

## デザイナーとアーティストの為の基礎的なプログラミングの教育方法について

### STUDY ON EDUCATIONAL METHODS OF FUNDAMENTAL PROGRAMMING FOR DESIGNER AND ARTIST

尹 智博 大学院芸術工学研究科 助教  
大内 克哉 基礎教育センター 教授  
榮元 正博 芸術工学部ビジュアルデザイン学科 助教

Jibak YOON Graduate School of Arts and Design, Assistant Professor  
Katsuya OUCHI Center for Liberal Arts, Professor  
Masahiro EIGEN Department of Visual Design, School of Arts and Design, Assistant Professor

#### 要旨

2013年6月に日本国政府は、「世界最先端IT国家創造宣言」を閣議決定した。その宣言の中には、2020年代から初等教育の中にプログラミング教育を導入する事が含まれていた。また新しいモノづくりであるデジタル・ファブリケーションやデバイス制御などの技術が修得できる環境整備を、教育環境のIT（Information Technology）化とともに推進していく事が決められた。

神戸芸術工科大学（以下、KDU）には、7学科（環境デザイン、プロダクト・インテリアデザイン、ビジュアルデザイン、映像表現、まんが表現、ファッションデザイン、アート・クラフト）27コースある。各学科では、それぞれの専門領域のもとに様々なカリキュラムが生まれ、必要に応じてプログラミングの授業（主にWeb系や3DCG）が開講されている。しかし今後は、義務教育の中にプログラミング教育が含まれることを考えると、ひとつの専門に偏るのではなく、デザインやアートの表現を行うための基礎的なプログラミング教育を構築する必要がある。

本研究は、KDUにおけるデザインやアートの表現を行う為の基礎的なプログラミングの導入教育方法の構築を目的とし、KDUインタラクションデザインコースの授業を中心に様々な教育実験を行った。

#### Summary

Japanese Government decided “Declaration on the Creation of the World’s Most Advanced IT Nation.” at the Cabinet meeting in June 2013. There was introducing programming education in primary education from 2020 in the declaration. And there was establish an environment in which you can acquire digital fabrication and device control technologies in the declaration.

Kobe Design University has 27 courses seven departments (Environmental Design, Product and Interior Design, Visual Design, Image Arts, Manga Media, Fashion and Textile Design, Arts and Crafts). In each department, various curriculums were organized under each specialized field, and programming classes (mainly web system and 3DCG) were held as needed. In the future, there is a need to construct basic programming education for expressing design and art in the KDU curriculum.

This research aimed at the introduction of basic programming introduction education method for expressing designs and arts in KDU, and conducted various educational experiments mainly on classes of KDU interaction design course.

### 1. 研究目的

本研究は、神戸芸術工科大学（以下、KDU）におけるデザインやアートの表現を行う為の基礎的なプログラミングの導入教育を構築する事を目的とする。

### 2. 研究背景

2013年6月に日本国政府は、「世界最先端IT国家創造宣言」を閣議決定した。その宣言の中には、2020年代から初等教育の中にプログラミング教育を導入する事が含まれていた。また新しいモノづくりであるデジタル・ファブリケーションやデバイス制御などの技術が修得できる環境整備を、教育環境のIT（Information Technology）化とともに推進していく事が決められた。

KDUでは、7学科（環境デザイン、プロダクト・インテリアデザイン、ビジュアルデザイン、映像表現、まんが表現、ファッションデザイン、アート・クラフト）27コースの専門領域のもとに様々なカリキュラムが組み立てられ、必要に応じてプログラミングの授業（主にWeb系や3DCG）が開講されている。しかし今後は、義務教育の中にプログラミング教育が含まれることを考えると、ひとつの専門に偏るだけではなく、デザインやアートの表現を行う為の基礎的なプログラミング教育を構築する必要があると考える。

### 3. 研究の到達目標

執筆者が考えるデザインやアートの表現を行う為の基礎的なプログラミングの導入教育の到達目標は、「Processing」と「Arduino」を「シリアル通信」で連携して、アナログセンサーを用いてモーショングラフィックスを制作出来るようになる事である。

「Processing」は、MIT(Massachusetts Institute of Technology)メディアラボのケイシー・リース (Casey Reas) とベン・フライ (Ben Fry) によって、デザイナーとアーティストの為のプログラミング入門として2001年に開発された言語である。「Processing」は、当初グラフィックスとインタラクションに焦点を絞り

開発されていたが、2002年のアルファ版公開から最新版のVer.3.3.5（2017年7月31日時点）に至るまで、その使用が容易いことから世界中にそのユーザーが増え、様々なライブラリがユーザーによって開発され、現在では音楽制作や視線トラッキング、AR(Augmented Reality)、VR(Virtual Reality)、「Arduino」との連携なども行える様になった。

