

NICOLE SCHULTHESS  
IAD 1. SEMESTER PROCESSING

ZUSAMMENFASSUNG 16. - 30. SEPTEMBER 2013

Wir konzentrieren uns die ersten zwei Wochen auf das Processing. Das Sprechtempo im Unterricht empfand ich als schnell, (neues Voci im Englisch) die Erklärungsweise jedoch sehr angenehm.

Schritt für Schritt mache ich mich mit dem Programmieren vertraut.

Am Anfang übte ich Ellipsen, Linien, Farben, Grössen, Positionen, Texte usw. Hier eine Übung in der viele der Ersten Schritte vorkommen.

HELLO WORLD:

```
char letter;
String words = "HELLO WORLD";

void setup() // funktion
{
  size(700, 700); // def. fenstergroesse
  randomSeed(millis()); // noch zufälliger
  textFont(createFont("Georgia", 12)); // textart und grösse

  smooth(); // aktiviere antialiasing
  strokeWeight(4); // linienbreite
  stroke(0, 0, 0, 150); // linienfarbe / 4. position = transparenz
  noLoop(); // nicht automatisch durchlaufen
}

void draw() // funktion
{
  background(0); // 0 = komplett schwarz
  textSize(98); // grösse
  fill(255, 255, 255); // füllung sprich farbe
  text(words, 50, 120, 540, 300); // siehe words funktion

  pushMatrix(); // momentaufnahme speichern
  translate(350, height -5); // height = fensterhöhe
  wurzel(7); // siehe funktion wurzel (7 = häufigkeit/ dichte)
  popMatrix(); // weiter machen
}

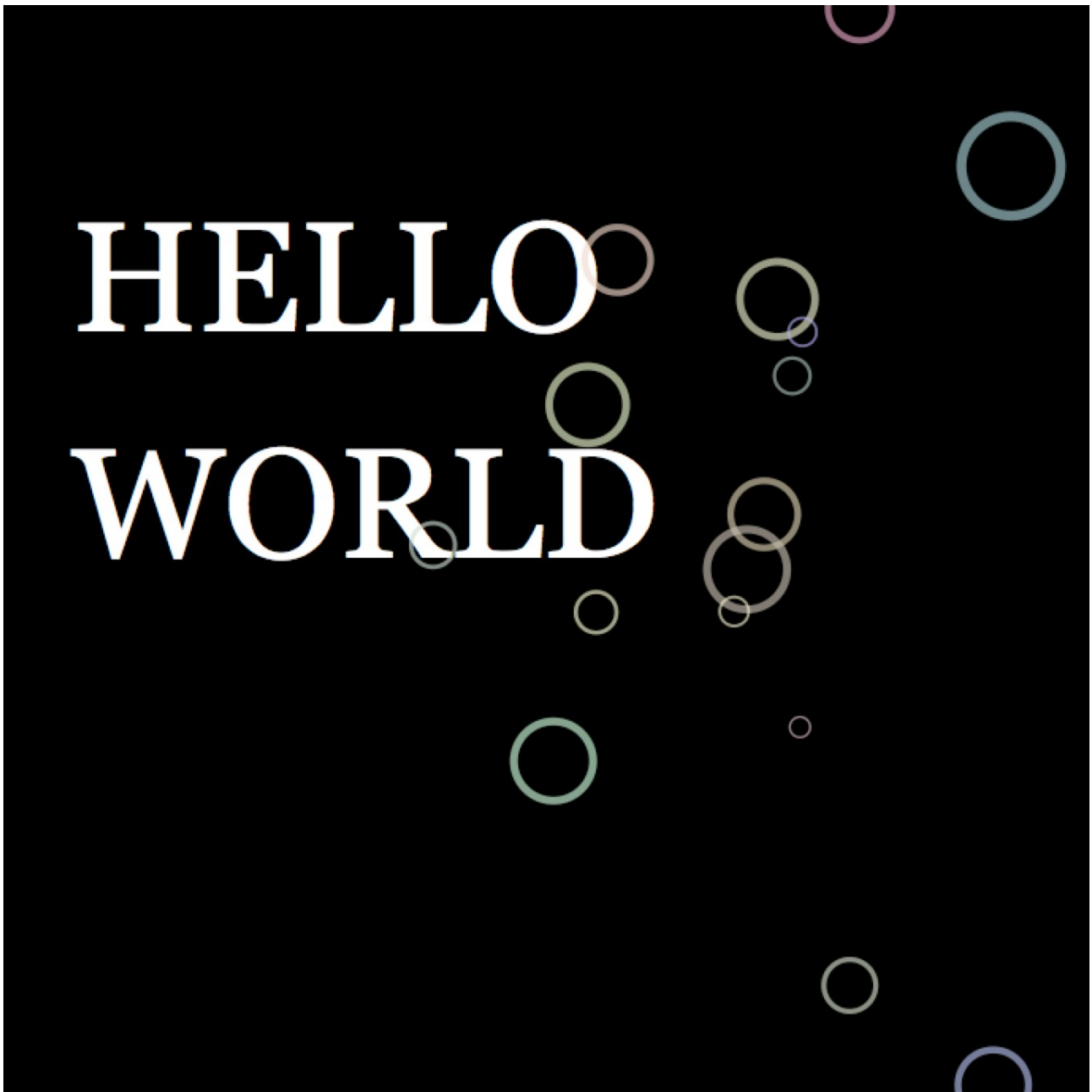
