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

```
arc( x, y, width, height, start, stop );
```

An arc is a section of an ellipse

`x, y, width, height`

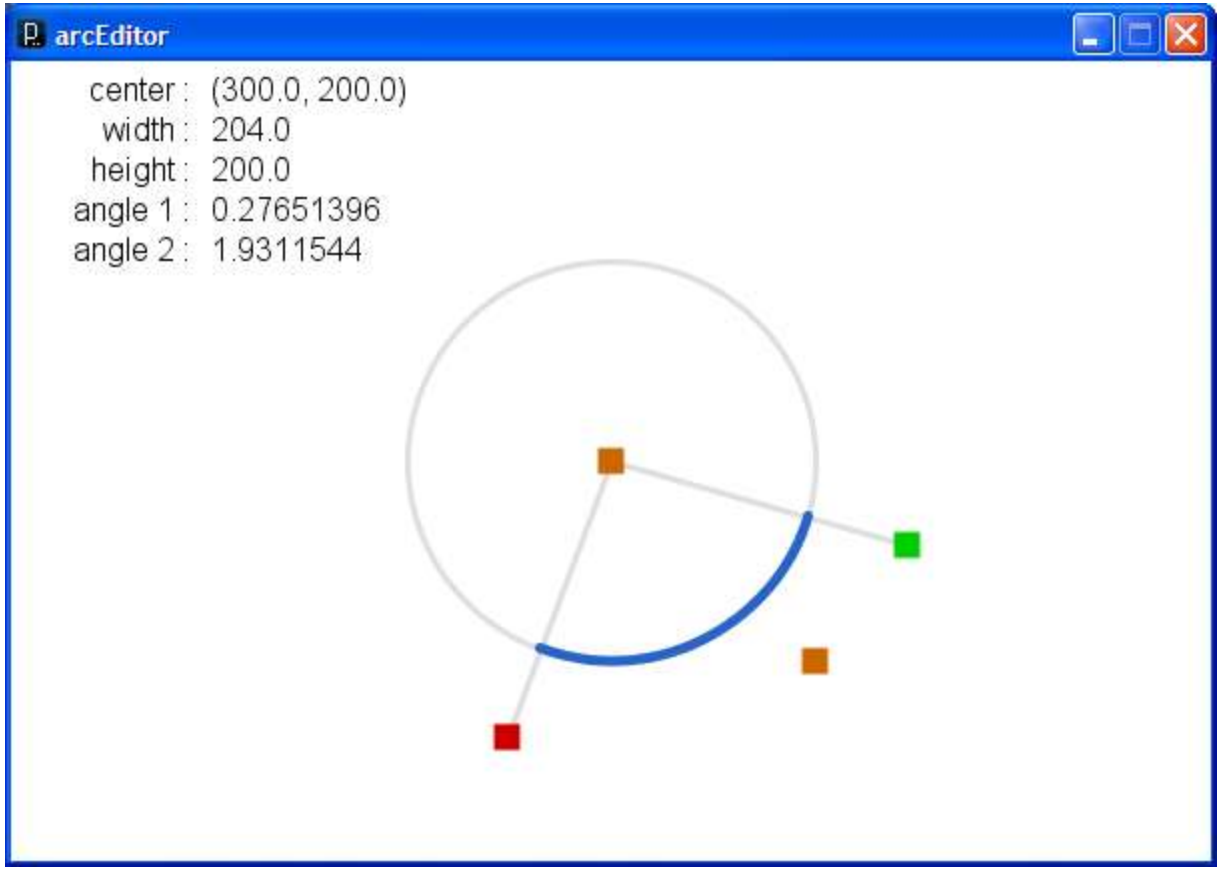
location and size of the ellipse

`start, stop`

arc bounding angles (in radians)

