

Lesson Plan for Activity: Pixel Art

Subject: Creating

Length of Lesson: 90 minutes

Concept or Skill Focus: Students can create artwork through method calls

Goal:

Objectives/Outcomes:

- Students can learn about method calls
- Students can learn how to pass parameters to methods
- Students can use RGB values to color their artwork
- Students can practice using the computer's coordinate plane

Activities and Time Line (Introduction, Middle, Conclusion)

- Students can start by exploring various pixel art
- On the Graphics Sheet, students can plan their pixel art
- On a projector, show students the coordinate plane (Graphics Sheet), top left-corner is (0, 0)
- On a projector, show students the various methods on Creating Visual Designs:
 - `line(x1,y1,x2,y2);`
 - `rect(x,y,w,h);`
 - `ellipse(x,y,w,h);`
 - `triangle(x1,y1,x2,y2,x3,y3);`
- Using the Graphics Sheet, start drawing a simple pixel art example
- Open the Processing IDE and code a few statements to draw the sample pixel art example
- On a projector, show students the various methods on Creating Visual Designs – Adding Color
 - `background(r,g,b);`
 - `stroke(r,g,b);`
 - `fill(r,g,b);`
- Make a connection between the RGB frosting activity – students should be able to choose colors based on RGB, or from the Color Selector under the Tools menu option in the Processing IDE
- Code a few more statements in the Processing IDE for adding color

Materials

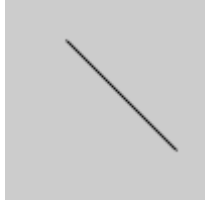
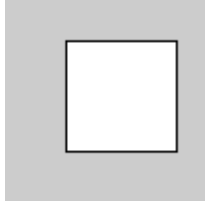
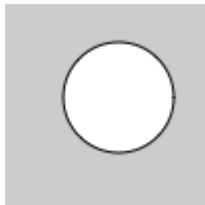
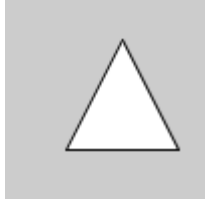
- Computers with Processing 3 installed downloadable here: <https://processing.org/download/>
- Graphics Sheet (below)
- Creating Visual Designs Activity (below)
- Pencils

Creating Visual Designs

Using Processing found at processing.org

Draw a picture made of 2D primitives!

Use the following methods to make a self-portrait, an alien creature, or something more abstract.

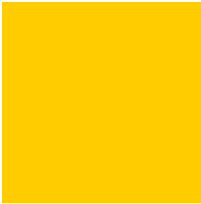
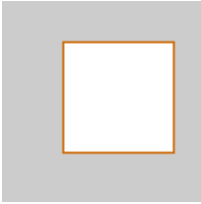
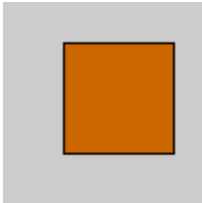
Methods	Examples
<p>To draw a line: <code>line(x1,y1,x2,y2);</code></p> <p><code>x1</code> – x-coordinate of the first endpoint <code>y1</code> – y-coordinate of the first endpoint <code>x2</code> – x-coordinate of the second endpoint <code>y2</code> – y-coordinate of the second endpoint</p>	<pre>line(30, 20, 85, 75);</pre> 
<p>To draw a rectangle: <code>rect(x,y,w,h);</code></p> <p><code>x</code> – x-coordinate of the shape's upper-left corner <code>y</code> – y-coordinate of the shape's upper-left corner <code>w</code> – width of the shape (in pixels) <code>h</code> – height of the shape (in pixels)</p>	<pre>rect(30, 20, 55, 55);</pre> 
<p>To draw an ellipse: <code>ellipse(x,y,w,h);</code></p> <p><code>x</code> – x-coordinate of the shape's origin (center) <code>y</code> – y-coordinate of the shape's origin (center) <code>w</code> – width of the shape <code>h</code> – height of the shape</p>	<pre>ellipse(56, 46, 55, 55);</pre> 
<p>To draw a triangle: <code>triangle(x1,y1,x2,y2,x3,y3);</code></p> <p><code>x1</code> – x-coordinate of the first point <code>y1</code> – y-coordinate of the first point <code>x2</code> – x-coordinate of the second point <code>y2</code> – y-coordinate of the second point <code>x3</code> – x-coordinate of the third point <code>y3</code> – y-coordinate of the third point</p>	<pre>triangle(30, 75, 58, 20, 86, 75);</pre> 

For more 2D primitives, check out the [arc](#), [quad](#), and [point](#) pages on processing.org.

For 3D primitives, check out the [box](#), and [sphere](#) pages on processing.org.

Add color to your design!

Don't forget, you need to call `stroke()` and `fill()` **before** you draw the shape.

Methods	Examples
<p>Set the background color: <code>background(r,g,b);</code></p> <p>r – amount of red, from 0 (none) to 255 (max) g – amount of green, from 0 (none) to 255 (max) b – amount of blue, from 0 (none) to 255 (max)</p>	<pre>background(255, 204, 0);</pre> 
<p>Set the outline color: <code>stroke(r,g,b);</code></p> <p>r – amount of red, from 0 (none) to 255 (max) g – amount of green, from 0 (none) to 255 (max) b – amount of blue, from 0 (none) to 255 (max)</p>	<pre>stroke(204, 102, 0); rect(30, 20, 55, 55);</pre> 
<p>Set the interior color: <code>fill(r,g,b);</code></p> <p>r – amount of red, from 0 (none) to 255 (max) g – amount of green, from 0 (none) to 255 (max) b – amount of blue, from 0 (none) to 255 (max)</p>	<pre>fill(204, 102, 0); rect(30, 20, 55, 55);</pre> 

For more on color, check out the [colorMode](#), [saturation](#), and [hue](#) pages on [processing.org](#).

Here are some examples:

