

Class 05-1

SpriteKit is Apple's exciting 2D game framework that was first released in September 2013 with iOS 7.

Sprite Kit

Spritekit Basics

- SKNode
 - ◆ SKNode是所有內容的基礎類別(包含SKScene)
 - ◆ SKLabelNode、SKSpriteNode、SKVideoNode 等皆是SKNode的子類別
- SKScene
 - ◆ 在Scene中，所有元素皆屬於樹狀的資料結構，而SKScene是根節點(root)，每個節點的位置會由父節點的座標系決定。
 - ◆ 一個遊戲可能會需要建立多個SKScene的子類別
(e.g. 遊戲主選單、遊戲場景、遊戲結束場景)
- SKAction
 - ◆ 此類別用來對SKNodes 設定動作 (e.g. 移動 縮放 旋轉 播放音訊)
- SKPhysicsBody
 - ◆ 此類別將物理特性添加到SKNodes，使其可以模擬出複雜的變化(碰撞 重力 摩擦力)

Spritekit Basics

- **SKView:** This is a subclass of the UIView. SKView is the view in which all of the SpriteKit contents are added and displayed on the screen.
- **SKNode:** This is the building block of SpriteKit. It is just an empty node. No visual elements. On a SKNode we can add child nodes such as SKSpriteNode. These nodes can be visual elements or a scene.
- **SKScene:** An SKScene is a subclass of the SKNode. It represents a single scene from the game. SKScene acts as a root node of the SKNode. SKScene provides the content to the SKView. You just need to call the function presentScene over the SKView and pass the scene instance.
- **SKAction:** An SKAction is used to set actions such as animations/movements on SKNodes.
- **SKPhysicsBody:** This class adds the physics components to the SKNodes. Components such as gravity, collision are commonly used in games.

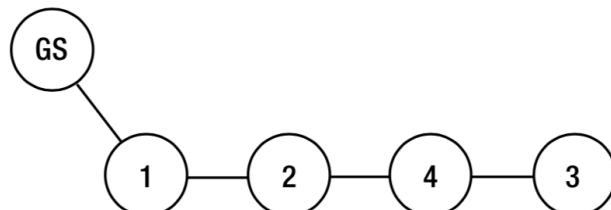
SpriteKit Node Tree

Method	Purpose
addChild()	The <code>addChild(_:)</code> method adds a node to the end of the receiver's collection of child nodes.
insertChild(_:at:)	The <code>insertChild(_:at:)</code> method inserts a child node at a specific position in the receiver's collection of child nodes.
removeFromParent()	<code>removeFromParent()</code> removes the receiving node from its parent.

```
var node1 = SKSpriteNode()
var node2 = SKSpriteNode()
var node3 = SKSpriteNode()
```

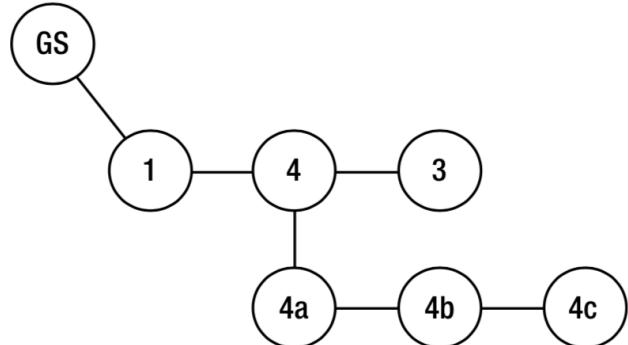
```
gameScene.addChild(node1)
gameScene.addChild(node2)
gameScene.addChild(node3)
```

```
var node4 = SKSpriteNode()
gameScene.insertChild(node4, at: 2)
```

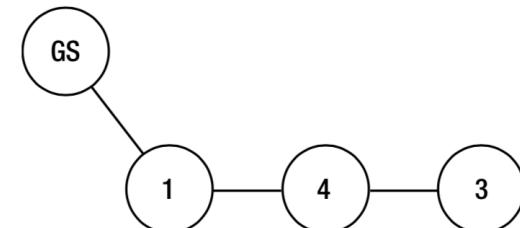


```
var node4a = SKSpriteNode()
var node4b = SKSpriteNode()
var node4c = SKSpriteNode()
```

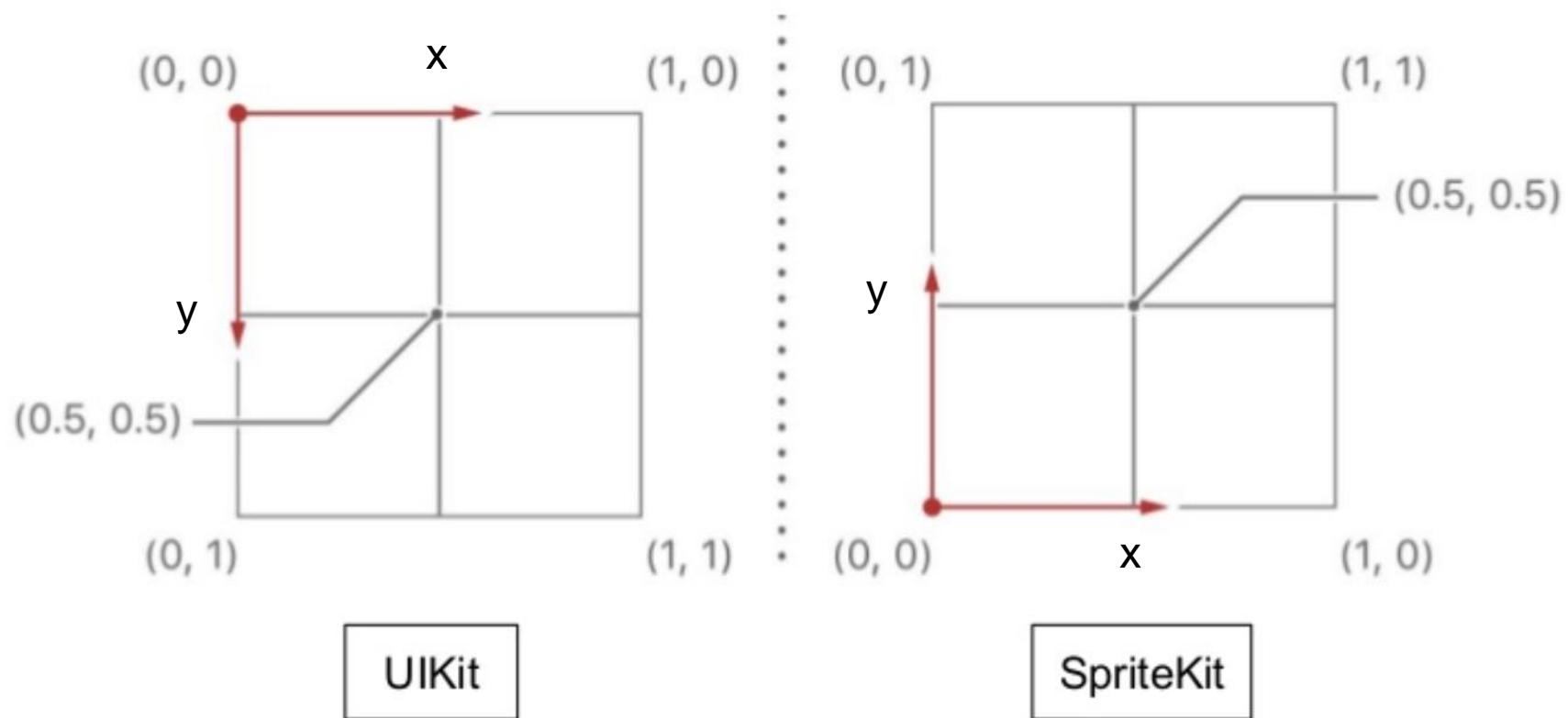
```
node4.addChild(node4a)
node4.addChild(node4b)
node4.addChild(node4c)
```



