



Title	メディアコンテンツの制作力を育成する情報教育プログラムの開発と実践に関する研究
Author(s)	有賀, 妙子
Citation	大阪大学, 2015, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/53935
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

メディアコンテンツの制作力を育成する
情報教育プログラムの開発と実践に関する研究

提出先 大阪大学大学院情報科学研究科

提出年月 2015 年 7 月

有賀 妙子

関連論文リスト

学術論文

1. 有賀妙子, 森公一, 大下福仁, 角川裕次, 増澤利光："フィジカル・インターラクションを使ったプログラミング学習のための教材開発と実践",情報処理学会論文誌コンピュータと教育(ショートペーパー),採録決定.
2. T. Ariga, T. Watanabe, T. Otani, and T. Masuzawa: "Learning program for enhancing visual literacy for non-design students using a CMS to share outcomes", International Journal of Technology and Design Education, 16 pages, DOI:10.1007/s10798-014-9297-5, Dec. 2014.
3. 有賀妙子, 森公一："フィジカル・インターラクションを使ったメディア造形基礎教育におけるプログラミング学習の実践", 情報処理学会論文誌, Vol.52, No.12, pp.3096-3105, Dec. 2011.
4. T. Ariga, and K. Mori:"Sensory vision-Development of a course for physical interaction and graphics", Computers & Graphics, Vol.34, No.6, pp.800-810, Dec. 2010.

国際会議

5. T. Ariga, T. Watanabe, and T. Otani: "Look and think exercises on visual communication design for non-designers" (Poster), Proceedings of the 2nd International Conference on Design Creativity, pp.99-107, Glasgow, UK, Sep. 2012.
6. T. Ariga, and K. Mori:"Learning course for sensory interaction", ACM SIGGRAPH ASIA 2009 Educators Program, Article No.11, 7 pages, Yokohama, Dec. 2009.
7. T. Ariga, and K. Mori: "Sensory vision – development of toolkits and training course using an I/O module and sensors", Proceedings of International Symposium on Electronic Art (ISEA2008), pp.32-34, Singapore, July 2008.

著書

8. 有賀妙子, 渡部隆志, 由良泰人："表現メディアの編集と表現", 分担執筆 pp.6-12, pp.128-172, 実教出版, Mar. 2015.
9. 森公一, 有賀妙子："インタラクティブなかたちの生成, インタラクティブなかたちの多様性", かたち創造の百科事典, pp.196-199, 丸善出版, Oct. 2012.
10. 立木秀樹, 有賀妙子："すべての人のための Java プログラミング 第 2 版", 分担執筆 pp.111-244, 共立出版, Sep. 2007.

研究報告

11. 有賀妙子, 渡部隆志, 大谷俊郎："CMS を活用した, 非デザイン学生向けビジュアルデザイン思考育成のための基礎教育プログラム", 情報処理学会第 76 回全国大会講演論文集, 第 4 分冊, pp.401-402, Mar. 2014.
12. 渡部隆志, 有賀妙子, 大谷俊郎："情報デザイン能力育成のための基礎教育プログラム開発, 情報[世界]を見る・考えるデザイン演習", デザイン学研究, 第 58 回研究発表大会概要集, pp.248-249, June 2011.
13. 有賀妙子, 森公一："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", コンピュータと教育研究報告, 情報処理学会研究報告, Vol.2010, No.6 (コンピュータと教育研究会報告 2011-CE-108), pp. 1-5, Feb. 2011.
14. 森公一, 有賀妙子："フィジカル・インタラクションと感覚教育 I/O デバイスやセンサを用いた造形教育プログラムの開発", デザイン学研究, 第 56 回研究発表大会概要集, pp.68-69, June 2009.
15. 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, June 2007.
16. 有賀妙子："デザイン・メディアアート関連学科におけるプログラミング教育-Java プログラミング教育の教材として使える外部入力デバイスの比較検討", 日本デザイン学会誌デザイン学

研究, 第 52 回研究発表大会概要集, pp.198-199, June 2005.

17. H. Tsuiki, and T. Ariga: "Programming for students of information design", ACM, SIGCSE Bulletin, Vol.33, No.4, pp.59-63, Oct. 2001.

その他の発表論文リスト

学術論文

18. T. Ariga, and T. Watanabe: "Teaching materials to enhance the visual expression of web pages for students not in art or design majors", Computers & Education, Vol.51, No. 2, pp.815-828, Sep. 2008.
19. B. Susser, and T. Ariga: "Teaching e-commerce web page evaluation and design: a pilot study using tourism destination sites", Computers & Education, Vol.47, No.4, pp.399-413, Dec. 2006.
20. 有賀妙子, 吉田智子: "ネットワークリテラシー教育の授業設計と教材開発"(資料), 日本教育学会論文誌, Vol.27, No.2, pp.181-190, Oct. 2003.

著書

21. 有賀妙子, 吉田智子, 大谷俊郎: "改訂新版インターネット講座", 分担執筆 pp.178-259, 北大路書房, Feb. 2014.
22. T. Ariga, and T. Yoshida: "Course design and teaching materials for network literacy" in Peer-to-Peer Networks and Internet Policies, pp.135-158, Hauppauge NY:Nova Science Publishers, Inc. Jan. 2010.
23. 有賀妙子, 渡部隆志, 由良泰人: "マルチメディア表現", 分担執筆 pp.6-12, pp.131-189, 実教出版, Mar. 2005.
24. 有賀妙子, 吉田智子: "新・インターネット講座", 分担執筆 pp.73-219, 北大路書房, Jan. 2005.

25. 立木秀樹, 有賀妙子："すべての人のための Java プログラミング", 分担執筆 pp.98-217, 共立出版, Oct. 2000.
26. 有賀妙子: "Java のキホン", 秀和システム, Sep. 2000.
27. 有賀妙子, 吉田智子："学校で教わっていない人のためのインターネット講座", 分担執筆 pp.22-49, pp.85-188, 北大路書房, Sep.1999.
28. 有賀妙子, 竹岡尚三："Java プログラミング徹底マスター", 分担執筆 pp.1-270, pp.283-301, ソフトバンクパブリッシング, Apr. 1998.

研究報告

29. 渡部隆志, 有賀妙子："LineViewer の開発とコンテンツ制作", 情報処理学会ヒューマンインターフェース研究報告, Vol. 2005, No.95 (ヒューマンインターフェース研究会 2005-HI-115), pp.1-5, Sep. 2005.
30. 有賀妙子："Web ページ制作演習におけるページデザイン技法の学習", 情報処理学会第 60 回全国大会講演論文集, 4-307, Mar. 2000.
31. 有賀妙子, 吉田智子："ネットワークリテラシー教育のシラバスと教材研究", 日本教育工学会第 15 回全国大会, 2aC-08, Oct.1999.
32. 有賀妙子："初等 Java プログラミング教育におけるデザインパターンの学習", 情報処理学会第 56 回全国大会講演論文集, 4-349, Mar. 1998.
33. 有賀妙子, 吉田智子："チェックリストを使った Web ページの批判的評価とその効果", 情報処理学会第 56 回全国大会講演論文集, 4-260, Mar 1998.

内容梗概

21世紀の情報化社会では、「情報処理」の技術や知識は限られた専門家にだけ属するものではなく、誰もがそれらを理解し、アクセスできる状況が浸透している。専門家ではない人たちが、情報を表現したり、データを処理したり、グラフィックスを加工、制作できる環境にある。学生たちがメディアコンテンツの消費者の立場だけでなく、制作者としての視点をもって行動できることは、学びやその後の仕事にとって大きなメリットが生まれる。そのために求められる教育内容には、メディア・テクノロジーを理解する力、使いこなすスキル、コミュニケーションのため情報を表現する力の育成がある。特に情報の表現に関する制作力を身に付けることで、消費者から制作者へと学生が変わることが期待できる。本論文では、メディア・テクノロジーを使った基礎的制作力の育成を念頭に、インタラクティブ性、プログラムによる表現、グラフィックイメージによる意味伝達に焦点をあて、2つの教育プログラムの構築、学習環境の開発と実践を行った。

ひとつは、コンピュータの特徴的要素であるインタラクティブ性の探求と、プログラムを表現の手段として使えるようになる基礎力の養成の2つの目的をもつ教育プログラム(SV [Sensory Vision] 教育プログラム)である。そのために、音センサ、赤外線センサなどを用い、新しいインタラクティブな経験を人に提供するメディアインスタレーション(グラフィックスや音声を使った体験型の立体展示作品)の制作を行う。I/O モジュールやセンサ類を含むツールキット、展示環境などのハードウェアを開発し、教育実践に適用した。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の学習が必要である。模倣する→再生産する→創作するという段階を通してプログラミングの理解を進める目的で、その出発点となるサンプルライブラリを用意した。本論文では、それらを紹介するとともに、作品におけるサンプルプログラムの流用、学生のプログラミング理解、プログラミング学習に対する意欲について考察する。

サンプルライブラリは、シンプルなグラフィックスの運動によるインタラクションが人の感覚にどう捉えられるかの発想の基本として、また、ソースコードを流用することでプログラム制作の土台として、有効に機能した。結果として8割の学生にプログラミングの理解の向上をもたらした。一方、1割程度の学生はプログラミングに否定的な徒労感を覚えることが観察された。プログラミング学習のための教材の開発、サンプルプログラムの提示方法の工夫を中心とした改善が必要と考えられた。

そこで、続いて、プログラミング理解を促進する目的で、主に自習に使用する教材を開発した。

プログラム制作をする同じモニタ上で参照でき、プログラムの実行結果を見られるように、Webコンテンツとして開発した。教材はSV教育プログラムに使用するのに加え、それとは独立した一般プログラミング学習の場においても活用できる。この教材を使ったプログラミング勉強会と、教材を適用したSV教育プログラムの、2つの教育実践を行った。本論文では、これら実践で得られた結果を考察する。Web教材を使うことでプログラミングに対する理解認識が高まり、SV教育プログラムにこの教材を適用することは有効であると示された。

ふたつめは、ビジュアルイメージのもつ意味を理解し、使い、制作する能力を高めることを目的とする教育プログラムである。知覚や記号的な意味について考える3つの演習を提案する。演習では素材として写真を使い、学生の描画力や美術的スキルとは無関係に、全ての学生が同じ質の素材から出発できるように工夫した。

この教育プログラムのひとつの特徴は、Webコンテンツとして各演習の成果物をクラス内で公開する点である。学生は観察したもの、発見したもの、制作したものをWebページ上に置き、その後クラス内の他のメンバーの成果を互いに鑑賞、評価する。加えて、成果の公開Webページは自己リフレクションのためのアーカイブにもなる。そのためのブログ形式のWebシステムを、コンテンツマネジメントシステム(Web CMS)を使って開発した。また、演習のプロセスや成果を振り返って評価するため、「目的に沿った探究をしたか」、「独創性があるか」、「表現に注意を払い取り組んだか」の3つの項目に対するループリックを用意した。

Webシステム、ループリックを使い、教育実践を行った結果を考察する。ループリックを使った学生の自己評価と教員の評価の分析から、演習の連続的な実施により、学生の自信が増加したこと、また、開発したWebサイトを通して他のメンバーの成果を見ることで、学生は刺激を受け、新しい気づきを得られたことが、学生の自由記述コメントの質的分析から示された。

目次

第1章	序論	1
1.1	はじめに	1
1.2	背景	3
1.3	本論文の構成	4
第2章	センサリービジョン教育プログラム	7
2.1	はじめに	7
2.2	背景と目的	8
2.2.1	背景	8
2.2.2	目的	9
2.3	教育プログラムの内容と教材	11
2.3.1	テーブル	11
2.3.2	行為を検知する装置	12
2.3.3	学習のためのプログラミング環境	18
2.3.4	行為に応答する画像生成サンプルライブラリ	20
2.3.5	教育プログラムの流れ	22
2.3.6	インタラクションを構想する演習	23
2.4	教育プログラムの実践と評価	27
2.4.1	作品に現れた行為と感覚のインタラクション	27
2.4.2	作品プログラムコードの分析	31
2.4.3	プログラミングの理解に関するアンケート調査	33
2.4.4	自由記述コメントの分析	35
2.4.5	プログラミング理解の認識と作品のプログラム部分の評価	37
2.5	結論	39
第3章	インタラクティブなビジュアル表現を作り出すプログラミング学習教材	41
3.1	一般情報処理教育におけるプログラミング	41
3.1.1	必要性についての提言	41
3.1.2	その実際と問題点	42
3.2	本教材の目的	43
3.3	Web プログラミング学習教材の内容	45
3.3.1	教材の構成と内容	45
3.3.2	ラーニングトレイル部の詳細	47

3.4 教育実践.....	50
3.4.1 Web 教材を使った Processing 勉強会	50
3.4.2 Web 教材を使った SV 教育プログラムの実践.....	51
3.5 考察	54
3.5.1 実践結果の解釈	54
3.5.2 本 Web 教材と既存教材との比較	55
3.6 結論	56
第4章 ビジュアル表現の意味を考える教育プログラム	57
4.1 背景と目的	57
4.1.1 研究の背景.....	57
4.1.2 教育プログラムの目的	58
4.2 教育プログラムの内容	60
4.2.1 基本構成	60
4.2.2 成果物共有のための Web ブログサイト	61
4.2.3 演習後の自己評価.....	63
4.2.4 演習 1：視覚による知覚の理解.....	64
4.2.5 演習 2：ビジュアル変数	67
4.2.6 演習 3：記号論的意味	68
4.3 教育実践とその考察.....	72
4.3.1 教育実践	72
4.3.2 ループリックによる成果の評価.....	72
4.3.3 学びの実感についての振り返り	74
4.4 結論	76
第5章 まとめ.....	79
謝辞.....	83
参考文献	85

図目次

図 2-1 インタラクションのテーブル	12
図 2-2 第 1 バージョンツールキット	13
図 2-3 GAINER 用デモ基板	13
図 2-4 第 1 バージョンの入力部品	14
図 2-5 第 1 バージョンのケーブル	14
図 2-6 第 1 バージョンの出力部品	15
図 2-7 物理的オブジェクトを使ったインスタレーションの例	15
図 2-8 第 2 バージョンツールキット	17
図 2-9 オリジナル I/O モジュール(SV モジュール)	17
図 2-10 第 2 バージョンの入力部品	18
図 2-11 基本運動のサンプル：直線運動	21
図 2-12 基本運動のサンプル：回転運動	22
図 2-13 インタラクションのサンプル	22
図 2-14 行為と感覚の関係の例	24
図 2-15 行為と感覚の演習	25
図 2-16 行為と感覚の記述演習の記録	25
図 2-17 作品例：皮をむくように白線が現れる	26
図 2-18 学生作品例：散	30
図 2-19 学生作品例：KINGYO	30
図 2-20 学生作品例：弾	30
図 2-21 再生産フェーズの学生プログラムにおける流用	32
図 2-22 創作フェーズの学生プログラムにおける流用	32
図 2-23 プログラミング認識理解の変化と学びの実感との関係	34
図 2-24 自由記述コメントのカテゴリと図 2-23 上のデータとの関係	36
図 2-25 プログラムの評価と図 2-23 上のデータとの関係	37
図 3-1 PROCESSING 学習 WEB 教材	44
図 3-2 PROCESSING 学習用 WEB コンテンツの構成	45
図 3-3 クイズの例(条件分岐)	50
図 4-1 デザインプロセス	58
図 4-2 カスタム投稿ページ(演習 3)	62
図 4-3 演習 3 の入力ページ	62

図 4・4 演習 1 の投稿ページの例	66
図 4・5 演習 2 の投稿ページの例	68
図 4・6 意味作用の連鎖.....	69
図 4・7 演習 3 の第一プロセスの投稿ページの例.....	70
図 4・8 演習 3 の第二プロセスの投稿ページの例.....	71
図 4・9 目的に沿った探求に対する学生と教員の評価の比較.....	73
図 4・10 興味と学びの実感に対するアンケートの回答.....	75

表目次

表 2-1 ハードウェアツールキット	13
表 2-2 教育用プログラミング環境の比較	19
表 2-3 センサリービジョン教育プログラムの実践	27
表 2-4 作品中の行為と感覚のインタラクション	29
表 2-5 プログラム理解認識のプロジェクト前後での変化	34
表 2-6 領域(図 2-23)ごとの人数とアンケートの回答	35
表 2-7 自由記述コメントのオープンコード化によるカテゴリ	36
表 2-8 プロジェクトで学んだことに関する自由記述コメント	38
表 3-1 学習の対象となるプログラミング基礎要素	46
表 3-2 動的グラフィックス生成のためのプログラミング学習項目	49
表 3-3 PROCESSING 勉強会と JAVA 授業の比較	52
表 3-4 自由記入コメントの PROCESSING 勉強会と JAVA 授業の比較	53
表 3-5 SV ワークショップの流れ	53
表 3-6 SV 教育プログラム実施前後でのプログラミング理解に対する認識の変化	53
表 4-1 自己評価のためのループリック	63
表 4-2 演習成果物に対する評価	73
表 4-3 学生自身の評価尺度の頻度	74
表 4-4 自由記述コメントとそのカテゴリ	75
表 4-5 カテゴリごとのコメントの数	76

第1章 序論

1.1 はじめに

21世紀の情報化社会では、「情報処理」の技術や知識は専門家にだけ属するものではなく、誰もがそれらにアクセスできる状況にあり、メディア・テクノロジーを使って、専門家ではない人たちが、情報を表現したり、データを処理したり、グラフィックスを加工、制作できる環境が生まれている。ここでメディア・テクノロジーは、人のコミュニケーションをサポートするデジタル技術のことを指し、コンピュータ、センサやI/Oモジュール、映像装置といったハードウェア、インターネット技術、グラフィックスや映像の制作用ソフトウェア、プログラミング環境などを含む技術である。それら技術を、与えられた既成のフレームの中で使っている状況では、私たちはメディアコンテンツの消費者である。学生たちがメディアコンテンツの消費者の立場だけでなく、制作者としての視点をもって行動できることは、学びやその後の仕事にとって大きなメリットが生まれる。そのための基礎教育プログラムを構築することは、一般情報教育において大きな研究課題になっていく。

1919年にドイツで創設されたバウハウスでは、建築を中心に工芸やデザインなど幅広い造形教育が行われ、20世紀のデザイン教育の基礎を築いた。その予備課程では機械工業化時代における新しい素材や技術を使って、材質感、空間構成などの表現技法の習得を目的に、基礎的な造形訓練が実践された。情報通信技術の浸透した社会において、センサやプログラミングが新しい素材として発展した今、バウハウスが工業化社会におけるデザイン教育を新たに築いたように、メディア・テクノロジーという新しい技術を題材として使った一般情報教育の基礎教育プログラムを構築するのが、本研究の大きな視点での動機である。そこで求められる教育内容には、メディア・テクノロジーを理解する力、使いこなすスキル、コミュニケーションのため情報を表現する力の育成が挙げられる。特に情報を表現することに係る制作力を身に付けることで、情報の消費者ではなく制作者として情報技術にアクセスできるようになる。この論文では、情報の表現を対象に、インタラクティブ性、プログラムによる表現と、グラフィックイメージによる意味伝達に焦点をあて、2つの教育プログラムの構築と実践を行った。

ひとつは、コンピュータの特徴的要素であるインタラクティブ性に注目した教育プログラム(SV [Sensory Vision] 教育プログラム)で、メディア・テクノロジーを表現の手段とする造形基礎学習

と、プログラミング学習の双方の目的をもつ。具体的には、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インスタレーション(グラフィックスや音声を使った体験型の立体展示作品)の制作を行う。センサ技術や I/O モジュールを使った教育プログラムとしては、Lego Mindstorms[1] や PicoCriket[2]があるが、これらは主たる対象が子供であり、本格的な作品制作に応用するには発展性に欠ける。前者の高等教育への応用もあるが、それらはロボティックスに関係した課題を解くという工学的な視点で教育が実施されることが多い。一方、後者を使って動くおもちゃや絵本を制作するなど、個人の关心や動機づけを元に、オープンエンドのテーマでワークショップが行われているが、子供が対象ということもあり、人とそれら「もの」とのインタラクションによる関係性の意味を考えるという観点はない。本研究では、人とグラフィックスとのインタラクションが人に新しい感覚経験を与えるような表現をプログラムによって作り出すことを目指して、教材を開発し、教育内容を構築した。I/O モジュールやセンサ類を含むツールキット、展示環境などのハードウェア、サンプルプログラムライブラリを開発し、教育実践に適用した。数理的表現の工夫がインタラクティブなモーショングラフィックスとして目に見える形で実現し、それが新しい感覚経験を与えるインタラクションを生み出すことを制作過程を通して体験した。結果として 8 割の学生にプログラミング理解の向上をもたらした。9 割の学生はプログラミングはむずかしいと考えながらも、プログラミング学習に期待される利点を反映する認識を示しており、モーショングラフィックスを生成するプログラム制作が学習意欲を向上させたと考える。しかし、1 割程度の学生はプログラミングは難しいだけでおもしろさがないと否定的認識をもって終えた。

プログラミング基礎要素を理解していれば、自作プログラムの修正、発展も容易であり、別の機会での問題解決にプログラムを有効に適応でき、さらには達成感、主体感も上がる。そこで、プログラミング基礎の学習面を強化するため、インタラクティブなモーショングラフィックスの描画と、プログラミング基礎要素を運動しながら学べるような、主に自習に使用する Web 教材を開発した。それを使ったプログラミング学習を実践し、プログラミング基礎項目の理解と自己効力感の獲得に、Java 言語入門科目の通常授業での学習と同等以上に効果を上げることを確認した。加えて、Web 教材を適用して SV 教育プログラムを実施することで、プログラミングに対する理解が向上することを示した。

ふたつめは、デザインを専門としない学生が、ビジュアル表現のもつ意味を理解し、使い、制作する能力を高め、デザイン思考力を養成することを目指す教育プログラムである。コミュニケーションのために情報を表現する力を育成する学びの中で、ビジュアル表現の意味を考えることに焦点をあてた。リーフレットや Web ページの制作といった具体的直接的なデザイン演習ではなく、知覚認識やイメージの記号的な意味について考えるなど抽象度の高い視点のテーマの演習からな

る。視覚形態を使って効果的に情報を伝えるためのビジュアルデザインのハウツー本やノンデザイナーを対象とした本は、フォントや色、レイアウトを考えるための実用的ヒントを提供する。しかし、それを個々の場面で対象としている情報表現にどう適用したらいいのか、できた形にはどんな意味が生まれるかについての考察は不足している。本教育プログラムでは、形を作る前に意味を考える観点を養うという意図で、3つの演習を提案する。意識的な観察を通して、ビジュアルイメージに対する知覚認識について理解し、それらを使うことを、また、ビジュアルイメージのもつ明示的意味と共示的意味を考えることを目指す。これら演習を通して、今まで気づかなかつたビジュアル表現のもつ意味に気づき、新しい意味を作り出すトレーニングを行う。教育実践において成果の共有と評価に使用するため、ブログ投稿の形で成果を共有する Web サイトをコンテンツマネジメントシステム(CMS)を使って開発した。加えて、演習のプロセスや成果を振り返って評価するため、「目的に沿った探究をしたか」、「独創性があるか」、「表現に注意を払い取り組んだか」の3つの項目に対する評価基準(ループリック)を開発した。

ループリックを含んだアンケートへの学生の回答と同じループリックを使った教員の成果物評価の分析から、演習の連続的な実施により、学生の自信が増加したことがわかった。また、開発した Web サイトによって共有した他のメンバの成果を見ることで、学生は刺激を受け、異なる視点や新しい気づきを生んだことが、学生の自由記述コメントの分析から示された。

1.2 背景

情報技術が限られた専門家だけのものではないという指摘は、1990 年代からなされている。国家戦略の観点からさまざまな提言を行うアメリカ National Research Council は、情報技術教育に関する報告[3]で、情報技術に関して顕著に重要な点は、「専門家としてのデザイナーやエンジニアの世界から情報技術が取り出され、我々の生活や仕事場に置かれる」ことだと述べ、全ての人に対して「IT 技術に対する流暢さ(FITness)」を育成する必要性を説いた。

この報告が総括しているように、優れたユーザビリティをもつ制作ツールの開発やネットワーク技術の発展により、メディアコンテンツの制作は広く一般のユーザにも容易なものとなっている。そして、情報をいかに構造化し表現するかという情報デザイン能力が、コンテンツ制作に取り組むあらゆる人に対して、求められる状況が生まれている。デザインを専門とする人だけではなく、さまざまな専門性をもつ人がそれぞれの研究や仕事のために、情報をデザインする必要性に迫られ、情報デザインの観点からの素養が求められる。高等教育の一般情報教育においても、従来からのアプリケーションの利用技術獲得やコンピュータ科学の基礎知識の学習に留まらず、それらを利用し

た結果作成されるメディアコンテンツに含まれる情報の構造やその表現の質への関心が不可欠となっている。

中橋と水越は「個人が情報を表現し、情報発信をし、多様なコミュニケーションを生み出していくことができる」今日において、メディア・リテラシー教育の新しい構成要素を整理し、従来中心に置かれていたマスメディアの情報の読み解きに加えて、メディアメッセージの表現技法やクリエイティブ・センスを構成要素として挙げた[4]。また、辰巳らは、FITnessの考え方をベースに大学の一般情報教育の内容として「情報デザイン」を含めることを提案している[5]。

高等学校の情報科目にも、この観点からの学びはすでに含まれている。平成25年から適用された高等学校学習指導要領の共通教科情報科の科目「社会と情報」では、情報を分かりやすく表現し効率的に伝達するために「情報の表現と伝達」を学ぶ。また、専門教科情報科で新設された科目「情報デザイン」では、「表現と心理」「意味の演出」「要素の構成」が学びの要素とされている。

気を付けなければならないのは、それらの教育内容では、単にフォントや色、レイアウトなど表面的ななかたちの表現だけを学ぶのではないということである。ビジュアル表現がもつ意味を考え、それらを効果的に使うというビジュアルリテラシー[6]の学びが求められる。

加えて、National Research CouncilはFITnessの全体像を説明する中で、プログラミング知識の重要性について節を変えて特に強調している。さらに、認知心理学者のNormanは、デザイン教育の文脈の中で、描画やスケッチなどの古いスキルに代わり、人間の認知やプログラミング、インタラクションについて学ぶ必要性を提言している[7]。このように情報教育、デザイン教育の双方において、プログラミングの理解は重要な要素として位置付けられている。

本論文では、制作者としての感性と基礎的知識の育成のため、インタラクティブ性、プログラムによる表現、ビジュアルイメージによる表現を題材に、新しい一般情報教育プログラムを開発した。

1.3 本論文の構成

本論文は序論に続く第2章で、インタラクティブ性の探求、プログラミング理解を目的としたSensory Vision教育プログラムについて述べる。背景と目的を説明した後、展示のためのインタラクションの場としてのテーブル(台)、I/Oモジュールやセンサ類を含むツールキット、プログラミング理解のためのサンプルライブラリについて詳説する。次に学習の具体的な流れに加え、インタラクションを構想するために考案した演習について述べる。2年間の教育実践を紹介し、実践から得られた知見を考察する。それは(1)学生作品に現れるインタラクション、(2)作品のプログラムコ

