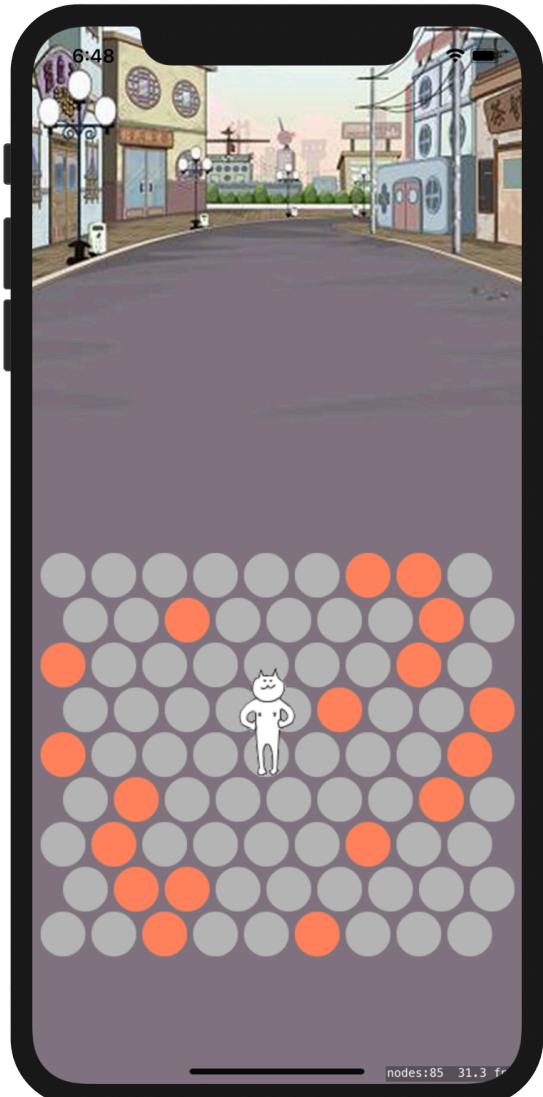


Class 05-2



Create A Game Using SpriteKit

Cat



iPhone 11 — 13.3

Create the game project

The screenshot shows the Xcode welcome screen. At the top, there's a large icon of a hammer and a pencil on a blueprint. Below it, the text "Welcome to Xcode" and "Version 11.3.1 (11C504)" are displayed. On the left, there are three main options: "Get started with a playground", "Create a new Xcode project", and "Clone an existing project". The "Create a new Xcode project" option is highlighted with a red rectangle. At the bottom left, there's a checkbox for "Show this window when Xcode launches". On the right side, a sidebar lists several recent projects with their names and paths:

- MySpritekit
~/Desktop/MySpritekit
- Myspritekit
~/Desktop/xcode
- Angelica Fighti
~/Desktop/Legend-Wings-master
- Firstsprite
~/Desktop/xcode
- Myspritekit02
~/Desktop/xcode
- Cat
...16 iOS/範例程式/CHT/第8章 程式碼/8.7 零路
- hash-table
~/Desktop/xcode
- 4-1-2
~/Desktop/xcode
- MyPlayground
~/Desktop/xcode
- 4-1
~/Desktop/xcode

At the bottom right, there's a link "Open another project...".

Choose options for your new project:

Product Name: **spritekit_cat**

Team: [Add account...](#)

Organization Name: **hpc**

Organization Identifier: **hpc**

Bundle Identifier: **hpc.spritekit-cat**

Language: **Swift**

Game Technology: **SpriteKit**

Integrate GameplayKit

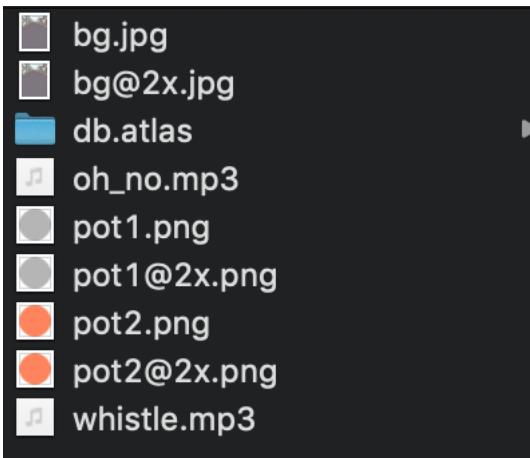
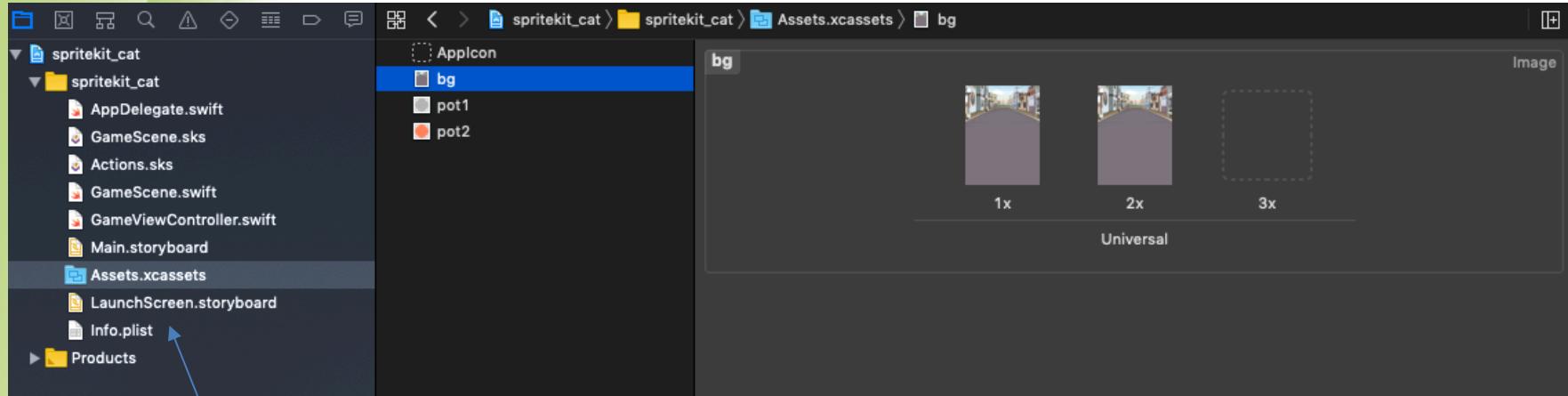
Include Unit Tests

