

未来のプログラム好きを育てる

兼宗 進 (大阪電気通信大学 医療福祉工学科) kanemune@acm.org

久野 靖 (筑波大学 ビジネス科学研究科) kuno@gssm.otsuka.tsukuba.ac.jp

A Proposal of Language and Educational Tool for Programming Education

Susumu Kanemune (Osaka Electro-Communication University)

Yasushi Kuno (University of Tsukuba)

概要

中学生から高校生を想定した「プログラミングが好きな子どもを育てる」ための言語環境ドリトルと「情報科学の楽しさを伝える」ための手法 CS アンブラグドを紹介する。どちらも中学校から大学までの授業等で利用が進んでおり、専門教育を受けたことのない教員がプログラミングやコンピュータ科学の楽しさを伝えられることが特徴である。

1 はじめに

中学生から高校生を想定した「プログラミングが好きな子どもを育てる」ための言語環境ドリトルと「情報科学の楽しさを伝える」ための手法 CS アンブラグドを紹介する。

ドリトル [1-6] は筆者らが設計した言語環境であり、初心者が楽しさを感じて「プログラムを作りたい」と思える言語を目指している。以前冬のプロシンドで紹介し、多くのコメントをもらうことができた。言語は Smalltalk 風のブロックと JavaScript 風のオブジェクトのプロトタイプ階層を持ち、英語のほかに利用される現地の言葉での命令語や変数名を使用できる。

処理系を Web で公開し、ダウンロードすることに加えて Web ブラウザ上で Java アプレットの形で動くようにしたところ、端末教室でのインストールが不要なことから、多くの高校や大学の授業で利用されるようになった。

ドリトルの特徴のひとつに、「プログラミング経験のない高校などの教員が、生徒にプログラミングの楽しさを伝えることができる」ことを目指して取り組んでいる点がある。そのために用意した教材と、実際に Web を見て授業をしてほしいいくつかの高校での報告を紹介したい。

CS アンブラグドは、ニュージーランドの Tim Bell 博士が提唱している「子どもたちにコンピュータ科学

の楽しさを伝える」ための教育手法である。内容は、「画像の符号化」、「二分探索やハッシュ探索」、「並列ソーティング」、「オートマトン」のような大学の専門教育で扱うような高度な概念が並んでおり、それらを小学生から楽しく学ぶことができる。筆者は 2007 年 9 月に翻訳書を日本で出版した [7-9]。それ以降は多くの高校や大学で実践されている。国内では、翻訳と並行して中学校、高校、大学で実験的な授業を実施し、情報教育に効果があることを確認してきた。これらが実際にどのように使えるのかを、小学生を対象にした科学イベント、中学校と高校での授業、大学での講義などの事例で紹介する。

2 CS アンブラグド

表 1 にアンブラグドの内容を示す。ここでは、第 4 章「カード交換の手品」について、授業の進め方の例を示す。登場人物は女性教員と男子生徒である。

1. 先生は「今から、ひっくり返されたカードを当てる手品をします」と言い、生徒に白黒のカードをランダムに縦横 5 枚ずつに並べるように伝える。
2. 生徒は黒板にカードをランダムに貼る。

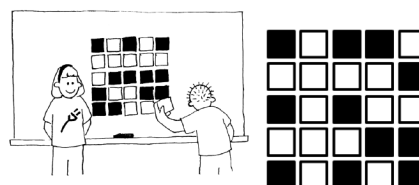
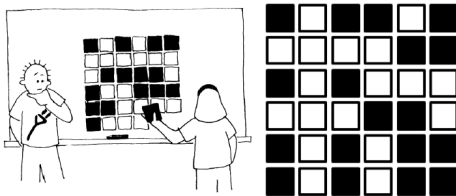


