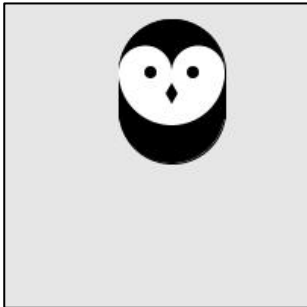
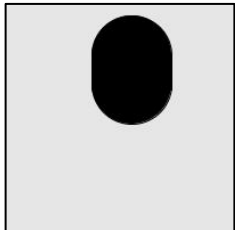
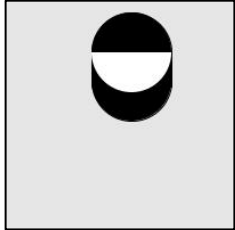


2013 年度情報メディア基盤ユニット

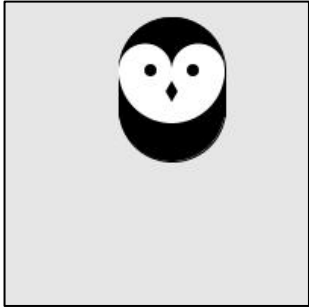
6 月 7 日分課題と宿題

授業関連資料は <http://www.sato-lab.jp/imfu2013> 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】本文中のサンプルプログラムを実行して見て下さい。
2. 【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。

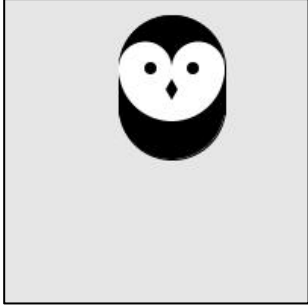
未完成のプログラム	実行例
<pre>size(200,200); smooth(); background(230); // 胴体 stroke(0); fill(0); ellipse(110,45,70,70); ellipse(110,70,70,70); rect(75,45,70,30); noStroke(); fill(255); arc((a) , (b) ,70, 70, 0, PI); // あご // 左目 fill((c)); ellipse(92.5, 45, 35, 35); fill((d)); ellipse(96, 45, 8, 8); // 右目 fill((c)); ellipse(127.5, 45, 35, 35); fill((d)); ellipse(124, 45, 8, 8); fill(0); quad(110, 52, 114, 59, 110, 66, 106, 59); // くちばし</pre>	 <p>最終表示</p>  <p>あごを描く直前</p>  <p>あごを描き終わった直後</p>

3. 【目コピ問題】未完成プログラムは、問 2 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。

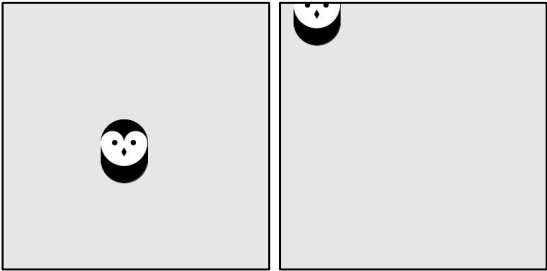
未完成のプログラム	実行例
<pre> // 左目を描く void drawLeftEye(float x,float y){ noStroke(); fill(__(a)); ellipse(__(b),__(c),35,35); fill(__(d)); ellipse(__(e),__(f),8,8); } // 右目を描く void drawRightEye(float x,float y){ noStroke(); fill(__(g)); ellipse(__(h),__(i),35,35); fill(__(j)); ellipse(__(k),__(l),8,8); } void 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>	

4. 【目コピ問題】 未完成プログラムは、問 3 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。

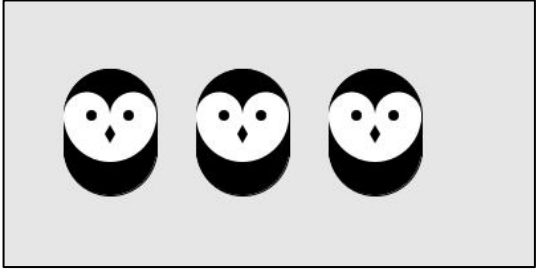
未完成のプログラム	実行例
<pre> // 左目を描く void drawLeftEye(float x,float y){ // この部分に追加 } </pre>	

