

情報の科学的な理解を深める学習法

2011年8月5日 兼宗 進 (大阪電気通信大学)

1 はじめに

1.1 自己紹介

- 兼宗 進 (かねむね すすむ)
- 大阪電気通信大学 医療福祉工学科 教授
- 専門はプログラミング言語、情報科学教育

1.2 今日の内容

楽しく本質を学ぶ授業のヒントを3つの切り口から紹介

- ドリトル: ソフトウェアを学ぶ
- ビーバー: クイズ問題で学ぶ
- アンプラグド: グループ活動で学ぶ

1.3 共通する考え方

生徒は理解したいと思っている。わかるから楽しい。わからないからつまらない。

生徒が知りたいのは自分と関係すること。無関係のことがわかって嬉しくない。教える側が教えたことではない。

ヒントは、自分たちの身近なもの。「何気なく使っていたけど、実はこういう仕組みだったのか!」という意外な驚きは重要。

ICTはデジタル世代の生活の一部。しかし、原理はわからない。その謎を解くのが教科「情報」の役割。情報科学と情報工学を活用しよう。

2 ドリトル言語

- 初心者が楽しくプログラムを書ける。
- 中身は本格的なオブジェクト指向言語。
<http://dolittle.eplang.jp>
- 兼宗が2000年に開発
- 高校では「1時間で学ぶソフトウェアの仕組み」が有効。^{*1} 資料として添付します。

3 ビーバーコンテスト

- 児童・生徒向けの情報科学コンテスト
- 問題を考えさせる体験を通して興味を喚起
- 日本では2010年から情報オリンピックが試行
- 実施は11月。学校で実施(授業、部活など)
- 過去問題の日本語訳を準備中。授業で利用可能

4 CS アンプラグド

- コンピュータ科学を楽しく学ぶ教育手法
- ニュージーランドのTim Bell博士たちが提唱
- 2007年に兼宗らが翻訳し12章までを出版
- 13章以降とオリジナルの内容を出版準備中
- 特徴は「教具の活用」「体験学習」「考えて、自分で発見させる」
- 12章の内容

学習	内容
1	点を数える(2進数)
2	色を数で表す(画像表現)
3	それ、さっきも言った!(テキスト圧縮)
4	カード交換の手品(エラー検出とエラー訂正)
5	20の扉(情報理論)
6	戦艦(探索アルゴリズム)
7	いちばん軽いといちばん重い(整列アルゴリズム)
8	時間内に仕事を終えろ(並び替えネットワーク)
9	マッディ市プロジェクト(最小全域木)
10	みかんゲーム(ネットワーク通信)
11	宝探し(有限状態オートマトン)
12	出発進行(プログラミング言語)

5 関連資料

- 「ドリトルで学ぶプログラミング」兼宗進、久野靖、イーテキスト研究所
- 「コンピュータを使わない情報教育」兼宗進監訳、イーテキスト研究所
- 「カメ太の日記」(Blog)
<http://kanemune.eplang.jp/diary/>

^{*1} <http://kanemune.eplang.jp/diary/2008-11-06-1.html>
<http://kanemune.eplang.jp/diary/2009-01-28-1.html>

1 時間で学ぶソフトウェアの仕組み

兼宗 進 (大阪電気通信大学)

1 はじめに

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 時間の授業で可能です。オンライン版のドリトルを使えば、インストールの手間も不要です。

2 宝物拾いゲーム

題材は「宝物拾いゲーム」です。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。資料は生徒には見せません。

2.1 前半

ゲームを作ることを伝え、「画面に主役を作ろう」と言いながら 1 行入力します。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。

かめた=タートル！作る。

続いて、「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。

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わからないんだね」と理由を説明し、「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。

左ボタン：動作=「かめた！ 30 左回り」。

続いて「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」。

次に「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。

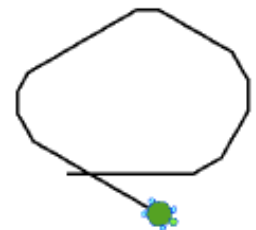
このとき、「いちどに 200 歩歩くと一瞬で動いてしまうので、0.1 秒ごとに 10 歩ずつ歩くようにしてみよう」という説明を黒板に書いて、アニメーションのように動かす原理を説明します。

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ふので、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 秒間で終わります。

時計=タイマー！作る。

時計！「かめた！ 10 歩く」実行。



2.2 後半

生徒に数分遊ばせた後、「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。

入力したら実行させます。表示された宝物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ね」と言いながら1行入力します。

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。裏技として、2行目と4行目には"LEFT"と"RIGHT"を加えて、左右の矢印キーで操作できるようにしています。

かめた＝タイトル！作る。

左ボタン＝ボタン！"左" "LEFT" 作る。

左ボタン：動作＝「かめた！ 30 左回り」。

右ボタン＝ボタン！"右" "RIGHT" 作る。

右ボタン：動作＝「かめた！ 30 右回り」。

時計＝タイマー！作る。

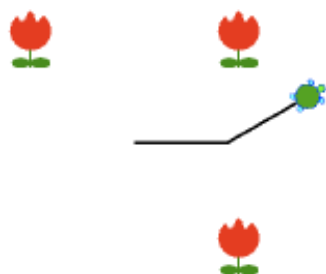
時計！「かめた！ 10 歩く」実行。

タイトル！作る "tulip.png" 変身する ペンなし 100 100 位置。

タイトル！作る "tulip.png" 変身する ペンなし 100 -100 位置。

タイトル！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」。



進度の早い生徒には、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ます。

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

2.3 まとめ

最後に振り返りを行いません。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。

- プログラムがどんなものかわかった
- 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
- 文法が違くとエラーになる
- 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)

3 おわりに

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。1時間で構いませんので、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ください。

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ります。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よ。

以上