# 2015 年度情報メディア基盤ユニット
# 7 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。【自己確認問題】は先生や TA の人にチェックしてもらう必要はありません。できたら先生か TA の人に確認をしてもらって下さい。

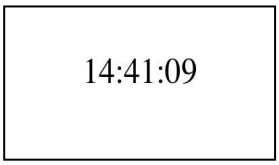
1. 【目コピ問題】数字を表示する際に、表示桁数を指定して表示を行いたい場合があります。その目的のために、Processing では nf 関数があります。この関数は、数値を指定された桁数の文字列に変換するものです。123 とい値を 5 桁の文字列に変換したいときには、nf(123,5)とすると、”00123”という 5 文字の文字列となります。なお、通常のこの関数には、非負の数を対象としています。

   この関数を使ったものがサンプル 1 です。このサンプルは、マウスをクリックするとストップウオッチのように経過時間を計るものです。

| サンプルプログラム 1 | 実行例 |
|---|---|
| PFont font;<br>int startTime_msec;<br>boolean counting = false;<br>void setup(){<br>  size(400,200);<br>  smooth();<br>//font は各自のものに変更してください。<br>  font = loadFont("Serif-48.vlw");<br>  textFont(font,48);<br>  textAlign(CENTER);<br>  fill(0);<br>}<br>void mouseClicked(){<br>  counting = true;<br>  startTime_msec = millis();<br>}<br>void draw(){<br>  int t=0;<br>  background(255);<br>  if(counting){<br>  t = millis();<br>  }<br>  int ms = t % 1000;<br>  int s = t/1000;<br>  int m = s/60;<br>  String elapsed=nf(m,4)+":"+nf(s%60,2)+":"+nf(ms,3);<br>  text(elapsed,width/2,height/2);<br>} | 0000:00:000<br><br>0001:53:063 |

この nf 関数を使って、14 時 41 分 09 秒であれば、「14:41:09」と表示されるなプログラムを作成したい。下の未完成のプログラムの空欄を埋めて、プログラムを完成させて

ください。

| 未完成プログラム | 実行例 |
|---|---|
| PFont font;<br><br>void setup(){<br>  size(400,200);<br>  smooth();<br>  font = __(a)__;<br>  textFont(font,48);<br>}<br><br>void draw(){<br>  background(255);<br>  int h = hour();<br>  int m = minute();<br>  int s = second();<br>  String time = __(b)__+":"+__(c)__+":"+__(d)__;<br>  fill(0);<br>  textAlign(CENTER);<br>  text(time,width/2,height/2);<br>} |  |

nf 関数の使用方法

| 関数の呼び出し | 呼び出し例 | 戻り値 |
|---|---|---|
| nf(整数値,桁数) | nf(123,5) | “00123” |
| nf(実数値,整数部分桁数,小数部分桁数) | nf(3.1415,3,5) | “003.14150” |

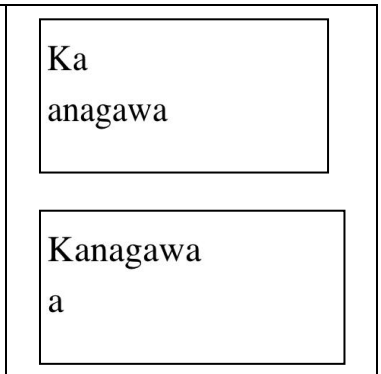
2. 【目コピ問題】String 型(String クラス)は、いつかの便利にメソッドを持っています。下の呼び出し例では、「String msg=”Makise Riho”;」となっています。

| メソッド | 機能 | 呼び出し例 | 戻り値 |
|---|---|---|---|
| length() | 文字列の長さを求めるます。 | msg.length() | 11 |
| substring(start) | Start 番目から最後までの<br>文字取り出します。配列と同じように<br>先頭の文字が 0 番目です。 | msg.substring(3); | “ise Riho” |
| substring(start,pos) | Start 番目から pos-1 番目のまでの文字列を取り出します。 | msg.substring(2,4) | “ki” |
| | | msg.substring(0,3) | “Mak” |

これらのメソッドを利用すると、次のようなプログラムを作成することが出来ます。このプログラムは、0.5 秒ごとに表示される文字数が増えたり減ったりするというものです。

| サンプルプログラム 2 | 実行例 |
|---|---|
| String msg = "Kanagawa"; | |

```
PFont font;
void setup(){
  size(400,200);
//font は各自のものに変更してください。
  font = loadFont("Serif-48.vlw");
  textFont(font,48);
  fill(0);
}
void draw(){
  background(255);
  fill(0);
  int pos = ((millis()/1000) % msg.length());
  text(msg.substring(0,pos+1),10,height/3);
  text(msg.substring(pos),10,2*height/3);
}
```