表1 アンブラグドの章構成

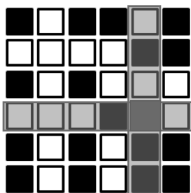
章	タイトル	サブタイトル	対象年齢	内容
1	点を数える	2進数	7歳以上	2進表現
2	色を数で表す	画像表現	7歳以上	画像のビット表現
3	きみの言うとおりに！	テキスト圧縮	9歳以上	LZ法
4	カード交換の手法	エラー検出とエラー訂正	9歳以上	パリティ
5	20の扉	情報理論	9歳以上	情報量
6	戦艦	探索アルゴリズム	9歳以上	線形、二分、ハッシュ
7	いちばん軽いといちばん重い	整列アルゴリズム	8歳以上	選択、クイック
8	時間内に仕事を終えろ	並べ替えネットワーク	8歳以上	並列処理
9	マッディ市	最小全域木	9歳以上	最小全域木
10	みかんゲーム	ルーティングとデッドロック	9歳以上	デッドロック
11	宝探し	有限状態オートマトン	9歳以上	オートマトン
12	出発進行	プログラミング言語	7歳以上	人工言語

3. 先生は「簡単すぎるので、もう少し難しくしましょうか」と言いながら、右端と下端に1列ずつ追加して縦横6枚ずつにする。



4. 先生は後ろを向き「1枚だけ裏返して」と伝える。
 5. 生徒は1枚を裏返す。
 6. 先生は黒板を見て、裏返されたカードを当てる。

これを授業で行うと、生徒はどうしてわかったのかと不思議に思う。数回繰り返すうちに、少しずつ気づく生徒が現れてくる。すると、残りの生徒は必死に手品のタネを考え出す。そして、わかる生徒が増えてきたところで、「先生が追加した縦横の1列ずつがパリティビットの役割をしていた」ことを解説する。ここでは白(または黒)の枚数が縦横で偶数になるように、パリティのカードを追加している。



このように、学習の中にゲーム性を取り入れ、学習意欲が低い生徒を含めて全員が必死に頭を使う学習を行うことが、アンブラグドの本質である。おそらく生徒は「パリティ」という言葉は忘れてしまうかもしれないが、「データに検査用のビットを追加することで誤りを検出できる」という手法そのものは、一生覚えているかもしれない。それくらい強烈な教育効果がアンブラグドには存在する。

コンピュータ科学の教育では、原理を教えた後で、実際にどのような身近な場面で活用されているかに結び付けることで理解を深めることができる。パリティの場合には、ネットワーク通信のエラー検出やCDのエラー訂正を説明する。

説明だけでは結び付きが弱いと、書籍のISBNや商品のJANコード(商品コード)のチェックディジットを計算させる演習を行うことが効果的である。生徒の文房具など持ち物に貼られたバーコードの右の桁はパリティに相当するチェックディジットであり、商店のレジで正しく読み取れたことを確認するのに使われていることを説明すると、生徒は自分の生活に関係する技術であることを理解し、強い興味を示す。

学習に対するモチベーションが高まったところで、12桁の数値から1桁のチェック桁を求める計算問題をワークシートで実習したり、情報システムの信頼性の学習などに発展させることが可能になる。アンブラグドを利用することで、生徒の興味を引き出し、発展させた学習に展開できるのは大きな利点と考えている。

3 CS アンブラグドの利用

3.1 海外の動向

CS アンブラグドは海外ではすでに米国や欧州を中心に一定の地位を築いており、各国語への資料や映像の翻訳が進められている。現在、Advisory Boardには筆者を含め、26名が登録されている。

3.2 科学イベントでの利用

国内でも利用が進んでいる。小学生向けには、科学イベントでの利用が行われている。もともとCSアンブラグドは、アルゴリズムの研究者である原著者が、小学校に入ったばかりの自分の子どもにコンピュータ科学の楽しさを伝えるために作り始めたこともあり、小学校の中学年以上であれば問題なく実施が可能である。筆者らは高校生を対象とした情報オリンピックの裾野を広げる活動として、小中学生を対象としたジュニア部会を作り、昨年から毎年夏に富士通と共同で科学イベントを行っている。