ードから見た、プログラムの創作性の程度とサンプルプログラムの流用, (3)学生のアンケートへの回答を基にしたプログラミング理解の主観的認識の変化, 学びの実感, (4)自由記入コメントの質的分析, (5)メディアコンテンツ作品としての教員の評価とプログラムの創作性の程度の関係についての考察からなる。

第3章では、第2章で述べた教育プログラムにおいてプログラミング基礎の学習面を強化するために開発したWeb自習教材について述べる。はじめに一般情報教育におけるプログラミング学習についてその必要性に対する提言を概観し、今までに実践されているプログラミング教育の例を紹介する。その後、本教材の目的と具体的な内容について述べる。本Web教材を使っての勉強会でのプログラミング学習と通常のJava言語によるプログラミング授業での学習について比較する。また、2章で述べた教育実践と、本Web教材を使っての教育実践を比較して、その結果を考察する。

第4章では、ビジュアル表現の意味を考えるための教育プログラムを提案する。教育プログラム全体の目的に統一して、視覚認知、ビジュアル変数、記号論的意味をテーマとした3つの演習を紹介する。演習成果共有のためのWebシステム、評価のためのループリックについて述べ、それを使った教育実践について経験的結果をもとに考察する。

第5章では、全体を総括し、結論を記す。

第2章 センサリービジョン教育プログラム

2.1 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が、1990 年代初頭からアートの世界に現れ、インタラクティヴァートやメディアアートと呼ばれるジャンルを切り開いた。この 20 年余りのメディアアートの歴史の中で、これまで実に多様な表現実験が行われてきた。例えば、鑑賞者と作品とのインタラクティブな関係性を重視するもの、プロジェクトを用いた映像インスタレーション作品、鑑賞者の存在や行為などの身体性を重視する作品、インターネット上のデータをリアルタイムで収集する作品、テレプレゼンス技術を利用した作品、人工生命や人工知能の技術を用いた作品、仮想現実(VR)システムを応用した作品、目に見えないデータの視覚化を中心的課題とする作品などがある。特に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してきた。

これらメディアアートの多様な表現は、基本的にコンピュータの技術的特性に由来する。特にデジタル・コンピューティングにおける情報の流れ、すなわち入力／アルゴリズムによる処理／出力のデータフローは、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鍵となる技術である。今後もメディアアートは、新たな技術開発とともに、いっそう多様な表現を生み出すであろうが、コンピュータがもたらすインタラクティブ性は、最も基本的な要素であると言える。この要素を軸に、メディア・テクノロジーの時代にふさわしい題材を使った教育の基礎的な教育プログラムを構想し、そのための学習環境を開発するのが、本章で述べる研究の目的である。

ビジュアル表現を生成するためのプログラミング環境である Processing[8] の開発者のひとり Reas[9] はプログラムによるメディアアート作品の価値を「メディアとしてのソフトウェアのもつとも独特的な側面は、それが応答するという点にある」と述べている。しかも、デジタル技術がもたらすインタラクションは、入力と出力の機械的な応答にとどまらない。鑑賞者の感覚や意識の覚醒に深く関わるような表現やコミュニケーションの新しい技法の考案などを目指して始めて価値が生まれる。これはアート作品の視点に限らず、プロダクトデザインやヒューマンインターフェースデザインの場面で起こるインタラクションにもいえる。

本教育プログラムの中で受講生は、音センサ、赤外線センサなどからの入力に応答して動的に

生成されるグラフィックスと人の行為がインタラクションをとることで鑑賞するインタラクティブ・コンテンツを制作する。体験型の立体展示作品(インスタレーション)の制作を行うこの教育プログラムは、メディア・テクノロジーを表現手段とする造形基礎学習だけでなく、プログラミング学習としての目的をもつ。教育実践に必要となる I/O モジュールやセンサ類を含むツールキット、展示環境などのハードウェア、プログラミング理解をサポートする教材を開発し、この教育プログラムをセンサリービジョン(Sensory Vision) プロジェクトと呼んだ。

2.2 背景と目的

2.2.1 背景

人の創造性や知性を、コンピュータを使って実現するには、ユーザとしてコンピュータを操作するだけでなく、作り手としてプログラミングを学ぶことが重要で、それが多くの可能性を広げる。プログラミング学習により、コンピュータがどう働くかといった基礎的な知識に加え、複雑な考えを単純な問題に分解して考える力、系統的に推論する力などを習得できる。そして、その考え方やスキルは理工学系の学生だけでなく、人文科学分野の学生にも必要である。認知心理学の立場からデザインを研究する Norman は、「なぜデザイン教育は変わらなくてはならないか」という論考の中で「古典的 (classical) 工業デザインでは、形や素材についての知識やスケッチやドローイング、成形のスキルが求められたが、新しい領域では人間の認知や情動、知覚の理解が必要だ」と述べ、具体的に「デザイナーはマイクロプロセッサやディプレイ、アクチュエータやセンサを効果的に展開する必要がある。(中略) ドローイング、スケッチなどの古いスキルは、プログラミング、インタラクションや人間認知のスキルによって補完される、あるいは置き換わられるはずだ」[7]と、デザインの文脈においてプログラミング教育の必要性を提言している。

多くの一般向けプログラミング教育の手法が提案、研究されているが、そのひとつが物理的な「もの」を用いて、それをプログラムで制御することで、プログラミング学習の楽しさや効果を上げようという工夫である。また、センサ技術や I/O モジュールを使った教育プログラムも開発されている。例えば Lego Mindstorms[1]は、LOGO のタートルグラフィックスがモニタ上を動くというバーチャルな世界にとどまらず、LEGO ブロックとセンサ、アクチュエータなどを組み合わせて、プログラムによって動くロボットを作成する教材として開発された。これを使ってプログラミング入門[10]やプロジェクトベースの学び[11]を行う研究が報告されていが、多くは工学系教育の文脈で実施され、ロボティクスに関する学習に焦点がある。

電子工学の知識やハンダ工作が不要な I/O モジュールやセンサとして、他に PicoCricket Kit[2] や pri/pro[12]がある。Mindstorms がロボットを作るキットとして特化されているのに対し、これ

らは光や音声、アクチュエータなどを使ったメディアコンテンツ制作のためにデザインされたキットで、子供向けのワークショップなどで使用されている。PicoCricket Kit はキットとしての完成度は高く、ロボティックスのような工学的視点ではないソフトな素材(毛糸やビーズなど)を使い、幅広い関心の学習者の興味を引き付ける内容である。しかし、子供を主たる対象にしていることもあり、本格的な作品制作に応用するには発展性に欠けるところがある。また、そこで作られる動くおもちゃやアクセサリーなどは、インターラクションの観点から見ると、センサからの入力に応じた出力という単なるメカニズムにとどまっている。

センサを使ったフィジカル・インターラクションを取り入れた教育において、機械的な応答を実装することで満足してはならないことは、次の理由から重要である。マウス操作やボタン押下といった、いわば 20 世紀型のソフトウェアや電気機器の操作から、ジェスチャによるインターフェースへと技術が進化していく中で、ソフトウェアのヒューマンインターフェースやインターラクティブなメディアコンテンツを開発・制作するにあたっては、固定的認識から離れ、新しい意味や関係性をもつアイディアを生み出すことが求められる。そこでは人の行為と知覚が生み出す新しいインターラクションを探求しようというような多様な観点が必要となる。

Kim ら[13]は、芸術とコンピュータ科学の双方の基礎的要素を学ぶためのコースとしてインターラクティブな「触れる」展示を行うプロジェクトを提案した。芸術と IT 技術の視点をベースにインターラクティブなアート作品を制作するという点で、本研究の教育内容と共通点があるが、プログラムによる表現には焦点をあてていない。また、芸術とコンピュータ科学の領域の学生が分野を横断して互いに学び会うことを目指した教育も多く試みられている。例えば、Ursyn ら[14][15]や Ribner ら[16]の教育プログラムは、コンピュータ科学専攻と芸術専攻の学生が同じクラスでマルチメディアコンテンツを制作する内容で、双方の理解とスキルの向上を目的にしている。リベラルアーツ教育の一環としてメディア・テクノロジーを題材に制作プロジェクトを行う点で、本教育プログラムと共通する内容である。

2.2.2 目的

本教育プログラムで行う課題のゴールは、鑑賞者の行為とインターラクティブに応答するインスタレーションの制作である。そこで要求されるのは、新しい感覚経験、驚きや楽しさを創出すること、それをプログラミングを使って実現することである。特定の機能を実現するような問題解決型の課題ではなく、自ら問題を発見、設定して、構想したアイディアをプログラムで表現することを求める。教育プログラムの目的は大きく次の 2 点である。

(1) 人の行為とグラフィックとのインターラクティブな関係への気づき

携帯端末を日常的にもつ生活を送る中で、人はコンピュータ上で起こる入力→データ処理→出

力という情報の流れを当然のものとして利用している。そこで起こるインタラクションは、ある入力に応じてある出力(反応)が現われるという固定的なメカニズムとして認識される。しかし、ソフトウェアのヒューマンインターフェースやインタラクティブなメディアコンテンツを開発・制作するにあたっては、固定的認識から離れ、新しい意味や関係性をもつインタラクションを生み出す必要がある。物理的、身体的なインタラクションを使ったシステム(インスタレーション)を開発する課題を通して、その探求へ向けての基礎的なトレーニングを行う。人の行為とそれがもたらす応答との間に生じる感覚を元に、新しいインタラクティブな意味関係を探る教育プログラムを構築する。これは、学生の専門分野と関係なく、デジタル技術によるメディアを使ったコンテンツや装置に、使い手としてあるいは作り手として、否応なく関わっていくことになる学生にとって、重要な感覚教育となると考える。

(2) コンピュータプログラミングの理解

センサによって検知する人の行為に応じ、動的にグラフィックスを生成するプログラムの作成を通して、プログラミングを理解する。プログラミングの学習によって得られる利点は、次のような点である。

(a) 数理的な手法による制作力をつける。

数理的手法に基づく表現方法を見つけ出し、実現する経験をする。

(b) 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、プログラムの工夫することで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。

(c) 制作・表現の幅を広げる。

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。

一般情報教育の文脈の中でプログラムを学ぶ利点は、今まで多く論じられている。その先駆者で LOGO 言語の開発者 Papert は、「タートルに行動や『思考』するのを教えることが、自分自身の行動や思考を熟視することにつながる」とタートルグラフィックスを使ったプログラミング学習の利点を述べている[17]。本教育プログラムでは、プログラムの工夫がグラフィック要素の動きとなって見えることで、驚きや楽しみを体験しながら、新たな表現を求めて数理的思考を深めることに特徴がある。そしてプログラムによる表現を試しながら作り出し(上記(a))、作品を完成させることで主体感の獲得(b)を経て、プログラムを今後の制作に適用して(c)いけるようになることが、学習の目指すところである。プログラムの実行結果を手に取るように(タングibleに)、確認できる点で、本教育プログラムはプログラミング理解に効果的に働くと期待する。

本研究での教育実践は、メディアコンテンツ制作を学ぶ情報メディア学科の学生に対して行つたが、教育プログラムは芸術やデザインを専門に学ぶ学生を対象とするものではなく、リベラルアーツの枠組みの中での一般教育科目として想定している。バウハウスの教授でもあったモホリ＝ナギは、「初年度、学生は感覚的経験、情感的価値の豊かさ、思考の開発に向けて導かれる。」と書いている[18]。彼は感覚的経験は素材特性の基礎的知識や技術の理解、そして道具を使って作ることを通して得られると考えていた。感覚的側面の教育、そのために技術の理解が必要なことに関して、本研究の教育プログラムはモホリ＝ナギの意図と重なる。本プログラムは、現代アートあるいはプロダクトデザインのような限られた分野に限らず、既成メディアのフレームを離れ、メディアコンテンツの創造的な利用を考えるための基礎教育で、プログラムによる表現力や構想力の育成を目指す。

この教育の実現のために、次のような学習環境が必要となる。(1)展示形態の設定、(2)センサ技術を容易に使える制作環境、および、(3)プログラミング理解を助ける学習教材。これらを開発するのが研究の目的である。(1)は2.3.1項で述べるテーブルの形で用意し、(2)は2.3.2項で述べるハードウェアツールキットとして開発し、電子工作スキルが十分でなくても、簡単にセンサが感知したデータをコンピュータへ読み込むことができるよう制作環境を工夫した。(3)は2.3.3項に述べるサンプルプログラムライブラリとして開発した。

2.3 教育プログラムの内容と教材

2.3.1 テーブル

インスタレーションの展示形態にはさまざまなバリエーションが考えられる。この部分を学生の自由な構想に任せると、形態の決定で思考が発散したり、大掛かりな装置が必要となったり、本来の教育目的が損なわれる可能性がある。そこで、シンプルな構成の中でインタラクションの意味の追求がなされることを目的に、制約的な空間環境を設定することとし、人の行為をテーブル面に対してのふるまいに限った。手を添える、撫でるなどの行為を誘引するために、白い木製のボックス(W500×D500×H900 mm、図2-1)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。その天板上で、鑑賞者の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、多様な感覚経験を生み出す場となる。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、天板は箱本体から取り外しが可能で、穴開けや接着などの加工が容易にできるようにした。天板にセンサを埋めこむことで、テーブル面がインタラクティブなインターフェースとして機能する。

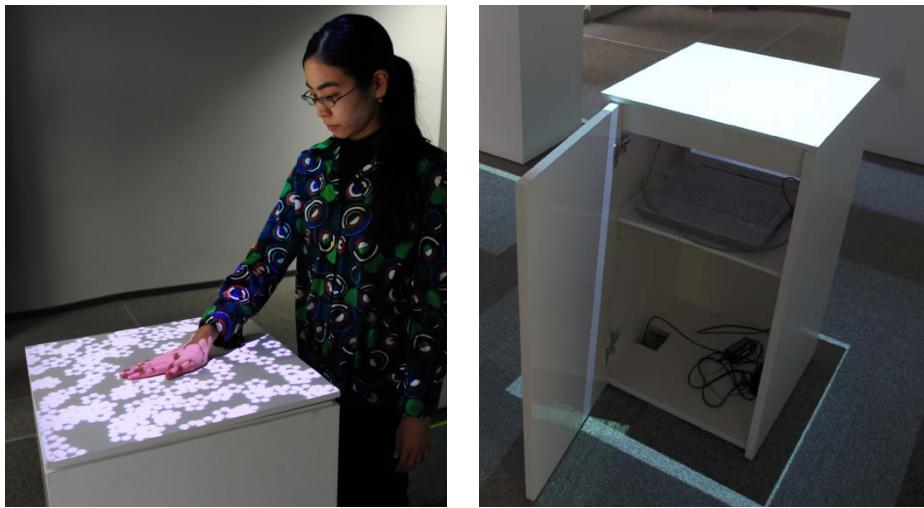


図 2-1 インタラクションのテーブル

2.3.2 行為を検知する装置

鑑賞者の身体的行為をデータ化し、コンピュータに取り込むにはセンサなどの入力装置と、入力装置からのアナログ信号をデジタル信号に変換する I/O モジュール(AD コンバータ)が必要である。一般にそれらを使うには、最低限の電気の知識や、ハンダづけなどの電子工作のスキルが必要である。しかし、本教育プログラムが対象とする学生は、そのような知識やスキルを持たず、センサなどの電子部品を自身で準備し、それらを組み立てることは困難である。電子工作をせずとも、インタラクティブな「装置」を制作できるように、I/O モジュールと接続すればそのまま使えるセンサをパッケージにしたハードウェアツールキットを教材として用意した。第 1 バージョンの開発、実験的使用を経て、オリジナル I/O モジュールを含む第 2 バージョンのツールキットを開発した(表 2-1)。このツールキットを使うことで、学生は、容易に人の行為を検知し、それへ反応するプログラムを実験することができる。

第 1 バージョンのツールキットでは(図 2-2)、I/O モジュールとして Gainer を採用した[19]。プログラミング環境として Processing(Java)をサポートしている点、また、I/O モジュールとして広く使われている Arduino と互換性があるという点で、柔軟性を考慮して選択した。Gainer はプリント基板にピン端子がついた形態のモジュールであり、このままではセンサやアクチュエータなどの装置に接続できない。ブレッドボードとジャンプワイヤ、ワニ口クリップを使って実験することはできるが、接続の安定性の点から実際の制作には向かない。そこで電子工作をすることなく接続が行え、かつ必要に応じてアクチュエータへ外部から電源を供給することができる機能をもつデモ基板を開発した(図 2-3)。

表 2-1 ハードウェアツールキット

	第1バージョン	第2バージョン
I/O モジュール	Gainer (IAMAS 開発) + オリジナル基板： アナログ入出力，デジタル入出力	SV I/O モジュール(オリジナル)：アナログ入力，デジタル出力
入力部品	タッチセンサ，光センサ，音センサ，赤外線センサ，可変抵抗，マイクロスイッチ，リードスイッチ，水銀スイッチ	タッチセンサ，光センサ，音センサ，赤外線センサ，可変抵抗
出力部品	LED，振動モータ，半導体リレー，小型モータ，ギアセット	LED
ドライバ・ケーブルなど	USB ケーブル，回路テスター，ドライバーセット，メスピンヘッドプラグ付き三線ケーブル	USB ケーブル，回路テスター，ドライバーセット，ミニピン延長ケーブル



図 2-2 第1バージョンツールキット

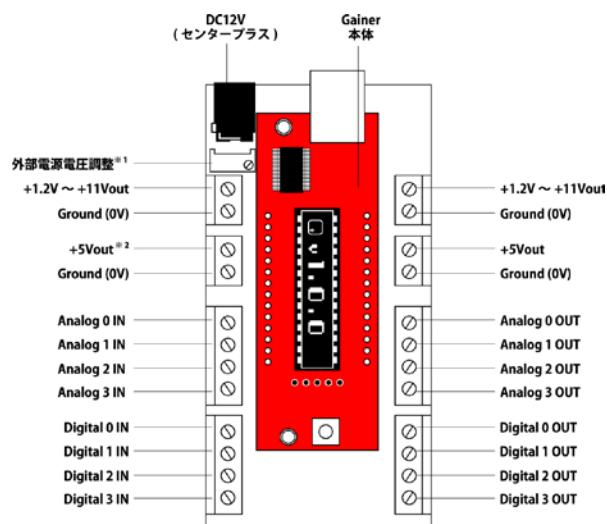
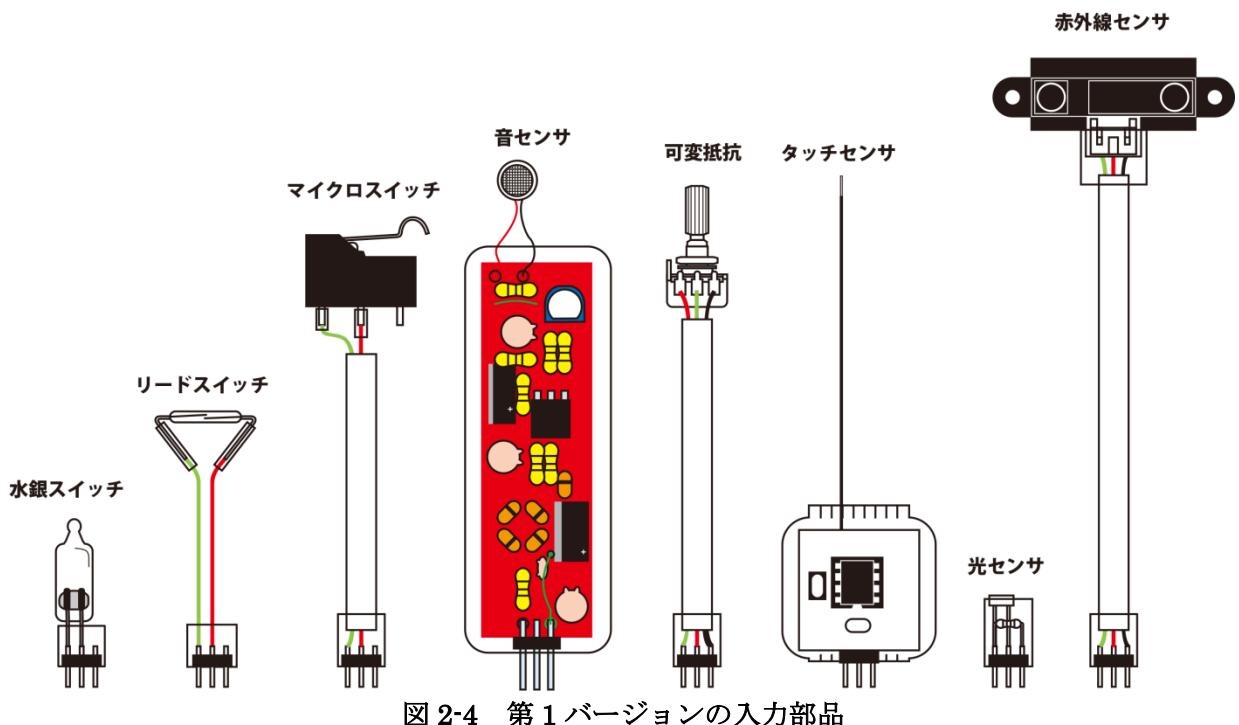


図 2-3 Gainer 用デモ基板

Gainer の入出力ポートとセンサやアクチュエータの接続には、ハンダ工作が不要なように差込式端子を使った。これにより入出力部品を安定して I/O モジュールに接続できる。また外部電源用に DC ジャックを備え、ここから受け取る 12V 直流電圧は半固定抵抗器で調整できる。

Gainer とデモ基板だけでは、まだ実験をすることはできない。電子工作の経験がないと、センサに適切な抵抗を接続したり、ケーブルを用意することさえ、むずかしいであろう。そこで、Gainer とデモ基板に加え、すぐに使えるセンサやモータ、LED、ケーブルなど必要最小限の部材を含む制作キットを含めた。これにより、学習者は Gainer を I/O モジュールとして、キット内のセンサや LED などを使い、サンプルプログラムで制御する実験を簡単に行える。



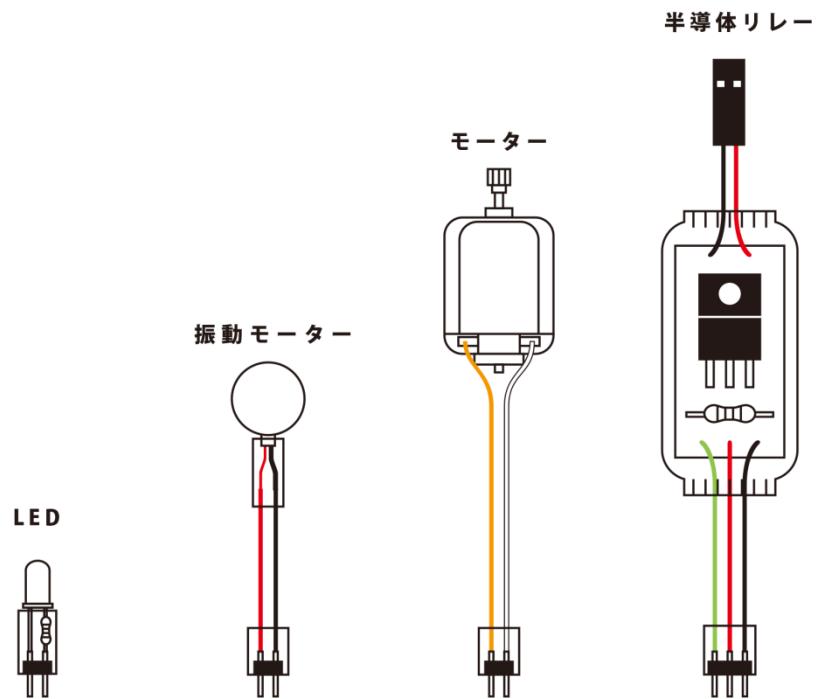


図 2-6 第 1 バージョンの出力部品



図 2-7 物理的オブジェクトを使ったインсталレーションの例

ツールキット内の入力用センサ類(図 2-4)には、必要に応じてあらかじめ抵抗などの部品を取り付け、接続端子としてオス 3 ピンヘッダを付けた。ピンソケットをもつ図 2-5 のケーブルを使ってデモボードの電源、グランド、入出力の差込式端子に接続できるように工夫している。また、図 2-6 は出力用部品であり、人間の行為と物理的オブジェクトがインタラクションをとるような構成の作品の実装も想定し、小型モータ、ならびに外部機器を制御できるようにリレーを同梱した。

この第 1 バージョンを使った教育プログラムの実験的実践では、グラフィックスではなく、「もの」とのインタラクションをとるインスタレーションが制作された。図 2-7 はその例で、左はテーブル面が洗面台となっていて、可変抵抗が埋め込まれた蛇口を回すと、その程度により異なる水流音が流れる。右はテーブルが日本風箱庭のように装飾され、中央に砂が敷かれている。砂の下には振動モータが、砂の手前には赤外線センサが仕込まれている。テーブル面に手を伸ばすと、振動が起り、砂状に波紋が現れる。

これら物理的オブジェクトを使った作品のインタラクションは、「前に立つとドアが開く」のように単純な機械的な応答になりがちで、その応答は固定的な文脈に依存した説明的な内容であった。これは、学生が具体的シンボルがもつ典型的な記号的意味に捉えられて、結果として、新しい感覚的意味表現を生み出すことから遠ざけられてしまったことを示す。

そこで、具象的オブジェクトがもつ固定的意味やそれから派生する紋切り型の文脈を避けるため、物理的な「もの」の存在を排除し、応答としてグラフィックスのみを採用することにした。それにより、アクチュエータの出力部品が不要となり、第 2 バージョンのツールキット(図 2-8)では、出力部品をテスト用の LED のみとした。また、入力部品からスイッチ類をはずした。物理的に「もの」を動かすのではなく、テーブル面に対しての「こする」や「たたく」などの動作を検知するには不要なためである。オンオフスイッチのような動作をさせる場合は、タッチセンサや光センサを使い、条件分岐のプログラムで対応できる。また、第 2 バージョンでは、オリジナルの I/O モジュール(Sensory Vision[SV]モジュール)を開発した(図 2-9)。4 つのアナログ入力ポート、4 つのデジタル(PWM: Pulse Width Modulation)出力ポートを備えた AD コンバータで、擬似 RS-232 シリアル信号で通信する。センサからのアナログ値(0-5V)を 0-255 の 8bit 信号に変換する。本教育プログラムに必要となる最小限の機能に留め、Gainer より機能をシンプルにした。直接アクチュエータを制御しないのでアナログ出力をなくし、またスイッチ類を除いたのでデジタル入力をなくした。

第 1 バージョンでは Gainer 用デモボードに差込式端子を使い、入出力部品と I/O モジュールのポートとを接続したが、どのケーブルをどの端子につなぐのかを迷ったり、ドライバーで締める作業が適切でなく、接続が不安定になることがあった。確実に接続することができるよう、SV モジュールはミニピンジャックのコネクタを採用した。センサ側にはミニピンプラグをあらかじめ付

けており、差し込むだけで簡単に実験を行うことができるよう工夫した(図 2-10).



