

課題の提出方法

- プログラムの冒頭にコメントを必ず書くこと！
 - プログラム名とその目的（日本語でも英語でもOK）
 - プログラムの作成日（例： 2021/05/10）
 - プログラムの作成者（年-組-番号-名前）
 - 追加で他の情報を記載してもOK

```
// Olympic.pde
// Draw the Olympic rings
// 2021/05/10
// 1-3-99 Satoshi Nakamura
background( 255, 255, 255 );
fill( 255, 0, 0 );
:
```

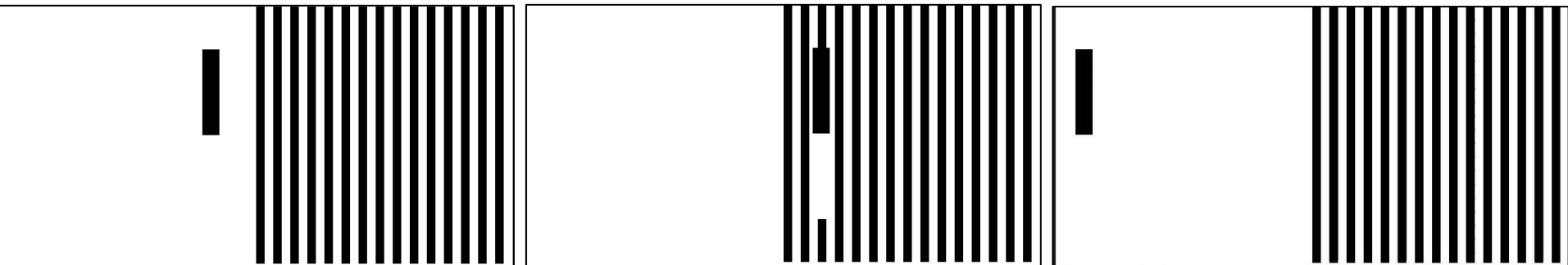
修正：課題の評価について

- 基本課題は、講義中の指定時間までに提出必須
 - －それ以降の提出は採点対象外となります
- 発展課題は、講義中の指定時間までに提出したものを100%として、指定時間以降、次の講義開始までに提出されたものを50%として評価
 - －なので、発展課題ができていなくても、次回までに提出したら50%分は点が付きます！
 - －他人に見せない！見せてもらわないように！

プログラミング演習(3) 課題

• 基本課題① スケッチ名: **basic_FootStep**

- 横600x縦300のウィンドウを作成せよ
- プログラムの画面右半分（x座標が300のところから）横幅10ピクセルの黒色の四角形を間を10ピクセル飛ばして画面右端まで描画するようにせよ（わかる人はfor/whileを使ってもよい）
- また、画面の左端からy座標が50ピクセルの位置に、縦100ピクセル、横20ピクセルの黒色の四角形を描画し、同じくy座標が150ピクセルの位置に、縦100ピクセル、横20ピクセルの白色の四角形を描画し、その2つの四角形をdrawのたびに1ピクセルずつ右へと移動させよ。なお、右端まで来ると左端から登場させよ
- 注意：四角形に枠線は付けないようにせよ！



プログラミング演習I (3) 課題

• 基本課題② basic_LissajousCurve

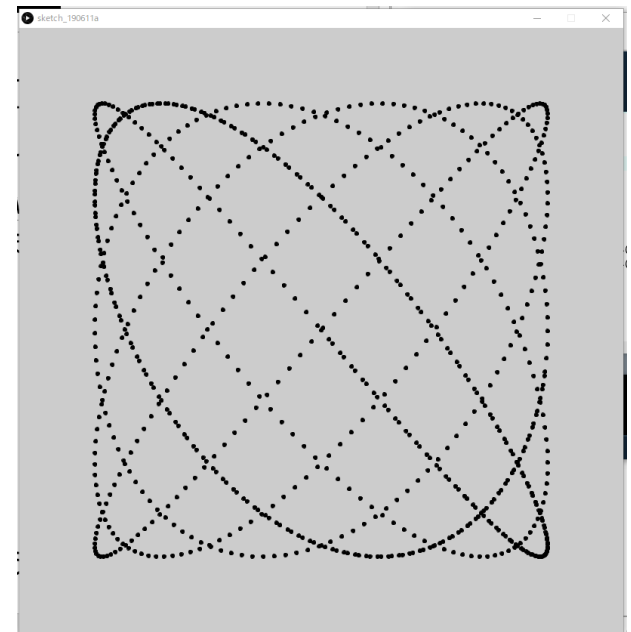
– xとyの座標がtによって変化する下記の数式の計算結果の座標に直径5ピクセルの黒色の円を描画せよ。ただし, tは draw() 毎に0から0.01ずつ増加するようにせよ。またウィンドウサイズは800x800とせよ。

