

授業で活用するアプリ開発と 教員のプログラミングスキル向上

兵庫県立神戸甲北高等学校 主幹教諭 教務部長
Microsoft MVP Data Platform 松本 吉生

1. はじめに

筆者は、平成 15 年の教科「情報」の出発点において、プログラミング教育に焦点が当たらなかったことに強い不満と危機感を持っていた。時を経て、平成 25 年 6 月に第 2 次安倍内閣による成長戦略「日本再興戦略－JAPAN is BACK－」¹⁾が閣議決定され、産業競争力の源泉となるハイレベルな IT 人材の育成・確保のために、義務教育段階からのプログラミング教育等の IT 教育を推進することが提言された。平成 28 年 6 月 16 日には、文部科学省「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」が議論の取りまとめを行い²⁾、学校教育としてプログラミング教育を進める方向が示された。そして平成 29 年 3 月に告示された小学校学習指導要領において、プログラミング教育を必修化することが示された。この頃から新聞テレビなどマスコミやインターネットの記事でプログラミング教育を取り上げるものが多く見られるようになった。

学習指導要領の告示により、小学校教員の間でプログラミング教育に対する関心が高まり、様々な実践報告、授業研究、教材研究が行われるようになった。しかし、高等学校の教員はプログラミング教育に対する関心が低いように思われたため、筆者は平成 29 年度に兵庫県教育研究会情報部会のグループ研究で「プログラミング教育に関するアンケート調査」を行い、兵庫県内高等学校における情報科教員のプログラミング教育に対する意識とスキルを調査した。³⁾この調査では、普通教科「情報」の授業でプログラミング実習は必要だと思うか、の問いで「思う」の回答が 68.2% でしかなく、自由記述で求めた回答の理由では「情報教育の根幹である」「これからの社会背景上必要」という積極的な意見から「全員がプログラミングを学ぶ必要を感じない」といった意見など大きな温度差があることがわかった。

また、実践されているプログラミング教育の中には、Excel の VBA やスクラッチ、HTML など純粋なプログラミング言語ではないものが報告され、プログラミングの概念が整理されていないことがうかがえた。HTML 言語がプログラミング言語でないことはもとより、Excel の VBA やスクラッチも純粋な意味でプログラミングではない。コンピュータプログラムはコンピュータに対する命令を記述したものであり、汎用的なハードウェアを制御するプログラムを作ることがプログラミングの本質である。その意味で、Excel の VBA は表計算ソフトウェアのオートメーション機能であり、スクラッチはプログラミングの学習環境である。

これらのことから、高等学校教員がプログラミングに対する理解を深め、スキルを向上させるには、自ら役に立つプログラミングを経験することが良いと考え、平成 31 年度に兵庫県教育研究会情報部会⁴⁾で「学力向上のためのアプリの開発と授業」のテーマでグループ研究を行った。本稿ではこの研究についてアプリ開発の実際を中心に報告する。

2. 研究の目的

この研究の目的は、高等学校の情報科教員がプログラミングによってアプリを開発し役立てることに、より、プログラミングに対する理解を深めスキルを向上させることである。校務にアプリを活用することも考えられるが、もとより教員の本務は授業にあるので、授業で活用するためのアプリを開発の対象とした。授業でアプリを効果的に活用することは、生徒の学習に役立つものであるはずだから、学力向上につながるものである。そのためにはアプリの特性を活かした授業展開も必要と考え、授業のデザインも検討した。同時に、作成に手間がかかりすぎるものは教材開発として現実的でないため、短時間で作成できるものを検討した。

3. 学習テーマとアプリの検討

研究グループでは、各校で使用している「社会と情報」または「情報の科学」の教科書を持ち寄り、学習内容のうちアプリを使うことが効果的であるものを検討した。概念を視覚化できるもの、条件を変えて試行錯誤が体験できるもの、繰り返し学習に役立つもの、などが効果的と考えられ、そのうち高度な技術を要するものや開発に手間がかかりすぎるものを除き、「ランレングス圧縮」「2進数の基礎、ビットとバイト」「2, 10, 16進数と2進数の加算」「文字コード」「光の三原色と色の三原色」の5つについてアプリを開発して授業を行った。

4. アプリの開発手法

アプリ開発は Microsoft Visual Studio 2019 Community により、言語を C# として開発した (Fig.1)。Visual Studio を選んだのは、無料で使える Community 版があること、作ったアプリの実行が一般的に各校で導入されている Windows OS で簡単なこと、効率よく開発できる統合開発環境が充実していることである。



