Module 13
Wrap-up
CS 106 Winter 2018
Module 01

Processing Recap

• Types
• Variables
• Control flow
• Functions
• Classes
Module 02

Arrays and Strings

- Array operations
- Arrays as values
- Functions on arrays
- Special characters
- Concatenating strings
- String equality
- Outputting text
Module 03

Input/Output

• Reading and writing images
• Reading and writing illustrations
• Reading and writing text
• splitTokens()
Module 04

Advanced Shapes

- `beginShape()` / `vertex()` / `endShape()`
- **Using** `PVector` **to represent points**
- Angles
- Polar coordinates
Module 05

User Interfaces

• Model-View-Controller paradigm
• Direct Manipulation
• Hit testing
• UI Toolkits
• ControlP5
Module 06

**Geometric Context**

- `translate()`, `rotate()`, `scale()`
- `pushMatrix()`, `popMatrix()`
- **Order of operations**
- **Building up complex transformations**
- **Hierarchical modelling**
Module 07

Recursion

- Recursion in recursivedrawing.com
- Anatomy of recursion (base case / recursive case / making progress)
- Writing simple recursive functions
Module 08

Randomness

• The random() function
• Generating random integers
• Flipping a coin
• Bias
• Pseudorandomness
• Using randomSeed()
Module 09

Noise

• Using the `noise()` function
• How `noise()` works in 1D, 2D and 3D
• How `noise()` is different from `random()`
Module 10
Data Processing and Text

• Shapes of data: text, sequence, dictionary, table, tree, graph
• Working with text in external files
• Using join(), trim()
• Working with Dictionaries
• Regular expressions—what are they?
Module 11

Tables

- Rows and columns (records and fields)
- CSV files
- Loading tables
- Reading data out of tables
- Sorting by column values
Module 12

Tree-Structured Data

• JSON files
• Types in JSON
• Loading JSON files
• Reading values out of JSON objects
The final exam

• Friday, April 20th
  12:30pm–3:00pm
  PAC 7, 8
• Similar in style to the midterm
• Covers entire course, weighted towards post-midterm material
• Memorization is *not* the key
Review sessions

• TBD — preferences?
• Watch Piazza for announcements about office hours
Study aids

• The midterm
• Assignment and lab questions
• Assignment and lab questions from previous offerings
• Practice programming exercises (updated!)
• Clicker questions
• Final exam review, midterm review
• Reviews from previous offerings
• Your imagination
Practice on paper, not just in Processing
Marking scheme reminder

• Assignment mark based on best 8 of 9 assignments, plus project
• Participation mark based on best 75% of clicker responses
• Must pass exam portion of course
Not appearing

- 3D
- Physics and Animation
- Sound
- Video, live camera input
- Idioms, software engineering
- Testing and debugging
Issues with Processing

• Geared more towards artistic practice than teaching
• Java is becoming a bit problematic

• But still a fun, practical tool, and useful for designers
Where to go from here

- More Processing
- Java
- Python
- Javascript
- C++, Swift
- Max/MSP
- Haxe/OpenFL
void setup()
{
    size( 400, 760 );
    rectMode( CENTER );
    noFill();
    background( 255 );

    for ( int y = 0; y < 24; ++y ) {
        float r_amt = map( y, 0, 23, 0, 1 );
        for ( int x = 0; x < 12; ++x ) {
            pushMatrix();
            translate( r_amt * random( -5, 5 ), r_amt * random( -5, 5 ) );
            translate( 20 + (x+0.5)*30, 20 + (y+0.5)*30 );
            rotate( radians( random( -45, 45 ) ) * r_amt );
            rect( 0, 0, 30, 30 );
            popMatrix();
        }
    }
}
P5js mode in Processing

function setup()
{
    createCanvas( 400, 760 );
    rectMode( CENTER );
    noFill();
    background( 255 );

    for ( var y = 0; y < 24; ++y ) {
        var r_amt = map( y, 0, 23, 0, 1 );
        for ( var x = 0; x < 12; ++x ) {
            push();
            translate( r_amt * random( -5, 5 ), r_amt * random( -5, 5 ) );
            translate( 20 + (x+0.5)*30, 20 + (y+0.5)*30 );
            rotate( radians( random( -45, 45 ) ) * r_amt );
            rect( 0, 0, 30, 30 );
            pop();
        }
    }
}
def setup():
    size( 400, 760 )
    rectMode( CENTER )
    noFill()
    background( 255 )

for y in range( 24 ):
    r_amt = map( y, 0, 23, 0, 1 )
    for x in range( 12 ):
        pushMatrix()
        translate( r_amt * random( -5, 5 ), r_amt * random( -5, 5 ) )
        translate( 20 + (x+0.5)*30, 20 + (y+0.5)*30 )
        rotate( radians( random( -45, 45 ) ) * r_amt )
        rect( 0, 0, 30, 30 )
        popMatrix()
Redmonk rankings for Q1 2018

<table>
<thead>
<tr>
<th></th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JavaScript</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
</tr>
<tr>
<td>3</td>
<td>Python</td>
</tr>
<tr>
<td>4</td>
<td>PHP</td>
</tr>
<tr>
<td>5</td>
<td>C#</td>
</tr>
<tr>
<td>6</td>
<td>C++</td>
</tr>
<tr>
<td>7</td>
<td>CSS</td>
</tr>
<tr>
<td>8</td>
<td>Ruby</td>
</tr>
<tr>
<td>9</td>
<td>C</td>
</tr>
<tr>
<td>10</td>
<td>Swift</td>
</tr>
<tr>
<td>10</td>
<td>Objective-C</td>
</tr>
</tbody>
</table>
Here are the best programming languages to learn in 2018

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>JavaScript</td>
</tr>
<tr>
<td>2</td>
<td>Swift</td>
</tr>
<tr>
<td>3</td>
<td>Python</td>
</tr>
<tr>
<td>4</td>
<td>Java</td>
</tr>
<tr>
<td>5</td>
<td>C++</td>
</tr>
</tbody>
</table>
Java
• Good all-purpose programming language
• Natural successor to Processing

Javascript
• Best choice for client-side web apps
• Probably easiest to deploy

Python
• Very practical language
• Useful for data processing, science
• Not as much support for graphics or application development
D3 (Javascript)

d3js.org
# Cinder (C++)

## Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>GBUFFER_TEMP</td>
</tr>
<tr>
<td>Debug</td>
<td>OFF</td>
</tr>
</tbody>
</table>

### SSAO
- **Iterations**: 4
- **SampleRadius**: 1.1
- **Intensity**: 1.72
- **Scale**: 0.08
- **Bias**: 0.05
- **Jitter**: 0.12
- **SelfOcclusion**: 0.06
- **Blur**: 1

### Anti-aliasing
- **Weight**: 1.00
- **LScale**: 1.00

### Physics
- **Simulate**: OFF
- **Damping**: 0.40
- **Friction**: 0.10
- **Gravity**: -0.07
- **CubeScale**: 2.00

[libcinder.org](http://libcinder.org)
openFrameworks (C++)
Books

Casey Reas and Ben Fry

Processing for Visual Artists: How to Create Expressive Images and Interactive Art
Andrew Glassner
Books

Introduction to Computing and Programming with Java: a Multimedia Approach
Mark J. Guzdial and Barbara Ericson
Online resources

- Daniel Shiffman’s web page (shiffman.net): courses, videos, lots of inspiration.
- codeacademy.com, khanacademy.org, code.org
- courseware.cemc.uwaterloo.ca
- Online courses at lynda.com
Code is a creative medium.