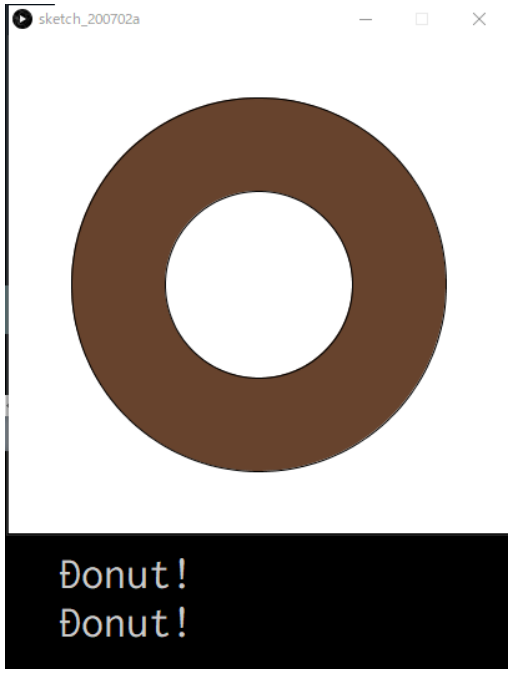
プログラミング演習I （第5回） 課題

- **基本課題①　スケッチ名：basic_Donut**
  - 400x400の大きさのウインドウの背景を白色にし、そのウインドウの中央に、内径75ピクセル、外径150ピクセルの茶色(103，67，45)のドーナツを描け
  - また、ドーナツの部分（茶色の部分）をクリックすると、Donut!と標準出力するようにせよ（ドーナツ以外の部分をクリックしても反応しないようにせよ）



プログラミング演習I （第5回） 課題

- **基本課題②　スケッチ名：basic_Janken**
  - 300x300のウインドウを作成し，(50,150)(150,150)(250,150)の位置にそれぞれ半径50ピクセルの円を表示せよ．いずれかの円をクリックしたとき，左ならグー、中ならチョキ，右ならパーの手をあなたは選んだものとする．このときコンピュータはランダムにグーチョキパーを選び，両者のじゃんけんの手が表示され，結果を下図のように標準出力するプログラムを作成せよ．円の外をクリックした場合は反応しないようにし，何度でもじゃんけんできるようにせよ

クリック!

```
あなたはグー
コンピュータはチョキ
あなたの勝ち
```

クリック!

```
あなたはグー
コンピュータはグー
引き分け
```

クリック!

