

プログラミング演習I (第7回) 課題

• 基本課題① スケッチ名: basic_Donut

- 中心座標と内径 (内直径)、外径 (外直径) を引数として与えると茶色(103, 67, 45)のドーナツを描く showDonut関数を作成せよ。
- また、400x400の大きさのウィンドウを背景白色で作成し、showDonut関数を利用してウィンドウの中央に、**内径が150ピクセル (半径75ピクセル)、外径が300ピクセル (半径150ピクセル)** 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、true、それ以外をクリックするとfalseを返す関数 isDonut関数を作成せよ。
- isDonut関数は、引数を中心座標と内径 (内直径)、外径 (外直径)、マウスのX座標、Y座標とせよ。
- またその関数を使って、ドーナツ部分をクリックしたとき (isDonutがtrueのとき) Donut!と標準出力するプログラムを作成せよ (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7回) 課題

• 基本課題② basic_Janken

- 300x300のウィンドウを作成し, (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.
- いずれかの円をユーザがクリックしたとき, 左ならグー(0)、中ならチョキ(1), 右ならパー(2)の手を選んだものとする.
- ユーザが手を選ぶと, コンピュータもランダムにグー(0)、チョキ(1)、パー(2)を選ぶ
- ユーザの手(0, 1, 2)とコンピュータの手(0, 1, 2)を引数として, ユーザがかった場合は1、引き分けだと0、負けた場合は-1を返す関数judgeJankenを作成せよ.
- 以上の仕組みと関数を利用し、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!

あなたはパー
コンピュータはパー
引き分け

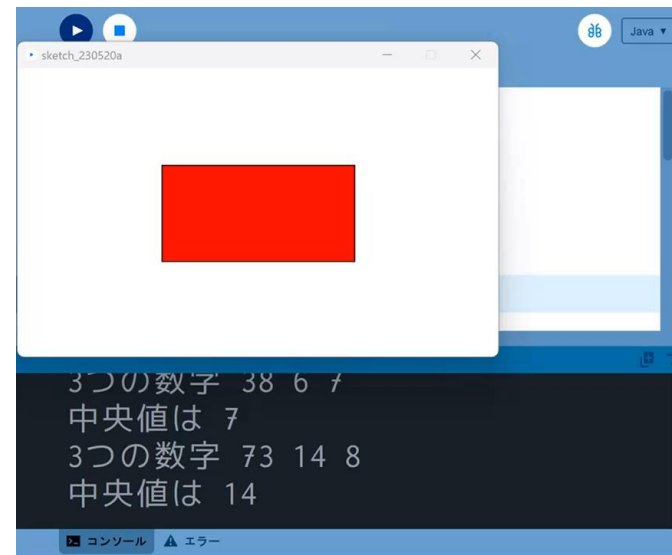
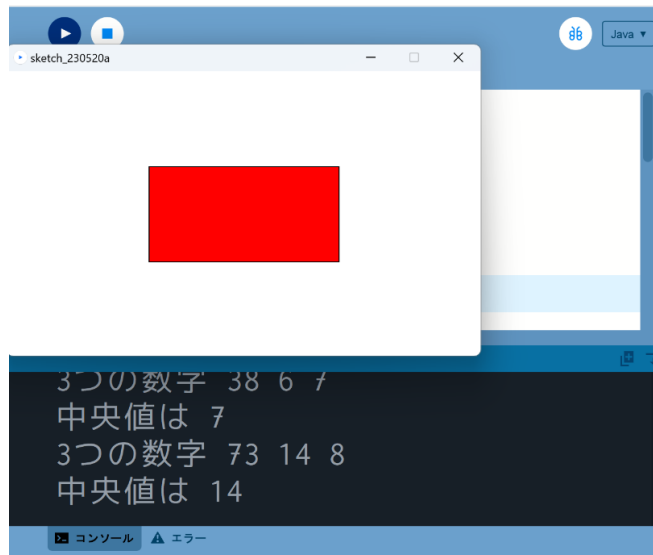
クリック!

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7回) 課題

• 基本課題③ スケッチ名：basic_Middle

- 500x300の白色背景のウィンドウ内に、(150, 100)を左上座標とし、横幅200、縦幅100の赤色の四角形を描画せよ。
- また、その赤色の四角形内をクリックしたときに、1~100までの整数値で3つの数値をランダムに生成し、標準出力に出力せよ。
 - (int)random(1, 101) で1~100までの整数は取得できるよ！
- さらに、3つの整数値を引数として、その中央値を求めて返す関数getMiddleを作成し、その関数を利用して中央値も標準出力せよ。
 - 中央値は、真ん中の値



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か？
 - ドーナツ部分は外径より小さく内径より大きいとき

- 基本課題②

- どういう条件になるか、手書きしてみよう
 - ユーザがグー(0)でコンピュータがチョキ(1)はユーザの勝ち
 - ユーザがグー(0)でコンピュータがパー(2)はユーザの負け
 - ユーザがグー(0)でコンピュータがパー(2)はユーザの負け
- mousePressedの中でユーザとコンピュータの手を取得して、judgeJanken関数の結果を利用して標準出力しよう