図 2-8 第 2 バージョンツールキット

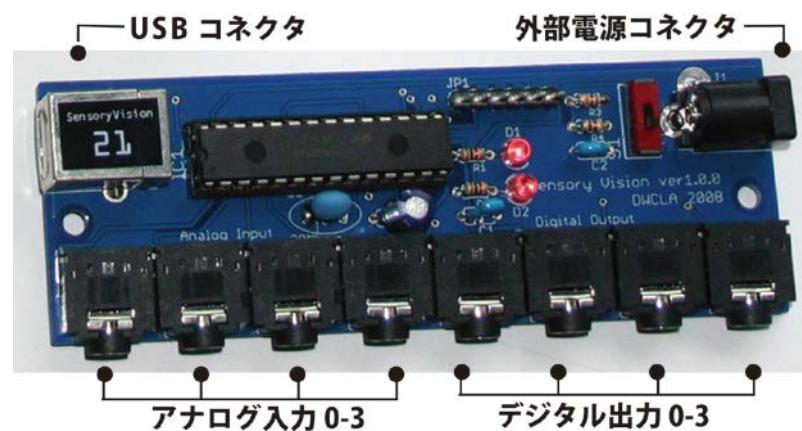


図 2-9 オリジナル I/O モジュール(SV モジュール)

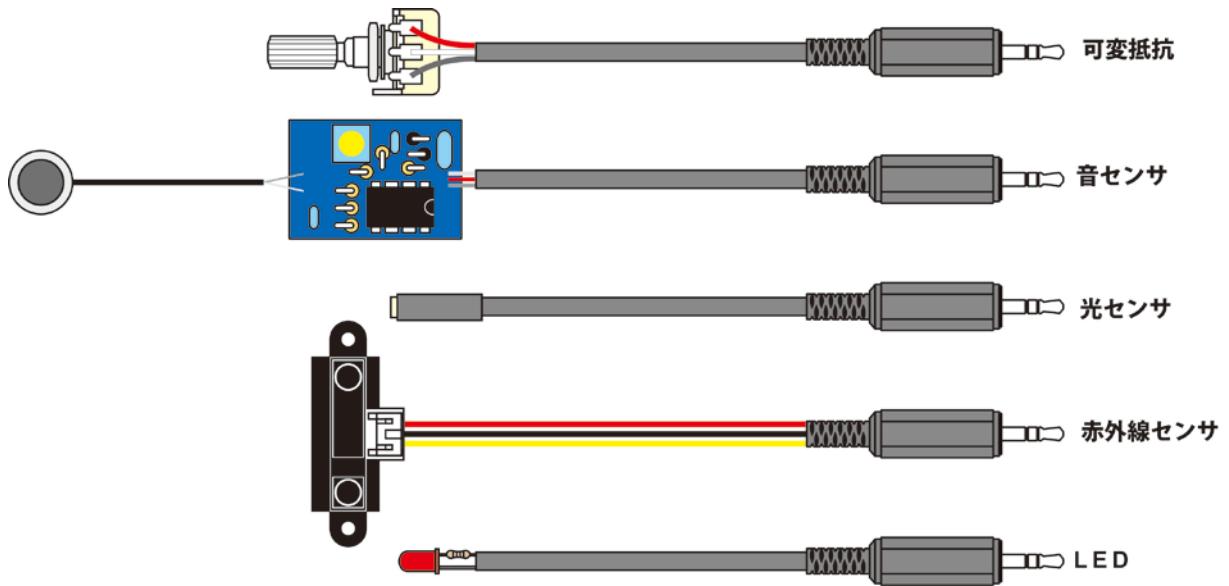


図 2-10 第 2 バージョンの入力部品

2.3.3 学習のためのプログラミング環境

鑑賞者の行為に基づき生成し変化する動的なグラフィックイメージを生成するため、プログラミング環境として Processing を採用した[8]. Java 言語を使って開発されているシステムで、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機能をもつ。

プログラミングの基礎を学ぶのに使われる言語の特性については多数の調査、研究が行われている。例えば、兼宗[20]は、教育に使われるプログラミング言語を 4 つのグループ(実用言語、伝統的教育言語、新しい教育言語、特定の用途向け言語)に分け、それぞれの特徴をまとめ、教育用プログラミング言語にふさわしい性質を述べている。Giangrande[21]は入門科目に使われる言語環境のうち、C++, Java, マイクロソフトの.NET フレームワークを比較している。Gross と Powers[22]はビジュアルプログラミング環境を対象に教育での適用を評価した。教育用プログラミング環境に求められる特徴を網羅的に比較したのが Mannila と Radtt[23]で、教育用言語とされている LOGO[17], Pascal[24], Python[25], Eiffel[26]の開発者がそれぞれ提案しているプログラミング言語に求められる基準を抽出し、実際に使われている言語環境がそれらを満たしているかどうかを検討した。17 項目の基準を示し、11 の言語で比較している。表 2-2 にその内 C, C++, Eiffel, Java, LOGO, Pascal, Python の比較結果を転載し、Processing に対する評価を追加した。Processing は統合型開発環境とは違い、コードが正しく、バグがないかを確かめるのをサポートする機能には欠けており、また教材、特に日本語の教材は限られるという欠点がある。しかし 13 項目を満たしていて、プログラミング基礎の学習言語として優秀と判断できる。

表 2-2 教育用プログラミング環境の比較

求められる特徴	C	C++	Eiffel	Java	Logo	Pascal	Python	Processing
シンプルな構文、混乱を招きにくい				✓	✓	✓	✓	✓
物理的比喩を使える				✓	✓	✓		✓
他の汎用的言語の原則、基本を学べる	✓	✓	✓	✓		✓	✓	✓
ソフトウェア開発全体をカバーするツールがある			✓	*1	✓			
初心者がアイディアを直ちに試し、結果を得られる					✓		✓	✓
コードの正しさ、バグがないかを確かめる方法がある		*2	✓	*2				*2
モジュール化できる	✓	✓	✓	✓	✓	✓	✓	✓
統合型開発環境を提供する			✓	*1				
サポートしてくれるユーザコミュニティがある	✓	✓	✓	✓			✓	✓
オープンソースである							✓	✓
多様なプラットフォームで使える	✓	✓	✓	✓	✓	✓	✓	✓
無償で簡単に利用可能			✓	✓	✓	✓	✓	✓
よい教材がある			✓	✓	✓	✓	✓	✓
教育以外でも使われる	✓	✓	✓	✓			✓	✓
拡張が可能	✓	✓	✓	✓			✓	✓
信頼できる	✓	✓	✓	✓		✓	✓	✓
過去だけでなく、未来においても有用		✓	✓	✓	✓		✓	✓

Processing 以外の評価は、Mannila と Raddt[23]から転載

*1 別の統合型開発環境を使えば可能

*2 単体テストでは可能

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開発環境としては、他に主に年少者を対象に Scratch や Etoys などのビジュアルプログラミング環境がある[27]が、入門を超えて多様なレベルで使える、汎用言語の基礎を学べるという点で不足がある。Processing は Java コードをそのまま使えるのに加え、多くの外部ライブラリが開発され、プロトタイプ制作にとどまらず、実用的な作品にも利用できる長所がある。また、ビデオや音声を扱う拡張機能も豊富なコミュニティライブラリとして提供されている。しかし、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。ただ、描画のために用意された関数群のおかげで、簡単な描画プログラムを書くにも、多くの文法要素を理解しなければならないという欠点はない。そのような点を

踏まえた上で、本教育プログラムではプロジェクト後の実際の作品制作への発展を考え、拡張性、応用性の高い Processing を選択した。

2.3.4 行為に応答する画像生成サンプルライブラリ

本教育プログラムの流れは次項で説明するが、プロジェクト実践の中でプログラミングの学習の時間は限られている。またプログラミングを専門科目としては学ばない学生を対象とすることを考慮し、次の 3 つのフェーズを通して、主に自習により理解を進められるよう、サンプルプログラムのライブラリを用意した。それらを模倣することからプログラム制作を始め、それを修正し工夫を積み重ねながら、作品を完成するよう、指導した。

- (i) 模倣する…既存のプログラムを真似て、使う段階。テストをしながら、プログラム内の記述を調べ、プログラム要素の使われ方、その意味を知る。
- (ii) 再生産する…既存のプログラムを改造して実験する段階。学生はグラフィックスの動きが数理的メカニズムを使ってどう制御されるかを理解できる。その理解が次のフェーズの独自の動きの創作につながる。
- (iii) 創作する…これまでのプログラミングの理解をもとに独自の動きを生み出す数理的手続きを考案出し、それをインタラクティブアニメーションの制作に適用できる。

サンプルプログラムライブラリは、「基本運動」と「センサリーインタラクション」の 2 種である。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、プログラムにより動的グラフィックスを生成する基本を学生に示す。後者はセンサからの入力により基本グラフィックスを動かす方法を示すサンプル集である。「基本運動のライブラリ」では、グラフィックオブジェクトの運動を、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。サンプルプログラムをシンプルにし、プログラム同士の比較をしやすくするため、グラフィックオブジェクトには円を使った(ただし、回転運動のサンプルは正方形を使用)。例えば、一方向の運動パターンには、円が直線、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、大きさや個数を変えて運動するプログラムが含まれる。

図 2-11 は一方向の運動のうち、軌道が直線となるプログラムの実行結果で、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。このような単純な動きを生成するプログラムによって、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。

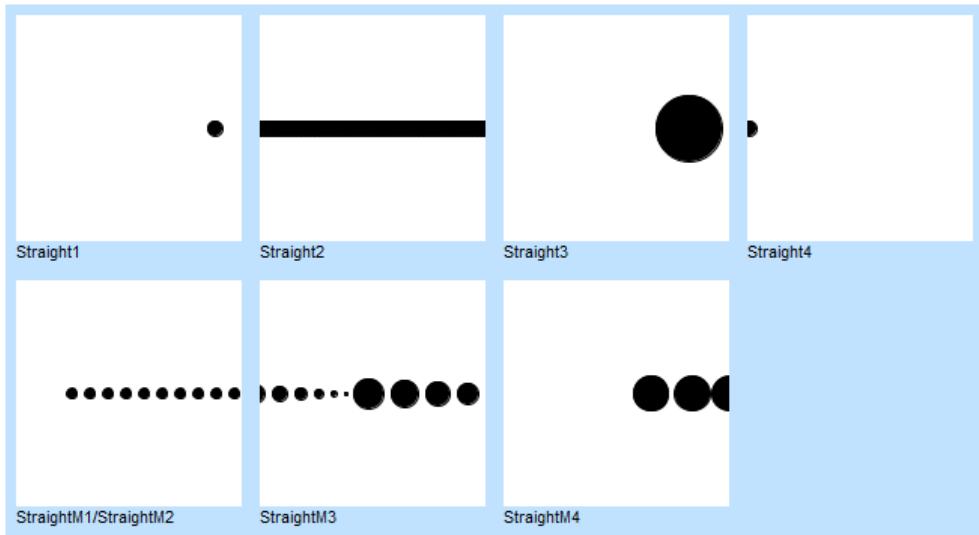


図 2-11 基本運動のサンプル：直線運動

図 2-12 は正方形の回転運動プログラムの実行例で、その場での回転、軌跡を残しながらの回転、大きさを変化しながら回転、軌跡を残しながら大きさを変化して回転、位置を変えながら回転というように、同じ回転運動でも、大きさや位置の変化を加えることで生まれるビジュアル効果を知る。

「センサリーインターラクション」と名づけたライブラリは、センサ入力に応じて動的にグラフィックを制御する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。

(a)センサからの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除き、音の大きさとオブジェクトの大きさを反比例させる方法をこのサンプルを通して知る。

(b)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、あるいは停めるといった制御の方法を学ぶサンプルである。

(c)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、動きを開始後、一定時間経過した後に停める方法を知るサンプルである。

(d)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い、粒子の動きと入力値とを連動させる。

図 2-13 は、センサからの入力に応答するサンプルプログラムの実行例で、左から順に入力値の変化量により描く円の大きさが変化する(a)、入力値が閾値を超えると円が弾けるように放射線上に

広がる(b), 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(d).

ライブラリは, オブジェクトの運動が人の視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験を行い, サンプルが示す運動を観察することで, 人の行為とグラフィックスの動きが生み出す感覚の関係を発見できる. このような実験と観察の後, 学生は制作内容を企画するが, その実現のため, 企画内容に合わせてまずは真似るプログラムを選び, それを修正, 拡張し, 加えて必要に応じて順次新たな動きを追加していくよう, 学生を導く. この際, サンプルプログラムは模倣する→再生産する→創作するというプログラミング学習の3つのフェーズの出発点となる.

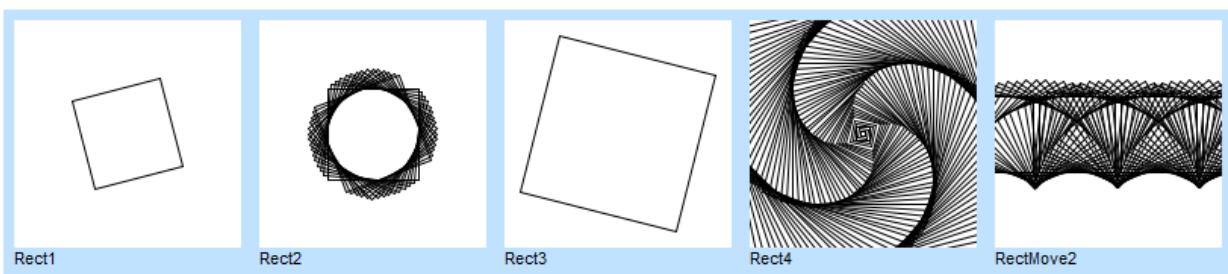


図 2-12 基本運動のサンプル：回転運動

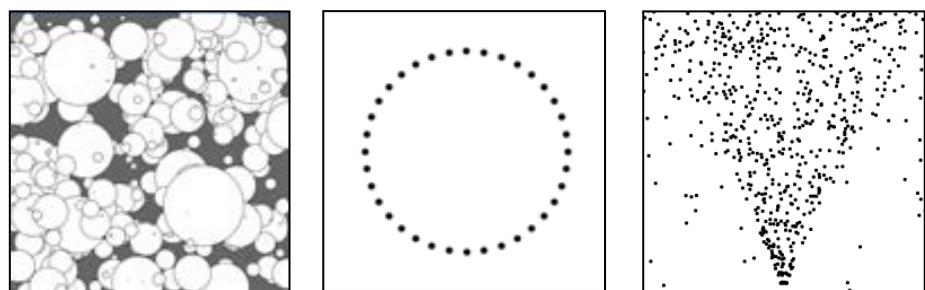


図 2-13 インタラクションのサンプル

2.3.5 教育プログラムの流れ

センサリービジョンプロジェクトでは, ひとつのインタラクティブインсталレーションを, 2名がチームとなって, 次のようなプロセスで制作し, 展示発表する.

- (1) プロジェクトの目的の理解
- (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

半期 14 回の授業の中で上記(1)から(8)を実施し、その後学内ギャラリーで展示する。プログラミングの理解は、4回のクラスセッションで次のようなプログラム要素の説明をし、その後はツールキットとサンプルライブラリを使い、実験しながら自習する。

- ・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)
- ・音声ファイルの再生
- ・クラスの活用
- ・画像ファイルの描画

2.3.6 インタラクションを構想する演習

インタラクションの場として用意したテーブルは、鑑賞者に触る、たたく、なでる、息を吹きかけるなどの行為を誘い、天板上に投影されるグラフィックスがその行為に応じて変化する。それは、単なるセンサからの入力とそれへの応答という技術的なメカニズムを越えた行為と感覚の多様な感覚的な経験を作り出す。赤外線センサからの入力に応じて、投影される円の直径が変化する図 2-14 の例を考えてみよう。鑑賞者が天板上で手を上下に動かす動作に連動したグラフィック要素の変化は、大変単純なメカニズムである。しかしながら、それがもたらす感覚的経験はそう単純ではなく、手を押し返すような感覚、あるいは何かしらの力が手から発せられているかのような感覚が起こる可能性がある。さらに、弾性のある動きは、見えないバリアがそこにあるような反発力を感じさせる。インタラクティブインсталレーションをしくみとして見れば、「人間の行為を検知し、プログラムにより入力値を処理し、出力としてビジュアルイメージを生成する」というだけもので

あるが、目指すのは入力を出力に変換するような単純なメカニズムではなく、鑑賞者が自身の行為にリンクしたビジュアルイメージを見て、それがもたらす行為と感覚の関係を体験することである。つまり、インタラクティブインスタレーションがそのような感覚経験を提供するように、インタラクションを構想する必要がある。

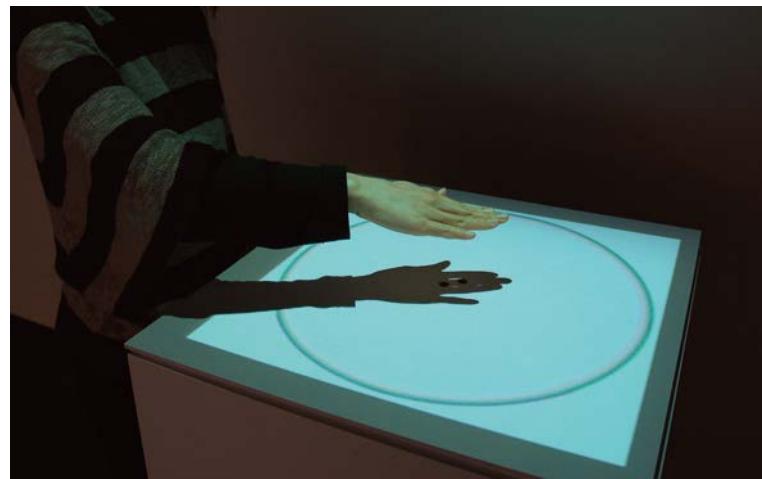


図 2-14 行為と感覚の関係の例

プロジェクトの構想段階で、学生はどのような行為を検知し、どのような動きをデザインするかを検討する。鑑賞者のどのような動きを誘導し、それに応じてグラフィックスを生成するかという基本的コンセプトを考えるために、その前段階の演習を考案した。それは、物理的なオブジェクト(みかん1個、あるいはポテトチップス1袋)を使い、学生はまず手でそれらを触り、触りながら解体していく(図 2-15)。そしてその過程で得られた感覚を言葉で記録し、行為と感覚のインタラクションを体験するというものである。

具体的には次のように行う。例えば、チームの一人がみかんをなでたり、つついたり、皮をむいたり、つぶしたりと、みかんと「戯れる。」そうしながら、どのような行為をしたか、何を感じたかを語り、他のひとりがそれを記録する。図 2-16 にその記録の例を示す。みかんと「戯れる」中で、学生はその行為が感覚を引き起こし、その感覚が新しい行為を起こすこと、そして両者が密接に関係していることに意識的に気づく。この演習後、チームは記録を見直し、行為と感覚の際立った関係を選択し、そこからインスタレーションのアイディアを得ていく。

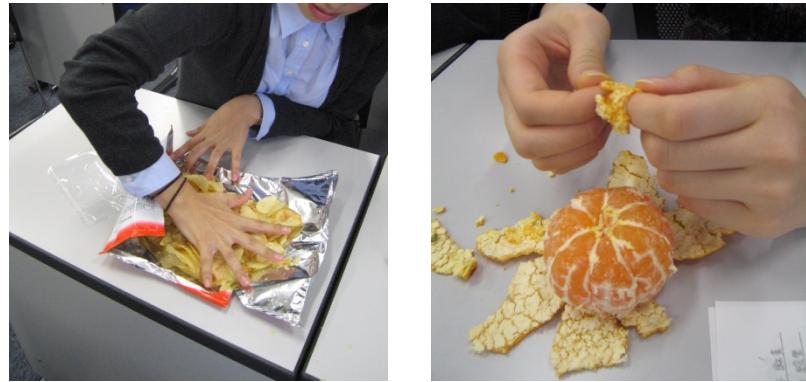


図 2-15 行為と感覚の演習

行為	感覚
<p>さわる</p> <p>にぎる。 細かい二つの見えた</p> <p>かわる かわりかわりする</p>	<p>→ つめたい。 表面さらさら。ほんほん。 強くやるとかゆいそう。 表面は棘川と強くにぎると弾力がある。</p> <p>へとへとす。 たんてんにあいがす。 へしゃへしゃ においでてきた!! スリッとしてくる。 不快。</p>
<p>皮</p> <p>→</p> <p>皮のうのせんぐく バストボーリング しあわせうな</p>	<p>→ 収とりしてる(う) ボアの満んだやつ。 エビフライ作る過程の 卵と小麦粉の工程のとき の手につくやつみたい。 かと—。</p>

図 2-16 行為と感覚の記述演習の記録

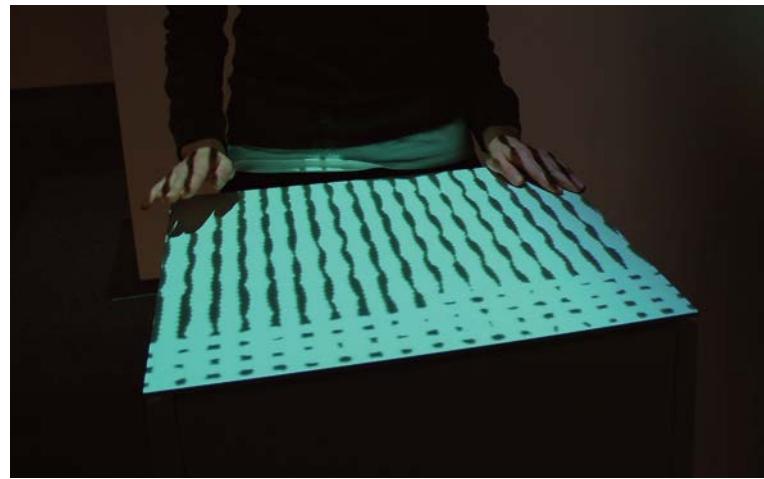


図 2-17 作品例：皮をむくように白線が現れる

例えば、あるチームはみかんの皮の白い部分をはがし取った経験に着目した。白い部分を全部取り除くまで夢中になってやめられず、きれいにはがした後満足したという経験を元に、鑑賞者がテーブルをこすると、皮をはぐようにぎざぎざの線が現れ、盤面を覆っていくというビジュアルイメージを生成した(図 2-17)。鑑賞者は皮を全部はぐようテーブル上が白くなることで心地よく感じるかもしれないし、痛みのような不快感を味わうかもしれない。

2.4 教育プログラムの実践と評価

2.4.1 作品に現れた行為と感覚のインタラクション

本教育プログラムを 2006 年から 2009 年の 4 年間、約 30 名のクラス(2 年次生)に対して実施した(表 2-3)。初年は第 1 バージョンのツールキットを使い、ビジュアルイメージではなく、砂や水といった物理的素材をインターフェースとして使用した。しかし、2.3.2 項で述べたように、例えば鑑賞者が手をかざすとミニチュアの街に光が付くといった固定的慣習的内容、あるいは単純な機械的反応にとどまった。作品が提供する体験により行為と感覚の新しい関係を認知するような試みを探求してはしかったが、物理的な素材の加工に工夫や高度なスキルが必要なこともあり、メカニズムを実現することだけに、満足してしまう傾向に陥った。そこで 2 年目(2007 年)は、センサをテーブル内部に埋め込み、グラフィックスを天板上に投影することで、天板自体をインターフェースの場とする制限を加えた。2007 年は 15 チームのうち、11 チームが円や四角などの基本グラフィック要素を使ってモーショングラフィックスのプログラムを制作した。一方、4 チームはイラスト画像ファイルを使ったが、具象的な画像を使った作品は、プログラムが生成する幾何学的グラフィックスを使ったものに比べて、説明的で文脈依存的な内容であった。枝や実の画像ファイルを使い、手を置くと木が育ち、実を付けるといったアニメーションを生成するプログラムはその例である。学生が「雪が降る」や「花が咲く」といった具象的イメージに固定的に捕らわれていると、ありふれた意味関係を押し付けることになり、それでは新しい行為と感覚の関係を提供できない。

表 2-3 センサリービジョン教育プログラムの実践

	ツールキット	応答部分	サンプルライブラリ	クラス人数
2006 年	第 1 バージョン	物理的オブジェクト (入出力サンプルのみ)		32
2007 年	第 1 バージョン	基本グラフィックス + イラスト (入出力サンプルと一部の運動サンプル)		30
2008 年	第 2 バージョン	基本グラフィックス	2 種のプログラムサンプルライブラリ	30
2009 年	第 2 バージョン	基本グラフィックス	2 種のプログラムサンプルライブラリ	32

そこで、ビジュアルイメージから具象的要素を排除するため、基本グラフィック要素のみを使うことにした。物理的素材を使わない、グラフィックスもプログラムが生成する基本グラフィックスにとどめるという方針を決定し、3年目の実践に入る前に、第2バージョンのツールキット、サンプルライブラリを開発した。ただ、制作するインスタレーションには物理的素材や具象的イメージを使わないとはいえ、人が「もの」に対してどのような感覚関係をもっているかの観察はアイディアを練る上で重要であるので、そのプロセスを助けるため、2.3.5項で述べた「もの」を使ったインタラクションを構想する演習を2008年から導入した。

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く、赤外線センサの使用が2009年に増えたという違いは、どのような行為をインタラクションの中心となる要素にするかに因り生じた。2008年の音センサを使った作品では、鑑賞者が最初は柔らかくタップしていたのが、反応とは無関係なのに徐々にバンバンと強くテーブルを叩く行為をする傾向が見られた。強く叩く行為が誘発されたと言える。この荒い行為は、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2009年の開始時に強く叩くような行為は適切ではないことを指摘した。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2-4に、チームが作品に採用した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2-18は表2-4中aの展示時の写真で、テーブルを叩くことで、白い板面上に、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、赤い染みがテーブルに飛散るような感覚をもつ。テーブル全体が赤く染まると、白に戻る。この作品は、インタラクションを構想する演習で、ポテトチップスを握りつぶし、その油が紙にしみていく感覚から生まれた。

図2-19は表2-4のfで水槽の中の金魚とのインタラクションの感覚をイメージしている。作品名から金魚鉢の中の金魚が音で驚いたり、手の回りに寄ってくるという固定的な文脈を想起させるが、具象的な形を使っていないため、純粹に手と円の動きの関係性から、鑑賞者それぞれが異なる感覚を体験できる。図2-20は表2-4のgで、手を押し付けることで、中央の赤い円の不透明度と大きさが増加し、その後多数の小さい円が飛散る。この作品では、例えばペンキの入った風船が膨らみ、手が弾性を感じ、手の圧力に耐え切れず破裂するというような感覚を体験する。

表 2-4 作品中の行為と感覚のインタラクション

	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. 絵具が飛散る ように全体が真っ赤になっていく
b	The steam	吹く (音センサ)	多数の小さい粒子が天板上を浮遊. 二人の鑑賞 者が天板上の小さい穴に息を吹きかけると, 粒 子は煙のように揺らめく.
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 カリと皮をむく様に, 面を白くする.
e	Perfume	手をかざす, 押さえる (赤外線センサ)	小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. 匂いの粒子 を手で押さえようとする. いったん収束する が, 一定時間後あふれ出る.
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)	中央の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. 粘着性の液体が染 み出し, 面を埋め尽くす感覚.
h	モノ黒	手を軽く押す, 叩く (音センサ, 赤外線センサ)	手で周回スピード, 方向が変化. 音で複数の小 円に放射状に弾ける. みかんをみかんらしくな く再構成. 押せば転がる, 叩けば弾ける.

