





コンセプト

このロゴはアルゴリズムにより生成される。まず11x8のグリッドを用意する。そこに以下の点を設定する。

S: (3,1) (1,3) (3,5) (1,7)

F: (7,1) (5,2) (5,7)

C: (10,1) (8,4) (10,7)

この、それぞれの点を結ぶベジェ曲線を、乱数により発生させたコントロールポイントを使用して、SFCのそれぞれ25本ずつ描いていく。これはSFCを巣立っていった多くの卒業生が、それぞれ自分なりのパス（経路）を描きながら目標に到達していく様を表現している。

ロゴは、Processingでプログラミングされており、使用する環境に応じて（印刷/スクリーン、カラ/グレースケール）解像度や色、使用サイズ適化した状態でPDFファイル形式で生成して使用する。ソースコードはGithubのリポジトリに公開されている。

<https://github.com/tado/SFC25th>

田所 淳（たどころ あつし）

政策・メディア研究科 博士課程2年

email. tadokoro@gmail.com

生成プログラムソースコード（Processing）

```
import processing.pdf.*;

float screenWidth = 640; // set logo width (pixel)
boolean drawColor = true; // color or grayscale
boolean drawGrid = false; // draw 11 x 8 grid
boolean drawFrame = false; // draw outline frame

void setup() {
  int gridSize = int(screenWidth / 11.0);
  size(gridSize * 11, gridSize * 8, PDF, "25th.pdf");
  background(255);
  strokeWeight(0.5);
  noFill();

  float stepX, stepY;
  stepX = width / 11.0;
  stepY = height / 8.0;

  // grid
  if (drawGrid) {
    stroke(0, 31);
    for (int i = 0; i < 11; i++) {
      line(i * stepX, 0, i * stepX, height);
    }
    for (int i = 0; i < 8; i++) {
      line(0, i * stepY, width, i * stepY);
    }
  }

  stroke(0, 63);
  // S
  for (int i = 0; i < 25; i++) {
    beginShape();
    curveVertex(3 * stepX, 1 * stepY);
    curveVertex(3 * stepX, 1 * stepY);
    curveVertex((1 + random(1.0)) * stepX, 3 * stepY);
    curveVertex((3 - random(1.0)) * stepX, 5 * stepY);
    curveVertex(1 * stepX, 7 * stepY);
    curveVertex(1 * stepX, 7 * stepY);
    endShape();
  }
```

```
// F
for (int i = 0; i < 25; i++) {
  beginShape();
  curveVertex(7 * stepX, 1 * stepY);
  curveVertex(7 * stepX, 1 * stepY);
  curveVertex((5 + random(1.0)) * stepX, 3 * stepY);
  curveVertex(5 * stepX, 7 * stepY);
  curveVertex(5 * stepX, 7 * stepY);
  endShape();
}

// C
for (int i = 0; i < 25; i++) {
  beginShape();
  curveVertex(10 * stepX, 1 * stepY);
  curveVertex(10 * stepX, 1 * stepY);
  curveVertex((8 + random(1.0)) * stepX, 4 * stepY);
  curveVertex(10 * stepX, 7 * stepY);
  curveVertex(10 * stepX, 7 * stepY);
  endShape();
}

// dot
if (drawColor) {
  fill(#e84e40);
} else {
  fill(127);
}

noStroke();
float radius = stepX / 3.0;
ellipse(3 * stepX, 1 * stepY, radius, radius);
ellipse(1 * stepX, 7 * stepY, radius, radius);
ellipse(7 * stepX, 1 * stepY, radius, radius);
ellipse(5 * stepX, 7 * stepY, radius, radius);
ellipse(10 * stepX, 1 * stepY, radius, radius);
ellipse(10 * stepX, 7 * stepY, radius, radius);
ellipse(6 * stepX, 4 * stepY, radius, radius);
// frame
if (drawFrame) {
  stroke(31);
  noFill();
  rect(0, 0, width, height);
}
}
```