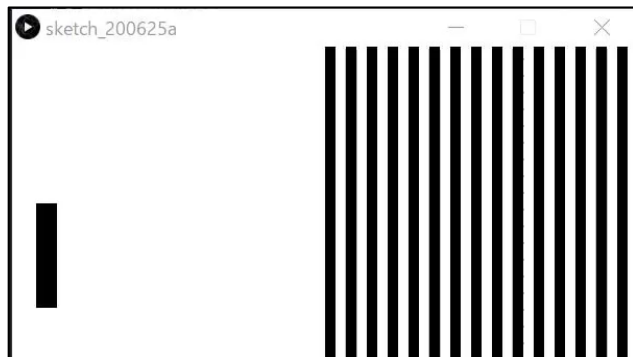


プログラミング演習(3) 課題

• 基本課題① 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）に、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）
- 画面の左端からy座標が50ピクセルの位置に、横20ピクセル、縦100ピクセルの白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、横20ピクセル、縦100ピクセルの黒色の四角形を描画し、その2つの四角形をdrawのたびに1ピクセルずつ右へ移動させよ。
- 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(3) 課題

• 基本課題②アルキメデスの螺旋 : basic_Spiral

- 800x800のウィンドウを作成し、xとyの値が θ (theta) 【度】によって変化する、下記の数式の計算結果の座標(x, y)を中心に直径3ピクセルの円 (黒色で塗りつぶす)を描け
- ただし、thetaはdraw()毎に2ずつ増加するようにせよ (採点が面倒なので、frameRateはいじらないこと)
- thetaをcos、sinの中で使う場合はradiansで度からラジアンに変換すること

$$x = a\theta \cos\theta + b$$

$$y = a\theta \sin\theta + c$$

$a=0.1$, $b=400$, $c=400$ とする

- frameCountを使っても良い

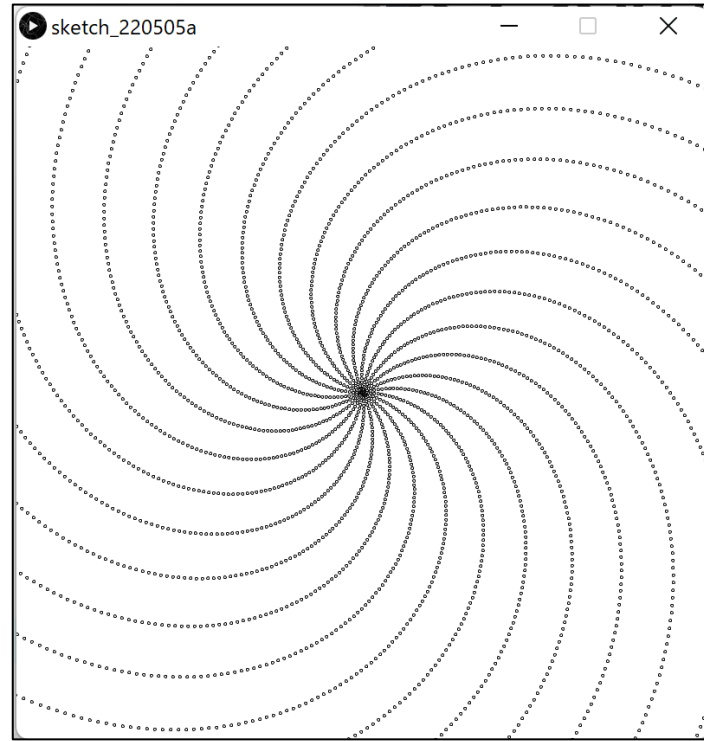
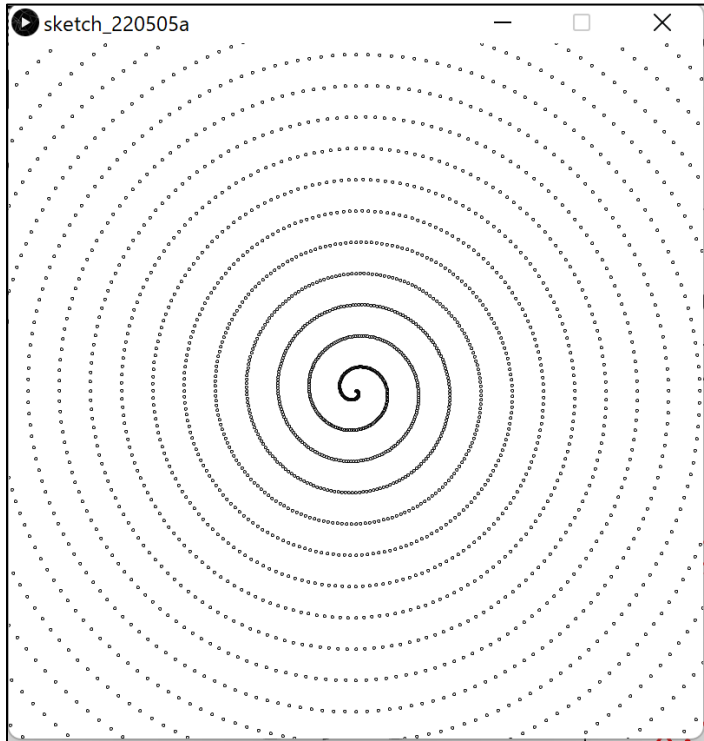
```
int theta = 0;

void setup(){
    初期化処理
}

void draw(){
    x,yを定義
    x,yの計算
    circle(x, y, 3);
    theta = theta + 2;
}
```

