

2016 年度情報メディア基盤ユニット

6 月 3 日分課題と宿題

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。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】以下の 4 個のサンプルプログラムを実行して見て下さい。

7-3,7-16,7-17,7-19

入力したソースコードをキャリアポートフォリオ上に提出して下さい。

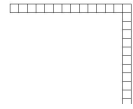
2. 右のプログラムは、マウスカーソルの位置から右下方向に円を表示していくプログラムのつもりでした。円の描画色は赤色から白色に変化し、色の変化量や位置の変化量は乱数で決めています。描く円の中心がウィンドウ外に出たり、彩度の値が負になったら、円の描画を終了します。このプログラムは思うように動作しませんでした。正しく動作するようにプログラムを修正して下さい。変数宣言を行う際には、なるべく有効範囲が小

未完成のプログラム	実行例
<pre>void setup(){ size(400,400); colorMode(HSB,359,99,99); smooth(); } void draw(){ background(0,0,99); s = 99; x = mouseX; y = mouseY; noStroke(); while(s >= 0 && (0 <= x x < width) && (0 <= y y < height)){ fill(0,s,99); ellipse(x,y,20,20); x += int(random(4,10)); y += int(random(4,10)); s += int(random(2,5)); } }</pre>	

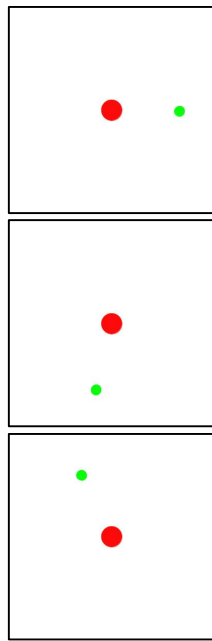
さくなるようにして下さい。面倒くさいからといって、すべての変数を大域変数として、宣言してはダメです。正しく修正したプログラムをキャリアポートフォリオ上にも提出して下さい。

3. 次のプログラムは、1 辺の長さが 10 の正方形を右方向に描いていき、右端に来る直前に下方向に描くようなプログラムです。変数の宣言をおこなっていないため、このままでは動作しません。適切な変数宣言を追加して、プログラムが動作するようにして下さい。なお、変数宣言を行う場合には、なるべく変数の有効範囲が狭くなるようにして下さい。全ての変数をグローバル変数にするでは、ダメです。for 文バージョンか while バージ

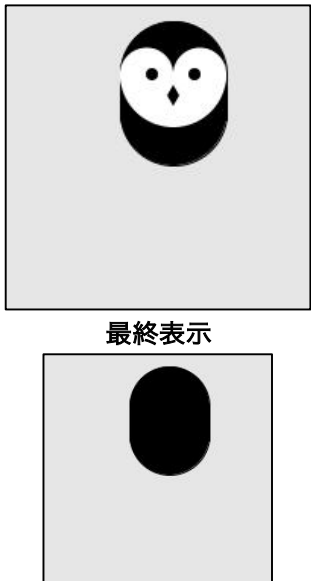
ョンのどちらかを解答すれば OK です。正しく修正したプログラムをキャリアポートフォリオ上にも提出して下さい。

未完成のプログラム (for バージョン)	実行例
<pre>void setup(){ size(400,400); smooth(); len = 20; } void draw(){ background(255); noFill(); stroke(0); for(x=mouseX;(x+len) < width;x += len){ rect(x,mouseY,len,len); } x = x-len; for(y=mouseY;y < height;y += len){ rect(x,y,len,len); } }</pre>	
未完成のプログラム (while バージョン)	
<pre>void setup(){ size(400,400); smooth(); len = 20; } void draw(){ background(255); noFill(); stroke(0); x = mouseX; while((x+len) < width){ rect(x,mouseY,len,len); x = x + len; } x = x-len; y = mouseY; while(y < height){ rect(x,y,len,len); y = y + len; } }</pre>	

