

中学生を対象としたプログラミング講座の実施計画について

土肥紳一*

Email: dohi@mail.dendai.ac.jp

*: 東京電機大学システム デザイン工学部 デザイン工学科

◎Key Words 中学生, プログラミング, 技術家庭(技術分野), Processing

1. はじめに

東京電機大学の東京千住キャンパスは、足立区の北千住の駅前に立地している。この度、足立区から中学生を対象としたプログラミング講座の依頼があり、その実施計画について検討を行っている。開催日は2025年9月27日(土)の13:00~15:00に学内の教室で予定している。ノートPCは15台準備し、親子での参加も歓迎している。プログラミング言語はスケッチブックに絵を描く感覚で学べるProcessing言語を使う⁽¹⁾。少ないキー入力でお絵描きを行え、実行結果はグラフィカルに表示できるため、興味を喚起できることを期待している。本論文では、中学校の技術家庭の技術分野でどのようなプログラミング言語を使っているのかを探り、Processingを使った実施内容を述べ、その後続く講座についても述べる。中学生を対象とした同様の講座を担当された経験のある先生方から、本提案に対する意見やコメントをうかがえるとありがたい。

2. 技術家庭(技術分野)のプログラミング言語

中学校の技術家庭(技術分野)でどのようなプログラミング言語が取り上げているのかを探るために、3つの出版社から教科書を購入し調査した^{(2)~(4)}。教科書の中に記載のあったプログラミング言語を表1に示す。JavaScript, Python, Scratchは、調査した3社が共通に取り上げていた。次に多かったのがHTML, Ruby, ドリトルであった。

ビジュアル(ブロック)型言語としてScratchが取り上げられている。Scratchはスクラッチと表記されているものもあり、ここではScratchとして記述した。テキスト型言語として、なでしこ、ドリトル、HTML、JavaScriptが取り上げられている。

表1 プログラミング言語

言語	A社	B社	C社	計
BASIC	1			1
C	1			1
CSS	1			1
HTML	1	1		2
JavaScript	1	1	1	3
PHP	1			1
Python	1	1	1	3
Ruby	1		1	2
Scratch	1	1	1	3
WebAPI			1	1
スモウルビー			1	1
ドリトル	1	1		2
なでしこ			1	1

各教科書の中で掘り下げて記載されている言語は、Scratchであった。パズル型の言語で、パズルをはめることによって、プログラムを作ることができ、スペルミスなどによる文法の誤りを防げる。

3. Processingについて

3.1 高大連携の実績

情報環境学部時代に高大連携の授業を担当する中で、プログラミング入門体験を実施したことがある。受講者は高校1年生であった。土曜日の10:00~12:00の時間帯で、途中、10分の休憩を挟みながら実施した。受講者数は20名程度であった。この内容は、システムデザイン工学部の誕生を広報する目的で、オープンキャンパスのワークショップとしても開講した。少ないキー入力でお絵描きでき、入力を開始すると同時に文法チェックが始まり、誤りを発見し易く、初学者に親切である。

3.2 デザイン工学科のプログラム言語

システムデザイン工学部は、情報環境学部が発展的に改組を行い2017年4月に東京千住キャンパスに開校した。デザイン工学科のプログラミングの授業は1年生の後期に「コンピュータプログラミングⅠ」の授業が始まる。デザイン工学科ではProcessingをプログラミング言語として採用した。お絵描き感覚でプログラミングを学べることが、最大の理由である。

4. 計画している講座の内容について

4.1 Processingについて

中学生を対象とした講座ではあるが、Processingに関する概要を説明する必要がある。スライドの様子を図1に示す。Javaで作られていることを入れているが、Java自体が何なのかを分からせることは難しそうである。