小学校4年生から6年生の子どもたちが、2進数やパリティ手品、オートマトンを使った宝探しゲームなどを体験し、見学の保護者を含めて好評である。

大学でも、高校生を対象としたオープンキャンパスや、大学祭での子ども向け科学イベントなどでも利用が増えている。

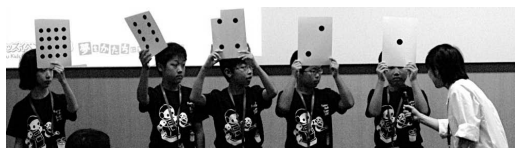


図1 科学イベント (2進数とパリティ手品)

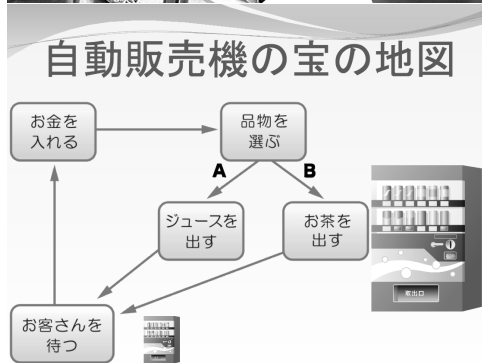


図2 科学イベント (オートマトンと状態遷移図)

3.3 授業での利用

学校でいちばん利用されているのは高校の「情報」の授業であろう。全国のすべての普通高校で1年間の授業が行われるが、多くの生徒にとっては受験科目でないため学ぶ動機を持ちにくく、教員もコンピュータの専門教育を受けていない場合が多い。そこで、CSアンブラグドは楽しい授業を行うための教材として注目されている面がある。情報の教員の勉強会で取り上げられることが多く、自治体が実施する教員研修でも取り上げられることが増えて来た印象がある。

大学でも、共通教育や、専門課程の初年度教育で利用され初めている。筆者の所属学科でも、医療工学を専攻する学生の初年度教育で、CSアンブラグドを利用しながらコンピュータ科学の基礎を学ぶ講義を計画している。その際、表1に示した内容だけではすべてをカバーできないため、ネットワーク通信などの項目を、必要に応じて新たに教材開発を行うことを考えている。

4 プログラミングへの発展

アンブラグドはゲーム性を取り入れた優れた教育手法だが、残念ながらプログラミングは扱われていない。しかし、コンピュータ科学やゲームプログラミングの楽しさを知るためには、プログラミングを避けて通るわけにはいかないだろう。

ここで、高校までの教育で、どのようにプログラミングが扱われているかを簡単に振り返る。

4.1 1時間で体験するプログラミングの提案

高校までのプログラミングの位置付けは、実に曖昧である。従来から工業や商業などの専門高校では、職業人を養成することを目的に、プログラミングを教えてきた。そこではプログラミングは「技能」であり、プログラムを作ることの楽しさは目的とされていない。

一方、中学校の技術家庭や普通高校の情報科目では、「専門教育をすることは目的でない」という理由から、プログラミングはほとんど扱われていないのが現状である。これは、2つの点で問題がある。

1つは、「身の回りの電化製品や社会インフラがブラックボックスになっているため、それらにコンピュータが内蔵されており、ソフトウェアで動いている」というモデルを理解することが難しく、特にソフトウェアを理解するためにはプログラムを書いてみる体験が不可欠であるという点にある。

もう1つは、現代では大学での学習や社会での職業において、ソフトウェアと関係する比率が高まっていることにある。情報を冠する学科に進学する生徒や、職業でソフトウェアやネットワークにかかわる社会人は増えている。たとえ情報関係の進学や就職が2、3割だと仮定しても、「全員が専門家になるわけではない」という理由で、プログラムやソフトウェア開発が何かをまったく知らせない理由にはならないと考える。