- $x = 300 \sin(5t) + 400$

- $y = 300 \sin(6t) + 400$

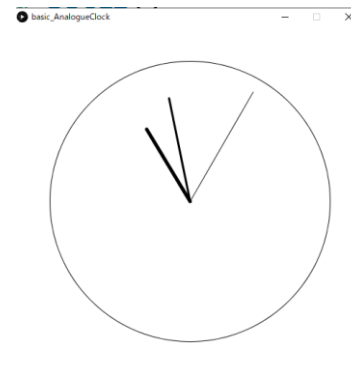
– 5と6の値を変えると他の図形になるよ！

– 出来た人は遊んでみましょう



プログラミング演習(3) 課題

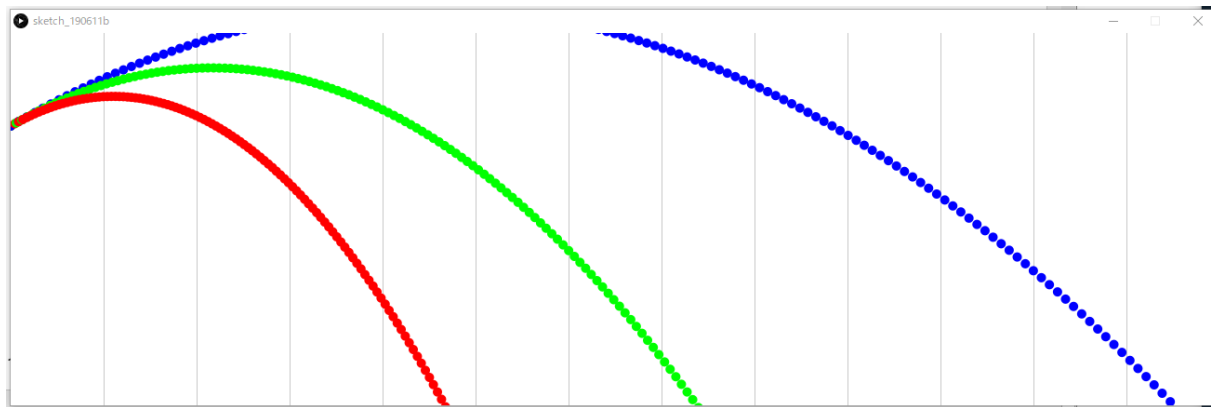
- 基本課題③ スケッチ名: **basic_AnalogueClock**
 - 縦横500x500のウィンドウを作成せよ
 - (250,250)を中心に直径400ピクセルの円を描け
 - drawの中でローカル変数として定義されている現在の時間h、分m、秒sという3つの変数を利用して、長針短針秒針のある時計を描画せよ（長さや太さで表現せよ）
 - 装飾をこだわってもよいが、時間はできるだけ正確に描画するようにせよ（短針が1時間に1回、長針が1分に1回動くものでも良いが、できれば徐々に動くようにすると良い）



プログラミング演習(3) 課題

・ 発展課題① スケッチ名：advanced_ThrowBalls

- 建物の屋上（300メートルの高さ）から斜め上30度方向にボールを投げ、落下するときの様子をシミュレーションするプログラムを作成してください（ウィンドウサイズは1300x400とせよ）。
- 3種類の投射速度（50m/s、70m/s、100m/s）で同時に投げたときの比較結果を下図のように軌跡が残るようにして示せ。
 - 50m/sを赤，70m/sを緑，100m/sを青色とせよ（輪郭の色は自由とする）
- 距離を把握しやすくするため，100mごとに灰色の縦線を描画せよ



