

# FlashCamp: Enhanced Drawing API in Flash Player 10

Trevor McCauley

Flash Player Quality  
Engineer 

October 11, 2008





- Streamlined path data with **Vectors**
- APIs Targeting **3D** Applications
- Drawing API **Data Objects**



- Typed arrays
  - Can only contain values of a certain type
- Improved performance

```
var myVector:Vector.<Number> = new Vector.<Number>( );  
myVector.push(1, 2, 3.5, 6.75);
```



- `Graphics.drawPath()`
- `Graphics.drawTriangles()`
- `Graphics.drawGraphicsData()`

- **Graphics.drawPath()**

```
drawPath( commands:Vector.<int> = null,  
          data:Vector.<Number> = null,  
          winding:String = "evenOdd" ):void
```



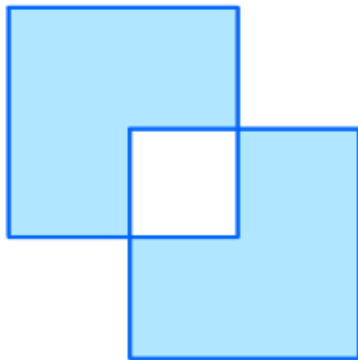
# Path Commands

Command	Value	Constant	Data Points
<code>moveTo()</code>	1	<code>GraphicsPathCommand.MOVE_TO</code>	2
<code>lineTo()</code>	2	<code>GraphicsPathCommand.LINE_TO</code>	2
<code>curveTo()</code>	3	<code>GraphicsPathCommand.CURVE_TO</code>	4
<i>wideMoveTo</i>	4	<code>GraphicsPathCommand.WIDE_MOVE_TO</code>	4
<i>wideLineTo</i>	5	<code>GraphicsPathCommand.WIDE_LINE_TO</code>	4

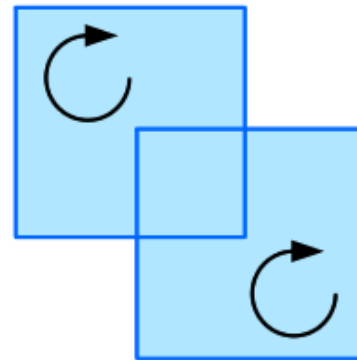
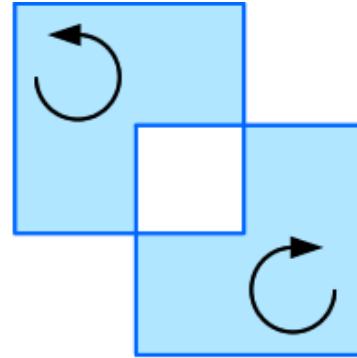


- Lists x and y coordinates
- Format:  
 $x_1, y_1, x_2, y_2, \dots, x_n, y_n$
- Length should always be even
- Data points should match commands

# Winding Fills

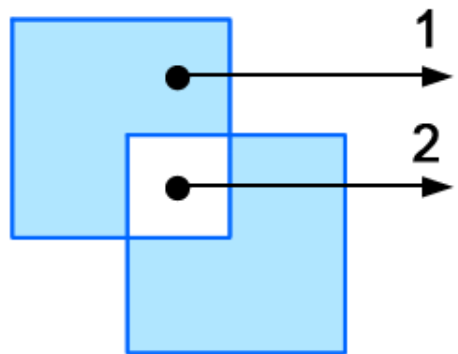


Even-odd Winding Rule

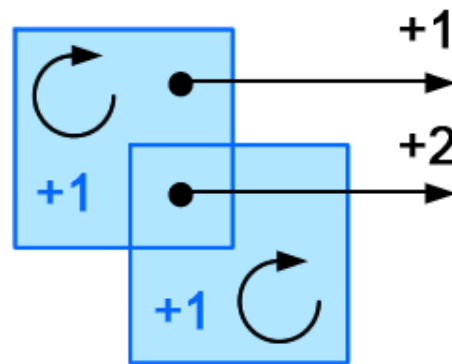
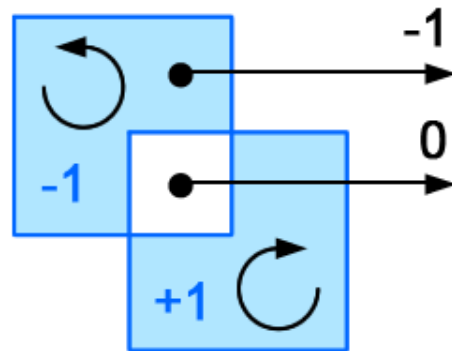


