

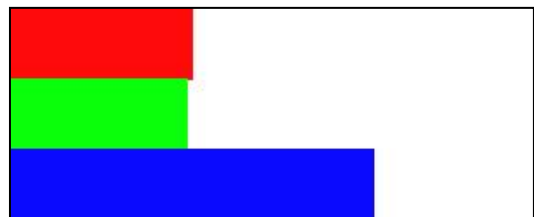
2013 年度情報メディア基盤ユニット

6 月 14 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。【自己確認問題】は先生や TA の人にチェックしてもら必要はありません。できたら先生か 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】下のプログラムは、現在の時刻の時の情報を赤色のバーで、分の情報を緑色のバーで、秒の情報を青色のバーで表すものです。バーの長さが長いほど値が大きいような表示となっています。バーの高さは 3 つとも同じとしています。空欄を埋めて、プログラムを完成させて下さい。

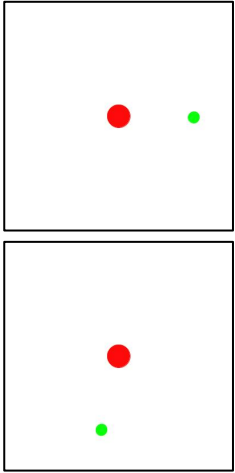


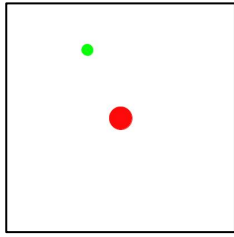
未完成のプログラム
<pre>void setup(){ size(300,120); } void drawRect(float x,float y,float w,float h,color c){ rectMode(CORNER); stroke(__(a)__); fill(__(b)__); rect(__(c)__,__(d)__,__(e)__,__(f)__); } void draw(){ background(255); float x = map(hour(),__(g)__,__(h)__,__(i)__,__(j)__); drawRect(0,0,x,height/3,color(__(k)__,__(l)__,__(m)__)); x = map(minute(),__(n)__,__(o)__,__(p)__,__(q)__); drawRect(0,height/3,x,height/3,color(__(r)__,__(s)__,__(t)__)); x = map(second(),__(u)__,__(v)__,__(w)__,__(x)__); drawRect(0,2*height/3,x,height/3,color(__(y)__,__(z)__,__(aa)__)); }</pre>

3. 【目コピ問題】下のプログラムは、マウスをドラッグしている際には”Dragging...”と表示し、単にマウスを移動している時には”Moving...”と表示するプログラムです。空欄を埋めて、プログラムを完成させて下さい。

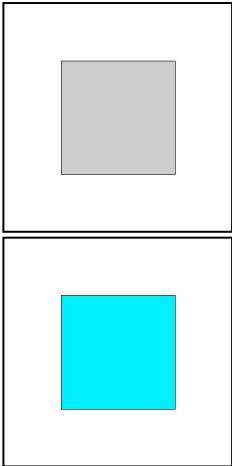
未完成のプログラム	実行例
<pre> PFont font; String msg; __(a) displaying = false;// 表示するかどうかを決める変数 void setup(){ size(400,400); smooth(); font = loadFont(__(b)); textFont(font,48); textAlign(CENTER); } void draw(){ background(255); fill(0); if(displaying){ text(msg,width/2,height/2); displaying = false; } } void mouseMoved(){ displaying = __(c) ; msg = __(d) ; } void mouseDragged(){ displaying = __(e) ; msg = __(f) ; } </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。



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); </pre>	

<pre> theta = theta + 1; rotate(radians(__(c))); translate(__(d), 0); stroke(10, 255, 10); fill(10, 255, 10); ellipse(0, 0, 20, 20); } </pre>	
--	--

5. 【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は、ランダムに決めた色の四角形を表示し、マウスボタンが押されていないときには、灰色の四角形を表示するものです。なお、描画する四角形の枠線の色は黒です。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	実行例
<pre> color fColor; // 四角形の塗りつぶし色を保存している変数 void setup(){ size(400, 400); rectMode(CENTER); colorMode(HSB, 359, 99, 99); fColor = color(0, 0, 80); } void draw(){ background(0, 0, 99); stroke(__(a)); fill(__(b)); rect(width/2, height/2, width/2, height/2); } void __(c) () { __(d) = color(random(__(e)), 99, 99); } void mouseReleased(){ __(f) = color(__(g) , __(h) , __(i)); } </pre>	

