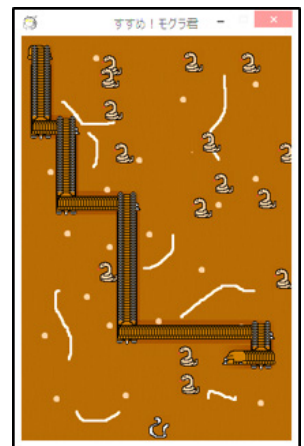
これまで学んだ内容やサンプルを参考にして、生徒にオリジナルのアプリを作成するよう指示する。ただし、授業で扱ったサンプルだけでは、アイデアがなかなか出ないので、呼び水代わりに教員も様々なサンプルを作成して生徒に紹介した。この実践の際に教員が作成して生徒に例示したサンプルは次のようなものである。

- ① 星空アプリ (図形, 位置, 乱数)
- ② 数当てゲーム (文字入力, 乱数, 分岐処理)
- ③ じゃんけんゲーム (変数, 乱数, 分岐処理)
- ④ クリックで座標表示 (位置, 変数, キー入力)
- ⑤ カラーコード (色情報, 変数, ボタン)
- ⑥ 当たり判定 (情報取得, 分岐処理, キー入力)
- ⑦ スクリーンショット (画像, ファイル保存)

いずれも HSP の基本命令だけで作成できるものである。生徒には、課題を作成する際、1 からコーディングしても、サンプルを改造してもよいと指示した。結果的には、80%程度の生徒がサンプルの改造を選択したが、20%程度の生徒は、一部サンプルからのコピーペーストがあるものの、自分で1からアプリを作成することができた。完成したコードを実行ファイルに変換する際にやや教員の手助けが必要な生徒もいたが、全員が期日までに課題のプログラムを提出することができた。

<生徒の作品の例>

- ・モグラがエサを探して地下に潜るアプリ
- ・入力した文章を保存するアプリ
- ・ランダムな3文字の組み合わせを作るアプリ
- ・分岐で結末が変わるノベル(読み物)アプリ
- ・歴史人物占いアプリ
- ・ギターコードの押さえ方を表示するアプリ
- ・タイピング練習用アプリ
- ・時間を知らせるタイマーアプリ
- ・この人は誰だ!クイズが楽しめるアプリ



▲モグラを操作してエサを探すアプリ

■ 10 時限

最後には、提出された課題を共有フォルダに入れて、作成者を伏せたまま、クラス全員で課題プログラムを実行させ、相互評価をさせた。おもに、

- ① アプリとして起動し、必要な動作をするか
- ② 画面構成や使い方の説明など、利用者の目線での工夫がなされているか (ユーザビリティ)
- ③ アプリとしての目新しさや斬新さを基準に評価させた。

■ 4 結果と反応

■ 成果と学習者の反応

最後に、生徒たちに自己評価として、プログラミングでアプリを作成した感想を書かせたところ、次のような回答が多かった。

- ・面白い。もっと本格的なアプリを作りたい。
- ・スマートフォンでも遊べるようにしたい。
- ・ダウンロードできるようにして、他の友達にもプレイしてもらいたい。
- ・将来、販売できるようなものを作りたい。
- ・プログラミングはかなり難しかったが、結果が目に見えるのは楽しかった。
- ・たった1文字の間違いでうまく動かないのが難しいと思った。

プログラミングは難しいと回答する生徒が多かったが、将来的な可能性を感じた生徒も多かったことがわかる。当初、生徒が作るアプリはゲームタイプのものが多いのではと予想していたが、メモアプリや自分の趣味に活かせるアプリなど、幅広いジャンルの課題が出来上がったことは大きな驚きだった。プログラミングを単なるゲーム作りで終わらせず、課題解決の手段であることを一部の生徒には実感させることができたと思う。

■ 5 まとめ

「情報 I」のスタートを見据えたプログラミング教材の開発は、全国の情報科の先生方が四苦八苦している最中である。そもそもプログラミングを通して何を学ばせるべきなのか、一言で断じるのは難しい。プログラミング言語の選択ひとつ取っても、それぞれの言語には強みもあれば弱みもあり、今回紹介した HSP もあくまでその一例に過ぎない。

今後発展させたい単元としては、HSP でチャットアプリを作成させて、情報通信ネットワークの原理を理解させるなどの方法も試してみたいと考えている。HSP を活用したプログラミングの授業実践は 3 年程度続けているが、毎年、生徒のアイデアの斬新さには感心させられる。今後も生徒の創造性を伸ばせるようなプログラミング授業を考えていきたい。

■ 6 参考文献等

- ・おにたま・悠黒喧史・うすあじ『はじめてのプログラミング HSP3.4+3Dish 超入門』秀和システム (2014)
- ・山田友梨『「CuteHSP」ではじめるプログラミング入門』工学社 (2017)