Non-zero Winding Rule

# Winding Fill Rules



Even-odd Winding Rule



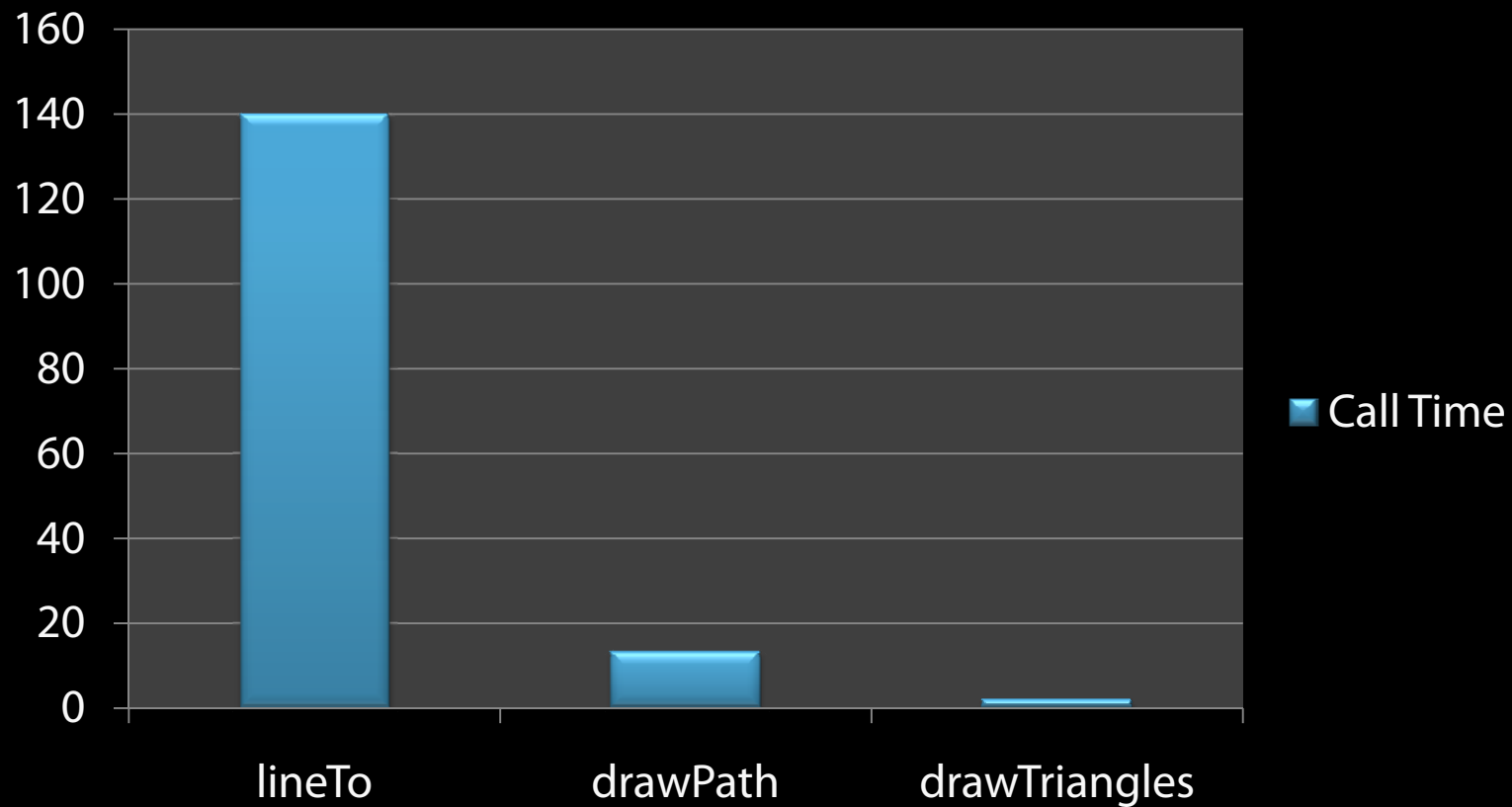
Non-zero Winding Rule



# Demo

## Shopping Cart Vectors

## API Call Time (ms)



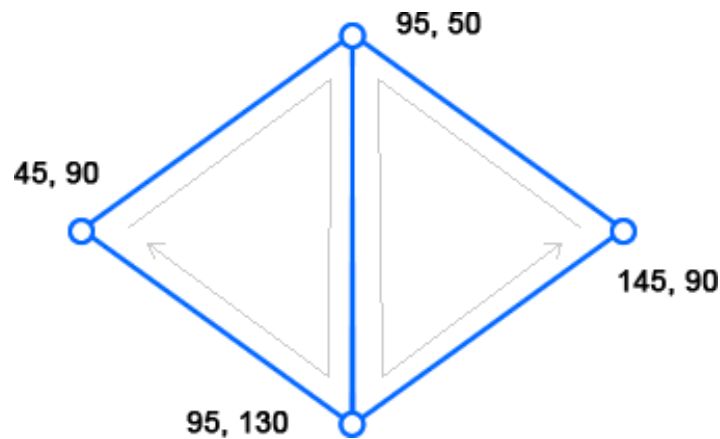
\*Mac OS 10.4, Intel Xeon, dual core 2.6 GHz, 2GB RAM

