

I/O デバイスやセンサを用いた 造形教育におけるプログラミングの理解

有賀妙子[†] 森 公一[†]

コンピュータやセンサ、I/O デバイスなどを用いた表現は、鑑賞者と作品とのインタラクティブな関係性を実現する。このインタラクティブ性はメディア・アートだけでなく、インタフェースデザインなどの分野においても重要な要素である。メディア・テクノロジーがもたらした表現の特性に注目し、メディア造形教育とでも呼ぶべき基礎的な教育内容の構築を試みてきた。人の行為を検知してプログラムによりグラフィックイメージを生成するインタラクティブインストール作品を生成するのに用いるツールキットを開発し、教材として使用している。インタラクティブなメディア表現を実現するには、コンピュータプログラムが重要な役割をする。本教育内容の中で行ったプログラムの理解、作品への展開について考察する。

Development of understanding computer programs in a basic course for interactive media contents

Taeko Ariga[†] and Koichi Mori[†]

Our study aims to develop a basic course for teaching interactivity in media arts and design. Interactivity produced by media technologies is a crucial element for a broad range of studies including software interface design and product design as well as media arts and design. We developed original toolkits for a training course that enables students to create an interactive installation and learn how to interrelate graphic images generated by a program that senses human actions. This report considers the way of learning computer programs in the course and how to apply them practically to artworks.

1. はじめに

コンピュータ、センサ、I/O モジュールのような情報技術、電子技術を使った表現手法は、アートやデザインの分野に新しい様相をもたらした。この新しい様相は 1990 年代初頭に始まり、多様な表現実験が行われている。とりわけ鑑賞者の身体的行為とそれに呼応する映像や音声のフィードバックを表現の重要な要素とする傾向、すなわちフィジカル・インタラクションを前提とする表現が数多く試みられ、メディア・テクノロジーに特有の感覚領域が実験されてきた。インタラクティブ性は、メディア・テクノロジー、殊にコンピュータがもたらした表現手法の中心となる特徴である。

映像イメージや音声など、鑑賞者の行為に応答するフィードバックをもつインタラクティブアートと呼ばれる作品は、物理的なインタラクションを通し、新しい感覚を鑑賞者に提供する。この行為と感覚の関係は、固定的なものではなく、本来の関係性を変容、増幅することで、新しい感覚的な体験をもたらす。また、インタラクティブ性はメディアアートだけでなく、ヒューマンインターフェースデザインのような分野の研究においてもキーとなる要素である。

メディア・テクノロジーが提供するインタラクティブ性の表現特性に注目し、メディア造形教育とでも呼ぶべき基礎的な教育内容とそのためのツールキットを開発し、実践してきた[1]。これをセンサリビジョン(Sensory Vision)プロジェクトと呼んでいる。本報告ではそのうち、コンピュータプログラムの理解に着目し、その効果について考察する。

2. プロジェクトの目的

プロジェクトでは、音センサ、赤外線センサなどを用い、インタラクティブなメディア造形作品(インストール作品)の制作を行う。センサからの入力に応答して動的に生成されるグラフィックと人の行為がインタラクションをとることで鑑賞する作品である。

画家が絵画制作の前段階において行うデッサンあるいは習作を通じて、光や色彩、形態や質感などの感性的次元を獲得するように、メディア・テクノロジーを前提とする表現手法の領域で、感性的次元の気づき、制作力獲得のためのトレーニングを、初習者向けの基礎的な教育プログラムとして構築した。デジタル・メディア・エイジにおける造形教育基礎課程と位置づけられ、この教育プログラムを通して、入力とそれに

[†] 同志社女子大学 学芸学部 情報メディア学科
Department of Information and Media, Doshisha Women's College of Liberal Arts

応じた出力といった単なるメカニズムにとどまらない、固定的な意味関係を越えた新しい行為と感覚の関係に、学生たちが気づくことを期待している。

加えて、メディア・テクノロジーの扱いに関する知識やスキルの向上も目指し、表現的側面と技術的側面の双方の力の育成を目的とする。技術的側面では、プログラミングの理解とスキル獲得が重要な位置を占める。

3. プロジェクトの内容

3.1 テーブル・インタラクション

行為と感覚がインタラクティブに関係し合う場として、白い木製のボックス(W500×D500×H900 mm、図1)を用意した。天井から吊るしたプロジェクタの下にボックスを設置し、天板に映像を投影する。鑑賞者が天板を触る、たたくなどの行為を行うと、それに応じたグラフィックイメージが生成、投影される。つまり、天板は鑑賞者とのインタラクションのテーブルとして機能する。箱の内部にはコンピュータやI/Oモジュールなどを置くことができる。天板は箱本体から取り外しが可能で、穴を開けるなどして、センサを埋めこむことが可能である。

テーブルは触る、たたく、なでる、息を吹くなどさまざまな行為を誘発し、鑑賞者のそのような行為に応じて、生成されたグラフィックイメージが動き、変化し、それ



図1 インタラクションの場

が多様な感覚経験を生み出す。それは、単なる原因と結果という固定的なメカニズムではない。例えば、赤外線(距離)センサからの入力に応じて円の直径が変わるインタラクティブなグラフィックイメージを考える。上下する手の動きにリンクした円の大きさの変化、それ自体は単純なメカニズムであるが、感覚的な経験はそれほど単純ではない。インストールの前で実際に体験すると、重み、あるいはあたかも手のひらから何かしらの力が発生しているような感じが生じる。これが行為と感覚の関係がもたらす、新しい感覚経験で、プロジェクトでは、このテーブル・インタラクションをいわば制約として、行為-感覚の関係をシンプルな構成の中で追求する。

3.2 ツールキット

人の行為を感知し、それを処理するには、センサやアナログ-デジタル(AD)変換に関係する最小限の電子工学、ハンダ工作そして、プログラミングなどの知識が必要となる。しかし、デザインや芸術、人文科学を学ぶ学生は多くの場合、そのような知識を持っていない。そのような学生にとって、センサや抵抗などの電子部品を集め、組み立て、入力を読み込むための簡単なプログラムを書くことは大変困難である。これを避けるため、ハードウェアとソフトウェアそれぞれツールキットを開発した。

ツールキットは学生が、人の行為をセンシングし、それをデータ処理して、応答を作り出す実験を簡単に行う、そしてそれに基づき、プロトタイプを開発する手段を提供する。

(1) ハードウェアツールキット

ハードウェアツールキットはI/Oモジュール、センサなど入出力用部品、ケーブルなどをパッケージしたもので、センサとコンピュータを接続し、入力を得るために必要となるデバイスと電子部品を含んでいる。図2にツールキットのパッケージの中身を示す。センサからのアナログ入力をデジタル信号に変換しコンピュータへ送る、またコンピュータからの信号を出力するため、オリジナルのI/Oモジュールを開発した(Sensory Vision(SV)モジュールと呼ぶ、図3)。ArduinoやGainer[2]など類似機能をもつI/Oモジュールは存在するが、ブレッドボードが必要である、あるいは接続が不安定になりやすいといった、本プロジェクトにとって実装上の欠点がある。SVモジュールはセンサとコンピュータ間の通信機能を実現する最小限の機能に絞る方針で設計された。4つのアナログ入力ポート、4つのデジタル出力ポートを備え、擬似RS-232シリアル信号で通信し、ハンダ工作をせずに、センサとI/Oモジュールを安定して接続できるように、センサ部品にはミニピンプラグをつけ、I/Oモジュールの入出力ポートはミニピンジャックとした。さらにセンサには必要に応じて抵抗などの部品もつけ、プラグを差し込むだけで、入力を得られるように工夫した。仮にI/Oモジュール



図 2 ハードウェアツールキットの内容

と裸のセンサが教材として与えられたとしても、ほとんどの学生にとっては、それらを使って基本的な入力実験を行うことさえ、ハードルが高い作業になる。

パッケージされているセンサは(図 2 で左 2 つめから順に)光センサ、距離センサ、音センサ、ボリューム(可変抵抗)である。SV モジュールは、センサからの 0-5V の値を 0-255 の 8bit 信号として送る。図 2 の一番左は、ツールキットに含まれる唯一の出力部品である LED で、チュートリアルで実験に使用する。

(2) ソフトウェアツールキット

本プロジェクトでは、プログラミング環境として Processing を使う[3]。Processing はグラフィックイメージを扱うことに特化して、Java 言語を使って開発されたもので、アニメーションやインタラクティブなグラフィックコンテンツを比較的容易に作成でき、Processing を用いて多くのメディアアート作品が制作されている。

どのような動的グラフィックをプログラムで生成するかに関しての着想を学生に与えるため、Processing で書いた 2 種のサンプルライブラリを用意した。基本運動のライブラリと、センサリインタラクションのライブラリである。前者は、グラフィックオブジェクトの基本的な運動を示すためのサンプルプログラム集で、学生にプログラムによって動的グラフィックスを生成する方法の基本を示す。グラフィックオブジェクト(例えば円)の運動を、一方向の運動、往復運動、回転、振動、拡張の 5 つの基本パターンに分類し、それぞれのパターン内で軌道や速度などを変えた 70 種の運動のプログラムを用意した。

