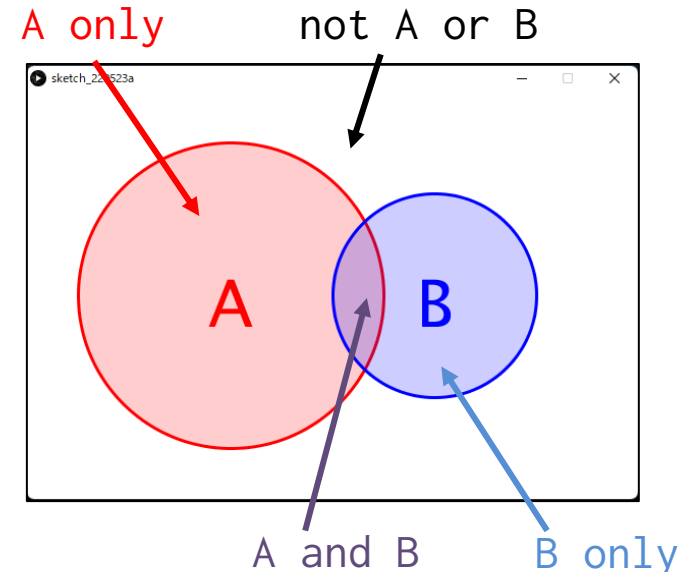
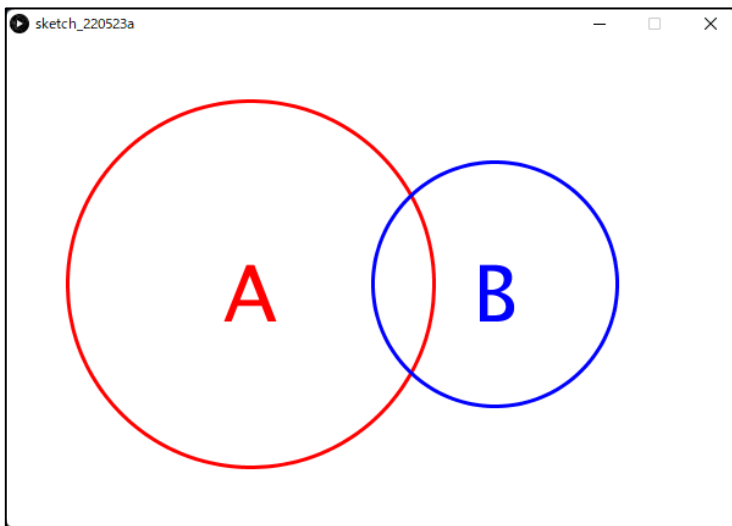


プログラミング演習I (第7回) 課題

基本課題① basic_VennDiagram

- 600x400の大きさのウィンドウの背景を白色にし、ウィンドウ内に、中心が(200, 200)で直径が300ピクセルの赤枠の円、中心が(400, 200)で直径が200ピクセルの青枠の円を描け。また、それぞれの円の中に「A」「B」と表記せよ。Aは赤文字、Bは青文字とせよ。
- isAという関数と、isBという関数を作成せよ。isA、isBはマウスのx, y座標を引数としてとり、それぞれA内、B内であればtrueを、そうでなければfalseを返すようにせよ。引数が整数2つで、戻り値が真偽値。
- isAとisBを利用し、ベン図のクリック場所に応じて「A only」「B only」「A and B」「not A or B」と標準出力するようにせよ。

画面の出力例



プログラミング演習I (第7回) 課題

• 基本課題② basic_Janken

- 300x300の背景白色のウィンドウを作成し、x座標が100と200の位置にそれぞれ黒色の縦線を描画せよ。
- ウィンドウ内をユーザがクリックした時、線で区切られた3つの領域の左ならグー(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。引数が整数2つで、返り値が整数。
- 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。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(0)
コンピュータはチョキ(1)
あなたの勝ち

クリック!

あなたはパー(2)
コンピュータはパー(2)
引き分け

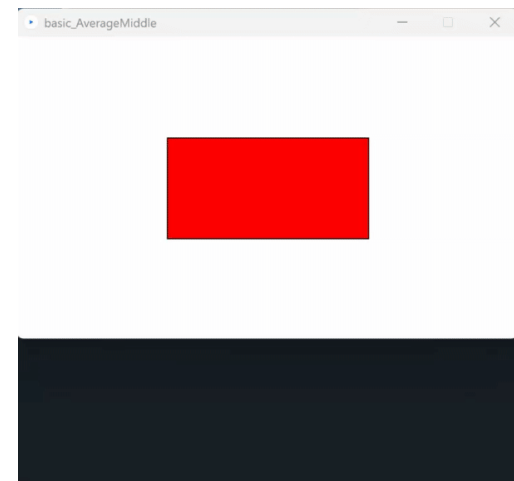
クリック!