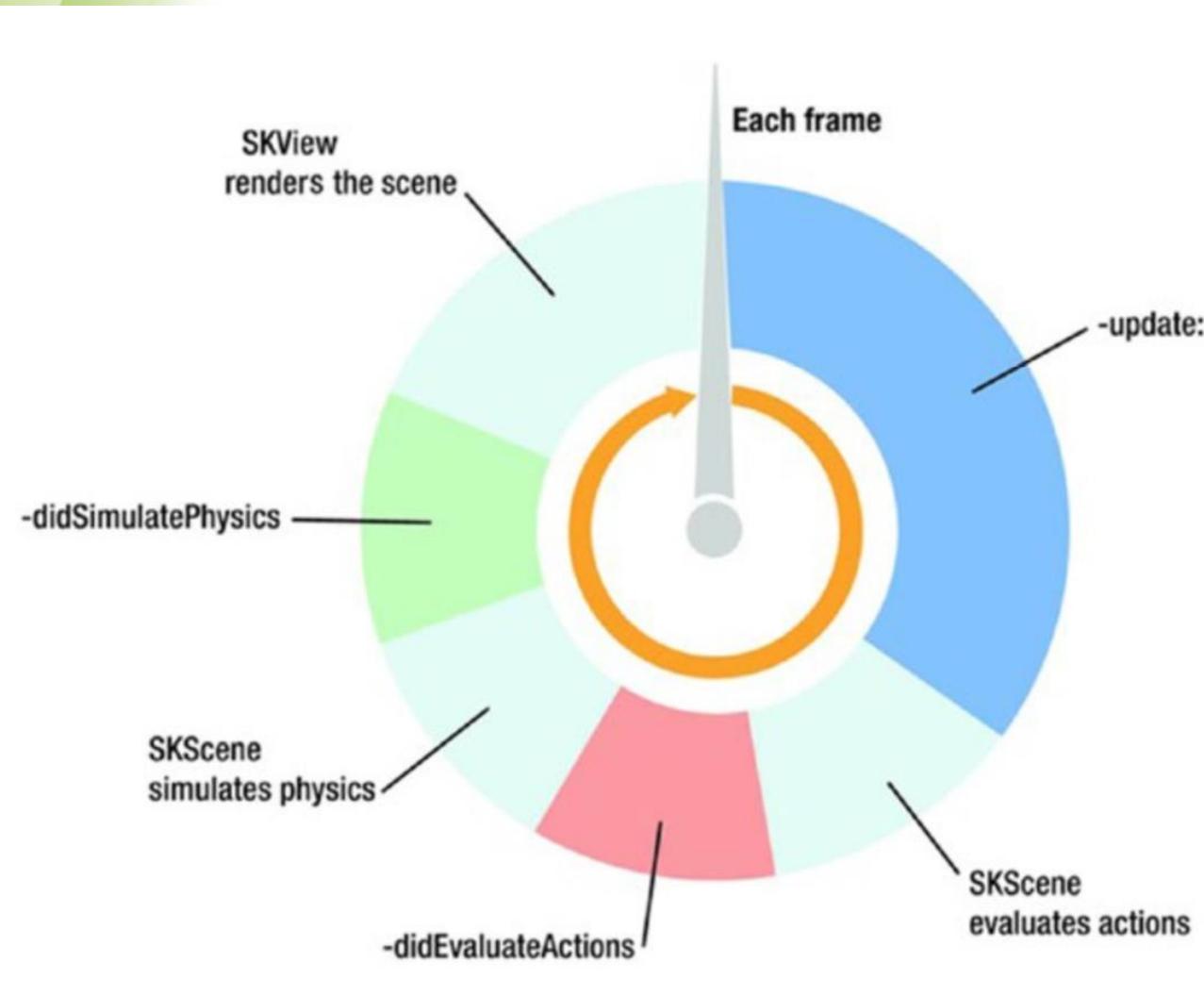
`node2.removeFromParent()`



SpriteKit Coordinate



Spritekit rendering loop



SpriteKit Example

Hello World !



Hello World !

10:25

1:46

iPhone 11 — 13.3

Tap here

Hello World !

iPhone 11 — 13.3

The image shows the Xcode welcome screen. On the left, there's a large icon of a hammer on blueprints with the word "XCODE" at the bottom. Below it, the text "Welcome to Xcode" is displayed in a large, white, sans-serif font. Underneath that, "Version 11.3.1 (11C504)" is shown in a smaller, gray font. To the right, a sidebar lists several projects:

- Firstsprite (~/Desktop)
- cpractice (~/Desktop)
- MyPlayground (~/Desktop)
- hellosprite (~/Desktop/2016 iOS/pr6 4h)
- hw1 (~/Desktop)
- 2-1 (~/Desktop)
- Calculator (~/Desktop)

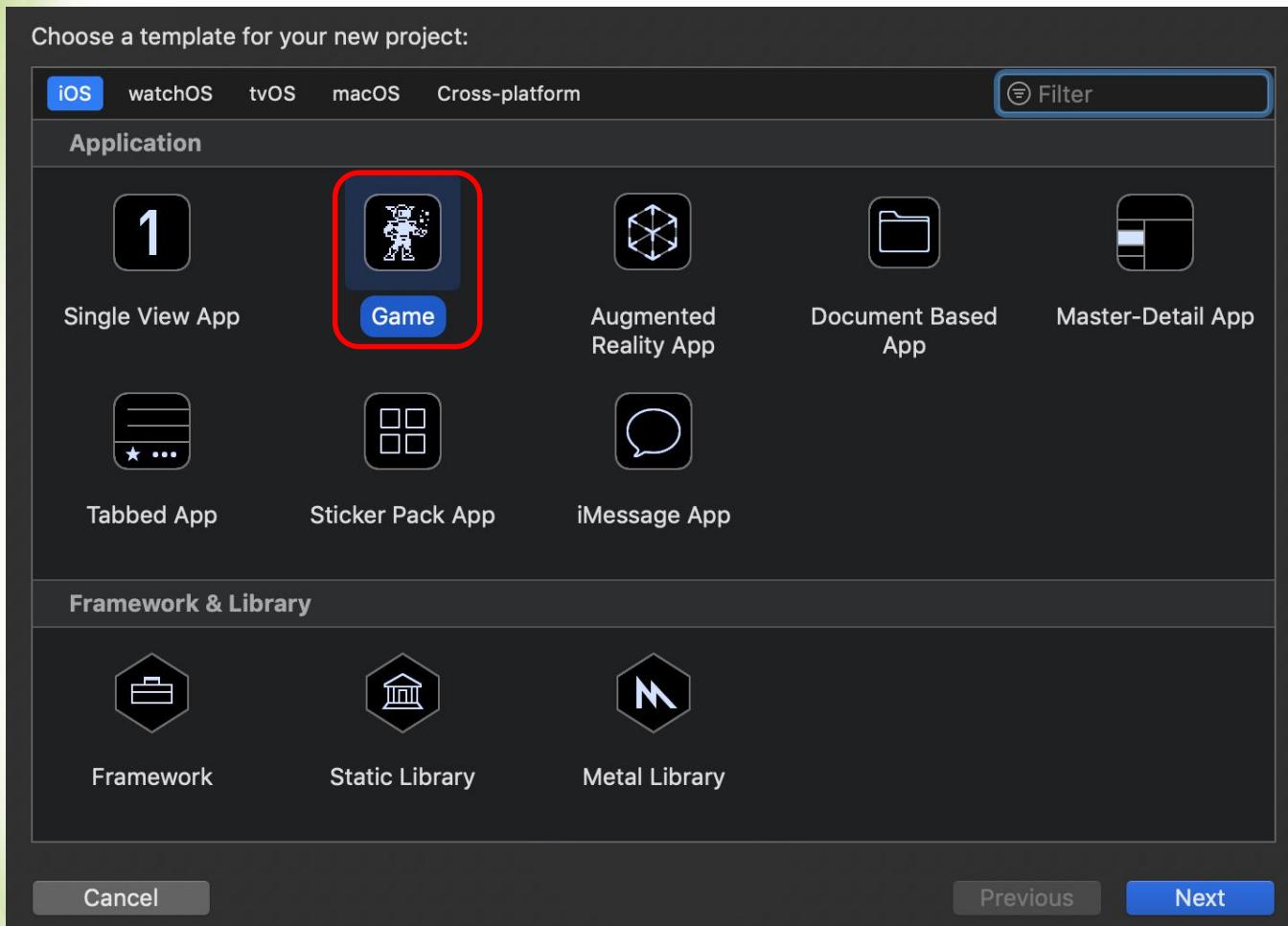
At the bottom left, there are three options with icons:

- Get started with a playground**: An icon of a bird on a document. Description: Explore new ideas quickly and easily.
- Create a new Xcode project**: An icon of a document with a play button symbol. Description: Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV. This option is highlighted with a red rounded rectangle.
- Clone an existing project**: An icon of a document with a circular arrow symbol. Description: Start working on something from a Git repository.

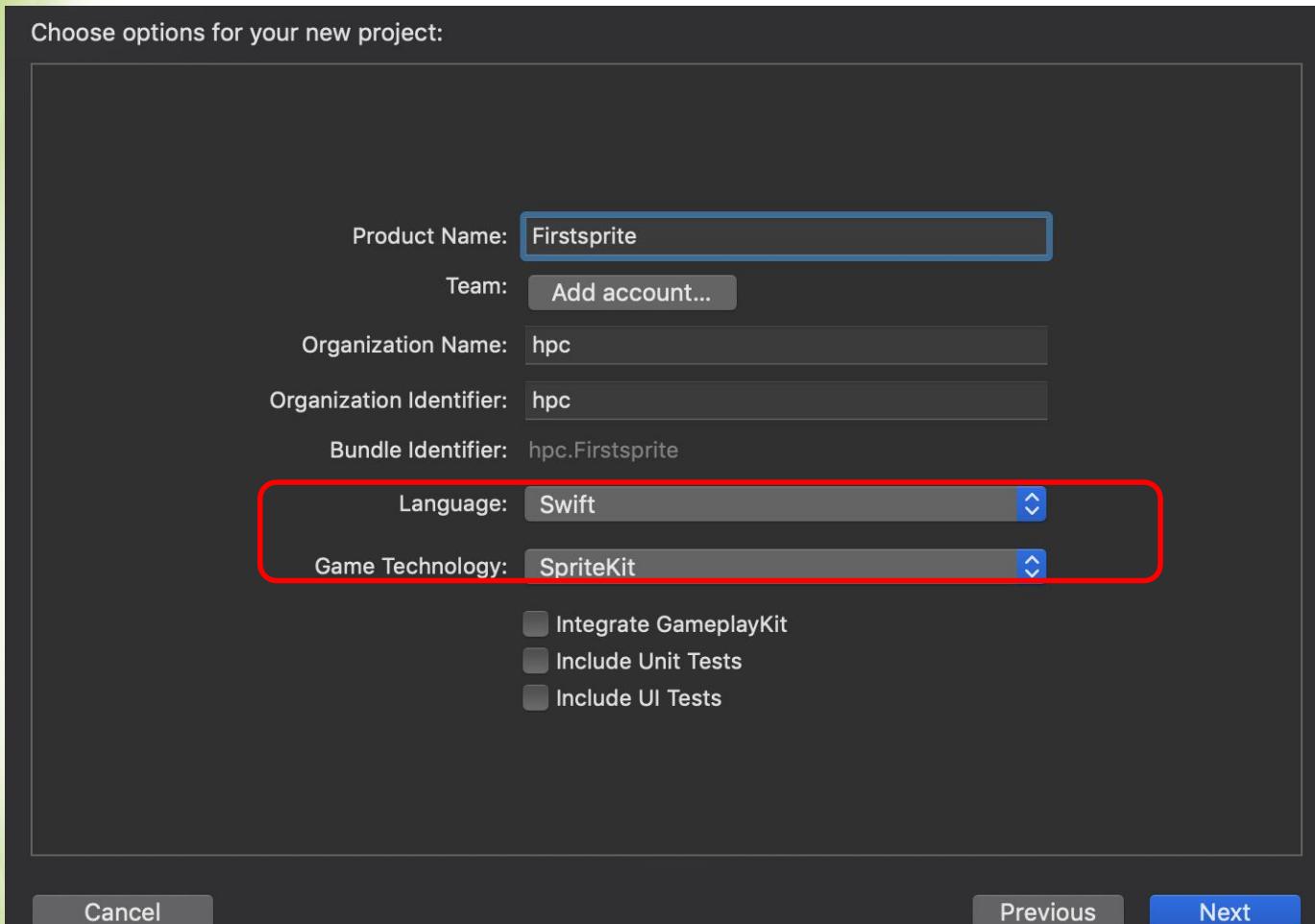
At the bottom left, there's a checked checkbox labeled "Show this window when Xcode launches". At the bottom right, there's a link "Open another project...".