プログラミング演習(3) 課題

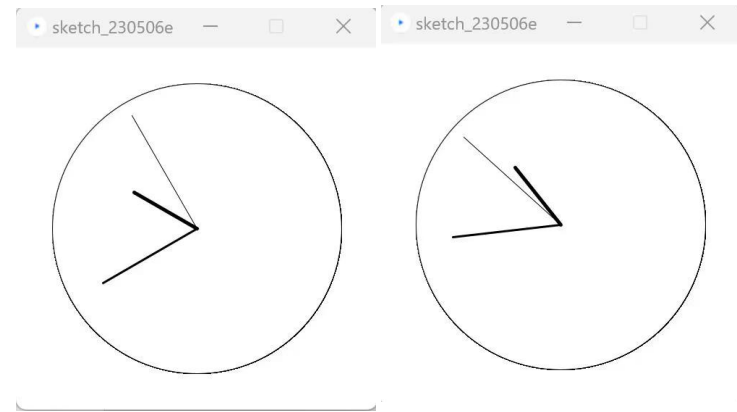
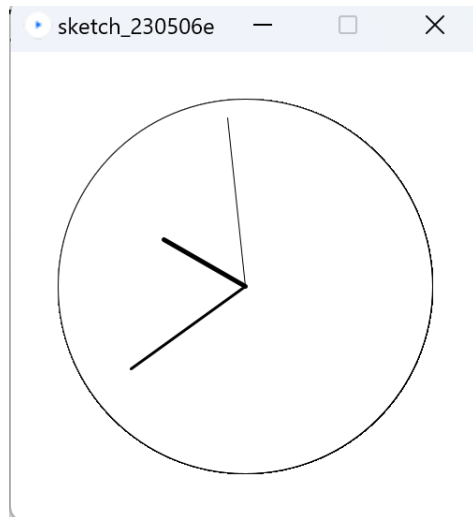
- アルキメデスの螺旋（左）、 \sin や \cos の中で使う θ をradiansでラジアン単位に変換しないと右のようになるよ（これは不正解）
- ただ、こうしたものもかっこいいので課題を提出したら、色々値を変えたりしながら楽しんでみよう！



プログラミング演習(3) 課題

• 基本課題③ スケッチ名: **basic_Clock**

- 縦横500x500のウィンドウを作成せよ
- ウィンドウの中央から、直径400ピクセルの円を描け
- 時間 hour()、分 minute()、秒 second() を取得する関数を利用し、12時間で1回転する時針(長さ100・太さ5)、60分で1回転する分針(長さ150・太さ3)、60秒で1回転する秒針(長さ180・太さ1)を直線で描け。
- 時針は1時間に1回、分針は1分に1回動くようにしてもよいし、時間の経過に合わせて少しずつ動くようにしても良い

