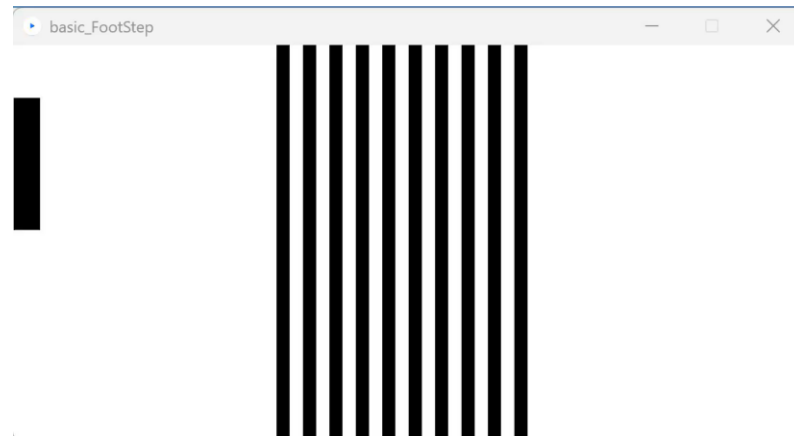
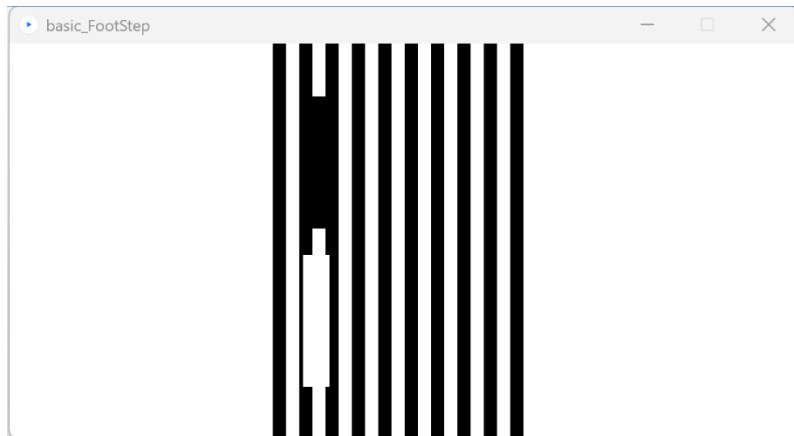


プログラミング演習(4) 課題

• 基本課題① basic_FootStep

- 横600x縦300のウィンドウを作成せよ
- プログラムの画面中央 (x座標が200~400) 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10本描画せよ
- 画面の左端からy座標が40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、同じくy座標が160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、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I (4) 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3ピクセルの円を描画せよ。円は線を描画せず、塗りつぶしの色はRGBで(x 座標の値, y 座標の値, 0)とせよ

