

ToonTalk を利用したプログラミング教育環境の構築と実践

A Trial Instruction of Programming with ToonTalk

松澤 芳昭^{†1}

Yoshiaki Matsuzawa

大岩 元^{†2}

Hajime Ohiwa

^{†1}慶應義塾大学 SFC 研究所

^{†2}慶應義塾大学 環境情報学部

^{†1}Keio Research Institute at SFC

^{†2}Department of Environmental Information, Keio University

<あらまし> ToonTalk は、子供たちのためにデザインされたビジュアルプログラミング環境である。子供たちはマウスを使い、直感的な操作でプログラミング作業を行うことができる。筆者らは、この教材を利用したプログラミング教育の可能性を調査するために、小中学生を公募して 10 時間程度の実験授業を実施した。

実験授業において、ほとんどの子供たちが 1 時間程度で基本操作を習得し、創造的な作品を完成させることができた。しかし、「プログラム中の属性を変更することができる」という能力は身につけても、「新しいプログラムを考え、組み立てることができる」という能力を身に付けるに到るまでには大きな壁があることが分かった。

本稿では、この実践的試みの概要と成果を報告するとともに、ToonTalk がプログラミング教育の教材としてどのように位置づけられるかを考察する。

<キーワード> プログラミング教育、小中学生、ToonTalk、ビジュアルプログラミング、授業実践

1.はじめに

プログラミング入門者にとって、プログラミング言語の「文法」を習得しなければならないことは大きな障壁である。そのため、プログラミングの入門教育は、学習者が文法誤りの対処などに振り回されずに、できるだけ「創造的作業」に集中できる環境で行われることが望ましい。

そのような環境を探るために、筆者らは「ToonTalk」というソフトウェアを利用して、実際に子供たちに対して実験授業を行った。

本稿では、この実践的試みの概要と成果を報告するとともに、ToonTalk がプログラミング教育の教材としてどのように位置づけられるかを考察する。

2.ToonTalk の概要

ToonTalk は、Ken Kahn によって開発された、子供たちを対象に設計されたビジュアルプログラミング環境である[1]。プログラミングの各要素が、例えば、変数は「箱」、手続きは「ロボット」というようなメタファーに置き換えられており、マウスを利用して、直感的に、ゲーム感覚の操作でプログラムを構成できる。

ToonTalk において、「プログラムを記述して実行する」ということは、「ロボットを訓練して、実行する」ということに対応する。ロボットの訓練法はともユニークで、「一度ロボットの思考の中で仕事の手順を一度やって見せる」というものである。具体

的な仕事をやって見せて、その後抽象的な仕事としてロボットに実行させるという仕組みであり、抽象的な仕事として最初から記述しなければならない従来のインターフェイスとは異なっている。



<図 1 : ToonTalk フリーモード>

ToonTalk は、2 つのモードから構成されている。チュートリアルとして、「パズルモード」が用意され、パズルを解きながら基本操作を習得することができる。もう一つのモードである、「フリーモード」(図 1)では、自分が考えたものを自由にプログラミングすることができる。

3.実験環境

実験授業は 2001 年度は小学校高学年を対象に、2002 年度は、中学生を対象に実施した。参加者は、

「コンピュータ経験は問わない」という条件で公募した。2001年度は約半数の子供たちがコンピュータ未経験者であり、2002年度はほとんどの子供たちがコンピュータ経験者であった。参加人数は、2001年度が35名、2002年度は15名程度である。いずれの実験授業も2日間、計10時間の授業を行った。1人1台のコンピュータを配置し、参加者4~5名に対して1名のTAを配置した。

内容は、まずパズルモードで基本操作を習得し、その後、フリーモードでオリジナルゲームの作成を行うというものである。オリジナルゲームの作成は、テンプレートとして与えられたピンポンゲームを改造することによって行った。

4. 結果と考察

ToonTalkは、実際に操作してみると、当初TAを担当した大学生の間で、「このような操作が子供たちができるはずがない」という意見がでるほどに操作は複雑である。しかし、実験では1時間ほどで全ての子供たちが基本操作を習得してしまった。ゲーム感覚であるために、子供たちは夢中でToonTalkに取り組むので、理屈よりも先に操作を覚えてしまうようである。

結果、子供たちの創造的作業が阻害されるという状況はほとんど見受けられなかった。子供たちは最後までToonTalkに夢中で、飽きる子供はほとんどいなかった。ほとんどの子供がアンケートに、「ToonTalkをもっとやりたい」と答えた。そして、最終的に全ての子供たちが創造性豊かな作品を完成させることができた。

ただし、ここで考えなければならないことは、子供たちの作品が「ボールを増やす」、「画像の色を変える」、「音を変える」といった、テンプレートプログラムの属性の変更や、既にあるロボットを組み合わせるといった作品にとどまっていることである。確かに、この作業は試行錯誤しながらゲームのチューンアップをする作業であるので、創造的なプログラミング作業の一過程ではある。

しかし、プログラミング教育で真に求められるのは、「自分で作りたいロボットを訓練し、実行できる」という能力ではないだろうか。

2002年度の試みでは、この目標をより明確にし、ロボットの理解を深めるために、ロボットの基本的な訓練法から段階的に学習できるようにしたパズルゲームを開発し、少し時間をかけて指導した。しかし、ロボットの訓練に興味をもつ子供は増えたものの、自分の思うように訓練できるようになるという

ことについては、授業時間の制約もあり果たせなかった。

この問題点をToonTalkインターフェイスと、パズルを利用した教育方法の2つの側面から考察する。

第1に、ToonTalkにおけるロボットの訓練インターフェイスの問題を挙げる。ロボットの訓練は2章で述べた仕組みで行われ、具体的な仕事を実行してみせる場面と、ロボットが抽象的な仕事として実行する場面が、同じ場面として表現される。そのため、逆に具体的な仕事と抽象的な仕事の区別がなくなってしまう、子供たちは、「抽象化」という概念の習得に苦労してしている様子であった。

第2に、今回実施したような、パズルを利用した教育方法の問題を挙げる。子供たちの特徴として、器用な子はパズルを解いて先に進んでいくが、難しいパズルになるとロボットがうまく使いこなせず、そこでつまづいてしまうケースが多かった。逆に、不器用でも試行錯誤をしながら一步一步プロセスを確認していく子は、ロボットが使いこなせるようになっていった。

パズルのように、「答え」を目標とする教育は、子供たちへの強い動機付けとなる。しかし、答えを求めることをが主な目標となってしまうと、プログラミングの学習で重要な、「試行錯誤の過程を通して、プロセスの意味を考える」という目的が子供たちに伝わりにくくなってしまふ。動機付けをしつつ、試行錯誤の過程を学ぶことができるような指導をする必要があり、適切な環境を考える必要がある。

5. さいごに

ほとんどの子供たちが、操作を苦にせず、プログラミングという作業を楽しんでくれた。このことから、少なくとも「文法」を意識せずに創造的なプログラミング教育ができる環境として、ToonTalkの有用性が示唆されたと筆者は考える。

4章の最後で提起された課題については、子供たちにどのようなプログラミング教育をすべきかということも含めて今後考えて行かなければならないと筆者は考えている。

さいごに、実験授業を支えてくれたスタッフの皆様と、2002年度の実験授業の指導もしていただいた、ToonTalkの作者であるKen Kahn氏に感謝する。

6. 参考文献

[1]ToonTalk - www.toontalk.com