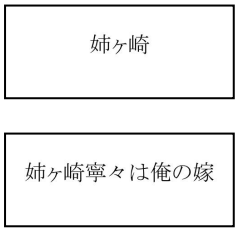



次のプログラムはマウスの X 座標(mouseX)値を利用して、文字列の表示範囲を決めるものです。マウスが左端にいるときは無表示、右端にいるときには全てを表示となっています。空欄を埋めて、プログラムを完成させてください。

未完成プログラム

```
//msg は各自のものに変更してください。
String msg = "姉ヶ崎寧々は俺の嫁";
PFont font;
void setup(){
  size(500,200);
//font は各自のものに変更してください。
  font = createFont("MS-PMincho",48);
  textFont(font,48);
  textAlign(CENTER);
  fill(0);
}
void draw(){
  background(255);
  fill(0);
  float dx = width/msg.length();
  int pos = round(mouseX/dx);
  text(msg.__(a)__(__(b)__,__(c)__),
       width/2,height/2);
}
```

実行例




プログラム中の createFont 関数は、loadFont 関数に似た動作をする関数です。最初の引数で指定した名前のフォントを利用して、最後の引数で指定した大きさのフォント情報を作り出すものです。createFont 関数の最初の引数に日本語フォントを指定すると、日本語の表示を行うことが出来るようになります。また、直接日本語の入力が出来ないので、メモ帳などで入力した文字列をコピー＆ペーストで貼り付けることで、入力をしています。

3. 【工夫問題】次の未完成プログラムは、RPG のマップのような画面を表示し、カーソルキーを押すと、地図がスクロールするというものです。未完成のプログラムは、上矢印キーでの移動にしか対応していません。左右下キーでの移動もできるようにプログラムを変更して下さい。可能なら、地図の中央部分にキャラクタを表示してみて下さい。地図の画像データは、ぴぽや http://piposozai.blog76.fc2.com/のデータを使っています。読み込んでいる画像ファイルは、1 枚の画像に複数のパーツ（32X32 の画像）が含まれています。画像ファイルから 1 つのパーツを取り出し表示するために、copy 関数を使っています。Copy 関数は copy(img,sx,sy,sw,sh,x,y,w,h)と実行すると、画像 img の一部をウインドウ上に表示するものです。画像 img 上の点(sx,sy)を左上の座標、幅 sw,高さ sh とする矩形領域を、ウインドウ上の点(x,y)を左上とする幅 w、高さ h の場所に表示します。なお、画像ファイルはいつものページからダウンロードできるようにしておきます。

```
PImage img;
int mapID[][];

int xSize;
int ySize;

void setup(){
  size(640,480);
  img = loadImage("pipoya_mcset1_at_grass2.png");
  xSize = width/32;
  ySize = height/32;
  mapID = new int[xSize][ySize];
  for(int i = 0;i < xSize;i++){
    for(int j= 0;j < ySize;j++){
      mapID[i][j] = int(random(10));
    }
  }
}

void draw(){
  background(255);
  for(int i=0;i < xSize;i++){
    for(int j=0;j < ySize;j++){
      int x = mapID[i][j] % 2;
      int y = mapID[i][j] % 5;
      copy(img,32*x,32*y,32,32,32*i,32*j,32,32);
    }
  }
}

void keyPressed(){
  if(key == CODED){
    if(keyCode == UP){
      int[] tmp = new int[xSize];
      for(int i=0;i<xSize;i++){
        tmp[i] = mapID[i][ySize-1];
      }
```

```
      for(int j=ySize-1;j >= 1;j--){
        for(int i=0;i<xSize;i++){
          mapID[i][j] = mapID[i][j-1];
        }
      }
      for(int i=0;i< xSize;i++){
        mapID[i][0] = tmp[i];
      }
    }else if(keyCode == DOWN){
      // Insert here...
    }else if(keyCode == LEFT){
      // Insert here...
    }else if(keyCode == RIGHT){
      // Insert here...
    }
  }
}
```