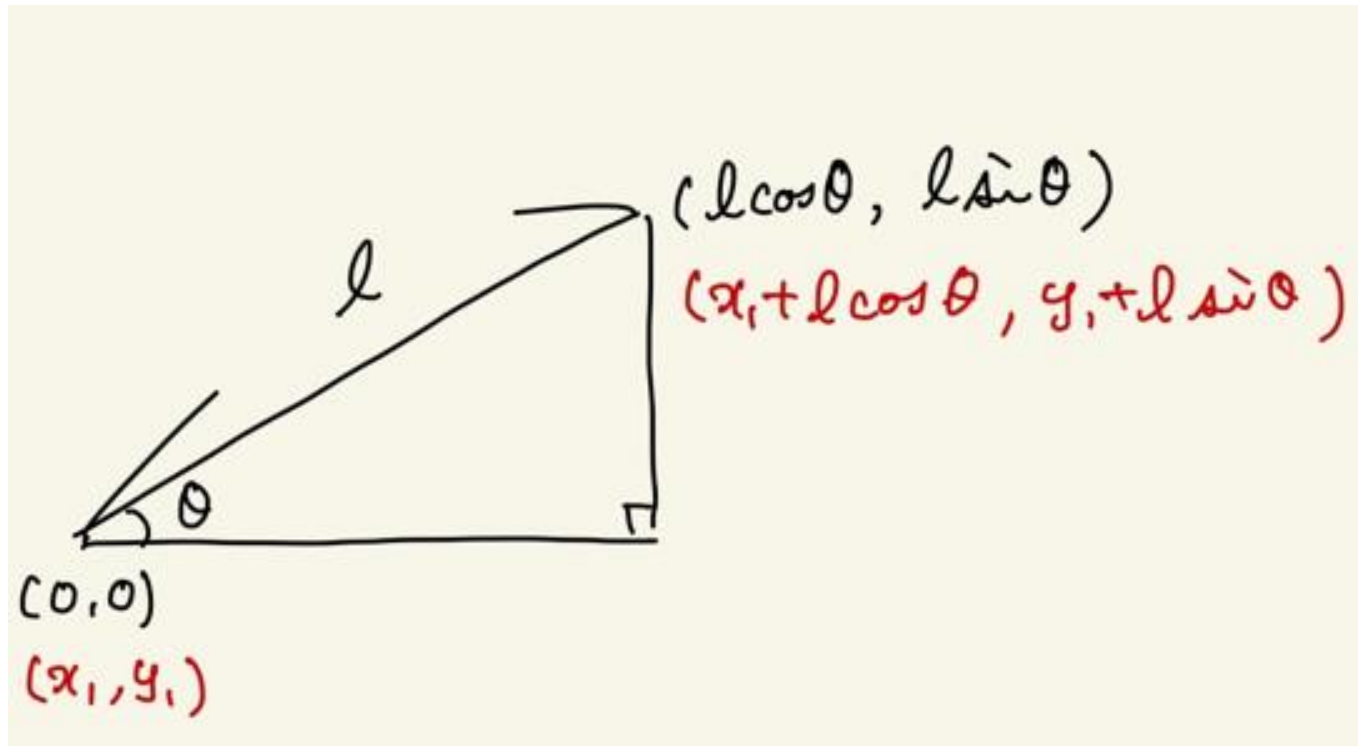


ヒント

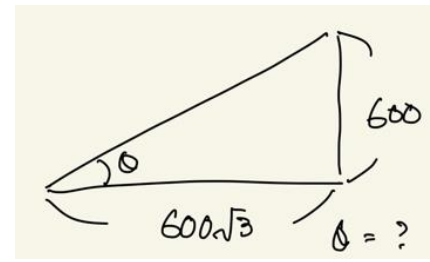
- 時針は12時間で1周（360度）する
 - つまり、1時間は角度としては 30 度分
 - 例えば10時は0時から角度が $30 * 10$ 度
 - cosとsinはラジアンを使うので、10時の角度は $(30 * 10 / 180) * \text{PI}$ となる
(radiansを使う場合は $\text{radians}(30 * 10)$ でOK!)
 - 時針は真上から始まるけれど、Processingの中での角度は右からスタートなので、90度調整すると $((30 * 10 - 90) / 180) * \text{PI}$ になる
($\text{radians}(30 * 10 - 90)$ でもOK)
 - 時針は長さ100ピクセルでウィンドウ中央(250, 250)から、上記の角度を使った下記の座標まで線を引くことに
 - $x = 250 + 100 * \cos(((30 * 10 - 90) / 180) * \text{PI})$
 - $y = 250 + 100 * \sin(((30 * 10 - 90) / 180) * \text{PI})$

