

小学生のプログラミング教育の実践と検証

学籍番号：79857843

学部学年：環境情報学部4年

氏名：坂野敦史

アブストラクト

プログラミング教育に重要なことは、入門段階で動機付けと方向付けをすることである。また、一般の予想に反して子どもたちは算数やプログラミングが好きである。しかし、プログラミングの楽しさを伝えるには、TAや考える仕掛けなどの学習環境が必要である。このレポートでは、サマースクールの実践を通して得られた上記の内容を検証する。

プログラミング教育の目指すもの

プログラミングは、プログラマや学者などの一部の専門家だけではなく、コンピュータを利用しようとする様々な人が学ぶべき内容である。なぜなら、プログラミングは、コンピュータの仕組みや有効な使い方を学ぶ上で重要な内容だからである。この章では、プログラミングを学ぶ目的と、学習のロードマップについて述べる。

なぜプログラミングを学ぶのか

プログラミングは、プログラマや学者などのプログラミングを職業的に直接利用する者だけのものではない。インターネットを利用して新しいビジネスを始める人、企業の情報システムを改善したりする人にとっても重要である。コンピュータを使って、新しいシステムを創り出そうとする時、プログラミングの知識は有効である。それは、コンピュータは万能な道具ではなく、使うべきではない問題も存在するからである。例えば、SF小説のように、コンピュータが人間を裁くということはコンピュータにとって非常に難しい。この裁判システムのようなコンピュータにできないものを作らないためには、プログラミングを理解することが有効なのである。

何を学ぶべきか

大きく3つある。どう使えば良いのか知っていることと、動いている仕組みを知ること、考え方を学ぶことである。

コンピュータをどう使えば良いのか知っていなければ、新しいシステムを創り出すことは出来ない。まず、人間にはできない処理を理解する必要がある。例えば、大量に保存されたメールの中から、先月に送られてきた友人のメールを探す、という作業があったとする。手作業でやると10分かかるところが、

コンピュータなら1秒で終わる。次に、コンピュータが苦手な処理も理解しなければならない。そうしないと裁判システムを作ることになる。

単に、どういう処理が得意か知っているだけではなく、なぜ得意だったり苦手だったりするのかを理解していなければならない。新しいシステムを創り出す人は、自分でコンピュータがその処理が得意かどうか判断しなくてはいけないからである。その判断をするには、動いている仕組みを知ること重要である。また、動いている仕組みを知るとは、コンピュータが問題を起こしたときに対処する上でも重要である。

プログラミングを通して、考え方を学ぶことも重要である。普通、プログラミング教育で、考え方を学ぶとか論理的思考を身に付けるというと、ソートなどのアルゴリズムの勉強が挙げられる。しかし、私が考えるプログラミング教育は、そういったアルゴリズムだけではない。例を挙げよう。レストランで下げ膳をしているとする。手当たり次第に両手に抱えて運んでも効率が悪い。テーブルの上で、食器を種類別に重ねて運ぶ方が効率がよい。こうした作業効率の問題は、プロジェクト管理やデスクワークにも存在する。アルゴリズムの学習はこうした作業一般を含むべきである。しかし、学習者はソートのアルゴリズムの勉強からでは、考え方を学ぶ、という意識が起こらない。今、やっていることが何のために学ぶのかを抑えながら学習を進めなくてはならない。

プログラミングの入門教育

最終的には、新しいシステムを創り出す人を目的としているが、その学習には順序が重要である。特に入門教育が重要である。入門教育を間違えると、その後の学習に影響が出るからである。

動機付けと方向付け

入門教育において重要なことは、動機付けと方向付けである。

動機付けとは、それを学習したいという意欲を湧かせることである。学習意欲がなければ、学習効果も半減する。例えば、プログラミングがパズルのように楽しいと思わせることかも知れないし、自分が希望する研究をするにはプログラミングが必要なのだと認識させることかも知れない。

方向付けとは、なぜ学習するのか、何を学習するべきなのかを明らかにすることである。例えば、コンピュータは煩雑な仕事をすばやく処理することを知っている。プログラミングをすると、自分だけのシステムが作れることを知っている。

内容を絞る

入門教育では、あまり多くの内容を盛り込まないことである。それは混乱を招かないためである。次のステップに進む前に、時間をかけて確認し、よくまとめるべきである。先を急いでも何も身につかない。先に進みたい者へ指示を出すのは良いと思う。

例えば、反復、分岐を必要なときに使える。コンピュータが苦手な処理を理解する。自分のアイデアをプログラミングで実現できる。モデルを作る。といった内容は、ごく初期の内容としては扱うべきではないと考える。