Arcs

```
arc( x, y, width, height, start, stop );
```



arcEditor.pde

Spline Curves

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

Spline: A smooth line drawn through a series of points

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

`x2, y2` *and* `x3, y3`

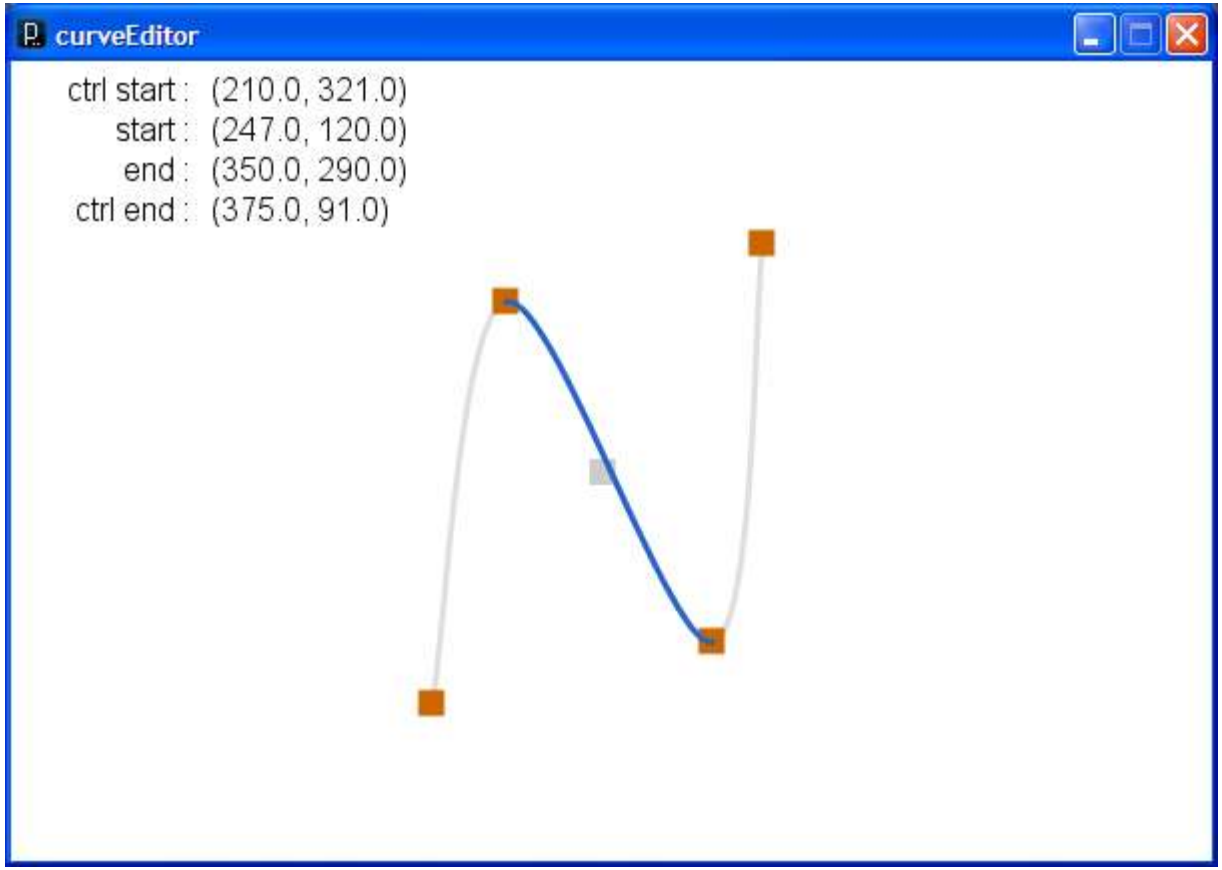
beginning/end points of visual part of curve

`x1, y1` *and* `x4, y4`

control points that define curve curvature

Spline Curves

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



curveEditor.pde

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`

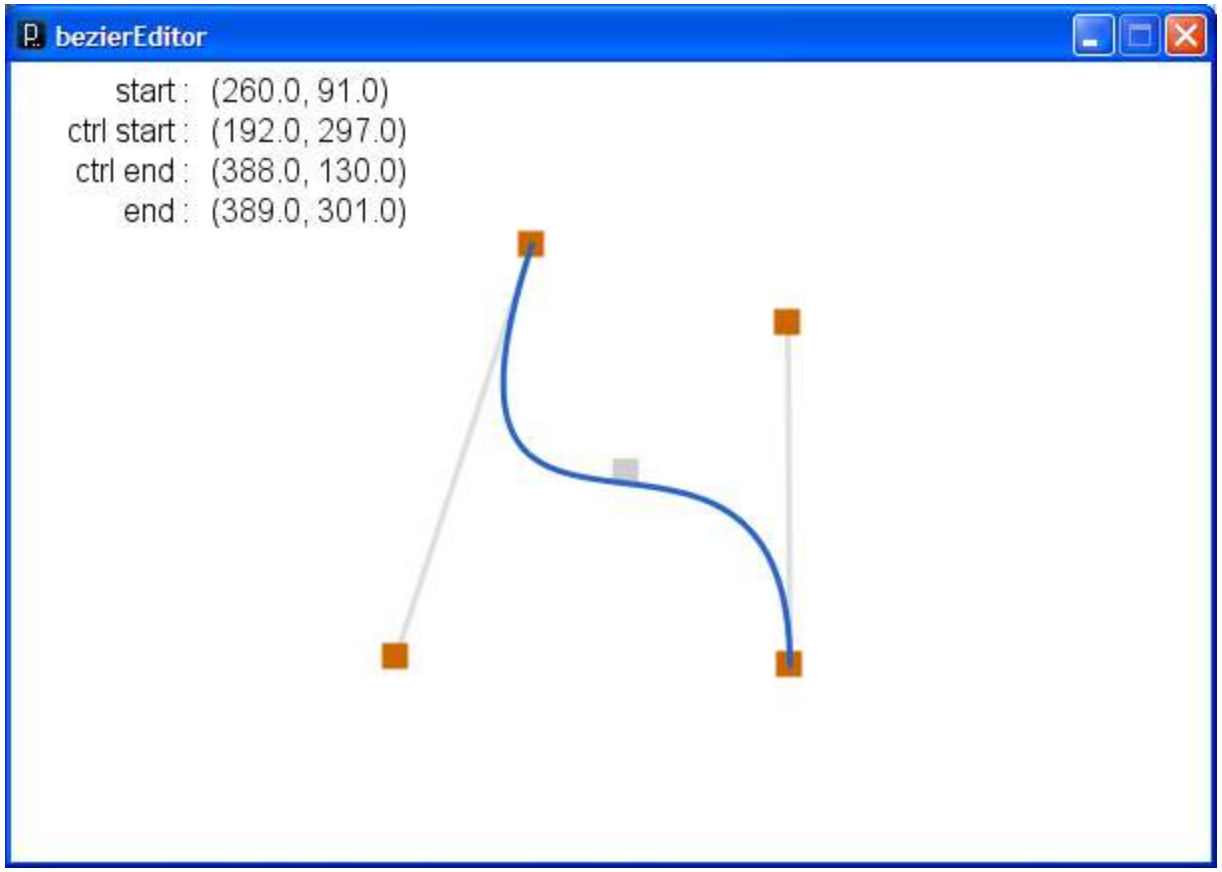
anchor points of bézier curve

`cx1, cy1` *and* `cx2, cy2`

control points that define curvature

Bézier Curves

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



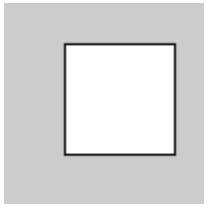
bezierEditor.pde
Inkscape

Custom Shapes

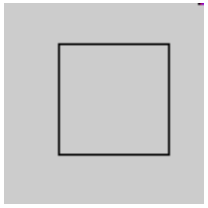
- Composed of a series of vertexes (points)
- Vertexes may or may not be connected with lines
- Lines may join at vertexes in a variety of manners
- Lines may be straight, curved, or bézier splines
- Shapes may be closed or open

