

情報教育のための学習環境デザイン

Designing Learning Environments for Information Education

上田信行*・古堅真彦**・森石峰一***

Nobuyuki UEDA・Masahiko FURUKATA・Minekazu MORIISHI

*甲南女子大学, ** (財)国際メディア研究財団,

***帝塚山学院大学 勤務, ***大阪電気通信大学 大学院生

*Konan Women's University

**International Media Research Foundation

***Tezukayama Gakuin University

***Graduate School Student, Osaka Electro-Communication University

【概要】本発表の目的は「プログラミングを学ぶ」ことの教育的意義を検討するために行った4つのワークショップを分析し、情報教育のための学習環境デザインへのアプローチを提案することである。教材コンテンツは学びの文脈から切り離して考えられないという視点から、今回は、情報教育のための学習環境デザインの重要性についての具体的提案を行いたい。playfulでreflectiveな学習環境をデザインするためには(1)課題のデザイン,(2)ツールのデザイン,(3)協同作業の場のデザイン,(4)批判的賞味の場のデザインについて考える必要があるということを強調したい。

【キーワード】アルゴリズム, プログラミング, 学習環境デザイン, メタ認知, 学びのコミュニティ

■研究の背景と動機

我々は、「プログラミングを学ぶ」ことの教育的意義を検討するために2年にわたって実験的ワークショップを行ってきた。このプロジェクトの目的は、プログラミングに関する様々なワークショップを通して情報教育の方向性を探るとともに、そのための学習環境デザインのための具体的な方法を開発することにあった。具体的にはプログラミング教育の本質であるアルゴリズムの認識を深めていける学びの場をどうデザインすればいいか、そして、その中で子どもたちは何を学んだのかを問いつつ吟味してきたのである。アルゴリズムというのは、ある課題をコンピュータに実行させるにはどのように手順を記述していけばいいかという論理的思考(道筋)のことであり、プログラミングとは、アルゴリズムをコンピュータが理解できるように正確に手続きとして記述していくことである。

アルゴリズムを考え、それをプログラム化するプロセスにおいては、プランニングや課題分析能力に加えて内省やモニタリングといった自分の認知過程を理解し、コントロールする意識的で意図的なメタ認知的スキルが要求される。しかし、現在の学校教育の中では、子どもが自らの学びのプロセスを振り返ったり、問題を深く吟味したり、といった反省的思考(reflective thinking)を通して学んでいく機会が少な

いように思われる。reflectionなどのメタ認知に注目してこれからのプログラミング教育を考えようというのが筆者らの視点であり、研究の動機である。

■研究の経緯

我々が行った実験的ワークショップは、(1)プログラミング活動における内省的認知の重要性に注目した「Reflective Design(1998)」,(2)Java言語を使ってreactiveな作品づくりを可能にした「Playful Design(1998)」,(3)新しいプログラム言語をつくるという「ことばのデザイン」に取り組んだ「Tangible Design(1999)」,そして、(4)LEGOブロックを使って作成した自動車をコンピュータで制御し、1メートルを1分で走行するようにプログラミングするという「Dynamic Design(1999)」の4つである。

■研究の目的と方法

本研究の目的は、これら4つのワークショップを分析することによって、情報教育のための学習環境を構築していくための重要なポイント(デザイン・エレメント)を明らかにすることである。

研究の方法は、それぞれのワークショップの目的、アクティビティの内容、開発したツール、結果を概観し、ワークショップのデザイン・エレメントになりうるものを抽出する。

■workshop 1 : Reflective Design

このワークショップは、アルゴリズムを考え、それをプログラム化するプロセスに内在するプランニングと内省的認知に焦点をあてて行ったものである。

2人1組になってあらかじめ用意した図形を1つ受け取る。受け取った参加者たちは、その形状をできるだけ確に日本語で記述する。つまりプログラミングをする。そしてその記述したものを、図形を見ていない他の参加者たちのグループにわたし、受け取った参加者たちはその記述からもとの形状を再現していく。最初は比較的単純な図形からはじめ

1. 目が2つだけである。
2. それはコンパスで描いたまる。
3. 左目の瞳は2で描いたまるにくっついていて、中心から右にある。
4. 右目の瞳は2で描いたまるにくっついていて、中心から下にある。
5. まつげはない。
6. 瞳は黒で塗りつぶしたコンパスで描いたまる。
7. 左目と右目（2で書いたやつ）はくっついていて。
8. 2で書いたやつ直けいは瞳の直径の2倍。
9. よく考えたまえ〜
ワッハッハー



ワークショップ課題

で順次複雑にしていった。

ここでは既存のプログラミング言語などは一切使用せず、参加者自らが自己の中でプログラミング言語（のようなもの）を創造することを体験することによって、筆者たちがプログラミングという作業において、教育的観点からも重要と考えている「アルゴリズム的な感性」、「論理的思考」、「デバッグという考え方」などを育むことができるのではないだろうかと考えた。

