

2019 年度情報メディア基盤ユニット

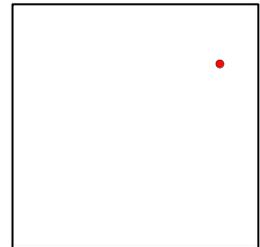
6 月 21 日分課題と宿題

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。【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。一部の問題はキャリアポートフォリオでも解答を入力して下さい。

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。
特に、以下に 4 個のサンプルプログラムは実行して見て下さい。

8-2,8-6,8-7,8-13

右の実行例はどのサンプルプログラムの実行例でしょうか？

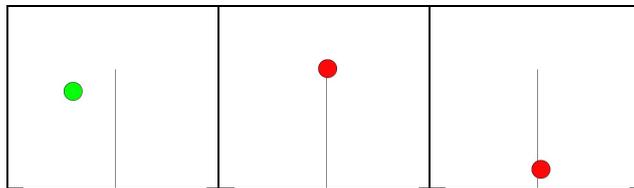


2. 【目コピ問題】下のプログラムは、マウスをドラッグしている際には”Dragging...”と表示し、単にマウスを移動させている時には”Moving...”と表示するプログラムです。空欄を埋めて、プログラムを完成させて下さい。正しく動くプログラムをキャリアポートフォリオ上にも提出して下さい。

未完成のプログラム	実行例
<pre>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);</pre>	

<pre> msg = <u>(d)</u>; } void mouseDragged(){ displaying = <u>(e)</u>; msg = <u>(f)</u>; } </pre>	
--	--

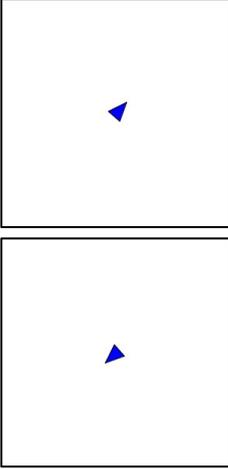
3. 【目コピ問題】 ウィンドウの下から垂直に立っている線分と円が表示されている。円はマウスの動きに合わせて移動し、線分の位置は固定されている。円と線分がぶつかった場合には、円の色を赤色で表示し、そうでない場合には円を緑色で表示するようなプログラムを作成して下さい。正しく動くプログラムをキャリアポートフォリオ上にも提出して下さい。



4. 【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。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。

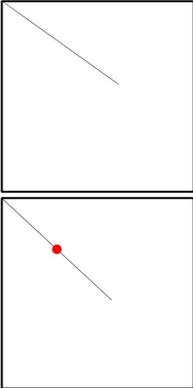
未完成のプログラム	実行例
<pre> color fColor;//四角形の塗りつぶし色を保存している変数 void setup(){ size(400,400); rectMode(CENTER); colorMode(HSB,359,99,99); fColor = color(0,0,80); } void draw(){ background(0,0,99); stroke(<u>(a)</u>); fill(<u>(b)</u>); rect(width/2,height/2,width/2,height/2); } void <u>(c)</u>(){ <u>(d)</u> = color(random(<u>(e)</u>),99,99); } void mouseReleased(){ <u>(f)</u> = color(<u>(g)</u>,<u>(h)</u>,<u>(i)</u>); } </pre>	

5. 【目コピ問題】 次のプログラムは、左右のカーソルキーが押されている時のみ、ウィンドウの中央に表示されている三角形が回転するプログラムです。左カーソルキー（←）を押すと反時計回り、右カーソルキー（→）を押すと時計回りに回転します。空欄を埋めて、プログラムを完成させて下さい。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 angle; // 回転角度を表す変数 float angleVel; // 回転角度の増分を指定する変数 void setup(){ size(300,300); angle = 0; angleVel = 0; } void draw(){ background(255); translate(width/2,height/2); (a) ((b) (angle)); fill(0,0,255); triangle(0,-20,-10,5,10,5); angle += (c) ; } void keyPressed(){ if(key == (d)){ if(keyCode == (e)){ angleVel = -1; }else if(keyCode == (f)){ angleVel = 1; } } } void (g) (){ if(key == (d)){ if(keyCode == (e) keyCode == (f)){ angleVel = (h) ; } } } }</pre>	

