Review

- Random numbers
- mouseX, mouseY
- setup() & draw()
- frameRate(), loop(), noLoop()
- Mouse and Keyboard interaction
- Arcs, curves, bézier curves, custom shapes
- Hue-Saturation-Brightness vs. Red-Green-Blue color
- Example Sketches
- OpenProcessing website

Odds and Ends

- Dropbox installation is a two-step process
  - Sign up for an account with dropbox
  - Install the dropbox application on your computer
- After you have installed dropbox
  - Invitation to join a shared folder named with your email user name
  - This is where all the submissions go!
- Processing programs carry the extension .pde
- Processing programs must be in a folder with the same name
  - myProgram.pde must be inside a folder called myProgram

Syntax

- Function call
  - line( 10, 10, 50, 80 );
  - Name
  - The commas
  - The parens ()
  - The semicolon
- Code block
  - The curly braces {}
- Comments
  - //
  - /* and */

Images

- loadImage(filename);
  - Loads an image from a file in the data folder in sketch folder.
  - Must be assigned to a variable of type PImage.
- image(img, X, Y, [X2, Y2]);
  - Draws the image img on the canvas at X,Y
  - Optionally fits image into box X,Y and X2,Y2
- imageMode(CORNER);
  - X2 and Y2 define width and height.
- imageMode(CORNERS);
  - X2 and Y2 define opposite corner.

Image Example

```pde
imageExample.pde

PImage img;
void setup()
{
  size(500, 400);
  img = loadImage("natura-morta.jpg");
  image(img, 50, 40);
}
```

Variables

- A name to which data can be assigned
- A variable name is declared as a specific data type
- Names must begin with a letter, “_” or “$” and can contain letters, digits, “_” and “$”

```pde
boolean bReady = true;
int i;
int j = 12;
float fSize = 10.0;
color _red = color(255,0,0);
String name123 = "Fred";
PImage img;
```
Variable Uses

• Use a value throughout your program,
  – but allow it to be changed
• As temporary storage for an intermediate computed result
• To parameterize – instead of hardcoding coordinates
• Special variables (preset variables)
  – width, height
  – screen.width, screen.height
  – mouseX, mouseY
  – pmouseX, pmouseY

Other "things" ...

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Default</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>a series of chars in quotes &quot;abc&quot;</td>
<td>null</td>
<td>?</td>
</tr>
<tr>
<td>PImage</td>
<td>an image</td>
<td>null</td>
<td>?</td>
</tr>
<tr>
<td>PFont</td>
<td>a font for rendering text</td>
<td>null</td>
<td>?</td>
</tr>
</tbody>
</table>

String message = "Hello World!";

Mixing types and Integer Division

• 3*1.5
  – value?
  – type?

• 3/2

• 2/3

• x/y

Primitive Data Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Default</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>{ true, false }</td>
<td>false</td>
<td>?</td>
</tr>
<tr>
<td>byte</td>
<td>(0..255)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>int</td>
<td>(-2,147,483,648 .. 2,147,483,647)</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>long</td>
<td>(-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807)</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>float</td>
<td>(-3.40282347E+38 .. 3.40282347E+38)</td>
<td>0.0</td>
<td>4</td>
</tr>
<tr>
<td>double</td>
<td>much larger/smaller</td>
<td>0.0</td>
<td>8</td>
</tr>
<tr>
<td>color</td>
<td>(#00000000 .. #FFFFFFFF)</td>
<td>black</td>
<td>4</td>
</tr>
<tr>
<td>char</td>
<td>a single character 'a', 'b', ...</td>
<td>'\u0000'</td>
<td>2</td>
</tr>
</tbody>
</table>

Data Type Conversion

• Variables of some types can be converted to other types.
• Type conversion function names are the types to which data will be converted

// binary(), boolean(), byte(),
// char(), float(), str()

float f = 10.0;
int i;

/i = f;     // Throws a runtime error
i = int(f);
println( char(65) ); // Prints the character 'A'

Conditionals: if-statement

**Programmatic branching** ...

if ( boolean_expression ) {
  statements;
}

// What does this do?
void draw() {
  if ( mouseX > 50 && mouseY > 50 ) {
    ellipse( mouseX, mouseY, 10, 10 );
  }
}
Logical Expressions

&& logical conjunction (and)
  • both expressions must be true for conjunction to be true

|| logical disjunction (or)
  • either expression must be true for disjunction to be true

! logical negation (not)
  • true → false, false → true

Relational Expressions

< less than
> is greater than
<= is less than or equal to
>= is greater than or equal to
== is equivalent
!= is not equivalent

Relational Expressions: Examples

1. if (true) { .. }
2. if (10 > 10) { .. }
3. if (10 >= 10) { .. }
4. if ('a' == 'a') { .. }
5. if ('Bryn Mawr' != 'bryn mawr') { .. }

Logical Expression Examples

1. if ((2 > 1) && (3 > 4)) { .. }
2. if ("blah" == "blah") && (1 + 2 == 3) { .. }
3. if (!false) { .. }
4. if (!true) { .. }
5. if (true || false) && true { .. }
6. if (true && false) || true) { .. }
7. if (true || false) && true | | true) { .. }
8. if (true && false) || true) { .. }
9. ..

Conditionals: if-else-statement

if (boolean_expression) {
  statements executed when boolean_expression is true;
} else {
  statements executed when boolean_expression is false;
}

// What does this do?
void draw() {
  if (mouseY < 50) {
    println("the sky");
  } else {
    println("the ground");
  }
}

Conditionals: if-else-if-statement

if (boolean_expression_1) {
  statements;
} else if (boolean_expression_2) {
  statements;
} else if (boolean_expression_3) {
  statements;
} else {
  statements;
}
void setup() {
  size(500, 500);
  ellipseMode(CENTER);
}

void draw() {
  if (mouseX < width/2) {
    stroke(255, 0, 0);
    if (mouseY < height/2) {
      fill(0, 255, 0);
    } else {
      fill(0, 0, 255);
    }
  } else {
    fill(0, 0, 255);
  }

  if (mouseY < height/2) {
    fill(255, 0, 0);
  } else {
    fill(255);
  }

  ellipse(mouseX, mouseY, 50, 30);
}

What will this do?

void setup() {
  size(500, 500);
}

void draw() {
  if (mouseX > 100) {
    background(255, 0, 0);
  } else if (mouseX > 200) {
    background(0, 0, 255);
  }
}

What does this do?

void setup() {
  size(500, 500);
}

void draw() {
  if (mouseX > 200) {
    background(0, 0, 255);
  } else if (mouseX > 100) {
    background(255, 0, 0);
  }
}

Does this work better?

Equations of Motion (Simplified)

r = displacement (position)
t = time
v = velocity
a = acceleration

- Constant acceleration (a)
  \[ r_{t+1} = r_t + v_t \Delta t \]
  \[ v_{t+1} = v_t + a \Delta t \]

- Assume small time intervals – i.e. \( \Delta t = 1 \)