\*\* drawTriangles called with indices

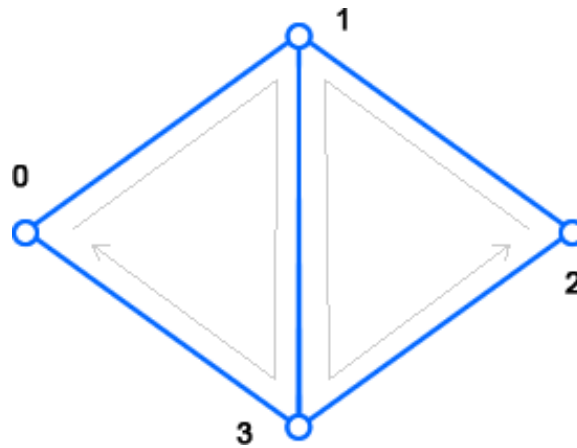
## ■ Graphics.drawTriangles()

```
drawTriangles(vertices:Vector.<Number>,  
              indices:Vector.<int> = null,  
              uvaData:Vector.<Number> = null,  
              culling:String = "none"):void
```

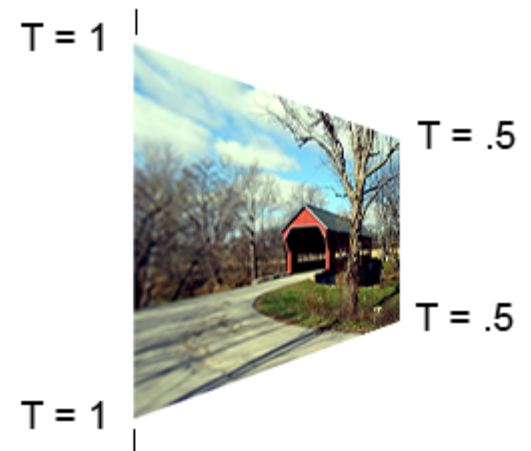
- Similar to drawPath data
  - List of x, y, coordinates
- Coordinates must specify triangles



- Specifies triangles in vertices list
- Reduces redundancy
  - Coordinates being shared by triangles are only defined once



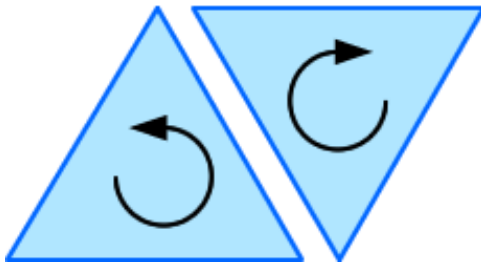
- UV Maps a bitmap to vertices
  - U: horizontal location (x) in % (0-1)
  - V: vertical location (y) in % (0-1)
- T determines perspective
  - Equal to scale ratio at that location in z space
  - In UVT, T is Optional



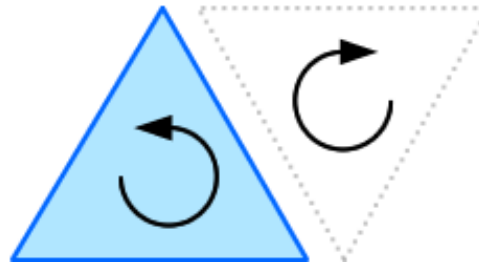


- Determines triangle sidedness (visibility)
  - Based on winding

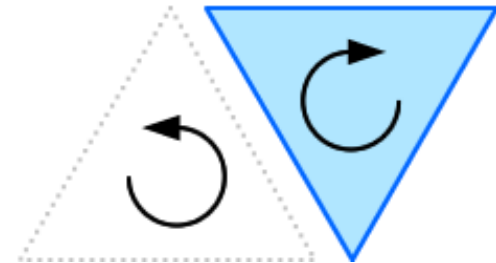
None



Positive



Negative





# Demo

Perspective Mark Sousa  
Rotating 3D Earth

- **Graphics.drawGraphicsData()**

```
drawGraphicsData(  
    graphicsData:Vector.<IGraphicsData>):void
```



# IGraphicsData Objects

Method	Data Object Class
beginFill	GraphicsSolidFill
beginBitmapFill	GraphicsBitmapFill
beginGradientFill	GraphicsGradientFill
beginShaderFill	GraphicsShaderFill
lineStyle	GraphicsStroke + GraphicsSolidFill
lineBitmapStyle	GraphicsStroke + GraphicsBitmapFill
lineGradientStyle	GraphicsStroke + GraphicsGradientFill
lineShaderStyle	GraphicsStroke + GraphicsShaderFill
moveTo lineTo curveTo drawPath	GraphicsPath
drawTriangles	GraphicsTrianglePath

# Data Object Advantages

- Portable
- Editable
- Native to Flash Player



# Demo

Wendy Carter as IGraphicsData

## Things to watch out for

- Graphics Path objects do not instantiate their own `Vectors`
- For bitmap fills, repeat defaults to `true` and `smooth` to `false`
- Be careful of `new Vector()` vs. `Vector([])` conversions



**Adobe**