- $x = 150 \sin(5t) + 200$

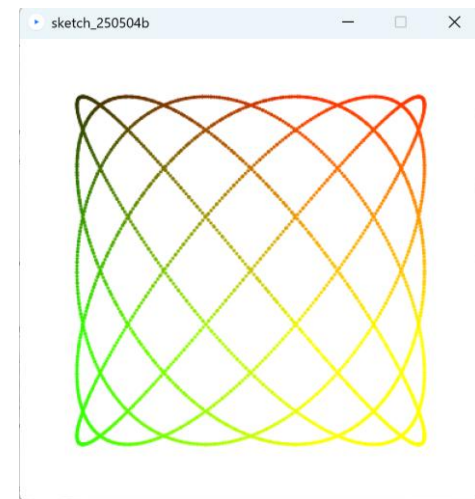
- $y = 150 \sin(6t) + 200$

– t は0から、`draw()` 毎に0.01ずつ増加するようにせよ。

– ウィンドウサイズは400x400とせよ

– 5と6の値を変えると他の図形になるよ！

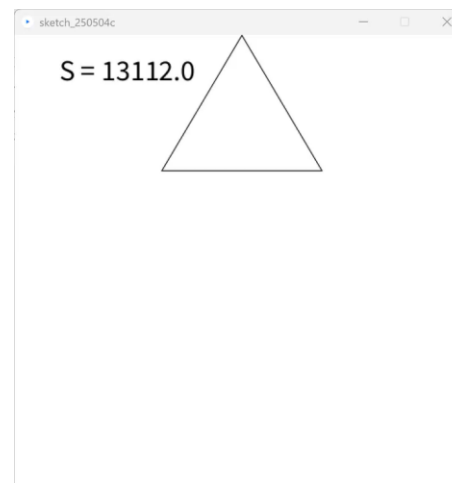
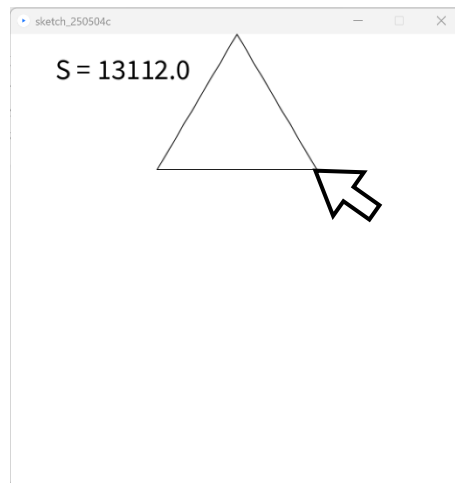
– 出来た人は遊んでみましょう



プログラミング演習(4) 課題

• 基本課題③ スケッチ名: `basic_TriangleArea`

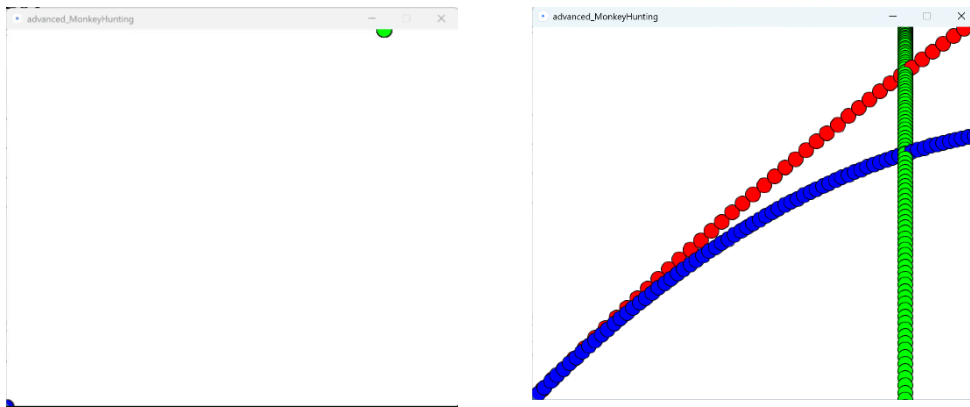
- 縦横500x500のウィンドウを作成し、(250, 0)の位置を頂点として、マウスの座標(mouseX, mouseY)を頂点とした下図のような白色の二等辺三角形を描画せよ
- また、三角形のSを求め、textを利用して下図のように画面出力せよ。描画の順序に注意すること。
- 標準出力はしないようにせよ。
- 絶対値は`abs()`という関数を用いれば良い



プログラミング演習(4) 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
- 600x500のウィンドウを作成し、x座標500、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
- 1フレームを0.1秒とみなし (frameRateで調整は不要)、1フレームごとにtの値を0.1秒ずつ増やせ。また1ピクセルを1mとせよ。