6. 【目コピ問題】 次のプログラムは、ウィンドウの左上から、マウスカーソルがいる位置に線分を引くプログラムです。さらに、マウスをクリックすると、クリックした時のマウスカーソルの位置に向かって、移動する赤い円を表示するものです。空欄を埋めて、プログラムを完成させて下さい。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> boolean alive= <u>(a)</u>; //赤い円を表示するかを決める変数 float xPos = 0; // 赤い円の中心座標 float yPos = 0; float vx = 0; // 赤い円の移動ベクトル float vy = 0; float speed = 0; // 赤い円の速さ void setup(){ size(400,400); smooth(); } void draw(){ background(255); stroke(0); line(0,0,mouseX,mouseY); noStroke(); fill(255,0,0); if(alive){ ellipse(<u>(b)</u> , <u>(c)</u> ,20,20); xPos += <u>(d)</u> ; yPos += <u>(e)</u> ; // ウィンドウの外に出たら赤い円の表示をやめる if((xPos-20/2 >= width) (yPos-20/2 >= height)){ alive = <u>(f)</u> ; } } } void mouseClicked(){ float theta = atan2(mouseY,mouseX); xPos = 0; yPos = 0; speed = 2; vx = speed * cos(<u>(g)</u>); vy = speed * sin(<u>(g)</u>); alive = <u>(h)</u> ; } </pre>	
---	---

7. 【目コピ問題】 次のプログラムは、問 7 のプログラムに変更を加え、左右のカーソルキーが押されている時のみ、ウィンドウの中央に表示されている三角形が回転し、上カーソルキー（↑）が押されている間は、前に進むようなプログラムです。また、rキーまたは R キーを押すと、三角形の位置などが初期状態となります。空欄を埋めて、プログラムを完成させて下さい。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
-----------	-----

```

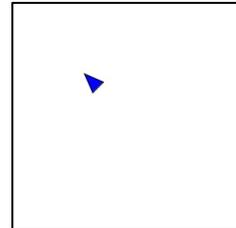
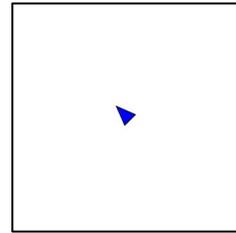
float angle; // 回転角度を表す変数
float angleVel; // 回転角度の増分を指定する変数
float xPos; // 三角形の位置座標
float yPos;
float vx; // 三角形の移動ベクトル
float vy;
float speed; // 三角形の速さ
(a) initTri(){
    angle = 0;
    angleVel = 0;
    xPos = width/2;
    yPos = height/2;
    vx = vy = 0;
    speed = 0;
}
void setup(){
    size(300,300);
    initTri();
}
void drawShip(){
    pushMatrix();
    translate(xPos,yPos);
    (b);
    fill(0,0,255);
    triangle(0,-20,-10,5,10,5);
    popMatrix();
}

void draw(){
    background(255);
    drawShip();
    angle += angleVel;
    xPos += (c);
    yPos += (d);
}

void keyPressed(){
    if(key == (e) || key == (f)){
        (g);
    }else if(key == CODED){
        if(keyCode == LEFT){
            angleVel = -1;
        }else if(keyCode == RIGHT){
            angleVel = 1;
        }else if(keyCode == UP){
            float theta = radians((h));
            speed = 2;
            vx = speed * cos(theta);
            vy = speed * sin(theta);
        }
    }
}

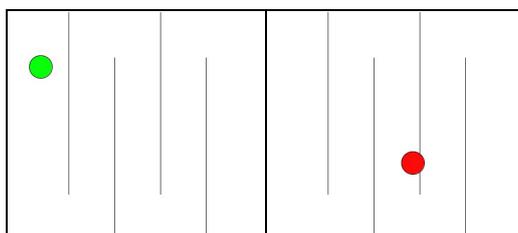
void keyReleased(){
    if(key == CODED){
        if(keyCode == LEFT || keyCode == RIGHT){