「Arduino」は、IDII(Interaction Design Institute Ivrea)のマッシモ・バンジ (Massimo Banzi) によって、電子制御を容易く行える基盤として開発された。またバンジは、基盤制御用の開発環境として「Arduino IDE」を「Processing」をベースに開発した。これによって、電子工学を主に学習しなかったデザイナーやアーティストでも、プログラミングの延長でインタラクティブな作品を制作する手段を手に入れる事が可能となった。

この2つの言語および基盤は、「シリアル通信」や「OSC(Open Sound Control)」などの各種通信を用いてインタラクティブな関係を構築する事が可能であり、双方の言語がメディア系の高等教育機関の授業カリキュラムに取り入れられ、世界中で様々な教育が行われている。

KDUにおいては、2014年度から各専門領域を横断するコースであるインタラクティブデザインコース（以下、IDC）の「サイエンティフィック・プログラミング演習」（担当教員：大内克哉、尹智博、小山明）において「Processing」と「Arduino」の連携をベースとしたプログラミング教育を行っている。履修した学生達は、初めてプログラム学習を行った人達が多くあるものの最終的には、「Processing」と「Arduino」を「シリアル通信」で連携して、アナログセンサーを用いてモーショングラフィックスをプログラムで制作する事が出来る様になっている。一方で、授業開始から数回については、「if文」や「for文」などの簡単な構文を理解出来ずに立ち止まってしまう学生も確認される。また「Arduino」でも電子制御における回路図や配線が理解出来ずに同様の事が起きてしまう。現在、上

記科目の履修生はインタラクティブデザインに関心がある学生であり、その学生達でもプログラミング教育の最初には理解出来ずに立ち止まってしまう事が確認される。しかし、これからはそうした学生と合わせ、ITにそこまで関心が無い学生にも理解し易いカリキュラムを構築していかなければならない。

そこで本研究では、MITが開発したアルゴリズムの教育用言語でもある「Scratch」と、電子工作における回路図や配線などを考えずに直感的電子パーツを繋ぎ合わせる事によって様々な動きを簡単に制作する事が出来る「littleBits」を用いて、デザインやアートの表現を行う為の基礎的なプログラミングの導入教育を構築する事を試みた。

#### 4. 研究方法

研究方法は、執筆者が担当する IDC のプログラミングやデバイス制御に関する科目(「サイエンティフィック・プログラミング演習」、「プログラミングカルチャー」、「学科横断型プログラム(インタラクティブデザインユニット)」、「ワークショップ(夏季学外演習)」)で様々な教育実験を実施した。

右掲の図版は、本研究で考案した導入教育のステップ案である。

##### ■ステップ 1 (図 1)

Scratch によるアルゴリズム学習

littleBits による電子工作

Scratch と littleBits の連携

##### ■ステップ 2 (図 2)

littleBits と Arduino の互換

電子回路図の学習

##### ■ステップ 3 (図 3)

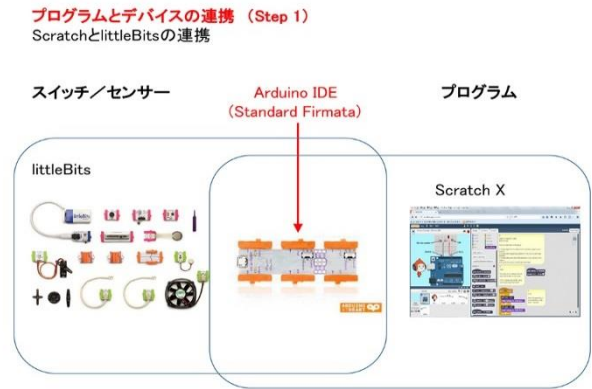
各種プログラミング言語の学習

(IDC では、Processing、MAX、Grasshopper、Sonic-Pi などの言語を使用している)

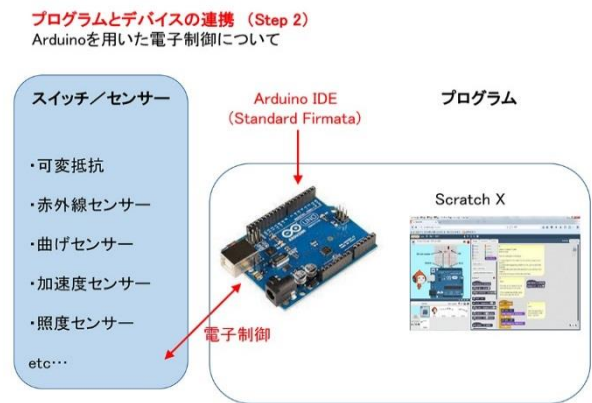
##### ■ステップ 4 (図 4)

