

スマートフォンのアプリ開発の授業について

秋田県立仁賀保高等学校教諭 早藤 素史

1. はじめに

秋田県立仁賀保高等学校の情報メディア科では、平成25年度からの新教育課程実施にあたり、ネットワークの学習を中心としたNetシステムコースの教育課程を見直し、プログラムに関する学習についても力を入れることにした。この学習に関しては、ただパソコンでコードを書いて終わるだけではあまり意味がなく、もっと生徒に興味を持って取り組めるものはないかと考えたとき、生徒が常に身につけているスマートフォンのアプリ開発をさせてみるのはどうだろうかと考えた。

通常Androidアプリの作成・開発を行うにはEclipseというツールを使用し、プログラミング言語はJavaで開発する。ネットや本の情報では「アプリ開発は無料で簡単にできる」などと甘い言葉が書かれてあるが、実際にやってみると、動くといわれるコードをそのまま記入しても動かず、アプリ開発に関する様々な知識が必要であることがわかった。このままでは授業で使うことは無理である。何か手はないものかと調べたところ、AppInventorという、コードを入力せず、部品を組み合わせでAndroidアプリが開発できるツールがあることを知った。早速試してみると生徒も簡単にできそうだったので、これを利用することにした。

2. AppInventorについて

(1) AppInventorとは何か

AppInventorは、Webブラウザ上で画面をデザインし、Javaアプリ上でパズルのような部品を

組み合わせてイベントを組み立て、Androidアプリを作成するため、コードを書き込むことがないツールである。まさに、ブロックを組み合わせるだけで、スマートフォンのアプリが作れる。

このツールは2011年までは、Googleが提供していたが、現在はマサチューセッツ工科大学(MIT)が提供しているため、現在の正式名称はMIT App Inventorとなっており、MITのサーバで管理されている。そもそも教育向けに開発されているものであるため、プログラムの構造や基礎を理解するのに適している。また、GoogleはAppInventorのソースコードをオープンソースにしているため、例えば、校内にWebサーバを立て、それを利用してAppInventorを使用できるようにすることも可能である。

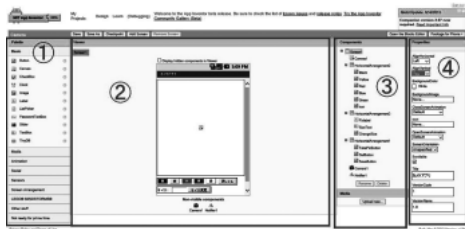
アプリ開発には非常に便利なAppInventorであるが、MITのサーバではベータ版として運用されているため、たまに接続が不安定になることもある。また、画面サイズが320×430で固定されているため、大型タブレットでは、表示がうまくいかないなど、機能が限定的である。しかし、実際にアプリ作成をしてみると、タッチセンサー、GPS、加速度センサー、音声マイク、カメラ機能など、スマートフォンの主たる機能は一通り使用できるようになっており、授業で使用するには十分である。

このツールは、プログラムの基礎を学び、アプリ作成の基礎を学ぶものである。限界を感じ、もっと高度なことをやりたくなったときには、Eclipseを使用することを紹介し、自学自習をすすめていくのがよいのではないかと感じている。

(2) AppInventorでの大まかなアプリ作成の手順について

さて、AppInventorによるアプリ作成のしかたを簡単に紹介する。

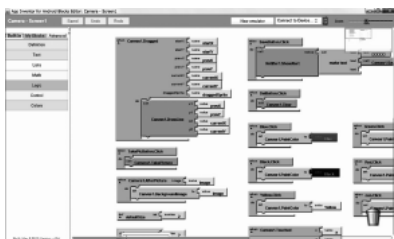
・デザイナーによるアプリ画面の作成



まず、URLで<http://appinventor.mit.edu/>にアクセスし、Inventのボタンを押すと、上図のような「デザイナー」という画面が表示される。中央の②の部分がViewerといい、スマートフォンのアプリの画面を作る部分である。これに①のPaletteから、「ボタン」や「テキストボックス」「ラベル」といった必要な部品（コンポーネント）をドラッグして置く。また、置いた部品は③のComponentsで確認し、さらに④のプロパティ画面で部品の表示やサイズなどを変更する。

・ブロックエディタでプログラム作成

デザイナー画面の上部のOpen the Blocks Editorのボタンを押すと、次のようなブロックエディタの画面が表示される。



左側にある「Built-in」と「My Blocks」等のタグにあるボタンを使い分けながら、アプリの動作をプログラムしていく画面である。

・スマートフォン（実機）によるエミュレート

作ったアプリの動作確認は、USBケーブルで接続したスマートフォン（実機）でエミュレートすることによって、動きを確認することができる。AppInventor上でも、ソフトで動作をするエミュレータを用意しているが、起動が遅いしパソコン