ヒント： 斜方投射の式

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

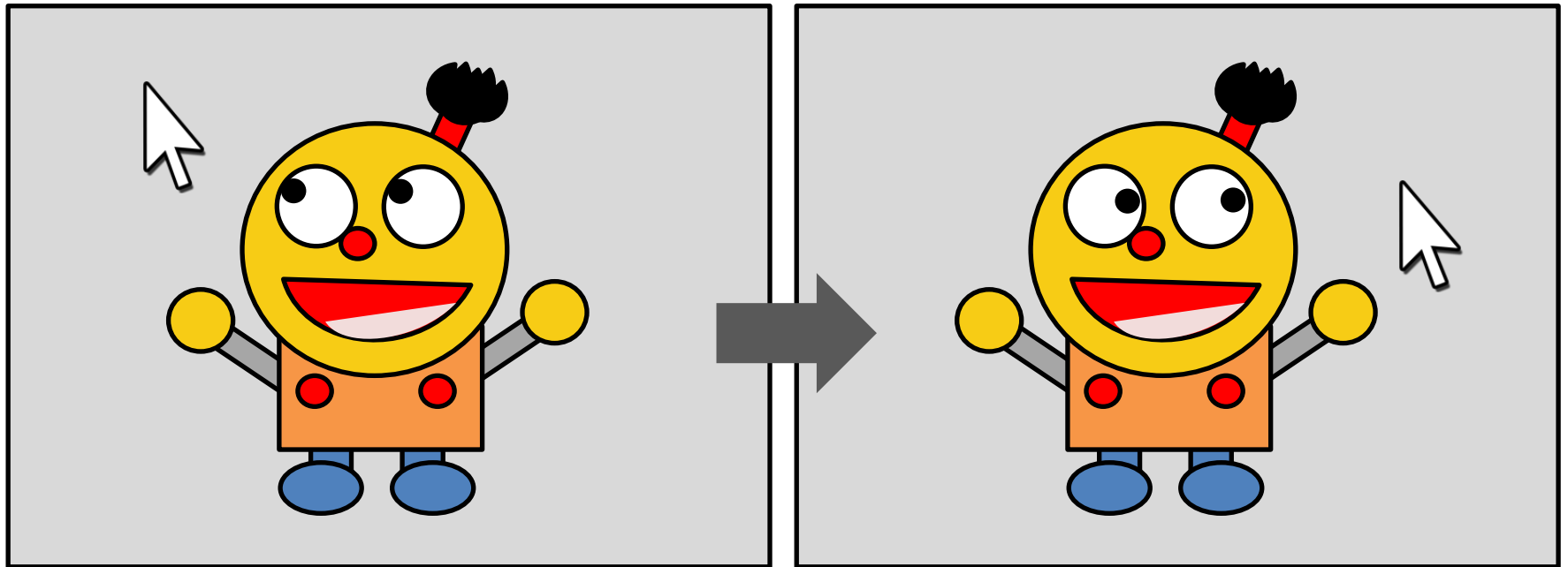
$$y = v_0 \sin(\theta) t - \frac{1}{2} g t^2$$

$$g = 9.8$$

プログラミング演習(4) 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、可能なら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）