4. 【目コピ問題】 未完成プログラムはウインドウの中心に赤色の円を表示し、その外側を緑色の円が回転するようなプログラムです。空欄に追加して、プログラムを完成させて下さい。赤い円の中心と緑色の円の中心の距離は、ウインドウの幅の 3 分の 1 となっている。正しく動くプログラムをキャリアポートフォリオ上にも提出して下さい。

未完成のプログラム	実行例
<pre>float theta; void setup(){ size(400,400); smooth(); theta = 0; } void draw(){ background(255); translate(__(a),__(b)); stroke(255,10,10); fill(255,10,10); ellipse(0,0,40,40); theta = theta + 1; rotate(radians(__(c))); stroke(10,255,10); fill(10,255,10); ellipse(__(d),__(e),20,20); }</pre>	

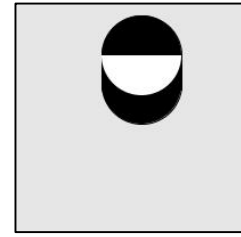
5. 【目コピ問題】 未完成プログラムは、フクロウ(owl)のような画像を表示するものです。くちばしの部分は四角形で描いています。空欄に追加して、プログラムを完成させて下さい。この四角形の中心の座標はいくつになるでしょうか？考えてみて下さい。なお、このサンプルは「Processing をはじめよう」(オライリージャパン)を参考にしました。

未完成のプログラム	実行例
<pre>size(200,200); smooth(); background(230); // 胴体 stroke(0); fill(0); ellipse(110,45,70,70); ellipse(110,70,70,70); rect(75,45,70,30); noStroke(); fill(255); arc(__(a),__(b),70, 70, 0, PI); // あご // 左目 fill(__(c)); ellipse(92.5, 45, 35, 35); fill(__(d)); ellipse(96, 45, 8, 8); // 右目 fill(__(c)); ellipse(127.5, 45, 35, 35); fill(__(d));</pre>	 最終表示 あごを描く直前

```

ellipse(124, 45, 8, 8);
fill(0);
quad(110, 52, 114, 59, 110, 66, 106, 59); // くちばし

```

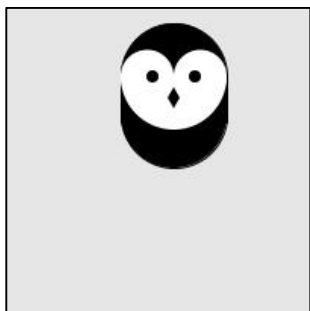


あごを描き終わった直後

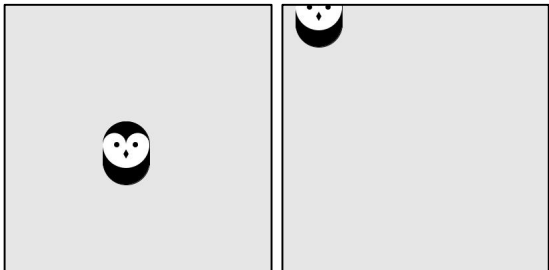
6. 【目コピ問題】 未完成プログラムは、問 5 と同じ動作をするものです。フクロウ(owl)のような画像を表示するものです。空欄に追加して、プログラムを完成させて下さい。

未完成のプログラム	実行例
<pre> // 左目を描く void drawLeftEye(float x,float y){ noStroke(); fill(__(a)); ellipse(__(b),__(c),35,35); fill(__(d)); ellipse(__(e),__(f),8,8); } // 右目を描く void drawRightEye(float x,float y){ noStroke(); fill(__(g)); ellipse(__(h),__(i),35,35); fill(__(j)); ellipse(__(k),__(l),8,8); } void setup() { size(200, 200); smooth(); } void draw() { background(230); // 胴体 stroke(0); fill(0); ellipse(110, 45, 70, 70); ellipse(110, 70, 70, 70); rect(75, 45, 70, 30); noStroke(); fill(255); arc(110,45, 70, 70, 0, PI); // あご // 左目 drawLeftEye(96,45); // 右目 drawRightEye(124,45); quad(110, 52, 114, 59, 110, 66, 106, 59); // くちばし } </pre>	