Include UI Tests

[Cancel](#)

[Previous](#)

[Next](#)



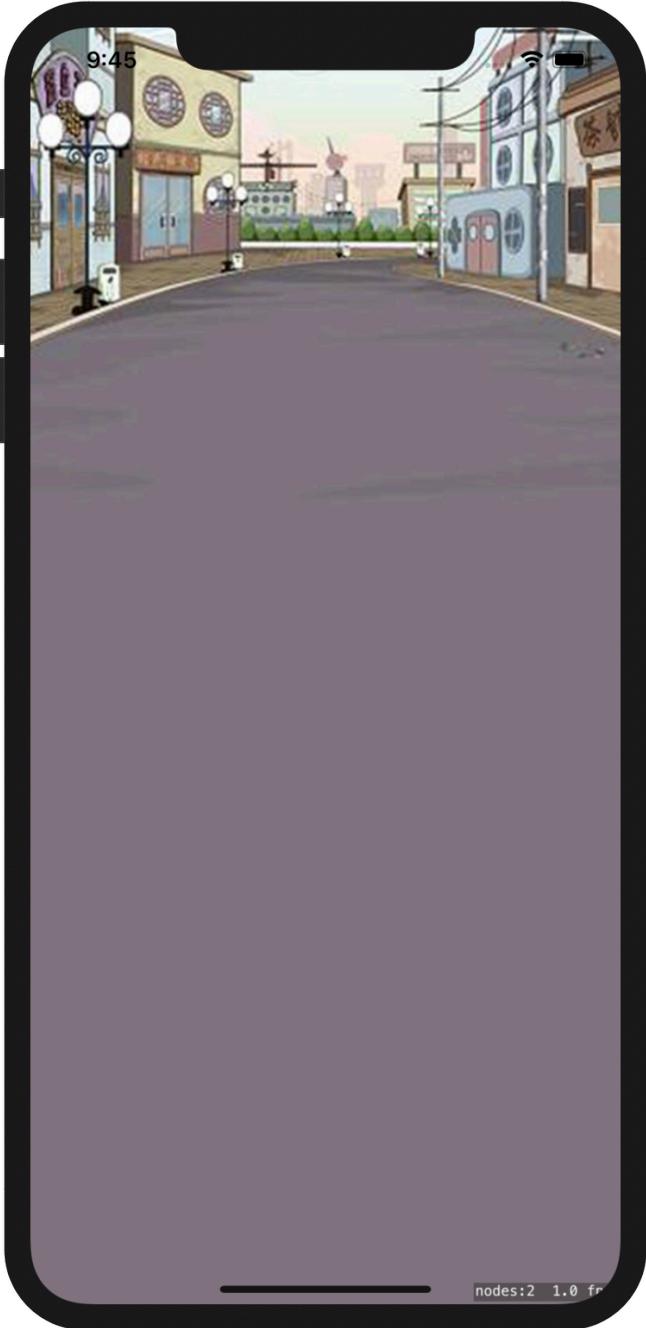
將db.atlas與mp3檔直接拖入專案，
其餘對應的放入Assets

GameViewController.swift

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

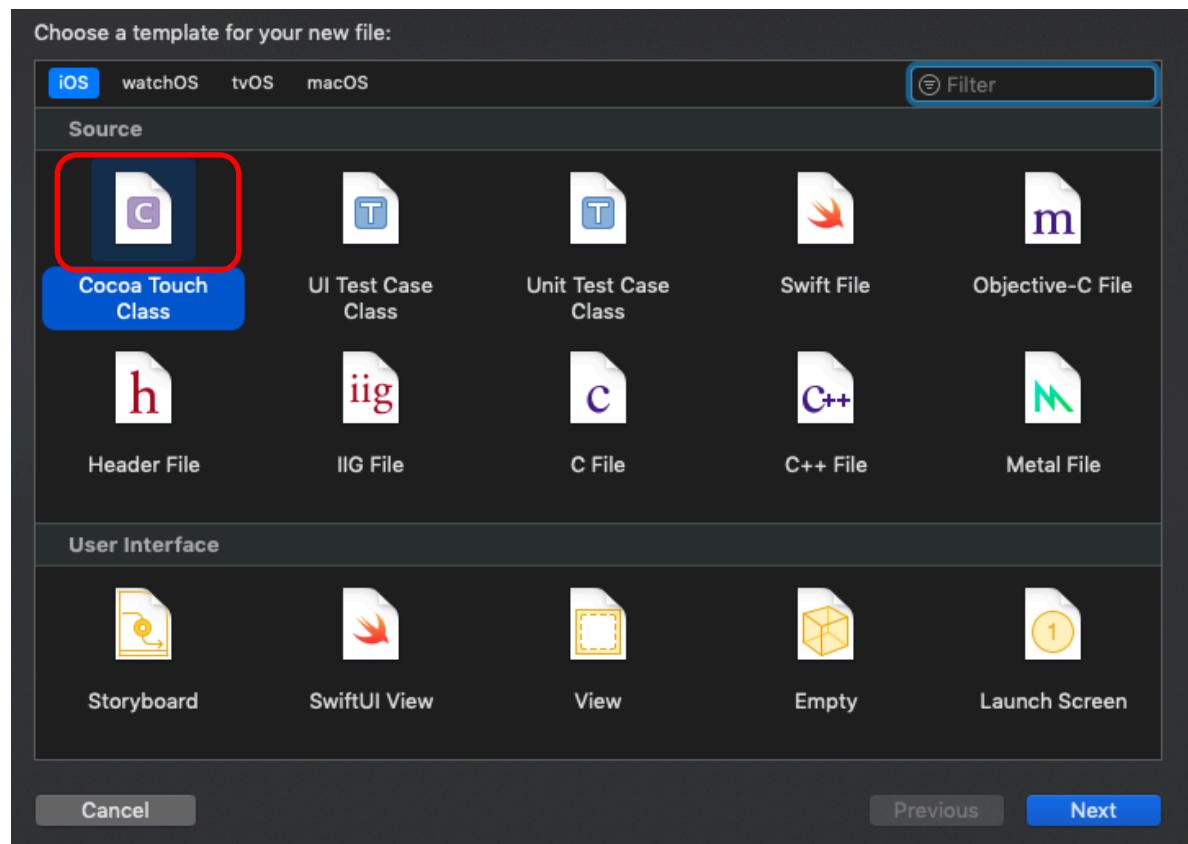
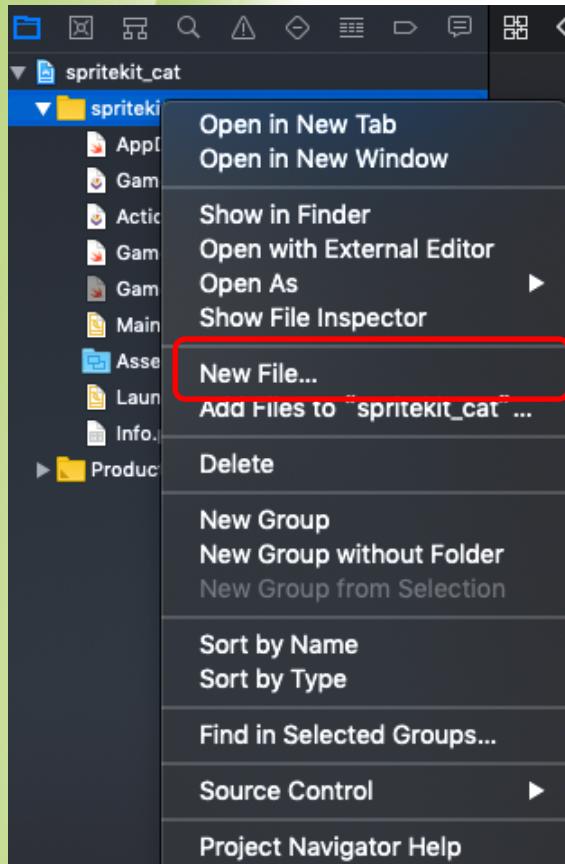
GameScene.swift

```
1 import SpriteKit  
2  
3 class GameScene: SKScene {  
4  
5     override func didMove(to view: SKView) {  
6         let background = SKSpriteNode(imageNamed: "bg")  
7         background.position = CGPoint(x:self.frame.midX,y:self.frame.midY)  
8         addChild(background)  
9     }  
10    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {  
11    }  
12 }  
13 }
```