Hello Sprite Kit!

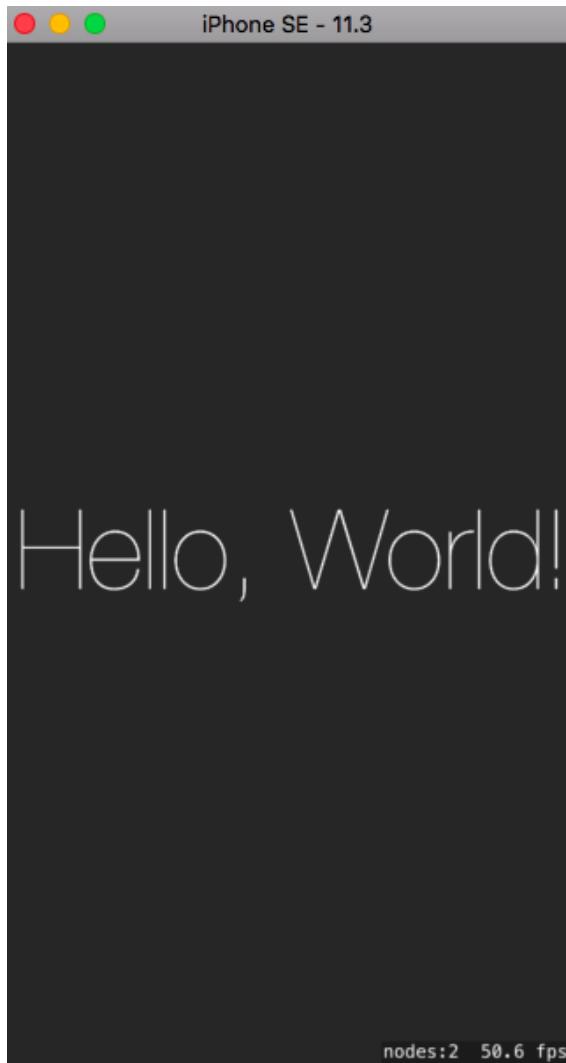
- Create Sprite Kit Game

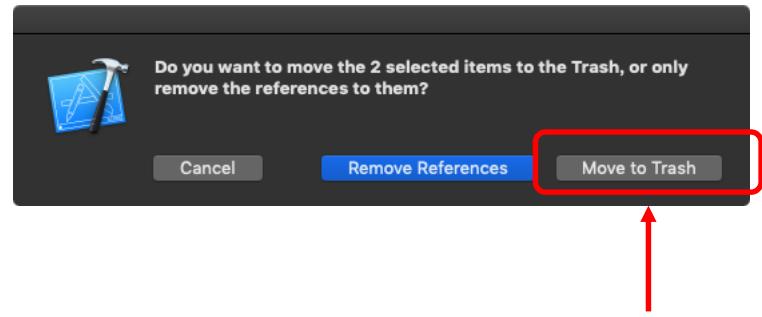
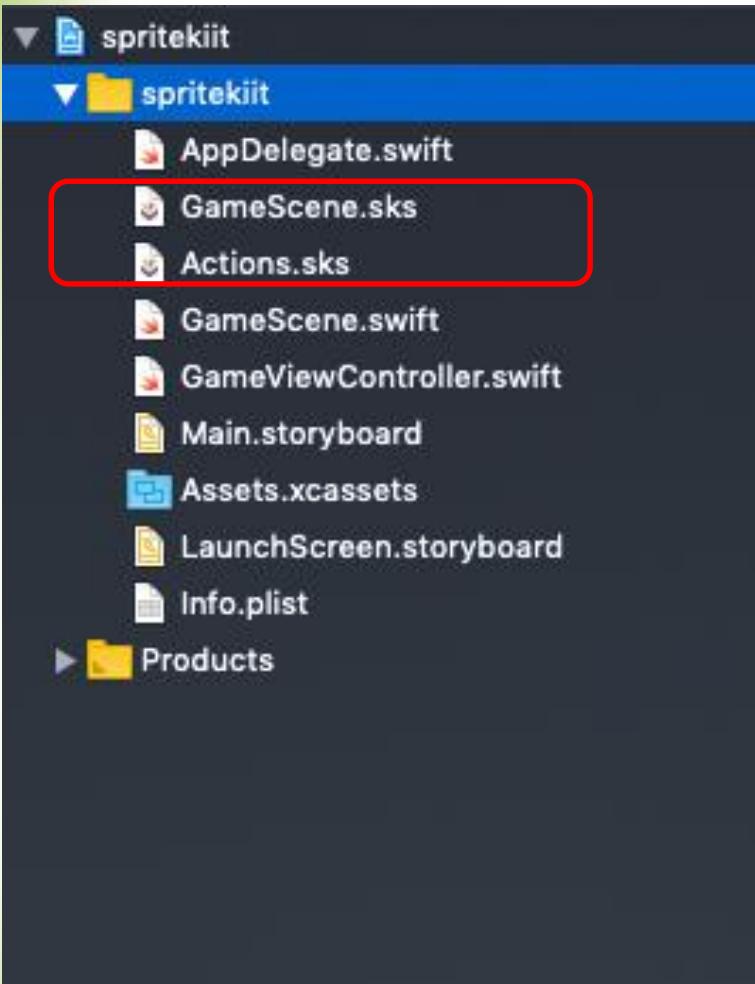


- Named project

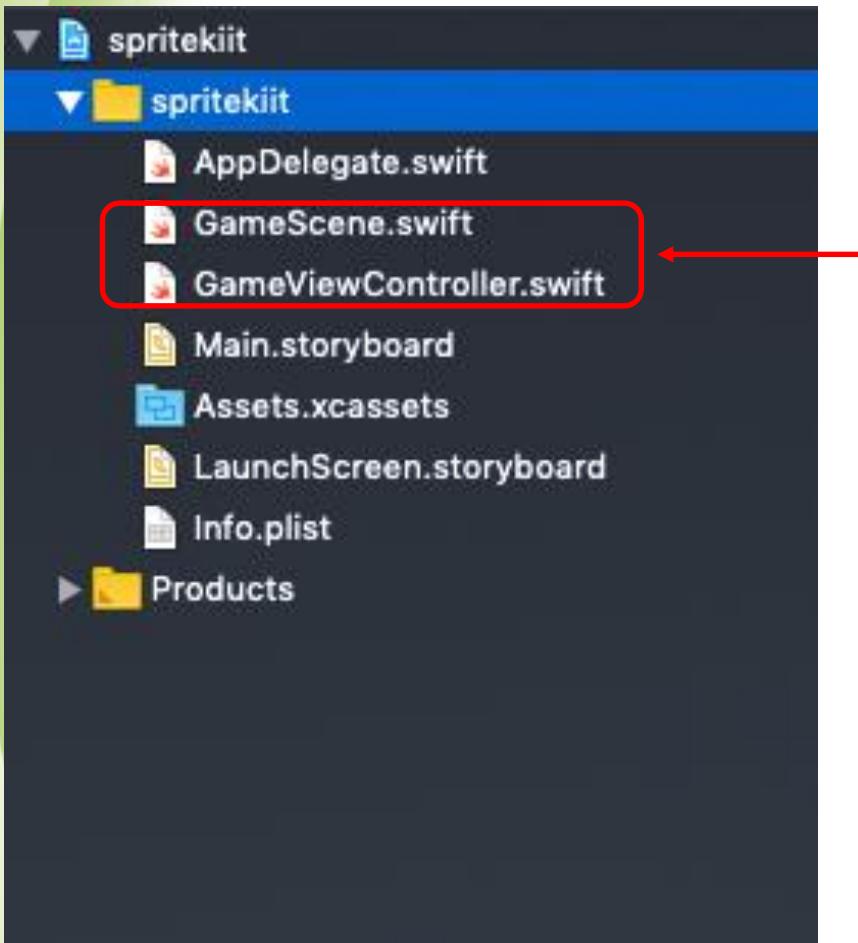


Run it and dragging your mouse to make some rectangles





Make sure you permanently
move the files to the trash



Delete everything inside
Not the entire file !

