Examen de 2nde session 2012-2013 (2h, barème /21)

1 Dessin&Animation (7 pts)

Dans cet exercice, il vous est demandé de fournir un certain nombre d'éléments nécessaires à la réalisation d'une application écrite en langage Processing.

```
float x[] = new float[64];
float y[] = new float[64];
float r[] = new float[64];

void setup (){
  size(800, 600);
  for (int i=0; i<64; i++){
    x[i] = random(300, 500);
    y[i] = random(400, 600);
    r[i] = random(2.0, 10.0);
  }
}

void draw(){
  background(255, 255, 196);
  fill(255, 255, 128);
  for (int i=0; i<64; i++){
    ellipse (x[i], y[i], r[i], r[i]);
    y[i] = y[i]-r[i]/4;
    if (y[i]<0) y[i] = random(400, 600);
  }
}}
```

1.1 ***(1pt)*** *Quel est la couleur de fond (en français) du programme ci-dessus?*

1.2 ***(2pts)*** *Décrivez ce que dessine le programme .*

1.3 ***(2pts)*** *Décrivez l'animation du programme.*

1.4 ***(2pts)*** *Que faut-il modifier dans le programme pour que l'animation se fasse dans l'autre sens (effet miroir par rapport à l'axe où se fait l'animation).*

2 Animation (7 pts)

On souhaite créer une animation de pluie qui tombe dans une flaque d'eau. Nous vous demandons d'écrire un programme Processing qui affiche des cercles concentrique bleus ciel sur un fond bleu-roi. Plus les cercles grandiront plus la couleur des cercles seront proche de la couleur de fond. Lorsque la taille et la couleur atteindront un certain seuil le cercle apparaitra ailleurs de taille minimale (1). En même temps 64 cercles doivent s'animer comme nous venons de le décrire (à défaut vous pouvez obtenir 2/3 des points en en affichant qu'un seul) mais chaque cercle a une taille initiale différente (tous les cercles atteignent donc le seuil a des moments différents).

2.1 (6pts) donner le code des fonctions setup() et draw()

2.2 ***(1pt)*** *Que Faut-il modifier pour que les cercles aient une apparence ovale (deux fois plus large que haut) ?*

3 Traitement d'image (7 pts)

Soit la fonction suivante qui copie une image (src) sur une autre (dst) de même taille sans tenir compte de la transparence de l'image src :

```
void blitimg(PImage dst, PImage src) {
  dst.loadPixels();
  for (int j=0; j<src.h; j++) {
    for (int i=0; i<src.w; i++) {
      int c = src.pixels[i+j*src.width];
      int r = red(c);
      int g = green(c);
      int b = blue(c);
      dst.pixel[i+j*dst.width] = color(r,g,b, 255) ;
    }
  }
  dst.updatePixels();
}
```

3.1 ***(4pt)*** *Refaire une nouvelle fonction de recopie blitimg3 qui prenne trois images de même taille (Source=**`src`**, Destination=**`dst`** et Masque = **`mask`**)). L'image résultat (**`dst`**) devra être identique à la source (y compris la transparence) là où le masque est blanc et complètement transparente là où le masque est noir.*

3.2 ***(4pt)*** *Que faut-il modifier pour que la transparence du résultat soit égale au produit de la transparence normalisée de la source et de la luminosité du masque (i.e si le masque est de luminosité 64 sur 255 a un endroit ou la source q une transparence de 192/255 alors la transparence résultante sera de 192*64/255)*