7. 【目コピ問題】 未完成プログラムは、問 6 と同じ動作をするものです。フクロウ(owl)のような画像を表示するものです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数を完成させ、プログラムが動作するようにして下さい。正しく動作するプログラムをキャリアポートフォリオ上にも提出して下さい。

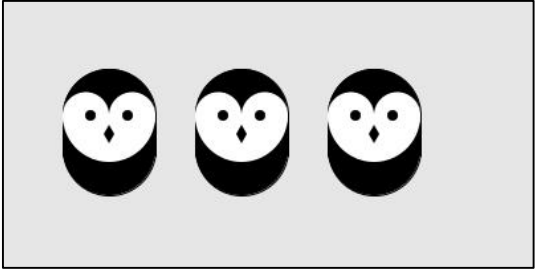
未完成のプログラム	実行例
<pre>// 左目を描く void drawLeftEye(float x,float y){ // この部分に追加 } // 右目を描く void drawRightEye(float x,float y){ // この部分に追加 } // フクロウを描く void drawOwl(){ // この部分に追加 } void setup() { size(200, 200); smooth(); } void draw() { background(230); drawOwl(); }</pre>	

8. 【目コピ問題】 未完成プログラムは、問 7 で作成した drawOwl 関数を変更して、マウスカーソル位置にフクロウのくちばしの中心が来るように、移動するプログラムです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数を完成させ、プログラムが動作するようにして下さい。

未完成のプログラム	実行例
<pre>// 左目を描く void drawLeftEye(float x,float y){ // この部分に追加 } // 右目を描く void drawRightEye(float x,float y){ // この部分に追加 } // フクロウを描く void drawOwl(float x,float y){ pushMatrix(); // この部分に追加 popMatrix(); }</pre>	

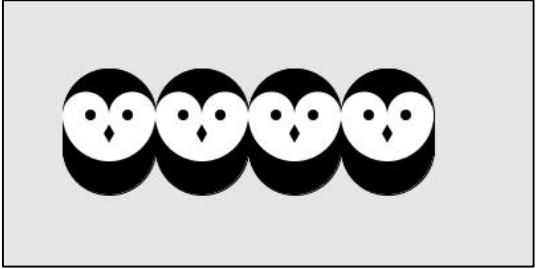
<pre> } void setup() { size(400, 400); smooth(); } void draw() { background(230); drawOwl(mouseX,mouseY); } </pre>	
--	--

9. 【目コピ問題】 未完成プログラムは、問 8 で作成した drawOwl 関数を利用して複数のフクロウを表示するものです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数、draw 関数を完成させ、プログラムが動作するようにして下さい。

未完成のプログラム	実行例
<pre> // 左目を描く void drawLeftEye(float x,float y){ // この部分に追加 } // 右目を描く void drawRightEye(float x,float y){ // この部分に追加 } // フクロウを描く void drawOwl(float x,float y){ pushMatrix(); // この部分に追加 popMatrix(); } void setup() { size(400, 400); smooth(); } void draw() { background(230); drawOwl(__(a)__,height/2); drawOwl(__(b)__,height/2); drawOwl(__(c)__,height/2); } </pre>	

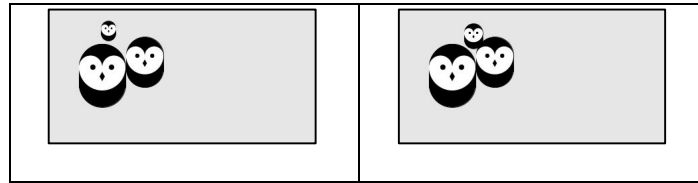
10. 【目コピ問題】 未完成プログラムは、問 8 で作成した drawOwl 関数を利用して複数のフクロウを表示するものです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数、draw 関数を完成させ、プログラムが動作するようにして下さい。なお、フクロウは隙間無く並んでいるものとする。