サマースクールの試み

実施概要

企画名称：コンピュータキャンプ2001

目的：プログラミングを楽しく学ぶ

期間：平成13年8月27日（月）・28日（火）

場所：「新川崎・創造のもり」K2タウンキャンパス

対象：川崎市在住の小学校4年生～6年生(パソコン経験は問わない)

人数：35人

授業体制：インストラクタ1名＋学生アシスタント10名

主催：慶應義塾大学・川崎市

共催：川崎市産業振興財団

後援：川崎市教育委員会

協賛：（株）内田洋行、（株）デジコン

企画：慶應義塾大学環境情報学部「大岩研究室」

カリキュラム



サマースクールは成功したのか

生徒の反応

一般的には、子どもは算数が苦手であり嫌いであるという認識がある。さらに、プログラミングも子どもにはつまらないだろうし、できないと考えられている。しかし、実際には子どもは算数のパズルが好きであるし、プログラミングが好きなのである。二日目の朝、多くの生徒が席に着くなり他のゲームをそっちのけで初日に解けなかったパズルに挑戦していたことから、算数やパズルが好きなのが分かる。アンケート結果から子どもたちがプログラミングを楽しんでいたことも明らかだ。

子どもにとって、プログラミングは楽しいのである。確かに、できなくて投げ出してしまう子もいる。しかし、TAが補佐をすることによって、子どもたちはやる気を取り戻すことができるのである。プログラミングがつまらないのではなく、できないことに嫌気がさしているだけであり、TAが補佐さえしてあげれば、ほとんどの子どもたちはプログラミングで自分の作品を作ることができるのである。

受講者は普通の小学生である。特に選んだ子どもたちではない。確かに、親は教育に関心を持っていることは間違いないだろうが、特にプログラミングができそうな子どもを集めたわけではない。学年も性別もバラバラである。

保護者の反応

初めは大学生がコンピュータを教えるという理由で来た親も、プログラミングを教えるということに理解を示してくれた。しかし、保護者へ十分な説明ができたわけではない。保護者の意思が子どもの学習内容を大きく左右するので、保護者への理解を深めることも重要であると考えます。

TAの反応

子どもに教えるということ十分に楽しんでいたようである。気になったのは、楽しかった、最初は子どもとの会話に戸惑ったなど、感想の関心が自分に向いていたのが気になった。もう少し、生徒が予想もつかないアイデアを出して驚いた、などの生徒へ関心が向くように仕掛けることができれば良いと感

じた。

楽しい学習環境

子どもたちは算数やプログラミングが好きである。ただし、算数やプログラミングの楽しさに気づくには工夫が必要である。ただ、そこに算数の問題用紙やプログラミング環境があるだけでは、子どもたちに、それらの楽しさを伝えることはできない。楽しい学習環境を構築することで、プログラミングの楽しさを伝えることができる。

TAの存在

TAの果たす役割

インストラクタが教えられない個別の問題に対応する。生徒は、相性の良い人に教えてもらうことができる。学習には、人間的な要素も影響が大きいため、こうした仕組みを作ることができるのは大きい。面白い点は、必ずしも社交的で明るい人がTAに向いているとは限らない。いろんな生徒がいるものである。休み時間に走り回らない小学生だっているのである。

TAの有効な活用法

授業はTAではなく、インストラクタがするものである。確かに、TAが個別に教える方が分かりやすいかもしれないが、基本的にインストラクタが教えるべきなのである。TAが個人指導してしまうと、TAが何人いても足りなくなってしまう。インストラクタが学生アシスタントに指示して、全体を集中管理した方がよい。学生アシスタントが足りないのではなく、現状の人数で足りるようにインストラクタが教える必要がある。学生アシスタントが個別に教えると教え方やペースがバラバラになり、負担も大きい。

教えればできるが、教えないとできない。ヒントを出して気づかせるには時間がかかり、生徒は考えているうちにやる気を失ってしまう。かといって、学生アシスタントが手を出してはいけない。

考える学習形式

プログラミング教育における良いカリキュラムとは何だろうか。プログラミングで教えることは、身体的技能や宣言的知識ではなく考え方である。そのため、操作の説明を聞いて作業をやらされるだけでは、ゲームを完成させることができても、生徒はプログラミングできるようにはならない。そこで、試行錯誤を繰り返すことで、何故そのような操作をする必要があるのかという意味を考えさせる。つまり、操作の意味を意図的に考えさせることが、良いプログラミング教育のカリキュラムには重要である。

