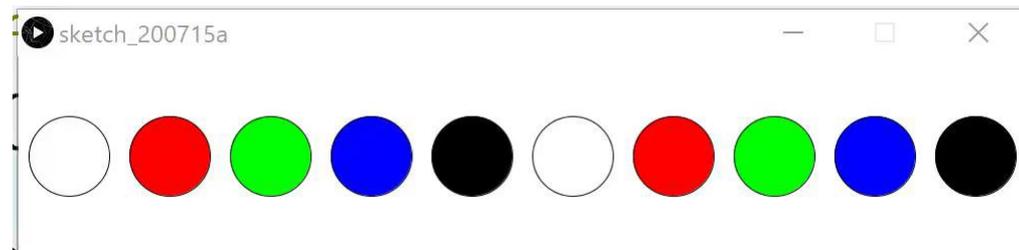
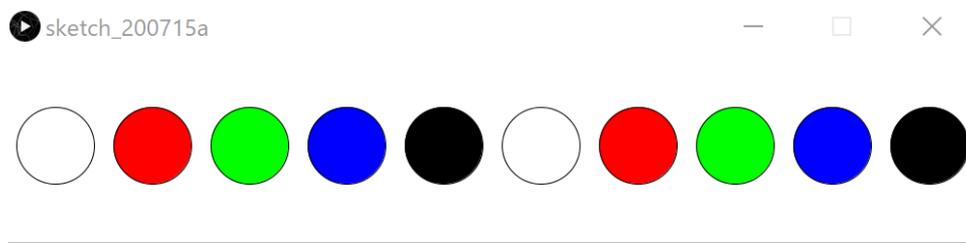


プログラミング演習I (第8回) 課題

• 基本① スケッチ名: basic_LineBoard

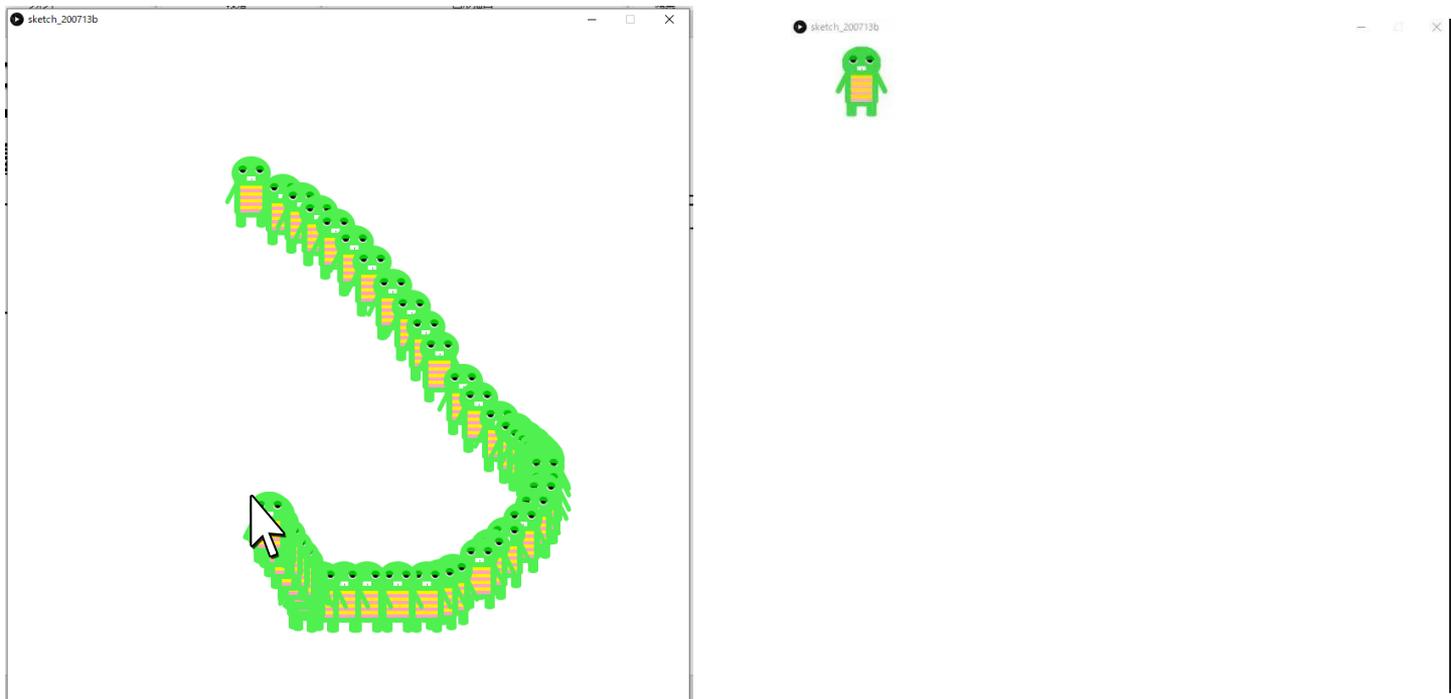
- 1000x200のウィンドウを作成し、直径80の円を横に等間隔に10個並べよ (円の中心のY座標は100とせよ)。また 円の内部 をクリックする度に、そのクリックされた円の色が【白→赤→緑→青→黒→白 (以後ループ)】と変化させるようにせよ。なお、起動したときの円の色は下記のようになるようにせよ。



