

## 2015 年度情報メディア基盤ユニット

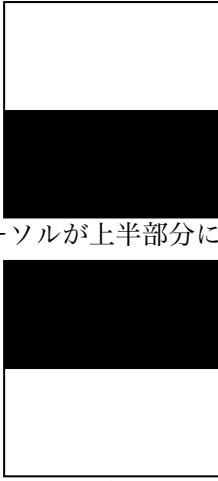
### 5 月 1 日分課題

授業関連資料は <http://www.sato-lab.jp/imfu> からダウンロード出来ます。授業中に配布したプリントに誤りを見つけた際には、修正版をのせてあります。問 3 以降は、出来たら先生か TA の人に確認をしてもらって下さい。

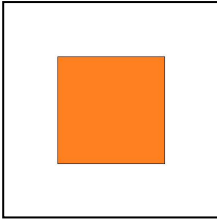
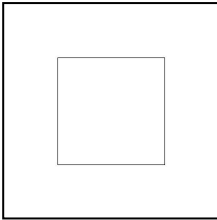
1. 【自己確認】 火曜日の授業時に配布した資料中のサンプルプログラムを実行して見て下さい。
2. 変数  $x$  の値が 5, 変数  $y$  の値が 6 の時に、以下の論理式の値(true か false)を求めよ。

$x > 6$		$x \neq y$	
$x < y$		$y \leq x$	
$!(x > 6)$		$(x == 6 \ \&\& \ x == 5)$	
$(x == 6 \    \ x == 5)$		$(x > -1 \ \&\& \ y < 10)$	
$5 < x \ \&\& \ x < 6$		$!(x \leq 5 \ \&\& \ x < 6)$	

3. 次のプログラムは、マウスカーソルがウインドウ上半部分にあるときには、ウインドウ下半分を黒色に、マウスカーソルがウインドウ下半部分にあるときには、ウインドウ上半分を黒色になるようなプログラムである。空欄を埋めて、プログラムを完成させよ。

プログラム	表示
<pre>void setup(){   size(400,400); }  void draw(){   background(255);   fill(0);   if(mouseY &lt; height/2){     rect(_____, _____, _____, _____);   }else{     rect(_____, _____, _____, _____);   } }</pre>	 <p>マウスカーソルが上半部分にあるとき</p> <p>マウスカーソルが下半部分にあるとき</p>

4. 次のプログラムは、ウインドウの中央の長方形内にマウスカーソルがいる場合には、長方形内部を塗りつぶし、そうでないときには、長方形内部を塗りつぶさない（白色で塗りつぶす）プログラムである。空欄を埋めて、プログラムを完成させよ。

プログラム	表示
<pre> int xLeft = 100; // 長方形の左上の X 座標値 int yTop = 100; // 長方形の左上の Y 座標値 int w = 200; // 長方形の幅 int h = 150; // 長方形の高さ  void setup(){   size(400,400); }  void draw(){   background(255);   stroke(0);   if(_____){ // この条件式はちょっと複雑     fill(255,128,32);   }else{     noFill();   }   rect(_____,_____,_____,_____); } </pre>	<div data-bbox="1024 302 1241 519" style="text-align: center;">  </div> <p data-bbox="917 521 1343 548" style="text-align: center;">マウスカーソルが長方形にあるとき</p> <div data-bbox="1024 595 1241 813" style="text-align: center;">  </div> <p data-bbox="917 815 1343 842" style="text-align: center;">マウスカーソルが長方形にないとき</p>

5. コメントのサンプルのように、プログラム A とプログラム B にコメントを入れて下さい。コメントは、プログラムのソースコード内に、そのソースコードを読む人のために書かれた覚え書き注釈のことです。コメントとして書く内容には、次の3つのものがあります。