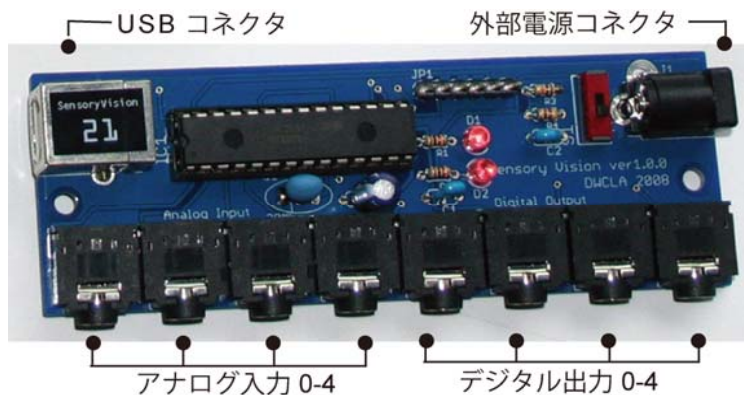


図 3 Sensory Vision I/O モジュール

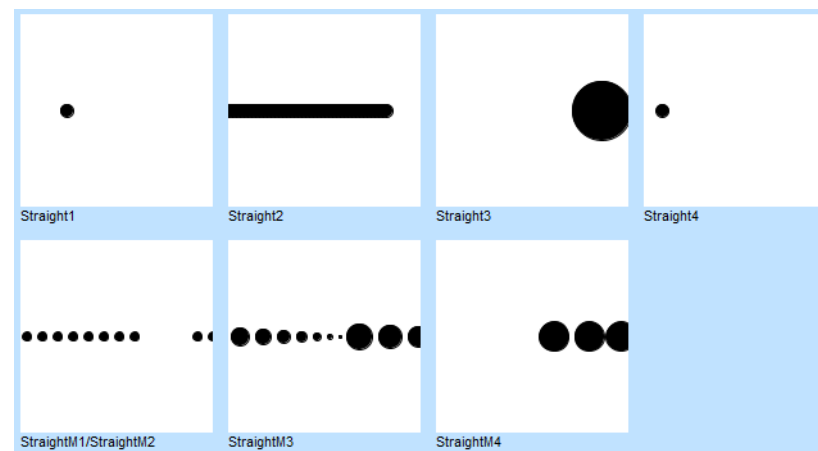


図 4 一方向の運動(直線)のプログラム実行例

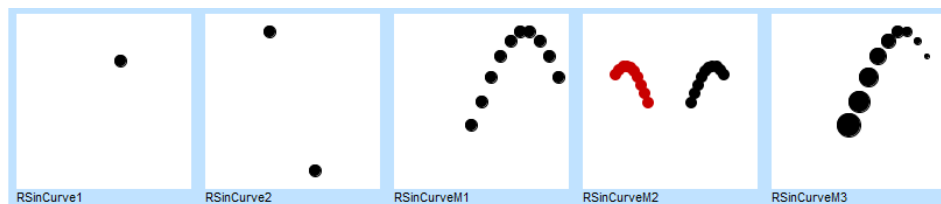


図5 往復運動(波)のプログラム実行例

例えば、一方向の運動パターンには、直線、波、円、螺旋の軌跡で運動するプログラム、さらに加えて等速/加速運動、大きさやオブジェクトの数を変えて運動するプログラムが含まれる。図4は一方向の運動のうち、軌道が直線のプログラムの実行結果のスナップショットで、左上から順に、等速運動、軌跡を残す等速運動、大きさが変化しながら等速運動、加速運動、同じ大きさの円列の移動、大きさが異なる円列の移動、大きさが変化しながら移動する円列である。

図5は、往復運動のうち軌道が波の運動で、左から順に、円が左右の壁で往復、2つの円が中央で衝突、同じサイズの円の列が左右の壁で往復、2つの円の列が中央で衝突、円列が大きさを変化しながら往復という動きをする。

このような単純な動きを生成するプログラムを実行し、オブジェクトの動きが人の感覚にもたらす効果を確認する。大きさが変化することで奥行きが感じられたり、加速することで力を感じたりすることを観察する。

センサリインタラクションのライブラリは、動的にグラフィックを制御するための方法を示すためのサンプル集で、20種のプログラムを含み、次の4種に分けられる。

- ① センサからの生データを別の範囲の値にマップして、大きさ、速度、角度、長さなどの変化に使う。例えば音センサからの180-220の値を、100-0の値に変換し、オブジェクトの大きさの値とする場合、音センサのノイズを無視し、音の大きさとオブジェクトの大きさを反比例させることになる。
- ② センサからの値をスイッチとして使う。閾値を設定し、それによって運動を開始する、あるいは止める制御をする。
- ③ 時間経過を計測することでタイマー機能を実現する。例えば、入力がある閾値を越えた後動きを開始し、一定時間後に止めるというようなサンプルである。
- ④ 物理的力場における粒子の位置を計算する粒子システムを使う。

図6は、センサからの入力に反応するサンプルプログラムの実行例で、(i) 入力値の変化量で描く円の大きさが変化する、(ii) 入力値が閾値を超えると円が弾けるように