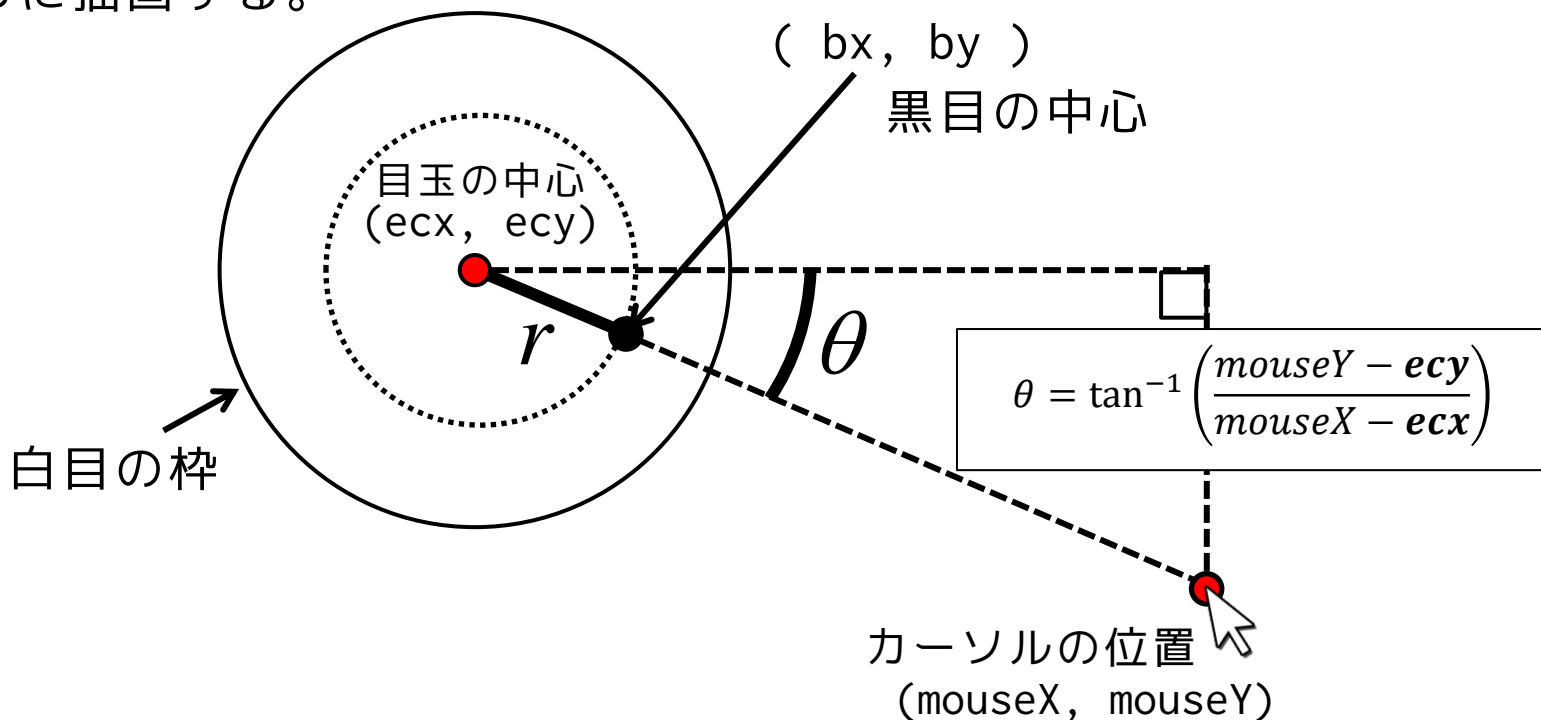


※例えば目玉が楕円形の場合、黒目の動きは楕円軌道にこだわらなくてもよいです。

プログラミング演習(4) 課題

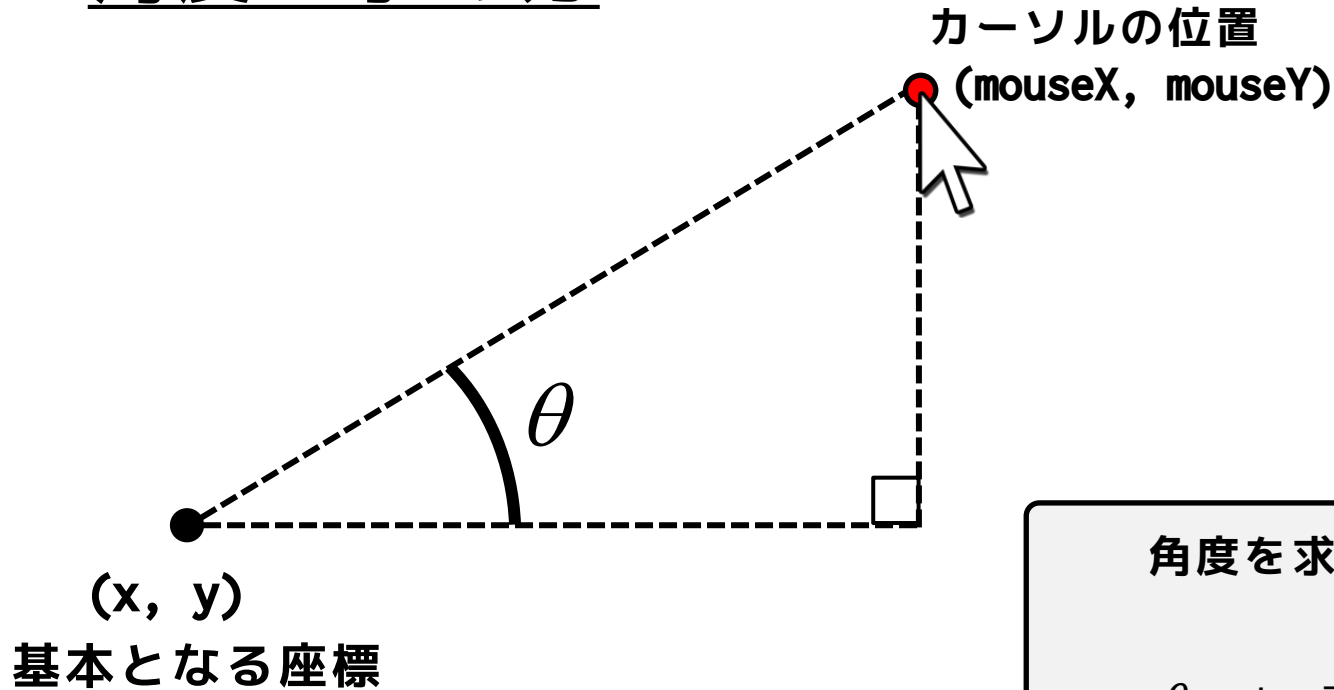
目玉と黒目がある場合の考え方 (1つの目に関する解説)