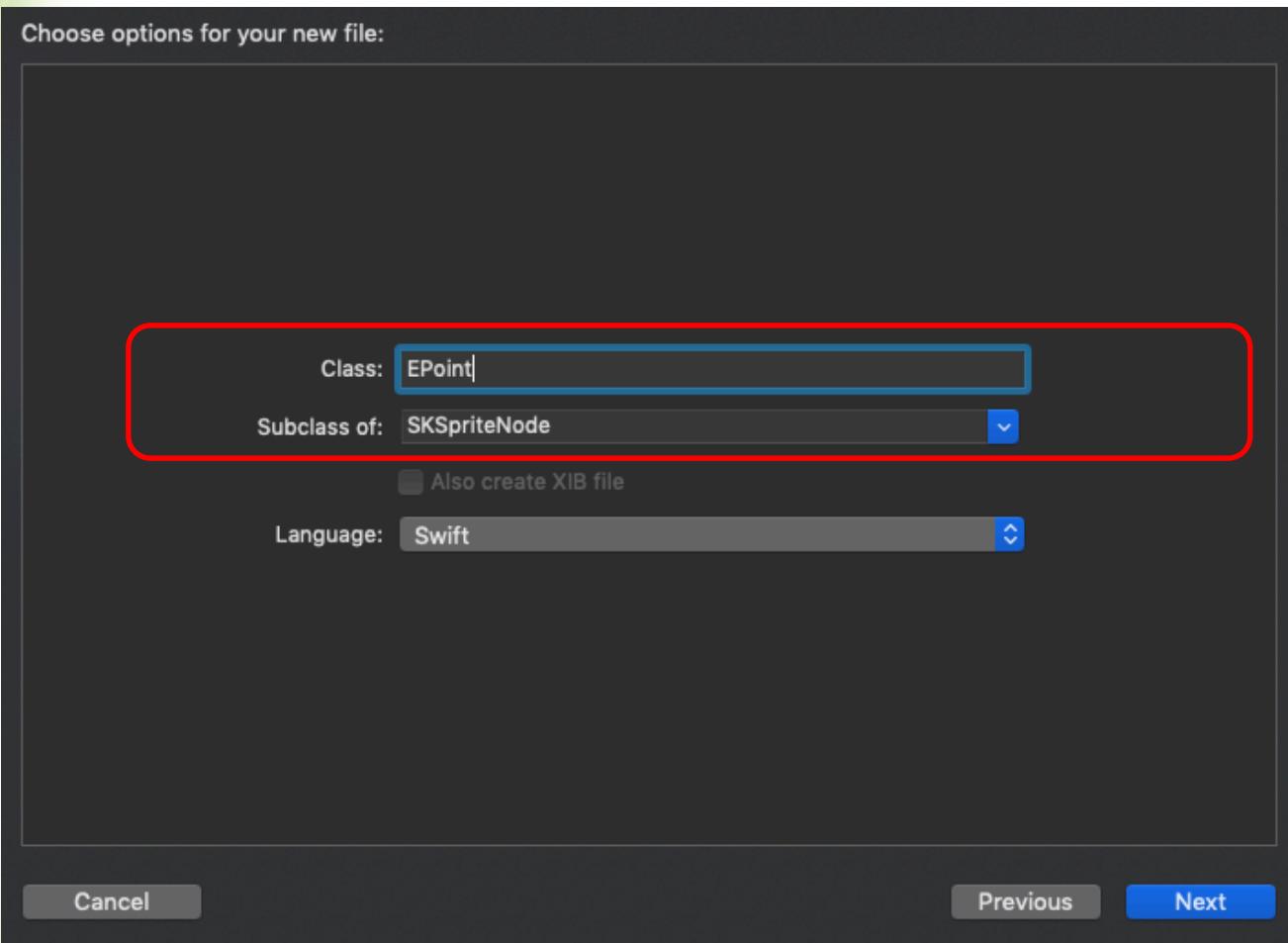


iPhone 11 Pro Max — 13.3

- New File



- New File

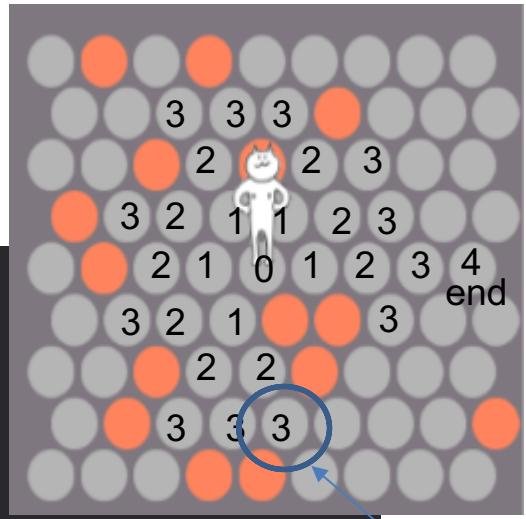


Class EPoint

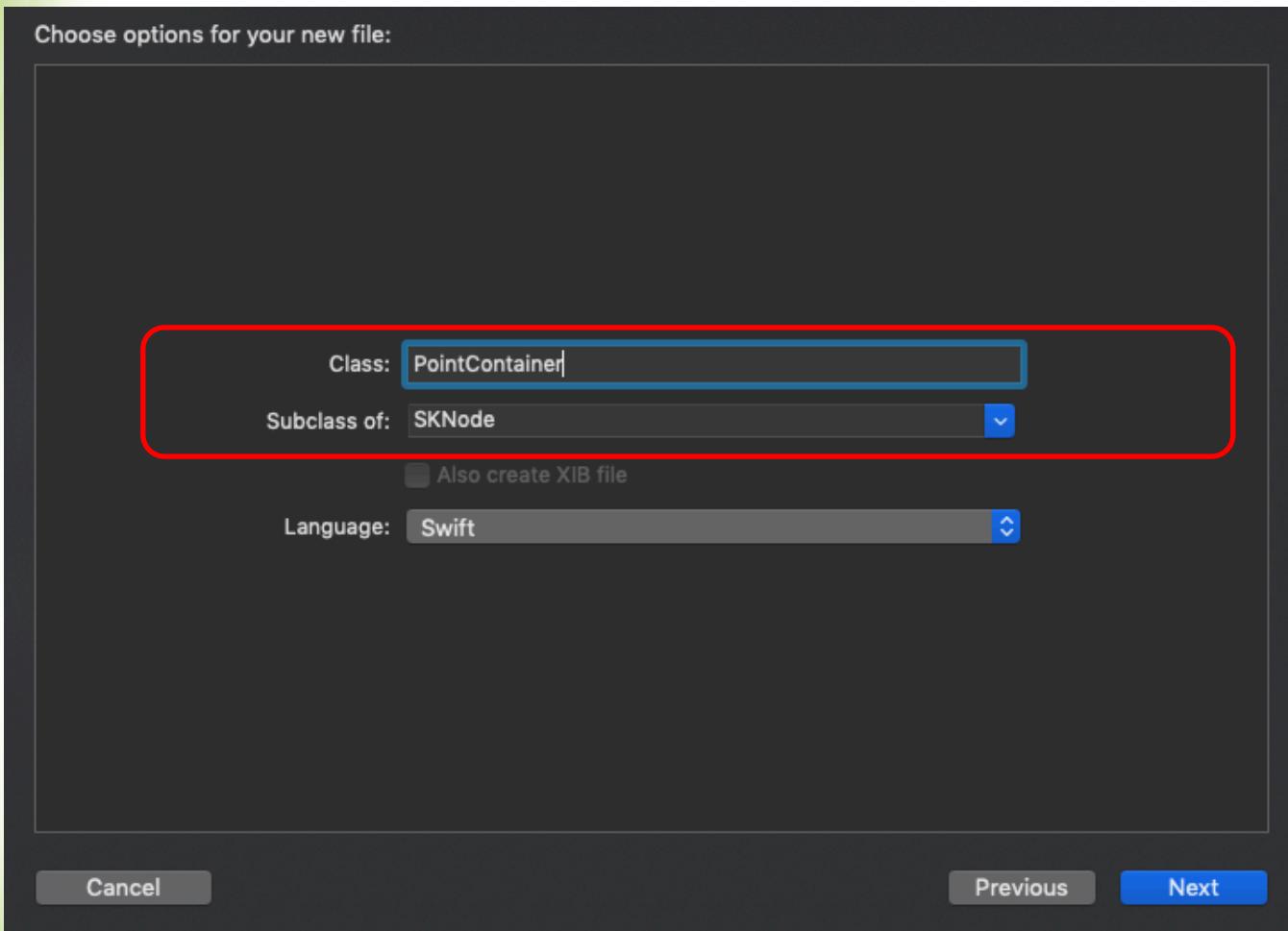
```
1 import SpriteKit  
2  
3 enum pointtype : Int {  
4     case gray = 0  
5     case red = 1  
6 }  
7 class EPoint: SKSpriteNode {  
8     var prePointIndex = -1  
9     var aroundPoint = [Int]()  
10    var step = 99  
11    var index = 0  
12    var type = pointtype.gray  
13    var isEdge = false  
14 }
```

此點是否為邊緣點

用陣列來儲存周圍的點之index



- New File



SKTexture 是一個 object 用於渲染
SKShapeNode 與 SKSpriteNode。
Texture 通常是由專案裡的圖集創建，
當被載入到記憶體後，它將保存到你
刪除它為止

