Review

• What is Computing?
• Occupations in CS
• What can be Programmed?
• Creative Computing
• Processing
• Downloading Processing
• Sketchpad

• Primitive Shapes
  – point
  – line
  – triangle
  – quad
  – rect
  – ellipse
• Processing Canvas
• Coordinate System
• Shape Formatting
  – Colors
  – Stroke
  – Fill
Comments

• Used to explain your source code
• Ignored by Processing

/* This is a comment
   that spans multiple lines */

// This is a comment that is restricted to a single line

line(0, 0, 10, 10);  // Can start anywhere, continue to line end

Note the color of the various items in the processing editor.
void setup()
{
   // Called once when program starts
}

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

print( something );
println( something );
    Print something to the Processing console.

mouseX
mouseY
    Built-in predefined variables that hold the current mouse X and Y locations.
randomEllipse

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

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

`frameRate(fps);`

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

`noLoop();`

Stops continuously calling `draw()`.

`loop();`

Resumes calling `draw()`.
More Graphics

arc(…)
curve (…)
bézier(…)
shape(…)

Arcs

\[ \text{arc}(x, y, \text{width}, \text{height}, \text{start}, \text{stop}); \]

An arc is a section of an ellipse

\[ x, y, \text{width}, \text{height} \]
location and size of the ellipse
\[ \text{start}, \text{stop} \]
arc bounding angles (in radians)
Arcs

\[ \text{arc}(x, y, \text{width}, \text{height}, \text{start}, \text{stop}); \]
Spline Curves

\texttt{curve( x1, y1, x2, y2, x3, y3, x4, y4 );}

\emph{Spline}: A smooth line drawn through a series of points

A \texttt{curve} is a Catmull-Rom (cubic Hermite) spline defined by four points

\begin{itemize}
  \item \(x_2, y_2\) \texttt{and} \(x_3, y_3\) \texttt{beginning/end points of visual part of curve}
  \item \(x_1, y_1\) \texttt{and} \(x_4, y_4\) \texttt{control points that define curve curvature}
\end{itemize}
Spline Curves

curve( x1, y1, x2, y2, x3, y3, x4, y4 );
Bézier Curves

bezier( x1, y1, cx1, cy1, cx2, cy2, x2, y2 );

A smooth curve defined by two anchor points and two control points

x2, y2 and x2, y2
anchor points of bézier curve
cx1, cy1 and cx2, cy2
control points that define curvature
Bézier Curves

`bezier( x1, y1, cx1, cy1, cx2, cy2, x2, y2 );`

![Bézier Curve Diagram](image)
Custom Shapes

- Composed of a series of vertexes (points)
- Vertexes may or may not be connected with lines
- Lines may join at vertexes in a variety of manners
- Lines may be straight, curved, or bézier splines
- Shapes may be closed or open
Custom Shapes

beginShape( [option] );

vertex( x, y );

curveVertex( x, y );

bezierVertex( cx1, cy1, cx2, cy2, x, y );

endShape( [CLOSE] );
strokeJoin()
Example Sketches...

- LadyBug1
- Monster1
- Ndebele
- Penguin1
- SouthParkCharacter1
- Sushi
- GiorgioMorandi
Dropbox

• https://www.dropbox.com/