補足

- ある点から、角度 θ 方向に、距離 l だけ離れた点の座標は？

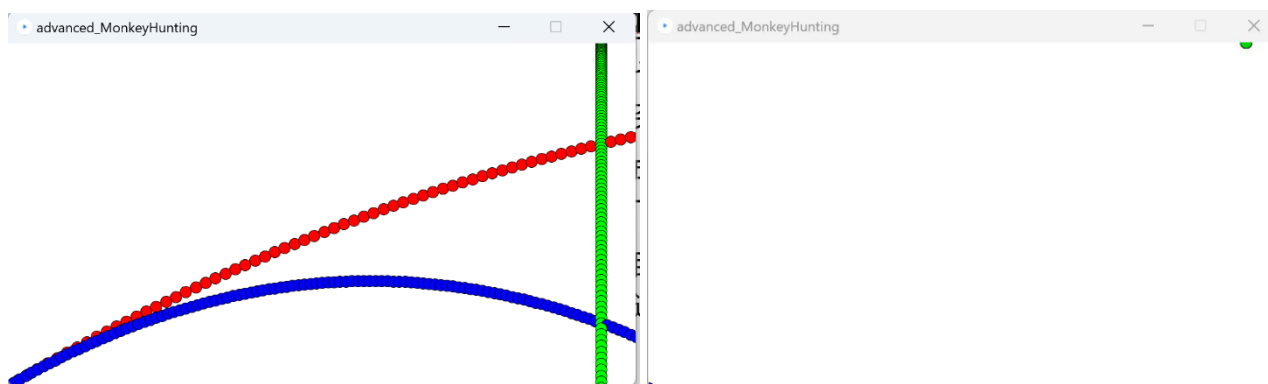


プログラミング演習(3) 課題



発展課題① advanced_MonkeyHunting

- モンキーハンティングという物理現象を確認するため、その動きを軌跡として残しつつ、衝突を確認するプログラムを作成せよ
- 1100x600のウィンドウを作成し、x座標 $600\sqrt{3}$ 、ウィンドウの上端から緑色のボールを自由落下させる
- また、ウィンドウ左下から自由落下させる緑色のボールに照準を合わせ、赤色のボールは初速200m/s、青色のボールは初速120m/sで投射する
- いずれも直径20ピクセルのボールとし、空気抵抗は無視せよ



