1時間で学ぶソフトウェアの仕組み

RIGHT:兼宗 進（大阪電気通信大学）2009年11月6日

はじめに

プログラミングは、うまく使うと生徒の集中力や考える力を伸ばします。そして、身の回りの多くのソフトウェアがどのような仕組みで動いているのかを理解できるようになります。

ソフトウェアの仕組みの体験的な学習は、ドリトルを使うと以下に説明する1時間の授業(小中高では45〜50分、大学等では90分)で可能です。オンライン版のドリトルを使えば、あらかじめ教室のパソコンで動くことを確認しておくだけで、インストールの手間もありません。情報関係の授業をなさっている先生は、1時間だけ実施してみてはいかがでしょう。生徒の反応を見てから、改めて数時間の授業に発展させることも可能です。

宝物拾いゲーム

題材は「宝物拾いゲーム」です。プログラムは書籍「ドリトルで学ぶプログラミング」の5章にあります(ここでは教えやすいように若干変えてあります)。進め方は簡単で、生徒に見えるように自分の画面で1行ずつ入力しながら、生徒に入力させて、1行ずつ実行させて行きます。本は生徒には見せません。

前半

まず最初に、「ゲームを作るために、画面に主役を作ろう」と言いながら1行入力します。入力が簡単なように、主役の名前はひらがなにしました。また、必要に応じて「＝」「！」「。」などの記号の入力方法を伝えます。

生徒には先生の画面を見ながら同じように入力させ、実行させます。全員が実行できたか見て回りましょう。感度のいい学級では、早くもこの時点で「何か出た！」「カメだ！」と歓声が上がります。

```
かめた＝タートル！作る。
```

続いて、「次にかめたを操作するボタンを作ろう」と言いながら1行入力します。生徒が入力したら実行させて、「ボタンを作ったけど、押すとどうなる？」と問いかけます。

```
左ボタン＝ボタン！"左" 作る。
```

生徒全員がボタンを押しても何も起きないことに気付いたところで、「押したときの動作を教えていないから、コンピュータは何をしたらいいかわからないんだね」と理由を説明します。そして、「押したときの動作をボタンに定義しよう。押したときにかめたを左に回すんだよね」と言いながら、1行入力します。

生徒が入力したら実行させます。キー入力の速い生徒と遅い生徒がいますので、エラーになっている生徒のフォローを含めて、ここで全体を見て回りながら足並みを揃えましょう。

```
左ボタン：動作＝「かめた！30　左回り」。
```

高校生以上では、ここで「左回転のボタンができたから、右回転のボタンを追加してごらん」と言って、自分で作らせるのも効果的です。

続いて、(自分で作らせた場合は2,3分後に答え合わせを兼ねて)右ボタンを定義します。先生はプログラムをキーボードから入力していきますが、この2行と後から出てくる宝物の定義だけは、コピーして修正するのが便利です。入力が遅い生徒が多い場合には、コピーできることに気付かせるのもよいでしょう。

生徒が入力したら実行させます。ボタンでカメを左右に回転できるようになったことを確認させます。

```
右ボタン＝ボタン！"右" 作る。
右ボタン：動作＝「かめた！30　右回り」。
```

1分くらい操作させたところで「カメを回せるようになったけど、動かないと面白くないのでカメにエンジン(モーター)を付けてみよう」と言いながら2行入力します。

このとき、「いちどに200歩歩くと一瞬で動いてしまうので、0.1秒ごとに10歩ずつ歩くようにしてみよう」という説明を黒板に書くと、アニメーションのように動いて行く原理が伝わります。

生徒が入力したら実行させます。「動いた！」と声が出ることでしょう。すぐに画面から出て行ってしまうので、実行ボタンで何度でも実行できることと、ボタンで操作できることを伝えます。

これで、簡単なドライブゲームになりました。実行は10秒間で終ります。

```
時計＝タイマー！作る。
時計！「かめた！10　歩く」実行。
```

参考までに、ここまでのプログラムをまとめて書いておきます。小学校では、ここまでのプログラムで十分かもしれません。

```
かめた＝タートル！作る。
左ボタン＝ボタン！"左" 作る。
左ボタン：動作＝「かめた！30　左回り」。
右ボタン＝ボタン！"右" 作る。
右ボタン：動作＝「かめた！30　右回り」。
時計＝タイマー！作る。
時計！「かめた！10　歩く」実行。
```

&show(treasure1.png,nolink,画像の説明);