2008 年の 16 チームのうち 13 チームが, 2009 年の 15 チームのうち 12 チームが, グラフィック要素として円を採用した. これはサンプルライブラリで使用しているグラフィックオブジェクトが円であることに起因している. プロジェクトのプロセスでは, サンプルを元に修正しながら再生産し, オリジナルなプログラムを創作する試みがなされるので, 円が多く使われるのは自然である. サンプルのグラフィックオブジェクトをそのまま使っているが, それは否定的なことではない. 円というシンプルなビジュアルを使うことで, 具象的シンボルのもつ慣習的固定的なイメージにとらわれずに, 作品の内容を練ることができた.

例えば, 表 2-4 の h は, 円が円周上を運動するサンプルと, 図 2-13 の中央のサンプルプログラムから基本的な着想を得ている. 円が円周上を動く速度が赤外線センサの値で変化し, 音がトリガーとなって円が破裂するように小さい円が飛び散る. 手を上下させる行為と速度との関係が微妙な張力の感覚を生み出し, 叩くことで弾け, その緊張感を解き放つ. プログラムの動きはシンプルで, サンプルからの直接の借用であるが, この作品は独自に工夫した感覚経験を鑑賞者に与える例であり. このチームは本教育プログラムの教育目的の成果を示したと考える.

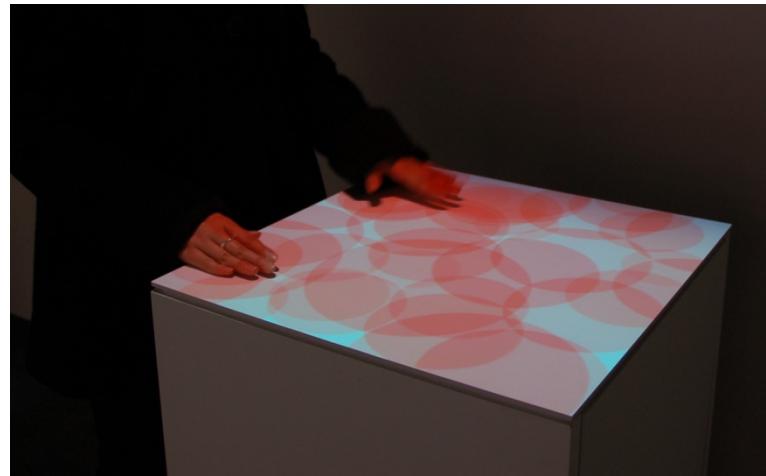


図 2-18 学生作品例：散



図 2-19 学生作品例：KINGYO



図 2-20 学生作品例：弾

2.4.2 作品プログラムコードの分析

インタラクティブなメディア作品としての魅力と、それを実現するプログラムの記述の適切さや独自性とは一致しない。メディアアート作品として鑑賞者に与える感覚経験や新しい発見の評価とは切り離して、ソースコードを把握できている 24 の作品を、サンプルプログラムの流用に着目して分析した。そして、プログラミング学習のフェーズと対応した次の 3 段階で評価した。(i) 模倣 (3 作品)：既存プログラムの模倣にとどまる。サンプルのクラスを直接使用している、あるいは単に描画関連メソッド呼出しの追加や引数の変更を加えたもの。(ii) 再生産 (9 作品)：既存プログラムのメカニズムを流用し、修正したもの。サンプルコードの制御構造に手を加えて流用しているが独自な動きを含まない。(iii) 創作 (12 作品)：独自のメカニズムによる独自の動きを実現した。

創作フェーズの 12 作品のうち、ソースコードにプログラムの直接の流用が見られるのが 6、流用がないのが 6 であった。全作品の 75% 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。図 2-20 の作品は模倣フェーズ、図 2-18 は再生産フェーズ、図 2-19 は創作フェーズの例である。

図 2-21 と図 2-22 は学生作品のプログラムソースの一部を流用元のサンプルと比較したものである(網掛け部分が変更点)。図 2-21 は図 2-18 の作品のプログラムで、入力値の変化量を大きさに反映する部分を流用している。そこに、色や位置の変更、入力値による場合分け、描画した円の個数による場合分け、新たな図形の描画を加えて「絵具を散らす」表現をプログラムで実現しているところが単なる模倣から進んだ部分である。図 2-22 は図 2-19 のプログラムで、サンプルライブラリの振動する円のクラスに対して中央に移動するメソッドを追加し、赤外線センサからの入力値が一定値になった時(手が近づいたら)、呼出すようにしている。このわずかな追加が運動に独自性を与える、サンプルを流用しつつ、創作フェーズといえる発展が見られる例である。

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、「模倣」し、「変更して再生産」し、「作り出す」出発点として重要な役割を果たしている。

サンプルプログラム Bubble(図 2-13 の左)	学生のプログラム：散
<pre>void draw0 { pa = sv.input[0]; // 以前の入力ポートの値を保存 sv.getInputAll(); // 新しい値を入力 henka = abs(sv.input[0]-pa); //入力 0 変化量 x = int(random(width));//円の中央ランダム y = int(random(height)); ellipse(x, y, henka, henka); if(sv.input[1] <= level){ //入力 1 が level 以下なら background(102); // 画面クリア } }</pre>	<pre>void draw0 { pa = sv.input[0]; // 以前の入力ポートの値を保存 sv.getInputAll(); // 新しい値を入力 henka = abs(sv.input[0]+pa); //変化量 if(henka > 150){ cnt=0; x = int(random(width));//円の中央ランダム y = int(random(height)); fill(0,henka,300,50); ellipse(x, y, henka/2, henka/2); } if(cnt > 100){ fill(0,0,255,20); //半透明の四角形を描く rect(0,0,width,height); }else{ cnt++; } }</pre>

図 2-21 再生産フェーズの学生プログラムにおける流用

サンプルプログラム VibrateBall(ランダムに振動する円)	学生のプログラム：KINGYO
<pre>class VibrateBall{ float x, y; // 位置(座標) float originX, originY; float sx, sy; // 移動速度 int wide; //動く範囲幅 // 中略 void display0 { sx = random(min, max); //変化量はランダム sy = random(min, max); x = x+sx; y = y+sy; if(x>originX+wide){//右 x = originX+wide; sx = -sx; } // 中略 ellipse(x, y, d, d); } }</pre>	<pre>class VibrateBall{ float x, y; // 位置(座標) float originX, originY; float sx, sy; // 移動速度 int wide; //動く範囲幅 // 中略 void display0 { sx = random(min, max); //変化量はランダム sy = random(min, max); x = x+sx; y = y+sy; if(x>originX+wide){//右 x = originX+wide; sx = -sx; } // 中略 ellipse(x, y, d, d); } void Shuugou(int kinjichi,int step){ if(abs(x-width/2)>kinjichi) x= x + (width/2-x)/step; else x = width/2; if(abs(y-height/2)>kinjichi) y=y+ (height/2-y)/step; else y = height/2; } }</pre>

図 2-22 創作フェーズの学生プログラムにおける流用

2.4.3 プログラミングの理解に関するアンケート調査

プログラムによってビジュアル表現を生み出す制作を経て、学生がプログラミングの理解に対してどのような認識をもったかを調べるために、プロジェクト開始前にプロジェクトへの期待(自由記述)と技術的知識の理解認識を自ら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技術的知識の自己判定アンケートを実施した。振り返りに関する設問は、プロジェクトは「大いにおもしろかったか／全くおもしろくなかったか」と、新しいことを「大いに学んだか／まったく学べなかつたか」の2つで、それぞれ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。技術的知識を自己判定する設問はプロジェクト前後とも同じ内容で、次の項目について「よく知っている(10)」から「全く知らない(1)」の10段階のいずれかの選択を求めた。

- ・電気の流れや抵抗について
- ・センサから値を得ることについて
- ・A/D(アナログ/デジタル)変換について
- ・プログラムを書くことについて
- ・グラフィックスをプログラムで生成することについて
- ・インタラクティブなメディアアートについて

学生自身の主観的自己評価を問うこれらのアンケートの項目のうち、「プログラムを書くこと」項目は2.2.2項で述べたプログラミング学習の利点の(a)数理的手法による制作力の獲得を反映し、「新しいことを学んだか」項目は(b)達成感・主体感の獲得と関係する。プログラミングの知識を問うテストによって、知識量の変化を推測することも考えられるが、それは行っていない。本プロジェクトのプログラミング学習は、プログラミング知識の獲得よりも、制作表現の幅を広げるためにプログラムを道具として使う意欲と自信を持たせるところに主目的があり、その点に関する学生の主観的認識を、主にこの2つの項目の回答から考察する。

プログラムを書くことについての回答の、プロジェクト前後での変化を表2-5に示す。2年間を合わせると49人中39人(80%)がプログラムについての理解認識の変化がプラスであったと回答し、26人(53%)が2段階以上増加したと回答した。クラスには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人いた。この13人中プログラム理解の認識が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。初学者の理解認識の増加が、既習者よりも大きいが、どちらも約80%の学生がプログラミング理解の認識が向上した。

次に、プログラミング理解の認識と学びの実感の関係を確認するため、回答のあった49名の理解認識の変化量を縦軸に、新しいことを学んだかどうかの回答を横軸にプロットした散布図(バブルチャート)を図2-23に示す。相関係数は2008年0.18、2009年0.15であり、ともに有意な相関

はない。図 2-23 を 4 つの領域に分け、散布図上に A, B, C, D の名を示した。「学んだ感」が平均の 7(2008 年の平均値は 7.3, 2009 年は 7.1) 以上かそれ未満か、また理解認識の変化がプラスかゼロ以下かの 4 領域である。各領域内にプロットされた人数を、アンケートの「面白かったか」項目の回答平均と中央値、プログラミング理解認識変化の平均と中央値とともに、表 2-6 に示した。学生の自己評価として、理解認識の変化がプラスで、高い学びの実感を示すことを期待するが、両年あわせると 63% の学生がそれに相当する A 領域に属する回答をしている。加えて、「面白かったか」項目の回答はどの領域も同じ程度で、プログラミング学習の困難さの認識はプロジェクト全体への興味、学びの意欲を減じてはいない。

表 2-5 プログラム理解認識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%)	3 以上增加	6(22%)	17 人 (81%)	3 以上増加	9(43%)	
変化なし		2 の増加	7(25%)		2 の増加	4(19%)	
		1 の増加	9(32%)		1 の増加	4(19%)	
変化なし			2 人 (9.5%)				
減少	4 人 (14%, -1)			2 人 (9.5%, -1 : 1 人, -2 : 1 人)			

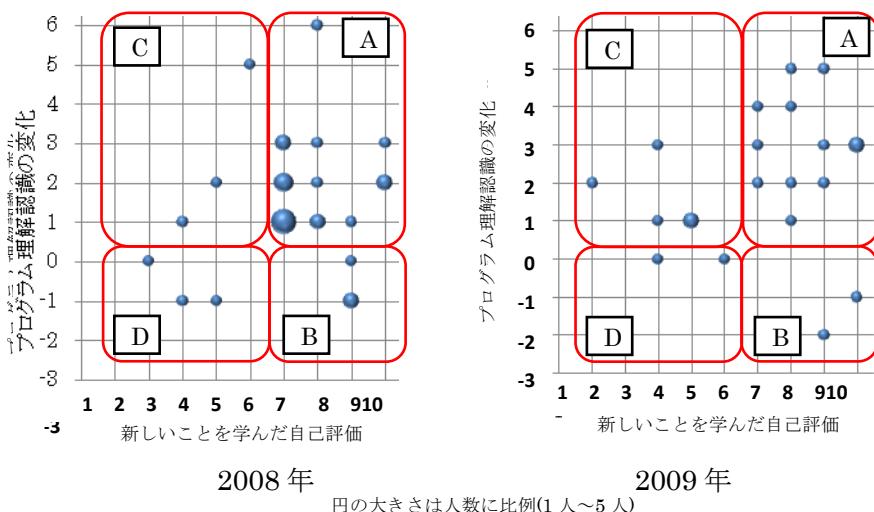


図 2-23 プログラミング認識理解の変化と学びの実感との関係

表 2-6 領域(図 2-23)ごとの人数とアンケートの回答

図 2-23 の領域	2008 年(回答数 : 28 人)				2009 年(回答数 : 21 人)			
	A	B	C	D	A	B	C	D
人数	19 (67%)	3 (11%)	3 (11%)	3 (11%)	12 (57%)	2 (10%)	5 (23%)	2 (10%)
面白かった感平均と中央値*	6.2 6	8.3 9	7.0 7	5.7 5	6.3 6	6.5 6.5	7.0 7	6.0 6
プログラミング理解認識の変化平均と中央値*	2.0 2	-0.6 -1	2.7 2	-0.7 -1	3.1 3	-1.5 -1.5	1.6 1	0 0

* 上段 : 平均値, 下段 : 中央値

2.4.4 自由記述コメントの分析

プログラミング理解の認識や学んだ感についてどのような認識の学生が、どのようなコメントを書いたかを考察するため、プロジェクト終了後アンケートの自由記述に対し、コーディングとカテゴリ化を行った。小規模な質的データの分析手法である SCAT(Steps for coding and theorization)[28]のオープンコーディングの手法を援用し、8つのカテゴリを抽出し、肯定的と否定的の2つに分類した(表 2-7)。カテゴリ「プログラムは困難」は、「プログラムはむずかしい！」と言い放って、理解の困難さの原因をプログラミング側におく姿勢であるのに対し、カテゴリ「制作姿勢への自省」は、「他の課題と重なり集中できなかった」などうまくできなかつた原因を自分側におくコメントである。肯定的カテゴリが、2.2.2 項で述べたプログラミング学習の利点(a)(b)(c)のいずれと関係するかを表 2-7 のカテゴリ名の横に合わせて示した。そして、表 2-7 でカテゴリにつけたマークを図 2-23 の点の上に重ねて図 2-24 とした。実際のコメントの一部を表 2-8 に示す。表 2-8 のマークは表 2-7 と対応しており、カテゴリに属するコメントの具体例がわかる。図 2-23 上のどの学生のコメントかがわかるように、図 2-23 上の領域名と座標も示した。

肯定的カテゴリは、図 2-23 の A,B,C いずれの領域にも現われる。肯定的カテゴリのうち、利点(a)に対応するカテゴリのコメントは、1 例を除き、A ならびに C 領域に現われる。当然ともいえるが、プログラム理解認識の変化がプラスであることが、利点(a)に対応するカテゴリの肯定的コメントの発現へと繋がっている。また、利点(b)の「主体感達成感あり」カテゴリも A,C 領域だけに現われる。プログラミングへの理解が進んだという認識や数理的手法による表現のおもしろさの発見は、主体感達成感を高める関係にある。「将来へ向けての意志」や「表現の可能性」カテゴリは、A,B,C 領域に偏りなく見られる。これらのコメントにはプログラミング理解認識がマイナスの変化であっても、「もう一度やってみよう」「プログラムを動かすことは広い分野への可能性が感じられて新鮮」と、プログラム理解の認識に関係なく、プログラムの知識を次の活動に活用しようとする

意欲が表現されている。自由記述コメントのカテゴリ分析から、9割の学生がA,B,C領域にあたる認識を示しており、これらの学生はプログラミング学習の利点(a)(b)(c)のいずれかを得ていると考える。

否定的カテゴリの「プログラムは困難」のコメントは、4領域いずれにも現われるが、A,B,C領域の学生は「プログラミングは難しく」ても、学ぶ意味を肯定的に捉えているのに対し、D領域には肯定的コメントはない。また、否定的カテゴリの「主体感なし」はDのみに見られる。表2-6に示したように、一割に相当するD領域の学生もプロジェクト全体の面白さの認識は他の領域の学生と違いはないが、プログラミング学習の利点を得られていない。

表2-7 自由記述コメントのオープンコード化によるカテゴリ

肯定的カテゴリ	否定的カテゴリ
◆ プログラムを(少しあは)理解した(a)	✗ 主体感なし
■ 数理的手法による表現(a)	● プログラムは困難
● 主体感達成感あり(b)	◆ 制作姿勢への自省
▲ 将来へ向けての意志(c)	
★ プログラムによる表現の可能性(c)	

表内のマークは表5、図11内のマークと対応。

(a) (b) (c)は、2.2 背景と目的のプログラミング学習の利点と対応

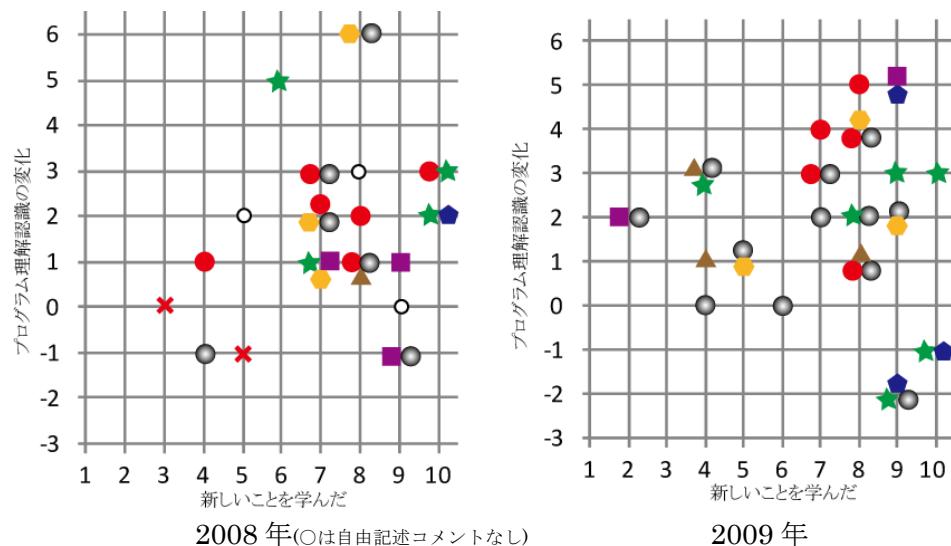


図2-24 自由記述コメントのカテゴリと図2-23上のデータとの関係

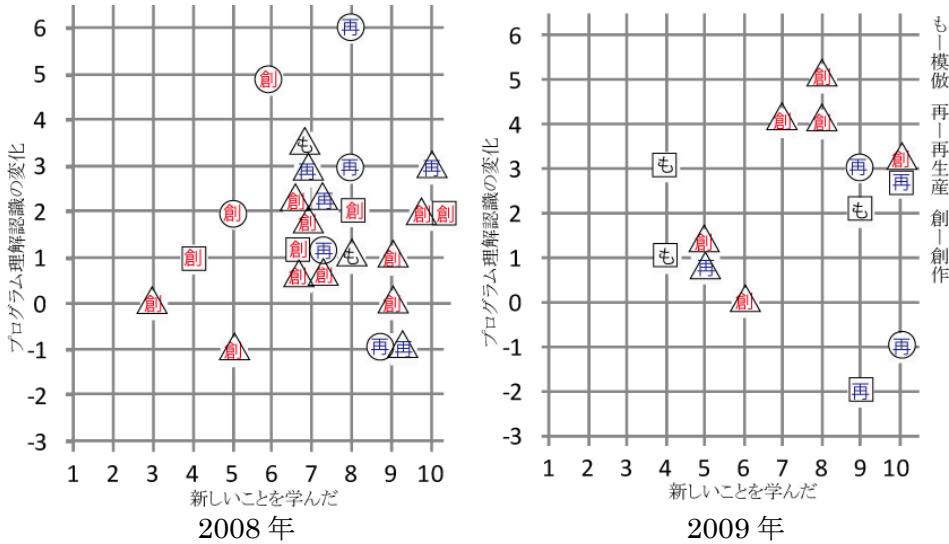


図 2-25 プログラムの評価と図 2-23 上のデータとの関係

2.4.5 プログラミング理解の認識と作品のプログラム部分の評価

図 2-24 にプロットされているどの学生が、2.4.2 項で評価したどのフェーズのプログラムを書いたかを、図 2-25 に示した。アンケート回答とソースコード提出がともに整っている 39 名の学生をプロットしている。A,B,C 領域では大きな偏りはなく、プログラミング理解に関する回答とプログラム部分の評価との間にはっきりとした傾向や関係は見出せなかった。一方、データ数は少ないが、D 領域にプロットされた回答の学生のプログラムは皆創作フェーズであった。創作フェーズのプログラムを書くには、プログラム要素を理解した上に、新しい動きを実現する数理手続きを生み出す必要があるが、D 領域にプロットされた学生の回答とそのプログラムを見ると、高度な動きを目指しすぎて、プログラミングに苦労し、構想を変更して完成させたものの、困難さが勝って主体感を得られない様子がわかる。例えば、2009 年の[6,0]の学生は粒子システムを使い、オリジナルな動きを含む創作フェーズのプログラムを制作したが、ライブラリを使いこなせず、構想通りにはならなかつたため達成感が得られなかつた。粒子システムを使ったサンプルは、数理的表現の美しさや発展性を示すよい例であるが、シンプルな運動と比較すると高度である。そのようなプログラムを模倣の出発点とし、独自の動きにチャレンジしたことが、結果として反対の効果を招いた。途中で挫折せず、プログラミングの学習意欲を持続するためには、出発点となるサンプルプログラムのシンプルさが重要であることを示唆している。

表 2-8 プロジェクトで学んだことに関する自由記述コメント

図 2-23 での位置*1	マーク *2	肯定的振り返りコメント	マーク *2	「プログラミングが大変」だったことに関するコメント
A 2008 年 [7,3]	●	思った動きが実現した時はとてもうれしい。	●	もっている能力が少なすぎて、できることが少なかったのが残念。
A 2008 年 [8,6]	◆	以前はわからなかつたプログラムを理解できるようになった。	●	プログラムが難しく、思うとおりに動かなかつたときは苦しかった。
A 2008 年 [7,1]	●	プログラムを使って自分の感覚を形にしていくのがおもしろかった。	●	プログラムがなかなかうまくいかなかつたので、ちょっとしんどかった。
A 2009 年 [10,3]	★	自分(助けは要るが)にもできるという可能性。	●	プログラムが難解で、サポートしてもらわないとわからなかつた。
A 2008 年 [8,1]	▲	もっとプログラミングを理解してもう一度作ってみたい。	●	やりたいこととプログラミング能力が一致せずにとても大変だつた。
B 2008 年 [9,-1]	■	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	●	プログラムの難しさを学んだ。かなりつらかった。
B 2009 年 [9,-2]	★	センサ, I/O モジュールを使い、プログラムを動かすことは広い分野への可能性が感じられて新鮮。	◆	多忙な状況の中、忙しくちゃんとプログラムについて考えられなかつた。
C 2009 年 [5,1]	◆	プログラムを前よりは少し理解できるようになった。	●	プログラムが思った通りに進まないことが多かつた。
C 2009 年 [4,1]	▲	思った通りにならなくとも、それに近づけるようにする向上心を学べた	●	プログラミングが難しすぎた
C 2009 年 [4,3]	★ ▲	プログラムによっていろいろな結果が得られるところがおもしろい。思い通りにならなくとも、理想に近づけようとするくじけない心を学んだ。	●	プログラミングはとても難しい。
D 2008 年 [3, 0]			✗	プログラミングが複雑で、自分で作り上げた気がしない。
D 2008 年 [4,-1]			●	プログラミングの知識のなさを学んだ。

*1 : A,B,C,D は図 2-23 の領域を, [M, N] は散布図上の座標で, M は学んだ感を, N はプログラミング理解度の変化を表す.

*2 : 表 2-7, 図 2-24 のマークと対応している.

メディアアート作品としての評価は、鑑賞者の主觀で成立するので、プログラミング学習の認識や意欲との関係を論じられないが、参考までに、授業終了時の教員による作品の評価を図 2-25 にプロットの形状で示した。鑑賞者に与える体験の豊かさ、アイディアの斬新さを教員 2 名が主觀的に 3 段階で評価したものである(○は A, □は B, △は C の 3 段階)。創作フェーズのコードを書いた 21 人中 A は 2(10%), B は 4(19%), C は 15(71%), 再生産フェーズ 13 人中 A は 6(46%), B は 2(15%), C は 5(38%), 模倣フェーズ 5 人中 A はなし, B は 3(60%), C は 2(40%) の評価であった。創作フェーズのプログラムは作品としての評価は C が多い。オリジナルな動きを導入して行為と感覚の体験を実現しようとしたが、実装が中途半端な形にとどまった結果、作品としての評価が低くなっている。模倣フェーズのプログラムは、作品としての評価が B または C で、模倣のままだと行為と感覚の新しい体験を生み出すまでに至らない結果が表れている。

2.5 結論

フィジカル・インタラクションによる鑑賞者の行為に応じて生成し変化する動的なグラフィックコンテンツの制作を行う教育プログラムを構築し、実践のためのツールキット、サンプルライブラリを開発した。実際の教育に適用したところ、人の行為と感覚の繊細で多様性に満ちた関係性を実現した課題作品が制作され、インタラクティブ性を探求する面において、この教育プログラムが有効に働くことを確認できた。

授業内ではプログラミングの学習に取れる時間が限られるため、サンプルライブラリを使い、「模倣する→再生産する→創作する」の段階的フェーズを通して自学により理解を進めた。サンプルライブラリは、シンプルなグラフィックスの運動によるインタラクションが人の感覚にどう捉えられるかの発想の基本として、また、ソースコードを流用することでプログラムの土台として、有效地に機能した。数理的表現の工夫がインタラクティブなモーショングラフィックスとして直ちに反映されることが、驚きを生み、それが 3 つのフェーズの学習過程を通して繰り返されることが、学習の意欲を高めた。結果として 8 割の学生にプログラミングの理解の認識の向上をもたらした。9 割の学生はプログラミングはむずかしいと考えながらも、プログラミング学習に期待される利点を反映する認識を示しており、モーショングラフィックスを生成するプログラム制作が学習意欲を向上させたと考える。しかし、1 割程度の学生はプログラミングに否定的な認識のみを覚えることになった。プログラミング学習のための教材の開発、サンプルプログラムの提示方法の工夫を中心とした改善が必要と考える。

第3章 インタラクティブなビジュアル表現を作り出すプログラミング学習教材

3.1 一般情報処理教育におけるプログラミング

3.1.1 必要性についての提言

学生の専門に関わらずプログラミングの基礎を学ぶことは、次のような重要な利点をもたらす。プログラミングの知識はソフトウェアによって創り出せるものの幅を広げ、ユーザを情報の消費者から制作者に変える。また、プログラミング学習を通してアルゴリズミックな思考体験することで、数理的な問題解決のための繰返しやモジュール化などの手法を理解することができる。プログラミング学習のこのような利点やその必要性に関して、例えば次のような提言や研究がある。

アメリカ National Research Council は情報技術リテラシーに関する調査報告[3]で「Fluent with Information Technology (FIT)」の重要性を訴えた。その中で Digital Fluency を自己を創造的に表現する能力、知識を形を変えて表現する能力、情報を単に理解するだけでなく生成する能力と定義した。そのうえで、情報技術に関する知識のタイプを 3 つの種類(Intellectual Capabilities, IT Concepts, IT Skill)に分け、それぞれ 10 項目をあげている。このうち、IT Concepts の「Algorithmic thinking and Programming」がプログラミングに関する項目で、アルゴリズミックな考え方は IT 技術の多くの側面を理解するキーであると位置づけ、プログラミングに付随したスキルは Digital Fluency の発達において重要な役割を果たすと述べている。日本でも FIT の考えを踏襲する形で、例えば上智大学の情報フルエンシーに関するクラス[29]において、プログラミング教育が展開されている。

