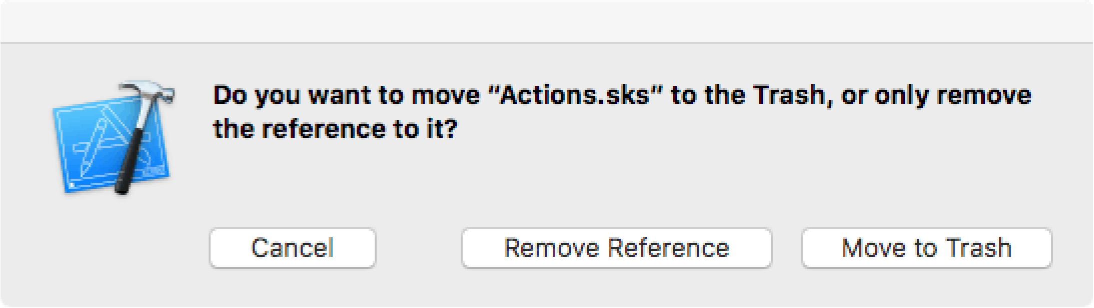
Inclass Activity on SKSimpleGame

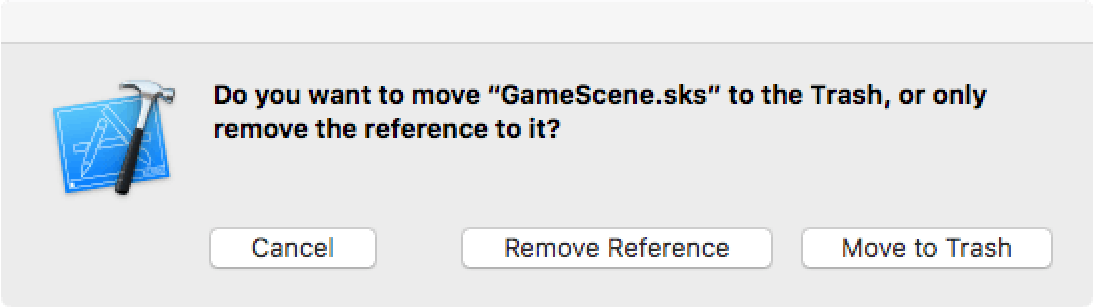
We'll learn how to create an object and add it to the scene. Next we'll learn how to move the object using UITapGestureRecognizer which listens to the tap event.

1. Create a new XCode project → select **Game** → Next → provide a Product Name: **GameSpriteAction** → Next → Create

2. The game template provides a number of files, a couple of them (GameScene.sks and Actions.sks) are related to Scene and Action which we don't use for a simple game. Delete both of them **to the Trash!**



Do you want to move "Actions.sks" to the Trash, or only remove the reference to it?



Do you want to move "GameScene.sks" to the Trash, or only remove the reference to it?

3. Empty the code inside the **GameViewController.swift** file leaving a few lines in the file like this below:

```
import UIKit
import SpriteKit
import GameplayKit

class GameViewController: UIViewController {

    override func viewDidLoad() {

    }

}
```

4. Next add a few lines of code inside **viewDidLoad()** method to create a GameScene object and present it to show and run the GameScene.swift file:

```
import UIKit
import SpriteKit
import GameplayKit

class GameViewController: UIViewController {

    override func viewDidLoad() {
        let scene = GameScene(size: view.frame.size)
        let skView = view as! SKView
        skView.presentScene(scene)
    }

}
```

5. Next switch to **GameScene.swift** file → empty the code inside the GameScene class leaving a few lines of code like this below:

```
import SpriteKit
import GameplayKit

class GameScene: SKScene {

    override func didMove(to view: SKView) {

    }
}
```