<pre> // 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>	
--	---

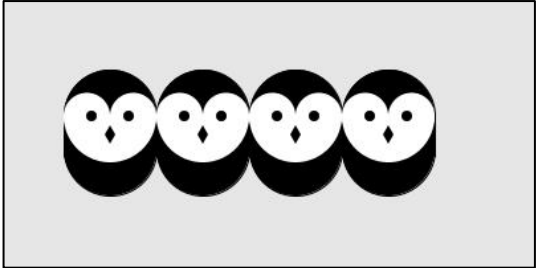
5. 【目コピ問題】 未完成プログラムは、問 4 で作成した drawOwl 関数を変更して、マウスカーソル位置にフクロウのくちばしの中心が来るように、移動するプログラムです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数を完成させ、プログラムが動作するようにして下さい。

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

6. 【目コピ問題】 未完成プログラムは、問 5 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>	

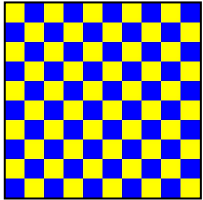
7. 【目コピ問題】 未完成プログラムは、問 5 で作成した drawOwl 関数を利用して複数のフクロウを表示するものです。drawLeftEye 関数、drawRightEye 関数、drawOwl 関数、draw 関数を完成させ、プログラムが動作するようにして下さい。なお、フクロウは隙間無く並んでいるものとする。

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 } </pre>	

```
// 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);
}
```

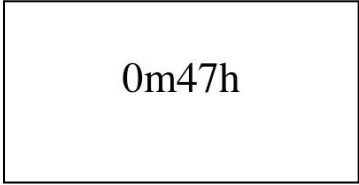
8. 下の2つのプログラムは、青と黄のチェック模様を描くプログラムです。プログラム中で使用している関数 check は boolean 型を返す関数で、2つの int 型の引数 m と n の和が偶数の時には true、奇数の時には false を返します。空欄を埋めて、プログラムを完成させて下さい。偶数を2で割ると余りは…。



未完成のプログラムその1	未完成のプログラムその2
<pre>void setup(){ size(400,400); } void draw(){ for(int x=0;x <10;x++){ for(int y=0;y <10;y++){ if(check(x,y)){ stroke((a) , (b) , (c)); fill((a) , (b) , (c)); }else{ stroke((d) , (e) , (f)); fill((d) , (e) , (f)); } rect(40*x,40*y,40,40); } } } (g) check(int m,int n){ if((m+n) % 2 == 0){ return (h) ; }else{</pre>	<pre>void setup(){ size(400,400); } void draw(){ background(255); for(int x=0;x <10;x++){ for(int y=0;y <10;y++){ if(check(x,y)){ strokeAndFill((a) , (b) , (c)); }else{ strokeAndFill((d) , (e) , (f)); } rect(40*x,40*y,40,40); } } } (g) check(int m,int n){ return (j) ; } (k) strokeAndFill(int r,int g,int b){ stroke((l) , (m) , (n)); fill((l) , (m) , (n)); }</pre>

<pre> return __(i); } } </pre>	
--------------------------------	--

9. 【目コピ問題】 下のプログラムは、プログラムの起動時からの経過時間をウインドウの中央部分に表示するものです。また、マウスボタンを押すと、そのときのウインドウの内容を test.jpg というファイルに保存するという動作をします。このプログラム中の関数 millisToMS は String 型の値を返す関数です。これは、ミリ秒で表された時間を「分 m 秒 s」の形の文字列(String 型)に変換するような関数です。例えば、900 ミリ秒の時には、「0m0s」、2000 ミリ秒の時には「0m2s」、100000 ミリ秒の時には「1m40s」となります。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> PFont font; void setup(){ size(400,200); font = loadFont(__(a)); textFont(font,48); textAlign(__(b)); smooth(); } void draw(){ background(255); fill(0); String time = millisToMS(millis()); text(time,__(c),__(d)); } __(e) millisToMS(int ms){ int ts = ms / __(f); // ミリ秒を秒に変換 int s = __(g); // ts から秒の部分を求める int m = __(h); // ts から分の部分を求める __(i) str(m) + "m" + str(s) + "s"; } void __(j)(){ save("test.jpg"); } </pre>