失敗させる、考えさせる、まとめる、というプロセスが重要である。

見せ方を変える

キュー、スタック、プライオリティーキューを教えるとする。ならば、お手伝いをする順番を考えてみる。頼まれた順に取り掛かる。最近、頼まれたものから取り掛かる。おこづかいの額が多いものから取り掛かる。それぞれ、キュー、スタック、プライオリティーキューに相当する。題材や見せ方を変えるだけで、小学生は理解することができる。しかし、題材がおこづかいだから面白いだけではない。どうやったら効率の良いアルゴリズムができるのか考えるという、一番教えた内容を小学生は楽しいと思うことだろう。ただし、あくまでも楽しい学習環境は、教えたことの楽しさに気づかせるきっかけに過ぎないことを忘れてはならない。

ToonTalkについての考察

ToonTalkとは

ToonTalkというソフトウェアを用いた授業カリキュラムを構築した。ToonTalkは、子どもにプログ

プログラミングを学ばせることができる教育ソフトである。キーボードでプログラム言語を打つ代わりに、インタラクティブなアニメの世界で、ロボット、掃除機、自転車の空気入れといった身近なキャラクターを使ってプログラミングをすることができる。

有効性

小学生でもプログラミングを体験するインターフェースを実現している。

問題点

ローカライズ

日本語訳がおかしいことがある。マーティのヒントも理解しにくい。パズルで「生まれた月を入力しなさい」と言われても英単語が分からない。「ピクチャ」ではなく「画像」とか「絵」にした方がよい。ダスティの鼻が「E」だと「消す」、というのは分かりにくい。

インターフェース

指で指しているものを、揺れていることで表現するのは分かりにくい。指の当たり判定が分かりにくい。ロボットの思考の中で、操作を間違えると、一からやり直すのは辛い。しかも、そのロボットはもう使い物にならない。作業中は部品やブロックだらけなので作業スペースが狭い。道具があんなに大きくなくても良い。パンピーで全てのを小さくするとちょうど良くなる。ネズミが来るとよけても無駄である。かつてにネズミが来るのではなく、道具が動くタイミングを自分で決めたい。特に、床に物がたくさんある場合、せっかく作ったものがなくなることがある。小さいピクチャに大きなピクチャを落とすと、ダスティですい出せなくなる。これもやり直すことができない。ダスティが吸い込みつづけるのは使いにくい。

パズル

いきなり難易度がアップするので理解がついていかない。問題数を増やし、ゆっくり確実に理解できるカリキュラムが必要である特にロボットの使い方が難しいので、多くの類題が必要である。一人でもできるようになるには、段階をふんで類似問題を出すか、ヒントを多くして欲しい。マーティのヒントを工夫する必要がある。マーティがヒントではなく、答えを言うのが早い。ヒントは、何回目かではなく、パズルの状況に応じてヒントを出して欲しい。同じ問題でも何通りも解き方があることを説明してほしい。

サンプルデモ

一時停止、早送りができてほしい。デモを見ながら、自分の作業をしたい。15分は長いかも。同時再生ができれば嬉しい。

取り扱い説明書

一人でやるなら絶対に必要である。現状のドキュメントやサンプルデモ、パズルでは絶対に無理である。起動の仕方、ロボットの使い方などを説明したものが必要だろう。

展望

ToonTalkは、自分のアイデアを実現するツールよりも、入門学習用の教材として有効であると考えられる。ToonTalkはビジュアルでプログラミングするだけではなく、ビジュアルにコンピュータの仕組みを理解する教材としても有効であると思う。逆に、コンピュータの仕組みを理解する上でビジュアルであることは重要であると考えられる。

入門学習用のためには、改善すべき点がある。上(右)スピードはロボットがピクチャの裏で動いていると説明しやすい。座標の値を変えるだけでもボールを動かすことができるが、スピードの変数が用意されている。本来なら、動かす処理をロボットに命令するはずが、隠されてしまっている。恐らく、処理

速度の異なるマシン間でも同じ速度で動くようにするためだと思われる。しかし、このスピードがあるために、コンピュータの処理がユーザから見えなくなっている。確かに、マシンの処理速度に依存する問題は本当に教えたことではないかも知れないので、スピードという変数が用意されていても良いと思うが、スピードの裏でロボットが動いていると良いと思う。全ての処理がロボットで成り立っていて、その動きを見ることができると、コンピュータの仕組みに対する理解が深まると思う。同様に、本物の爆弾の裏にロボットが欲しい。

プログラミングへの動機付けとして、オリジナルゲームを創ることが出来ることは有効である。そのためには、画像や音声を簡単に使えるようにしてほしい。パスの知識が必要なのは小学生に使いやすいインターフェースとはいえない。ビジュアルプログラミングの良さが大きく損なわれてしまう。透過GIFが使えると、ゲームの画面を綺麗にすることができる。