GameViewController.swift

```
1 import SpriteKit  
2  
3  
4 class GameViewController: UIViewController {  
5     override func viewDidLoad() {  
6         let scene = GameScene(size: view.frame.size)  
7         let skView = view as! SKView  
8         skView.presentScene(scene)  
9     }  
10 }  
11 }
```

The func Runs after the viewcontroller is loaded into memory

force downcast the controller's view to a SpriteKit view

GameScene.swift

Change to your style

#selector is from objective-c syntax

```
import SpriteKit

class GameScene: SKScene {
    let label = SKLabelNode(text: "Hello World !")

    override func didMove(to view: SKView) {
        label.position = CGPoint(x: view.frame.width / 2, y: view.frame.height / 2)
        label.fontSize = 45
        label.fontColor = SKColor.red
        label.fontName = "Avenir"
        label.speed = 5

        let recognizer = UITapGestureRecognizer(target: self, action: #selector(tap))
        view.addGestureRecognizer(recognizer)
        addChild(label) //add the label as a child node to the scene's node tree.
    }
}
```

Create a tap gesture recognizer
and add it to the scene's view.

Select which
method do you
want to call

GameScene.swift

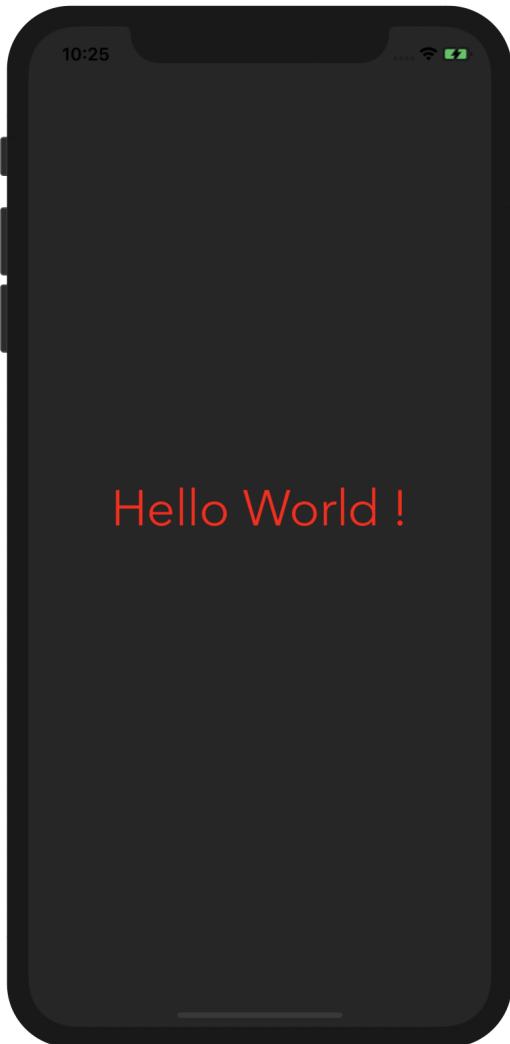
Not in func didMove

convertPoint(fromView:) Converts a point from the coordinate system of a given view to that of the receiver.

```
22 @objc func tap(recognizer: UIGestureRecognizer) {  
23     let viewLocation = recognizer.location(in: view)  
24     let sceneLocation = convertPoint(fromView: viewLocation)  
25     let moveToAction = SKAction.move(to: sceneLocation, duration: 1)  
26     label.run(moveToAction)  
27 }
```

This is the expected duration of an action's animation.

Run it



```
1 import SpriteKit  
2  
3 class GameScene: SKScene {  
4     let label = SKLabelNode(text: "Hello World !")  
5     var txtchange :Bool = false  
6  
7     override func didMove(to view: SKView) {  
8         label.position = CGPoint(x: view.frame.width / 2, y:  
9             view.frame.height / 2)  
10        label.fontSize = 45  
11        label.fontColor = SKColor.red  
12        label.fontName = "Avenir"  
13        label.speed = 5  
14  
15        let recognizer = UITapGestureRecognizer(target: self, action:  
16            #selector(tap))  
17        let dtaprecognizer = UITapGestureRecognizer(target: self,  
18            action: #selector(doubletap))  
19        dtaprecognizer.numberOfTapsRequired = 2  
20        view.addGestureRecognizer(recognizer)  
21        view.addGestureRecognizer(dtaprecognizer)  
22        addChild(label) //add the label as a child node to the scene's  
23        node tree.  
24    }  
25  
26    @objc func doubletap(recognizer: UIGestureRecognizer) {  
27        if txtchange == false {  
28            label.text = "I ❤️ SpiteKit"  
29            txtchange = true  
30        }  
31        else {  
32            label.text = "Hello World !"  
33            txtchange = false  
34        }  
35    }  
36}  
37}
```

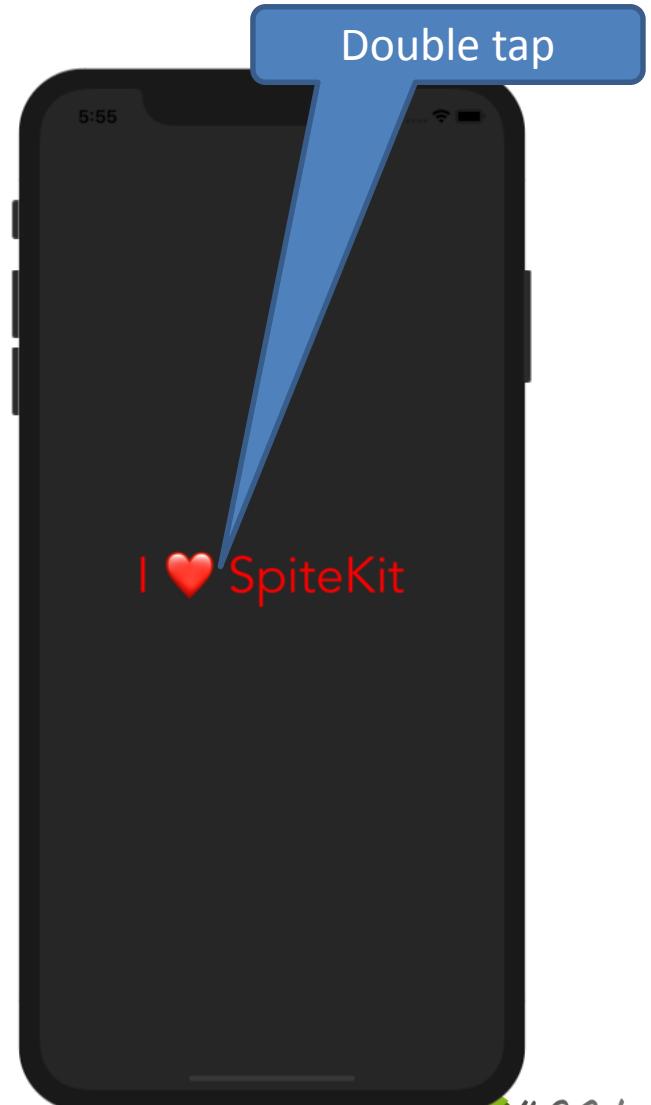
Add the double tap
recognizer

```
28    @objc func doubletap(recognizer: UIGestureRecognizer) {  
29        if txtchange == false {  
30            label.text = "I ❤️ SpiteKit"  
31            txtchange = true  
32        }  
33        else {  
34            label.text = "Hello World !"  
35            txtchange = false  
36        }  
37    }
```

Run it



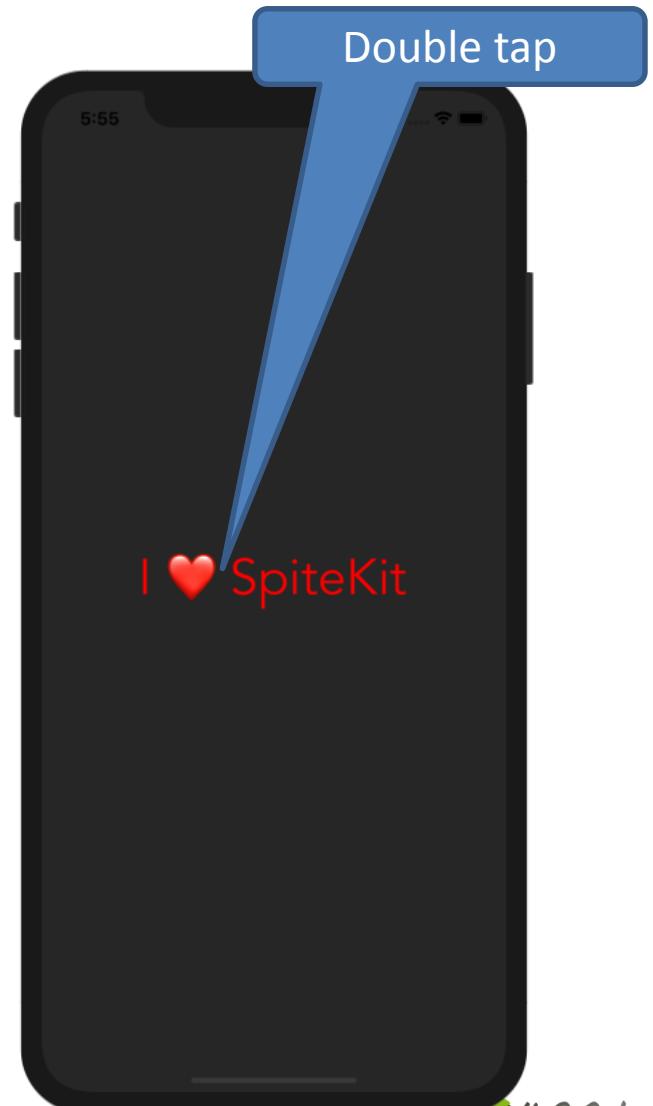
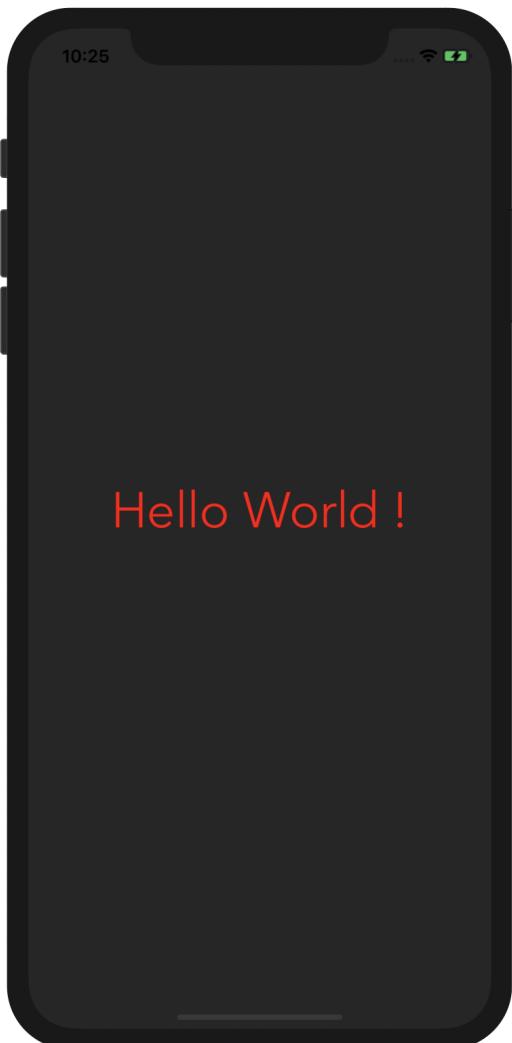
iPhone 11 — 13.3