PointContainer.swift

```
1 import SpriteKit
2
3 class PointContainer: SKNode {
4     let textPoint1 = SKTexture(imageNamed: "pot1")
5     let textPoint2 = SKTexture(imageNamed: "pot2")
6     var arrPoint = [EPoint]()
7     let startIndex = 40
8     var currIndex = 40
9     var isFind = false
10
11    var arrNext = [Int]()
12 }
```

此陣列儲存所有點，共81個

記錄外圈點
用於廣度優先搜尋路徑

代表是否找到出口(邊緣點)

此函式來初始化共81個點

```
15     func onInit(){
16         for i in 0...80 {
17             let point = EPoint(texture: textPoint1)
18             let row = Int(i/9)
19             let col = i%9
20             var gap = 0
21             //奇偶行判斷,0為奇數行,1為偶數行
22             if Int(row % 2) == 1 {
23                 gap = 19
24             }else {
25             }
26             //取得紋理寬度,用來計算位置
27             let width = Int(textPoint1.size().width)
28             let x = col * (width+5) - (9*width)/2 + gap
29             let y = row * width - (9*width)/2
30             point.position = CGPointMake(x:CGFloat(x),y:CGFloat(y))
31             //判斷是否為邊緣點
32             if row == 0 || row == 8 || col == 0 || col == 8 {
33                 point.isEdge = true
34             }
35             point.index = i
36             point.zPosition = 10
37             addChild(point)
38             arrPoint.append(point)
39         }
40     }
```

偶數行需要右移位置

將此點加入到PointContainer
與點陣列中

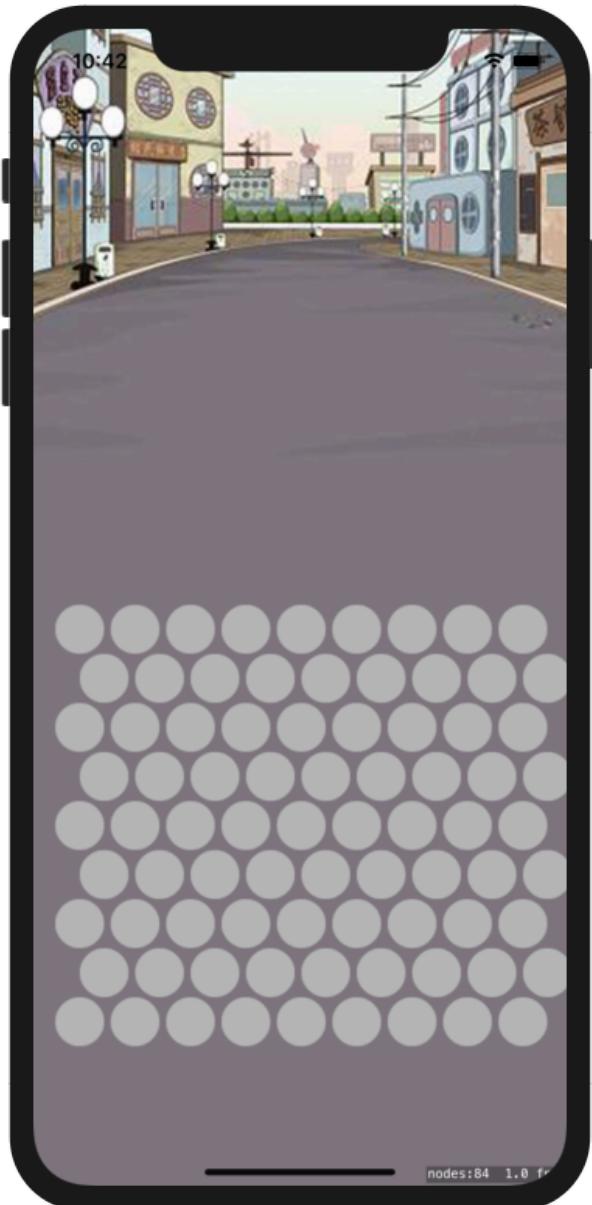
GameScene.swift

宣告一個PointContainer
物件

設定在Scene中的位置，
並呼叫onInit函式來初始化
所需的Nodes

```
1 import SpriteKit
2
3 class GameScene: SKScene {
4     let pcontainer = PointContainer()
5
6     override func didMove(to view: SKView) {
7         let background = SKSpriteNode(imageNamed: "bg")
8         background.position = CGPoint(x:self.frame.midX,y:self.frame.midY)
9         addChild(background)
10
11         pcontainer.position = CGPoint(x: self.frame.midX, y: self.frame.midY-150)
12         addChild(pcontainer)
13         pcontainer.onInit()
14     }
15     override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
16
17     }
18 }
```

Run it



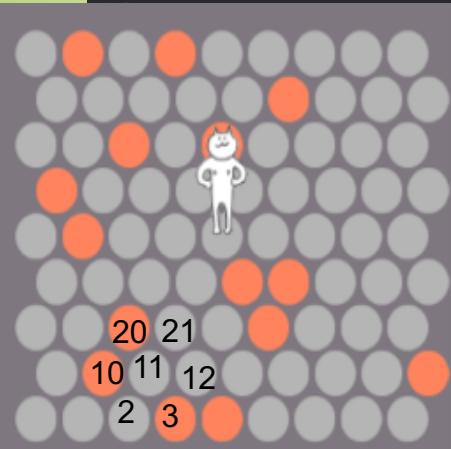
iPhone 11 — 13.3

PointContainer.swift

此函式將arrPoint陣列裡的每個點之
周圍點都記錄下來

觀察其規律，並判斷奇偶
行

```
43 func onData(){
44     for point in arrPoint {
45         let row = Int( point.index / 9 )
46         if Int(row % 2) == 1 {
47             if point.index - 1 >= 0 && point.index - 1 <= 80 {
48                 point.aroundPoint.append(point.index - 1)
49             }
50             if point.index + 9 >= 0 && point.index + 9 <= 80 {
51                 point.aroundPoint.append(point.index + 9)
52             }
53             if point.index + 10 >= 0 && point.index + 10 <= 80 {
54                 point.aroundPoint.append(point.index + 10)
55             }
56             if point.index + 1 >= 0 && point.index + 1 <= 80 {
57                 point.aroundPoint.append(point.index + 1)
58             }
59             if point.index - 8 >= 0 && point.index - 8 <= 80 {
60                 point.aroundPoint.append(point.index - 8)
61             }
62             if point.index - 9 >= 0 && point.index - 9 <= 80 {
63                 point.aroundPoint.append(point.index - 9)
64             }
65 }
```



