BEGINNING METAL

HANDS-ON CHALLENGES

Beginning Metal

Caroline Begbie

Notice of Rights

Notice of Liability

Trademarks

Challenge #11: Touch Handlers

By Caroline Begbie

In this challenge, you're going to add touch ability to your scenes. By the end of the challenge you'll be able to rotate the mushroom along its x and y axes.

Touch events are detected by the view controller, so we'll set up the touch event handlers there.

In **ViewController.swift**, add these four touch methods to `ViewController`:

```
override func touchesBegan(_ touches: Set<UITouch>,
                           with event: UIEvent?) {
  renderer?.scene?.touchesBegan(view, touches:touches,
                                with: event)
}

override func touchesMoved(_ touches: Set<UITouch>,
                           with event: UIEvent?) {
  renderer?.scene?.touchesMoved(view, touches: touches,
                                with: event)
}

override func touchesEnded(_ touches: Set<UITouch>,
                           with event: UIEvent?) {
  renderer?.scene?.touchesEnded(view, touches: touches,
                                with: event)
}

override func touchesCancelled(_ touches: Set<UITouch>,
                               with event: UIEvent?) {
  renderer?.scene?.touchesCancelled(view, touches: touches,
                                    with: event)
}
```

That's one touch handler for every touch event. Your app should have a build error until we've added the scene methods.

In `Scene`, add the touch handlers so that each scene subclass can override them:

```
func touchesBegan(_ view: UIView, touches: Set<UITouch>,
                  with event: UIEvent?) {}

func touchesMoved(_ view: UIView, touches: Set<UITouch>,
                  with event: UIEvent?) {}

func touchesEnded(_ view: UIView, touches: Set<UITouch>,
                  with event: UIEvent?) {}

func touchesCancelled(_ view: UIView, touches: Set<UITouch>,
                      with event: UIEvent?) {}
```

Your app should now compile again.

In **LightingScene.swift**, override `touchesBegan(_:touches:with:)` and `touchesMoved(_:touches:with:)`:

```
override func touchesBegan(_ view: UIView, touches: Set<UITouch>,
                           with event: UIEvent?) {
}

override func touchesMoved(_ view: UIView, touches: Set<UITouch>,
                           with event: UIEvent?) {
}
```

Still in `LightingScene`, set up a property to hold the previous touch location:

```
var previousTouchLocation: CGPoint = .zero
```

Store the first touch location. Add this to `touchesBegan(_:touches:with:)`:

```
guard let touch = touches.first else { return }
previousTouchLocation = touch.location(in: view)
```

In `touchesMoved(_:touches:with:)`, calculate the difference between the current touch location and the previous touch location:

```
guard let touch = touches.first else { return }
let touchLocation = touch.location(in: view)

let delta = CGPoint(x: previousTouchLocation.x - touchLocation.x,
                    y: previousTouchLocation.y - touchLocation.y)
```

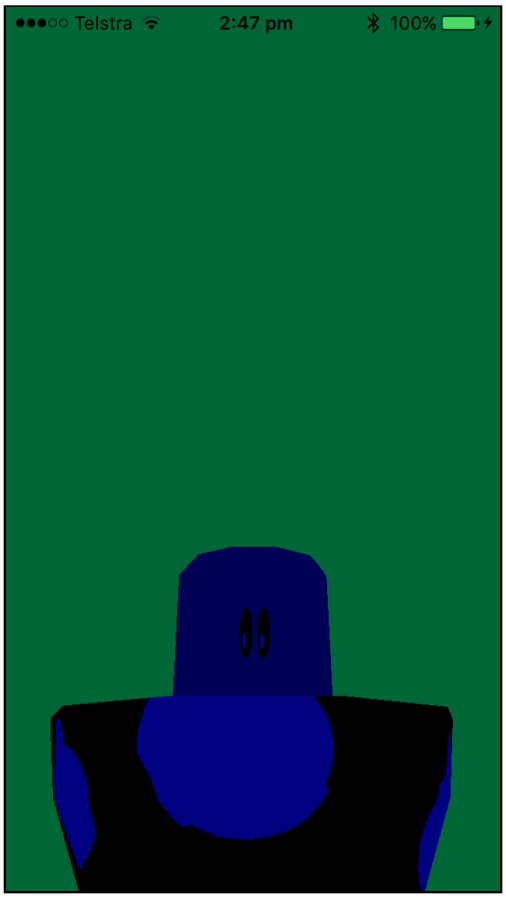
Still in `touchesMoved(_:touches:with:)`, rotate the mushroom by this difference, but stepping it down by a sensitivity factor:

```
let sensitivity: Float = 0.01
mushroom.rotation.x += Float(delta.y) * sensitivity
mushroom.rotation.y += Float(delta.x) * sensitivity
```

Then at the end of `touchesMoved(_:touches:with:)`, store the current touch location ready for the next move:

```
previousTouchLocation = touchLocation
```

Build and run, and you should be able to rotate the mushroom around its origin.



Now you've implemented touch, your future games will be so much more interesting :].