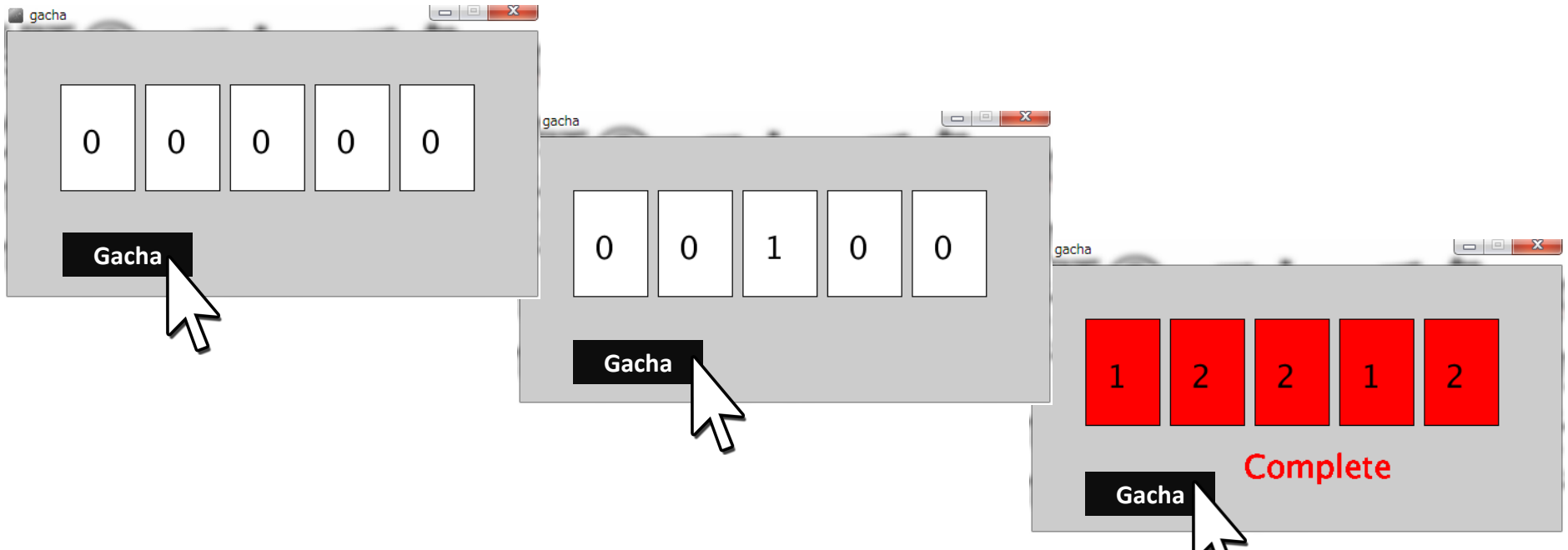
```
あなたはチョキ
コンピュータはチョキ
引き分け
```

クリック!

```
あなたはグー
コンピュータはパー
コンピュータの勝ち
```

プログラミング演習I （第5回） 課題

- **基本課題③ スケッチ名：basic_CompleteGacha**
  - ウインドウ下部のガチャボタンをクリックする度に，5種類のカードの1種類がランダムに選ばれ，枚数が１加算されるプログラムを作成し，それぞれのカードが選ばれた枚数を表示するプログラムを作成せよ（ただしボタン以外では反応しないようにせよ）
  - また，すべてのカードが1枚以上になったら，Completeとウインドウ内にtextを用いて表示し，カードの色を赤色にせよ