- 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(4) 課題

• 角度の求め方



角度を求めるための数式

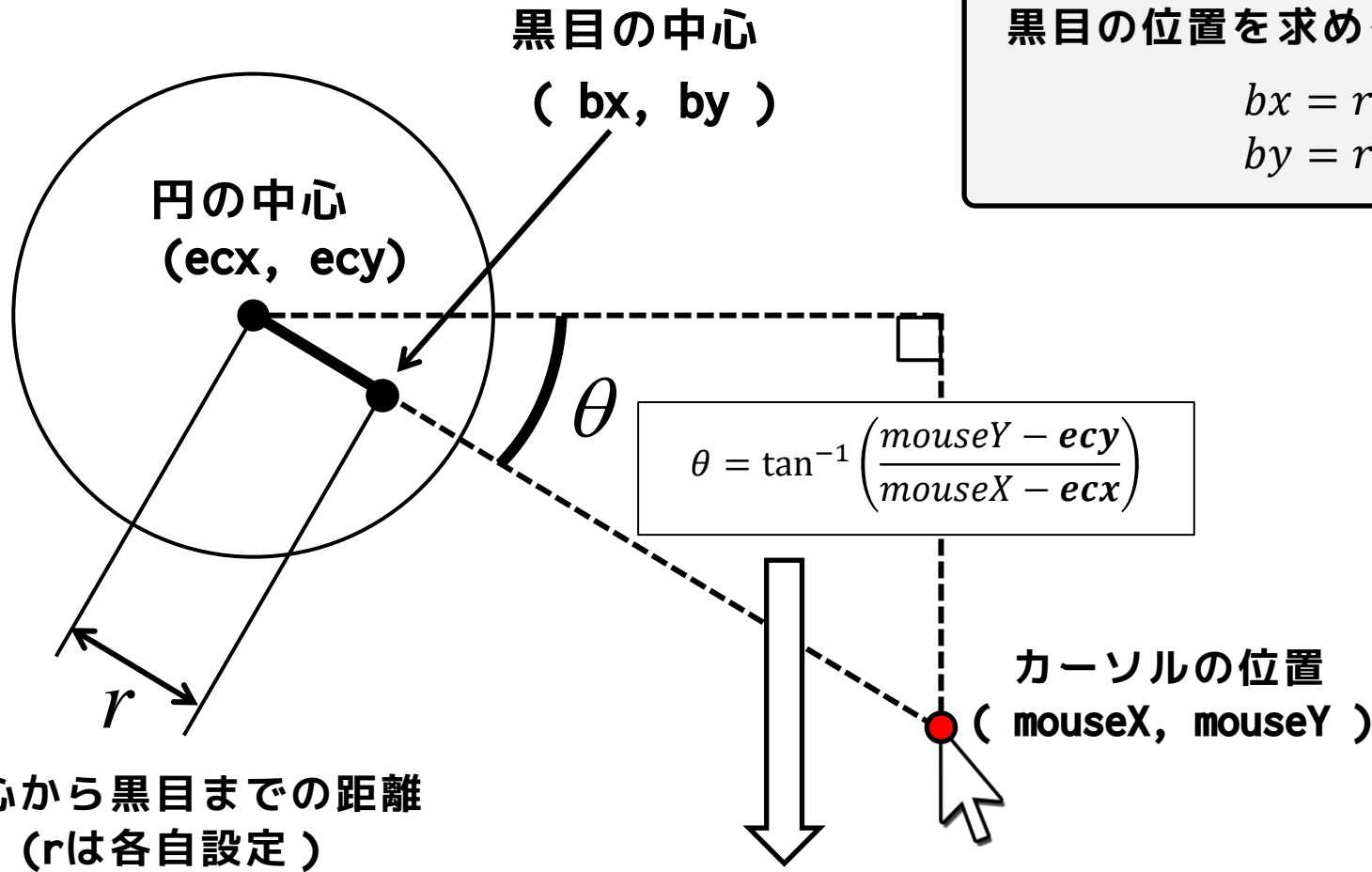
$$\theta = \tan^{-1} \left(\frac{\text{mouseY} - y}{\text{mouseX} - x} \right)$$



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

プログラミング演習(4) 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);
```