4. 【工夫問題】次の未完成プログラムは、RPG のマップのような画面を表示しするものです。問 3 との違いは、読み込んだ画像から、画像をそれぞれのパーツに分け、その情報を PImage 型配列 groundImage に保存しています。パーツに分けるために、get 関数を利用しています。img.get(x,y,w,h)とすると、画像 img から左上(x,y)、幅 w、高さ h の矩形領域を切り出し、あたらな画像データとして利用できるようにします。groundImage の中には道のような画像も入っています。これらのデータを利用して、自分なりの地図を描いて下さい。ただし、道は繋がっていることが必要です。可能なら、カーソルキーで移動させたり、キャラクタを表示させてみて下さい。地図の画像データは、ぴぽや http://piposozai.blog76.fc2.com/のデータを使っています。

```
PImage[] groundImage;
int[][] mapID;
int xSize;
int ySize;

void setup(){
  size(640,320);

  PImage img0 = loadImage("pipoya_mcset1_at_grass2.png");
  PImage img1 = loadImage("pipoya_mcset1_at_gravel1b.png");
  groundImage = new PImage[20];
  for(int i=0;i < 10;i++){
    int x = i % 2;
    int y = i % 5;
    groundImage[i] = img0.get(32*x,32*y,32,32);
    groundImage[i+10] = img1.get(32*x,32*y,32,32);
  }
  xSize = width/32;
  ySize = height/32;
  mapID = new int[xSize][ySize];
  for(int i = 0;i < xSize;i++){
    for(int j= 0;j < ySize;j++){
      mapID[i][j] = int(random(groundImage.length));
```

```
    }
  }
}

void draw(){
  background(255);
  for(int i=0;i < xSize;i++){
    for(int j=0;j < ySize;j++){
      println(mapID[i][j]);
      image(groundImage[mapID[i][j]],32*i,32*j);
    }
  }
  for(int i=0;i<xSize;i++){
    image(groundImage[i % groundImage.length],32*i,0);
  }
}
```

5. 「サンプルプログラム３：早撃ちゲーム」は、

1) 最初は開始画面が表示する。

2) 表示されている正方形（ターゲット）をクリックしたら、hit 表示がされる。

3) hit 表示は少し経ったら(0.5 秒)、再び正方形（ターゲット）を表示する。

4) 正方形（ターゲット）は 1 秒間表示されたら、別な場所に移動する。

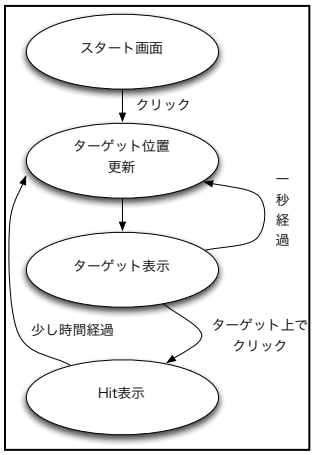
といった感じのゲームのようなプログラムです。

サンプルプログラム３：早撃ちゲーム

```
 1  PFont font;
 2  float xTarget;
 3  float yTarget;
 4  int targetWidth;
 5  int targetHeight;
 6  color targetColor;
 7
 8  int GAME_READY = 0;
 9  int GAME_UPDATING = 1;
10  int GAME_RUNNING = 2;
11  int GAME_HIT = 3;
12
13  int currentState;
14  int time0_msec;
15
16  void startElapsedTime(){
17    time0_msec = millis();
18  }
19
20  int elapsedTime_msec(){
21    return millis()-time0_msec;
22  }
23
24  void setup(){
25    size(400,400);
26    smooth();
27    font = createFont("Serif",48);
```

```
  currentState = GAME_READY;
}

void updateTarget(){
  targetWidth = 50;
  targetHeight = 50;
  xTarget = random(targetWidth, width-targetWidth);
  yTarget = random(targetHeight, height-targetHeight);
  targetColor = color(10,10,255);
}

boolean isOnTarget(int x,int y){
  if((xTarget <= x && x < (xTarget+targetWidth)) &&
     (yTarget <= y && y < (yTarget+targetHeight))){
     return true;
   }else{
     return false;
   }
}

void showReadyMessage(){
  textAlign(CENTER);
  textFont(font,48);
  fill(0);
  text("Start to click",width/2,height/2);
}

void showHitMessage(){
  textAlign(CENTER);
  textFont(font,48);
  fill(255,10,10);
  text("Hit!!",width/2,height/2);
}

void showTarget(){
  stroke(targetColor);
  fill(targetColor);
  rectMode(CORNER);
  rect(xTarget,yTarget,
      targetWidth,targetHeight);
}

void draw(){
  background(255);
  if(currentState == GAME_READY){
    showReadyMessage();
  }else if(currentState == GAME_UPDATING){
    updateTarget();
    currentState = GAME_RUNNING;
    startElapsedTime();
  }else if(currentState == GAME_RUNNING){
    showTarget();
    if(elapsedTime_msec() >= 1000){
      currentState = GAME_UPDATING;
    }
  }else if(currentState == GAME_HIT){
    showHitMessage();
```

```
85      if(elapsedTime_msec() >= 500){
86        currentState = GAME_UPDATING;
87      }
88    }
89  }
90
91  void mouseClicked(){
92    if(currentState == GAME_READY){
93      currentState = GAME_UPDATING;
94    }else if(currentState == GAME_RUNNING){
95      if(isOnTarget(mouseX,mouseY)){
96        currentState = GAME_HIT;
97        startElapsedTime();;
98      }
99    }
100 }
```