void mousePressed() // funktion
{
  redraw(); // beim klicken nochmals die funktion draw ausführen
}

void wurzel(int tiefe)// definition funktion wurzel
{
  if (tiefe <= 0) // bis zum ende
  {
    pushStyle(); // momentaufnahme anhalten // zeichne kreischen:
    int clr = (int)random(150, 255); // farbe random von bis
    int clr2 = (int)random(150, 255); // "
    int clr3= (int)random(150, 255); // "
    stroke(clr, clr2, clr3, 150); // abwechslung von den drei obigen
    // random varianten
    strokeWeight(10); // dicke kreis also ellipse rand
    noFill(); // keine füllung, sonst wird es ein ausgefüllter kreis
    ellipse(100, 0, 100, 100); // ist das unsichtbare(weil no Fill) zentrum
    //vom farbigen kreis (der stroke)
    popStyle(); // und weiter gehts...
    return;
  }
}
```

```
int x;
int y;
int count = (int)random(1, 13); // zufällige anzahl von kreisen zwischen 1 und 13
for (int i = 0; i < count; i++) //
{
  x = (int)random(-300, 300); // position von den kreisen - aufgangslage
  y = (int)random(-600, 600); // position von den kreisen - ausgangslage

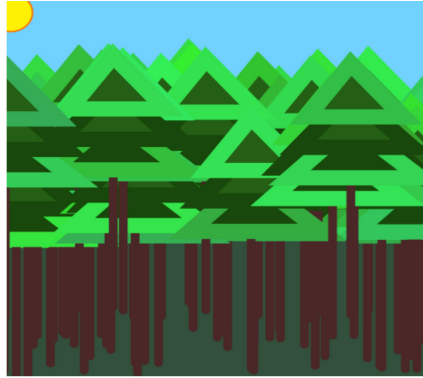
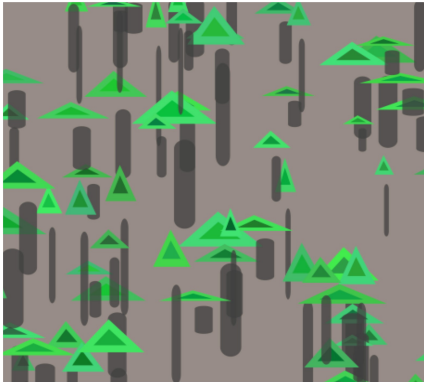
  pushMatrix(); // anhalten, bewegung reinbringen und wieder fortsetzen
  translate(x, y);
  scale(random(.3, .95));
  wurzel(tiefe-6);
  popMatrix();
}
}
```