PointContainer.swift in Function onData()

```
66         else {
67             if point.index - 1 >= 0 && point.index - 1 <= 80 {
68                 point.aroundPoint.append(point.index - 1)
69             }
70             if point.index + 8 >= 0 && point.index + 8 <= 80 {
71                 point.aroundPoint.append(point.index + 8)
72             }
73             if point.index + 9 >= 0 && point.index + 9 <= 80 {
74                 point.aroundPoint.append(point.index + 9)
75             }
76             if point.index + 1 >= 0 && point.index + 1 <= 80 {
77                 point.aroundPoint.append(point.index + 1)
78             }
79             if point.index - 9 >= 0 && point.index - 9 <= 80 {
80                 point.aroundPoint.append(point.index - 9)
81             }
82             if point.index - 10 >= 0 && point.index - 10 <= 80 {
83                 point.aroundPoint.append(point.index - 10)
84             }
85         }
86     }
87 }
```

PointContainer.swift

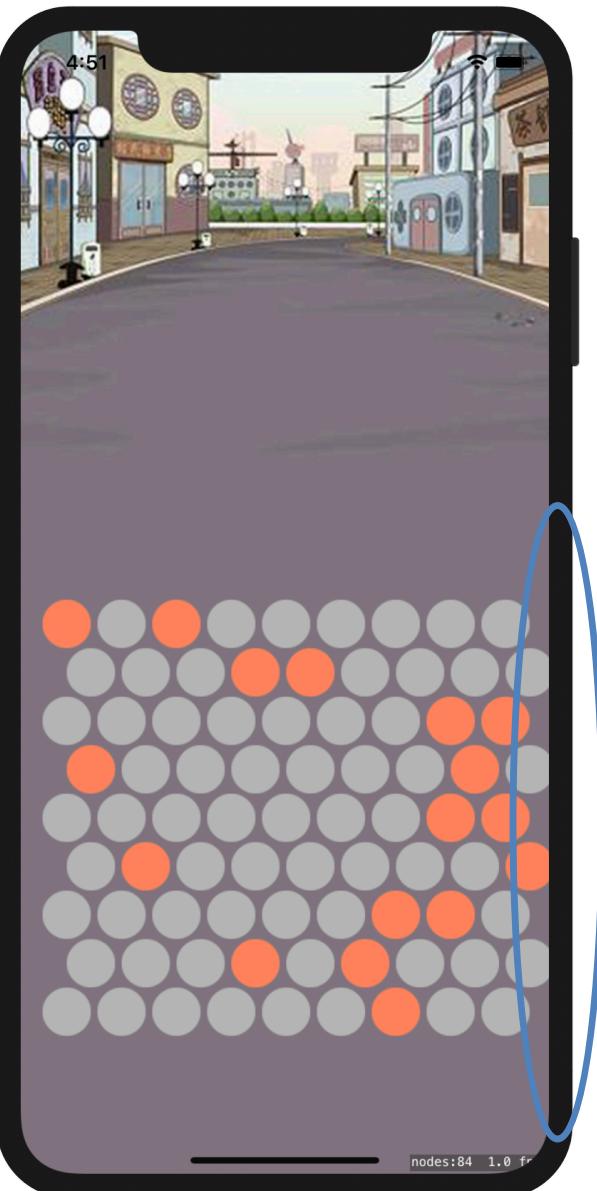
```
95     //改變圓點類型
96     func onSetRed(_ index:Int){
97         arrPoint[index].type = pointtype.red
98         arrPoint[index].texture = textPoint2
99     }
100    //隨機生成紅色點
101    func onCreateRed(){
102        for i in 0...8 {
103            let r1 = Int(arc4random() % 9) + i * 9
104            let r2 = Int(arc4random() % 9) + i * 9
105            if r1 != startIndex {
106                onSetRed(r1)
107            }
108            if r2 != startIndex {
109                onSetRed(r2)
110            }
111        }
112    }
```

PointContainer.swift In func onInit()

```
29         // 判斷是否為邊緣點
30         if row == 0 || row == 8 || col == 0 || col == 8 {
31             point.isEdge = true
32         }
33         point.index = i
34         point.zPosition = 10
35         addChild(point)
36         arrPoint.append(point)
37     }
38     onData()
39     onCreateRed()
40 }
```