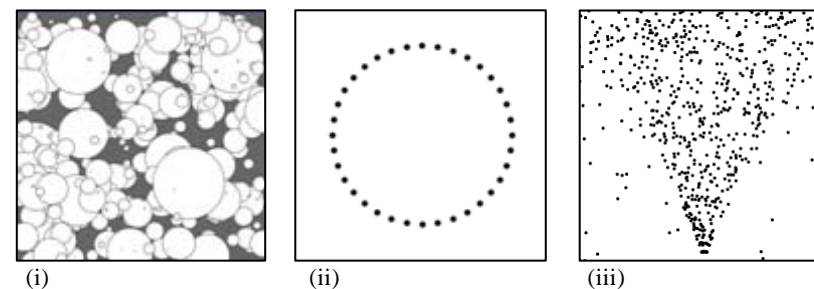


図6 センサリインタラクションライブラリの実行例

広がる、(iii) 噴水のように粒子が吹き出し、入力値により高さが変化する。

両ライブラリのサンプルプログラムに登場するオブジェクトは円だけである。それには次のような理由がある。本センサリビジョンプロジェクトに先立つ研究として2年間の類似プロジェクトを実施した[4]が、そこではアクチュエーターのような物理的応答部分を持ったり、キャラクタなど具象的グラフィックを使ったりしたインタラクティブ作品が制作された。それらのインタラクティブ性は結果として説明的な文脈あるいは単純な機械的応答に依ったものばかりであった。物理的な物の存在や具象的なイメージがその傾向を助長していると考えられた。そこで、本プロジェクトではそれを避けるため、応答部分は映像のみとし、かつシンプルな幾何学的形態のオブジェクトの運動だけを使って、新しい感覚体験を探求することを目指した。それを受けて円の運動に絞ってサンプルプログラムを開発した。

ライブラリは、学生にプログラムの書き方を理解させる役割があるとともに、それぞれのサンプルプログラムが示す運動を観察することで、オブジェクトの運動が人の感覚にどのような刺激を与えるかを考察するのを手助けする。学生は、ライブラリ中のサンプルをセンサを変えて実行する実験を行い、そこから人の行為とグラフィックの動きとの関係を発見していく。そのような実験と観察の後、学生はサンプルプログラムを変更、拡張、結合などして、自分たちのコンテンツを形作る。

3.3 授業プロセスと作品例

センサリビジョンプロジェクトの授業では、ひとつのインタラクティブインストールを、2名がチームとなって、次のようなプロセスで制作、発表する。

- (1) プロジェクトの目的的理解

- (2) ハードウェアツールキットと基礎エレクトロニクスの理解
- (3) コンピュータプログラミング(Processing)の理解
- (4) ハードウェア・ソフトウェアツールキットを使った実験
- (5) 行為と感覚の記述演習
- (6) インタラクションの構想設計
- (7) 実装
- (8) 作品プレゼンテーション
- (9) 作品展示

2008年と2009年の2年間、約30名のクラス(2年次生)に対して、3.2節で説明したツールキットを使い、上記のプロセスの(1)から(8)を半期14回の授業の中で実施し、その後学内ギャラリーで展示した。

2008年の16のチームのうち、13チームが音センサを使い、テーブルを叩く、弾く、こする、引っかく、あるいは息を吹きかけるといった行為を誘発するコンテンツを制作した。また、4つのチームが赤外線センサを使い、手を置くあるいはかざすといった行為に基づいたコンテンツを制作した。2009年は15チームの内、12チームが赤外線センサを、8チームが音センサを、4チームが光センサを使った。2008年は音センサを使った作品が多かったが、これらの作品では鑑賞者がバンバンと強くテーブルを叩く行為をする傾向があり、行為と視覚のフィードバックによる繊細な感覚の体験を妨げることになりがちだった。これを受けて2009年の開始時に荒い行為は適切ではないことを指摘し、その結果柔らかい、ゆっくりとした行為を想定した赤外線センサの使用が2009年に増えた。表1に、チームが作品に採用した人の行為とインタラクションの例を、センサの種類とともに示す。

図7は表1のaの展示時の写真で、テーブルを叩くことで、半透明の円が現われ、テーブルを真っ赤に染めていく。鑑賞者は絵具の入った見えない風船が破れて、赤い絵具がテーブルに飛散するような感覚をもつ。図8は表1のcで、赤い円の動きがテーブルをこすったり叩いたりすると変化する。距離に応じているだけだが、円周上を移動する速度が変化したり、直線的に動いたり、手の動作に応じて円がじゃれて遊んでいるような印象をもたらす。図9は表1のdでテーブルをカリカリと搔くと、白いぎざぎざの線が縦横に現われる。ペンキを塗るようにあるいは拭くように線がテーブル面を埋めつくすまでやり続けてすっきりするという感覚を狙っている。