そのものに、センサーやカメラなどがついていないと動作確認ができないため、実機で動作確認をした方がよいようである。

・AppInventorを使う利点

AppInventorを使う利点は、生徒の授業の進度が揃いやすいことにある。コードを打ち込まないことにより、キーボード操作が速い生徒と遅い生徒の差がなくなり、アルゴリズムやプログラムの説明にきちんと時間を使えることが非常に魅力的である。また、エラーがあってもビジュアル的に見やすいため、エラー部分を探しやすく、そういったところも指導がしやすい。

3. AppInventorを用いた授業例

AppInventorに関する書籍が少ないため、授業の教材は、それらの本とネット上の情報、さらに私が作るオリジナルのアプリを使用するなど、独自のプリント教材を作成し授業を行っている。

プリントには、デザイナーに配置するコンポーネントとそのプロパティに関する情報及び、ブロックエディタで作成するブロックが記載されており、生徒は、それとアプリの画面を参考にしながら作成する。デザイナーに関しては、2年生の初心者でもほとんど説明なしにアプリ画面を作っている。コンポーネントの場所やコンポーネントを置く順番がわからないことはあるようだが、全くできない生徒はいない。



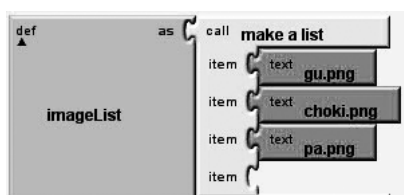
「じゃんけんゲーム」（上図）を例にとって説明する。これは、どのプログラム言語でも基本的なものとして扱われるため比較しやすい。特に、IFに関するアルゴリズムに関しては、生徒にとっても理解しやすいものである。

まず、デザイナー画面には、「Label」、「Image」、

「Button」などを図のように配置をする。ブロックエディタに関しても、動作を確認しながら作成する。コンピュータの手を表す変数と自分の手を表す変数は、Variableのブロックの名前を変更することによって作ることができる。



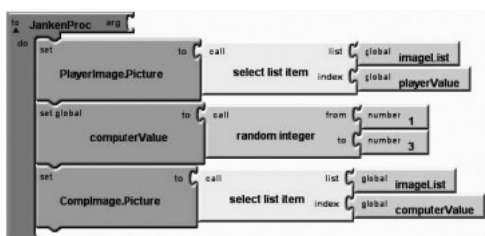
コンピュータの手も自分の手も、グー=1、チョキ=2、パー=3と決めておく。じゃんけんで使用する画像ファイルはサーバにアップロードしておき、ファイル名をリスト化して番号でファイルが呼び出せるようにしておく。



「グー」のボタンを押した場合の処理は、



となる。GuButton.Clickとなっているブロックを探して置くだけで、あとはその中にボタンが押されたときに処理するブロックを配置すればよい。JankenProcは、プロシージャであり、「チョキ」や「パー」のボタンを押したときにも同じ処理がされるので、一つのまとめたプログラムとして用意している。そのプロシージャの一部は次図のようなブロックでできている。

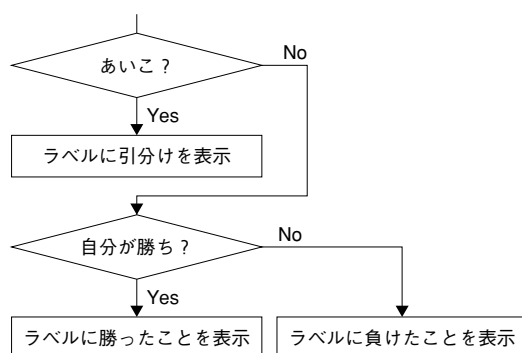


プログラムを作成する際に一番難しいのは、IF文に関するところである。今回のじゃんけんの

ここでは、「コンピュータの手」と「自分の手」に関して、次のような表を作らせた。

		コンピュータの手 computerValue		
		1 (グー)	2 (チョキ)	3 (パー)
自分の手 playerValue	1 (グー)	引き分け (あいこ)	勝ち	負け
	2 (チョキ)	負け	引き分け (あいこ)	勝ち
	3 (パー)	勝ち	負け	引き分け (あいこ)