Custom Shapes

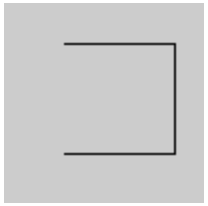
```
beginShape( [option] );

vertex( x, y );

curveVertex( x, y );

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

endShape( [CLOSE] );
```



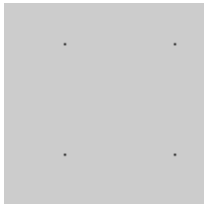
```
beginShape();
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape(CLOSE);
```



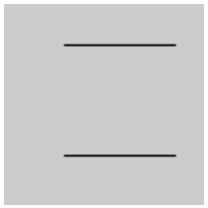
```
noFill();
beginShape();
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape(CLOSE);
```



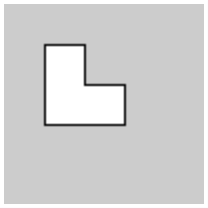
```
noFill();
beginShape();
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape();
```



```
beginShape(POINTS);
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape();
```

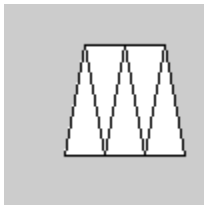


```
beginShape(LINES);
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape();
```

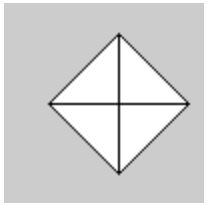


```
beginShape();
vertex(20, 20);
vertex(40, 20);
vertex(40, 40);
vertex(60, 40);
vertex(60, 60);
vertex(20, 60);
endShape(CLOSE);
```

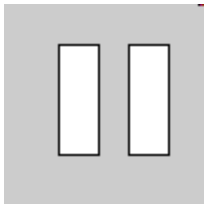
```
beginShape(TRIANGLES);
vertex(30, 75);
vertex(40, 20);
vertex(50, 75);
vertex(60, 20);
vertex(70, 75);
vertex(80, 20);
endShape();
```



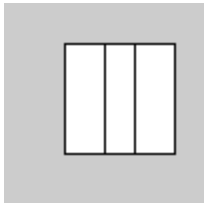
```
beginShape(TRIANGLE_STRIP);
vertex(30, 75);
vertex(40, 20);
vertex(50, 75);
vertex(60, 20);
vertex(70, 75);
vertex(80, 20);
vertex(90, 75);
endShape();
```



```
beginShape(TRIANGLE_FAN);
vertex(57.5, 50);
vertex(57.5, 15);
vertex(92, 50);
vertex(57.5, 85);
vertex(22, 50);
vertex(57.5, 15);
endShape();
```

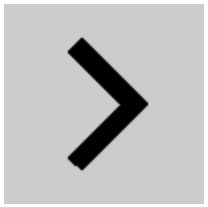


```
beginShape(QUADS);
vertex(30, 20);
vertex(30, 75);
vertex(50, 75);
vertex(50, 20);
vertex(65, 20);
vertex(65, 75);
vertex(85, 75);
vertex(85, 20);
endShape();
```

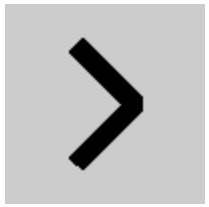


```
beginShape(QUAD_STRIP);
vertex(30, 20);
vertex(30, 75);
vertex(50, 20);
vertex(50, 75);
vertex(65, 20);
vertex(65, 75);
vertex(85, 20);
vertex(85, 75);
endShape();
```

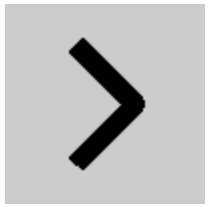
strokeJoin()



```
noFill();
smooth();
strokeWeight(10.0);
strokeJoin(MITER);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();
```



```
noFill();
smooth();
strokeWeight(10.0);
strokeJoin(BEVEL);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();
```



```
noFill();
smooth();
strokeWeight(10.0);
strokeJoin(ROUND);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();
```

Example Sketches...

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

Dropbox

- https://www.dropbox.com/