続いてアメリカでは、Wing が「Computational Thinking」の概念に基づき、情報教育への戦略を提案している[30]。これは、読み、書き、算術に次ぐ 4 つめの基礎的能力と位置づけられた概念で、人の創造性や知性をコンピュータを使って実現するための基本的な考え方やスキルを対象とする。主にアルゴリズミックな手法で問題解決を図ろうとする思考や能力を指し、その向上のための教育を推進する活動が、Wing を中心に進められている。

また、日本情報処理学会は情報処理教育委員会「日本の情報教育・情報処理教育に関する提言 2005」[31]において、「大学の一般教育において、『手順的な自動処理』についての制作体験をさせる」ように提案している。2007 年、日本情報処理学会は、情報教育のカリキュラム標準 J07 を策

定し、その中の一般情報教育(J07-GE)[32]では、一般情報教育の知識体系(GEBOOK)にアルゴリズムとプログラムを含めている。

Mateasは「コンピュータによる数理的手続きを読み書きする能力」をProcedural Literacyと呼び、プログラミングスキルをその重要な要素として位置づけ、このリテラシーは理工学系の学生だけでなく、人文科学分野の学生にも必要な能力であると主張した[33]。ただ、工学・科学の学生と芸術・人文を学ぶ学生の間にはギャップがあるので、コンピュータ科学の入門科目を実施するだけでは不適切であるとも述べている。

このような情報教育の立場からだけでなく、ソフトウェア・スタディーズ分野の研究者であるManovichは、21世紀において、文化的生産物の作成、保存、配布、アクセスの方法をソフトウェアが担い、現代社会のエンジンになっている状況を踏まえ、ソフトウェアの理解が重要と主張した[34]。その中で、メディア・文芸理論家のKittlerの「現在の学生は少なくとも二つのソフトウェア言語を習得すべきだ。それでようやく現在の文化について語ることができるようになる」との言葉[35]を引用している。

3.1.2 その実際と問題点

プログラミング教育の重要性を受けて、大学においても、これまで多くの人文系学科で一般プログラミング基礎教育が実施されている。専門課程と同様のプログラミング入門教育を実施する、あるいは一般プログラミング教育に特化した環境と内容を用いるなど、そのアプローチはさまざまである。

MoonsとBacker[36]は、プログラミング入門教育のアプローチを、(1)ある確立した順番に従う、(2)構成主義(Constructivism)的なアクティブラーニングの手法を使う、(3)初心者向けに開発された言語を使う、(4)特別な目的をもって開発されたプログラミング環境を使う、という4つに分類している。(1)(2)は使用言語に関係ない教育方法に、(3)(4)は使用言語や環境に着目した分類である、例えば、(3)の初心者向けプログラミング言語を使い、(1)あるいは(2)の手法で教育を展開するというアプローチがとれる。MoonsとBackerは構成主義的手法を考慮することが重要との考えに基づき、独自のプログラミング環境を開発した。

上記(1)のアプローチは変数、制御構造、関数(メソッド)呼出しなどの基礎要素を知識として順序だてて伝える手法である。現在多くの場合、このアプローチがとられている(例えば[37])。C、C++やJavaといった実用言語を使い、このアプローチによる人文系学科でのプログラミング教育は、例えば数や文字列を「うまく処理」するプログラムを作るような題材を取り上げることが多く、数学的パズルを解くのに興味をもつ学生にはおもしろいが、一般に人文系学科の学生には興味のわかない内容となりがちである。加えて、一番単純なプログラムを書くにも、多くの文法要素を説明

しなければならない問題も繰返し指摘されている[38]。BASIC や Pascal, Lisp のような教育用プログラミング言語では、文法的な困難さは避けられるが、取り上げる題材の適切さの問題は残る。

この問題に対しては、ビジュアル表現を作り出す題材を取り上げることで、解決を図る工夫がなされてきた。Papert は構成主義の考えを発展させて、人工物を構築することで学びを進める構築主義(Constructionism)を唱え、教育用プログラミング言語 LOGO を開発した[17]。タートルグラフィックスの考えを取り入れたプログラミング教材[39]やプログラミング環境[40]が開発されている。さらに発展したビジュアル表現を使う Scratch[41], SqueakEtoys[42]や 3D アニメーションを作る Alice[43]などの教育用プログラミング環境が研究されている。

2.2.1 項でも触れたように、モニタ上でグラフィックスが動くだけではなく、物理的な「もの」を用いて、それをプログラムで制御することで、プログラミング学習の楽しさや効果を上げようという工夫もなされている。例えば LEGO Mindstorms[1]を使い、ロボティックスに焦点をあてたプログラミング入門教育(例えば[10], [11])の多くは、車のある経路に沿って走らせるといった比較的狭い問題設定のなかで、その解決を図る教育内容である。しかし、このように特定の工学的視点での問題解決に焦点をあてた題材では、ロボティックスに関心がない学生にはプログラミング学習の動機づけとして有効ではない。それに対して、Rusk と Resnick らは Scratch をプログラミング環境として使うセンサ、アクチュエータのキット PicoCricket を開発し、子供向けの教育に適用した[44]。構築主義的学びの考え方のもと、個人の関心や動機づけを元に、オープンエンドのテーマでワークショップを行い、定められた問題を解決するだけでなく、問題を発見する能力の発展に繋がると述べている。Resnick らの教育的アプローチはすばらしいが、大学においては、子供向けのプログラミング入門教育と違って、ただ楽しい充実感だけではなく、また汎用的な機能をもつ、実際の問題解決にも使えるプログラミング環境で教育ができるのが望ましい。

2 章で述べたセンサリービジョン(SV)教育プログラムは、新しい形のインタラクションを発想できる基礎的素養の養成につながるような題材を取り入れ、プログラミング教育の新しいアプローチを構築することを目指した[45][46]。

3.2 本教材の目的

SV 教育プログラム内では、十分な時間を直接的なプログラミングの説明にあてられないこともあり、2.3.4 項で述べたようにサンプルを出発点としてプログラム制作を進めた。しかし、それだけではプログラミング学習の観点からは十分でなかった。プログラミング基礎の理解がなくても、「作品」としてのプログラムを作ることはできる。模倣、再生産の過程で、プログラミングの基礎

要素を学ぶことを想定しているが、模倣したサンプルに含まれない基礎要素は学ばないままになってしまう。Processing 自体が、プログラミングの基礎知識がなくても、グラフィックスを描くことができるプログラミング環境であるため、理解のないまま、とりあえずは動けばいいと考え、プログラム内部の理解をおざなりにする傾向もあった。また、約 1 割の学生はプログラミングを理解したという認識を持てなかった。きちんと基礎要素を理解していれば、自作プログラムの修正、発展も容易であり、別の機会での問題解決にプログラムを有効に適応でき、さらには達成感、主体感も上がる。そこで、プログラミング基礎の学習面を強化するため、Processing で描画することと、プログラミング基礎要素を連動しながら学べるような、主に自習に使用する教材を開発した。教材には、次のような内容を含むことが求められる。

- (1)センサリービジョン教育プログラムへの適用のため、インタラクティブなモーショングラフィックスのプログラムの作成を段階を追って学べる。
- (2)プログラミング基礎要素理解のため、基礎文法項目を順序だてて学べる。
- (3)サンプルプログラムの実行結果を教材の中で説明とともに確かめられる。
- (4)知りたい項目に直ちにアクセスできる
- (5)プログラムの制作時に、必要に応じて、いつでもどこでも参照できる。

教材の形態として、紙あるいは Web 媒体の選択肢がある。(3)(4)(5)の点を考えると、Web 教材として開発するのが適切で、それによりプログラムを開発している同じモニタ上で、動きを比較しながら参照できる。図 3-1 は開発した Web 教材のスナップショットである。



図 3-1 Processing 学習 Web 教材

導入	ラーニングトレイル	基本文法知識	サンプルプログラム
1 Processing とは 2 線を描く・動かす	1 まずは図形を描く 2 図形を動かす 3 マウスに応答 4 関数を作る 5 イベント処理 6 運動と位置の計算 7 ビジュアル・ハーモニー 8 フラクタルを描く 9 画像ファイルを描く 10 クラスを使う 11 音を使う 12 音に反応する	1 変数、式、文 2 算術演算子 3 条件分岐と繰返し 4 配列 5 関数の定義 6 クラスとオブジェクト 7 クラスの定義 8 クラスの拡張	1 運動 ・一方向 ・往復 ・振動 2 インタラクション ・マウスの位置 ・マウスの移動速度 ・閾値によるスイッチ
基本関数			
1 ウィンドウと色 2 基本図形 3 移動回転拡大縮小 4 算術計算 5 タイポグラフィ 6 画像データ			

図 3-2 Processing 学習用 Web コンテンツの構成

これを SV 教育プログラムの実践において、プログラミングを理解するために自習教材として使用する。加えて、この教材は、SV 教育プログラムとは独立した一般プログラミング教育の場においても活用できると考えている。SV ツールキットのようなセンサーデバイスのない場面においても、パソコンの内蔵マイクを用いることにより、マウスやキーボード以外のインタラクションをもつモーショングラフィックスを制作することで、学びのモチベーションを高められる。

3.3 Web プログラミング学習教材の内容

3.3.1 教材の構成と内容

教材は、3つの学びの流れを提供する：

- ・ラーニングトレイル(円をひとつ描くところから始め、それを動かす、反応させるという流れの中で、モーショングラフィックスのプログラム作成を学ぶ)
- ・基本文法知識を学ぶ
- ・サンプルプログラムから学ぶ

教材全体はこれら3つの中核部分に加え、導入部、Processing 基本関数のリファレンス的解説部からなる(図 3-2)。Web の特性を活かして、各部分の関連する内容には相互にリンクが貼られ、3つの学びの流れは相互に行き来できる。プログラムのソースコードを提示する部分では、コードとその解説だけではなく、Web ページ上でプログラムがインタラクティブに応答して動くようにしてある。つまり、Processing を起動し、ソースコードをコピーして実行しなくとも、Web ブラウザ上で、直接実行結果を確かめられる。

プログラミングの初学者は、導入部からラーニングトレイルへと進み、その中に出てくる基本

文法項目についての解説を途中読みながら、音に反応するインタラクティブなグラフィックスを作成する最終部分に至る。「基本文法知識」を学ぶ流れは、プログラミング文法の基本項目の全体を見通すためのもので、プログラミング入門書と同等の構成である。ラーニングトレイルの中の関連する文法要素からリンクが張られており、学習者は必要に応じて参照する。それだけでなく、ラーニングトレイルでの一連の学びが終わった後に、「基本文法知識」の流れを追うことで、基本文法知識全体の理解を整理できる。「サンプルプログラム」から学ぶ流れでは、SV 教育プログラムで開発した基本運動ならびにセンサリーインタラクションのサンプルプログラムとその解説を示している。作品制作の企画段階で、これらサンプルの実行を見ながら、合わせてソースを理解することで、SV 教育プログラムでの作品作りのアイディアを掴むことを促す。センサリーインタラクションのサンプルは、Web ブラウザ上での実行のため、マウスの画面上での位置をセンサの入力値の代わりに使って、動きを確認できるようにしている。

本研究がプログラミング学習の基礎要素と考え、教材に含めた内容を表 3-1 に示す。ACM と IEEE が共同で定めた Computer Science Curricula 2013[47]の中で、これらがどのように扱われているかを合わせて示した。Software Development Fundamentals(SDF)内の知識ユニット Fundamental Programming Concepts の Core-Tier1(基礎段階で学ぶ内容)として提案されている全項目を含み、それを基本として、複数のグラフィック要素を扱う際に欠かせない配列とクラスの基礎概念を加えた。クラスとオブジェクトの概念は、同じふるまいをするグラフィック要素を複数扱う際に必要となるのに加え、Processing が提供するサンプルでも使われており、その理解に必要となるからである。

表 3-1 学習の対象となるプログラミング基礎要素

学習項目	Computer Science Curricula 2013 での対応する箇所
変数、データ型	SDF, Fundamental programming concepts, Core-Tier1
演算子、代入	
条件分岐、繰返し	
シンプルな入出力	
関数と引数渡し	
再帰の概念	
配列	SDF, Data structures, Core-Tier1
クラスの定義(フィールド、メソッド、コンストラクタ)	PL, Object-oriented programming, Core-Tier1

SDF と PL は Computer Science Curricula 2013 の Knowledge Area(KA)のひとつ。
SDF: Software Development Fundamentals, PL: Programming Language

アメリカ National Research Council はその提言([3])で， Fluent with Information Technology(FIT)が求めるプログラミング経験として，条件分岐，繰返し，関数(functional decomposition, functional abstraction)を上げており，FITの要求を満たす内容となっている。

3.3.2 ラーニングトレイル部の詳細

センサからの入力に応じてグラフィックスを動かすプログラムを作成することが，3つの学びの流れの中で中核となるラーニングトレイル部の目的である。そのようなプログラムを書くために学ぶべき要素として，表 3-1 に示した基礎要素全体に加え，基本描画，位置の計算，応答(イベント処理)を含むこととした。またセンサリービジョン教育プログラムへの適用を考慮し，インターフェイティブな応答に音声入力処理を加えた。

ラーニングトレイルで扱うサンプルはコードを少しでも簡潔にするため，素材グラフィックスには単純な円を使った。項目ごとに取り上げる題材と，そこで学ぶプログラミング基礎項目，学習の目的について述べる。

(1) まずは図形を描く

「円をひとつ描く」プログラムを通して，関数，引数渡しを理解する。次に「円を複数描く」プログラムに進み，規則性を発見し，変数を使って変更を容易にする工夫をする。さらに繰返しを理解し，プログラムを見通しよくしていく。合わせてデータ型，式，代入，演算について学ぶ。

(2) 図形を動かす

図形の大きさ，位置を変えることで，プログラムの実行に動きを作りだす。アニメーションを繰り返すため，条件分岐を学び，応用できるようにする。

(3) マウスに応答

マウスに追随する図形を描くプログラムを通して，位置やその変化量に対応する処理を理解する。イージングの処理を理解し，その効果を加えることで，動きの印象が大きく変わることを知り，自分のプログラムに適用できるようにする。

(4) 関数をつくる

同心円を描く関数の定義を通して，関数による抽象化の概念を理解する。新しい関数を定義し，使えるようにする。

(5) イベント処理

マウスクリック，ドラッグ，キーボード入力に応答するため，イベント処理関数を定義する。SVツールキットを使う教育プログラムでは，マウスやキーボード入力は不要であるが，PC モニタ上で実行されるプログラム作成の学習のことを考慮して，この項目を設けた。

(6) 運動と位置の計算

基本的な物理の計算式を使い、加速運動、ばねの運動をシミュレーションする。数式による位置の計算が新しい動きを生み出すことを理解し、自分のプログラムに応用できるようにする。

(7) ビジュアル・ハーモニー

古典的モーショングラフィックスの作品である Whitney の Digital Harmony[48]を再現するプログラムを作成する。これを通して、シンプルな数式の表現が、驚きの効果を生み出す体験をする。段階を踏んだ作成の途中で、複数の円の位置を保持するために配列について理解する。

(8) フラクタルを描く

簡単なフラクタル画像を描くプログラムを書くことを通して、再帰処理を学ぶ。

(9) 画像ファイルの描画

あらかじめ用意した画像ファイルを描画することで、パラパラアニメを実現する。1000 を超えるような多数のオブジェクトを描く場合、画像ファイルを使う方が描画処理が早い。ファイルからのデータ入力とともに、画像ファイルによる描画方法を理解する。

(10) クラスを使う

クラスからオブジェクトを生成して使うことを学ぶ。UI 部品としてのスライダークラスを取り上げ、その API からコンストラクタ、メソッドについて知る。スライダークラスのオブジェクトを生成し、フラクタルの描画をスライダーで操作するプログラムを作成する。自分でクラスを定義することを理解するとともに、Processing を拡張するために開発されているライブラリを使う準備となる。

(11) 音を使う

音声を扱うための拡張ライブラリ Minim を使い、あらかじめ準備した音声ファイルの再生と、その音量に応答して動くグラフィックスを描くプログラムの作成を行う。クラスの API から使い方を知って、利用することを学ぶ。

(12) 音に反応する

外部から得る値を描画の位置や大きさに反映させる実例として、パソコン内蔵マイクの音を入力して、それをグラフィックスの大きさの値に使うプログラムを取り上げる。入力値の範囲を、大きさや位置に使う別の範囲に割り当てる(マップする)ことで、入力値を大きさや位置に適切に反映させる方法を理解し、使えるようにする。SV ツールキットを使う場合、センサからの入力値はセンサの種類や周辺環境によってさまざまなので、その値に応じて調整して入力値を使うことを知る。

このラーニングトレイルの内容が、インタラクティブなモーショングラフィックスを作り出すプログラムを書く際に学ぶ項目として十分かを検討する。そのような場面における学習項目のひとつの提案として、Processing 開発者の Reas と Fry の入門書[49]がある。それと比較して検討するた

め、表 3-1 の基本要素に加え、入門書が取り上げている項目、本教材のラーニングトレイルが取り上げた項目を表 3-2 に示し、それらの扱いについて整理した。

表 3-2 動的グラフィックス生成のためのプログラミング学習項目

学習項目	本教材*1	Getting started with Processing[49]*2
基礎要素 (表 3.1 の項目)	変数、データ型	ラーニングトレイル(1) ○
	演算子、代入	ラーニングトレイル(1) ○
	条件分岐、繰返し	ラーニングトレイル(2) ○
	シンプルな入出力	ラーニングトレイル(9) 画像ファイルの描画 ○
	関数と引数渡し	ラーニングトレイル(4) ○
	再帰の概念	ラーニングトレイル(8) ×
	配列	ラーニングトレイル(7) ○
	クラスの定義	ラーニングトレイル(10) ○
図形の描画	ラーニングトレイル(1) ○	
応答(イベント処理)	ラーニングトレイル(3)(5) ○	
動きの記述	スピード・方向	ラーニングトレイル(6) ○
	motion tween	ラーニングトレイル(6) ○
	乱数	基本運動サンプル ○
	タイマー	センサリサンプル ○
	移動・回転・拡大・縮小	関数のみ紹介 ○
音声入力	ラーニングトレイル(11)(12)	△ 拡張機能を紹介

*1 項目を取り上げている教材内の場所

*2 項目を取り上げているか(○)、いないか(×)

表 3-1 の基礎要素に関してみると、再帰の概念以外は共通している。再帰の概念が入門書[49]には含まれないのは、生成系のグラフィックスを制作しない場合には再帰は必ずしも必要ではなく、入門レベルを超える内容と判断されたと想像される。「動きの記述」に関連する項目では、オブジェクトの位置を式で計算して描画し、変数の値を変えることで動かすという点は共通している。また「動きの記述」の中の乱数、タイマーは本教材ではラーニングトレイルの中では取り上げず、サンプルプログラムとして使い方を示している。意外性のある動きを実現したり、場面転換には欠かせない要素であるが、円の位置を計算して動かすという中心の流れの理解を進めるため、それから離れる要素はラーニングトレイルから除いた。また、同じ理由で、移動・回転・拡大・縮小も関数を紹介するにとどめた。

3.4 教育実践

3.4.1 Web 教材を使った Processing 勉強会

プログラミングの自習教材として開発した Web 教材を使ったプログラミング学習が、プログラミング基礎の学習に効果を生むかを、通常の Java によるプログラミング授業と比較することで検証した。

Processing による学習は 2013 年秋学期に自主的勉強会の形で行い、プログラミング初学者 8 名(4 年次 : 1 名, 3 年次 : 1 名, 2 年次 6 名)が参加した。週 1 回 90 分のセッションを 15 回行った。Web 教材のラーニングトレイルを読んできた後、講師役の筆者が勉強会にて質問に答え、Web 教材中の演習問題のプログラムを各自作成した。一方、Java 言語による学習は、2013 年と 2014 年春学期の入門プログラミング科目で週 1 回 90 分 2 コマの授業を 15 週実施した。受講生は 2013 年 19 名、2014 年 36 名であったが、アンケート未回答、クイズを 2 回以上欠席した学生、Processing の勉強会に参加した学生(2014 年の 3 名)を除き、2013 年 18 名(4 年次 : 1 名, 3 年次 : 2 名, 2 年次 : 15 名)、2014 年 27 名(3 年次 : 7 名, 2 年次 : 20 名)を比較の対象とする。いずれも情報メディア学科にて実施した。

どちらの教育実践でも 15 週のクラスセッションの中で、基礎知識を問うクイズを 5 回実施し、また学習開始前と後で基礎知識への理解認識の程度をアンケートにより回答してもらった。クイズは、変数とデータ型、繰返し、条件分岐、配列、メソッド(Processing では関数)の 5 つの基本的項目の理解を問う問題を、学習の進行に合わせて実施した。問題は、Java SE Bronze Oracle Certification Exam. で出題される内容と同等レベルとし、文法的な正誤を問う、プログラム実行時の変数の値を問う、コードの断片を記入する形の 5-10 の質問からなる。図 3-3 は、条件分岐の問題の一例である。

次のコードの後ろでは a の値はいくつになりますか
int x = 50;
int y = 100;
int a = 0;
if (x >= 100 && y >= 100) {
 a = x+5;
} else {
 a = x-5;
}

図 3-3 クイズの例(条件分岐)

プログラミング基礎要素に対する理解の認識については、クイズで問うた文法要素である変数、データ型、条件分岐、繰返し、配列、引数の 6 項目に対して、「よく理解している」から「全く理

解していない」の 4 段階で回答を求めた。学習終了後のアンケートでは、プログラミングに対する自己効力感(self-efficacy)についても尋ね、加えて「学習を通して何を学んだか」に対して自由に記入してもらった。自己効力感は自分のプログラミング遂行能力に対する予測を問うもので、次の 7 項目に対して、「全く自信がない」から「大いに自信がある」の 7 段階で回答を求めた。

- (1) 文法的に正しい Processing(Java) のプログラムを書ける
- (2) 自分が書いたプログラムなら多少長くて複雑でもデバッグできる(エラーを直せる)
- (3) 解説サイトや本、リファレンスがあれば、プログラムの課題を完成できる
- (4) 私が詰まった時、誰か助けてくれれば、課題を完成できる
- (5) プログラムを完成させる十分な時間があれば、課題を完成できる
- (6) 詰まった時でも、そのままプログラムの課題を続ければ、解決する方法を見つけられる
- (7) ソースコードを見て、処理がどう進むかを頭で追って考えられる

自己効力感は Bandura[50]が提唱した概念で、具体的な目標を成し遂げられる力があるという感覚を指す。教育やビジネスの幅広い分野で自己効力感と学習の達成度や意欲などの関係が研究され、自己効力の信念が学業課題に対する努力とねばり強さを強化するという結果が報告されている。プログラミング教育においても、多くの研究があり、Ramalingam と Widenbeck は、プログラミング学習に関する Self-Efficacy Scale を考案し[51]、プログラミング教育の効果に関する研究で使われている(例えば、[52][53])。Ramalingam と Widenbeck の Self-Efficacy Scale は 32 項目からなるが、学習内容や言語環境に全く依存しない上記の 7 項目を選択し、学習終了後アンケートに使用した。

表 3-3 に、Processing 勉強会と Java プログラミング授業でのクイズ成績(1)、プログラミング基礎要素に対する理解認識の学習前後での変化(2)、学習後の値(3)、自己効力感(4)のクラス平均値と中央値を示した。さらに、個々人の認識や自信について詳しく見るために、「何を学んだか」に対する自由記入的回答から、コメントを切り出し、SCAT[28]の手法に従い、8 つのカテゴリに分類した(表 3-4)。表 3-3 には両者の平均値に対するウェルチの t 検定の結果を載せた。リッカート尺度データの解析に関して、t 検定とノンパラメトリック法であるマン・ホイットニー・ウイルコクソン検定の検定力に差はないという Winter と Dodou の研究報告[54]があり、本論文では両群の傾向を示すのに有用と考え、平均値の t 検定を示した。

3.4.2 Web 教材を使った SV 教育プログラムの実践

SV 教育プログラムを、表 3-5 に示すような流れの 11 週のワークショップ形式(週 1 回 90 分のセッション)で 2014 年に実施した。Web 教材をプログラミングの自習に使い、チュートリアルでも参照した。受講生は 10 名(大学 3 年次 : 4 名、大学 1 年次 : 3 名、高校 3 年次 : 1 名、高校 2 年

次：2名)で、2名ずつがペアとなって制作を行った。プログラミングの理解にあてるチュートリアルは、2008年、2009年の実践よりも1週短い3週である。

2章で述べた2008年と2009年のSV教育プログラムの実践[45][46]では、Processing学習用のWeb教材はまだなく、サンプルプログラムのソースファイルを個別に参照することで、プログラミングの理解を進めながら、SV教育プログラムのワークショップを実施した。これらの実践とプログラミング学習Web教材を導入した2014年の実践とを比較するため、「プログラムを書くことについての知識量や理解度はどの程度か」を10段階で学習前後に問うた結果を表3-6に示す。

表3-3 Processing勉強会とJava授業の比較

	Processing 勉強会 ^{*5} (N=8)	Java授業 ^{*5}		勉強会とJava授業の値 のウェルチのt検定	
		2014(N=27)	2013(N=18)	2014	2013
(1) クイズの平均 (各人の平均のクラス平均) ^{*1} と中央値	6.3(SE: 0.90) 7.0	4.5 (SE: 0.35) 4.6	4.1 (SE: 0.46) 3.8	t(9)= 1.87 p=0.09	t(11)=2.14 p=0.06
(2) 基礎項目の理解認識の変化値のクラス平均 ^{*2} と中央値	6.9(SE: 1.3) 6.5	5.9 (SE: 0.50) 6.0	6.1 (SE: 1.1) 5.0	t(9)= 0.71 p=0.49	t(16)=0.44 p=0.67
(3) 学習後の基礎項目の理解認識のクラス平均 ^{*3} と中央値	3.2(SE: 0.21) 3.2	2.8 (SE: 0.079) 2.7	2.8 (SE: 0.11) 2.8	t(9)= 1.74 p=0.12	t(11)=1.58 p=0.14
(4) 自己効力感の平均(各人の平均のクラス平均) ^{*4} と中央値	3.7(SE: 0.33) 3.6	3.5 (SE: 0.21) 3.7	3.6 (SE: 0.21) 3.4	t(13)= 0.58 p=0.57	t(13)=0.28 p=0.79

*1 各クイズの満点を10点に標準化し、それを各人ごとに平均した結果をクラス全体で平均した値。

*2 理解認識は4段階のリッカート尺度で回答された値(4:よく理解している, 3:まあ理解している, 2:あまり理解していない, 1:全く理解していない)。6つの項目の回答を合計。学習前後の差のクラス平均。

*3 学習後の理解認識の回答値の各人の平均をクラス全体で平均した値。

*4 自己効力感に関する質問に、7段階のリッカート尺度で回答された値を各人で平均した値のクラス全体での平均。7:大いに自信がある, 6:けっこう自信がある, 5:まあ自信がある, 4:50/50, 3:少し自信がある, 2:ほとんど自信ない, 1:全く自信がない。

