

Beginning

Core Graphics

Part 3: Contexts 1

Core Graphics Hands-On Challenges

Copyright © 2016 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.



Challenge: Draw an Icon

In this challenge you'll draw the Pet category icon. You'll do this in a playground just as we did in the demo.

At the end of the challenge you'll end up with this path:



Create a new iOS playground. Ctrl click **Playgrounds** folder and choose **New File**. Choose **iOS \ Source \ Playground**. Save the file as **IconPet**.

The playground will be similar to IconFun except for the actual drawing paths.

First set up the variables needed for drawing – replace the existing playground code with:

```
import UIKit

let size = CGSize(width: 400, height: 400)
let rect = CGRect(origin: .zero, size: size)

let backgroundColor = UIColor.redColor()
let drawingColor = UIColor.blackColor()

let lineWidth:CGFloat = 5.0
```

Set up the context:

```
UIGraphicsBeginImageContextWithOptions(size, false, 0.0)
```



This creates a drawing context of 400 by 400 points with transparency at a scale determined by the current device.

End the context and extract the image. These three lines must stay at the end of the playground until you've completed drawing the paths.

```
let image = UIGraphicsGetImageFromCurrentImageContext()  
UIGraphicsEndImageContext()  
image
```

Show the image in the playground by clicking the + next to `image`.



Between beginning and ending the context, before extracting the image, get a reference to the current context:

```
let context = UIGraphicsGetCurrentContext()
```

Remember all the drawing commands must be done before you end the context.

Create a path for the outline of the icon. This is a rounded rectangle.

```
let edge = UIBezierPath(roundedRect: rect.insetBy(dx: lineWidth*2,  
dy: lineWidth*2), cornerRadius: 50)  
edge.lineWidth = lineWidth
```

Save the state of the context and add a clipping path:

```
CGContextSaveGState(context)  
edge.addClip()
```



Fill the context with the background color:

```
backgroundColor.setFill()  
UIRectFill(rect)
```

The image in the playground should now look like this:



The edge path clips the red background fill so that the image has rounded corners.
Restore the context to the state before the clip was added:

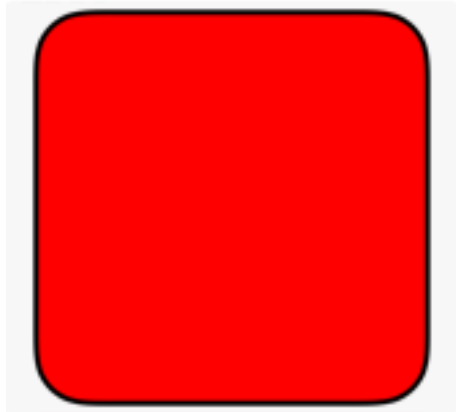
```
CGContextRestoreGState(context)
```

Stroke the outline with the edge path:

```
drawingColor.setStroke()  
edge.stroke()
```



The image now has a black outline:



Add the first part of the paw - this will be a circle:

```
let paw1 = UIBezierPath(ovalInRect: CGRect(x: 130, y: 174, width:
140, height: 154))
paw1.lineWidth = lineWidth
paw1.stroke()
```

and then add the three other smaller circles:

```
let paw2 = UIBezierPath(ovalInRect: CGRect(x: 71, y: 104, width:
71, height: 89))
paw2.lineWidth = lineWidth
paw2.stroke()

let paw3 = UIBezierPath(ovalInRect: CGRect(x: 161, y: 66, width:
71, height: 89))
paw3.lineWidth = lineWidth
paw3.stroke()

let paw4 = UIBezierPath(ovalInRect: CGRect(x: 254, y: 104, width:
71, height: 89))
paw4.lineWidth = lineWidth
paw4.stroke()
```



Now the image has the paw drawn:



Group all the paw paths into one path:

```
let path = UIBezierPath()  
path.appendPath(paw1)  
path.appendPath(paw2)  
path.appendPath(paw3)  
path.appendPath(paw4)
```

At the very end of the playground create a second drawing context so that you can see what the path looks like:

```
UIGraphicsBeginImageContextWithOptions(size, false, 0)  
  
// .. drawing goes here  
  
let image2 = UIGraphicsGetImageFromCurrentImageContext()  
UIGraphicsEndImageContext()  
image2
```

Inside the new context where it says `// .. drawing goes here`, stroke the path:

```
path.stroke()
```



Click the + next to `image2` to see the result.



You've now created a reusable path. Later on in the series, you'll use this path in the app code for your expense category icons.