6. 【目コピ問題】 下のプログラムは、ストップウォッチのような機能をもったプログラムです。マウスボタンをクリックすると時間の計測を始め、もう一度マウスをクリックすると計測を中止します。マウスのクリック時に、その時間を `startTime` に保存し、その時間と現在の時間との差を計算し、経過時間を数字で表示している。なお、秒未満の部分は数字でなく、扇型の表示で表している（映画などのカウントダウンで出てくるようなやつ）。この扇型は、時計の 12 時の場所から描画スタートします。そのスタートの場所の角度は $3 * \text{PI} / 2 (= 3 * \text{HALF_PI})$ となり、時間に応じた描画角度は `map` 関数を用いて計算しています。

未完成のプログラム	実行例
<pre> int startTime; (a) doing = false; // 時間計測中は true にする PFont font; void setup(){ size(400,400); smooth(); font = (b); textFont(font,48); textAlign(CENTER); } void draw(){ background(255); if(doing){ int ms = millis()- (c); fill(0); text(millisToMS((d)),width/2,height/5); float theta = map((e) ,0,1000,0,2*PI); stroke(0); fill(50); arc(width/2,0.6*height, width/2,height/2, (f) , (g) + (h)); } } (i) millisToMS(int ms){ int ts = ms / (j); // ミリ秒を秒に変換 int s = (k); // ts から秒の部分を求める int m = (l); // ts から分の部分を求める (m) str(m) + "m" + str(s) + "s"; } void (n) (){ if(doing == false){ doing = (o); startTime = (p); }else{ doing = (q); } } </pre>	<div data-bbox="1109 712 1342 943" style="border: 1px solid black; padding: 5px; text-align: center;"> 0m0s  </div> <div data-bbox="1109 949 1342 1180" style="border: 1px solid black; padding: 5px; text-align: center;"> 0m2s  </div>

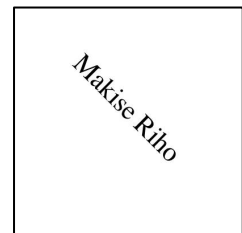
7. 【工夫問題】 ランダムな位置に四角形か円を表示し、その図形をクリックしたら、別の位置に図形を表示するという動作を繰り返すプログラムを作成してください。また、図形をクリックした回数を表示するような機能も付け加えてください。

8. 【工夫問題】 問 6 のプログラムでは、時間の計測を始めてから、再びマウスをクリックすると表示が消えてしまいます。そこで、表示が消えないようなプログラムを作成して下さい。つまり、「マウスクリック→計測開始→マウスクリック→表示したまま計測終了

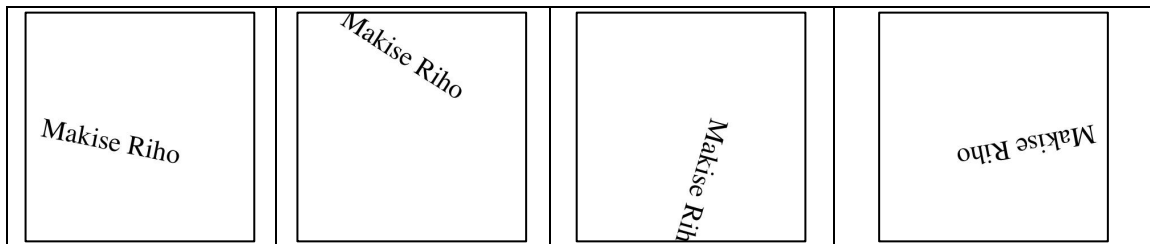
→マウスクリック→経過時間を 0 として、計測開始…」を繰り返すようなプログラムとしてください。

9. 【工夫問題】問 6 または問 8 のプログラムを変更し、秒未満の部分を表している扇型の部分は、「最初の 1 秒は黒で扇を描画→次の 1 秒は白で扇を描画（見かけ上は黒の部分が減っていく）」となるようにプログラムを変更してください。実行例で確認してみてください。