プログラミング演習I (第8回) 課題

• 基本②スケッチ名 : basic_MouseTrace

- 800x800のウィンドウを作成し, そのウィンドウの中でマウスカーソルを動かすと, マウスカーソルを追尾する50個のキャラクタを描画せよ
- ただし, マウスを追尾しているキャラクタのうち, マウスに近いもの(新しいもの)を, 手前に表示するようにせよ
- 最初に左上にキャラクタが集まっててもよい



キャラクタを小さくする

```
void drawCharacter(int cx, int cy)
{
    pushMatrix();
    // 座標移動
    translate(cx, cy);
    // 縮小 ( 以下の場合には0.2倍にする )
    scale(0.2);

    // ここにキャラクタの描画が来る

    popMatrix();
}
```

プログラミング演習I (第8回) 課題

• 基本②スケッチ名 : basic_MouseTrace

- 800x800のウィンドウを作成し, そのウィンドウの中でマウスカーソルを動かすと, マウスカーソルを追尾する256個の丸を描画せよ
- ただし, マウスを追尾している丸のうち, マウスに近いもの(新しいもの)のを手前に表示するようにせよ. また, 255フレーム前に作られた丸は(255, 255, 255)となるようにし, 一番新しい丸は(255, 0, 0)となるようにせよ
- 最初に左上に丸が集まっててもよい



プログラミング演習I (第8回) 課題

• 基本③ スケッチ名：**basic_CompleteGacha**

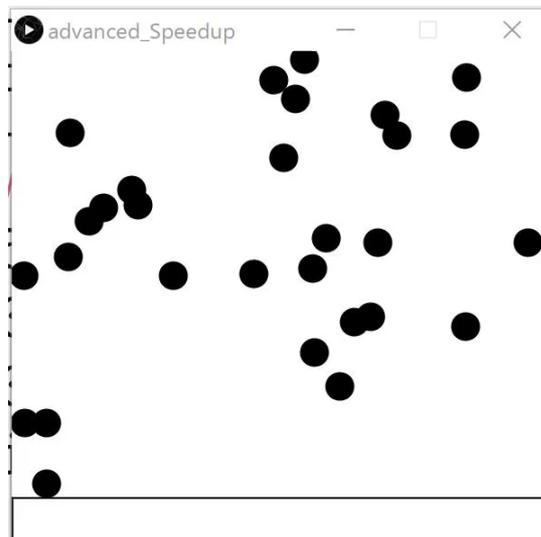
- 30個のアイテムがあり、クリックするたびに重複含め10個のアイテムをランダムに取得できるガチャがある。ここで取得したアイテムの数をカウントし、その数を表示するプログラムを棒グラフとして作成せよ。ただし、配列を用いて実現せよ
- また、30個のアイテム全てが1個以上揃った時にCompleteと表示するようにせよ
- なお、Complete以降は増やせるようにしても、増やせないようにしてもどちらでもよい



プログラミング演習I (第8回) 課題

• 発展① スケッチ名: advanced_Speedup

- 縦横600x550のウィンドウ内の、600x500の領域にX座標Y座標ともにランダムに配置された30個の直径20の円を描き、またそれぞれの円のX、Y両方向の基本速度を-0.1~0.1の間でランダムに設定せよ
- また、画面の下に600x50ピクセルのスライダ(縦線)を表示し、スライダ内をマウスクリックすることで、クリックされたX座標にスライダの場所を指定できるようにせよ
- 円はdrawのたびに移動するようにし、その速度は、スライダの位置のX座標に基づき、X倍するようにせよ。なお、円は両端で跳ね返るようにせよ。
- 円の色もスライダの位置に応じて左端なら(0,0,0)、右端なら(255,0,0)となるように変更せよ。



プログラミング演習I (第8回) 課題

• 発展② スケッチ名: advanced_Eratosthenes

- 下記に示すエラトステネスの篩のアルゴリズムに従い, 配列を利用して2~100万までの素数を高速に求め, 2~100万までの素数の個数を標準出力せよ (答えは78498個)
- また, 計算を開始してから終了するまでの時間をミリ秒単位で計算し, 出力せよ. この計算時間は100ミリ秒以内とすること.
- <https://ja.wikipedia.org/wiki/エラトステネスの篩>
- さらに, エラトステネスの篩ではなく自身のこれまでのやり方でも2~100万までの素数を求め, その個数を標準出力するとともに, 計算時間をミリ秒単位で出力せよ
- プログラムの出力例は下記のように, どちらがどちらなのかをわかりやすく表示するようにせよ

自分で組んだプログラム
2から100万までの素数の数は78498個あります
計算にかかった時間は  ミリ秒です

エラトステネスの篩で組んだプログラム
2から100万までの素数の数は78498個あります
計算にかかった時間は  ミリ秒です

今日使うテクニック

millis()でミリ秒単位の経過時間を取得する

- アプリケーションが起動されてからの時間は millis() で取得することが可能

```
int iStartMillis;
boolean bFlagStart = false; // スタートしたかどうかのフラグ
void setup() {
    size(300, 150);
    fill( 0 ); // 文字色を黒色に設定
}
void draw() {
    background( 255 );
    if( bFlagStart ){ // スタートしていたら～
        text( millis()-iStartMillis, 20, 90 ); // 差分で経過時間を表示
    }
}
void mousePressed(){
    bFlagStart = true; // クリックされたらスタートフラグを立てる
    iStartMillis = millis(); // スタートの経過時間をセット
}
```