iPhone 11 — 13.3

Run it

So, we can add
more gestures you
want...



Create A Game Using SpriteKit





Welcome to Xcode

Version 11.3.1 (11C504)



Get started with a playground

Explore new ideas quickly and easily.



Create a new Xcode project

Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.



Clone an existing project

Start working on something from a Git repository.



Show this window when Xcode launches



Firstsprite
~/Desktop



cpractice
~/Desktop



MyPlayground
~/Desktop



hellosprite
~/Desktop/2016 iOS/pr6 4h



hw1
~/Desktop



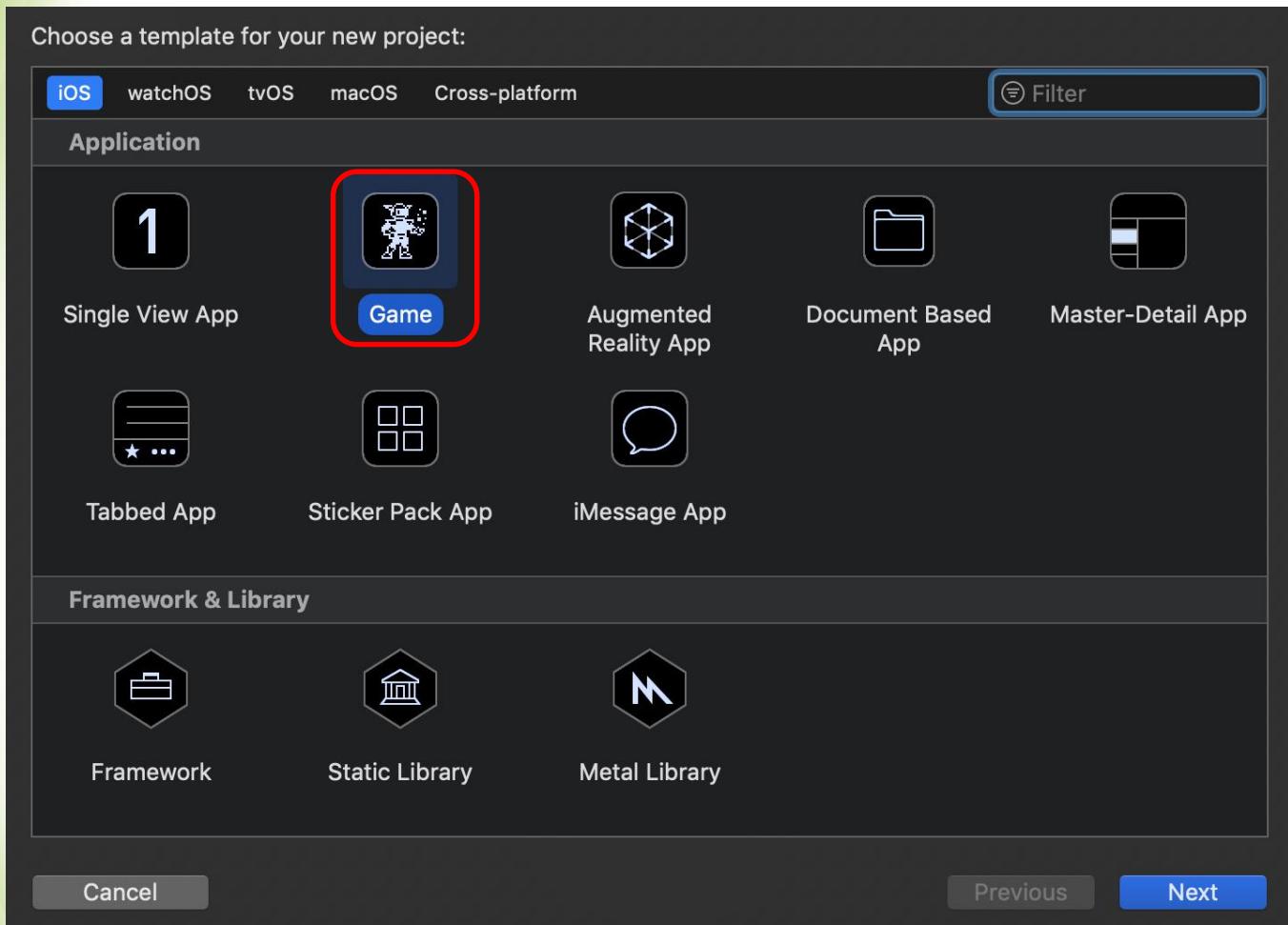
2-1
~/Desktop



Calculator
~/Desktop

Open another project...

- Create Sprite Kit Game



Choose options for your new project:

Product Name: Myspritekit

Team: Add account...

Organization Name: hpc

Organization Identifier: hpc

Bundle Identifier: hpc.Myspritekit

Language: Swift

Game Technology: SpriteKit

Integrate GameplayKit

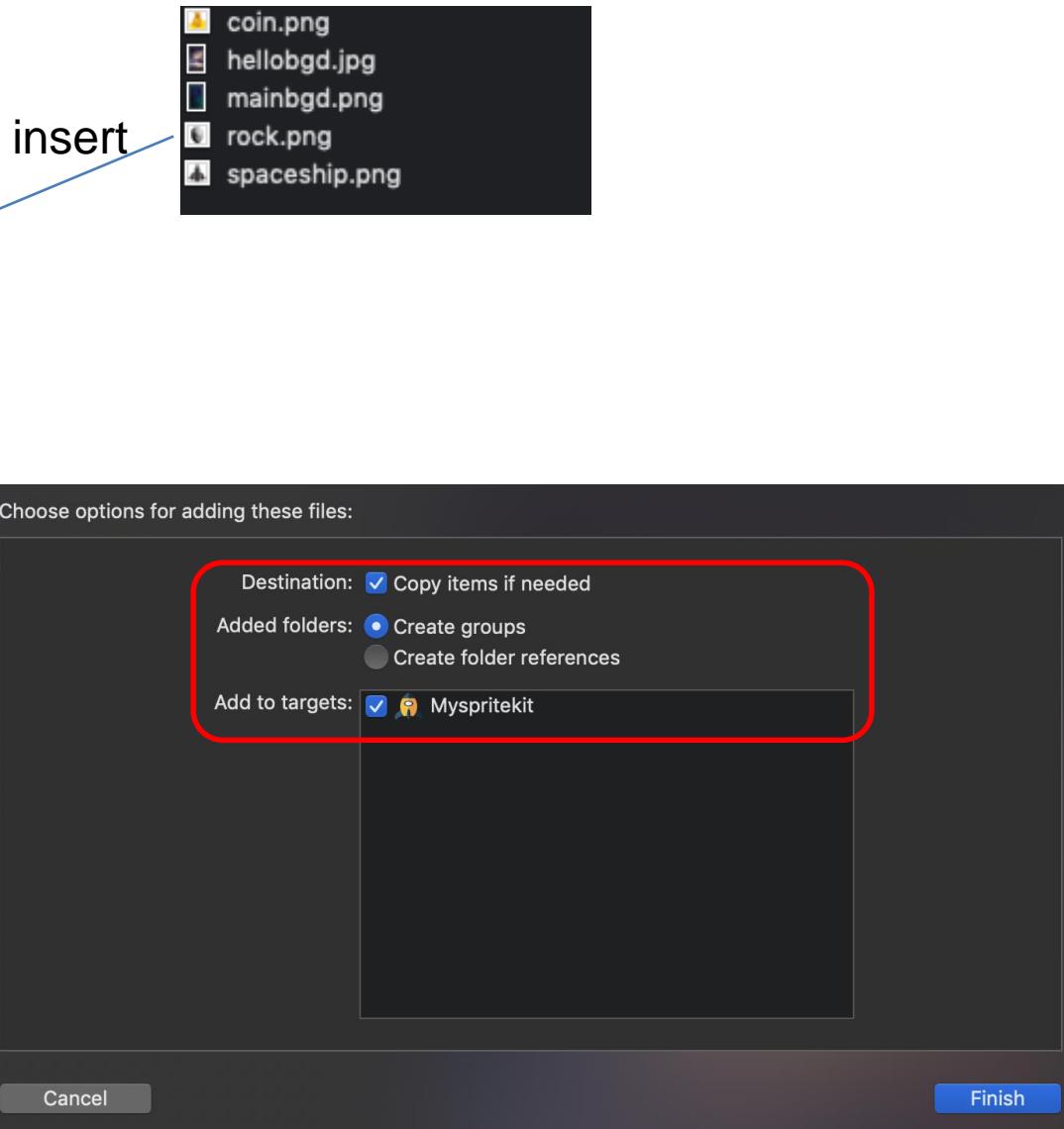
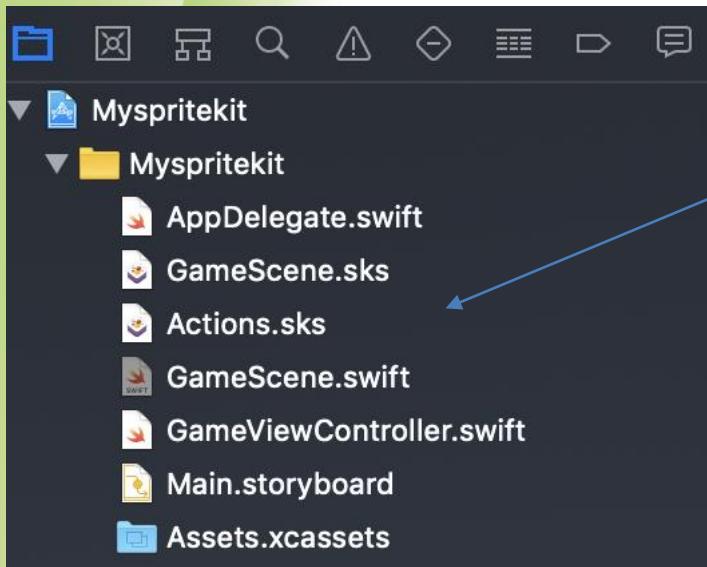
Include Unit Tests