ヒント： 斜方投射の式

$$x = v_0 \cos(\theta) t$$

$$y = v_0 \sin(\theta) t - \frac{1}{2} g t^2$$

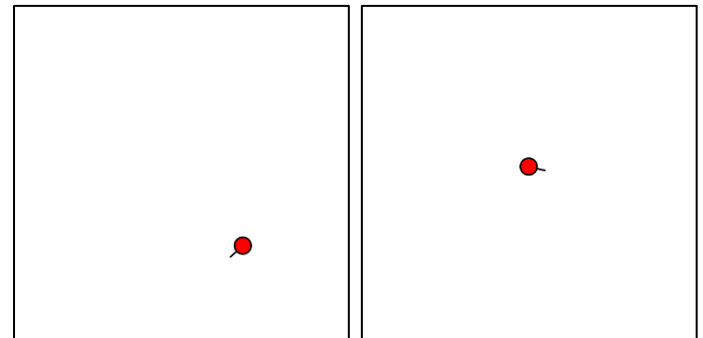
$$g = 9.8$$

※1フレームの経過時間は0.1秒とせよ。また、1ピクセル=1mと考えてください。

プログラミング演習(3) 課題

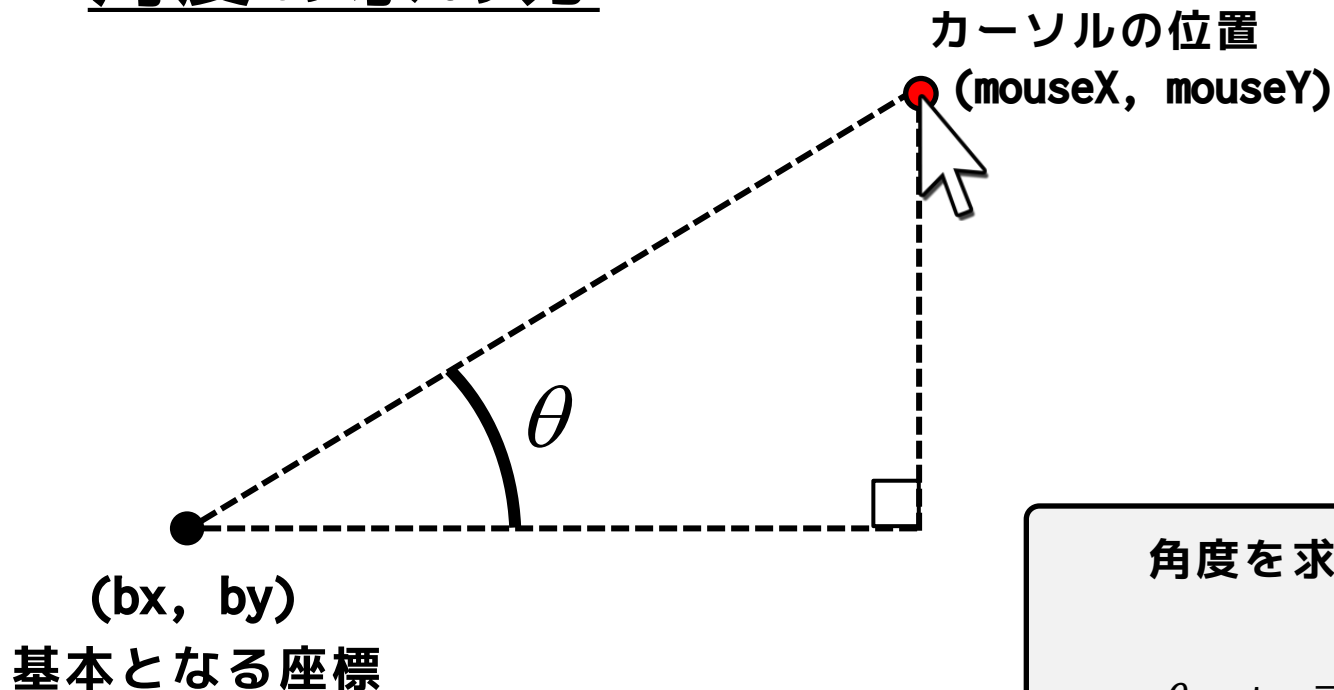
• 発展課題② スケッチ名：advanced_ChaseBall

- 800x800のウィンドウ内で、マウスカーソルを追いかけるオブジェクト（赤色円とカーソルの方向を向く棒線からなる）を描画するプログラムを作れ
- ここでは、まずオブジェクトを画面中央に配置せよ
- また、オブジェクトの速度は、「現在のオブジェクトの位置からマウスカーソルの位置までの距離の5分の1」とせよ（速度は距離が遠いと速く、近いと遅くなる）
- オブジェクトの進行方向は、現在のオブジェクトの位置からマウスカーソルの位置に直接向かう方向とせよ
- なお、角度については次ページの方法で `atan2` を利用することで求めることが可能である



プログラミング演習(3) 課題

• 角度の求め方



角度を求めるための数式

$$\theta = \tan^{-1} \left(\frac{mouseY - by}{mouseX - bx} \right)$$



```
theta = atan2(mouseY-by, mouseX-bx);
```