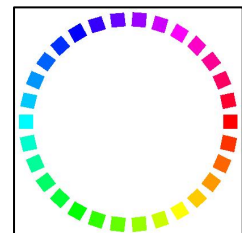
10. 【目コピ問題】右図のように文字列を 45 度傾けて表示するプログラムを作成して下さい。表示する文字列は何でもかまいません。



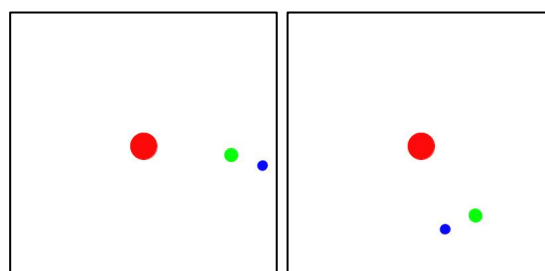
11. 【目コピ問題】下図のようにマウスカーソルの X 座標の位置に応じて、文字列の回転角度を変化させて文字列を表示するプログラムを作成して下さい。マウスカーソルが一番左にいるときは 0 度、一番右にいるときは 180 度回転しています。



12. 【目コピ問題】右図のように正方形を円周上に配置するプログラムを作成して下さい。明度と彩度は一定で、角度によって色相が変化しています。色相環のような色の配置です。



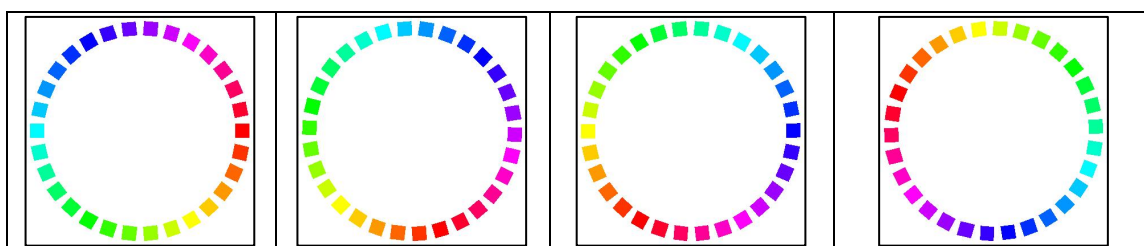
13. 【目コピ問題】ウインドウの中心に赤色の円盤を表示し、その外側を緑色の円盤が回転し、その緑色の円盤の周りを青色の円盤が移動するようなプログラムを作成して下さい。



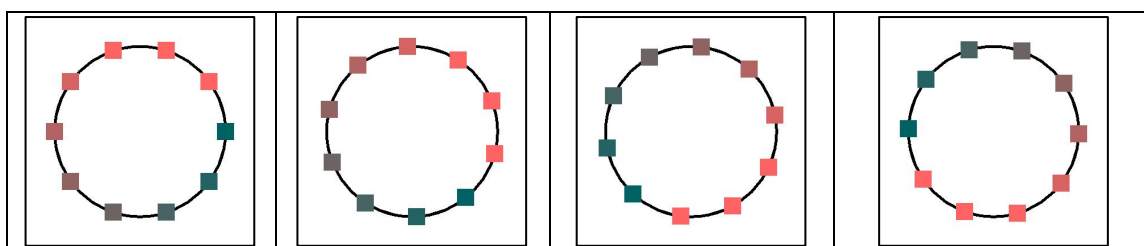
14. 【目コピ問題】 下図のように文字列が左から右に連続して、途切れることなく表示するプログラムを作成して下さい。プログラム中の size 関数で指定しているウインドウの大きさを変更しても、きちんと動作するプログラムとして作成して下さい。

Ri	RihoRihoR
RihoRihoRihoRihoRihoRihoR	hoRihoRihoRihoRihoRihoRiho

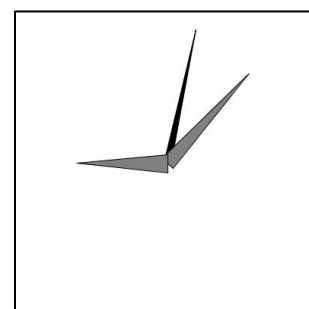
15. 【目コピ問題】 問 13 で作成した図形を回転させるプログラムを作成して下さい。