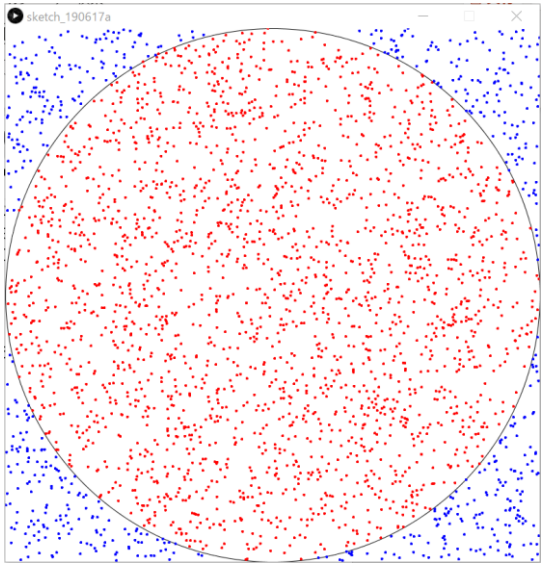
ヒント

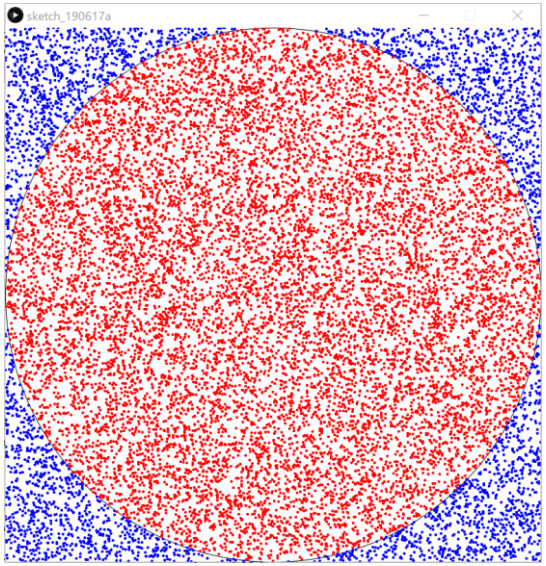
- 基本課題①
  - ドーナツの茶色の部分の判定条件はどうなる？
  - mousePressedの中で標準出力しよう！
- 基本課題②
  - どういう条件になるか、手書きしてみよう
    - ユーザのじゃんけんの手はどうとる？
    - コンピュータのじゃんけんの手はランダムに
    - 勝敗の条件は？
  - mousePressedの中で標準出力しよう！
- 基本課題③
  - 必要な変数はなんだろうか？
  - 全部が1枚以上というのはどういう条件か？
    - 全部が1枚以上かどうかを格納する変数もあるとよいのでは？

プログラミング演習I （第5回） 課題

発展課題① advanced_MonteCalro

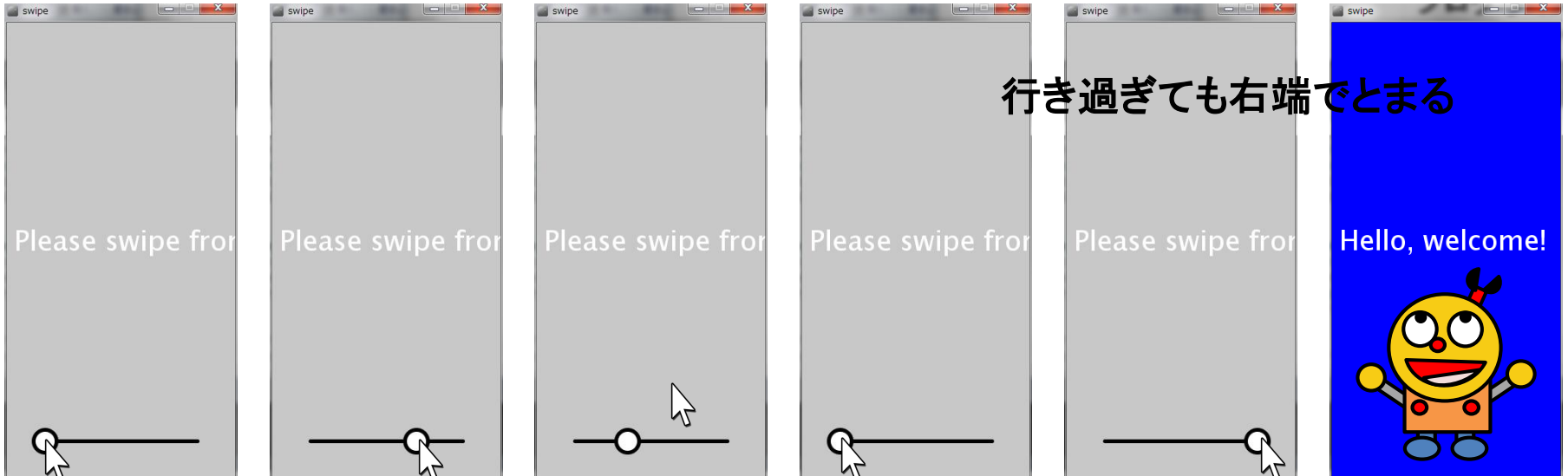
- モンテカルロ法とは，確率に関係のないものを確率を利用して計算するというものである．ここで円周率の近似値を求めたい．
- まず，800x800のウインドウ内に直径800ピクセルの円を描け
- 800x800の正方形の面積は800x800，直径800ピクセルの円の面積は800x800xπとなるため，円の面積/正方形の面積はπ/4となる
- draw()のたびに，画面内でランダムにX座標，Y座標を取得し，その点が円の外側の時には直径3ピクセルの青丸，内側の時には直径3ピクセルの赤丸を表示せよ
- 円内の点の数/全ての点の数 ≒ 円の面積/正方形の面積 ≒ π/4 を利用すると，π ≒ 4 * 円内の点の数 / 全ての点の数となる．これを利用して円周率を draw() を100回実施するたびに標準出力せよ