- (ア) 命令文や変数に関する説明
- (イ) ソースコードに関するまとめ
- (ウ) ソースコードの処理内容の意図

(ア) では、その変数がどのような用途で利用されるかや、その命令文で何をやろうとしているかなどを書きます。命令文の説明では、説明の繰り返しとなってしまう場合がありますが、そのような時には、コメントは必要ないと思います。例えば、「y = y+1;」という命令文に「変数 y の値を 1 増やす」というコメントを入れても、そんなこと言われなくてもわかると思ってしまうますが、「表示位置を変更する」などと書いてあれば、もう少し役立ちそうに見えます。(イ) と (ウ) では、なぜこの処理方法を選んだのかや、なぜその判断が正しいのかなどを書くと、価値の高いコメントになると思います。コメントに書く内容が上手く思い浮かばない時には、往々にして自分自身で処理内容を良く理解していないことがあります。コメントを書くという行為は、自分自身でのプログラムの理解度のバロメータとなっていると思います。

この問題中のコメントのサンプルは、「説明の繰り返し」となっている部分があるので、あまり良いサンプルとなっていないと思います。例えば、      の部分は Processing のプログラムを書いているときには当たり前の“説明の繰り返し”となっています。      の部分は意味のあるコメントになっていると思います。

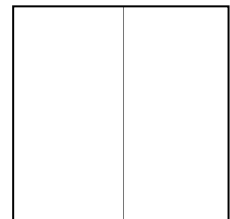
コメントのサンプル	
<pre>int y;  void setup(){   size(200,400);   y = 0; }  void draw(){   background(255);   stroke(0);   line(0,y,width,y);   y = y+1;   if(y &gt;= height){     y = 0;   } }</pre>	<pre>// <u>描画する線分の両端の y 座標</u>  // <u>初期設定に関わる処理を行う関数 setup</u> // <u>横 200、縦 400 のウィンドウを表示</u> // <u>表示する線分の y 座標の初期値を 0 にする</u> // // // <u>表示を行う関数 draw</u> // <u>白色で背景を塗りつぶす</u> // <u>線の描画色を黒色とする</u> // <u>高さ y の位置に水平な線を描画する</u> // <u>表示位置を下げる</u> // <u>線分の描画位置がウィンドウの下からはみ出たら、</u> // <u>線分の描画位置をウィンドウの先頭に移動する</u> // //</pre>
プログラム A	
<pre>int x;  void setup(){   size(400,200);   x = 0; }  void draw(){   background(255);   stroke(0);   line(x,0,x,height);   x = x+1;   if(x &gt;= width){     x = 0;   } }</pre>	<pre> //(1)  //(2)  //(3)  //(4)  //(5)  //(6)  //(7)  //(8)  //(9)  //(a)  //(b)  //(c)  //(d)  //(e)  //(f)  //(g)</pre>
プログラム B	
<pre>float x; float dx; int side;  void setup(){   size(400,200);   x = 0;   side = 50;   dx = 0.5; }  void draw(){   background(255);</pre>	<pre> //(1)  //(2)  //(3)  //(4)  //(5)  //(6)  //(7)  //(8)  //(9)  //(a)  //(b)  //(c)  //(d)</pre>

stroke(255,10,10);	//(e)
fill(255,10,10);	//(f)
rect(x,0,side,height);	//(g)
x = x+dx;	//(h)
if(x < 0){	//(i)
x = 0;	//(j)
dx = 0.5;	//(k)
}else if((x+side)	//(l)
>= (width-1)){	//(m)
x = (width-1)-side;	//(n)
dx = -0.5;	//(o)
}	//(p)
}	//(q)

プログラム中の赤字の部分  
は印刷の都合で折り返した所です。

6. 問4のプログラムに改良を加え、ウインドウの中央の長方形内にマウスカーソルがあり、かつマウスボタンを押した場合に、長方形内部を塗りつぶし、そうでないときには、長方形内部を塗りつぶさない（白色で塗りつぶす）プログラムを作成せよ。