10. 【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

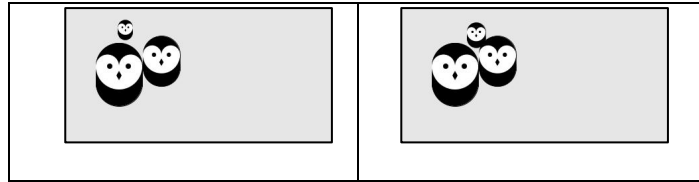


ンドウ全体を覆う四角形を描画することで、フェードアウト効果を出しています。このフェードアウトの機能は、fadeTo 関数と fadeToWhite 関数を定義して、実現しています。また、円の半径は 10~30 の乱数で決めています。空欄を埋めて、プログラムを完成させて下さい。

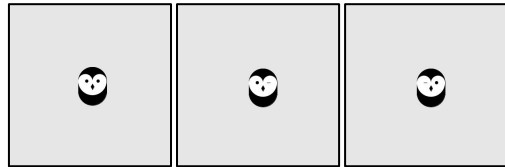
未完成のプログラム
<pre> void setup(){ size(700,200); colorMode(HSB,359,99,99); smooth(); } void draw(){ fadeToWhite(); color c = color(map(mouseX, (a) , (b), (c) , (d)),99,99); stroke(c); fill(c); float r = random((e) , (f)); ellipse(mouseX,mouseY,2*r,2*r); } void fadeTo(color c){ stroke(c,20); fill(c,20); rectMode(CORNER);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

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 7 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>

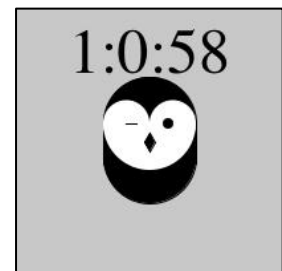


12. 【目コピ問題】 マウスボタンを押したら、目をつむったフクロウが表示されるようなプログラムを作成して下さい。右ボタンを押したら右目を、左ボタンを押したら左目を閉じるようにすること。



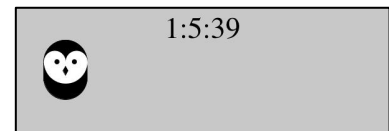
13. 【目コピ問題】 現在時刻の秒を 4 で割ったときの余りが、

- 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。

14. 【工夫問題】 問 13 のプログラムにおいて、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。



15. 【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

宿題はレポートとして 6 月 14 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. 【目コピ問題】下のプログラムは、現在の時刻が一日のうちでどれくらい経過したかをしめすプログラムです。つまり、現在の時刻が 0 時 0 分 0 秒から何秒経っているかを求め、それをもとにウインドウ中に表示する長方形の横幅を決めています。戻り値が int 型の seconds 関数は、0 時 0 分 0 秒から現在までの総秒数をもとめるものです。空欄を埋めて、プログラムを完成させて下さい。もし可能なら、長方形の代わりにグラデーションさせるようなプログラムを作成して下さい。

未完成のプログラム			
<pre>void setup(){ size(600,200); } void draw(){ background(50); fill(240); stroke(240); float xMax = map(seconds(), __ (a) __ , __ (b) __ , __ (c) __ , __ (d) __); rect(0,0,xMax,height); } // 0時0分0秒から現在までの総秒数を求める関数 __ (e) __ seconds(){ int h = hour(); int m = minute(); int s = second(); return __ (f) __ ; }</pre>			
			

お知らせ

6 月 21 日（金）に中間試験を行います。試験範囲は 6 月 4 日（火）の講義の関数定義までです。