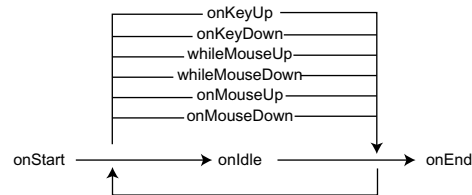
■workshop 2 : Playful Design

JAVA言語を使ってマウスに反応するreactiveな作品をつくり、その作品をweb上で公開するというものである。アトリエでは高校生と研究者たちが一緒にチームを組み、研究者がガイドしつつ作品をつくりあげ、その成果をweb上の展覧会として企画、発表し、参加者の友人たちからコメントをもらうというワークショップの流れをデザインした。JAVAを使ったのはweb上での発表が可能なことと、プログラマーにとってアルゴリズムを構造化しやすいという言語特性からである。

このワークショップには著者の一人（古堅）が

開発したProgramming Baseを使用した。

Programming Baseとはプログラミング言語であるJavaを利用した制作環境である。これはプログラミングという環境の中の、機械内部に依存した部分をしてできるだけ排除し可能な限りプログラミングの問題の中の「アルゴリズム」というところだけを浮き立たせるようにしたものである。プログラミングの初心者でも比較的容易で、最初はできるだけ直感的に、プログラミングという作業に没入できるような



Programming Baseの構造

仕組みにした。

Playful Designという名前は、前回のReflective Designの次のステップとして、作品づくりを参加者自身がエキスパートのサポートを受けつつ即興的にplayfulに展開していけるよう学習環境をデザインしたことによる。Programming Baseを使用することと、エキスパートが近くにおいて、いつでも質問できるという状況によって、参加者がモノづくりの際の技術的な障壁をあまり意識することなく、作品のできあがりのイメージと実際の作業におけるゴールを参加者自身が最初から思考の中で直感的につなげられるような環境づくりを考え、そこからプログラミングの中のアルゴリズムという側面が浮きあがり、何度もデバッグを繰り返すという環境を参加者自身が自然に体験することにより、参加者の内省的思考を育むことができるのではないだろうかと考えた。またプログラミングにおける内省的思考の重要性と同時に、作品をダイナミックに展開していく姿勢と環境からのサポートによって作品づくりの可能性が大きく広がっていくことを体験してもらうことを目指した。

■workshop 3 : Tangible Design

LEGOブロックというtangible（触知できる）オブジェクトを使って「ブロックを組み立てるための法則（ルール）」を考え、プログラミングのための「ことば」を共同でつくりあげるワークショップである。

ワークショップの形態としては、2人1組になりLEGOブロックで任意の形状を組み立て、その形状を他のグループに「ことば」を使って再現させる

というものである。これを

1. 再現している手元の様子を見て、
再現の状態を確認しながら再現させる。
2. 再現している手元の様子は見ないで
「ことば」のみで再現させる。
3. 「ことば」を紙の上に記述して、
その紙を渡すことによって再現させる。

という3段階で行った。そして最後にその「ことば」を参加者全体で体系化することでそのコミュニティの中での「プログラミング言語」を作成する



ワークショップ風景

ことを目標とした。

Reflective Designでは既存のプログラミング言語の枠にとらわれずに「アルゴリズム的な感性」、
「論理的思考」や「デバッグという考え方」を
学び、Playful DesignではJava言語を使用して実際の
プログラミング体験をすることにより、同様のことを
学ぶことを目的とした。そしてTangible Designで
は実際の「プログラミング言語」を自ら手で作る
ということを体験することで同様のことをさらに深く
学ぶことができ、さらにその重要性や有用性を感じ
とることができるのではないだろうかと考えた。

結果としては、作成された言語は参加者によつて「chovo言語」と名付けられ、それを実際にコンピュータ上で動作する環境を筆者の一人（古堅）がweb上に制作した。（<http://www.furukata.com/chovo>）

■workshop 4 : Dynamic Design

このワークショップは、1メートルを1分で走行するように、LEGOブロックで作成した自動車をコンピュータで制御することを目標にした。そしてハードウェアとソフトウェアの両方を融合させた設計と作成が必要になるようにデザインした。ただし、制御は難しいものと捉えられる傾向があるので、「制御」を「スイッチのON/OFF」であると言いかえることにした。この考えに基づき、ワークショップの前半は、スイッチの働きと構造を再確認するこ

とにした。

参加者は実際にモーターと電池をつないでから、なぜ、つないでいる配線ははずすと、モーターは止まるのかという問題に取り組み、導体と絶縁体の存在を知ることへの動機付けにした。スイッチの構造が分かったところで、日常生活で使っている、紙コップ、割り箸、輪ゴム、アルミホイル等を組み合わせてスイッチを創作した。

後半は、人間よりコンピュータの方がスイッチのON/OFFを正確に行えることを理解した上で、スイッチの操作を人間からコンピュータに変え自動車の制御を行った。

LEGOブロックで作成した自動車をコンピュータで制御するために、筆者の一人（森石）がCube TalkというMacintosh上で動作する制御ソフトを開発した。このソフトウェアはディスプレイの上部に命令を入力するフィールドと、下部の制御用のスイッチで構成されている。この制御用スイッチは容易に確認できる正方形とし、中央にcube kart（スズキ教育ソフト社）のセンサーを取り付けた。このスイッチが黒色のときはOFFであり、白色になればONとなることで、モーターの制御ができる。