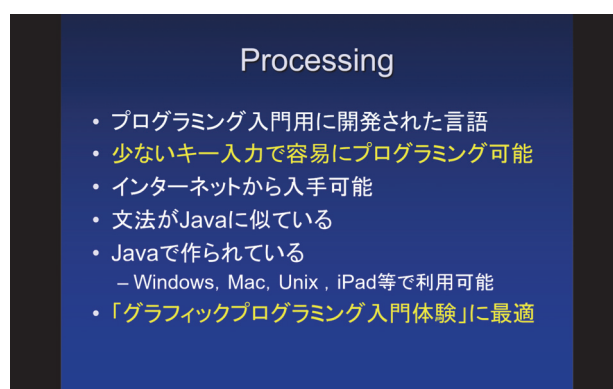


図1 Processingについて

4.2 座標系について

お絵描き感覚でプログラミングを学べる環境ではあるが、座標系の説明が欠かせない。Processing では原点が左上になり、Y 軸は下に向かって大きくなる。中学生が普段使う Y 軸は、上に向かって大きくなるため、注意を要する。スライドの様子を図 2 に示す。

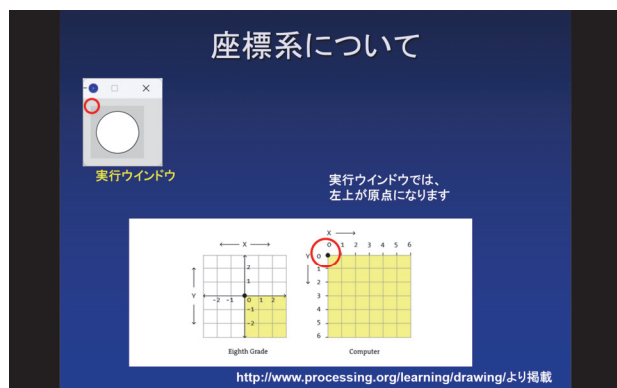


図 2 座標系について

4.3 円の表示

絵を描くときに最初に出てくるのが ellipse である。これは楕円を描くための関数である。中学生は楕円を教わっていないため、円を取り上げる必要がある。ただし円は英語では circle のスペルであるが、なぜ ellipse なのか理屈を説明することになる。つまり、楕円の幅と高さを同じにすることによって円になる。スライドの様子を図 3 に示す。まだ教わっていないことが出てくる難しさがある。



図 3 楕円の表示

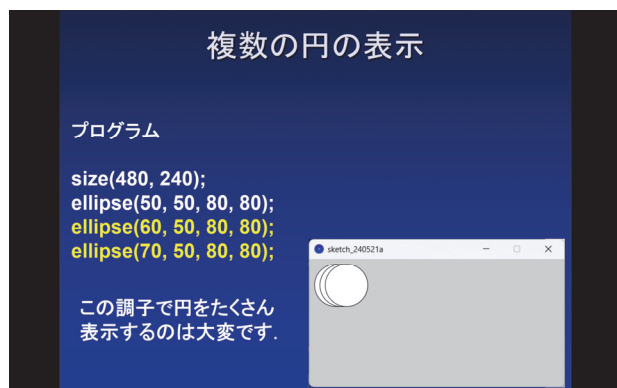


図 4 複数の円の表示

ellipse で絵を描くことが理解できると、たくさん ellipse

を並べることによって、円を増やすことができる。この様子を図 4 に示す。

この時注意しないといけないことは、円の中心の座標が同じだと重なってしまい、1つの円に見えてしまう。後から描いた図形が上に表示されることを理解させる必要がある。また、ウインドウの外の座標も指定できるが、ウインドウの中には表示されないため見えない。その他、大きさを変えたりできることを試すことで、形を色々変えることができることを体験させることができる。

4.4 繰り返しを使った円の表示

たくさんの円を表示するためには、ellipse をたくさん並べる必要があり大変である。そこで登場するのが繰り返しである。人間は同じことを何度も繰り返すことは苦手であるが、コンピュータは繰り返すことが得意である。for 文を使った繰り返しを取り入れると、繰り返す回数を変更するだけで、たくさんの円を瞬時に表示できる。この様子を図 5 に示す。