ヒント： 斜方投射の式

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

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

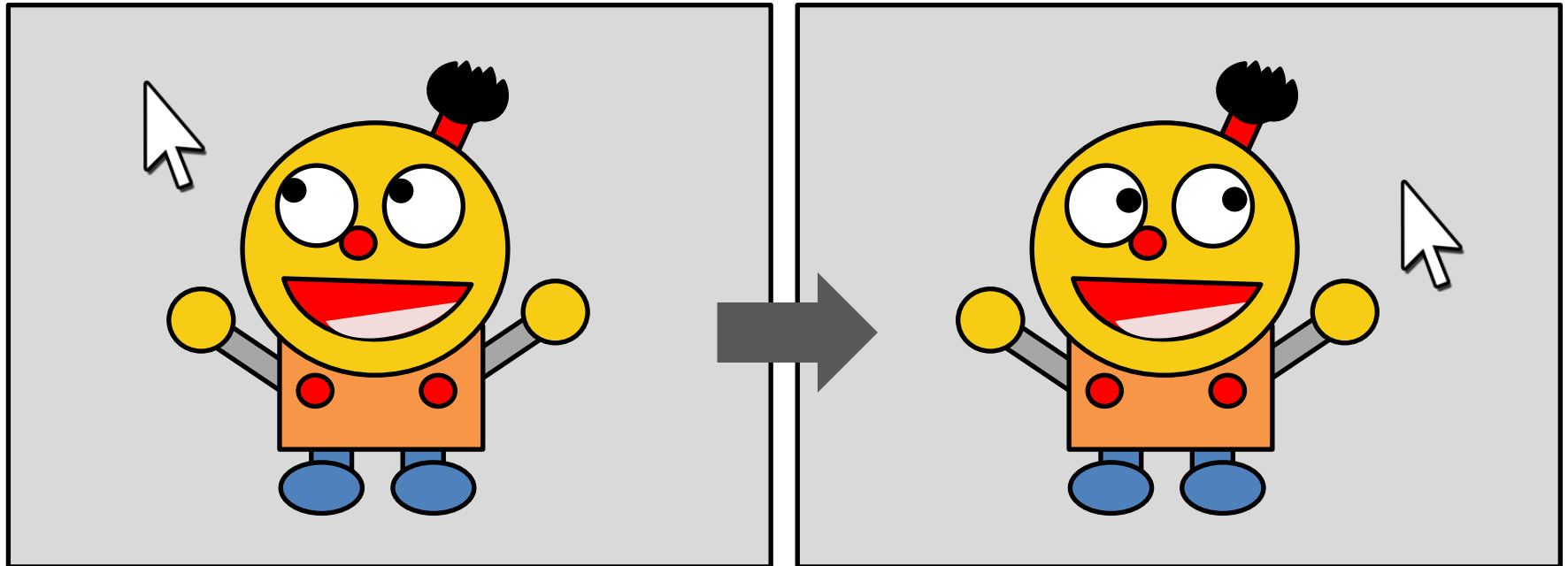
$$g = 9.8$$

※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 advanced_MoveEyes

- キャラクタを描くプログラムを改良し、キャラクタの目（黒目がある場合は黒目）がマウスマウスカーソルのある方向を常に追いかけるプログラムを作成せよ。なお黒目がない場合は、目を追加してもよい。少なくとも1つの目が動けば正解としますが、ぜひ全部の目を動かしてみてください（白目部分はなくともOKです）