*5 上段は平均値(標準偏差), 下段は中央値。

表 3-4 自由記入コメントの Processing 勉強会と Java 授業の比較

自由記入コメント	Processing 勉強会	Java 授業
自分で考えて作れる、工夫して応用できる。	-	6 (13.3%)
達成感があった	-	6 (13.3%)
楽しかった	5 (62.5%)	-
問題を冷静に分析できる。	-	12 (26.7%)
処理の流れを順序立てて考えられる。	-	4 (8.9%)
プログラミングの基礎、言語文法を理解した。	6 (75.0%)	27 (60%)
今後の学習に意欲がわいた。	5 (62.5%)	4 (8.9%)
プログラミングは難しい。	1 (12.5%)	4 (8.9%)
その他	・構造をしっかり把握していないとならない。	・諦めないで取り組むこと。 ・プログラムには何通りもの書き方がある。 ・簡単な動作にも長いコードが要る。

コメント総数：Processing 勉強会 18, Java 授業 66

括弧内の数字は、そのカテゴリのコメントを書いた人の割合

表 3-5 SV ワークショップの流れ

1 週	導入・準備(目的の理解、ソフトウェアのインストール)
2	チュートリアル 1(プログラム動作確認、入力実験)
3	チュートリアル 2(行為とグラフィックスとの関係性をセンサを変えて実験、関数、変数、演算の理解)
4	チュートリアル 3(グラフィックスの大きさ、位置を変化)
5	チュートリアル 4(音に応答する Interactive Mask(インタラクティブに仮面の表情が変化)の制作演習)
6	作品構想(行為と感覚のエクササイズ)
7	作品構想(動きのシナリオを作成)
8	制作
9	グループごとにプログラミングに関してのアドバイスをしながら進める
10	
11	プレゼンテーション

表 3-6 SV 教育プログラム実施前後でのプログラミング理解に対する認識の変化

変化	Web 教材あり 2014 年(N=9)			Web 教材なし 2008+2009 年(N=49)			
	増加 8 人 (89%)	4 以上	5(56%)	39 人 (80%)	4 以上	6(12%)	
変化なし		3 増加	1(11%)		3 増加	9(18%)	
		2 增加	2(22%)		2 増加	11(23%)	
		1 増加	0		1 増加	13(27%)	
		1 人	(11%)		4 人	(8%)	
減少	0				6 人	(12%, -1:5 人, -2:1 人)	

・2014 年はアンケートに回答した 9 名を対象とした。

3.5 考察

3.5.1 実践結果の解釈

本研究で開発した Web 教材を使った Processing 勉強会と Java プログラミング授業を表 3-3 に示した結果から比較する。クイズ成績、プログラミング基礎要素に対する理解認識、自己効力感のクラス平均値とともに、Processing 勉強会は Java 言語による学習に対して同等以上の結果であった。両者の平均値に対するウェルチの *t* 検定では、クイズ平均値は Processing 勉強会の方が高い有意傾向が見られ、クイズで質問しているような基礎項目の理解は、Web 教材を使った Processing 学習でも、Java 授業と同程度以上に得られることを確認した。他の項目は統計的には平均値に有意差を認められないが、どの平均値も Processing 勉強会の方が高い。これは人数が少ないことで、わからない部分をそのままにせずに留まって考え、個々のペースで学習を進めたことも一因である。Java プログラミング授業は、教師が説明し、学生は一部クラス内で、一部は宿題として演習を行う。それに対し、Processing 勉強会は事前に Web サイトを自習してきて、セッション内では、多くの時間を個別の演習に費やす。学習に使っている言語環境ではなく、学習のスタイルの違いが影響していると考えられる。

また、自由記入のコメントを比較する(表 3-4)と、Java 授業では「達成感があった」とのコメント(13.3%)があるが、Processing 勉強会では達成感という言葉は使われず、代わりに「楽しかった」というコメントが現れる(62.5%)。また「問題を冷静に分析できる」「処理の流れを順序だてて考えられる」というコメントは、Java 授業のみに現れる。Processing 勉強会では楽しく主体的に学べており、基礎項目については同等以上の理解に至っているものの、プログラミングの解説や演習問題に費やす時間が Java 授業に比べ短いこともあり、プログラミングの深い理解を得たという認識にまでは踏み込めていないことを示唆する。一方、「インタラクティブな作品作りに使える」といった今後の学習への意欲を表す記述が Processing 勉強会では多い。本教材が、音に反応するモーショングラフィックスという絞られたテーマでの学びを誘導し、その延長線上での展開をイメージしやすいことを反映している、今後プログラミングを自分の問題解決手段のひとつとして主体的に選択できるという点で、有効に働いている。

Web 教材を導入した SV 教育プログラムの実践(2014 年)と以前の実践(2008 年と 2009 年)とを比較した(表 3-6)。プログラミングに対する主観的理認識の学習前後での変化は、増加した学生が 2014 年は 89%，Web 教材なしの以前の実践では 80%，また 3 以上増加したと認識した割合が 2014 年は 67% であるのに対し、以前の実施では 30% で、自信につながる主観的理認識の変化が大きい参加者が多かった。SV 教育プログラムにおいて、自習用 Web 教材を導入することで、プログラミング基礎要素の学習を強化する目的に対して、効果を上げたと判断できる。

実際、2014年の実践では学習の場で、学生が構想した動きを実現するために、プログラムを書く段階において、何度もWeb教材を参照する場面が見られた。例えば、サンプルプログラムを修正しながら意図する動きを探る段階で、Web教材上で元にしたサンプルの解説や動きを確認する、文法の不明点を確認するなど、3つの中核部分が互いにリンクするWeb教材の特徴が有効に活用されていた。また、Web教材を学生同士、講師と学生が共有していることで、説明をする、教えあうプロセスが大変やりやすかった。これは通常のテキスト教材の共有とは異なる。サンプルプログラムがWebページ上で動作するWeb教材であるため、開発作業をしている同じ画面上で、ソースコードと動きを確認しながら、それを発展させるためのアイディアについての話を進められた。加えて、プログラミング基礎事項とサンプルが解説されたコンパクトな内容であることで、制作中、確認したい項目、探している事柄を何度も振り返って確認できる点も、有効であった。

3.5.2 本Web教材と既存教材との比較

既存のProcessing学習のためのWebコンテンツには、Processing自身に含まれる多くのサンプルプログラム、リファレンスに加え、Processingの公式Webサイトが提供するチュートリアルがある[55]。このうち、開発者であるReasとFryによるGetting StartedとShiffmanのビデオ講座は初学者向けコンテンツである。Getting Startedは本教材の導入部分の内容と一致し、またShiffmanのビデオ講座「Hello Processing!」は、形、色、マウスとの応答、条件分岐の項目からなり、本教材のラーニングトレインの1ならびに2と3の一部の内容と同等である。

導入部分について見れば、既存チュートリアルの内容と重なり、それで十分といえる。しかし、その次の段階に進むと、既存チュートリアルはその内容が多岐にわたり、どのように学習を進めていいかわからなくなるという欠点がある。公式Webサイトのチュートリアルでは、座標、色、イメージとピクセル、データ、曲線の描き方や三角関数といった項目ごとに解説するWebページが提供されている。これらは内容ごとに独立しており、必要に応じて学習者が選択して参照するようになっている。そのため、例えば色や配列など自分の知るべき観点が明確な場合は必要な内容を適切に選択ができるが、そうでない場合は難易度がさまざまな項目からどれを学習したらいいか判断しにくい。これに対して、本教材は導入に続く学習の道筋を示し、音に応答するサンプルプログラムの制作へ向けて必要な要素を学べるようになっており、初学者が迷うことがない。

また、Processingの開発グループとは別に公開されているチュートリアルには、Pazos[56]のビデオによる解説コンテンツがある。単純な図形の描画、音声や画像データの処理まで150のエピソードを提供している。視覚的に楽しい画像をプログラムで作り出す観点から必要となる内容をトピックとして取り上げて解説しているが、プログラミング基礎要素を学ぶという点からは内容は整理されていない。加えて、個々のエピソードは独立しておらず、連続したいくつかがシリーズとな

ってひとつの項目を解説している。そのため、利用者が自分の問題に適切なエピソードを探しにくい。それに対し、本教材はインタラクティブなモーショングラフィックスを制作するために必要となる知識と方法に絞り、コンパクトにまとまっているのに加え、リファレンスとして使えるプログラミングの基礎要素や Processing 関数の説明をともに簡潔に示し、忘れたら直ちに参照できるところに特徴がある。

3.6 結論

本章では、SV 教育プログラムでの適用を念頭に、プログラミングの基礎要素を理解しながら、モーショングラフィックスを段階を追って作る自習 Web 教材の開発とそれを使った実践について述べた。これを適用したプログラミング学習が、プログラミング基礎項目の理解と自己効力感の獲得に、Java 言語での入門科目での学習と同等以上に効果を上げることを確認した。基礎要素を理解していれば、学習の場で制作するプログラムの修正や発展も容易であり、別の機会での問題解決にプログラムを有効に適応できる。プログラミングに対する理解認識が高まるとともに、構想段階や作成段階で、サンプルプログラムへの参照が理解や発想を促したという点で、SV 教育プログラムにおいてこの教材を適用することは有効と考える。

さらに、この教材は、SV 教育プログラムとは独立した一般プログラミング教育の場においても活用できる。SV ツールキットのようなセンサーデバイスのない場面においても、マウスやキーボードに加え、音とのインタラクションをもつモーショングラフィックスの制作を題材に、楽しさの中でプログラミング理解を進められる。

第4章 ビジュアル表現の意味を考える教育プログラム

4.1 背景と目的

4.1.1 研究の背景

ビジュアルコミュニケーションデザインを対象とした基礎教育プログラムを、一般教育として、開発するのが本章で述べる研究の目的である。デザインを専門としない学生が、ビジュアルイメージのもつ意味を理解し、使い、制作する能力を高め、デザイン思考力を養成することを目指す。リーフレットやWebページの制作といった具体的直接的なデザイン演習ではなく、形を作る前に意味を考える観点を養う目的で、知覚や記号的な意味について考えるなど抽象度の高い視点のテーマの演習からなる。演習では、クラス内で受講生が制作物を共有し、互いに利用し、評価しあう目的で、Webを利用する。そのためのブログ形式のWebシステムを、コンテンツマネジメントシステム(Web CMS)を使って開発した。

高度なインタラクティブ性をもったアプリケーションの開発やネットワーク技術の発展により、メディアコンテンツの制作は広く一般のユーザにも容易なものとなり、それと同時に、情報をいかに構造化し表現するかという情報デザイン能力が、コンテンツ制作に取り組むあらゆる人に対して、求められる状況が生まれている。メディアコンテンツを制作するのは、デザインを専門とする人だけではない。さまざまな専門性をもつ人がそれぞれの研究や仕事のために、情報をデザインする場面に出会い、情報デザインの観点からの素養が求められる。高等教育の一般情報教育においても、従来からのアプリケーションの利用技術獲得やコンピュータ科学の基礎知識の学習に留まらず、それらを利用した結果作成されるメディアコンテンツに含まれる情報の構造やその表現の質への関心が不可欠となっている。Alway-Rosenstock[57]は1年次デザイン学生と他の学科の学生に対し、グラフィックデザインの授業を行い、デザイン教育がノンデザイン学生のそれぞれの専門の学びにも寄与すると述べている。

ただ、芸術・デザイン分野を専攻していない学生(ノンデザイン学生)はデザインや美的素質に欠けていると自ら思っていることが多く、もともと美術が得意なデザイン学生向けの導入教育をそのまま適用するのは適切ではない。デザイン学生とノンデザイン学生の特性を比較した研究からもその違いが報告されている[58][59]。筆者自身の教育経験においても、ノンデザイン学生の大多数は

アイディアを視覚的方法で表現するのに慣れていないことを観察してきた。ノンデザイン学生の特性に考慮した一般デザイン教育の開発が求められるが、それに関する研究は少ない。

視覚形態を使って効果的に情報を伝えるためのビジュアルデザインのハウツー本は多く出版されている(例えば、[60][61][62])。ノンデザイナーを対象とした本もあり、Williams[63]やSeddonとWaterhouse[64]の著書では、フォントや色、レイアウトを決定するのに役立つ提案を示している。また筆者は今までにノンデザイン学生のためのデザイン教育の教材として、Web ページ制作時に使うワークシートを開発した[65]。Web ページ制作をおこなう授業を導入している大学や高等学校に向けた教材で、デザイナーが行うデザインワークを追体験するように、視覚要素の生成プロセスに合わせた複数のワークシートを用意し、それらを段階的に用いてデザイン制作のサポートをするものである。高校学校専門教科「情報」の科目「マルチメディア表現」の教科書[66]の中に含まれ、授業において利用されている。

しかし、これらのノンデザイナーを対象とした本やワークシートは、形を作るための実用的な知識を学習者に提供する一方、できた形がどういう意味をもつかについての考察には欠けている。色やフォントの選択に関するヒントが示されていても、それを今対象としている情報にどう適用したらいいのか、わからず立ち往生することが多くある。ワークシートや本のヒントを適用して形を作るまでに、視覚的な形がどのような意味を生むかを考える必要がある。本研究では、視覚的表現を観察し、その意味を考えるために教育プログラムを、ノンデザイン学生向けに開発することを目的にする。

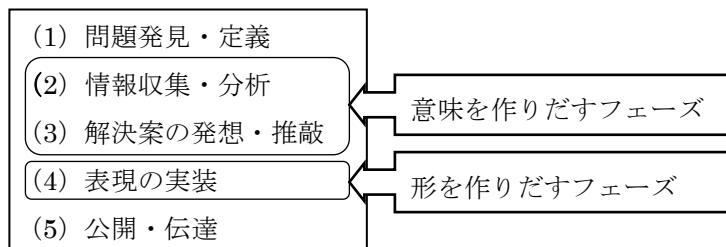


図 4-1 デザインプロセス

4.1.2 教育プログラムの目的

Webサイトやパンフレットなどビジュアル表現のデザインは、一般に図4-1に示すようなプロセスで進む。Luptonは著書[67]の中で、グラフィックデザインのプロセスを問題定義、アイディア出し、形の制作からなると定義している。図4-1の(2)と(3)はLuptonの2番目のフェーズと、(4)は3番目のフェーズと対応している。インタラクションとプログラム機能をもつWebサイトの場合はインターラクティブ機能を検討するプロセスが必要となるが、Webサイトのデザインプロセスも基本的に

は図4-1と同じである[68][69]. さらに図4-1の5つのプロセスはWongとSiuが教育場面でのデザインプロセスとして提案したデザインモデル(situation, research, ideation, developmentとrealizationの5つのステージからなる)とも合致する[70].

図4-1の(4)のプロセスは具体的な形態や色、レイアウトなどの形を作るフェーズであるのに対し、(2)と(3)のプロセスは意味を作るフェーズであると言える. Kolkoはデザインのアイディアを練りあげる統合のプロセスを、*abductive sense-making process*と表現したが、図4-1の「意味を作り出すフェーズ」は、Kolkoが指摘したプロセスと対応する[71]. このフェーズでの学びを対象に、発想と表現についての感性領域の育成をめざし、ノンデザイン学生向けの教育プログラムを構築するのが研究の目的である. ビジュアル表現のもつ意味を考え、それを作り出すことに対しての自信をもつことを期待する. 同様の観点からKellyらは、創造的自信はマインドセット(考え方)であり、ビジュアル表現のデザインを思考する上で必要な素養であると指摘している[72].

Association of College & Research Libraries(ACRL)はビジュアルコンテンツを見つけ、解釈し、評価し、使用し、制作する能力をビジュアルリテラシーと呼び、学部横断的に必要な能力として、その学びの標準化を提案している. 学生は学問的環境の中でイメージや視覚的メディアと関わり、その後の彼らの研究や仕事において、それらを批判的に使用し制作することが期待される. そしてこれらの能力は教えられる必要があると述べている[6]. ただ、中等教育においては多くのビジュアルリテラシー教育に関する論文があるが、高等教育の場面での研究はまだ多くないとの報告がある[73]. 本教育プログラムの目指すところは、ビジュアルリテラシーの学びの項目である、「イメージや視覚メディアの意味を解析する」、「それらを効果的に使う」、「意図的にメディアをデザインする」と一致する. この章では、それを学ぶための、高等教育現場での実施に適した演習を提案する. 加えて、その実践に必要な教材として、成果物をクラス内で共有するためのWebサイト、評価基準を学生と教員が共有するためのループリックを開発する.

本教育プログラムは、大学においてノンデザイン学生への教育に適用されることを意図して構築されたが、高等学校での指導にも活用できる. 平成25年から適用された高等学校学習指導要領の共通教科情報科の科目「社会と情報」での、「情報の表現と伝達」の学びの内容と関連する. また、専門教科情報科で新設された科目「情報デザイン」では、「表現と心理」、「意味の演出」が学びの要素とされているが、本教育プログラムが教育の対象とする内容と合致しており、そこでの適用も考えられる.

4.2 教育プログラムの内容

4.2.1 基本構成

教育プログラムは、「情報〔世界〕を見る・考える演習」と名付けた一連の演習からなる。芸術・デザインを専攻する学生向けには多くのワークショップやプロジェクトが提案されている[74][75][76][77]が、それらで採用されている方法の多くは、中心となるテーマを与えた後、学生にアイディアを考えるように促すというものである。このように学生個々人がすでにもっている制作構想能力に依存する方法は、ノンデザイン学生には有効に働かない。彼らは情報デザインの文脈で創造的な思考をした経験がなく、どう考え始めていいかわからない場合が多く、段階を追った導きが必要である。デッサンは描くことを通して対象を徹底的に観察し、今まで気づかなかつたことに気づくトレーニング方法であるが、まったくデッサンの素養のない人に「さあ、描こう」と言っても、どう手を付けていいか、どこを観察していいか、わからないのと同じである。ビジュアル表現のもつ意味に気づき、それを使うトレーニングを段階的に行えるように演習を構成した。

本教育プログラムでは、最終形として広告ポスターやWebページなどを制作するようなデザインタスクを課すという演習は行わない。演習の目的は、人の視覚情報の認知や記号の意味などを考えながら、ビジュアルイメージのもつ意味を発見するところにある。演習では、視覚要素として学生が自分の回りの世界を観察して撮影した写真を使用し、個人の描画力やデザイン的素養に左右されないように工夫した。演習をする過程で、学生はビジュアルイメージの意味を考察するよう指導される。Jewittは学びの場におけるビジュアルイメージに関する研究のレビューの中で、意味を作るプロセスにおいて使えるリソースを理解するためには、イメージや他のシンボリックな形態が社会的文化的な文脈の特性をどう表現するかを探ることが不可欠であると述べている[78]。

この章では、授業で実践を行った3つの演習について述べる。演習は、知覚、視覚変数、イメージの記号的意味の3つをテーマとしており、演習1から3へと順に求める考察の抽象度が上がるよう設計した。各演習は次のステップを含む。

- (1)注意深く観察し、視覚的イメージの意味を認識する。
- (2)観察し、発見した結果を記録、切り取り、選び取る。
- (3)発見したことを整理、再認識し、考察した結果を言葉で表現する。
- (4)結果をブログ形式のWebシステムを通して他のメンバと共有する。
- (5)クラス内で発表し、互いに評価する。

演習においては、成果を教員だけに見せるのではなく、クラス全体で共有し、互いに評価しあうプロセスが重要である。そのための手段として、ブログ形式のWebサイトをWebコンテンツマネジメントシステム(Web CMS)を使い独自に開発した。4番目のステップでは、それまでのステッ

プの成果物であるイメージとテキストをブログへの投稿の形で、クラス内に公開する。また評価のプロセスでは、学生自身と教師が共有する評価基準が必要である。それをループリックとして開発した。以下の項でブログシステム、ループリック、演習の詳細について述べる。上に記したステップ番号は、演習の説明で現れる番号と対応している。

演習では学習環境として、Web ブラウザとデジタルカメラ(あるいは携帯電話のカメラ機能)だけを使う。画像処理ソフトやグラフィックデザインソフトなどがない一般的な情報教育環境で、提案する教育プログラムを実施できるように設計した。

4.2.2 成果物共有のための Web ブログサイト

本教育プログラムのひとつの特徴は、各演習の成果物をネットワークを使いクラス内で公開する点である。学生は観察したもの、発見したもの、制作したものを Web ページ上に置き、その後クラス内の他のメンバの成果を互いに鑑賞、評価する。加えて、成果の公開 Web ページは自己リフレクションのためのアーカイブにもなる。授業の課題を紙ベースで提出する方法では、成果物を教員一人だけが見る・読むことになるが、ブログ投稿での提出によりピアレビューが行われ、学びのコミュニティが生まれる。伝統的な課題提出スタイルに代わって、ブログを使うことによる影響に関する多くの研究が行われている。Waycott らはブログによる課題の提出を導入している 20 人の講師にインタビューし、学生の成果物の可視性が増加したこと、クラスに協同して参加するという緊張感と挑戦する気持ちが高まると報告している[79]。

Web ブログサイトは、Web CMS のひとつである WordPress を使って開発した。学生が撮影、制作した画像やテキストをブログポストとして投稿することで、クラス内で成果を共有する機能をもつサイトである。WordPress はもっとも広く使用されている CMS であり、プラグインによる機能拡張が容易であるため、開発のプラットフォームとして採用した。Web ブログサイトは、読むべき資料と演習の説明を載せた固定ページと、演習ごとにカスタマイズした投稿ページからなる(図 4-2)。それぞれのページは Web サイト左側のメニューリストからアクセスする。各演習は学生に異なる数の画像と、キャプションやタイトル、説明文などのテキスト情報を提出することを求める。デフォルトの汎用投稿ページでは対応できないため、提出する画像やテキストの要素に合わせてレイアウトを変え、演習ごとに投稿ページをカスタマイズした。

図 4-3 は演習 3 の入力ページで、学生は用意した画像ファイルや説明文などを入力し、投稿する。投稿結果が図 4-2 で、アップロードした画像と説明文が表示される。個々の演習の Web ページは演習ごとに蓄積されて、演習リストからアクセスできるほか、学生ごとに提出した全演習の成果物を見ることもできる。つまり、Web サイトは成果物のクラス内での共有のためのツールとしてだけでなく、学生個人の学びのポートフォリオとしても使える。

管理画面

- 投稿
- メディア
- 固定ページ
- コメント
- 演習1
- 演習2
- 演習3-1**
- 演習3-1 新規投稿
- 演習3-2
- 演習4-1
- 演習4-2
- 演習4-3
- 外観
- プラグイン
- ユーザー
- ツール
- 設定
- CMS Press
- メニューを閉じる

編集 演習3-1 [新規投稿]

1712061ny

バーマリンク: <http://look-think.com/a2013/subject3-1/1712061ny> [編集] [確認 演習3-1]

カスタムフィールドテンプレート

氏名	氏名を入力してください。	初期化
動詞	言葉を選ぶ 例:出る 消える 逃れる 当たる 繋ぐ 外れる 直る 膨らむ 委む 伸びる 流れる 上がる 下がる 帰りる 切る	
演習3-1_画像1	言葉に関連する画像を追加してください。 [選択...] ファイルが選択されていません。	
	<input type="checkbox"/> 削除 	
演習3-1_画像2	言葉に関連する画像を追加してください。 [選択...] ファイルが選択されていません。	
	<input type="checkbox"/> 削除 	
演習3-1_画像3	言葉に関連する画像を追加してください。 [選択...] ファイルが選択されていません。	
	<input type="checkbox"/> 削除 	

公開

[ゴミ箱へ移動](#) [更新]

図 4-3 演習 3 の入力ページ

コンテンツメニュー

お知らせ

演習課題要項
概要 / 操作方法
演習1
演習2
演習3
演習4

演習提出一覧
演習1 一覧
演習2 一覧
演習3-1 一覧
演習3-2 一覧
演習4-1 一覧
演習4-2 一覧
演習4-3 一覧
演習提出一覧

課題提出サンプル
演習1 サンプル
演習2 サンプル
演習3-1 サンプル
演習3-2 サンプル
演習4-1 サンプル
演習4-2 サンプル
演習4-3 サンプル

検索

ログイン

演習3-1提出 学籍番号: 氏名: 2013/07/02 演習3-1

演習3-1 画像イメージと意味【ことば→画像イメージ→言語化】

氏名:

言葉を選ぶ: 流れる

(1) 

(2) 

(3) 

(4) 

(5) 

(6) 

(7) 

(8) 

(9) 

私が選んだ動詞は「流れる」です。写真は「雲」「電流（電線）」「音楽」「時間」「予定（約束）」「水」「景色」「血（血筋）」を表しています。それぞれ全く別のもので、意味合いも違うものを集めましたが、「流れる」という表現で表すことができます。どれも常に変化し、1つにとどまることできないという点において、全ての人に共通することだと思います。

前の投稿へ 演習3-1一覧へ 次の投稿へ

図 4-2 カスタム投稿ページ(演習 3)

表 4-1 自己評価のためのループリック

	すばらしい	よい	標準	まあまあ	まだまだ
目的に沿った探究 課題に示されたテーマを理解し、共感が得られるよう探求、その意図を文章で説明した。	徹底した情報収集や観察などをを行い、複数の視点からの十分な探求をした。	情報収集や観察などを十分に行い、複数の視点からの探求をした	求められた範囲で情報収集や観察などをを行い、複数の視点からの探求をした。	求められた範囲で情報収集や観察などをを行い、それを元にひとつの視点からの考察をした。	最小限の情報収集や観察を行い、ひとつの視点からのアイディアを検討した。
表現・意味の独創性 課題が対象としているテーマに対して、自分の考え、アイディアを示した。	今までにない新しいユニークな自分の視点、アイディアを示した。	自分の視点とアイディアで制作した。	既存のアイディアを参考にしながらも、自分のアイディアを強く示した。	普通にあるアイディアや表現を元に、自分の視点を一部加えて、再生産した。	普通にあるアイディアや表現を真似た。
表現への関心 前向きな姿勢で取り組み、注意を払って、ていねいに仕上げた。	高い完成度をめざして、ていねいに仕上げた。	よくなるように注意を払い、ていねいに仕上げた。	自分の考えを示せる程度にていねいに取り組んだ。	一部分、急いで終わらせたところがある。	最小限の時間と力で、急いで終わらせた。

4.2.3 演習後の自己評価

各演習成果を Web サイトに提出後、学生は演習から何を学んだかについてのアンケート用紙に回答する。アンケートは次の 2 つの部分からなる：(1)演習への興味や学びの実感について問う部分で、演習はおもしろかったか、新しい気づきがあったかについて、5 段階のリッカート尺度を使って回答し、自由記述でその理由を書く、(2)演習過程での自分の振る舞い(パフォーマンス)を評価する部分で、目的に沿った探究をしたか、独創性があるか、表現に注意を払い取り組んだかの 3 つの項目に対して、5 段階の評価基準を示したループリック(表 4-1)上でいずれかを選択する。

ループリックは伝統的な筆記試験ではその達成度を判定できない創造的な学習活動を評価するために使われるツールである。学習活動で発現される態度、思考、判断や習得されるスキルの質に関する評価基準を定義したもので、「不十分」から「優秀」の間の 3 段階から 5 段階程度の尺度に対して、文章で記述する。表 4-1 のループリックは学生の自己評価と教員による評価の両方に使われる。

情報デザインの学びにおいても、ループリックに関する研究が多くなされている。Knight は学部における情報リテラシー学習の達成度をループリックにより評価した結果を報告している[80]。Clary らは、景観デザインのクラスにおいて、創作的な成果物の評価ツールとして、ループリックが有効だと述べている[81]。