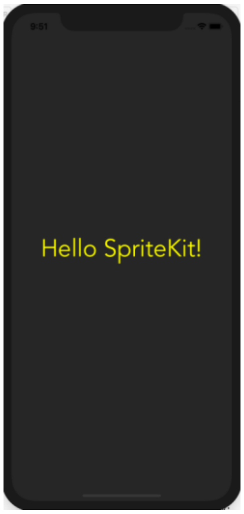
6. Add a node (e.g. SKLabelNode) to the scene inside the **didMove()** method, position it to the center, change the font size, font name and font color:

```
class GameScene: SKScene {

    let label = SKLabelNode(text: "Hello SpriteKit!")

    override func didMove(to view: SKView) {
        addChild(label)
        label.position = CGPoint(x: view.frame.width / 2, y: view.frame.height / 2)
        label.fontSize = 45
        label.fontColor = SKColor.yellow
        label.fontName = "Avenir"
     }
}
```

7. Run the app to see the result.



Hello SpriteKit!

8. Next we'll add a Gesture Recogniser to the scene. So add the following code into the didMove() method in the GameScene.swift file, followed by a tap method below the didMove() method:

```
    override func didMove(to view: SKView) {
        addChild(label)
        label.position = CGPoint(x: view.frame.width / 2, y: view.frame.height / 2)
        label.fontSize = 45
        label.fontColor = SKColor.yellow
        label.fontName = "Avenir"

        let recognizer = UITapGestureRecognizer(target: self, action: #selector(tap))
        view.addGestureRecognizer(recognizer)
    }

@objc func tap(recognizer: UIGestureRecognizer) {

}
```

9. Now we are creating SKAction to move the label node by adding the following code into the tap() method:

```
 @objc func tap(recognizer: UIGestureRecognizer) {
     let viewLocation = recognizer.location(in: view)
     let sceneLocation = convertPoint(fromView: viewLocation)

     let moveToAction = SKAction.move(to: sceneLocation, duration: 1)
     label.run(moveToAction)
}
```

10. Try different movements with the following sets of code”

i.

```
@objc func tap(recognizer: UIGestureRecognizer) {
    let viewLocation = recognizer.location(in: view)
    let sceneLocation = convertPoint(fromView: viewLocation)

    let moveByAction = SKAction.moveBy(x: sceneLocation.x - label.position.x, y: sceneLocation.y - label.position.y, duration: 1)
    label.run(moveByAction)
}
```

ii.

```
@objc func tap(recognizer: UIGestureRecognizer) {
    let viewLocation = recognizer.location(in: view)
    let sceneLocation = convertPoint(fromView: viewLocation)

    let moveByAction = SKAction.moveBy(x: sceneLocation.x - label.position.x, y: sceneLocation.y - label.position.y, duration: 1)
    let moveByReversedAction = moveByAction.reversed()
    let moveByActions = [moveByAction, moveByReversedAction]
    let moveSequence = SKAction.sequence(moveByActions)
    label.run(moveSequence)
}
```

iii.

```
@objc func tap(recognizer: UIGestureRecognizer) {
    let viewLocation = recognizer.location(in: view)
    let sceneLocation = convertPoint(fromView: viewLocation)

    let moveByAction = SKAction.moveBy(x: sceneLocation.x - label.position.x, y: sceneLocation.y - label.position.y, duration: 1)
    let moveByReversedAction = moveByAction.reversed()
    let moveByActions = [moveByAction, moveByReversedAction]
    let moveSequence = SKAction.sequence(moveByActions)
    let moveRepeatSequence = SKAction.repeat(moveSequence, count: 3)
    label.run(moveRepeatSequence)
}
```

iv.

```
@objc func tap(recognizer: UIGestureRecognizer) {
    let viewLocation = recognizer.location(in: view)
    let sceneLocation = convertPoint(fromView: viewLocation)

    let moveByAction = SKAction.moveBy(x: sceneLocation.x - label.position.x, y: sceneLocation.y - label.position.y, duration: 1)
    let moveByReversedAction = moveByAction.reversed()
    let moveByActions = [moveByAction, moveByReversedAction]
```

```
        let moveSequence = SKAction.sequence(moveByActions)
        let moveRepeatForeverSequence = SKAction.repeatForever(moveSequence)
        label.run(moveRepeatForeverSequence)
}
```

11. Run the app and tap on the screen to move the label node. Take screen a screen shot and submit it.