Randomness

Oregon State University
Mike Bailey
mjb@cs.oregonstate.edu

Start With Something We've Seen Before

```plaintext
void setup() {
  size(800, 800);
  colorMode(RGB);
  background(255, 255, 255);
  stroke(0, 8, 255);
  strokeWeight(4.0);
  fill(255, 58, 58);
  noFill();
}

void draw() {
  beginShape();
    vertex(188, 100);
    vertex(188, 400);
    vertex(288, 400);
    vertex(388, 300);
    vertex(488, 56);
  endShape();
}
```
Pure Randomness is Pretty Jarring

```java
void draw()
{
    background( 255, 255, 255 );
    beginShape();
    for( int x = 0; x < width; x += 5 )
    {
        int y = int( random( 0, height ) );
        vertex( x, y );
    }
    endShape();
}
```

A Better Approach – Add a Random Number to the Current Value

```java
void draw()
{
    background( 255, 255, 255 );
    int y = height / 2;
    beginShape();
    for( int x = 0; x < width; x += 5 )
    {
        int dy = int( random( -height/10, height/10 ) );
        y += dy;
        vertex( x, y );
    }
    endShape();
}
```
Computer Graphics Noise

- The built-in `noise()` function is a smoothly-changing sequence of values.
- It returns values from 0. to 1.
- It is centered around 0.5, i.e., the midline.
- It can be spread out (made smoother) by making the argument smaller.
- It can be compressed (made more jagged) by making the argument larger.
- It is **Coherent** in that the noise value at one point is close to the noise value at the next point.
- Setting `noiseSeed()` makes it **Repeatable** in that the same input always gives the same output.

Noise Octaves Create More Detail

A Noise Octave is another noise wave with lower amplitude (height) and higher frequency (jagginess). We add octaves together to get a combination of smoothness and jagginess.

1 octave 2 octaves

4 octaves 8 octaves
float NoiseFactor = 200.; // larger to make the noise gentler
int NoiseSeed = 22019; // start the random number sequence
int MinOctaves = 1;
int MaxOctaves = 8;

void setup( )
{
  size( 800, 600 );
  colorMode( RGB );
  noFill( );
  noiseSeed( NoiseSeed );
}

void draw( )
{
  background( 200, 200, 255 );
  stroke( 128, 0, 0 );
  strokeWeight( 1. );
  beginShape( );
  vertex( 0, height/2 );
  vertex( width, height/2 );
  endShape( );
  for( int octaves = MinOctaves; octaves <= MaxOctaves; octaves = octaves*2 )
  {
    noiseDetail( octaves );
    int green = int( map( octaves, MinOctaves, MaxOctaves, 0, 255 ) );
    stroke( 255, green, 0 );
    beginShape( );
    for( int x = 0; x < width; x = x + 5 )
    {
      int y = ( height / 2 ) + int( ( height ) * ( noise( x / NoiseFactor ) - 0.5 ) );
      vertex( x, y );
    }
    endShape( );
  }
}
int y = ( height / 2 ) + int( ( height ) * ( noise( x / NoiseFactor ) - 0.5 ) );

- Gives us 0. to 1.
- Gives us -0.5 to +0.5
- Gives us -height/2. to +height/2.
- Gives us 0. to height

Using Noise to Affect Size

float NoiseFactor = 200.;  // larger to make the noise gentler
int NoiseSeed = 22019;  // start the random number sequence

void setup( )
{
  size( 800, 800 );
  colorMode( RGB );
  background( 200, 200, 255 );
  fill( 255, 255, 0 );
  stroke( 0, 0, 0 );
  noiseSeed( NoiseSeed );
  noiseDetail( 4 );
}

if ( mousePressed )
{
  float nx = noise( mouseX/NoiseFactor );
  float ny = noise( mouseY/NoiseFactor );
  ellipse( mouseX, mouseY, 200*nx, 200*ny );
}
Using Noise to Affect Color

float NoiseFactor = 200.; // larger to make the noise gentler
int NoiseSeed = 22019; // start the random number sequence

void setup( )
{
    size( 800, 800 );
    colorMode( RGB );
    background( 200, 200, 255 );
    fill( 255, 255, 0 );
    stroke( 0, 0, 0 );
    noiseSeed( NoiseSeed );
    noiseDetail( 4 );
}

In draw( ):
    if ( mousePressed )
    {
        float nx = noise( mouseX/NoiseFactor );
        float ny = noise( mouseY/NoiseFactor );
        int red  = int( nx*255. );
        int green = int( ny*255. );
        fill( red, green, 0. );
        ellipse( mouseX, mouseY , 100, 100 );
    }

Using 2D Noise to Affect Color

float NoiseFactor = 200.; // larger to make the noise gentler

void setup( )
{
    size( 600, 600 );
    colorMode( RGB );
    background( 200, 200, 255 );
    fill( 255, 255, 0 );
    stroke( 0, 0, 0 );
    noiseDetail( 4 );
}
Using 2D Noise to Affect Color

// takes about 40 seconds to do 600x600 = approx 9,000 points/sec

void draw()
{
  for( int x = 0; x < width; x++ )
  {
    for( int y = 0; y < height; y++ )
    {
      noiseSeed( 0 );
      int red = int( 255.*noise( x/NoiseFactor, y/NoiseFactor ) );
      noiseSeed( 1000 );
      int green = int( 255.*noise( x/NoiseFactor, y/NoiseFactor ) );
      noiseSeed( 2000 );
      int blue = int( 255.*noise( x/NoiseFactor, y/NoiseFactor ) );
      stroke( red, green, blue );
      point( x, y );
    }
  }
  //noLoop( );
  //saveFrame( "ColorClouds.png" );
}

Here are some fun things to try (make the window size smaller first!):
• What happens if you make NoiseFactor larger? Smaller?
• What happens if you only stroke with (red, green, 0.)?
• What if you only use red and blue? Green and blue?