プログラミング演習I （第5回） 課題

- **発展課題②スケッチ名：advanced_Unlock**
  - 縦長のウインドウの下部に左の方に丸型のものを用意し、それを右側にスライド（スワイプ）させると、ロックが解除されキャラクタなどが描画されている画面へと遷移せよ（キャラクタじゃなく何か他のものでも良い）
    - 解除後にまたロック画面に戻らなくて良い
  - なお、右端まで移動せずに手を離した場合は、5ピクセル/フレームの速度で最初の位置に戻るようにせよ。また、指定の位置より右や左にはみ出ないようにせよ。



今日使うテクニック

① text()で表示する文字の大きさを変える方法

- 文字の大きさを変えるには　textSize(文字サイズ) を使う。

```
void setup() {
  size(300, 150);
}

void draw() {
  fill(0);
  textSize(50); // 文字の大きさを設定
  text( "Processing", 20, 90 ); // 文字を表示
}
```

- textSize()は、fill()やstroke()と同様に何回でもパラメータを変えて指定できるので、大きさの違う文字を混在させることができる。

今日使うテクニック

② text() で表示する文字の書体(フォント)を変える方法

- フォントを変えるには、PFont、createFont()、textFont() を使う。
- **日本語を使いたいときは日本語フォントの指定が必要**
- 以下はHGS創英角ポップ体で「Processing」と書く例

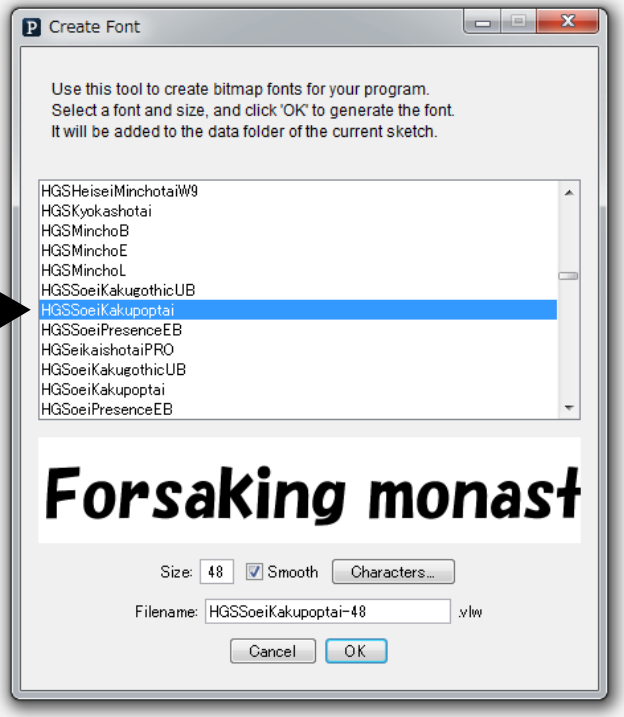
```
PFont myFont; // フォント

void setup() {
  size(300, 150);
  myFont = createFont("HGSSoeiKakupoptai",10);  // フォントを準備
  textFont(myFont);  // フォントを設定
  textSize(50);          // 文字サイズを改めて変更することもできる
}

void draw() {
  fill(0);
  text( "Processing", 20, 90 ); // 文字を表示
}
```

今日使うテクニック

- PFont はフォントを格納する変数につかうデータ型です。
  int や float などと同じような扱い。

- **createFont(フォント名, 文字サイズ)** でフォントを準備する。
  フォント名は、Processingのメニューの
  Tools -> Create Font...
  で出てくるパネルで確認できる。

**このリストにプログラム中で使える
フォント名が表示される。**



- 最後に、**textFont(フォント)** で
  フォントを設定する。