各種プログラミング言語と Arduino の連携

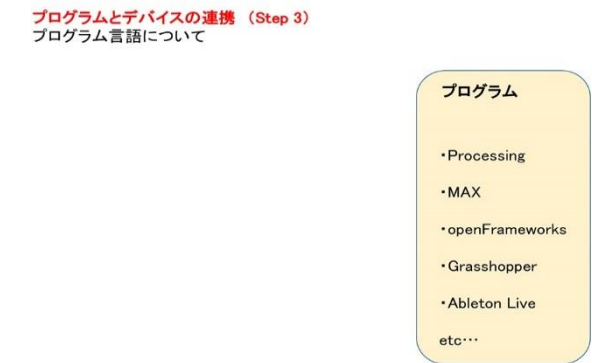
シリアル通信、OSC の学習



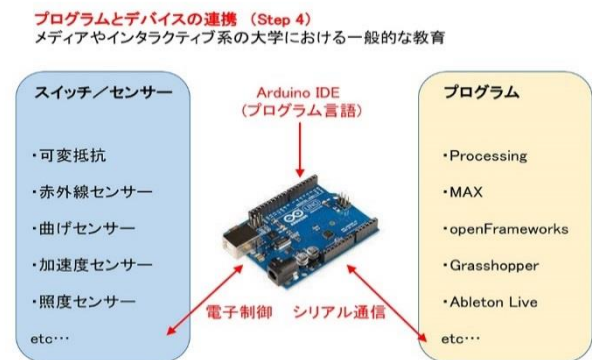
(図 1) ステップ 1



(図 2) ステップ 2



(図 3) ステップ 3



(図 4) ステップ 4

一般的なメディア系の高等教育機関および本学の IDC では、ステップ 3 とステップ 4 の教育が行われている。これらはインタラクションに特化したプログラミング教育ではあるが、2020 年代からの初等教育にプログラミングを含む IT 教育が始まると、デザイナーやアーティストにとっては、基礎的な制作ツールとして身につけておかなければならないものとする。

そこで、それらの導入教育として、プログラミングに不慣れた学生達でも容易く学習できる教育方法として、ステップ 1 とステップ 2 の教育実験を IDC 科目内で試みた。

## 5. 研究の教育実験

教育実験は、ステップ毎に個別の授業で実施した。

ステップ 1 の「Scratch によるアルゴリズム学習」と「littleBits による電子工作」(写真 1)については、「ワークショップ(夏季学外演習)」の授業において履修生 12 名に対して行った。ステップ 2 は、個別に実施しなかった。ステップ 3 の「各種プログラミング言語の学習」(写真 2)は、「学科横断型プログラム(インタラクションデザインユニット)」において履修生 11 名に対して行った。ステップ 4 の「各種プログラミング言語と Arduino の連携」と「シリアル通信、OSC の学習」については、「プログラミング・カルチャー」の授業において履修生 7 名に対して行った。

そして、ステップ 1~4 を通しては、「サイエンティフィック・プログラミング演習」にて履修生 12 名に対して行った(写真 3)。

各授業の履修生達には、個別のアンケートを実施しフィードバックを行った。アンケートで記入されたコメントには、「Scratch」と「littleBits」を用いる事によって得られるであろうと想定していたメリット・デメリットが学生達によっても記載されており、よりそのデメリット部分について分かり易く教育システムを構築する必要がある事を実感した。

双方のメリットについては、基本的には使用しやすい事やプログラミングの学習が思っていたよりも単純

であった事などがあった。一方のデメリットでは、「Scratch」に関して変数の設定などが煩わしい事や、「littleBits」に関してピンの設定情報が分かりづらいなど、コメント記入者のプログラミングに対する理解力の高さから記入されたものであった。

## 6. まとめ

IDC のプログラミングに関する授業内で様々な教育実験を行ったが、今後はより多くの学生に教育実験を行い、母数を大きくして IT 教育時代に対応する、デザインとアートの為の基礎的なプログラミングの教育方法について継続して検証していきたいと考える。



(写真 1) ステップ 1 のワークショップ風景。



(写真 2) 「各種プログラミング言語の学習」のワークショップ風景。ここでは「MAX」のワークショップを行った。



(写真 3) 「サイエンティフィック・プログラミング演習」の授業風景。「littleBits」と「Processing」を「シリアル通信」で連携して、モーショングラフィックスを制作した。  
註：図版は執筆者作成、写真は執筆者撮影