```



<pre> angleVel = 0; }else if(keyCode == UP){ vx = <u>(i)</u>; vy = <u>(j)</u>; speed = 0; } } } </pre>	
--	--

8. 【目コピ問題】 ウィンドウの上部と下部から垂直に何本かの線分を表示し、その線分につつからずにウィンドウの左から右に円を移動させるゲームを作成して下さい。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。



9. 【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。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> int startTime; <u>(a)</u> doing = false; // 時間計測中は true にする PFont font; void setup(){ size(400,400); smooth(); font = <u>(b)</u>; textFont(font,48); textAlign(CENTER); } void draw(){ background(255); if(doing){ int ms = millis()- <u>(c)</u>; fill(0); text(millisToMS(<u>(d)</u>),width/2,height/5); </pre>	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <p>0m0s</p>  </div> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-top: 10px;"> <p>0m2s</p>  </div>

<pre> 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>	
--	--

10. 【工夫問題】 問 7 のプログラムで三角形ではなく、それらしい画像を表示するように変更して下さい。また、背景もそれらしい画像に変更して下さい。適当な画像が無い場合には、web ページからダウンロードして下さい。

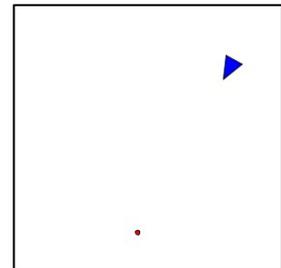


11. 【工夫問題】 ランダムな位置に四角形か円を表示し、その図形をクリックしたら、別の位置に図形を表示するという動作を繰り返すプログラムを作成して下さい。また、図形をクリックした回数を表示するような機能も付け加えてください。

これ以降の問題は余裕のある人向けです。

12. 【工夫問題】 問 9 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下さい。
13. 【工夫問題】 問 9 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下さい。

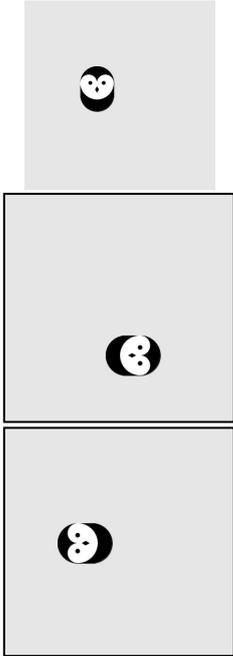
14. 【工夫問題】 マウスマウスカーソルの位置に三角形上の物体を表示し、マウスをクリックすると、この三角形の頂点から、真上に円盤状の球を発射するようなプログラムを作成して下さい。もし可能なら、発射された弾が消えるまで、次の弾が発射出来ないようにして下さい。
15. 【工夫問題】 問 14 に、ウインドウの上方から赤色の円が落ちてくるような処理を加えて下さい。さらに、赤色の円と弾との衝突判定を行う処理を付け加え、衝突していると判定した場合には、赤色の円の位置をウインドウ最上部に移動させ、再び下方向に移動するようにすること。また、衝突回数をカウントし、その回数を表示するようにして下さい。
16. 【工夫問題】。問 11 を変更し、一定時間図形が表示されたら、別の位置に移動するようにプログラムを変更して下さい。
17. 【工夫問題】。問 16 を変更し、一定回数時間切れで図形が別の位置に表示されるという動作が行われたら、“終了メッセージと得点”が表示されるようにして下さい。
18. 【工夫問題】 問 14 を変更して、マウスでなくキーボードで“自機”を操作できるようにして下さい。
19. 【工夫問題】 問 7 のプログラムを変更して、z キーまたは Z キーを押したら、三角形の頂点から赤い円が発射されるようなプログラムを作成して下さい。



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); } void (d) (){ rot = rot + (e) ; } }</pre>	 <p>The execution examples show three instances of the owl's face on a light gray background. The top instance is upright (0 degrees rotation). The middle instance is rotated 90 degrees clockwise. The bottom instance is rotated 180 degrees clockwise (upside down).</p>