

(2013/06/13 - 2013/06/27)

Processing で考えよう (1)

第3回：動きをつくる

川上 博

2013/06/25

今日のテーマ

第3回：7章 動き（パラパラまんが）をつくる


◎ 1回描く (static) vs 繰り返し描く (active)


```
frameRate(30);  
x += speed;
```


◎ 図形を動かす vs 座標系を動かす

```
translate(40, 20);  
rotate(PI/12);  
  
pushMatrix();           radians(90);  
popMatrix();
```

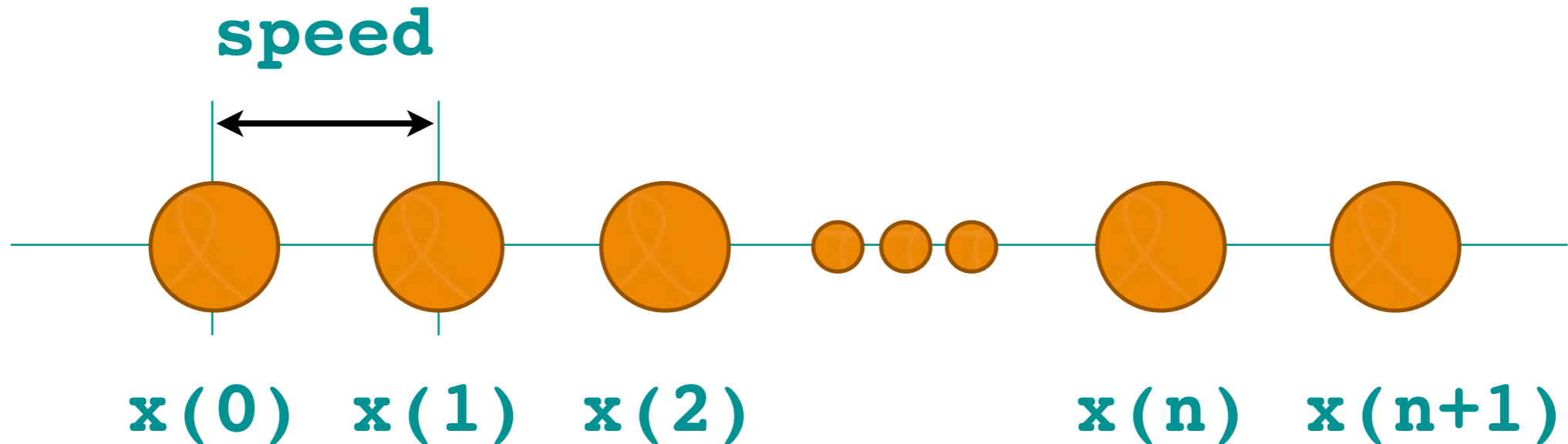
数列(sequence)

① 1, 4, 7, , 10, ...
 $x(n+1) = x(n) + 3$ 等差数列

② 1, 2, 4, , 8, ...
 $x(n+1) = 2 * x(n)$ 等比数列

③ 1, 1, 2, , 3, 5, ...
 $x(n+2) = x(n+1) + x(n)$ Fibonacci number

動かす : $x(n+1) = x(n) + \text{speed}$



初期値

$$x(0)$$

$$x(1) = x(0) + \text{speed}$$

$$x(2) = x(1) + \text{speed} = x(0) + 2 * \text{speed}$$

$$x(3) = x(2) + \text{speed} = x(0) + 3 * \text{speed}$$

$$x(n) = x(n-1) + \text{speed} = x(0) + n * \text{speed}$$

動かす : $x(n+1) = f(x(n))$ のプログラム

```
// file:lec703
// Example 7-3
// Modified by H. Kawakami
// TLT lecture on Processing

int radius;
float x0, x, speed;

void setup(){
  size(500, 500);
  ellipseMode(RADIUS);
  radius=40;
  x0=radius;
  speed=0.5;
  ellipse(x0,250,radius,radius);
}

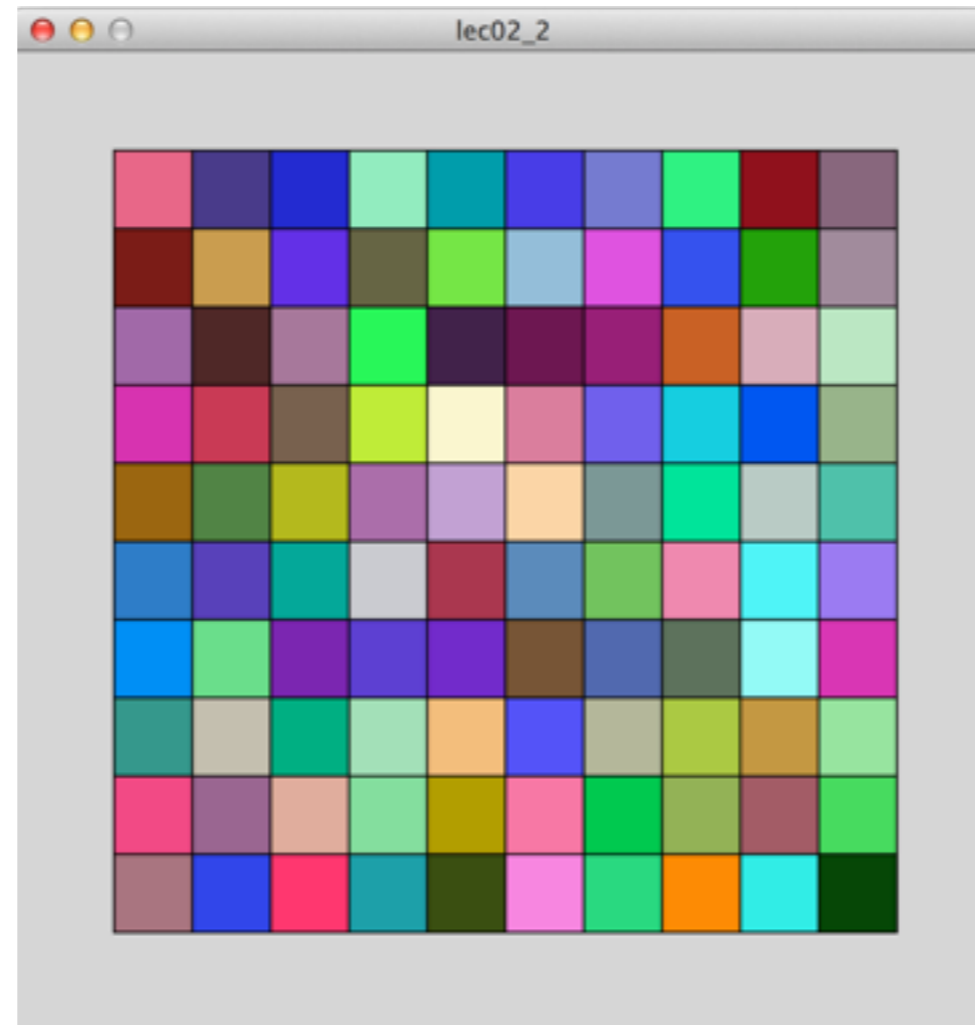
void draw(){
  background(0);
  x = x0 + speed;
  ellipse(x,250,radius,radius);
  x0 = x;
}
```