未完成のプログラム	実行例
-----------	-----

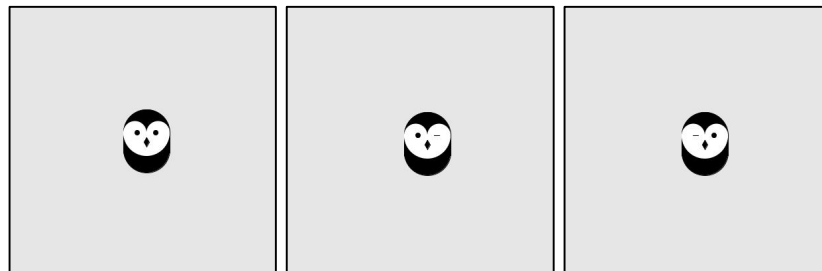
<pre>// 左目を描く void drawLeftEye(float x,float y){ // この部分に追加 } // 右目を描く void drawRightEye(float x,float y){ // この部分に追加 } // フクロウを描く void drawOwl(float x,float y){ pushMatrix(); // この部分に追加 popMatrix(); } void setup() { size(400, 400); smooth(); } void draw() { background(230); drawOwl(__(a)__,height/2); drawOwl(__(b)__,height/2); drawOwl(__(c)__,height/2); drawOwl(__(d)__,height/2); }</pre>	
--	--

11. 次の 2 つのプログラムの実行結果が異なる理由を説明して下さい。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数、draw 関数は問 8 で作成したものを使用して下さい。

プログラム 1	プログラム 2
<pre>void setup() { size(400, 200); smooth(); } void draw() { background(230); drawOwl(80,height/2); scale(0.8,0.8); drawOwl(180,height/2); scale(0.4,0.4); drawOwl(280,height/2); }</pre>	<pre>void setup() { size(400, 200); smooth(); } void draw() { background(230); drawOwl(80,height/2); pushMatrix(); scale(0.8,0.8); drawOwl(180,height/2); popMatrix(); scale(0.4,0.4); drawOwl(280,height/2); }</pre>



12. 【目コピ問題】 マウスボタンを押したら、目をつむったフクロウが表示されるようなプログラムを作成して下さい。右ボタンを押したら右目を、左ボタンを押したら左目を閉じるようにすること。



13. 【目コピ問題】 未完成プログラムは、`millis` 関数を用いてマウスを押している間の経過時間をマウスカーソルの位置に表示するものです。また、その時点での経過時間の最も長い時間をウインドウに左下に表示しています。動作するように、空欄を埋めて下さい。また、変数宣言がされていない変数があります。適切な変数宣言を加えて下さい。この変数宣言のための空欄は用意されていません。

未完成のプログラム	実行例
<pre> PFont font; int maxTime; (a) pressing=false; void setup(){ size(400,400); maxTime = 0; smooth(); (b) textFont(font,24); } void draw(){ background(255); fill(0); if(mousePressed){ if(pressing == (c)){ startTime = millis(); pressing = (d); } interval = (e)-startTime; text(interval,mouseX,mouseY); }else{ pressing = (f); } } </pre>	

<pre> } if((g)> maxTime){ maxTime = (h); } text(maxTime,0,height-textDescent()); } </pre>	
--	--

これ以降の問題は余裕のある人向けです。

14. 【目コピ問題】 次のプログラムは、
mouseX の値により色相の値を決め、
円を描くものです。マウスが左端に
いるときの色相の値は 0、右端にいると



きの色相の値は 359 としています。このプログラムでは、不透明度付きの色（白色）で
ウインドウ全体を覆う四角形を描画することで、フェードアウト効果を出しています。
このフェードアウトの機能は、fadeTo 関数と fadeToWhite 関数を定義して、実現して
います。また、円の半径は 10～30 の乱数で決めています。空欄を埋めて、プログラム
を完成させて下さい。

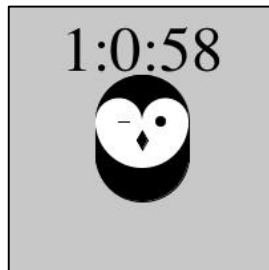
未完成のプログラム
<pre> void setup(){ size(700,200); colorMode(HSB,359,99,99); smooth(); } void draw(){ fadeToWhite(); color c = color(map(mouseX,__(a)__,__(b)__,__(c)__,__(d)__),99,99); stroke(c); fill(c); float r = random(__(e)__,__(f)__); ellipse(mouseX,mouseY,2*r,2*r); } void fadeTo(color c){ stroke(c,20); fill(c,20); rectMode(CORNER); rect(__(g)__,__(h)__,__(i)__,__(j)__); } void fadeToWhite(){ fadeTo(__(k)__); } </pre>

15. 【目コピ問題】 現在時刻の秒を 4 で割ったときの余りが、

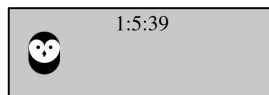
- 0 の時：両目を閉じているフクロウを表示する

- 1 の時：右目だけを閉じているフクロウを表示する
- 2 の時：左目だけを閉じているフクロウを表示する
- 3 の時：両目を開けているフクロウを表示する

ようなプログラムを作成して下さい。

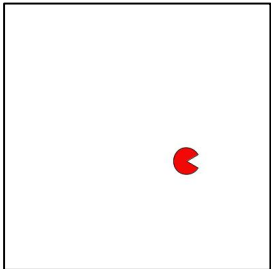


16. 【工夫問題】 現在時刻の分の値によって、フクロウの位置が変わるようなプログラムを作成して下さい。例えば、0 分の時には一番左、59 分の時には一番右に表示されるようにして下さい。

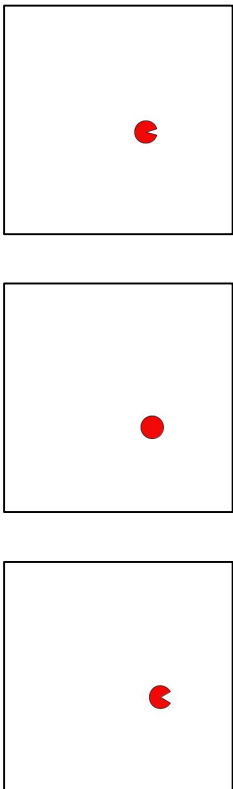


17. 【目コピ問題】 未完成プログラムは、arc 関数などを利用して、パッ○マン風の画像を表示するものです。動作するように、空欄を埋めて下さい。可能なら色は黄色にしてください。mouseX の値によって、口の開き具合を調節しています。

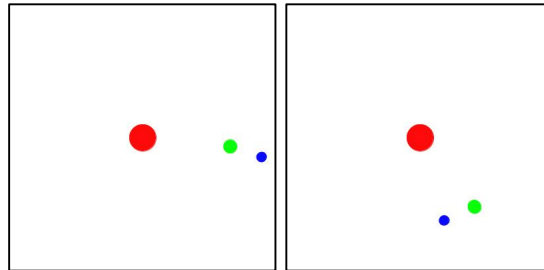
未完成のプログラム	実行例
<pre> void setup() { size(400, 400); colorMode(HSB,359,99,99); smooth(); } void draw() { background(0,0,99); fill((a), 95, 95); stroke(0,99,0); translate(mouseX,mouseY); if (mouseX % 10 < 5) { arc(0, 0, 40, 40, PI/6, 11*PI/6); rotate(PI/6); line(0,0,(b),0); rotate((c)); line((d)); }else if (mouseX % 10 < 9) { arc(0 ,0, 40, 40, PI/12, 2*PI-PI/12); rotate((e)); } } </pre>	

<pre> line((f)); rotate((g)); line((h)); } else { ellipse((i), (j), 40, 40); } } </pre>	
---	---

18. 【目コピ問題】 未完成プログラムは、arc 関数などを利用して、パッ○マン風の画像を表示するものです。動作するように、空欄を埋めて下さい。可能なら色は黄色にしてください。Processing には、pmouseX と pmouseY と呼ばれるシステム変数があります。これは、1 つ前のマウスカーソルの X 座標と Y 座標の値を保存しています。下のプログラムは、マウスカーソルが X 座標方向に動いた向きにパッ○マン風の画像を表示するものです。マウスカーソルが X 座標方向には移動していない場合には、向きを変えずに表示します。プログラムを適当におぎなつて、キチンと動作するようにして下さい。