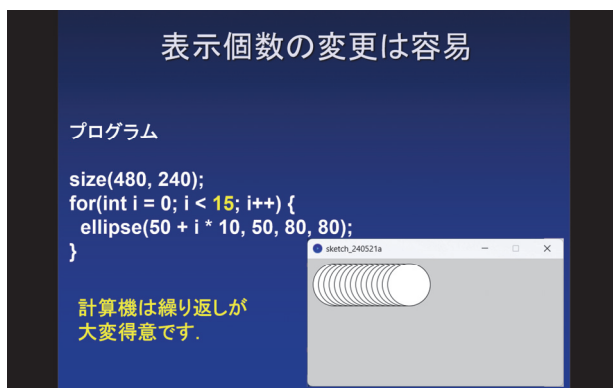


図 5 繰り返しを使った円の表示

大切なことは繰り返しの仕組みを理解することである。この様子を図 6 に示す。繰り返しの仕組みを中学生にどのように教えると理解されるのか、筆者は熟知していない。陸上のトラック競技で、何周したかを数える人がいて、その人が良いというまで何周もするように擬人化して理解させるのが良いかも知れない。

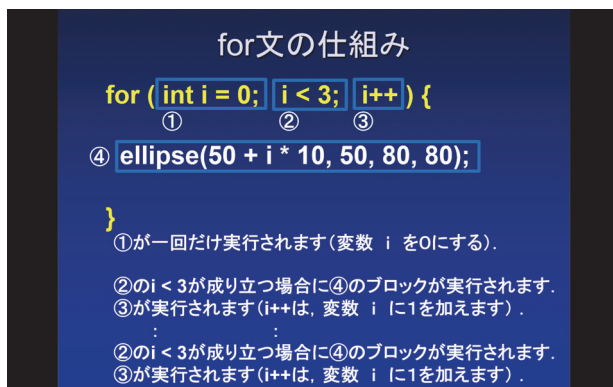


図 6 for 文の仕組み

4.5 基本図形の組み合わせでキャラクターを表示

円や線や四角等を組み合わせることによって、キャラクターを表示できる。元の絵は方眼紙等を使って絵を作成し、それを実現できる基本図形をあてはめ、位置の座標

を読み取る。最初の一つを表示するのは手間がかかるが、一度出できあがると何度描いても同じキャラクターが描ける。人間は同じ絵を何度も書くことは苦手であるが、コンピュータは間違えることなく何度も正確に描けるとことが凄いところである。キャラクターの様子を図 7 に示す。



図 7 キャラクターの表示

4.6 マウスを使った円の表示

Processing は実行ウィンドウ内のマウスの座標を読むことができる、システム変数 `mouseX` と `mouseY` を提供している。プログラムの実行時に一度だけ実行される `setup` 関数を定義でき、ウィンドウの大きさや変数の初期化を行う。さらに 1 秒間に 60 回の速さで無限の繰り返しを実行する `draw` 関数を定義できる。これらの機能を組み合わせることによって、手の動きに合わせて円を表示すると残像が残る、円の重なり具合でお絵描きすることができる。手をゆっくり動かすと円がたくさん重なり、素早く動かすと円がまばらに表示される。この様子を図 8 に示す。非常に短い記述であるが、興味深い絵を描けることは Processing の得意技である。



図 8 マウスを使った円の表示

4.7 マウスを使った線の表示

マウスを使った円の表示のプログラムを少し書き換え、原点とマウスの位置を線で結ぶようにプログラムを変更する。手をゆっくり動かすと線がたくさん重なり黒い部分が多くなり、素早く動かすと線がまばらに表示される。また、原点から離れた位置にマウスを移動し、原点に近い位置に移動することにより、興味深い残像が残る。この様子を図 9 に示す。`ellipse` を `line` に置き換えるだけで、まったく異なる描画を行えることは興味深く、中学生の興味を喚起できることを期待している。

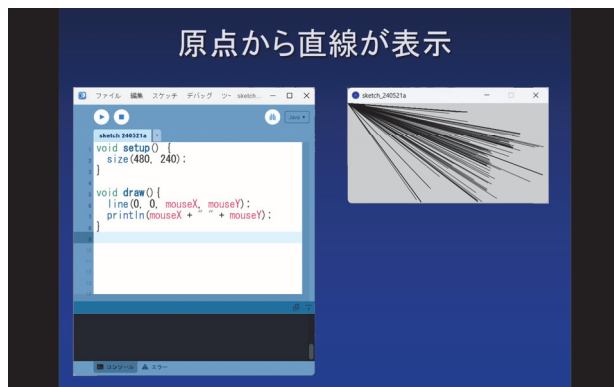


図 9 マウスを使った線の表示

4.8 マウスのボタンを使った円の塗りつぶし

マウスを使った円の表示のプログラムを少し書き換えて、マウスのボタンを押した時に、黒色で塗りつぶすように変更する。マウスのボタンが押されたか否かをシステム変数の `mousePressed` で知ることができ、その結果を `if` 文で判定する。この様子を図 10 に示す。

