

# Systèmes dynamiques

Guillaume Pelletier-Auger

8 avril 2017

## Résumé

Définition, exploration et catalogage des systèmes dynamiques.

## 1 Définition du prototype *System*

Qu'est-ce qui définit un système dynamique comme je les conçois ? Essayons un premier jet.

```
var System = function(system){
  this.pos = system.pos || {x : 0, y : 0, z : 0};
  this.n = system.n || 0;
  this.animation = system.animation || false;
  this.backgroundColor = system.backgroundColor || 0;
  this.setup = system.setup;
  this.iterativeFunction = system.iterativeFunction;
  this.displayFunction = system.displayFunction;
  this.animation = system.animation || false;
};

System.prototype.runIterativeFunction = function() {
  this.pos = this.iterativeFunction(this.pos.x, this.pos.y, this.pos.z, this.constants);
  this.n++;
};

System.prototype.runDisplayFunction = function() {
  this.displayFunction(this.pos, this.n, this.translate, this.scale);
};
```

## 2 Définition des systèmes individuels

```
var dune001 = new System({
  pos : {x : 0, y : 0, z : 0},
  n : 0,
  constants : {
    a : 5,
    b : 3,
    c : 2,
    d : 1,
    e : 2,
    f : 1
  },
  animation : false,
  iterationsPerFrame : 1500,
  backgroundColor : 0,
  setup : function() {
    frameRate(30);
    background(0);
    fill(255, 5);
    noStroke();
  },
  iterativeFunction : function(x, y, z, c) {
    //c is a set of constants.
    var v = {
      x : sin(c.a * x) + cos(c.b * y) - tan(c.c * z),
      y : cos(c.d * y) + cos(c.e * x) + tan(c.f * z),
      z : z + 0.1
    };
    return v;
  },
  displayFunction : function(v, n) {
    var red = map(abs(sin(n / 10)), 0, 1, 255, 100);
    var green = map(abs(cos(n / 30)), 0, 1, 200, 40);
    var blue = map(abs(sin(n / 10)), 0, 1, 0, 255);
    blendMode(ADD);
    fill(red, green, blue, 15);
    ellipse(v.x * 100, v.y * 100, 0.5);
  }
});
```

Les systèmes devraient-ils également avoir des valeurs *scale* et *translate*? Ça me permettrait de réutiliser les mêmes fonctions d’affichage mais de modifier la zone affichée. D’expérience, c’est quelque chose que je voudrai faire très souvent.

### 3 Définition de fonctions indépendantes

Les fonctions itératives peuvent être définies indépendamment des systèmes dynamiques, et utilisées par ceux-ci. Même chose pour les fonctions d’affichage.

```
var duneFunction = function(x, y, z, c) {  
  var v = {  
    x : sin(c.a * x) + cos(c.b * y) - tan(c.c * z),  
    y : cos(c.d * y) + cos(c.e * x) + tan(c.f * z),  
    z : z + 0.1  
  };  
  return v;  
};  
  
var colorFunction001 = function(v, n, translate, scale) {  
  var red = map(abs(sin(n / 10)), 0, 1, 255, 100);  
  var green = map(abs(cos(n / 30)), 0, 1, 200, 40);  
  var blue = map(abs(sin(n / 10)), 0, 1, 0, 255);  
  blendMode(ADD);  
  fill(red, green, blue, 15);  
  ellipse(translate.x + v.x * scale.x, translate.y + v.y * scale.y, 0.5);  
};
```

## 4 Une possible fonction pour dupliquer un système

Devrais-je me créer une fonction qui ne servirait qu'à dupliquer un système entier, pour ensuite le modifier ? Ça pourrait ressembler à ça :

```
var dune003 = dune002.clone();
dune003.scale = {x : 200, y : 0};
```

Et la fonction copiante ressemblerait à ça :

```
System.prototype.clone = function() {
  return {
    pos : {x : this.pos.x,
          y : this.pos.y,
          z : this.pos.z},
    n : this.n,
    constants : this.constants,
    animation : this.animation,
    iterationsPerFrame : this.iterationsPerFrame,
    backgroundColor : this.backgroundColor,
    setup : this.setup,
    iterativeFunction : this.iterativeFunction,
    displayFunction : this.displayFunction
  };
};
```