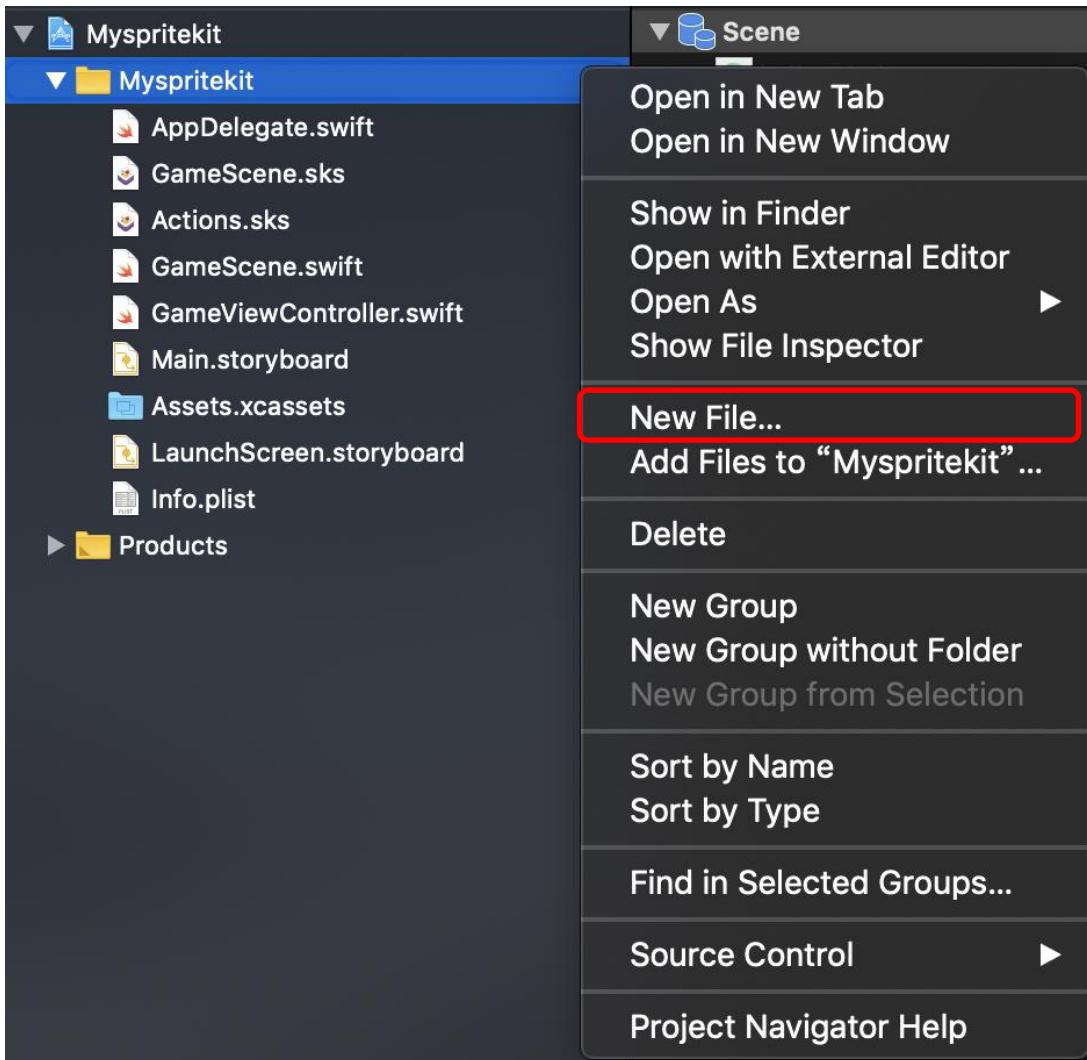
Include UI Tests

Cancel

Previous

Next





Choose a template for your new file:

iOS watchOS tvOS macOS Filter

Source

 Cocoa Touch Class	 UI Test Case Class	 Unit Test Case Class	 Swift File	 Objective-C File
 Header File	 IIG File	 C File	 C++ File	 Metal File

User Interface

 Storyboard	 SwiftUI View	 View	 Empty	 Launch Screen
---	---	--	--	--

Buttons

Cancel Previous Next

Choose options for your new file:

Class: HelloScene

Subclass of: SKScene

Also create XIB file

Language: Swift

Cancel

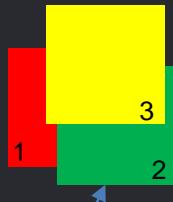
Previous

Next

GameViewController.swift

將UIView 強制轉為SKView

```
1 import UIKit
2 import SpriteKit
3
4 class GameViewController: UIViewController {
5
6     override func viewDidLoad() {
7         super.viewDidLoad()
8
9         if let view = self.view as! SKView? {
10             let scene = HelloScene(size: view.bounds.size)
11             scene.scaleMode = .aspectFill
12             view.presentScene(scene)
13             view.ignoresSiblingOrder = true
14             view.showsFPS = true
15             view.showsNodeCount = true
16         }
17     }
18 }
```



True: You need to set their zPosition .
False: Means that when multiple nodes share the same zPosition

HelloScene.swift

```
1 import UIKit
2 import SpriteKit
3
4 class HelloScene: SKScene {
5     override func didMove(to view: SKView) {
6         createScene()
7     }
8     func createScene(){
9         let bgd = SKSpriteNode(imageNamed: "hellobgd.jpg")
10        bgd.size.width = self.size.width
11        bgd.size.height = self.size.height
12        bgd.position = CGPoint(x:self.frame.midX, y:self.frame.midY)
13        bgd.zPosition = -1
14
15
16        let hellolabel = SKLabelNode(text: "Space 🚀 Adventure ")
17        hellolabel.name = "label"
18        hellolabel.position = CGPoint(x: self.frame.midX, y: self.frame.midY)
19        hellolabel.fontName = "Avenir-Oblique"
20        hellolabel.fontSize = 28
21
22        self.addChild(bgd)
23        self.addChild(hellolabel)
24
25 }
```

Add child node to
root(HelloScene)

HelloScene.swift

Override the function in class

When one or more touches occurred

```
27  override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {  
28      let labelNode = self.childNode(withName: "label")  
29      let moveup = SKAction.moveBy(x: 0, y: 200, duration: 1)  
30      let zoomin = SKAction.scale(to: 3.0, duration: 1)  
31      let pause = SKAction.wait(forDuration: 0.5)  
32      let zoomout = SKAction.scale(by: 0.5, duration: 0.25)  
33      let fadeaway = SKAction.fadeOut(withDuration: 0.25)  
34      let remove = SKAction.removeFromParent()  
35      let movesequece =  
36          SKAction.sequence([moveup,zoomin,pause,zoomout,pause,fadeaway,remove])  
37      labelNode?.run(movesequece)  
38      /* You can also group these actions , try it */  
39      /*  
40          let rotate = SKAction.rotate(byAngle: CGFloat.pi * 2, duration:3)  
41          let groupAction = SKAction.group([zoomout,rotate,moveup])  
42          labelNode?.run(groupAction)  
43      */
```