命令は、Logoのリスト処理的な処理形態を持ちながら、短時間で理解し操作ができることを主眼に



cube kartのセンサー



ワークショップ風景

て、[ON][OFF][WAIT] の3種類に制限した。

ワークショップの最後には、作成した自動車で、1] 1メートルを1分以上2分以内で走行すること、2] 自動車には2個のモーターを使うこと、3] パソコンで自動車の動作を制御すること等のルールに従って競技を行った。

Aグループは、1メートルをゆっくり1分で走行する自動車を作成し、Bグループは、スタート地点で自動車は停止したままで、通電チェックのためにプロペラの付いたモーターを5.6秒間動作させ、そ

の後一気にゴールに向かうように作成した。Cグループの自動車は、タイヤにいろんなLEGOブロックを付けることによって、滑りながら走行するように作成されている。

これら3グループの制御方法は明らかに違う。Aグループは、自動車が1メートルを1分で走行するための速度を算出しギヤ比を考えた。Bグループは、自動車が1メートルを走行する時間を測定し、1分から走行時間を引いた残りの時間を、スタート地点で止め、その間プロペラを回転させるようにプログラムを考えた。Cグループは、滑りながら走行することを考慮したプログラムを考えた。

このワークショップでは、参加者に明確なゴールと条件を与え、目標達成の方略に集中できるようにした。そのため、同じ素材を使っても、考え方の相違でいろんなアプローチがあるということを参加者全員が感じ取ることが出来たようである。

■結果と考察

今まで行ってきたワークショップから筆者たちが学んだことをもとに、ワークショップ・デザインのための重要なポイントを4つ提案したい。

(1) 課題のデザイン

ワークショップ・デザインを考える時に大切なのは、参加者にどんな課題をどのように与えるかということである。参加者が、まずおもしろそうだと感じると同時に、やれそうだと、やってみようという気持ちになるようにワークショップの課題を提案することが大切である。そしてワークショップ・デザイナーがテーマとして提案した課題を、参加者自らが独自に解釈を行い自分たちの課題として設定できるメタ課題になるものが望ましい。すなわち、参加者が自ら積極的に課題を設定できることや作品をつくりたいような学習環境が必要なのである。

思いを作品にして動作させ、改善を加えながら作品のイメージを固め、自らが設定した課題に向かって進んで行く。この改善のプロセスつまり試行錯誤の中で参加者たちはモノづくりを学んでいくのである。

(2) ツールのデザイン

モノづくりをサポートするための挑発的で直感的なツールの開発が必要である。例えば、Playful Designで開発したProgramming Baseは、アルゴリズム的な感性をはたらかせながら直感的にプログラミングができるようなツールになっている。

(3) 協同作業の場のデザイン

モノづくりを行っていく中で、協同作業を通し

て理解が深まっていくと考え、すべての活動のベースをcollaborationにした。またcollaborationは、自己と他者の考え方を対比しながら、課題を反省的に吟味できる場の形成にもなる。

そして、まわりからのサポートがあれば自分もプログラミングができるのだという自信を持てる雰囲気をつくっていくことができる。

(4) 批判的賞味の場のデザイン

協同作業を通して、ワークショップの場に学びのコミュニティが生まれてくる。アルゴリズムやプログラミングについて「ああでもない、こうでもない」と語る雰囲気が自然に立ち現れてくるのである。参加者たちがコミュニティの中でお互いの作品をクリティカルに味わいながら、モノづくりの楽しさを感じ、自分や他者の作品を批判的・建設的に吟味したり、賞賛したりする姿勢を身につけていくのである。

■今後の課題

「プログラミングを学ぶ」学習環境デザインには、魅力的でplayfulな課題を選び、そのための道具を開発し、挑発的な学びの場をデザインしていくことが大切であると感じている。

当日の発表では、ワークショップのメタアナリシスを通して考えたことをベースに、これからの情報教育のための学習環境デザインの可能性について討論していきたい。

■参考文献

- 森石峰一, 古堅真彦, 上田信行(1999) Dynamic Design 教育システム情報学会発表論文集: 397-398
- 森石峰一, 古堅真彦, 上田信行(1999) Thinking Toys 日本教育工学会第15回全国大会講演論文集: 513-514
- 上田信行, 古堅真彦, 岸本菜穂美(1998) Reflective Design 日本教育工学会第14回全国大会講演論文集: 85-86
- 上田信行, 古堅真彦, 岸本菜穂美(1998) Playful Design 日本視聴覚・放送教育学会発表論文集: 22-23
- 上田信行, 古堅真彦(1998) プログラミングを通じた論理的思考の外在化の実験 情報処理学会発表論文集: 49-56
- 上田信行, 古堅真彦(1999) タンジブル・デザインー Tangible Designー 日本教育メディア学会第6回大会: 31-32