16. 【目コピ問題】 下図のように円周上に配置した正方形を回転させるプログラムを作成して下さい。正方形の向きが問 15 と異なっていることに注意して下さい。



17. 【工夫問題】 右図のようなアナログ時計を作成して下さい。針の形や、背景の画像などに自由に工夫して作成して下さい。



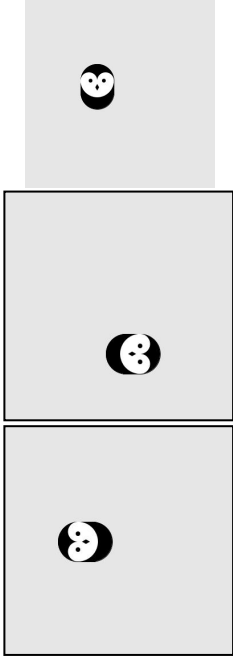
18. 【工夫問題】 5 つの長方形を表示し、その中から正解の長方形を選ぶようなゲームを作成して下さい。

19. 【自己確認問題】 中間試験の準備をして下さい。

宿題

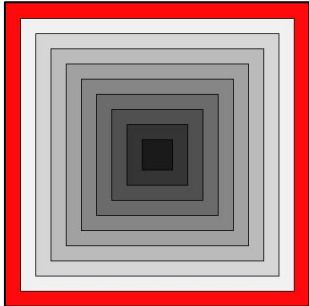
宿題はキャリアポートフォリオ上で解答して下さい。

1. 【目コピ問題】 下のプログラムは、マウスをクリックするたびにフクロウが時計周りに90度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。

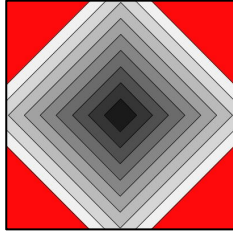
未完成のプログラム	実行例
<pre>float rot; // フクロウの回転角度を保存している変数 void setup() { size(400, 400); smooth(); rot = 0; } void drawLeftEye(float x,float y){ noStroke(); fill(255); ellipse(x-3.5,y,35,35); fill(0); ellipse(x,y,8,8); } void drawRightEye(float x,float y){ noStroke(); fill(255); ellipse(x+3.5,y,35,35); fill(0); 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); translate((a) , (b)); rotate((c)); drawOwl(0,0); }</pre>	 <p>The execution examples show three instances of the owl's head, each on a light gray rectangular background. The first instance is upright (0 degrees rotation). The second instance is rotated 90 degrees clockwise, appearing as a profile. The third instance is rotated 180 degrees clockwise, appearing upside down.</p>

<pre>void __ (d) __{ rot = rot + __ (e) __; }</pre>	
---	--

2. 【目コピ問題】 下のプログラムは、背景が赤色、外側の正方形は一辺が 360、内側に行くに従って、正方形の一辺の長さは 40 ずつ減っていきます。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>size(400,400); smooth(); rectMode(CENTER); background(255,10,10); stroke(0); int d = __ (a) __; for(int s=__ (b) __; __ (c) __; s -= __ (d) __){ fill(__ (e) __); rect(width/2,height/2, __ (f) __, __ (g) __); }</pre>	

3. 【目コピ問題】 下のプログラムは、背景が赤色、外側の正方形は一辺が 360、内側に行くに従って、正方形の一辺の長さは 40 ずつ減っていきます。また、正方形は 45 度 (PI/4) 傾いています。空欄に追加して、プログラムを完成させて下さい。

未完成のプログラム	実行例
<pre>size(400,400); smooth(); rectMode(CENTER); background(255,10,10); stroke(0); translate(__ (a) __, __ (b) __); rotate(__ (c) __); int d = __ (d) __; for(int s=__ (e) __; __ (f) __; s -= __ (g) __){ fill(__ (h) __); rect(0,0, __ (i) __, __ (j) __); }</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。ノートパソコンを含めて持ち込み可です。ただし、人の持ち込みは不可です。ネットに接続していることが判明した場合には、カンニングと見なします。