表1 採用された行為とインタラクション

	作品名	鑑賞者の行為 (使用したセンサ)	インタラクション
a	散	叩く、弾く (音センサ)	ランダムに円が現われ、重なる。絵具が弾け飛ぶように全体が真っ赤になっていく
b	The steam	吹く (音センサ)	ふたりの鑑賞者が息を吹きかけると、蒸気が揺らめく。
c	戯れ	叩く、こする、上下に動かす (赤外線センサ)	鑑賞者の手に、赤い円がじゃれつくように動く。
d	delete	引っかく、叩く (音センサ)	ぎざぎざの白い線が黒い画面に現われる。カリカリと引っかき、面を白くする。
e	Perfume	手をかざす、押さえる (赤外線センサ)	円が外周へ向けて弾ける、集まる、あふれ出す。しみだす匂いを手で塞ぐ感覚。
f	KINGYO	手を近づける、手を打つ (音センサ、赤外線センサ)	朱と黒の円が集まる、回転する、散る。水槽の金魚を覗いて遊ぶ感覚。
g	弾	手を押し付ける (赤外線センサ)	赤い円の不透明度と大きさが増加、その後多数の小さい円が飛散る。ペンキが染み出し、弾け飛んで、面を埋め尽くす。

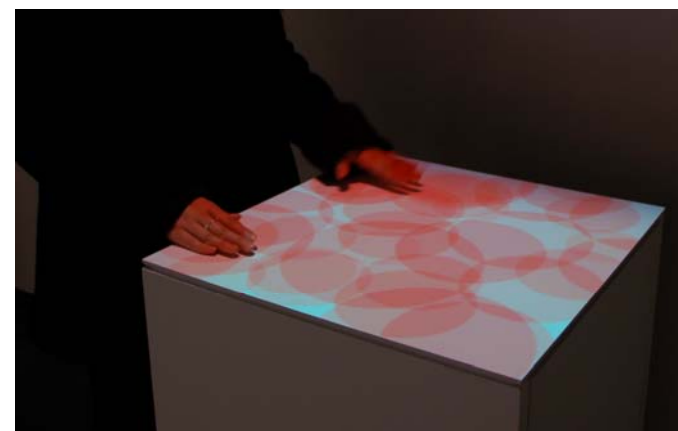


図7 作品例： 散

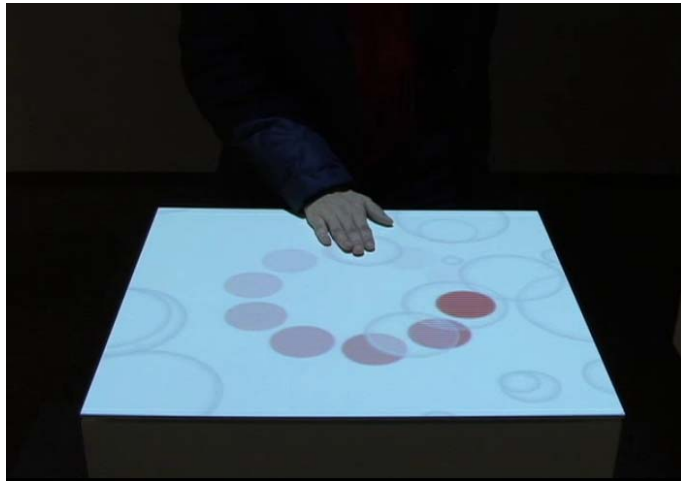


図8 作品例：戯れ

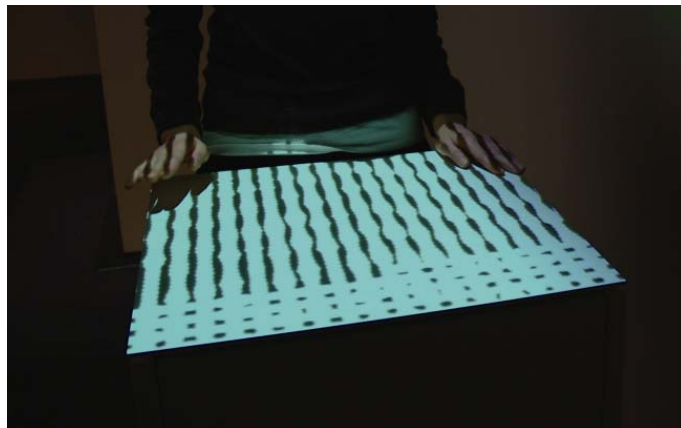


図9 作品例：delete

4. プログラミングの理解

4.1 プログラム要素の学習

14回の授業のうち、4回をプログラミングの理解に割り当て、次のような内容を説明した。

- ・変数
- ・メソッドの実行
- ・算術演算と代入
- ・制御(if else 文と for 文)
- ・配列
- ・位置の計算(等速・加速運動)
- ・クラスの活用
- ・音声ファイルの再生
- ・画像ファイルの描画
- ・センサからの入力値のマッピング(比例変換)
- ・粒子システムの利用