特にプログラミングについては、好き嫌いや向き不向きが強い傾向が存在する。そこで、就職または進学を決める前に、プログラミングを体験する機会を与えることが重要になる。そこで、高校1年生（遅くとも2年生）までに、すべての生徒に1時間のプログラミング体験授業を行うことを提案したい。これはプログラミングの教育が目的ではなく、身の回りで動いているソフトウェアの仕組みを体験することが目的である。そして副次的な作用として、プログラミングの楽

しさを伝え、プログラミングに対する好みや適性を体験を通して確認することが可能である。

具体的には中学校の「技術」や高校の「情報」の授業がターゲットになる。どちらも必修ではあるものの、生徒だけでなく、教える先生のほぼ全員もプログラミングの経験がない。

設計したカリキュラムは、通常のプログラミングの授業と比べて、次の特徴がある。

- 1時間*1で学べる。
- 自分の手でプログラムを入力して実行する。
- 「1行加えて実行する」を繰り返す。いちどに複数行を書く必要はなく、前の行に戻る必要もない。
- 先生の入力をまねて入力する。エラーが出たら、先生の1行と比べればよい（デバッグ不要）。
- 構文の理解は不要。プログラムを書けない先生でも教えられる。
- ソフトウェアがプログラムで書かれていることを理解できる。

参考までに、ドリトルの2種類の基本構文を紹介しておく。

変数=オブジェクト！パラメータ... メッセージ。

基本形である。オブジェクトにメッセージを送り、必要に応じて結果を変数に代入する。変数の宣言は不要である。オブジェクトは「作る」で生成する。

オブジェクト：プロパティ=「|パラメータ|...」。

オブジェクトにプロパティを定義する。右辺に「...」という形のブロックを指定するとメソッドの定義になる。

実際の授業では、先生は1行入力するたびに生徒への問いかけを行い、プログラムの意味と動作を考えさせながら進める。詳細は付録およびBlog [10]を参照されたい。

2008年秋に筆者のBlogで「1時間で学ぶソフトウェアの仕組み」として公開した後、多くの高校の授業で利用されたようである。その効果が驚きの声とともに、いくつかの高校教員のBlog [11-14]で報告されている。

4.2 計測・制御を学ぶ

2008年に改訂版が発表された中学校の指導要領では、技術・家庭科で「計測と制御」が必須の内容になっ

*1 中学校と高校の1時限は50分である。

た。全国のすべての中学校で、どのような教育が可能かは見えていないが、教科書の執筆と教材の開発が進められている。

現在は、現時点で入手できる 10 種類前後の教材を集めて評価を行っている。多くは CPU を搭載した車型のロボット教材で、ライントレースなどを行うための光センサなどを搭載している。プログラムはフローチャートや画面にブロックを並べる形のビジュアルなものが多かった。

ドリトルでは、作成したプログラムを転送して実行する仕組みをいくつかの教材に対応する形で対応を進めている。

4.3 ネットワークでのソフトウェアを学ぶ

現代において、生徒にとって身近なコンピュータは、パーソナルコンピュータではなく、携帯電話であり、携帯ゲーム機である。「1 時間で学ぶソフトウェアの仕組み」は、パーソナルコンピュータまたはゲーム機におけるソフトウェアの裏側を体験的に理解することができる教材だが、ネットワークで通信し合う、携帯電話や Web の通信を説明できるものではない。

そこで、教室内の端末を使い、P2P の通信や、サーバー・クライアント通信を体験できるドリトルによる教材の開発を進めている。ひとつは文字や数値をオブジェクトとして転送し合う機能で、中学校から大学までの授業でチャットプログラムなどを題材にした授業で利用されている。もうひとつは、Web サーバーとして教室内にサーバー上で動くプログラムを公開できる機能であり、来年からの授業に向けて開発を進めている。

5 まとめ

コンピュータはパソコンだけでなく携帯電話やゲーム機など多くの電子機器で活用されており、情報科学で学ぶ内容は、実は生徒にとって身近な興味のある題材である。

一方、探索やソートなど、個々の内容を取り出した学習では、日常生活との結び付きを理解することが難しいため、何のために学ぶのかを生徒が理解することは難しいという問題があった。