So verändern sich die Kreislein in der Farbe und in der Position sowie in der Anzahl und Grösse. Nach einiger Übung, fühle ich mich mit den Anfängen jetzt sicher.



## THE FORRESTS

Das folgende Bild links, zeigt einen einfacheren- und dann rechts einen animierten Wald. Beide möchte ich als Bild in meine „Lernzusammenfassung“ integrieren, da ich sehr viel gelernt habe bei diesen zwei Übungen. Angefangen beim farblichen aufteilen vom Hintergrund, über die Benutzung von Push und Popmatrix bis zu Scale und Translate.



## THE SOABUBBLEMACHINE

Diese „Seifenblasenmaschine“ ist beim weiterentwickeln der Aufgabe „Japanese Tree“ entstanden.

```
char letter;
String words = "SOAP BUBBLE MACHINE";

void setup()
{
  size(700, 700); // def. fenstergroesse
  randomSeed(millis()); // seed random
  textFont(createFont("Georgia", 12));
  smooth(); // aktiviere antialiasing
  strokeWeight(4); // linienbreite
  stroke(0, 0, 0, 150);
  noLoop();
}

void draw()
{
  background(178, 225, 255);
  fill(88, 66, 66);
  rect(0, 400, 700, 600); // unterer teil hintergrund
  fill(245, 62, 62);
  rect(300, 280, 160, 40, 10); // roter deckel
  fill(111, 109, 107);
  rect(300, 350, 160, 100, 10); // unterer teil der maschine
  rect(300, 310, 160, 50); // mittlerer teil der maschine
  fill(255, 255, 255);
  rect(302, 312, 156, 5); // verschluss
  fill(0);
  text(words, 308, 290, 300, 300); // schrift

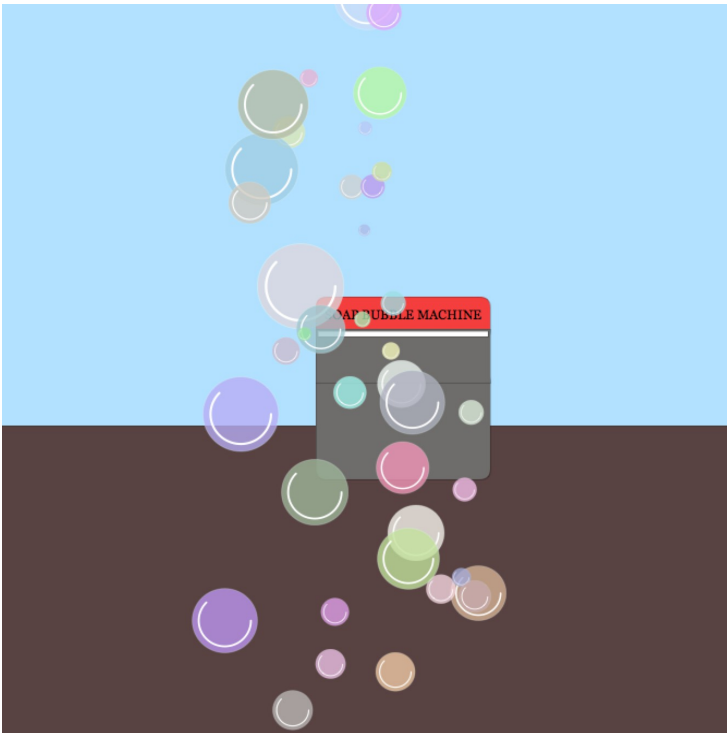
  pushMatrix();
  translate(width *.5, height - 10);
  wurzel(7);
  popMatrix();
}
```

```

void mousePressed()
{
  redraw();
}
// funktion
void wurzel(int tiefe)
{
  if (tiefe <= 0) // teste ob das ende erreicht worden ist
  {
    // zeichne seifenblasen
    pushStyle();
    int clr = (int)random(150, 255);
    int clr2 = (int)random(150, 255);
    int clr3= (int)random(150, 255);
    stroke(230, 255, 255);
    fill(clr, clr2, clr3, 200);
    ellipse(0, 0, 100, 100);
    noFill();
    stroke(255, 255, 255);//wie kann ich diese str transparent machen?
    strokeWeight(3);
    arc(0, 0, 80, 80, 0, PI+QUARTER_PI, OPEN);//der arc sollte auf der ellipse sein...
    popStyle();
    return;
  }
  int x;
  int y;
  int count = (int)random(1, 13);
  for (int i = 0; i < count;i++)
  {
    x = (int)random(-100, 100);
    y = (int)random(-600, 600);

    pushMatrix();
    translate(x, y);
    scale(random(.3, .95));
    wurzel(tiefe-6); // möchte weniger Äste
    popMatrix();
  }
}