プログラムの学習は、3.2節の(2)で述べたサンプルライブラリをまず使ってみるところから始め、説明されたプログラム要素がどう使われているかを理解する。その後自分の作品企画に合わせて参考となるサンプルを変更し、プロトタイプとしてテストする。「真似る」から「変更して再生産する」、そしてオリジナルな動きを「創作する」に至ることを期待する。

4.2 学生の理解度の自己認識

プロジェクト開始前にプロジェクトへの期待と自分の技術的な知識を判定するアンケートを、また終了後にプロジェクトの振り返りと同じ技術的知識を判定するアンケートを実施した。終了後アンケートでは、プロジェクトはおもしろいか／おもしろくないか、新しいことを大いに学んだか／学べなかったかを10段階で判断し、加えて自由記述でその理由の記入を求めた。回答の散布図を図10に示す。教育的プロジェクトとしては新しいことを大いに学びかつおもしろかったと振り返ることが望ましいが、両者には相関はない。2つの項目に極端に差のある回答をした学生のコメントを見ると、学んだの自己評価が10、おもしろいが3([10,3]と記す)の学生は、「視覚的に受ける印象は興味深かったが、制作過程は本当に意味がわからずつらかった」とゼロからかたちを作り出すプロジェクト型の学びに苦勞した。また[2,8]の学生は、「プログラムが映

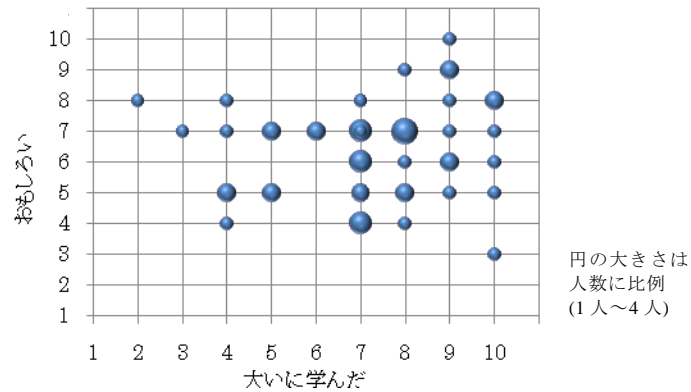


図 10 新しいことを学んだ—おもしろい

像として出てくるのが面白かった」が「プログラミングは結局理解できなかった」というプログラミングに対する否定的な気持ちでプロジェクトを終えている。

技術的な知識を判定する設問はプロジェクト開始前と終了後同じ内容で、プログラムの記述やセンサ、A/D 変換など技術的な事柄について、自分の知識量や理解度を 10 段階でどの程度かを回答する。このうち、プログラムを書くことについて、「よく知っている」から「全く知らない」の 10 段階で自己判定した結果のプロジェクト前後での変化を表 2 に示す。49 人中 26 人(53%)がプログラムについての知識理解が 2 段階以上増加したと回答した。

Java プログラミングの半期科目を履修済みの学生が全体で 13 人いる。この 13 人中プログラム理解度の自己認識が 2 以上増加したのは 5 人(32%)、1 の増加は 6 人(46%)であった。一方未履修者 36 人では 2 以上増加したのは 21 人(58%)、1 の増加は 8 人(22%)であった。予想されることであるが、プログラミングの初学者の理解度認識の方が、既習者よりも大きい増加を示している。

表 2 プログラムの理解度のプロジェクト前後での変化

変化	2008 年(回答数: 28 人)		2009 年(回答数: 21 人)	
増加	22 人 (79%)		17 人 (81%)	
	2 以上の増加	13(46%)	2 以上の増加	13(62%)
	1 の増加	9(32%)	1 の増加	4(19%)
変化なし	2 人 (7%, 2 と 3)		2 人 (9.5%, 2 と 4)	
減少	4 人 (14%, -1)		2 人 (9.5%, -1: 1 人, -2: 1 人)	