後半

生徒には、2,3分遊ばせましょう。その後、「ゲームだから、宝物を拾うようにしてみようか。かめたは花を集めるのが趣味なんだって」と言いながら、1行入力します。

```
タートル！作る "tulip.png" 変身する　ペンなし　100 100 位置。
```

生徒が入力したら実行させます。宝物は表示されましたが、カメと重なっても何も起きないことを確認させます。

次に、黒板にXY座標を描き、(100,100)の位置を図示しながら説明します。そして、生徒にこの1行をコピーして修正することで、画面上の異なる位置に3個の宝物を置くプログラムを作らせます。

続いて、「最後に宝物を拾えるようにしてみよう。たとえば、かめたと何かが重なったときに相手を消せば、拾ったように見えるよね」と言いながら1行入力します。縦棒記号は普段使わないので、入力方法を説明するとよいでしょう。

```
かめた：衝突＝「|相手|　相手！消える」。
```

これで宝物拾いゲームは完成です。参考までに、プログラム全体を書いておきます。10行程度ですので、中学校以上であれば、余裕を持って取り組めると思います。

```
かめた＝タートル！作る。
左ボタン＝ボタン！"左" 作る。
左ボタン：動作＝「かめた！30　左回り」。
右ボタン＝ボタン！"右" 作る。
右ボタン：動作＝「かめた！30　右回り」。
時計＝タイマー！作る。
時計！「かめた！10　歩く」実行。
タートル！作る "tulip.png"　変身する　ペンなし　100 100 位置。
タートル！作る "tulip.png"　変身する　ペンなし　100 -100 位置。
タートル！作る "tulip.png"　変身する　ペンなし　-100 100 位置。
かめた：衝突＝「|相手|　相手！消える」。
```

&show(treasure2.png,nolink,画像の説明);

進度の速い生徒には、宝物を増やして5個にさせたり、次のように乱数のサンプルを示して、実行するたびに宝物の位置が異なるように拡張させるのも効果的です。

```
タートル！作る "tulip.png"　変身する　ペンなし　(乱数(600)-300) (乱数(400)-200) 位置。
```

また、テキストのように、ボタンの定義に “LEFT” や “RIGHT” を加えて、キーボードからかめたを操作できるようにすることもできます。

まとめ

授業の最後では、5分くらいで構いませんので、今日体験したことの振り返りを行なってください。たった1時間の実習でしたが、生徒は実にいろいろなことを感じ取っているはずです。生徒から感想を引き出しつつ、次のようなことを板書などを含めて確認してみてください。

最後の「OS はオペレーティングシステムと呼ばれるソフトウェアで、Windows Macintosh Linux などがあります。少し高度になりますので、最後の部分だけは生徒の反応を見て解説するかどうかを判断してください。

- プログラムがどんなものかわかる
- ゲームなどのソフトはプログラムで作られている
- プログラムは人間が書いている
- プログラムは特別な「言語」で書く
- 文法が違うとエラーになる
- 間違って書くと間違って動く
- 書かれていないことは実行されない
- 上から順に実行される
- ある状態になったときに実行される命令もある（ボタン、衝突）
- ソフトは自分たちで作れる
- キー入力やマウスカーソルもプログラムが表示している OS

おわりに

小中高の授業で扱う多くの場合は、プログラミングの教育が目的ではありません。「ドリトルを教える」「プログラミングを教える」といった「〜を教える」という形の授業ではなく、上記のように適度なヒントを与えながら実習を行ない、そこから生徒が体験的に学んでいく学習が効果的です。

ドリトルを使うと、普段使っているゲーム機や携帯電話、ワープロなどのソフトウェアがどのような仕組みで動いているのかを体験的に学ぶことができます。ぜひ1時間だけでも授業に取り入れて、身近なソフトウェアの原理に触れさせてあげてください。生徒によっては一生に一度の貴重なプログラミング体験になるかもしれません。

中学校「技術・家庭」や高校「情報」の授業では、この実習を1時間やることで生徒が「ソフトウェア」と「プログラム」が何かを理解できるので、その後の授業がやりやすくなると思います。情報システムのような説明はもちろん、情報倫理のような話題でも「コンピュータウィルスは自然にできるものではなく、悪意を持った人間が作っている」と説明したときに、生徒に伝わるリアルさがまったく違って来るはずです。

（付録）体験から学べることの例

- プログラムがどんなものかわかる
- ゲームなどのソフトはプログラムで作られている
- プログラムは人間が書いている
- プログラムは特別な「言語」で書く
- 文法が違うとエラーになる
- 間違って書くと間違って動く
- 書かれていないことは実行されない
- 上から順に実行される
- ある状態になったときに実行される命令もある（ボタン、衝突）
- ソフトは自分たちで作れる
- キー入力やマウスカーソルもプログラムが表示している（OS）