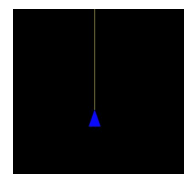
7. 【目コピ問題】直線がウインドウの左から右に向かって移動するようなプログラムを作れ。なお、直線が一番右側に到達したら、再び一番左側に戻り、右方向に移動するようにすること。



8. システム変数 mouseX,mouseY と三角形を描く triangle 命令（関数）を使用して、3点 (mouseX,mouseY-20),(mouseX-14,mouseY+20),(mouseX+14,mouseY+20)を頂点する、三角形を表示するプログラムを作成せよ。

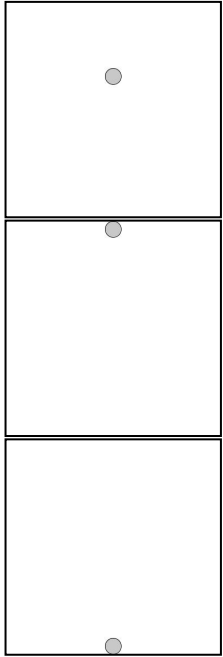
プログラム	表示
<pre>void setup(){   size(400,400); }  void draw(){   background(0);    noStroke();   fill(10,10,255);   triangle(_____, _____, _____, _____, _____, _____); }</pre>	

9. 問8のプログラムに「マウスボタンを押したときに、三角形の頂点

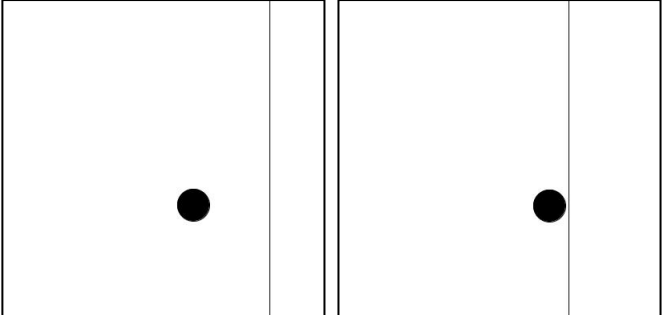


(mouseX,mouseY-20)から、真上に黄色の線分を表示する」という変更を加えたプログラムを作成せよ。

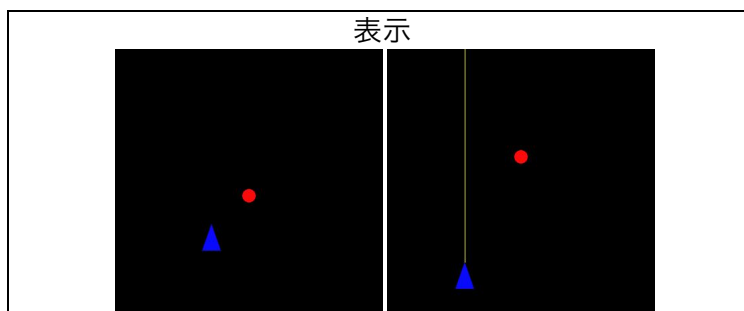
10. システム変数 mouseY を使用して、マウスカーソルと同じ高さに円を表示するプログラムを作れ。ただし、円の一部がウインドウ外に出てしまう場合は、それ以上移動しないようにすること。詳しくは、web ページ上の実行例を見て下さい。

プログラム	表示
<pre> int y; // 円の中心のY座標 int r=15; // 円の半径  void setup(){   size(400,400);   smooth(); }  void draw(){   background(255);   fill(200);    y = mouseY;   if(_____){     y = r;   }else if(_____){     y = _____;   }   ellipse(_____,_____,_____,_____); }         </pre>	