表 4-1 の評価項目は、Creative Product Analysis Matrix (CPAM) モデルの 3 要素と対応して

いる。CPAM はプロダクトの創造性を評価するモデルとして提案されたもので、問題解決(resolution), 斬新さ(novelty), 入念さと統合(elaboration and synthesis)の 3 つの要素で評価を行う。CPAM モデルはその有効性が検証される[82]とともに、さまざまな研究において、例えば椅子などの有形プロダクト[83]をはじめ、グラフィックデザインやコンピュータグラフィックスの制作物の創造性を評価するツールとして使われている[84][85][86][87]。

問題解決(resolution)要素はプロダクトに価値があり、使用に適しており、わかりやすいかを評価し、斬新さ(novelty)要素は対象の独創性や人に与える驚きに着目し、入念さと統合(elaboration and synthesis)はプロダクトがエレガントで、うまくできているか(well-craftedness)といった出来栄えに関する側面を評価する。製品の評価のみならず、このモデルを教育現場において学習者の成果物の創造性を評価するスケールとして使うことが提案されている[88]。

学生のパフォーマンスを評価する基本的項目として、ループリックにこの 3 つの要素を採用した。ループリックの 3 つの項目、目的に沿った探究、表現・意味の独創性、表現への関心は、それぞれ CPAM モデルの問題解決(resolution), 斬新さ(novelty), 入念さと統合(elaboration and synthesis)に対応している。それぞれの項目に対して、すばらしい、よい、標準、まあまあ、まだまだの 5 段階の尺度に対して、そのパフォーマンスの質を記述した。

学生は、Web サイトの投稿ページを参照しながら、ループリックの記述に従い、自身の演習のプロセスを振り返り、評価する。ループリックを含むアンケート用紙に記入することで、学生が自分のパフォーマンスをリフレクションするのを助け、自身の観察や制作への認識を高める。

4.2.4 演習 1：視覚による知覚の理解

演習 1 の目的是、人間の視覚による認知の傾向を理解することと、それを通じて意識的な観察を行うことである。具体的には、ゲシュタルト要因をテーマとしたビジュアルイメージ採集を行う。ドイツの実験心理学者であるヴィルヘルム・ヴント(Wilhelm M. Wundt)を中心に、20 世紀初めに提起されたゲシュタルト心理学 (Gestalt Psychologie) は「人間の精神は部分や要素の集合ではなく、全体性や構造こそ重要視されるべき」とし、人の知覚の普遍的な特性を研究した。その学派の名は、全体性をもったまとまりある構造を示すドイツ語の「ゲシュタルト (Gestalt : 形態)」にちなんでいる。ゲシュタルト心理学では、人間の知覚において多数の刺激が存在する時、それらは個々に知覚されるのではなく、より大きな範囲のいくつかの群として知覚される傾向があり、そのまとまりを決定するものを知覚体制化の法則として「ゲシュタルト要因 (Gestalt Factor)」と呼んだ[89]。ゲシュタルト要因は視覚デザインの分野でも応用され、例えば、Lidwell らはその著書で、より訴求力を高め、よりよい視覚デザインの決定をするためのデザインの原則の一部としてゲシュタルト要因を紹介している[61]。この演習では、6 つのゲシュタルト要因(図と地の関係、近接、類同、閉

合、滑らかな連続、同一の結合)に着目して、人がまわりの視覚世界を認知する傾向を理解する。視覚による知覚の理解は効果的なビジュアルコミュニケーションデザインへつながる。演習は次のステップで行う。

- (1) 学生は日常生活で出会う「もの」、あるいはシーンを観察し、6つのゲシュタルト要因の実例となるビジュアルイメージを写真に撮る。つまり、ゲシュタルト要因が現れるイメージの採集を行う。
- (2) 撮影された写真の中で、6つの要因それぞれが現れる一番適切な写真を選択する。
- (3) 選択された写真の中に現れるイメージはどう知覚され、それはどのゲシュタルト因子と関係しているかについての説明文を記述する。
- (4) 写真と説明文をWebサイトに投稿する(図4-4は学生の投稿ページの例)。
- (5) クラスセッションにおいて、画像のコレクションを発表し、ループリックで自己評価する。

情報【世界】を見る・考える演習

コンテンツメニュー

お知らせ

演習課題要項

概要 / 操作方法

演習1

演習2

演習3

演習4

演習提出一覧

演習1 一覧

演習2 一覧

演習3-1 一覧

演習3-2 一覧

演習4-1 一覧

演習4-2 一覧

演習4-3 一覧

演習提出一覧

課題提出サンプル

演習1 サンプル

演習2 サンプル

演習3-1 サンプル

演習3-2 サンプル

演習4-1 サンプル

演習4-2 サンプル

演習4-3 サンプル

ログアウト

検索

演習1提出 学籍番号： 氏名：

2013/05/17 演習1

演習1 ビジュアルイメージの採集

氏名：

(1) 図と地の関係

背景の赤の部分を「地」、人型の部分を「図」として図と地の関係が成り立っていると感じました。



(2) 近接

並んでいる洗濯バサミのうち、互いが近い位置にある真ん中の二つは、離れているものよりもより関係が近しいように感じたので、近接の要因が成り立っていると思います。



(3) 類同

形や色が似ているボタンは、そうでないボタンに比べひとたまりに認識されやすく関係性が深いようになるので、類同の要因が成り立つと思いました。



図 4-4 演習 1 の投稿ページの例

4.2.5 演習 2：ビジュアル変数

演習 2 は画像イメージがどのように量、質、区別を表現するかに関する意識を高める演習である。Bertin[90]は、位置、大きさ、形、濃淡、色、方法、肌理の 7 つをビジュアル変数として提案し、それらが地図や図表のようなグラフィカルな情報の中で、違いを表現するのにどう働くかを研究した。Mijksenaar は Bertin の原則に基づき、その変形として実用的ビジュアル変数を提案し、どのように使えるかを考察している[91]。重要さの違いを示す階層的(hierarchical)変数と、タイプの違いを示す特徴づけ(distinguishing)変数である。

演習 2 では Mijksenaar の試みのように、Bertin のビジュアル変数の考え方をベースに量、質などの差異を表すビジュアル変数のスケールを、自分の撮影した写真を使って作る。スケールの中で写真はビジュアル変数の一つの値として機能するが、それを決める過程でビジュアルイメージのもう意味を考えるのが目的である。ビジュアルスケールは 10 枚の写真からなり、1 枚の写真が変数の一つの値を表す。写真はスケールの中で段階的なビジュアル変数の値の変化を示すように並べられる。図 4-5 は学生の投稿ページの例で、紫陽花の色をビジュアル変数とし、その変化で量を表すスケールとしたものである。演習 1 は観察が演習の中心であったが、演習 2 では観察の上にビジュアルイメージの意識的な使用に焦点をあてる。

演習は次のステップで行う。

- (1) 学生は制作するスケールが表す値の性質、その範囲を決定する。そして、そのスケールの特性を表すのに適切な変数要素を Bertin の 7 つビジュアル変数から選ぶ。
- (2) 選んだビジュアル変数における段階的な値の変化を表す写真を撮影する。
- (3) スケールを表すのに適した 10 枚の写真を選ぶ。そのスケールについての説明文を書く。
- (4) 写真と説明文を Web サイトに投稿する(図 4-5)。
- (5) クラスセッションにおいて、スケールを発表し、ループリックで自己評価、また相互評価する。

情報【世界】を見る・考える演習

演習2提出 学籍番号： 氏名：

日 2013/06/20 白 演習2

演習2 情報を表すビジュアル変数

氏名：



土によって色が変化する紫陽花の色の変化で変数を表す。

図 4-5 演習 2 の投稿ページの例

4.2.6 演習 3：記号論的意味

ビジュアルイメージの記号論的意味について考えることを目的とした演習である。 哲学者 Barthes は写真のメッセージについて、「写真は現実を類似的に再現する。写真に写された事物や

風景などは、対象の類似的な内容そのもので、対象が指示する明示的な意味（denotation：外示）である。これを外示されたメッセージと呼ぶ。写真はそれと一緒に、写真を見た人が読み解く潜在的な意味（connotation：共示）をもつ。これを共示されたメッセージと呼ぶ」と述べている[92]。記号は明示的意味において記号がとる形であるシニフィアン（signifiant）と、それが表現する概念であるシニフィエ（signifie）からなり、これが一体となって外示的意味を伝達する。バルトはこれを「形式」と呼んだ。共示的な意味体系においては、この形式（第一次の意味作用）が新たにシニフィアンになって、それが表す概念であるシニフィエとさらに一体となって第二の意味を伝達する。これが共示的な意味である[93]。つまり、明示的意味が共示的意味の連鎖を生む（図4-6）。このメカニズムによれば、記号の意味を保持しながら、記号の形を変えることで、異なる共示的意味を生成することができる（Chandler[94]）。つまり、内容が同じでもスタイルやトーンを変えることで異なる共示的意味を生む。このような記号論の知見はグラフィックデザインにおいても、デザイナーがコミュニケーションの視覚的意味を検討する場合の有効な手段である[95]。



図4-6 意味作用の連鎖（Barthes[93]より一部変更して引用）

演習3は、写真的文化的、社会的文脈における共示的意味を考察するように構成され、2つのプロセスからなる。まず、あることばの明示的意味を写真に撮影する。風景、できごと、ものなどが写真の中に切り取られ、そのことばが表わす意味のビジュアルイメージが収集される。「出る」を例にすると、「太陽が昇る」「月が出る」「顔が出る」「突き出た窓」「巣立った後の巣」などが写真に撮られるだろう。それらの写真は「出る」ということばの直接的な多様な意味内容を表すことになる。クラス全体では、ことばと写真の集合のセットがいくつもできる。これが明示的意味を考える第一のプロセスである。出発点となる動詞として、多数の意味をもつ動詞のリストをあらかじめ用意した（出る、消える、逸れる、当たる、繋ぐ、外れる、壊れる、直る、膨らむ、萎む、伸びる、流れる、上がる、下がる、降りる、通る）。第一のプロセスは、次のように進める。

- (1) 学生は動詞のリストからひとつの動詞を選択する。

- (2) 選んだ動詞の明示的意味を表す写真を撮る。
- (3) 撮影した中から 9 枚の写真を選び、写真が表す明示的意味とそのビジュアルイメージについて関して説明を記述する。
- (4) 写真と文章を Web サイトに投稿する(図 4-7 「流れる」を選んだ学生の投稿ページの例)

情報【世界】を見る・考える演習

演習3-1提出 学籍番号： 氏名：
 2013/08/06 演習3-1

演習3-1 画像イメージと意味 [ことば→画像イメージ→言語化]

言葉を選ぶ：流れる

今回私が選んだ動詞は「流れる」です。撮ってきた写真は「ライブハウス」「朝日」「インスタレーション作品」「海」「恋人達」「街中」「紅葉」「工場の夜景」「空」です。これらは私の中で大切な思い出であり過去の時間軸を流れていってもう一度訪ねたい場所の写真達です。撮った瞬間は今であり現在なのにその一瞬でそれらの出来事は過去となり過ぎ去った（流れてしまった）過去の時間となってしまいます。時間という流れは綺麗であり悲しいものです。



(1)



(2)



(3)



(4)



(5)



(6)



(7)



(8)



(9)

**図 4-7 演習 3 の第一プロセスの投稿ページの例：
動詞「流れる」を明示的に表す写真のセット**

情報【世界】を見る・考える演習

演習3-2提出 学籍番号： 氏名：

2013/08/06 □ 演習3-2

演習3-2 画像イメージと意味【ことば→画像イメージ→言語化】

私は「不安感や憂鬱さを感じる一瞬」をテーマとして写真を選びました。一枚目の花の枯れた様子は季節の移り変わり時のなんともいえない不安定な気持ちや憂鬱さが現れていて、二枚目の写りこみの写真も雨の日に感じるあの憂鬱で気だるげな気持ちになってしまう瞬間を、そして飛行機の写真は別の地へと旅立つ時の期待感よりも勝る不安感を、最後に自分の写真はすごく印象的であったボルタンスキー氏のインスタレーション作品の写真を選びました。この作品に出会った時のなんとも言えない不安感は忘れられません。私はこの言葉にできない不安定な気持ちがすごく好きです。

(1) 

(2) 

(3) 

(4) 

図 4-8 演習 3 の第二プロセスの投稿ページの例：
「不安感や憂鬱さを感じる一瞬」というテーマの組写真

次に第二のプロセスで、複数のセットの写真の中から、共通、対比、連続、因果的要因などの観点から新しい関係性を生み出す4枚の写真を選ぶ。元のことばとの関連を断ち切った4枚で組写真を作ることで、写真がもつ共示的意味を発見する。組写真にタイトルをつけ、テーマを解説する文章を書くことで、組写真が表現する内容を言語化する。

第二のプロセスは次のように行う。

- (1) クラスメンバーによって投稿された写真のセットと説明文を見る。
- (2) すべての投稿の中から 4 枚の写真選び、独自のテーマを表現する組写真を構成する。4 枚の写真のうち一枚は自分の写真の中から選ぶ。4 枚の写真は個々のテーマの下で新しい共示的意味を生む。
- (3) 組写真にタイトルをつけ、そこに込められた意味について記述する。
- (4) 写真、タイトルと記述文を Web サイトに投稿する(図 4-8)。
- (5) クラスセッションにおいて、組写真を発表し、ループリックで自己評価、また相互評価する。

4.3 教育実践とその考察

4.3.1 教育実践

4.2 節で述べた 3 つの演習は、演習 1 から 3 へと段階を踏んで抽象度の高い考察が行えるように設計した。ビジュアル表現に対する学生の気づきや自信の観点から学生の学びの進展を考察するため、3 つの演習を順次実施した。ビジュアルコミュニケーション科目の受講生 71 名(2, 3, 4 年次の情報メディア学科生)を対象に、週 1 回のクラスセッションを 10 週に渡り行った。演習を通して学生がどう変化したかについて知るため、各演習後学生自身のアンケート用紙による振り返りと教員の評価を行った。4.2.3 項で述べたアンケート用紙を使って学生は自身の演習結果を振り返る。同時に 3 名の教員が学生の成果物を、ループリックにより評価する。3 名のうち 1 名はクラスセッションを担当した教員で、他の 2 名はビジュアルデザインを専門とする大学講師である。3 つの演習の成果の提出とアンケートへの回答がすべて揃っている 47 名の学生を対象に考察を行う。

4.3.2 ループリックによる成果の評価

各演習成果に対するループリックによる学生自身と教師の評価の平均を表 4-2 示した。学生自身による評価は演習が進むにつれてわずかに上がっているが、教員の評価には違いはない。教員の評価結果は成果物の質に演習間で差はないことを示しているが、学生は演習が進行するにつれて自分の成果を高く評価する傾向にある。これは進行に従い自信が増加していることを示唆する。この示唆は、教員と学生自身の評価を詳しく比較することで、さらにはっきりとする。

表 4-2 演習成果物に対する評価

	学生			教員		
	目的に沿った探求	独創性	表現への関心	目的に沿った探求	独創性	表現への関心
演習 1	3.28 (0.71)	3.43 (0.77)	3.49 (0.80)	2.90 (0.53)	2.92 (0.39)	2.75 (0.53)
演習 2	3.40 (0.77)	3.51 (0.72)	3.62 (0.80)	3.01 (0.49)	2.82 (0.47)	2.77 (0.47)
演習 3	3.64 (0.74)	3.66 (0.70)	3.68 (0.70)	2.94 (0.43)	2.80 (0.47)	2.65 (0.60)

* N=47, 括弧の中の値は SD.

* 5段階のリッカート尺度で評価: 5 = すばらしい; 4 = よい; 3 = 標準; 2 = まあまあ; 1 = まだまだ.

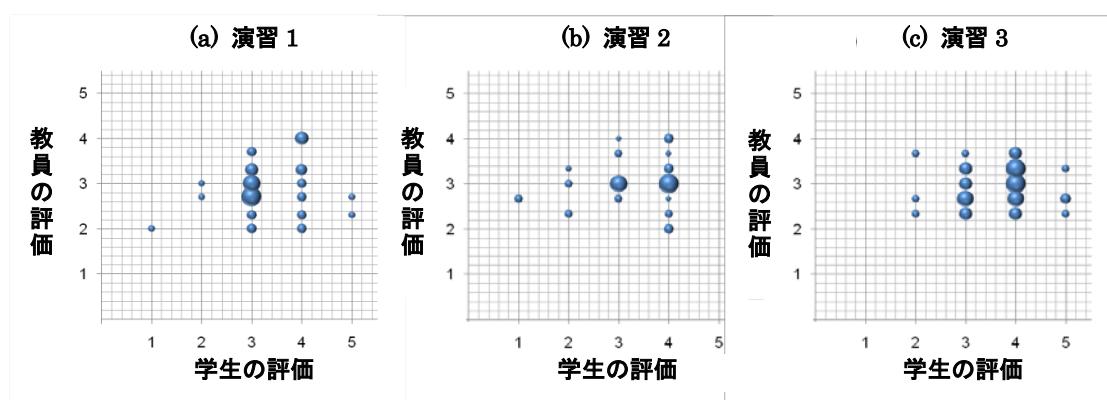


図 4-9 目的に沿った探求に対する学生と教員の評価の比較

個々の成果物に対して、学生と教員での評価の違いを詳しく見るために、ループリックの「目的に沿った探求」項目に関する学生自身と教員の評価をプロットしたのが図 4-9 である。プロットの x 座標値は学生の評価を、y 座標値は教員の評価を、大きさはその評価を得た学生の人数を表す。例えば、図 4-9(c)の座標(4,3)のプロットは、演習 3 の成果の「目的に沿った探求」項目に対する評価で、学生自身の評価が 4 で、教員の評価が 3 であった学生が 7 名いたことを意味する。学生自身の評価は明らかに教員の評価より高い傾向にある。教員より高い自己評価をする学生の数は、演習 1 で 26 名(55.3%)、演習 2 で 25 名(53.2%)、演習 3 で 37 名(78.7%)と演習 3 で増えている。同じ傾向はループリックの他の項目(独創性と表現への関心)でも見られた。

演習が進行するについての学生自身の評価の変化を見るために、評価尺度の頻度を表 4-3 に示す。「目標に沿った探求」では尺度の 4 あるいは 5 と評価した学生の数は、演習 1 で 15 名(32.0%)、演習 3 で 29 名(61.7%)と増加し、また「独創性」の項目では、演習 1 で 24 名(51.1%)、演習 3 で 30 名(63.8%)と増加した。これに対し、「表現への関心」では 4 あるいは 5 と評価した学生の数に

変化は見られなかった。入念に仕上げるという点に対しては、演習は影響を与えることができなかつたが、観察や探求を深め、アイディアを考え出す点に対しては学生を刺激し、意識を高めることができたと考える。ここで提案した演習は4.1節で述べたように、形ではなく、意味を作り出すプロセスに焦点を当てて構築されているので、探求や独創性より、表現に关心をもって入念に仕上げることに対して低い意識しか得られなかつたことは当然ともいえる。

表4-3 学生自身の評価尺度の頻度

評価の尺度	目的に沿った探求					独創性					表現への関心				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
演習1	0	2	29	13	2	0	6	17	22	2	1	4	15	25	2
演習2	1	5	15	26	0	0	4	17	24	2	0	4	15	23	5
演習3	0	3	15	25	4	1	0	16	27	3	0	1	18	23	5

N = 47

5段階のリッカート尺度で評価: 5 = すばらしい; 4 = よい; 3 = 標準; 2 = まあまあ; 1 = まだまだ

4.3.3 学びの実感についての振り返り

4.2.3項で述べたアンケート用紙で、演習に対する興味や学びの実感を問うた第一の部分に対する学生の回答を考察する。図4-10に演習ごとに、5段階のリッカート尺度の回答の人数を示す。明らかに、新しい気づき、おもしろさとともに肯定的な回答が演習2で減っている。新しい気づきに関する質問に4、あるいは5と回答した学生は、演習1で41(87.3%)であったのが、演習2で27(58.7%)と減少し、再び演習3で36(76.6%)と増加した。同じ傾向はおもしろさを尋ねる質問への回答でも見られ、演習1で34(72.4%)、演習2で20(42.6%)、演習3で33(70.2%)であった。演習2の目的は、Bertinの図の記号学に基づき、グラフィックイメージが量や質、違いをどう表せるかに関しての気づきを促すことである。演習1に比べると、演習において深い考察が必要であり、理解を得るのに多少困難さがある。このことが演習2での肯定的回答が減少した理由だと考える。ただ、教員の演習2の成果に対する評価は演習1や演習3と同等であることからも、演習2が要求するレベルが実際に高すぎるということはないと判断できる。

演習3はBarthesの写真がもつメッセージに関する論考をもとに、ビジュアルイメージの記号的意味についての認識を高めるのが目的である。演習1と2が図4-1のプロセス(2)の情報の収集と分析についての演習であるのに対して、演習3は、プロセス(3)のアイディアの発想と発展に焦点を当てる。そのため、演習3の要求は前の2つに比べて高い。それにもかかわらず、図4-10が示す回答では、演習3では肯定的回答(4ないし5に相当する回答)が増加している。さらに、1な

いし 2 に相当する回答は演習 3 ではなかった。イメージのもつ意味について考察することに慣れ、3 つの演習を実施する中で、そのことを楽しむことができるようになったことを示唆する。

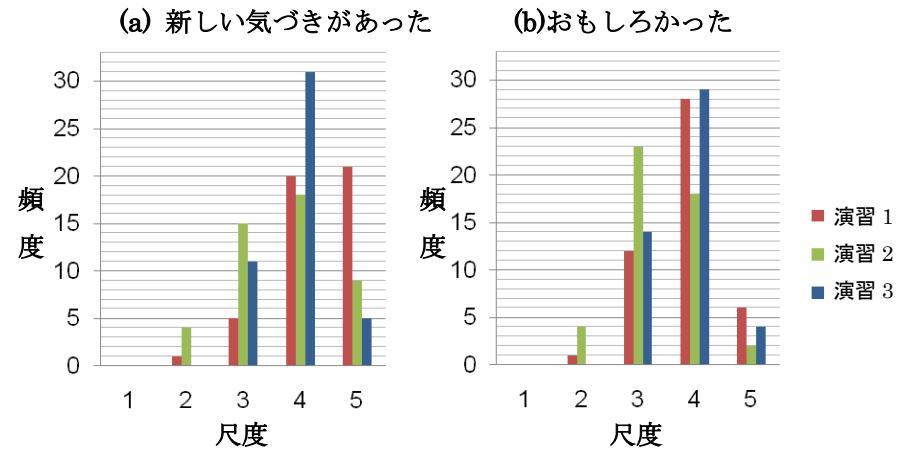


図 4-10 興味と学びの実感に対するアンケートの回答
大きいにそう思う(5)から全くそう思わない(1)の 5 段階のリッカート尺度で回答

表 4-4 自由記述コメントとそのカテゴリ

カテゴリ	説明	実際のコメントの例
1 直接的記述	感じたこと、見たことをそのまま記述したコメント	<ul style="list-style-type: none"> 適切な写真を選択するのがむずかしかった。 新しいことを考えるのは楽しかった。 自分が撮影した写真が投稿ページに載ったのを見て、うれしかった。
2 他のメンバからの刺激	他のメンバの投稿を見て、新しいことを発見した、視点が広がった	<ul style="list-style-type: none"> 他の人のアイディアはとても興味深く、そこに新しい発見があった。 私が今まで考えもしなかったアイディアを見られたのがよかったです。 他の人が感じることは、自分とは違うのだと気づいた。
3 理解や気付き	演習の目的に沿って、理解が進み、気付きが得られた	<ul style="list-style-type: none"> 私が普通考えもしなかったところに、デザインの意図が隠れていたことを発見した。 写真や言葉にはいろいろな意味があるのだと思った。 一つのイメージから複数の意味を生成できることを理解した。
4 概念的思考	多様な観点を通して、抽象的な考察ができた	<ul style="list-style-type: none"> 再構成して新しく意味づけするという発想に新しい気づきが生まれた。 全く意味の違う写真が集まって新たなテーマに変わっていく過程がおもしろかった。
5 意図しない発見	直接演習意図とは関係のない新しい気づき、発見があった	<ul style="list-style-type: none"> 写真を撮る時は、対象の背景にも注意を払わないとならないことに気づいた。

表 4-5 カテゴリごとのコメントの数

	カテゴリ	演習 1 (N=94)	演習 2 (N=89)	演習 3 (N=91)
1	直接的記述	41 (43.6%)	30 (33.7%)	20 (22.0%)
2	他のメンバからの刺激	6 (6.4%)	13 (14.6%)	23 (25.3%)
3	理解や気付き	39 (41.5%)	36 (40.4%)	36 (39.5%)
4	概念的思考	8 (8.5%)	4 (4.5%)	10 (11.0%)
5	意図しない発見	0	6 (6.8%)	2 (2.2%)

この 2 つの質問へのリッカート尺度を使っての回答について、そう判断した理由を自由に記述してもらった。47 名の学生のコメントから 274 の文章を切り出し、質的分析を行った。小規模な質的データの分析手法である SCAT(Steps for coding and theorization)[28]のオープンコーディングの手法を援用し、5 つのカテゴリを抽出した。表 4-4 にそのカテゴリとコメントの例を、表 4-5 にカテゴリごとのコメント数を示す。

表 4-5 を見ると、カテゴリ 3, 4, 5 に分類されたコメントの数は、演習によって大きな違いはない。それに対してカテゴリ 2 のコメントの数は、演習が進むにつれて増加している。カテゴリ 2 に属する演習 3 でのコメントに「他の人の組写真を見て、集めた写真から新しい意味が生まれるプロセスを味わった」、「人によってさまざまな見方があるのだということを実感した」、「いろいろな人の視点を疑似体験しているような気分になった。自分が見落としていた『よくあるモノ』の写真に新しい発見ができた」といったものがある。これらのコメントは、他のメンバの成果の発表を見ることで、刺激を受け、気づきを得るという演習のねらいを、学生たちが演習を通して達成していることを示している。

カテゴリ 1 のコメントは、何が起こって、何を感じたかの記録的内容にとどまり、考察や推論を含まない記述である。このような浅いリフレクションと言えるコメントの数が、演習を進めるに従って減っていくことは、演習の流れが学生に意味に関する考察を促し、その結果一歩進んだリフレクションができたと考える。

4.4 結論

この章で述べてきた研究は、デザインを専門としない学生を対象に、ビジュアルイメージの意味を考え、新しい意味を作り出すための教育プログラムを提案することである。教育プログラムは

3つの演習からなり、ビジュアルイメージの意味に関する理解や気づきを促すことを目的に設計されている。この章では、学習の意図、内容を述べ、教育実践において得られた観察データの考察を行ってきた。

演習1と2は図4-1のプロセス(2)の情報の収集と分析に関するトレーニングを行う演習であり、意識的な観察を通して、ビジュアルイメージに対する知覚認識について理解し、それらを使うことに焦点を当てている。演習3は、ビジュアルイメージのもつ明示的意味と共示的意味を考えるもので、図4-1のプロセス(3)のアイディアの発想と推敲に関係したトレーニングである。

ループリックを含んだアンケートへの学生の回答と同じループリックを使った教員の成果物評価の分析から、演習の連続的な実施により、演習が進むにつれて、学生の自信が増加したことがわかった。また、開発したWebサイトを通して他のメンバの成果を見ることで、学生は刺激を受け、新しい気づきを得ることができたことが、学生の自由記述コメントの分析から示された。Webサイトによる成果物の共有が異なる視点や新しい気づきを生み、有効に機能したと考えている。また、浅いリフレクションを示すカテゴリのコメント数の減少は、連続した演習の実施が学生を抽象度の高い考察へ導いたことを示唆する。

