Beginning

Core Graphics

Part 4: Contexts 2

Core Graphics Hands-On Challenges

Challenge: Expense Icon

Now you'll transfer your pet icon code that you created in the previous challenge from the playground to the DrawIcon class that we created in the demo. The pet icon will then show on the expense screen along with the smiley face Fun icon.

We already set up the cell drawing code in the demo, so you should just be able to define the drawing code and make sure it's called from the category.

First create a new class method in DrawIcon:

```
    class func drawPetIcon() -> UIBezierPath {
    }
```

In `drawPetIcon()` set the size of the context:

```
  let size = CGSize(width: 400, height: 400)
```

Copy the relevant code from your IconPet Playground from `UIGraphicsBeginImageContextWithOptions(size, false, 0.0)` to `UIGraphicsEndImageContext()` and paste it in `DrawIcon`'s `drawPetIcon()` method.

Clean up the code:

1. Remove the context variable and all edge drawing.
2. Remove all line width settings and stroke drawing.
3. Remove getting the image from the context

You should be left with this code:

```
  class func drawPetIcon() -> UIBezierPath {
    let size = CGSize(width: 400, height: 400)
    UIGraphicsBeginImageContextWithOptions(size, false, 0.0)

    let paw1 = UIBezierPath(ovalInRect: CGRect(x: 130, y: 174,
width: 140, height: 154))
    let paw2 = UIBezierPath(ovalInRect: CGRect(x: 71, y: 104,
width: 71, height: 89))
    let paw3 = UIBezierPath(ovalInRect: CGRect(x: 161, y: 66,
width: 71, height: 89))
    let paw4 = UIBezierPath(ovalInRect: CGRect(x: 254, y: 104,
width: 71, height: 89))

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

    UIGraphicsEndImageContext()
 }
```

At the end of `drawPetIcon()` return the path.

```
    return path
```

In `Category.swift` in `drawIcon()`, add this to the switch:

```
    case "Pet":
      path = DrawIcon.drawPetIcon()
```

When drawIcon is called, if the category's name is **Pet**, `drawPetIcon()` will be called and your drawing code will create the path which is stored in the category.

Build and run the application, choose the **Pet** category and see your paw print in action.