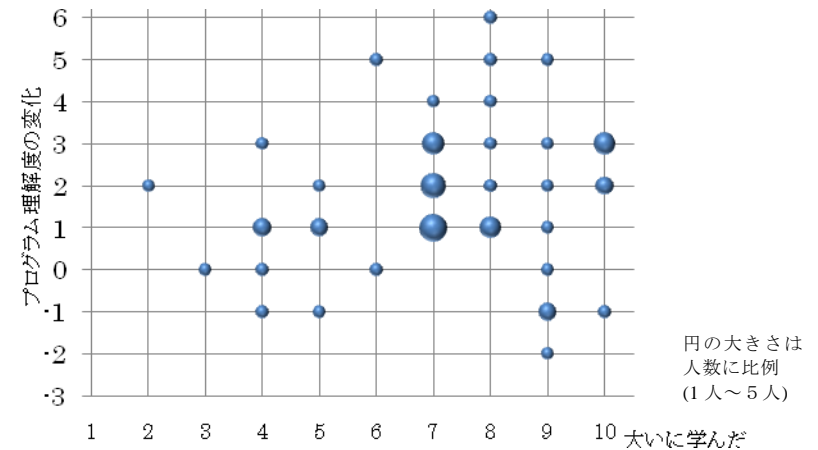


図 11 大いに学んだ—プログラミング理解度の変化

図 11 は、回答のあった 49 名の理解度の変化量を、大いに学んだかどうかの程度を横軸にプロットした散布図である。大いに学んだ感が 7 以上の学生 36 名(高グループ)の理解度変化の平均は 1.94 で、5 以下の学生 11 名(低グループ)の理解度変化平均は 0.82 であった(49 名全体の平均は 1.71)。一方、2 つのグループがプロジェクトに感じているおもしろさには差はない(全体平均: 6.45、高グループ: 6.42、低グループ: 6.45)。

低グループの学びに関する自由記述には「プログラミングが難しくよくわからなかったのがつらかった」「プログラミングが複雑で、あまり自分たちで作り上げた気がしなかった」などプログラミングに関しては、ネガティブなコメントだけで、ポジティブなコメントはまったくなく、プログラムの困難さを認識した結果が学んだ感を減少させている。これに対し、高グループでは、「持っている能力が少なすぎて、出来るものが少なかったのが残念」というネガティブな記述とともに、「自分が思ったような動きが実現した時はとてもうれしい」といったプラスのコメントがあり、プログラムを通して新しい表現(このプロジェクトではオブジェクトの運動)を探る主体感をもっている。

4.3 作品に見るサンプルプログラムの流用

インタラクティブなメディア作品としての魅力とプログラムの記述の適切さとは、必ずしも一致しない。サンプルプログラムをほとんどそのまま流用している場合でも、行為とグラフィックの関係により新しい発見や体験を提供する作品もある。サンプルプログラムをテストするところから始まるプログラミングの学習は、次のような 3 つ

のフェーズをいききしながら進む。

- (i) 模倣する
- (ii) 再生産(変更)する
- (iii) 創作する

プログラムソースコードを把握できている 24 の作品を対象に、オリジナルな動きやサンプルプログラムの流用について分析し、3 つのどのフェーズにあたるかを判定した(表 3)。サンプルのクラスを直接使用しているのが模倣フェーズ、サンプルのクラスを変更あるいはサンプルコードを一部流用し、かつオリジナルな動きを含まない作品を再生産フェーズ、オリジナルな動きを実装した作品を創作フェーズとした。創作フェーズの 12 作品のうち、ソースレベルでプログラムの流用が見られるのが 6、流用がないのが 6 であった。75%の作品がサンプルプログラムを明確に流用するかたちで制作されている。

表 3 プログラムから見た作品の進化

フェーズ	作品数
模倣	3
再生産 (変更)	9
創作	12 (流用あり:6, 流用なし:6)

図 7 と図 8 の作品は再生産フェーズのもので、図 7 は、センサリインタラクショナルライブラリのサンプル(図 6 の i)を参考に円をランダムに重ねて描く。サンプルのコードを流用、修正し、見えないボールを潰すというオリジナルな体験感覚を作り上げている。また、図 8 は、サンプル(図 6 の i)に加え、基本運動のライブラリから、円を軌道として運動するプログラム、直線を往復運動するプログラムを組み合わせ流用し、独自クラスを定義して実現している。図 9 は創作フェーズの作品例で、直線一方の運動プログラムを出発点として流用しながら、オリジナルな新しいプログラムを創作した。

プログラミング初学者が中心であるクラスで、プログラムの必要要素を理解し、制御プログラムを作成するにあたり、サンプルプログラムライブラリは、「真似て」、「学ぶ」出発点として重要な役割を果たしている。

5. 考察

非コンピュータ科学の文脈の中でプログラムを学ぶ利点は、いままで多く論じられ

ている[5,6,7,8]。メディア技術を表現の手段とした造形基礎教育であるセンサリビジョンプロジェクトにおいて、プログラミングの学習から得られるメリットは、次の点にある。

- (a) 制作の幅を広げる。
プログラムの知識を必要とする制作ツールを今後の制作に選択できる、またそれを使用する際の理解を容易にする。
- (b) 数理的な思考力を鍛える。
数理的手段を踏んで問題を解決する方法を見つけ出し、実現する経験をする。
- (c) 違いを生み出す力を実感する。
既存のアプリケーションが提供する表現方法を使用するだけでなく、違いを生み出す達成感、自分がやったという主体感をもつ。