Run it and see what happens

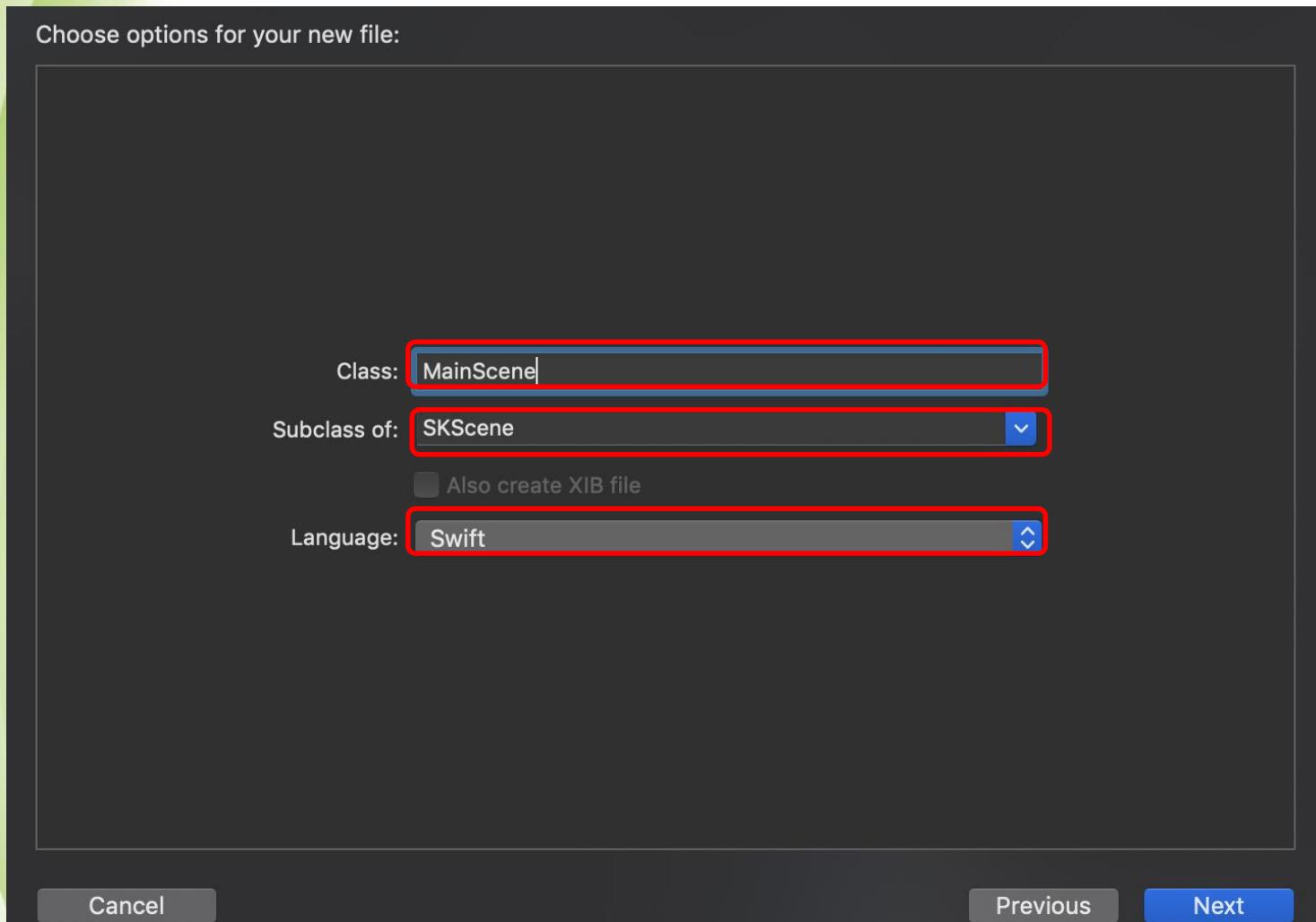


iPhone 11 Pro Max — 13.3



iPhone 11 Pro Max — 13.3

We need more Scene,
Create new SKScene



HelloScene.swift

In func touchesBegan

```
19     override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {  
20         let labelNode = self.childNode(withName: "label")  
21         let moveup = SKAction.moveBy(x: 0, y: 200, duration: 2)  
22         let zoomin = SKAction.scale(to: 3.0, duration: 2)  
23         let pause = SKAction.wait(forDuration: 0.5)  
24         let zoomout = SKAction.scale(by: 0.5, duration: 0.25)  
25         let fadeaway = SKAction.fadeOut(withDuration: 0.25)  
26         let remove = SKAction.removeFromParent()  
27         let movesequece = SKAction.sequence([moveup,zoomin,pause,zoomout,pause,fadeaway,remove])  
28         labelNode?.run(movesequece) delete  
29         /* You can also group the actions , try it */  
30         /*  
31             let rotate = SKAction.rotate(byAngle: CGFloat.pi * 2, duration:3)  
32             let groupAction = SKAction.group([zoomout,rotate,moveup])  
33             labelNode?.run(groupAction)  
34         */  
35     }
```

Insert the transition scene code

```
labelNode?.run(movesequece, completion: {  
    let mainScene = MainScene(size: self.size)  
    let doors = SKTransition.doorOpenVertical(withDuration: 0.5)  
    self.view?.presentScene(mainScene, transition: doors)  
})
```

When movesequence is complete

used to perform an animated transition to a new scene.

MainScene.swift

```
8     override func didMove(to view: SKView) {  
9         createScene()  
10    }
```

Need to define

Line 19

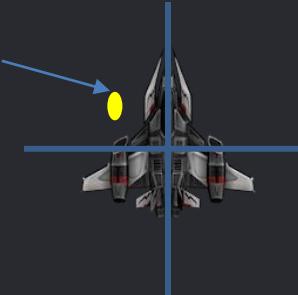
```
12    func createScene() {  
13        let mainbgd = SKSpriteNode(imageNamed: "mainbgd.png")  
14        mainbgd.size.width = self.size.width  
15        mainbgd.size.height = self.size.height  
16        mainbgd.position = CGPoint(x:self.frame.midX ,y:self.frame.midY)  
17        mainbgd.zPosition = -1  
18  
19        let spaceship = newSpaceship()  
20        spaceship.position = CGPoint(x: self.frame.midX, y: self.frame.midY-150)  
21  
22        self.addChild(mainbgd)  
23        self.addChild(spaceship)
```

MainScene.swift

```
53 func newSpaceship() ->SKSpriteNode {  
54     let ship = SKSpriteNode(imageNamed: "spaceship.png")  
55     ship.size = CGSize(width: 75, height: 75)  
56     ship.name = "ships"  
57  
58     let leftlight = newLight()  
59     leftlight.position = CGPoint(x:-20, y: 6)  
60     ship.addChild(leftlight)  
61  
62     let rightlight = newLight()  
63     rightlight.position = CGPoint(x:20, y: 6)  
64     ship.addChild(rightlight)  
65  
66     ship.physicsBody = SKPhysicsBody(circleOfRadius: ship.size.width / 2)  
67     ship.physicsBody?.usesPreciseCollisionDetection = true  
68     ship.physicsBody?.isDynamic = false  
69  
70     return ship  
71 }
```

Line 67

True : The simulation performs a more precise and more expensive calculation to detect these collisions. This property should be set to true on small, fast moving bodies.



A Boolean value that indicates whether the physics body is moved by the physics simulation.

creates a path of an rectangle.
CGRect : the rectangle to add

```
76 func newLight() -> SKShapeNode {
77     let light = SKShapeNode()
78     light.path = CGPath(rect: CGRect(x:-2,y:-4,width:4,height:8), transform: nil)
79     light.strokeColor = SKColor.white
80     light.fillColor = SKColor.yellow
81
82     let blink = SKAction.sequence([
83         SKAction.fadeOut(withDuration: 0.25),
84         SKAction.fadeIn(withDuration: 0.25)
85     ])
86     let blinkForever = SKAction.repeatForever(blink)
87     light.run(blinkForever)
88     return light
89 }
```



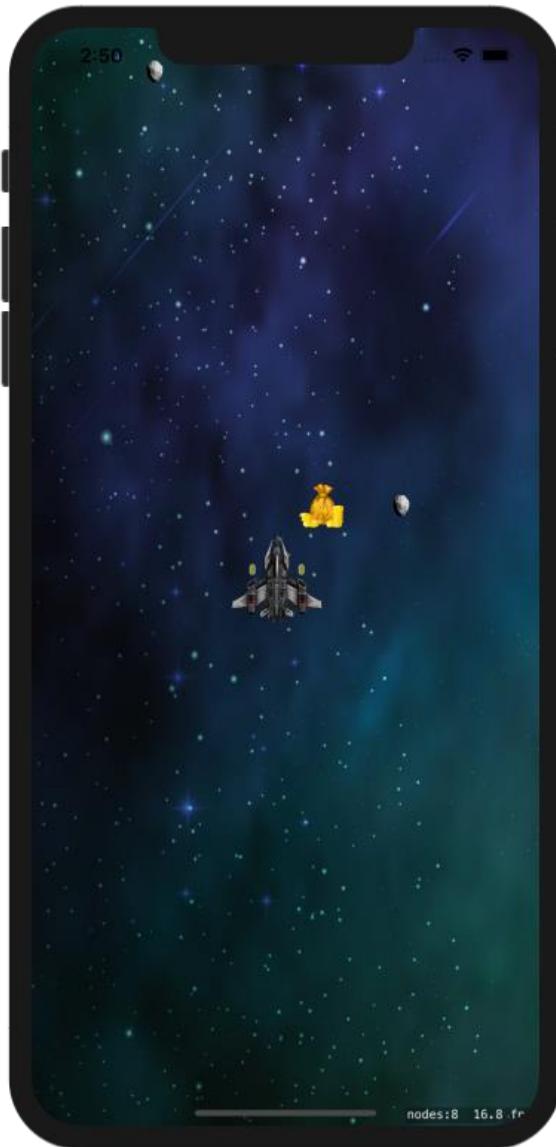
Run it
We have a cool spaceship with
cool lights now !



iPhone 11 Pro Max — 13.3

Run it
We have a cool spaceship with
cool lights now !