あなたはグー(0)
コンピュータはパー(2)
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_AverageMiddle`

- 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、1~10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, 11)` で1~10までの整数は取得可能
- さらに、3つの整数値を引数として、その平均値を実数値として求めて返す関数`getAverage`と、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平均値と中央値も標準出力せよ。
- 中央値は、真ん中の値 (3つの数字で真ん中の値は、最大値でも最小値でもないもの)



ヒント

- 基本課題①

- isAの条件はどうなる？
- isBの条件はどうなる？
- isAとisBの組み合わせで標準出力を設定しよう！

- 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

- 四角形の条件判定は？
- 3つの数字の平均値はどう計算する？（実数であることに注意！）
- 3つの数字の中央値はどういう特徴をもつ？

プログラミング演習I (第7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、キャラクタが地面に接しているときに限り、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)
 - ジャンプ中に上キーを押しても反応しないようにするという意味
- 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>

プログラミング演習I (第7回) 課題

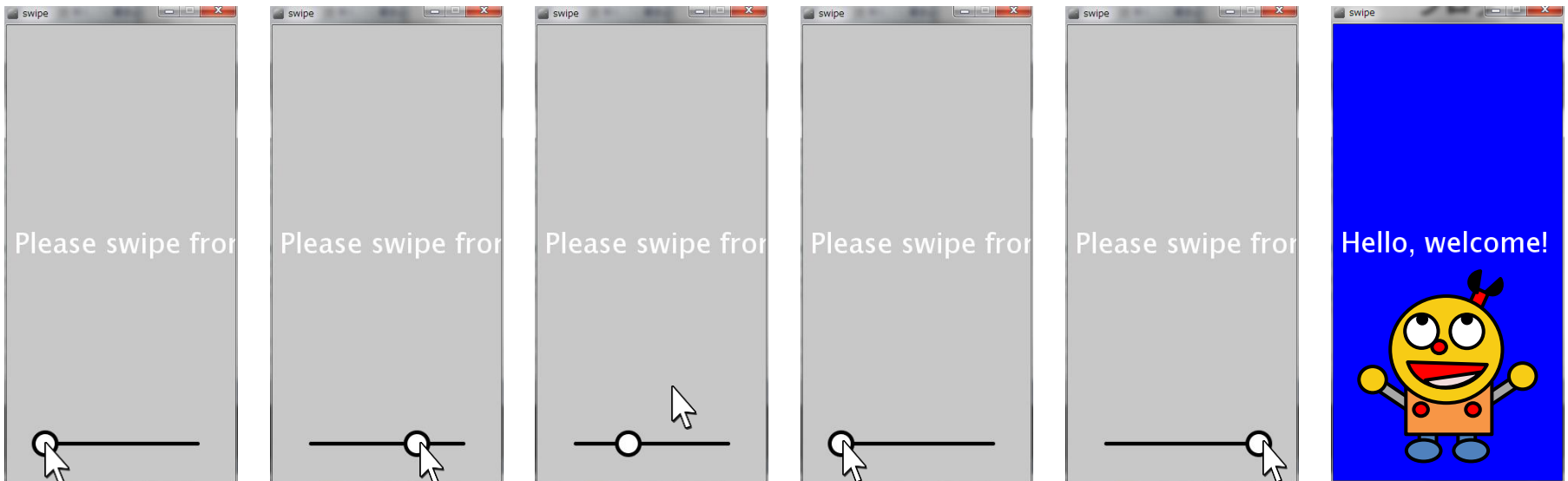
▶ sketch_220521a



プログラミング演習I (第7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、ロックが解除された画面へと遷移するプログラムを作成せよ。ロック画面とアンロック画面とで画面上に表示するものを切り替える(表示する文字列を切り替える、背景色を変更する、キャラクターを表示するなど)ことで、ロックとアンロックを表現せよ。
- 解除後にロック画面に戻らなくて良い。また、アンロック画面ではスライドさせるUIを消すこと。
- なお、丸をUIの右端まで移動せずにマウスのボタンを離した場合は、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。**行き過ぎても右端でとまる**



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！  
}
```