本プロジェクトでのプログラミングの学習で、80%の学生がプログラミング理解が向上したとの自己認識を持って終えているが、上記の点を実現したか考える。プロジェクト終了後の、何を学んだかについての自由記述アンケートで、プログラミングについてコメントを書いた学生 48 人のうち、ポジティブなコメントだけを書いた学生は 10 名、ネガティブなコメントだけをしたのは 9 人、両方のコメントを書いた学生は 29 人であった。コメントは次のようなものである。

【ポジティブなコメント】

- ・思った通りに動きが実現してうれしい、楽しい
- ・プログラムを変更していくうちにおもしろい動きを発見できた
- ・思う通りにならなくても、それに近づけようとする努力、くじけない心を学んだ
- ・自分にも(助けはいるが)出来るという実感
- ・広い分野への可能性が感じられ新鮮

【ネガティブコメント】

- ・難しくよくわからなかったのがつらかった
- ・やりたいこととプログラムできることとのギャップが大きかった

自己の振り返りコメントを見ると、8割の学生が上記(a)–(c)のいずれかのメリットを体験し、認識することができている。しかし、同時にプログラミングに苦勞もしている。否定的なコメントだけを書いた学生のプログラミングの理解度の変化が皆低いわけではないが、自分でやったという主体感を得られていない。プロジェクトの成果に見られるプログラムのレベルや習得度以上に、その後プログラミングを自分の道具として捉えられ、制作の幅を広げられるかどうか重要だが、このような学生に対してはプログラミング学習のメリットを受けるに至っていない。

プログラムを学ぶ困難さを避けて利点を享受する目的で、主に年少者向けに Scratch

や Etoys など、ゲームやアニメーションのようなインタラクティブメディア用のさまざまなプログラミング環境が開発されている[9]。本プロジェクトで使用しているプログラミング環境 Processing はグラフィック描画やアニメーションを容易に記述できる独自文法と Java 言語の双方を使用でき、プロトタイプの作品にとどまらず、実用的な作品にも利用できる長所がある。一方、テキストでコーディングするという伝統的なプログラミング言語の範疇にあり、ビジュアルプログラミング環境と比較すると学ばずともプログラムできるという簡便さはない。本プロジェクトでは、プロジェクト後の実際の作品制作への展開を考え、拡張性、応用性の高い Processing を選択した。

プログラミング学習の困難さを補うため、サンプルライブラリを用意した。プログラムコードの記述とオブジェクトの運動との直感的関連性がプログラム要素理解の助けになるとともに、真似て、再生産し、創作へつなげるプログラムの出発点として、またシンプルなグラフィックの運動によるインタラクションが人の知覚にどう捉えられるかの発想の基本として、有効に機能した。しかし、さらにプログラミング理解の難しさを軽減するため、サンプルプログラムの工夫を進めて、プログラムに対してネガティブなコメントだけを残してプロジェクトを終えることのないよう改良を加える必要がある。

謝辞 制作キットの開発、プロジェクトの実施における、真下武久氏、作花愛梨氏の協力を感謝する。なお、本研究は同志社女子大学研究助成金の支援を受けた。

参考文献

- 1) Ariga, T. and Mori, K.: Sensory Vision- Development of a Course for Physical Interaction and Graphics, Computers & Graphics, Vol.34, No.6, pp.800-810 (2010).
- 2) Kobayashi S, Endo T, Harada K, and Oishi S.: A Reconfigurable I/O Module and Software Libraries for Education, NIME06, IRCAM (2006).
- 3) Processing, <http://processing.org/>.
- 4) 有賀妙子, 森公一: メディアデザイン基礎教育プログラム - I/O モジュールを使ったインタラクションのデザインと制作, デザイン学研究 第 54 回研究発表大会概要集, pp.118-119 (2007).
- 5) Seymour P.: MINDSTORMS Children, Computers, and Powerful Ideas, Basic Books, New York, Second Edition (1993), 邦訳 マインドストーム—子供、コンピューター、そして強力なアイデア, 未来社 (1995)
- 6) National Research Council Committee on Information Technology Literacy. Being Fluent with Information Technology, National Academy Press, Washington, D.C (1999).
- 7) Wolz, U.: Teaching Design and Project Management with Logo RCX Robots, In Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education, pp.95-99 (2001).

- 8) Computational Thinking for Youth, The ITEST Small Group on Computational Thinking, White Paper Working Group, <http://itestlrc.edc.org/resources/computational-thinking-youth-white-paper> (2010)
- 9) Resnick, M. et al: Growing Up Programming: Democratizing the Creation of Dynamic, Interactive Media, CHI '09, In Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (2009).