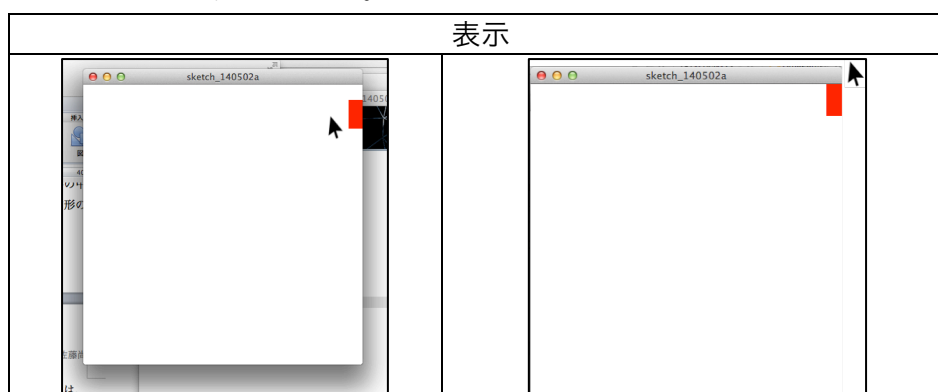
11. 【目コピ問題】 マウスカーソルの位置に円を表示し、この円と直線がぶつかった場合には、移動方向が変化するようにプログラムを作成せよ。なお、直線がウインドウの端に到達した場合にも、移動方向が変化するものとする。詳しくは、web ページ上の実行例を見て下さい。

表示


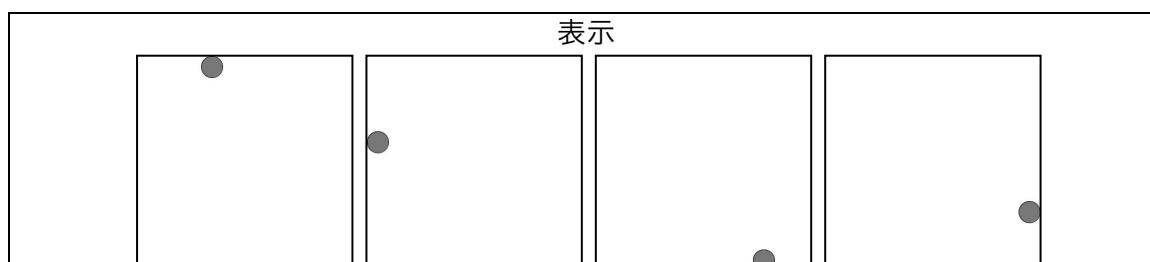
12. 【目コピ問題】 問 9 のプログラムに、ウインドウの上方から赤色の円が落ちてくるような処理を付け加えること。



13. 【目コピ問題】 問 12 のプログラムに、赤色の円と黄色の線分との接触判定を行う処理を付け加えること。接触している判定した場合には、赤色の円の位置をウインドウ最上部に移動させ、再び下方向に移動するようにすること。また、直線と円の接触回数をカウントし、println を利用して、接触が判定されるたびに、その値を表示すること。なお、赤色の円の出現位置をランダムにすると、ちよっぴりゲーム風になります。
14. 【目コピ問題】 システム変数 mouseY を使用して、マウカーソルの Y 座標のが長方形の中心となるように、ウインドウの右隅に長方形を表示するプログラムを作れ。ただし、長方形の一部がウインドウ外に出ってしまう場合は、それ以上移動しないようにすること。詳しくは、web ページ上の実行例を見て下さい。



