2D Shapes (using variables)

Creative Coding & Generative Art in Processing 2
Ira Greenberg, Dianna Xu, Deepak Kumar

Did you do this?

• Go the CS Computer Lab (Room 231 PSB)
• Log in
• Start the Processing application
  (Make sure it is Version 2.x)
• In a web browser, go to the Tutorials section of processing.org
  http://www.processing.org/tutorials/gettingstarted/
• Read the Getting Started tutorial (by Casey Reas & Ben Fry) and try out the
  two examples of simple Processing programs presented there
• If you’d like, install Processing 2.x on your own computer
• Do the Coordinate System and Shapes tutorial
• Read Ch. 2, pgs. 33-48 (for last class)
• Read Ch. 2, pgs. 48-63 (for this class)
• Start Assignment 1

Questions

• Assignment 1
• Reading for today
• Entry Survey (Please complete by next Tuesday)

Drawing Basics

• Canvas – computer screen
  size(width, height);
• Drawing Tools – shape commands
• Colors – grayscale, RGB, or RGBA
  background(125);
**Drawing Tools - Basic Shapes**

- Point
- Line
- Triangle
- Rectangle
- Ellipse
- Arc
- Quad
- Polygon
- Curve

**Drawing & Shape Attributes**

- **Anti-aliasing**
  - smooth();
  - noSmooth();

- **Stroke**
  - noStroke();
  - strokeWeight(<pixel width>);
  - stroke(<stroke color>);

- **Fill**
  - noFill();
  - fill(<fill color>);

**Drawing Tools - Basic Shapes**

- Point
- Line
- Triangle
- Rectangle
- Ellipse
- Arc
- Quad
- Polygon
- Curve
Basic Shapes: Arcs

• What is an arc?

arc(x, y, width, height, startAngle, endAngle);

- degrees vs radians

noFill();
stroke(255, 0, 0);
arc(200, 200, 150, 150, 0, PI);

fill(255, 255, 0);
stroke(255, 0, 0);
arc(200, 200, 150, 150, 0, PI);

start = 30 degs
end = 102 degs

start = 59 degs
end = 210 degs

start = 160 degs
end = 316 degs

start = 96 degs
end = 265 degs

start = 2 degs
end = 119 degs

start = 116 degs
end = 281 degs

start = 1 degs
end = 326 degs

start = 14 degs
end = 213 degs

W degrees vs radians
Basic Shapes: Quadrilaterals

```javascript
quad(x1, y1, x2, y2, x3, y3, x4, y4);
```

Basic Shapes: Polygons

```javascript
beginShape();
  vertex(x1, y1);
  quad(xN, yN);
endShape(CLOSE);
```

Basic Shapes: Curves

```javascript
curve(cpx1, cpy1, x1, y1, x2, y2, cpx2, cpy2);
```

More Complex Curves

```javascript
beginShape();
  curveVertex(x1, y1);
  curveVertex(xN, yN);
endShape(CLOSE);
```
Example: A Penguin

```
// penguin
size(400, 500);
smooth();
background(0);
stroke(245, 63, 55);
strokeWeight(3);
fill(0);

beginShape();
  curveVertex(105, 400);
  curveVertex(101, 392);
  curveVertex(117, 388);
  curveVertex(119, 342);
  curveVertex(106, 210);
  curveVertex(120, 160);
  curveVertex(125, 120);
  curveVertex(122, 99);
  curveVertex(116, 90);
  curveVertex(85, 72);
  curveVertex(112, 80);
  curveVertex(120, 81);
  curveVertex(129, 80);
  curveVertex(120, 77);
  curveVertex(112, 80);
  curveVertex(105, 400);
endShape();
```

Review: Drawing Basics

- **Canvas**
  ```
  size(width, height)
  ```

- **Drawing Tools**
  ```
  point(x, y)
  line(x1, y1, x2, y2)
  triangle(x1, y1, x2, y2, x3, y3)
  quad(x1, y1, x2, y2, x3, y3, x4, y4)
  rect(x, y, width, height)
  ellipse(x, y, width, height)
  arc(x, y, width, height, startAngle, endAngle)
  curve(cpx1, cpy1, x1, y1, x2, y2, cpx2, cpy2)
  ```

- **Colors**
  ```
  grayscale[0..255],
  RGB [0..255], [0..255], [0..255],
  alpha[0..255]
  ```