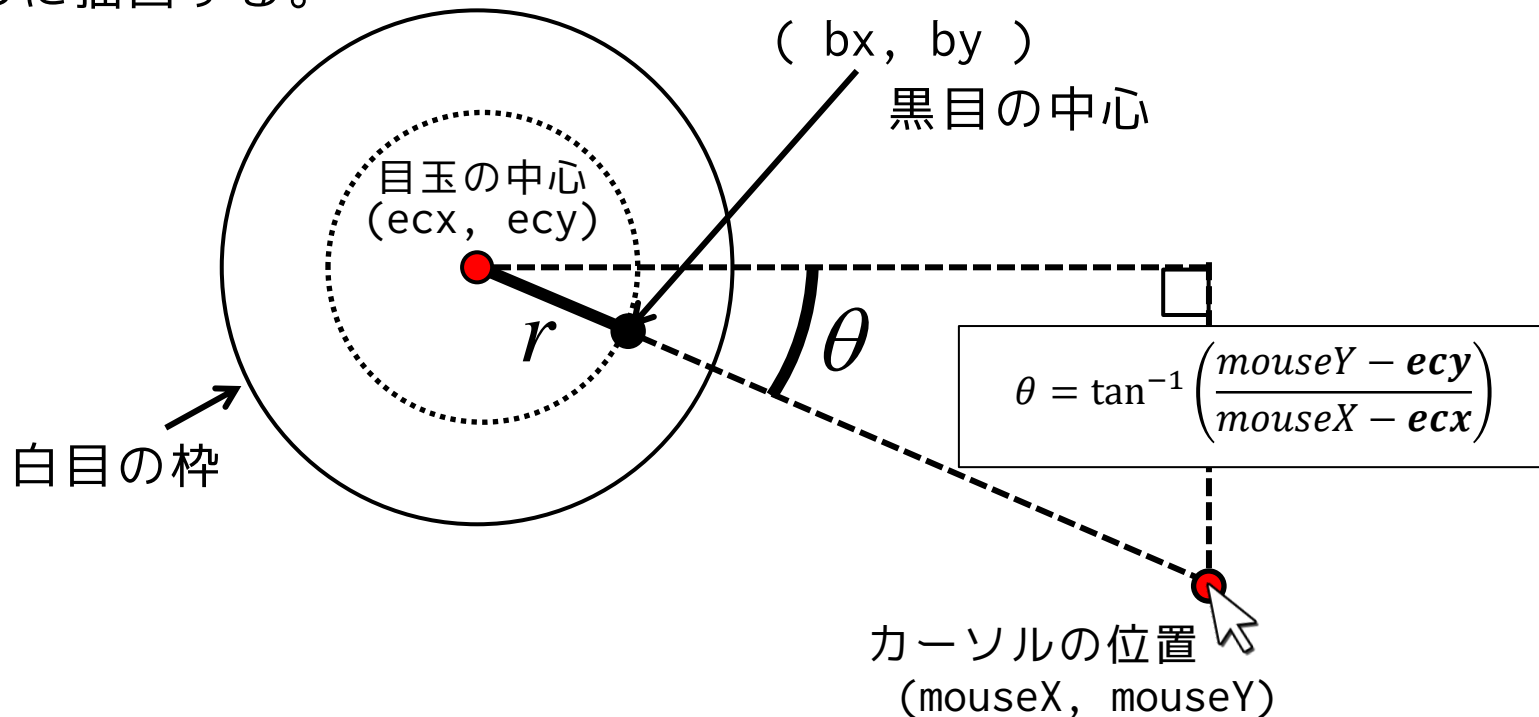


※例えば目玉が楕円形の場合、黒目の動きは楕円軌道にこだわらなくてもよいです。

プログラミング演習(3) 課題

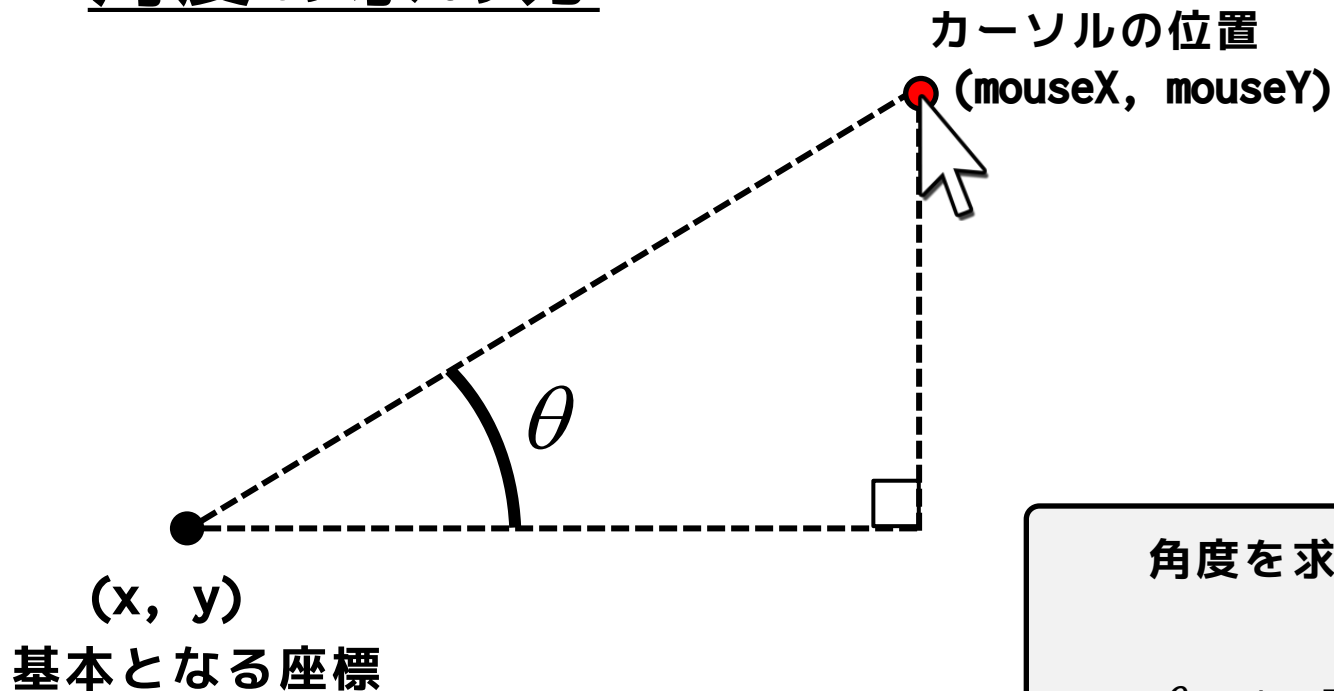
目玉と黒目がある場合の考え方 (1つの目に関する解説)

- まず、目玉の中心(ecx , ecy)を中心として、黒目(bx , by)を半径 r の円周上で動かすと考える。
 - ecx や ecy 、 r については、実際のキャラクタに基づき値を決め打ちする
- 黒目は、目玉の中心と、マウスの座標とを結ぶ線と、半径 r の円が交わるところに描画する。