Let's add some nodes



iPhone 11 Pro Max — 13.3

Do nothing for 3s and
then remove from parent

MainScene.swift

```
62 @objc func newRock()  {
63     let rock = SKSpriteNode(imageNamed: "rock.png")
64     rock.size = CGSize(width: 40 , height: 40)
65     let remove = SKAction.sequence([SKAction.wait(forDuration: 3),SKAction.removeFromParent()])
66     let w = self.size.width
67     let h = self.size.height
68     let x = CGFloat(Float(arc4random()).truncatingRemainder(dividingBy: w))
69     rock.position = CGPoint(x: x , y: h)
70     rock.name = "rocks"
71     rock.physicsBody = SKPhysicsBody(circleOfRadius: 4)
72     rock.physicsBody?.usesPreciseCollisionDetection = true
73     rock.run(remove)
74     self.addChild(rock)
75 }
```

MainScene.swift

```
77 @objc func newCoin()  {
78     let coin = SKSpriteNode(imageNamed: "coin.png")
79     coin.size = CGSize(width: 30, height: 30)
80     let remove = SKAction.sequence([SKAction.wait(forDuration: 3),SKAction.removeFromParent()])
81     let w = self.size.width
82     let h = self.size.height
83     let x = CGFloat(Float(arc4random()).truncatingRemainder(dividingBy: w))
84     coin.position = CGPoint(x: x , y: h)
85     coin.name = "coins"
86     coin.physicsBody = SKPhysicsBody(circleOfRadius: 4)
87     coin.physicsBody?.usesPreciseCollisionDetection = true
88     coin.run(remove)
89     self.addChild(coin)
90 }
```

MainScene.swift

In function createScene()

```
12 func createScene() {
13     let mainbgd = SKSpriteNode(imageNamed: "mainbgd.png")
14     mainbgd.size.width = self.size.width
15     mainbgd.size.height = self.size.height
16     mainbgd.position = CGPoint(x:frame.size.width / 2, y: frame.size.height / 2)
17     mainbgd.zPosition = -1
18
19     let spaceship = newSpaceship()
20     spaceship.position = CGPoint(x: self.frame.midX, y: self.frame.midY-150)
21
22     self.addChild(mainbgd)
23     self.addChild(spaceship)
24
25     Timer.scheduledTimer(timeInterval: 0.5, target: self, selector: #selector (newRock), userInfo: nil,
26                           repeats: true)
27     Timer.scheduledTimer(timeInterval: 2, target: self, selector: #selector(newCoin), userInfo: nil,
28                           repeats: true)
29 }
```

Insert Timer

MainScene.swift

```
1 import UIKit
2 import SpriteKit
3
4 class MainScene: SKScene {
5
6     override func didMove(to view: SKView) {
7         createScene()
8         let panrecognizer = UIPanGestureRecognizer(target: self, action: #selector(handpan))
9         view.addGestureRecognizer(panrecognizer)
10    }
```

declare a recognizer

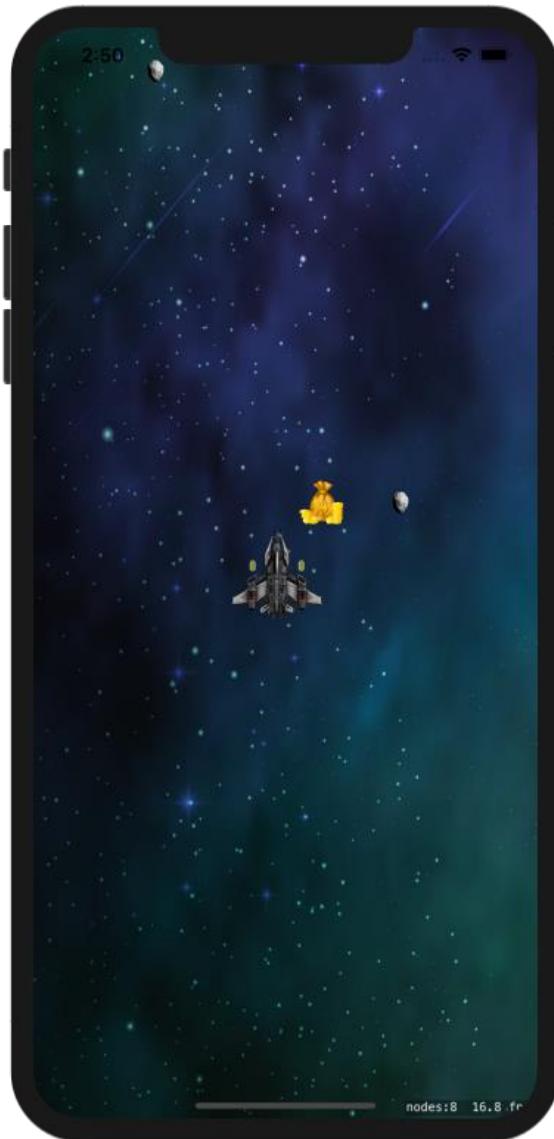
call which function
when happen

Target: MainScene

MainScene.swift

```
123 @objc func handpan(recognizer: UIPanGestureRecognizer) {
124     let viewLocation = recognizer.location(in: view)
125     let sceneLocation = convertPoint(fromView: viewLocation)
126     let moveAction = SKAction.moveTo(x: sceneLocation.x, duration: 0.1)
127     self.addChild(node: "ships")!.run(moveAction)
128 }
```

Run it



iPhone 11 Pro Max — 13.3

An object that implements the SKPhysicsContactDelegate protocol can respond when two physics bodies with overlapping contactTestBitMask values are in contact with each other in a physics world.

```
4 class MainScene: SKScene, SKPhysicsContactDelegate {
```

```
8     override func didMove(to view: SKView) {  
9         self.physicsWorld.contactDelegate = self
```

Func newSpaceship()

```
ship.physicsBody?.categoryBitMask = 0x1 << 1  
ship.physicsBody?.contactTestBitMask = 0x1 << 2
```

Type: UInt32 , Defines the collision categories to which a physics body belongs.

Determines which categories this physics body makes contact with

We don't know which one is spaceship, bodyA ? or bodyB ?



Determine which nodes collided

```
130 func didBegin(_ contact: SKPhysicsContact) {  
131     var firstBody = SKPhysicsBody()  
132     var secondBody = SKPhysicsBody()  
133     if contact.bodyA.node?.name == "ships"{  
134         firstBody = contact.bodyA  
135         secondBody = contact.bodyB  
136     } else {  
137         firstBody = contact.bodyB  
138         secondBody = contact.bodyA  
139     }  
140     if (firstBody.node?.name == "ships" && secondBody.node?.name == "rocks") {  
141         print("You lose !\n")  
142     }  
143     else if (firstBody.node?.name == "ships" && secondBody.node?.name == "coins"){  
144         contact.bodyB.node?.removeFromParent()  
145         print("Get point 100 !\n")  
146     }  
147     /* if (secondBody.node?.name == "rocks"){  
148         print("You lose !\n")  
149     }  
150     else{  
151         contact.bodyB.node?.removeFromParent()  
152         print("Get point 100 !\n")  
153     }  
154 */  
155 }
```

Get point 100 !

You lose !

Get point 100 !

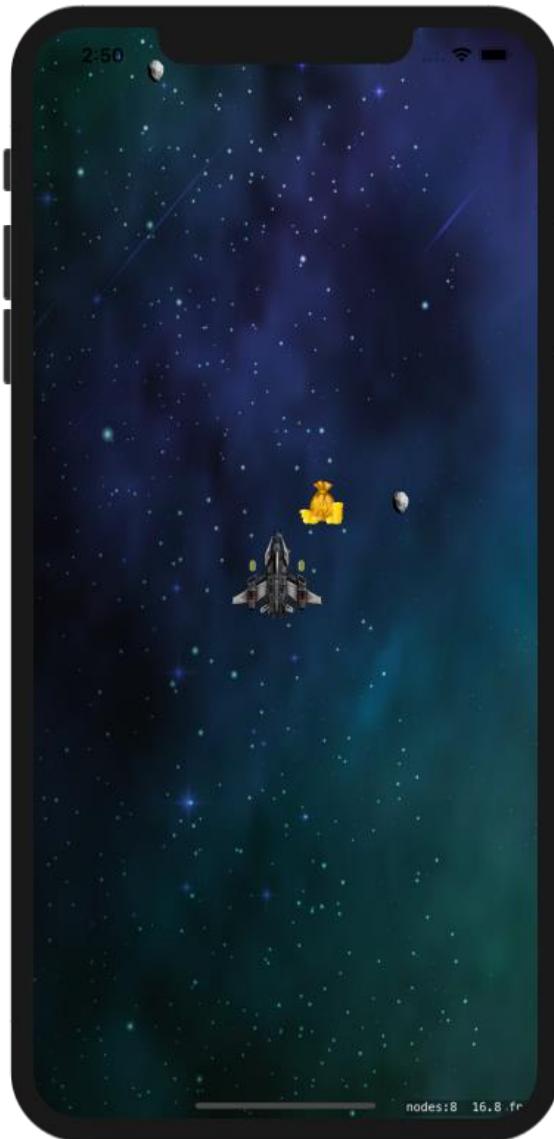
Get point 100 !

Get point 100 !

Get point 100 !

All Output ▾

Filter



iPhone 11 Pro Max — 13.3