今回紹介したアンブラグドでは、洗練された教材を使い、自分の手を動かしながら理解することで、情報科学の代表的な内容を、小学生以上の生徒が興味を持って意欲的に学習することが可能である。

情報科学は応用的な学問である。原理の楽しさを学ぶと同時に、それがどのように役立てられているかを知ることは大切である。アンブラグドを学習した子どもがドリトルなどの入門用言語を通してプログラミングの楽しさを体験できるように今後もカリキュラムなどを整備して行きたいと考えている。

参考文献

- [1] 兼宗進, 御手洗理英, 中谷多哉子, 福井真吾, 久野靖. 学校教育用オブジェクト指向言語「ドリトル」の設計と実装. 情報処理学会論文誌, Vol.42, No.SIG11, pp78-90, 2001.
- [2] 中谷多哉子, 兼宗進, 御手洗理英, 福井真吾, 久野靖. オブジェクトストーム: オブジェクト指向言語による初中等プログラミング教育の提案. 情報処理学会論文誌, Vol.43, No.6, pp1610-1624, 2002.
- [3] 兼宗進, 中谷多哉子, 御手洗理英, 福井真吾, 久野靖. 初中等教育におけるオブジェクト指向プログラミングの実践と評価. 情報処理学会論文誌, Vol.44, No.SIG13, pp58-71, 2003.
- [4] 兼宗進, 阿部和広, 原田康徳. プログラミングが好きになる言語環境. 情報処理, 特集 未来のコンピュータ好きを育てる, Vol.50, No.10, 2009.
- [5] 兼宗進, 久野靖『ドリトルで学ぶプログラミング』イーテキスト研究所, 2008.
- [6] プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp>
- [7] 兼宗進監訳『コンピュータを使わない情報教育: アンブラグド・コンピュータ・サイエンス』イーテキスト研究所, 2007. <http://www.etext.jp/unplugged.html>
- [8] 西田知博, 兼宗進. コンピュータ科学を楽しく学ぶ. 情報処理, 特集 未来のコンピュータ好きを育てる, Vol.50, No.10, 2009.
- [9] コンピュータサイエンスアンブラグド.
<http://dolittle.eplang.jp/?unplugged>
- [10] カメ太の日記『一時間で学ぶソフトウェアの仕組み』.
<http://kanemune.eplang.jp/diary/2008-11-06-1.html>
- [11] 情報科 Blog『1 時間でやってみたソフトウェアのしくみ』.
http://blog.goo.ne.jp/yoshisato_2004/e/dcdf6da2b6da91051fe98241ce8f5590
- [12] 情報科メモ帳『ドリトル, 恐るべし!』.
<http://jyohoka.exblog.jp/10222741>
- [13] ネットで教科「情報」日記『ドリトルで学ぶソフトウェアのしくみの授業』.
<http://htanaka.exblog.jp/10256122>
- [14] ありおり (日記)『ドリトル, マジ半端ねえ』.
<http://www.ariori.com/diary/2009/01/09/>

(付録) 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

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たいかがでしょう。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。

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。本は生徒には見せません。

2.1 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。入力が簡単のように、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!」「カメラ!」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。ボタンでカメラ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「カメラ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ラにエンジン(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秒間で終わります。

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。

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。

2.2 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」。

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。

2.3 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
- 間違っって書くと同違っって動く
- 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)

(さらに付録) 高校の授業の反響 (許可を得て Blog から転載)

1 「1時間でやってみたソフトウェアのしくみ」佐藤義弘先生 (都立東大和高校) [11]

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は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、「おお〜!」「わあ!」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、どれを書くかどうか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と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「-」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!なんて思ってしまうます。

まあ、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。

2 「ドリトル 恐るべし!」福原利信先生 (都立久留米西高校) [12]

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。

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くてという生徒が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。