FingerPainter

Lesson 5

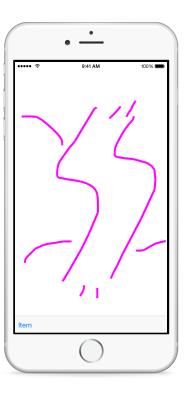


Description

Implement the drawing method with Core Graphics.

Learning Outcomes

- Recognize the procedural steps to draw lines with Core Graphics.
- Describe the procedure of performing drawing operations within an image context, and updating a UIImageView with the resulting image.



Vocabulary

context	point	path
stroke	CGPoint	CGContextRef
UIImage	UIImageView	

Materials

FingerPainter Lesson 5 Xcode project

Opening

How might you implement the actual drawFromPoint:toPoint: method in the controller?

Agenda

- Discuss how, similar to drawing a single line with Core Graphics in Lesson 1, one must follow a similar idiom when implementing drawFromPoint:toPoint:: obtaining a drawing context, setting the stroke color and width, creating a path with points, and then stroking the path.
- Update the implementation of drawFromPoint:toPoint:.

```
func drawFromPoint(start: CGPoint, toPoint end: CGPoint){
  // set the context to that of an image
  UIGraphicsBeginImageContext(canvas.frame.size)
  let context = UIGraphicsGetCurrentContext()
   // draw the existing image onto the current context
  canvas.image?.drawInRect(CGRect(x: 0, y: 0,
     width: canvas.frame.size.width,
     height: canvas.frame.size.height))
   // draw the new line segment
  CGContextSetLineWidth(context, 5)
  CGContextSetStrokeColorWithColor(context,
     UIColor.magentaColor().CGColor)
  CGContextBeginPath(context)
  CGContextMoveToPoint(context, start.x, start.y)
  CGContextAddLineToPoint(context, end.x, end.y)
  CGContextStrokePath(context)
   // obtain a UIImage object from the context
   let newImage = UIGraphicsGetImageFromCurrentImageContext()
  UIGraphicsEndImageContext()
  // Set the UIImageView's image to the new, generated image
  canvas.image = newImage
}
```

- Explain how the call to UIGraphicsBeginImageContext establishes that all drawing operations will take place on an image, rather than the view itself.
- Discuss how subsequent drawing operations will occur to this "cached" image managed by Core Graphics.
- Discuss how the first drawing operation is calling the drawInRect: method on the UIImageView's current image, which tells it to "draw itself onto the current context." In other words, the lines already drawn on the UIImageView's image are first drawn on the new context, so that they do not disappear after subsequent calls to drawFromPoint:toPoint:

- Discuss the drawing functions CGContextSetLineWidth, CGContextSetStrokeColorWithColor, CGContextBeginPath, CGContextMoveToPoint, CGContextAddLineToPoint and CGContextStrokePath.
- Explain how the results of the drawing operations are obtained as a UIImage object from the UIGraphicsGetImageFromCurrentImageContext function.
- Explain how the resulting image is then assigned to the image property of the controller's UIImageView canvas.
- Discuss how, for every call to drawFromPoint:toPoint:, a new UIImage is generated, the existing UIImageView image is drawn onto the new UIImage, the new line is drawn on the new UIImage, and then the new UIImage is assigned to the UIImageView for continuous display.
- Run the app (***R**), click and drag the mouse on the screen to simulate a finger movement, and observe the line drawing.

Closing

What happens if you delete the line that calls canvas.image?.drawInRect:?

Modifications and Extensions

• Modify the drawing implementation such that the line changes color or thickness the longer the entire moving touch event is.

Resources

Drawing and Printing Guide for iOS: iOS Drawing Concepts https://developer.apple.com/library/ios/documentation/2DDrawing/Conceptual/DrawingPrintingiOS/GraphicsDrawingOverview/GraphicsDrawingOverview.html

View Programming Guide: Creating Views https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/CreatingViews/CreatingViews.html

Quartz 2D Programming Guide http://developer.apple.com/library/ios/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d/Introduction/Introduction.html

CGContext Reference https://developer.apple.com/library/ios/documentation/ GraphicsImaging/Reference/CGContext/index.html

UllmageView Class Reference https://developer.apple.com/library/ios/documentation/UlKit/Reference/UllmageView_Class/index.html

Ullmage Class Reference https://developer.apple.com/library/ios/documentation/ UlKit/Reference/Ullmage_Class/index.html

JIColor Class Reference https://developer.apple.com/library/ios/documentation/ JIKit/Reference/UIColor_Class/index.html		