先生方の反響（Blogから無断転載）

「１時間でやってみたソフトウェアのしくみ」(2008/11/21)佐藤先生（東京都立東大和高校）

&show(ysato.png,nolink,画像の説明);

ドリトルを使った「1時間で学ぶソフトウェアの仕組み」を２クラスやってみました。話の進行に合わせて、スライドを作りました。

入力自体はそんなに難しくないはずなのですが、スペースが抜ける事が多く、最初の頃はよくエラーが出ていました。「隣の人に聞いてみて」と促すと、うまくデバッグできはじめました。同じ命令を打っているわけですから、成功体験を伝達できるのです。

動き始めると、「おぉ〜っ！」「わぁ！」などの声がいろいろなところから聞こえてきます。

左ボタンを作ったあと、「右ボタンもつくってみて」というと、自分で考えて作る生徒が大半です。チューリップも「数を増やしてみよう」というと、自分で考えて作ろうとするなど、積極的に「自分で考える」生徒が多いことがこの授業の特徴です。

授業の終わりに近づくと、どれを書くとどう反応するかが理解できてくるので、さまざまな改造が始まります。

最後に感想や自己評価をしっかり取りたかったのですが、熱中してなかなかやめてくれない生徒が数多くいました。自分で動くものを作るのは、やっぱり、おもしろいんですね。

生徒の意見を見ながらまとめをしようと思っていたのですが、仕方ないですね。自己評価の集計結果は以下のようになります。

1.プログラミングがどのようなものか少しわかった そう思う(57)、やや思う(22)、やや思わない(3)、思わない(0) 2.コンピュータがどのようにして動くか少しわかった そう思う(57)、やや思う(24)、やや思わない(1)、思わない(0) 3.ソフトウェアの役割が少しわかった そう思う(52)、やや思う(22)、やや思わない(8)、思わない(0) 4.宝探しゲームを楽しく作れた そう思う(62)、やや思う(15)、やや思わない(3)、思わない(2) 5.プログラミングの授業をまたやってみたい そう思う(64)、やや思う(12)、やや思わない(4)、思わない(0)

「4.宝探しゲームを楽しく作れた」や「5.プログラミングの授業をまたやってみたい」で肯定的でない回答をしている生徒は、キーボードの操作があまり得意でないことがわかっています。

「”」や「！」の入力で苦戦する生徒も結構いました。授業では説明しているんですけどね。中には「ー」が入力できない生徒もいました。今までレポートどうしてたんだよ！なんて思ってしまいます。

まぁ、それ以外はいたって好評です。授業自体は楽しく進みますし、生徒が自ら考えようとする場面がよく見られたことが収穫です。反応を見て他のクラスでもやろうと考えていたのですが、これは全クラスでやることにしようと思いました。

準備はドリトルが動くブラウザがあればOKです。共有フォルダにドリトルを置いて配布しても良いでしょう。スライドは使ってもらってかまいませんので、ぜひ一度やってみてください。

「ドリトル　恐るべし！」(2008/11/28)福原先生（東京都立久留米西高校）

&show(fukuhara.png,nolink,画像の説明);

今日のあるクラスは授業時間の関係で他のクラスより進んでいたため、ドリトルの体験を５０分でやってみました。授業の進め方はだいたい以下の通りです。

- プログラムがゲーム、車、ロボットなどの制御に使われている話しを簡単にする。
- NHKのデジタル進化論ビデオ第5回目の５分～８分ぐらいまでのドリトルプログラミング画面を見せる。
- ドリトルを起動して、こちらのシナリオ通りに進める。
- デジタル進化論のビデオ第5回の最新ゲームのプログラムを２分程見せる。
- Googleフォームでアンケートをとって終了。

欠席者が何人かいたのですが、な、なんとクラス全員が授業が、とても楽しかった、楽しかったと回答しています。（楽しくなかった生徒は一人もいなかったのです）

プログラミングは楽しかったですか とても(13)、楽しかった(21)、あまり(0)、楽しくなかった(0)

さらに、プログラミングの授業をやってみたいという生徒が９０％を越えています。

プログラミングの授業はこれからもやってみたいですか 是非(13)、やってみたい(18)、あまり(3)、やりたくない(0)

生徒が興味を持てるようなシナリオ（授業展開）、教材（サンプルプログラム）があればもっともっと色々な事が出来そうです。生徒から感想もとっていますのでもう少し分析してみたいと思います。

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/d1h?rev=1515059352

Last update: **2018/01/04 18:49**