初始化時要將每個點的周圍點記錄到陣列裡，
並隨機產生紅點

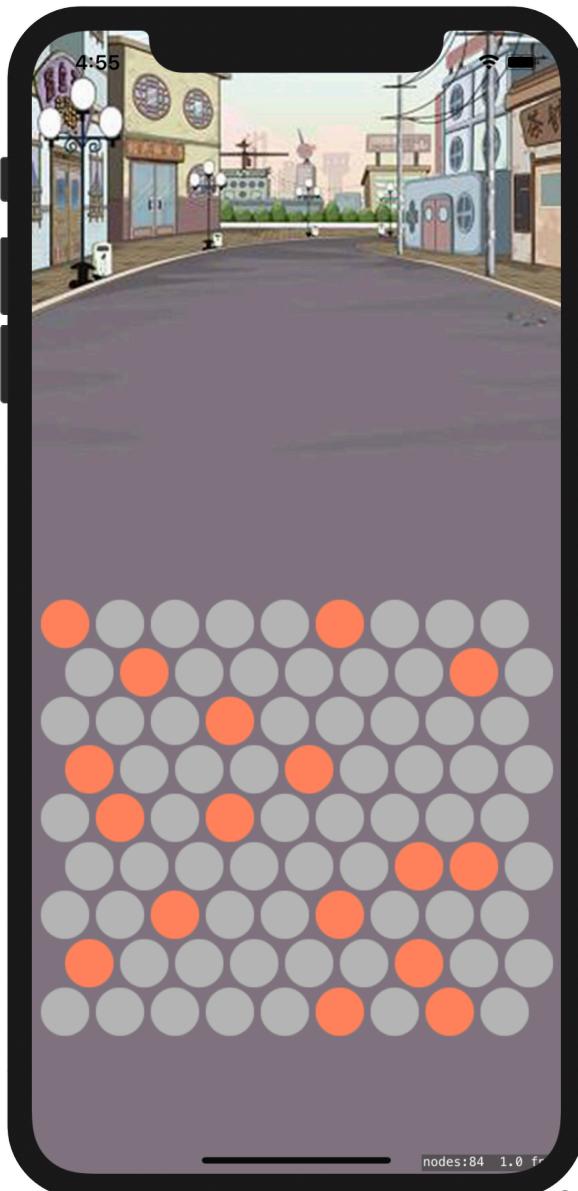
Run it



iPhone 11 — 13.3

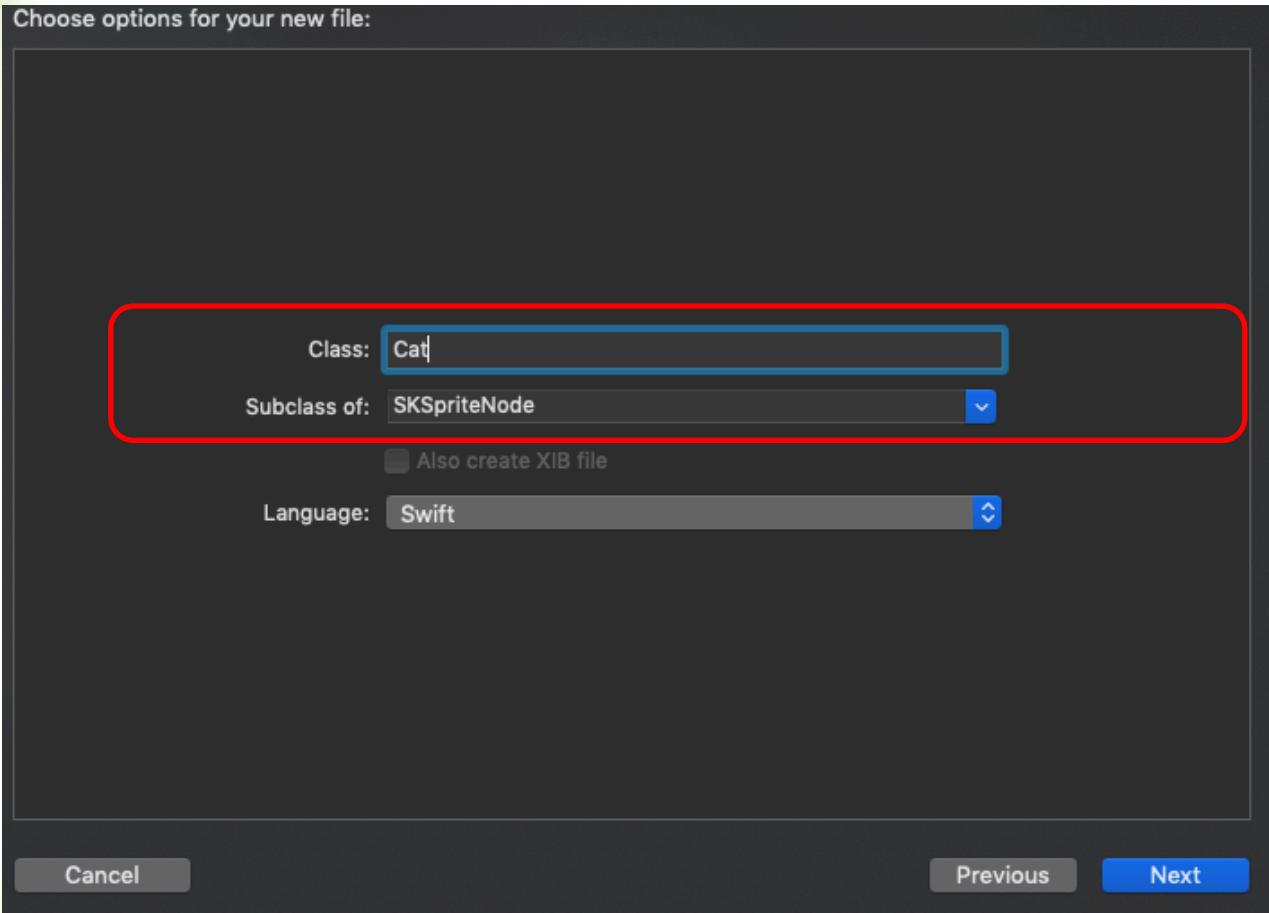
GameScene.swift

```
11     pcontainer.position = CGPoint(x: self.frame.midX-10, y:  
12                                     self.frame.midY-150)  
13     addChild(pcontainer)  
14     pcontainer.onInit()  
    }
```