未完成のプログラム	実行例
<pre> boolean right = true; void setup() { size(400, 400); colorMode(HSB,359,99,99); smooth(); } void draw() { background(0,0,99); fill(0, 95, 95); stroke(0,99,0); int t = millis(); if(pmouseX < mouseX){ // 左から右に移動 right = true; }else if(pmouseX > mouseX){ // 右から左に移動 right = false; // }else if(right){ //右向きのまま移動せず // }else{ // 左向きのまま移動せず // } } </pre>	

19. 【目コピ問題】 ウィンドウの中心に赤色の円盤を表示し、その外側を緑色の円盤が回転し、その緑色の円盤の周りを青色の円盤が移動するようなプログラムを作成して下さい。

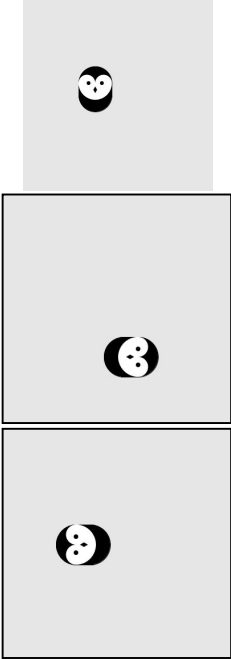


20. 【目コピ問題】 「現在の座標軸」を基準に原点から(200,0)まで直線を描き、その後、`translate(200,0)`と `rotate(radians(180-36))`を実行します。この操作を何回か繰り返し、星形を描く関数 `foo(int w,color c)`を定義して下さい。引数 `w` では描く直線の幅を指定し、引数 `c` では描く線の色を指定します。この関数では、`pushMatrix` 関数と `popMatrix` 関数を使用して、`foo` 関数実行前の「現在の座標軸」の状態に戻してから、`foo` 関数の実行を終了するようにして下さい。また、この関数 `foo` を利用して、次のプログラムを実行してみてください。

```
void setup() {
  size(400, 400);
  smooth();
}
void draw() {
  background(255);
  stroke(0);
  translate(mouseX, mouseY);
  foo(20,color(0,0,0)); // 太さ 20 で黒色の線で描く
  foo(10,color(0,0,255)); // 太さ 10 で青色の線で描く
  foo(2,color(0,255,0)); // 太さ 2 で緑色の線で描く
}
```

宿題

1. 【目コピ問題】 下のプログラムは、マウスボタンを押している間、フクロウが時計周りに回転するようなプログラムです。空欄を埋めて、プログラムを完成させて下さい。なお、回転角度は1度ずつ増えるようにして下さい。

未完成のプログラム	実行例
<pre> float rot; // フクロウの回転角度を保存している変数 void setup() { size(400, 400); smooth(); rot = 0; } void drawLeftEye(float x,float y){ noStroke(); fill(255); ellipse(x-3.5,y,35,35); fill(0); ellipse(x,y,8,8); } void drawRightEye(float x,float y){ noStroke(); fill(255); ellipse(x+3.5,y,35,35); fill(0); ellipse(x,y,8,8); } void drawOwl(float x,float y){ pushMatrix(); translate(x-110,y-59); stroke(0); fill(0); ellipse(110, 45, 70, 70); ellipse(110, 70, 70, 70); rect(75, 45, 70, 30); noStroke(); fill(255); arc(110,45, 70, 70, 0, PI); drawLeftEye(96,45); drawRightEye(124,45); quad(110, 52, 114, 59, 110, 66, 106, 59); popMatrix(); } void draw() { background(230); if(__(a)__) { rot = rot + __(b)__; } translate(__(c)__,__(d)__); rotate(__(e)__); drawOwl(0,0); } </pre>	 The execution examples show an owl on a light gray background. The first example shows the owl facing forward (0 degrees). The second example shows the owl rotated 90 degrees clockwise, appearing upside down. The third example shows the owl rotated 180 degrees clockwise, appearing as a mirror image. The fourth example shows the owl rotated 270 degrees clockwise, appearing on its side.