動かせる公式

```
for(;;){
  x=f(x0);
  xを使った作業

  x0=x;
}
```

網目状に線を引く



lec02-3

漸化式： $x(n+1) = x(n) + \text{speed}$

漸化式の解： $x(n) = x(0) + \text{speed} * n$

$$\begin{aligned} x(n+1) &= x(n) + w & \Rightarrow & x(n) = x(0) + w * n \\ y(n+1) &= y(n) + w & \Rightarrow & y(n) = y(0) + w * n \end{aligned}$$

```
int x=50, y=50;
int w=40;
```

```
size(500,500);
```

```
for(int i=0; i<10; i++){
  for(int j=0; j<10; j++){
    fill(color(random(255), random(255), random(255)));
    rect(x+w*i, y+w*j, w, w);
  }
}
```

$(x+w*i, y+w*j)$

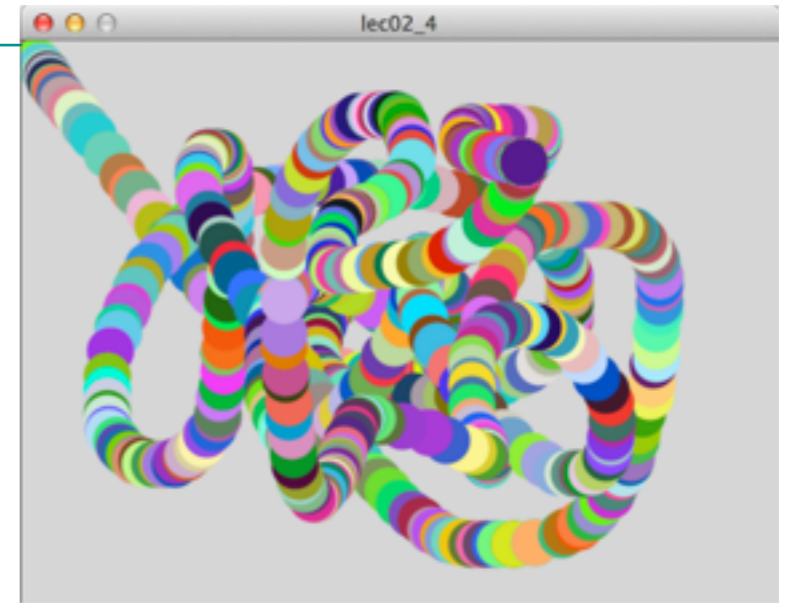


座標変換：移動，回転，拡大，縮小

座標は固定し，中心を指定

```
void setup() {  
  size(480, 360);  
  noStroke();  
}
```

```
void draw() {  
  fill(color(random(255), random(255), random(255)));  
  ellipse(mouseX, mouseY, 30, 30);  
}
```



中心は固定し，座標を移動

```
void draw() {  
  translate(mouseX, mouseY);  
  fill(color(random(255), random(255), random(255)));  
  ellipse(0, 0, 30, 30);  
}
```


pushMatrix() & popMatrix()

```
// file:lec302
// clock: an example of push & pop
```

```
void setup() {
  size(300,300);
  stroke(255);
  frameRate(30);
}
```

```
void draw() {
  background(255);
  float s = second();
  float m = minute();
  float h = hour()%12+1.0;
```

```
  strokeWeight(1);
  stroke(252,225,156);
  fill(252,225,156,50);
  translate(width/2, height/2);
  ellipse(0,0,width-26,height-26);
  noFill();
```

```
  stroke(255,0,0);
  strokeWeight(1);
  pushMatrix();
  rotate(radians(s*(360/60)));
  line(0,0,0,-height/2+16);
  popMatrix();
```

```
  stroke(0,0,255);
  strokeWeight(3);
  pushMatrix();
  rotate(radians(m*(360/60)));
  line(0,0,0,-height/2+36);
  popMatrix();
```

```
  stroke(0,255,0);
  strokeWeight(6);
  pushMatrix();
  rotate(radians(h*(360/12)));
  line(0,0,0,-height/3+20);
  popMatrix();
```

```
}
```