加えて、学びの素材として写真を使う方法は、学生の描画力や美術的スキルとは無関係に、全ての学生が同じ質の素材から出発できるということから、ノンデザイン学生を対象とした教育プログラムにおいて効果的である。

この研究は、評価のための手段であるループリック、成果共有のためのWebシステムという教材と、3つの演習を含む教育プログラムをセットとして提案するところに意義があると考えている。ノンデザイン学生向けの教育内容として、別の教育現場で実施できるように、教材の公開、内容の充実を図りたい。基本的な構成のパソコンと予めインストールされているWebブラウザ、デジタルカメラ(あるいはカメラ機能付き携帯電話)という一般的コンピュータ演習室で実施が可能である点では、受け入やすいと考える。ただ、成果物共有のためのWebシステムを機能させるにはWebサーバへのインストールが必要であり、それをさらに容易にするため、Web CMSのテンプレートを作成するなど工夫が求められる。

第5章 まとめ

18世紀半ばから19世紀にかけての産業革命によって引き起こされた工業化社会において、その基盤である蒸気機関の知識や技術は専門家の領域であった。日常生活がそれにより生産されたものや道具で支えられていても、その知識や技術は一般の人からは離れた存在であった。20世紀半ばから情報技術が発展し、情報化社会が進展したが、その当初はコンピュータやソフトウェアといった基盤技術は、やはり限られた専門家のものであった。21世紀に入りその状況は変わる。インターネットの浸透、マルチメディア処理技術やパーソナルコンピュータ、携帯端末などの発展は、社会のあらゆる分野の仕事や研究と関係し、日常生活における情報環境も大きく変化した。Manovichはその著書[34]で、手紙を書くときもビデオ映像を見る時もソフトウェアを使っている状況を差し、「ソフトウェアは世界を動かすユニバーサルなエンジンだ」と述べている。自動車に乗せてもらうだけでなく、エンジンの仕組みを知った上で自ら運転できれば、行動範囲が広がる。これと同様、21世紀の情報化社会では、「情報処理」の専門家ではない人たちが、気軽に使えるソフトウェアを用いて、情報を表現したり、データを処理したり、グラフィックスを加工したり制作したりしている。その技術や知識は限られた専門家にだけ属するものではなく、誰もが理解やアクセスが可能であり、使う側と作る側の垣根が低くなっている。このような状況の中で、利用者としてだけではなく、制作者としての視点をもって行動できることは、仕事や学びにとって大きなメリットが生まれる。本論文では、グラフィックイメージによる意味伝達、情報と身体とのインタラクティブな関係に焦点を当て、メディアコンテンツの制作者としての感性と基礎的スキルをもつことを目的に、教育プログラムの開発を行った。画家は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。これと同様にメディア・テクノロジーを使った表現や開発の領域で、感性的次元の気づき、制作力獲得のための基礎的トレーニングとしての教育プログラムである。

第2章では、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インストレーションの制作を行う教育プログラム(SV[Sensory Vision]教育プログラム)について述べた。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スと人の行為とがインタラクションをとることで鑑賞するインストレーションである。メディア・テクノロジーがもたらすインタラクティブ性の探求と、プログラミング理解との2つの目的をもつ。I/Oモジュールやセンサ類を含むツールキット、展示環境などのハードウェアを開発し、教育実践に適用した。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の学習が必要である。模倣する→再生産する→創作するという段階を通してプログラミングの理解を進める目的で、その出発点となるサンプルライブラリを用意した。第2章ではそれらを紹介するとともに、作品におけるサンプルプログラムの流用、学生のプログラミング理解の認識、プログラミング学習に対する意欲について考察した。

サンプルライブラリは、シンプルなグラフィックスの運動によるインタラクションが人の感覚にどう捉えられるかの発想の基本として、また、ソースコードを流用することでプログラムの土台として、有効に機能した。結果として8割の学生にプログラミングの理解の向上をもたらした。一方、1割程度の学生はプログラミングに否定的な徒労感を覚えることが観察された。プログラミング学習意欲を持続するためにそれを助ける学びのツールが必要であると考えられた。

そこで、プログラミング理解を促進する目的で、主に自習に使用するWeb教材を開発した。インターラクティブなビジュアル表現を作り出すプログラム制作を学ぶことを軸にした教材で、これについて第3章で述べた。教材は、SV教育プログラムに使用するのに加え、それとは独立した一般プログラミング学習の場においても活用できる。この教材を使い、教材を使ったプログラミング勉強会と、教材を適用したSV教育プログラムの、2つの教育実践を行った。前者の実践をJava入門プログラミング授業と比較した結果、本教材を使ったプログラミング学習が、基礎項目の理解と自己効力感の獲得においてJava言語での入門科目的学習と同等以上の結果であったことを確認した。また、後者と教材を使わないSV教育プログラムの実践と比較した結果、教材を使うことで、プログラミング理解に対する主観的理解度が高まったという結果が得られた。

第4章では、ビジュアル表現の意味を考える教育プログラムについて述べた。デザインを専門としない学生が、ビジュアルイメージのもつ意味を理解し、使い、制作する能力を高めることを目指す。この目的は、ビジュアルリテラシーの学びに含まれる「イメージや視覚メディアの意味を解析する」、「それらを効果的に使う」、「意図的にメディアをデザインする」と重なる。知覚や記号的な意味について考える3つの演習を提案した。演習では素材として写真を使い、学生の描画力や美術的スキルとは無関係に、全ての学生が同じ質の素材から出発できるように工夫した。演習成果をクラス内で共有し、互いに利用し、評価しあう目的で、Webを利用する。そのためのブログ形式のWebシステムを、コンテンツマネジメントシステム(Web CMS)を使って開発した。また、演習のプロセスや成果を振り返って評価するためのループリックを用意した。創造的な学習活動を評価するために使われるループリックには、演習過程で求められる姿勢、思考、判断や習得されるスキルの質に関する基準を定義する。提案したループリックは、「目的に沿った探究をしたか」、「独創性があるか」、「表現に注意を払い取り組んだか」の3つの項目に対して、5段階の評価基準を記述した。

Webシステム、ループリックを使い、教育実践を行った結果を述べた。ループリックによる学

生の自己評価と教員の評価の分析から、演習の連続的な実施により、演習が進むにつれて、学生の自信が増加したとわかる。また、開発した Web サイトを通して他のメンバの成果を見ることで、学生は刺激を受け、新しい気づきを得ることができたことが、学生の自由記述コメントの質的分析から示された。Web サイトによる成果物の共有が異なる視点や新しい気づきを生み、有効に機能したと考えられた。また、浅いリフレクションを示すカテゴリのコメント数の減少は、連続した演習の実施により、学生が抽象度の高い考察を行うようになったことを意味する。

情報化社会において、メディア・テクノロジーは人間の創造性や知性を拡張する身近な道具である。本論文では、その技術がもつ特性であるビジュアル表現やインタラクションの意味を考える基礎教育プログラムを提案し、その実践に使う教材を開発した。それらが他の教育現場でも適用されるように、教材の公開、教育プログラム内容の充実を図っていきたい。

プログラミングはメディア・テクノロジーを使った表現にとって重要な要素であって、プログラミングを理解することで、やりたいこと、表現の幅が広がる。ゲームソフト「パラッパラッパー」や「ビブリボン」の制作者である季理は、グラフィックスを動かしたいと考えて大阪大学工学部電子工学科の研究室でプログラムを学び始めた頃の経験を、講演の中でこう語っている[96]。「『プログラミング言語C』という本を渡され、1 個めの例題で Hello World とモニタに表示させました…(中略)… 7 つ目ぐらいの例題に、摂氏温度を華氏に変換しろと書いてあったんですね、実は私はそこで切れました。これは私に必要なことではないと思いました。」

Hello World から始めるプログラミングの学びは出発点としてごく普通のことであり、摂氏華氏変換計算はプログラミングの重要な基本要素を理解するための好例と言える。しかし、プログラミングを学ぶのに、必ずしも従来の手順や方法で進める必要はない。特に理工系の方向からの関心でなく、ゲームやアニメーションやあるいは文学など人文系の方向から、プログラミングを学ぼうとする動機に応えることのできるプログラミング学習の教材が求められる。発展性を担保するために基礎要素は確実に理解しながらも、グラフィックスや音声、インタラクションなどメディア・テクノロジーの魅力的要素を活用し、楽しみや驚きを生み出す題材を取り上げたプログラミング学習の教育プログラムをさらに開発していきたいと考える。例えば、人のジェスチャや表情とインタラクションをとることを取り入れることが考えられる。SV 教育プログラムに加えて、新しい形のインタラクションを発想できる基礎的素養の養成につながるような題材を使い、プログラミング教育の新しいアプローチを構築することを目指したい。

謝辞

研究の機会を与えてくださり、本研究の全過程において、懇切なるご指導とご支援をいただきました、大阪大学大学院情報科学研究科 増澤利光教授に心より深く感謝いたします。また、本論文作成にあたり、有益なご教示とご指摘をいただきました、大阪大学大学院情報科学研究科 萩原兼一教授、楠本真二教授に感謝を申し上げます。

本論文作成におけるご指導のみならず、研究活動において、多大なご助言をいただきました、大阪大学大学院情報科学研究科 角川裕次准教授に心より感謝の意を表します。

また、大阪市立大学大学院創造都市研究科 松浦敏雄教授には、筆者が大阪大学で研究を行うことに対してご助言いただき、その後の研究活動においても貴重なご支援をいただきました。心より深く感謝いたします。

加えて、研究を進めるにあたり、適切なご助言とご協力をいただきました大阪大学大学院情報科学研究科 大下福仁助教（現在、奈良先端科学技術大学院大学情報科学研究科准教授）に感謝を申し上げます。

これまでの筆者の研究において、さまざまな場面でご協力とご支援をいただいた、同志社女子大学学芸学部情報メディア学科 森公一教授、大阪電気通信大学大学院総合情報学研究科 渡部隆志教授に改めてお礼を申し上げます。また、研究活動において、ご協力いただいた京都芸術デザイン専門学校 大谷俊朗氏のご厚意に感謝いたします。

本研究を進める過程においてご協力いただきました、大阪大学大学院情報科学研究科増澤研究室の皆様に感謝いたします。さらに、同志社女子大学学芸学部情報メディア学科の教職員の皆様には、本研究を進めるための機会とご配慮をいただきました。深くお礼申し上げます。

最後に、このように研究を進められたことをきっと喜んでくれるだろう今は亡き両親、また今までの研究活動を陰で応援してくれた夫 雅夫に感謝します。

参考文献

- [1] Lego Mindstorms, available from <<http://mindstorms.lego.com/>>.
- [2] PicoCricket, available from <<http://www.picocricket.com>>.
- [3] National Research Council. *Being Fluent With Information Technology*. Washington, DC: National Academy Press, 1999.
- [4] 中橋雄, 水越敏行. メディア・リテラシーの構成要素と実践事例分析. 日本教育工学雑誌, Vol.27(suppl), pp.41-44, 2004.
- [5] 辰己丈夫, 中野由章, 野部緑, 川合慧. 情報フルーエンシーを意識した大学の一般情報教育のカリキュラム提案. 情報処理学会研究報告コンピュータと教育(CE), 2009-CE-100(9), pp.1-8, 2009.
- [6] Association of College & Research Libraries Board of Director. Visual literacy competency standards for higher education. *College & Research Libraries News*, Vol.73, No.2, pp.97-104, 2012.
- [7] Don Norman. Why design education must change. Core 77 Nov 26, 2010, available from <http://core77.com/blog/columns/why_design_education_must_change_17993.asp>.
- [8] Processing, available from <<http://processing.org/>>.
- [9] Casey Reas. Programming media. 邦訳 プログラミング・メディア. *Inter Communication*, No.47, Winter, pp.24-31, NTT 出版, 2004.
- [10] Pamela B. Lawhead, Michael E. Duncan, Constance G. Bland, Michael Goldweber, Madeleine Schep, David J. Barnes, and Ralph G. Hollingsworth. A road map for teaching introductory programming using LEGO© mindstorms robots. *ACM SIGCSE Bulletin*, Vol.35, No.2, pp.191-201, 2003.
- [11] Ursula Wolz. Teaching design and project management with lego RCX robots. *ACM SIGCSE Bulletin*, Vol.33, No.1, pp.95-99, 2001.
- [12] Ryota Kuwakubo. DIY hardware: Reinventing hardware for the digital do-it-yourself revolution. *Proceedings of ACM SIGGRAPH ASIA 2009 Art Gallery & Emerging Technologies*, pp.66-67, 2009.
- [13] Hyun Ju Kim, Diana Coluntino, Fred G. Martin, Linda Silka, and Holly A. Yanco. Artbotics: community-based collaborative art and technology education. *Proceedings of*

ACM SIGGRAPH 2007 educators program (SIGGRAPH '07), Article No. 6, 2007.

- [14] Anna Ursyn, Terry Scott, Benjamin R. Hobgood, and Lizette Mill. Combining art skills with programming in teaching computer art graphics. *ACM SIGGRAPH Computer Graphics*, Vol.31, No.3, pp.60-61, Aug 1997.
- [15] Anna Ursyn, and Terry Scott. Web with art and computer science. *Proceedings of ACM SIGGRAPH 2007 educators program*, Article No. 22, 2007.
- [16] Naomi Ribner, and P. Takis Metaxas. The art and science of multimedia: an interdisciplinary approach to teaching multimedia at a liberal arts college. *Proceedings of ACM SIGGRAPH 98 Conference abstracts and applications*, pp.46-51, 1998.
- [17] Seymour A. Papert. *MINDSTORMS: Children, Computers, and Powerful Ideas, Second Edition*. New York, NY: Basic Books, 1993. 奥村貴世子(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ィア.未来社, 1995, 2.2.2 項の引用部分は邦訳 pp.36.
- [18] László Moholy-Nagy. *The New Vision: Fundamentals of Bauhaus Design, Painting, Sculpture, and Architecture*. pp.18, New York, NY: Dover Publications, 2005. (Originally published in 1938).
- [19] Shigeru Kobayashi, Takanori Endo, Katsuhiko Harada, and Shosei Oishi. GAINER: a reconfigurable I/O module and software libraries for education. *Proceedings of the 2006 conference on new interfaces for musical expression* (NIME06), pp.346-351, 2006.
- [20] 兼宗進. 教育用プログラミング言語の動向. 情報処理, Vol.48, No.6, pp.589-593, 2007.
- [21] Ernie Giangrande, Jr. CS1 programming language options. *Journal of Computing Sciences in Colleges*, Vol.22, No.3, pp.153-160, 2007.
- [22] Paul Gross, and Kris Powers. Evaluating assessments of novice programming environments. *Proceedings of the first international workshop on computing education research (ICER '05)*, pp.99-110, 2005.
- [23] Linda Mannila, and Michael de Raadt. An objective comparison of languages for teaching introductory programming. *Proceedings of the 6th Baltic Sea conference on computing education research (Baltic Sea '06)*, pp.32-37, 2006.
- [24] Niklaus E. Wirth. The programming language pascal. *Acta Informatica*, Vol.1 No.1, pp.35-63, 1971.
- [25] Guidovan Rossum, Computer Programming for Everybody.
Available from <<https://www.python.org/doc/essays/cp4e/>>

- [26] Bertrand Meyer. Towards an object-oriented curriculum. *Proceedings of the eleventh international conference on technology of object-oriented languages and systems (TOOLS '93)*, pp.585-594, 1993.
- [27] Mitchel Resnick, Mary Flanagan, Caitlin Kelleher, Matthew MacLaurin, Yoshiki Ohshima, Ken Perlin, and Robert Torres. Growing up programming: democratizing the creation of dynamic, interactive media. *Proceedings of the computer-human interaction conference (CHI '09)*, pp.3293-3296, 2009.
- [28] 大谷尚. SCAT: Steps for coding and theorization - 明示的手続きで着手しやすく小規模データに適用可能な質的データ分析手法 -. 感性工学, Vol.10, No.3, pp.155-160, 2011.
- [29] 曾我部潔, 田村恭久, 高岡詠子. 上智大学における情報フルエンシー系科目設置までの経緯と経過(前). 情報処理, Vol.53, No.6, pp.619-622, 2012.
- [30] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, Vol.49, No.3, pp.33-35, 2006.
- [31] 情報処理学会情報処理教育委員会. 日本の情報教育・情報処理教育に関する提言 2005. Available from <<https://www.ipsj.or.jp/12kyoiku/teigen/v81teigen-rev1a.html>>, 2006.
- [32] 河村一樹. 一般情報処理教育(J07-GE). 情報処理, Vol.49, No.7, pp.768-774, 2008.
- [33] Michael Mateas. Procedural literacy: educating the new media practitioner. *Beyond Fun: Serious Games and Media*, pp.80-96, Pittsburgh, PA: ETC Press, 2008.
- [34] Lev Manovich. *Software Takes Command*. pp.20, New York, NY: Bloomsbury Academic, 2013.
- [35] Matthew Griffin, Susanne Herrmann, and Friedrich Kittler. Technologies of writing: interview with Friedrich A. Kittler. *New Literary History*, Vol.27, No.4, pp.731-742, 1996.
- [36] Jan Moon, and Carlos De Backer. The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, Vol.60, No.1, pp.368-384, 2012.
- [37] Stuart Reges. Back to basics in CS1 and CS2. *ACM SIGCSE Bulletin*, Vol.38, No.1, pp.293-297, 2006.
- [38] Frances Bailie, Mary Courtney, Keitha Murray, Robert Schiaffino, and Sylvester Tuohy. Objects first – does it work? *Journal of Computing Sciences in Colleges*, Vol.19, No.2, pp.303-305, 2003.
- [39] 立木秀樹, 有賀妙子. すべての人のための Java プログラミング第 2 版. 共立出版, 2007.

- [40] 兼宗進, 御手洗理英, 中谷多哉子, 福井眞吾, 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.SIG_11(PRO_12), pp.78-90, 2001.
- [41] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: programming for all. *Communications of the ACM*, Vol.52, No.11, pp.60-67, 2009.
- [42] Alan Kay. Squeakland. Available from <<http://www.squeakland.org>>.
- [43] Caitlin Kelleher and Randy Pausch. Lowering the barriers to programming: a taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, Vol.37, No.2, pp.83-137, 2005.
- [44] Natalie Rusk, Mitchel Resnick, Robbie Berg, and Margaret Pezalla-Granlund. New pathways into robotics: strategies for broadening participation. *Journal of Science Education and Technology*, Vol.17, No.1, pp.59-69, 2008.
- [45] Taeko Ariga, and Koichi Mori. Sensory vision – development of a course for physical interaction and graphics, *Computers & Graphics*, Vol.34, No.6, pp.800-810, 2010.
- [46] 有賀妙子, 森公一. フィジカル・インターラクションを使ったメディア造形基礎教育におけるプログラミング学習の実践. 情報処理学会論文誌, Vol.52, No.12, pp.3096-3105, 2011.
- [47] Association for Computing Machinery, and IEEE Computer Society. Computer Science Curricula 2013.
Available from <<http://www.acm.org/education/CS2013-final-report.pdf>>.
- [48] John Whitney. *Digital Harmony: On the Complementarity of Music and Visual Art*. New York, NY: McGraw-Hill Inc., 1981.
- [49] Casey Reas and Ben Fry. *Getting started with Processing*. Sebastopol, CA: O'Reilly Media, Inc., 2010.
- [50] Albert Bandura. *Self-Efficacy in Changing Society*. Cambridge: Cambridge University Press, 1997. 本明寛, 春木豊, 野口京子ほか(訳): 激動社会の中の自己効力, pp.18, pp.185, 金子書房, 1997.
- [51] Vennila Ramalingam, and Susan Wiedenbeck. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, Vol.19, No.4, pp.365-379, 1998.
- [52] Vennila Ramalingam, Deborah LaBelle, and Susan Wiedenbeck. Self-efficacy and mental

- models in learning to program. *ACM SIGCSE Bulletin*, Vol.36, No.3, pp.171-175, 2004.
- [53] Sankara Sethuraman, and M. Dee Medley. Age and self-efficacy in programming. *Journal of Computing Sciences in Colleges*, Vol.25, No.2, pp.122-128, 2009.
- [54] Joost C. F. de Winter, and Dimitra Dodou. Five-point Likert items: t test versus Mann-Whitney-Wilcoxon. *Practical Assessment, Research & Evaluation*, Vol.15, No.11, 12 pages, 2010.
- [55] Ben Fry, and Casey Reas. A collection of step-by-step lessons covering beginner, intermediate, and advanced topics. Available from <<http://www.processing.org/tutorials>>.
- [56] Abe Pazos. Fun programming. Available from <<http://funprogramming.org>>.
- [57] Sara Alway-Rosenstock. ReDesign: The design education of a non-designer. *Design Principles & Practices: An International Journal*, Vol.5, No.1, pp.173-180, 2011.
- [58] John Tyers. Personality and other attributes, qualities, abilities and opinions of some 'A' level designstudents. *International Journal of Technology and Design Education*, Vol.1, No.3, pp.159-170, 1991.
- [59] Sy-Chyi Wang, Kyle L. Peck, and Jin-Yuan Chern. Difference in time influencing creativity performance between design and management majors. *International Journal of Technology and Design Education*, Vol.20, No.1, pp.77-93, 2010.
- [60] Ronnie Lipton. *The Practical Guide to Information Design*. Hoboken, NJ: Wiley, 2007.
- [61] William Lidwell, Kritina Holden, and Jill Butler. *Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach through Design*. London: Rockport Publishers, 2010.
- [62] David Sherwin. *Creative Workshop: 80 Challenges to Sharpen Your Design Skills*. Blue Ash, OH: How Design Books, 2010.
- [63] Robin Williams. *The Non-Designer's Design Book* (3rd edition). Upper Saddle River, NJ: Pearson Education, 2008.
- [64] Tony Seddon, and Jane Waterhouse. *Graphic Design for Non-Designers*. San Francisco, CA: Chronicle Books, 2009.
- [65] Taeko Ariga, and Takashi Watanabe. Teaching materials to enhance the visual expression of web pages for students not in art or design majors. *Computers & Education*, Vol.51, No.2, pp.815-828, 2008.

- [66] 有賀妙子, 渡部隆志, 由良泰人. マルチメディア表現. 実教出版株式会社, 2005.
- [67] Ellen Lupton. *Graphic Design Thinking: Beyond Brainstorming*. New York, NY: Princeton Architectural Press, 2011.
- [68] Patrick Lynch, and Sarah Horton. *Web Style Guide, Basic Design Principles for Creating Web Sites*. New Haven, CT: Yale University Press, 1999.
- [69] WEB CREATORS LAB. プロセスオブウェブデザイン企画からデザインへ落とし込みの技術. 翔泳社, 2006.
- [70] Yi Lin Wong, and Kin Wai Michael Siu. A model of creative design process for fostering creativity of students in design education. *International Journal of Technology and Design Education*, Vol.22, No.4, pp.437-450, 2012.
- [71] Jon Kolko. Abductive thinking and sensemaking: the driver of design synthesis. *Design Issues*, Vol.26, No.1, pp.15-28, 2010.
- [72] Tom Kelley, and David Kelley. *Creative Confidence*. New York, NY: Crown Business, 2013.
- [73] Peter Felten. Visual literacy. *Change: The Magazine of Higher Learning*, Vol.40, No.6, pp.60-63, 2008.
- [74] Steven Heller. *Teaching Graphic Design: Course Offerings and Class Projects from the Leading Graduate and Undergraduate Programs*. New York, NY: Allworth Press, 2003.
- [75] Elizabeth Resnick. *Design for Communication: Conceptual Graphic Design Basics*. Hoboken, NJ: Wiley, 2003.
- [76] 日本デザイン学会, カリキュラム開発プロジェクトの活動報告. デザイン学研究特集号, Vol.13, No.3, pp.60-73, 2006.
- [77] Carolyn A. Knight, and Jessica Glaser. *The Graphic Design Exercise Book: Creative Briefs to Enhance Your Skills and Develop Your Portfolio*. Blue Ash, OH: How Design Books, 2010.
- [78] Carey Jewitt. *The Visual in Learning and Creativity: A Review of the Literature: A report for Creative Partnerships*. London: Arts Council England, 2008.
- [79] Jenny Waycott, Judithe Sheard, Celia Thompson, and Rosemary Clerahan. Making students' work visible on the social web: a blessing or a curse? *Computers & Education*, Vol.68, pp.86-95, 2013.
- [80] Lorrie Knight. Using rubrics to assess information literacy. *Reference Services Review*, Vol.34, No.1, pp.43-55, 2006.

- [81] Renee M. Clary, Robert F. Brzuszek, and C. Taze Fulford. Measuring creativity: a case study probing rubric effectiveness for evaluation of project-based learning solutions. *Creative Education*, Vol.2, No.4, pp.333-340, 2011.
- [82] Suzan Besemer, and Donald J. Treffinger. Analysis of creative products: Review and synthesis. *Journal of Creative Behavior*, Vol.15, No.3, pp.158-178, 1981.
- [83] Suzan Besemer, and Karen O'Quin. Confirming the three-factor creative product analysis matrix model in an American sample. *Creativity Research Journal*, Vol.12, No.4, pp.287-296, 1999.
- [84] Karen O'Quin, and Suzan Besemer. The development, reliability, and validity of the revised creative product semantic scale. *Creativity Research Journal*, Vol.2, No.4, pp.267-278, 1989.
- [85] Randy P. Howe. Uncovering the creative dimensions of computer-graphic design products. *Creativity Research Journal*, Vol.5, No.3, pp.233-243, 1992.
- [86] Alisa White, and Bruce L. Smith. Assessing advertising creativity using the creative product semantic scale. *Journal of Advertising Research*, Vol.41, No.6, pp.27-34, 2001.
- [87] Karen O'Quin, and Suzan Besemer. Using the creative product semantic scale as a metric for results-oriented business. *Creativity and Innovation Management*, Vol.15, No.1, pp.34-44, 2006.
- [88] Cropley Arthur. *Creativity in Education and Learning: A Guide for Teachers and Educators*. London: Routledge, 2002.
- [89] 野口薫. 美の感性の心理学 ゲシュタルト知覚の新しい地平. 日本大学文理学部叢書, 2007.
- [90] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. Redlands, CA: Environmental Systems Research Institute Press, 2010. (Original work published 1967)
- [91] Paul Mijksenaar. *Visual Function: An Introduction to Information Design*. New York, NY: Princeton architectural press, 1997.
- [92] ロラン・バート. 第三の意味—映像と演劇と音楽と. pp.1, みすず書房, 1984. (原著出版は1961).
- [93] ロラン・バート. ロラン・バート著作集 3, 現代社会の神話. みすず書房, 2005. (原著出版は1957).
- [94] Daniel Chandler. Denotation, connotation and myth, *Semiotics: The Basics*. London: Routledge, 2007.

[95] Ian Noble, and Russell Bestley. *Visual Research: An Introduction to Research Methodologies in Graphic Design*. London: Bloomsbury Publishing, 2011.

[96] 季里. 講演会デジタルの仕事について.

Available from <<http://www.dwcmedia.jp/archives/05l02>>.