Fig.1 Visual Studio による開発画面

Visual Studio による開発では、メニューの「デバッグ」から「デバッグなしで開始」を実行するか、メニューバーの緑の三角アイコン「開始」をクリックすることでビルドと実行ができる。ビルドに成功すると、「ドキュメント」フォルダの下にある Visual Studio のプロジェクトファイルを保存する source\repos フォルダに実行ファイルが生成する。このとき、source フォルダは非表示になっていることがあり、その場合は、たとえばユーザー名が「yomatsumoto」のとき、次のようにフォルダを直接開く必要がある。

```
C:\Users\yomatsumoto\source\repos
```

「ドキュメント」フォルダ下の source\repos フォルダには、プロジェクトごとのフォルダが生成している (Fig.2)。デバッグ時に生成した実行ファイルは、プロジェクト名のフォルダ下の \bin\Debug フォルダに生成している (Fig.3)。授業で生徒に実行させるには、コンピュータ教室で利用できる共有フォルダを用意し、そこに実行ファイルをコピーしておけばよい。



Fig.2 source\repos フォルダ

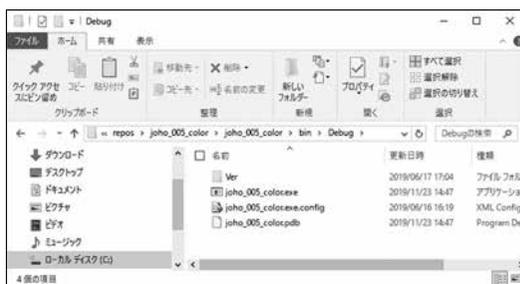


Fig.3 Debug フォルダ

5. 作成したアプリ

(ア) ランレングス圧縮

ランレングス圧縮を学べるアプリには2つのテキストボックスと2つのボタン、1つのラベルがある (Fig.4)。最初のテキストボックスにいくつかの連続するアルファベットを入力し「圧縮する」のボタンをクリックすると、ランレングス圧縮されたデータが次のテキストボックスに表示される。次のボタンをクリックすると圧縮率をラベルに表示する。このアプリでは、文字列中の連続するアルファベットを数えて返す Compress メソッドを作り、ボタンクリックで文字列を引数に入れて次のように実行する。

```
textBox2.Text = Compress(textBox1.Text);
```

圧縮率の計算は、圧縮後の文字数を最初の文字列の文字数で割って求める calcCompressRate メソッドで行う。

```
private float calcCompressRate(int a,int b)
{
    float answerRate = (float)a / (float)b * 100;
    return answerRate;
}
```



Fig.4 ランレングス圧縮アプリ

(イ) 2進数の基礎, ビットとバイト

このアプリでは, 10進数を2進数に変換するボタンを用意するとともに, 数値を1ずつ増やすボタンと減らすボタンを置いた(Fig.5)。これにより2進数の各桁の重みが10進数で理解できるようにした。10進数を2進数に変換するコードは次の1行でよい。

```
label4.Text = Convert.ToString(myNumber, 2);
```

また1時間の授業でビットとバイト, 接頭辞の概念も教えることとしたので, ビットとバイトの変換, バイトとキロバイトの変換もできるようにした。



Fig.5 2進数の基礎, ビットとバイトアプリ

(ウ) 2, 10, 16進数と2進数の加算

コンピュータ内部で数字は2進数で扱われ, 人間は日常的に10進数で数える。そのうえで, なぜ16進数表現が重要なのかを理解するために, 2進数の各桁と10進数, 16進数の関係がわかるアプリを開発した(Fig.6)。このアプリでは, 2進数の各桁をチェックボックスで加えることができるようにし,

数値を2進数と10進数, 16進数で表示させた。生徒は2進数の各桁をチェックすることで, 各桁の重みが10進数のどの値に対応するか, また2進数と16進数を比較することで, 2進数の4桁が16進数の1桁に相当し, 数値を16進数で表現する利点を実感できるようにした。

またこのアプリを使う授業でも, 1時間の授業で2進数の加算も教えることにしたので, アプリ上で確かめることができるようにした。このアプリでも2進数の値を増減できるボタンを配置し, 自分で試しながら2進数の桁上りが理解できるようにした。