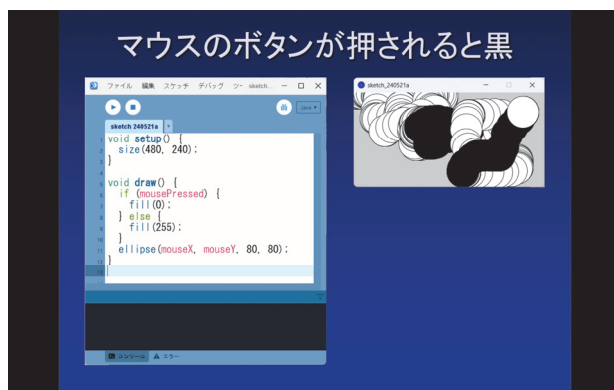


図 10 円の塗りつぶし

マウスのボタンが押されていないときは、白い円が表示されるが、マウスのボタンを押すと黒く塗りつぶされ、ウィンドウ全体を黒く塗りつぶすことを始めてしまう。このような操作は、筆者の憶測であるが、人間の好奇心をくすぐる何かが存在しているようである。`if` 文の仕組みを図 11 に示した。

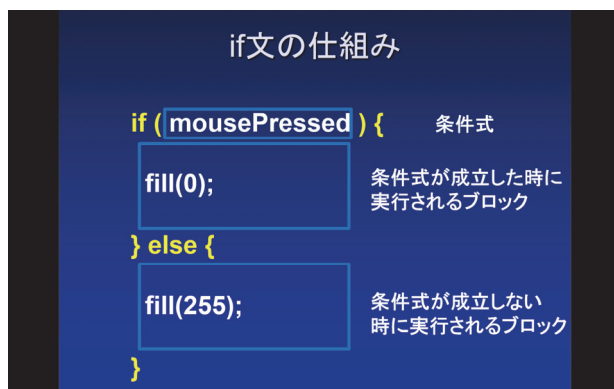


図 11 if 文の仕組み

4.9 塗りつぶしの色の指定

ここまでで紹介してきたプログラムは、モノクロの世界であったが、赤、緑、青の各色を 0~255 の値で指定す

ることができ、この組み合わせで色々な色を作ることができる。この様子を図 12 に示す。



図 12 塗りつぶしの色の指定

4.10 円の回転

講座の最後に予定しているプログラムは、円を円周上で公転させる内容である。この様子を図 13 に示す。このプログラムは三角関数を利用して円周上の座標を計算する必要がある。三角関数は高校 2 年生で教わるのが一般的なようである。中学生は三角比を教わっているが、三角関数のギャップが気になるところである。高校で教わるのを楽しみに待つよう伝えたいと考えている。プログラム中の $\sin$ と $\cos$ を逆にすると、回転方向が逆になる。

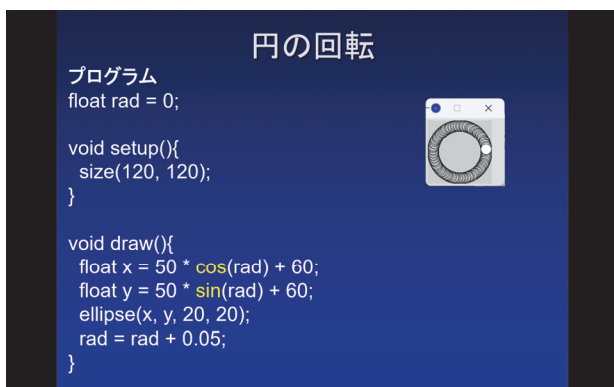


図 13 円の回転

5. さらなる発展について

まだ講座を実施する前であるが、この講座の次に続くものとして L チカを考えている。Processing は Raspberry Pi 4 などで利用できる。Raspberry Pi 4 はキーボード、マウス、ディスプレイをつなぎ、Raspberry Pi OS を入れると、デスクトップコンピュータとして利用できるようになる。Raspberry Pi 4 は GPIO を使って色々な電子部品と接続可能であり、L チカは、その典型例である。