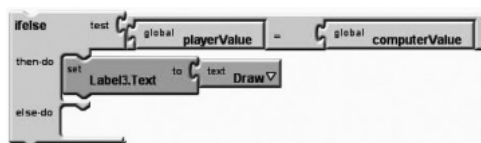
これにより、じゃんけんの勝ち負けのアルゴリズムをどのようなフローチャートにするかを考えさせた。いろいろな方法があるが、今回は、フローチャートとして次のようなものを提示した。



次は、条件について考えさせる。「あいこ」の条件は生徒もすぐに理解できた。

playerValue = ComputerValue

これをAppInventorのブロックで表示させると次のようになる。



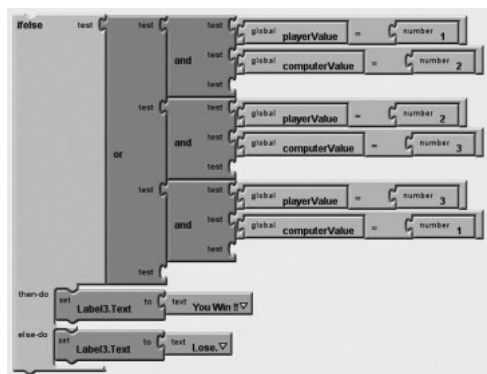
空欄の「else do」には、次の条件「自分が勝ち？」という部分から先の部分が入る。「自分が勝ち？」の条件は、生徒もかなり悩んでいた。今回は、自分が勝つためには3パターンあることを考えさせ、論理和 (or) と論理積 (and) を組み合わせればできることを理解させ、それを条件とさせた。

1. playerValue = 1 かつ computerValue = 2 の場合
2. playerValue = 2 かつ computerValue = 3

の場合

3. playerValue = 3 かつ computerValue = 1
の場合

1～3のいずれか一つが「true」の場合、「自分が勝ち」となる。これを、AppInventorのブロックで表示すると次のようになる。



最後は、実機でエミュレートさせ、エラーが出たり動かない部分は直させた。

4. 終わりに

今年度の2年生は、好奇心が旺盛でどんどんアプリを作ってしまうため、4月と5月のたった2か月ながら、「基本的な作り方の説明」「カメラアプリ」「音声検索ツール」「大小ゲーム」「電子さいころ」「じゃんけんゲーム」とかなりのハイペースでアプリ作成の授業が行われている。しかも、生徒の反応が非常によい。



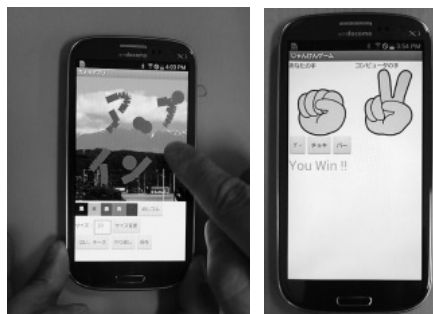
例えば、「カメラアプリ」は、スマートフォンで写真を撮り、撮った写真に指でなぞって落書きをして保存するというアプリだが、できあがると生徒から、「アンドロ機能はないのか?」「消しゴム機能はないのか?」などいろいろな質問が出た。スマートフォンの機能に関するものもあるためすべてはできないが、一つ一つ説明したとこ

ろ、それなりのアプリができあがっており、生徒も喜んでいた。他の授業では無反応なことが多いため、この反応には驚いた。

「大小ゲーム」は、次に出る数字が7よりも大きいかわ小さいかを当てるだけの単純なゲームだが、連勝に関する機能や音ファイルをつけたり、画面の色などデザインをいろいろ変更できることを教えると、かなりいろいろな画面ができあがっていった。

「じゃんけんゲーム」では、リスト機能に応用し、勝ったり負けたりした時のメッセージを数パターン表示できるようにすることに挑戦させてみた。「5回勝負させるにはどうすればいいですか?」という質問が出たが、別の生徒が「大小ゲームで続けてゲームができるようになっていたけどそれを応用すればできますよね」と答えるなど、今までとは違う生徒の反応に驚いた。

プログラムに関する学習は、クリエイティブな能力を鍛えさせる場である。そのために必要なことは、すべてを教えず、自分で考えて工夫しながら作らせることが重要である。3年生には、「シューティングゲーム」や「ブロック崩し」のベースとなる部分を教えている。そこから応用した完成品を作ってもらいたいと思い、いろいろとヒントを与えながら作らせようとしているが、なかなか難しいようである。スマートフォンは身近にあるため、授業と合わせて使っているうちに「これをこうやれば」とか、「このプログラムはあそこで使える」といった発想をする生徒が出てくることを期待している。ほんの少しでも、生徒がオリジナルのアプリを作れるようにさせたいと考えながら授業をしている。



アプリの完成画面