- 基本課題③

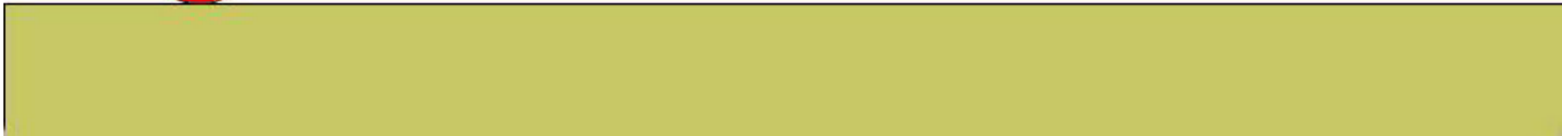
- 四角系の条件判定は？
- 3つの数字の中央値はどのような特徴をもつ？

プログラミング演習I (第5回) 課題

• 発展課題① advanced_Action

- 800x400のウィンドウを作成し、画面下部に地面のような領域を描画せよ。また、円(または自身のキャラクタ)をX座標を400、Y座標は地面に接するように初期配置し、静止させておけ。
- キーボードの左キーで左方向への速度を+1し、右キーで右方向への速度を+1せよ。例えば、初期状態から右キーを2回押すと、X座標の正の方向に1フレームあたり2ピクセル動くようになる。また、その状態で左キーを2回押すと静止する。さらに、その状態で左キーを1回押すと1フレームあたり1ピクセル左に動くようになる。
- さらに、キーボードの上キーで上方向にジャンプするようにせよ。上方向への速度は毎回同じで良い(最高到達点はおなじになる)
- 画面右端、左端まで来ると、逆側から登場するようにせよ。
- 動きはこんな感じ：
<https://gyazo.com/bce8324df8086995dd42f1cdfbbd0563>

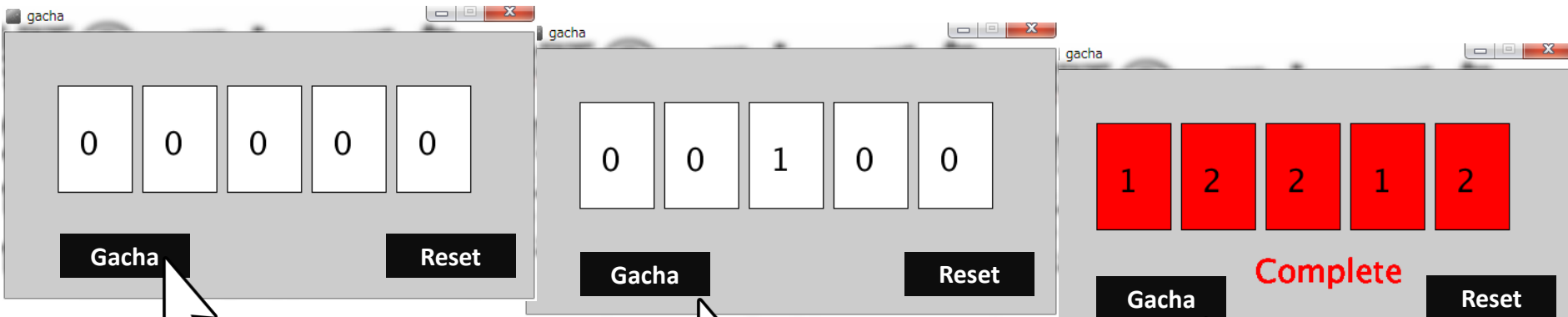
▶ sketch_220521a



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_CompleteGacha`

- ウィンドウ下部のGacha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、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
- また、すべてのカードが1枚以上になっているかどうかを判定する `isComplete` 関数 (すべてのカードが1枚以上の場合は `true`、そうでない場合は `false` を返す。引数はあってもなくてもよい) を作成せし、その関数を利用してすべてのカードが1枚以上の場合に `Complete` とウィンドウ内に `text` を用いて表示し、カードの色を赤色にせよ
- なお、どの状態でもウィンドウ下部の `Reset` ボタンを押すとすべてのカードの枚数を0枚にせよ (すでに `Complete` している場合もすべてを0枚に戻し、カードの色も元通りにせよ)



今日使うテクニック

text() で表示する場所をわかりやすくする方法

- text("出力する文字列", x座標, y座標); の、x座標とy座標が微妙に扱いにくい。
- textAlign(CENTER, CENTER); と書いて、縦横中央揃えしておくとうわかりやすくて良いよ！
 - LEFT/RIGHT, TOP/BOTTOMも設定できるよ！

```
void setup() {  
  size(300, 150);  
  textAlign(CENTER, CENTER);  
}  
  
void draw() {  
  text( "Processing", 150, 75 ); // 中央に表示されるよ！  
}
```