```



## CATCHBIRDY

Dies ist der Code zur Endaufgabe. (Kommentar unten)

```
Catchbirdy catchb1;
Catchbirdy catchb2;
Catchbirdy catchb3;
Catchbirdy catchb4;
Catchbirdy catchb5;
Catchbirdy catchb6;
Catchbirdy catchb7;
Catchbirdy catchb8;
Netz netz3;

PVector p1 = new PVector();
PVector p2 = new PVector();
PVector p3 = new PVector();
boolean drag = false;
PImage bgimage;
String words = "CATCH BIRDIE";
boolean animDir = false;

void setup()
{
  background(140, 172, 234);
  textFont(createFont("Georgia", 105));
  size(800, 800);

  catchb1 = new Catchbirdy(100,100, 0,0);
  catchb2 = new Catchbirdy(130,130, 300,0, true);
  catchb3 = new Catchbirdy(130,130,600,300,true);
  catchb4 = new Catchbirdy(800,80,60,height,false);
  catchb5 = new Catchbirdy(100,100,300,300);
  catchb6 = new Catchbirdy(80,80,0,0);
  catchb7 = new Catchbirdy(100,100, 120,0, true);
  catchb8 = new Catchbirdy(100,100, 30,0, true);
  netz3 = new Netz (80, 150);

  bgimage = loadImage("Threebirdy.png");

  // svg Bild soll zentriert sein
  imageMode(CENTER);
  smooth();
}

void draw()
{
  fill (0);
  text(words, 20, 20, 1000, 1000); // schrift
  image(bgimage,width/2,600);
  fill(189, 209, 250, 60); //weshalb ist der hintergrund mit einem
  rect(0, 0, width, height); //rect gelöst? Wegen dem ghosting?

  if (drag)
  { // zeichne die Flugrichtung
    p1.set(catchb1._pos.x, catchb1._pos.y, 0);
    line(p1.x, p1.y, p2.x, p2.y);
  }

  // zeichne die vögelein
  catchb1.draw();
  catchb2.draw();
  catchb3.draw();
  catchb4.draw();
  catchb5.draw();
```

```

catchb6.draw();
catchb7.draw();
catchb8.draw();
netz3.draw();

if (abs(catchb1._pos.x-netz3._pos.x) < 30 && abs(catchb1._pos.y-netz3._pos.y) < 30) {
  catchb1._dead = true;
}
if (abs(catchb2._pos.x-netz3._pos.x) < 30 && abs(catchb2._pos.y-netz3._pos.y) < 30) {
  catchb2._dead = true;
}
if (abs(catchb3._pos.x-netz3._pos.x) < 30 && abs(catchb3._pos.y-netz3._pos.y) < 30) {
  catchb3._dead = true;
}
if (abs(catchb4._pos.x-netz3._pos.x) < 30 && abs(catchb4._pos.y-netz3._pos.y) < 30) {
  catchb4._dead = true;
}
if (abs(catchb5._pos.x-netz3._pos.x) < 30 && abs(catchb5._pos.y-netz3._pos.y) < 30) {
  catchb5._dead = true;
}
if (abs(catchb6._pos.x-netz3._pos.x) < 30 && abs(catchb6._pos.y-netz3._pos.y) < 30) {
  catchb6._dead = true;
}
if (abs(catchb7._pos.x-netz3._pos.x) < 30 && abs(catchb7._pos.y-netz3._pos.y) < 30) {
  catchb7._dead = true;
}
if (abs(catchb8._pos.x-netz3._pos.x) < 30 && abs(catchb8._pos.y-netz3._pos.y) < 30) {
  catchb8._dead = true;
}
}

void keyPressed ()
{
  if (keyCode == RIGHT)
  {
    netz3.moveright();
  }
  else if (keyCode == LEFT)
  {
    netz3.moveleft();
  }
  if (keyCode == UP)
  {
    netz3.moveup();
  }
  else if (keyCode == DOWN)
  {
    netz3.movedown();
  }
}

```

Da mir Spiele in denen man was „runterschiesst“ um zu gewinnen, nicht so nah sind, war es mir lieber ein Spiel zu programmieren, bei dem Kinder etwas einfangen können. So kam ich auf die Idee, diese Vögelchen rumfliegen zu lassen. Das Netz kann man mit den Pfeiltasten steuern und das Ziel von CATCHBIRDY ist es, innerhalb von möglichst kurzer Zeit alle Vögelchen einzufangen. Gerne würde ich das Spiel verbessern, in dem man eine eingblendete Stoppuhr hat und Punkte bekommt, je schneller das man alle gefangen hat. Gerne würde ich auch verbessern, dass man die Vögelchen nur mit der offenen Seite vom Netzt einfangen könnte, dies ist mir leider noch nicht gelungen. Ebenfalls wäre es bestimmt besser, das Ganze mit einem Array zu machen, deshalb möchte ich noch dazu sagen, dass dies der Stand vom letzten Schultag in der ersten Processing Runde war. Ich befasse mich aktuell mit Arrays und Klassen, da ich wie besprochen noch am weiter üben bin. Es gefällt mir jedoch, dass ich umsetzen konnte was ich mir vorgestellt habe.

# CATCH BIRDIE