Processing でディスプレイ上に○を 3 つ表示し、左の○の中にマウスが入ると青に、中央の○に入ると黄に、右の○に入ると赤になるように交通信号に似せて画面を作成する。ブレッドボード上には青、黄、赤の LED を横一列に配置し、電流制限抵抗を介しジャンパー線で Raspberry Pi 4 の GPIO と接続する。画面上のマウスの動きと連動し、対応する LED を点灯させる。この様子を図 14～図 16 に示す。単純なプログラムだが、LED を Processing で制御することで、興味を喚起できることを期待している。

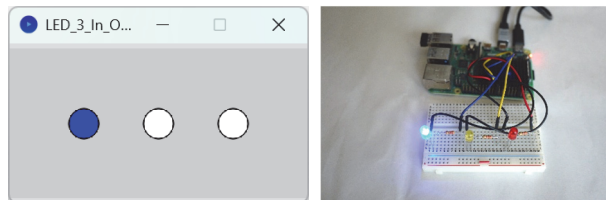


図 14 L チカ(青色)

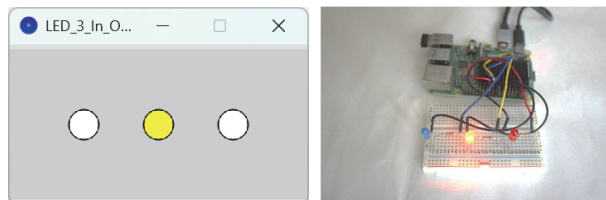


図 15 L チカ(黄色)

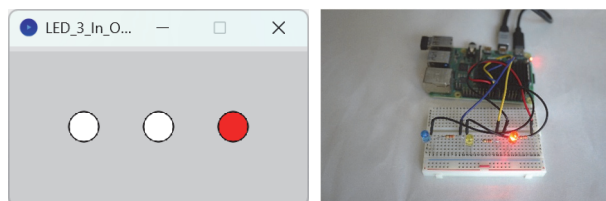


図 16 L チカ(赤色)

提案したプログラミングの講座ではノート PC の活用を想定しているが、その発展としてシングルボードコンピュータを使えるようになると、電子部品を活用した講座に発展させることができる。

ノート PC のディスプレイの中だけでの表示は、絵を描くプログラミングで興味を喚起できるが、さらに電子部品の制御につながると、もっと興味を喚起できると考えている。電子工作を行い、それをコンピュータで制御できることを知ることは、創造性の原動力になる。中学生の方が、電子工作に興味を持ち、その原理や仕組みを理解することは、とても重要である。デジタルネイティブな人たちは、お金を出せば欲しいものが手に入り、物作りから遠ざかっている。ハードウェアを自作し、コンピュータで制御できる喜びを知って欲しいものである。

6. おわりに

本論文では、中学生を対象としたプログラミングの講座の実施計画について述べた。本番では想定外のことも色々発生すると考えられるが、実施結果については今後の PC カンファレンスで報告できればと考えている。

参考文献

- (1) Processing をはじめよう第 2 版, Casey Reas, Ben Fry 著, 船田 巧訳, オライリー, <https://www.oreilly.co.jp/books/9784873117737/>, 2025 年 06 月 30 日閲覧。
- (2) 技術・家庭 技術分野, 開隆堂出版, <https://krdkrk.jp/tech/>, 2025 年 06 月 30 日閲覧。
- (3) 新編 新しい技術・家庭 技術分野 未来を創る Technology, 東京書籍, <https://www.tokyo-shoseki.co.jp/product/textbook/j/7/599/>, 2025 年 06 月 30 日閲覧。
- (4) 新 技術・家庭 技術分野, 明日を創造する, 教育図書, https://www.kyoiku-tosho.co.jp/item/c_g006-73_g006-74/, 2025 年 06 月 30 日閲覧。