- **Drawing & Shape Attributes**
  ```
  smooth(), noSmooth()
  stroke(color), noStroke(), strokeWeight(pixels/Width)
  fill(color), noFill()
  ```

Simple Program Structure

```
// Create and set canvas
size(width, height);
smooth();
background(color);

// Draw something
...

// Draw something else
...

// etc.
```

Simple Program Structure

```
// Draw a barn
size(300, 300);
smooth();
background(187, 193, 127);

// wall
fill(206, 224, 14);
rect(50, 150, 200, 100);

// Draw Door
fill(72, 26, 2);
rect(115, 160, 70, 90);

// Draw roof
fill(224, 14, 14);
quad(50, 150, 160, 75, 200, 75, 250, 150);
```
Variables: Naming Values

• Values
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

Variables: Naming Values

• Values
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

Variables: Naming Values

• Values
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

Variables: Naming Rules & Conventions

• Variables have a Name
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

• Variables have a Type
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

• Variables have a Name
  42, 3.14159, 2013, “Hi, my name is Joe!”; true, false, etc.
  – Numbers
    • Integers
      int meaningOfLife = 42;
      int year = 2013;
    • Floating point numbers
      float pi = 3.14159;
  – Strings
    String greeting = “Hi, my name is Joe!”;
  – Boolean
    boolean keyPressed = true;

• Names begin with a letter, an underscore (_), or a dollar sign ($)
  Examples: weight, meaningOfLife, $value

• Names may include numbers, but only after the initial character
  Examples: value1, score5, $bestFriends

• No spaces are permitted in names
  Examples: value1, dollar $sign

• Processing Conventions
  – Names begin with a lowercase letter
    Example: meaningOfLife, highestScore
  – Constants are written in all caps
    Example: DAYS_IN_WEEK, PI
### Variables: Declarations & Initialization

- Declaring variables
  ```
  int meaningOfLife;
  int year;
  float pi;
  String greeting;
  boolean keyPressed;
  ```

- Initializing values in declarations
  ```
  int meaningOfLife = 42;
  int year = 2013;
  float pi = 3.14159;
  String greeting = "Hi, my name is Joe!"; 
  boolean keyPressed = true;
  ```

### The color type

- Processing has a type called `color`
  ```
  color firebrick = color(178, 34, 34);
  color chartreuse = color(127, 255, 0);
  color fuchsia = color(255, 0, 255);
  ```

- Using Variables
  ```
  // Draw a barn
  // Create and set canvas
  size(300, 300);
  smooth();
  background(187, 193, 127);
  // wall
  fill(206, 224, 14);
  rect(50, 150, 200, 100);
  // Draw Door
  fill(72, 26, 2);
  rect(125, 200, 50, 50);
  // Draw roof
  fill(224, 14, 14);
  quad(50, 150, 100, 75, 200, 75, 250, 150);
  ```

### Expressions: Doing Arithmetic

- Assignment statement
  ```
  <variable> = <expression>;
  ```

- Examples:
  ```
  meaningOfLife = 42;
  area = length * height;
  perc = statePop/totalPop*100.0;
  ```

- Operators

  - `+` (addition)
  - `-` (subtraction)
  - `*` (multiplication)
  - `/` (division)
  - `%` (modulus)

- Example:
  ```
  mouth_x = ((leftIris_x + irisDiam) / 2 + eyeWidth) / 4;
  ```
// Draw a simple house
// Create and set canvas
size(300, 300);
smooth();
background(187, 193, 127);

// wall
fill(206, 224, 14);
rect(50, 150, 200, 100);

// Draw Door
fill(72, 26, 2);
rect(125, 200, 50, 50);

// Draw roof
fill(224, 14, 14);
triangle(50, 150, 150, 50, 250, 150);

What variables will make this scalable?

What variables will make this movable?

What variables will make this scalable and movable?

Homework

• Read Ch. 3 (pgs. 65-72)
• Read and do the Color tutorials on processing.org
• Review Processing commands:
  size(), background(), 2D shapes: point(), line(), triangle(), rectangle(), quad(), ellipse(). Attributes and modes: stroke(), noStroke(), strokeWeight(), fill(), noFill(), rectMode(), ellipseMode().
  Color values (grayscale and RGB) and transparency.
• Review the concepts of an algorithm, pseudocode, syntax, and sequencing
• Complete Assignment 1.