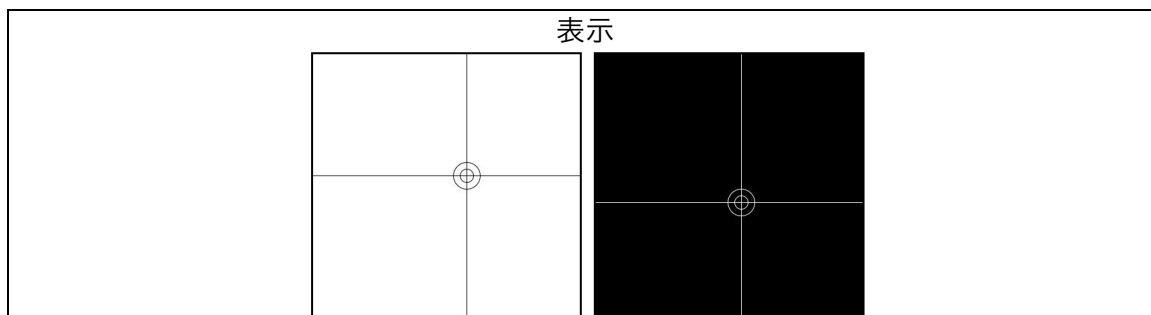
15. 【目コピ問題】 ウインドウの外枠に接しながら、反時計回りで円が移動するようなプログラムを作成せよ



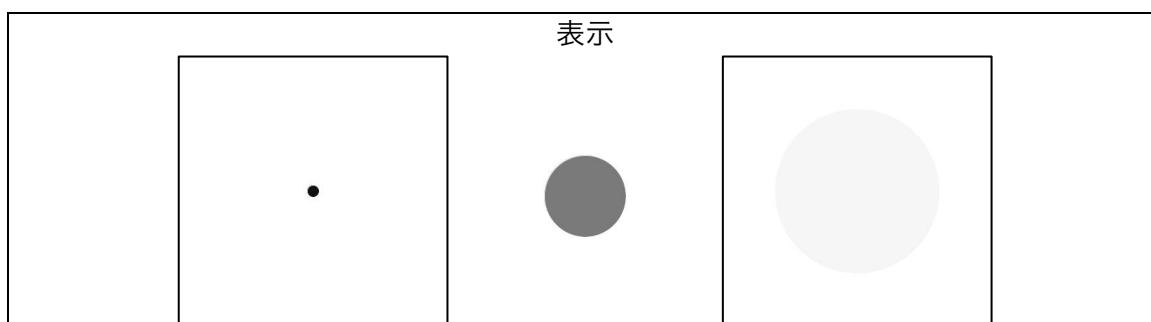
## 宿題

宿題はレポートとして5月8日（金）の授業の時に、提出して下さい。提出するレポートの表紙には、科目名、学籍番号、氏名、提出日、提出先（担当の先生の名前）レポート内容の概要を記載して下さい。

1. マウスマウスの位置を中心に2つの円と十字を表示し、マウスボタンを押したときに、色が反転するようなプログラムを作成せよ。詳しくは、web ページ上の実行例を見て下さい。



2. ウィンドウの真ん中を中心とする円が、マウスボタンを押している間、徐々に大きくなりながら、黒から白に変化するようなプログラムを作成せよ。詳しくは、web ページ上の実行例を見て下さい。



3. 問5のように、プログラムにコメントを入れよ。

<code>int x;</code>	<code>//</code>
	<code>//</code>
<code>void setup(){</code>	<code>//</code>
<code>size(400,100);</code>	<code>//</code>
<code>x = width-1;</code>	<code>//</code>
<code>}</code>	<code>//</code>
	<code>//</code>
<code>void draw(){</code>	<code>//</code>
<code>background(255);</code>	<code>//</code>
<code>stroke(0);</code>	<code>//</code>
<code>line(x,0,x,height);</code>	<code>//</code>
<code>x = x-1;</code>	<code>//</code>
<code>if(x &lt; 0){</code>	<code>//</code>
<code>x = width-1;</code>	<code>//</code>
<code>}</code>	<code>//</code>
<code>}</code>	<code>//</code>

おまけ：乱数を使ったプログラム

random 関数を利用すると、乱数を取り出すことができます。基本的には、random 関数は呼び出される度に異なる値を返します。float 型の値となっていますので、int 型の値を欲しい場合には、int(random(width))などとします。

random：乱数を返す関数	
random(high)	0 以上 high 未満の乱数を返す
random(low,high)	low 以上 high 未満の乱数を返す