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

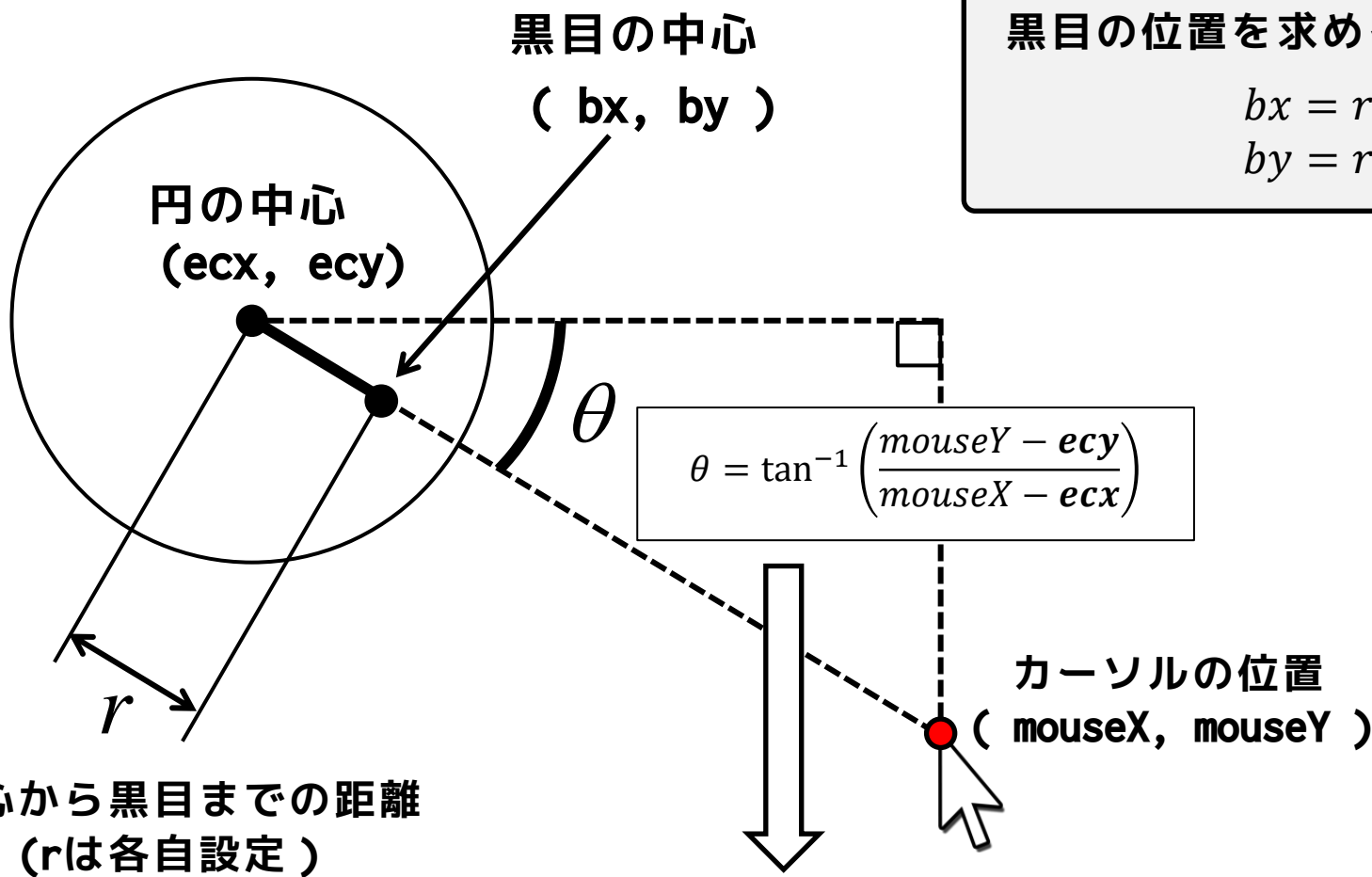
$$\theta = \tan^{-1} \left(\frac{\text{mouseY} - y}{\text{mouseX} - x} \right)$$



```
theta = atan2(mouseY-y, mouseX-x);
```

プログラミング演習(3) 課題

発展課題② のヒント



黒目の位置を求めるための数式

$$bx = r \cos \theta + ecx$$
$$by = r \sin \theta + ecy$$

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

```
theta = atan2(mouseY-ecy, mouseX-ecx);
```