上の 1)から 4)から、このプログラムには、下の表のような 4 つの状態（状況）があると考えられます。マウスをクリックしたり、一定時間経ったら、これらの状態間を遷移します。状態から状態への遷移は、右の図のようになっています。このような状態間の遷移の状態を表す図のことを状態遷移図と呼んでいます。このような状態遷移図を作成すると、全体の進行状況を捉えやすくなります。このような状態遷移図をプログラム化するには、一般的に 2 つのやり方があります。

1) 表示や入力の処理をする関数内(draw などや mouseClicked 関数など)で、状態を条件分岐(if 命令など)で状態を変更する。
2) 状態遷移クラスを作成し、各状態を子クラスとして継承する。

| 状態名 | 説明 | 状態定数名 |
|---|---|---|
| 開始画面 | 開始画面を表示している | GAME_READY |
| ターゲット位置更新 | 正方形（ターゲット）の位置を変更している | GAME_UPDATING |
| ターゲット表示 | 正方形（ターゲット）を表示している | GAME_RUNNING |
| Hit 表示 | Hit 表示を行っている | GAME_HIT |

このサンプルでは、1)の方法を利用してプログラムを作成しています。状態を表すために、各状態に数値を割り当て、状態を区別します。数字では、人間がどの状態かを識別ルことが難しいので、適当な変数名をつけた変数を利用しています。この値は変更さえ

ることがないので、定数と呼ばれています。この状態を識別する定数として、上の表の状態定数名という名前の変数を使用しています。また、現在の状態は currentState という変数に保存されています。

マウスをクリックして状態が変化する部分は、mouseClicked 関数で処理をしています。また、経過時間などで変化する部分は、draw 関数で処理をしています。

マウスクリック時の処理は、GAME_READY 状態であれば、直ちに GAME_UPDATE 状態に遷移します。また、GAME_RUNNING 状態であれば、マウスをクリックした場所が正方形（ターゲット）上であれば、GAME_HIT 状態に遷移します。経過時間を計るために、状態が遷移するときに、startElapsedTime 関数を呼び出し、elapsedTime_msec 関数で状態の経過時間をミリ秒単位で調べることが出来るようにしています。これらの処理は mouseClicked 関数内に書かれています。

GAME_UPDATE 状態であれば、正方形（ターゲット）の位置情報などを更新後、GAME_RUNNING 状態に遷移します。また、GAME_RUNNING 状態で、経過時間が 1 秒(1000 ミリ秒)経過したら、GAME_UPDATE 状態に遷移します。GAME_HIT 状態で、経過時間が 0.5 秒(500 ミリ秒)経過したら、GAME_UPDATE 状態に遷移します。これらの処理は、draw 関数内に書かれています。

【工夫問題】このプログラムに以下のような改良を加えて下さい。

(1) 正方形の表示時間を乱数で変更するようにして下さい。
(2) hit 回数を表示するようにして下さい。
(3) 5 回ミス hit したら、ゲームが終了するようにして下さい。
(4) hit 回数に応じて、正方形の表示時間が短くなるようにして下さい。
(5) 単純な正方形ではなく、画像や別な形となるようにして下さい。
(6) 効果音を付け加えて下さい。
(7) 繰り返しゲームが出来るようにし、開始画面にそれまでの最大 hit 回数が表示されるようにして下さい。

6. 【工夫問題】最終課題でどのようなものを作るのかを説明するレポートを作成して下さい。イラストや図などが入っていても OK です。提出は来週の金曜日です。これは宿題も兼ねています。

## 宿題

今週は問題を解くという形での宿題はありません。最終課題でどんなプログラムを作るかを考えて下さい。ただ、頭の中で考えるだけでなく、絵や文章として形にしてみて下さい。