Fig.6 2, 10, 16進数と2進数の加算アプリ

(エ) 文字コード

多くの教科書で文字コード表が示されており, このアプリでは実際に文字がコンピュータ内部でどのようなコードになっているかを表示できるようにした(Fig.7)。テキストボックスに任意の文字を入力し, 文字コードに変換して教科書の文字コード表と対照して確かめることができる。異なる体系のコードはインターネットで確かめさせる。C#は文字コードを変換するメソッドが豊富に用意されており, Encoding オブジェクトに実装されている。メソッドの引数を変えることで各コード体系の文字コードを取得できる。

```
Encoding.GetEncoding("iso-2022-jp");
Encoding.GetEncoding("SHIFT-JIS");
Encoding.GetEncoding("euc-jp");
Encoding.GetEncoding("unicodeFFFE");
```



Fig.7 文字コードアプリ

(オ) 光の三原色と色の三原色

RGB 光の三原色を組み合わせることで色を表示するアプリは様々なものがあり、Windows の「ペイント」でも「色の編集」で体験することができる。しかし既存のアプリを授業で使わせる場合には、操作方法を説明しなければならない。アプリを開発すれば使い方を説明することなく、目的的操作に集中させることができる。またこのアプリでは、すでに学習した 2 進数や 16 進数と 10 進数の関係も整理できるように、各色の値を 3 つの数値表現で表示するようにした (Fig.8)。

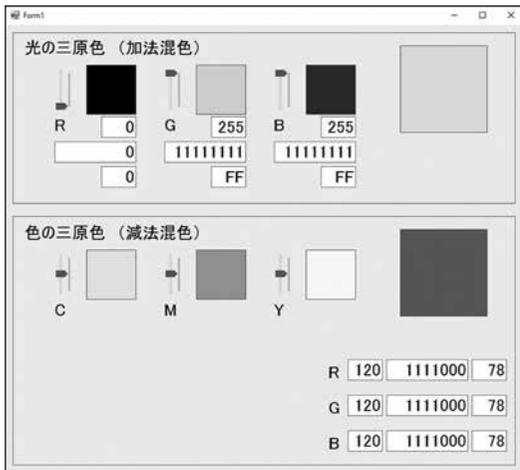


Fig.8 光の三原色と色の三原色アプリ

6. アプリを開発するメリット

これらのアプリは、いずれも 1 時間程度で開発できるものである。研究会ではアプリ開発の経験がない先生にもプログラミングを体験してもらい、今日的なプログラミング開発環境での開発の簡単さを実感してもらった。既存のアプリを利用するのではなく独自のアプリを開発するメリットは、生徒が操作しやすく、教師が意図した実習の目的に集中できるものが作れること、体験的に理解させたい概念を明確に作りこむことができること、1 時間の授業をデ

ザインしたときに、アプリを切り替えることなく時間中に体験する機能を組み合わせることができることがある。

7. おわりに

プログラミングには経験が必要だが、ここで紹介した程度のアプリを作るのは難しくない。思えば筆者が学生の頃、コンピュータは好事家のおもちゃだったが、今やプログラミングが一部の技術者による特殊技能である時代は終わり、表計算ソフトやワープロソフトを使いこなすように、一般のデスクワーカーがプログラミングをする「市民開発者」の時代がやってきた。

コンピュータに関することに限らず、技術には進歩の節目がある。長い歴史のある Visual Studio も日々進化してきたが、プログラミングの世界は、今まさに大きなトレンドを迎えようとしている。クラウドのデータ処理や AI の利用が簡単にできる Microsoft Power Apps はその先鋒といえ、目から鱗が落ちるほど簡単に、ほとんどコードを書くことなくアプリを作ることができる。

新しい技術はとっつきにくいものだが、その時代に求められる必要なことが、より簡単にできるようにデザインされているものだ。高等学校の情報科教員は、常に最新の技術に貪欲であり、チャレンジする姿勢を持ち、自らが積極的に体験することによってこそ、生徒にプログラミングの価値や面白さを伝えることができるだろう。

参考文献

- 1) 内閣府「日本再興戦略 -JAPAN is BACK-」, 平成 25 年 6 月 14 日
https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/saikou_jpn.pdf
- 2) 小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」, 平成 28 年 6 月 16 日
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/houkoku/1372522.htm
- 3) 松本吉生「プログラミング教育に関するアンケート調査の結果とプログラミング教育の課題」, i-Net 第 53 号, 数研出版, 令和元年 5 月
<https://www.chart.co.jp/subject/joho/inet/inet53/inet53-1.pdf>
- 4) 「兵庫県高等学校教育研究会情報部会」
<http://www.hyogo-c.ed.jp/~johobukai/>