Intro to Computing

Lecture 2
Introduction to Processing
Class Lottery

• Make sure to sign the class roster.

• If you are not “in” the lottery, indicate that. We will contact you by e-mail as soon as we have confirmation from other students.
Recap of Last Time

We are going to learn about computing (and computer science) through creating interactive computer-based art.
Computing as a Medium for Art

Processing 1.5.1, available at processing.org
Processing Canvas

\texttt{size( \textit{width}, \textit{height} );}
Set the size of the canvas.

\texttt{background( [0..255] );}
Set the background grayscale color.
Coordinate System

(0, 0)

+\text{y}

+\text{x}
Pixels
Computing as a Medium for Art

Choosing Stroke Width

```
strokeWeight(1);
strokeWeight(2);
strokeWeight(3);
strokeWeight(4);
strokeWeight(5);
```
How do we choose the shade (or color) of the stroke?
Processing Documentation

Language (API). The Processing Language has been designed to facilitate the creation of sophisticated visual and computational structures.

http://processing.org/reference/
Anatomy of a Function Call

```
line( 10, 10, 50, 80 );
```

- Function name
- Parentheses
- Arguments
- Statement terminator
Drawing a Line

Name

`line()`

Examples

```
line(30, 20, 85, 75);
```

```
line(30, 20, 85, 20);
stroke(126);
line(85, 20, 85, 75);
stroke(255);
line(85, 75, 30, 75);
```
smooth() vs. noSmooth()
Colors

Composed of four elements:

1. Red
2. Green
3. Blue
4. Alpha (Transparency)
Choosing a Color

For instance:
http://www.daviddurman.com/flexi-color-picker/
Why 0 ... 255?
Primitive 2D Shapes

• point
• line
• triangle
• rect (rectangle)
• quad (quadrilateral, four-sided polygon)
• ellipse
• arc (section of an ellipse)
• curve (Catmull-Rom spline)
• bezier (Bezier curve)
Let’s Write a Program to Draw Some Shapes

• ....

• We will also see how to save and open a processing sketch
Shape Formatting

1. Fill color
2. Line thickness
3. Line color

*These are properties of your paintbrush, not of the object you are painting.*
Fill Color

fill(gray);
fill(gray, alpha);
fill(red, green, blue);
fill(red, green, blue, alpha);

noFill();
Stroke (Line) Color

```javascript
stroke(gray);
stroke(gray, alpha);
stroke(red, green, blue);
stroke(red, green, blue, alpha);

noStroke();
```
ellipseMode

```java
ellipseMode(CENTER);
ellipse(35, 35, 50, 50);
ellipseMode(CORNER);
fill(102);
ellipse(35, 35, 50, 50);
```

rectMode

```java
rectMode(CENTER);
rect(35, 35, 50, 50);
rectMode(CORNER);
fill(102);
rect(35, 35, 50, 50);
```

http://processing.org/reference/ellipseMode_.html
http://processing.org/reference/rectMode_.html
Drawing Curves (Arcs)

center: (300, 200)
width: 200
height: 150
angle 1: 0.000 (0.0°)
angle 2: 1.571 (90.0°)
Drawing Curves (Beziers)

start : (158, 169)
ctrl start : (184, 125)
ctrl end : (307, 231)
end : (330, 170)
Curve Generation Tools


- There is also another type of curve called curve() (no tool currently exists to make this easy to use, but you are welcome to use it if you’d like)
Examples
So, What’s the Big Deal?

• We could do any of these things by hand, why do we need a computer?

• Any ideas?
The Setup and Draw Loop

```java
void setup()
{
    // Called once when program starts
}

void draw()
{
    /* Called repeatedly
       while program runs */
}
```
random(\textit{high});
random(\textit{low, high});
  Generate a random number in the range 
\textit{low} (or 0) to \textit{high}

\textit{mouseX}\textit{mouseY}
  Built-in predefined variables that hold the current mouse X and Y locations

\textit{print( something );}
\textit{println( something );}
  Print something to the Processing console.
Anatomy of a Function Call Revisited

Some function calls, like random, return a value that can be fed as input into another function call.
randomEllipse

void setup()
{
    size(300, 300);
    smooth();
}

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

```java
frameRate(fps);
```
Sets number of frames displayed per second.
i.e. the number of times draw() is called per second. Default = 60.

```java
noLoop();
```
Stops continuously calling draw().

```java
loop();
```
Resumes calling draw().
void mousePressed() {
    // Called when the mouse is pressed
}

void mouseReleased() {
    // Called when the mouse is released
}

void mouseClicked() {
    // Called when the mouse is pressed and released
    // at the same mouse position
}

void mouseMoved() {
    // Called while the mouse is being moved
    // with the mouse button released
}

void mouseDragged() {
    // Called while the mouse is being moved
    // with the mouse button pressed
}
Assignment 1

http://cs.brynmawr.edu/Courses/cs110/spring2013/assignments/assignment1.html
Grading Policy and Code Formatting Standards

• Available on website...
• Let’s check them out.
Electronic Submission Instructions

Use the same e-mail you put on the sign-up sheet
Electronic Submission Instructions

• I will invite you to a shared folder
• Please make a subdirectory for each of your assignments
• Inside each subdirectory place your processing sketch file as well as your hardcopy writeup
Example