iPhone 11 — 13.3

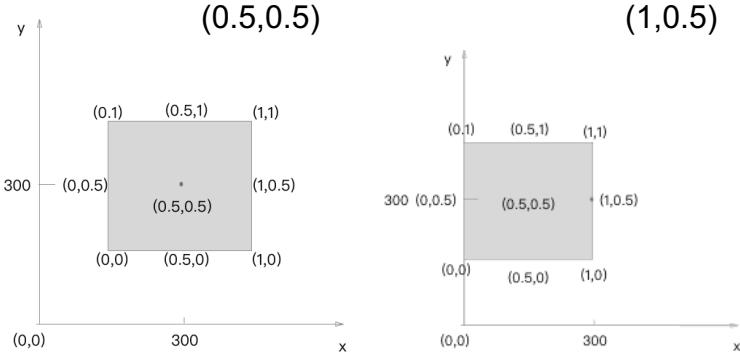
- New File



Class Cat

anchorPoint為中心點位置
更改此屬性令Cat站在點上

名為"db.atlas"的紋理集



```
1 import SpriteKit
2
3 class Cat: SKSpriteNode {
4     let textureAtlas = SKTextureAtlas(named: "db.atlas")
5     var textureFrames = [SKTexture]()
6
7     override init(texture: SKTexture?, color:SKColor,size:CGSize) {
8         let texture = SKTexture(imageNamed: "db_02.png")
9         //執行父類別的構造方法
10        super.init(texture:texture,color:SKColor.clear,size:texture.size())
11        //從紋理集中提取紋理加入陣列
12        var tempName:String
13        for i in 1...textureAtlas.textureNames.count {
14            tempName = String(format:"db_%2d",i)
15            let dbTexture = textureAtlas.textureNamed(tempName)
16            textureFrames.append(dbTexture)
17        }
18        anchorPoint = CGPoint(x: 0.5, y: 0.2)
19        showAtlas()
20    }
21    required init?(coder aDecoder: NSCoder) {
22        fatalError("init(coder:) has not been implemented ")
23    }
24    func showAtlas(){
25        self.run(SKAction.repeatForever(SKAction.animate(with: textureFrames, timePerFrame: 0.2)))
26    }
27 }
```

要用此陣列來處理動畫

將紋理集都加入到陣列裡

此函式來回傳position,
Cat會需要用此來更改位置

PointContainer.swift

```
196     func onGetPosition(_ index:Int) -> CGPoint{  
197         return CGPoint(x: arrPoint[index].position.x, y: arrPoint[index].position.y)  
198     }
```

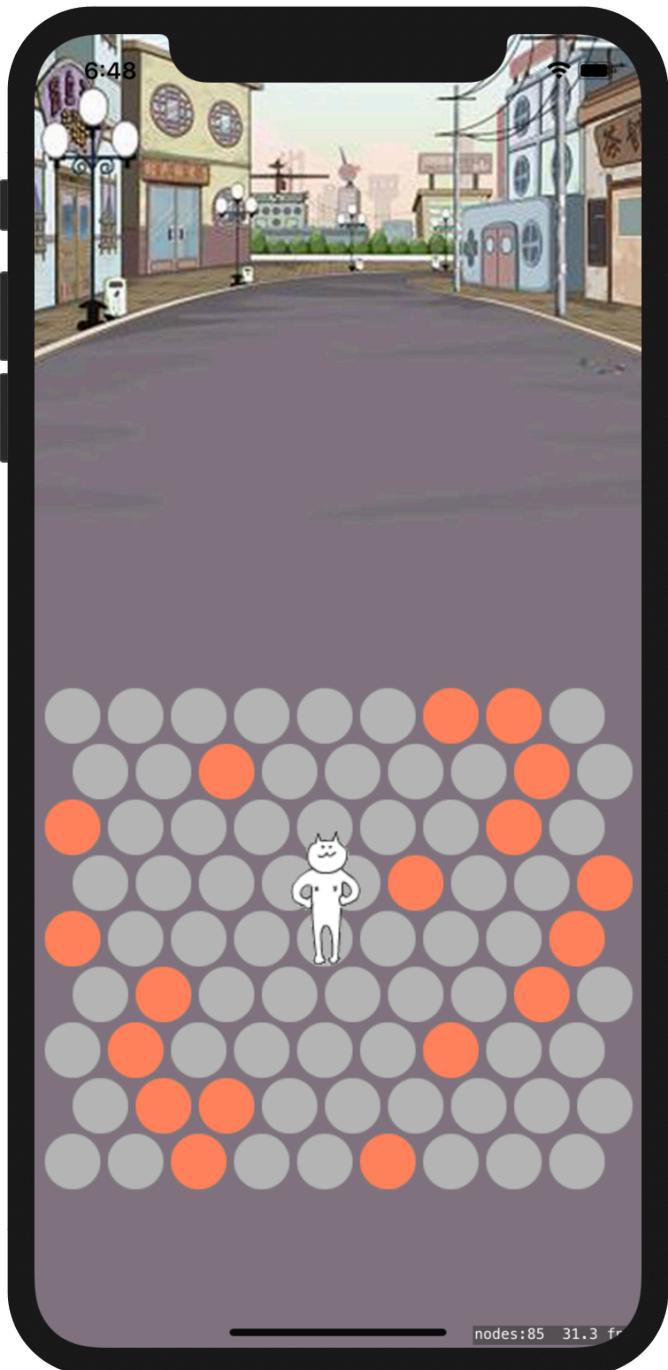
```
4  class PointContainer: SKNode {  
5      let cat = Cat()  
6      let textPoint1 = SKTexture(imageNamed: "pot1")  
7      let textPoint2 = SKTexture(imageNamed: "pot2")
```

onInit()

```
36             point.zPosition = 10  
37             addChild(point)  
38             arrPoint.append(point)  
39         }  
40  
41         cat.position = onGetPosition(startIndex)  
42         cat.zPosition = 20  
43         addChild(cat)
```

設定位置並加入
到PointContainer

Run it



iPhone 11 — 13.3

GameScene.swift

宣告一個SKAudioNode來播放
背景音樂

```
3 class GameScene: SKScene {  
4     let pcontainer = PointContainer()  
5     var backgroundMusic : SKAudioNode!  
6     let touchMusic = SKAction.playSoundFileNamed("oh_no.mp3", waitForCompletion: false)
```

GameScene.swift In func didMove()

當我們觸摸螢幕需要播放提示音

```
15         pcontainer.onInit()  
16  
17         if let musicURL = Bundle.main.url(forResource: "whistle", withExtension: "mp3") {  
18             backgroundMusic = SKAudioNode(url: musicURL)  
19             addChild(backgroundMusic)  
20         }  
21     }
```

宣告musicURL來存背景音樂位址
當加入到Scene後它將循環播放

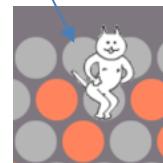
GameScene.swift

```
22     override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
23         run(touchMusic)
24         for touch:AnyObject in touches{
25             let location = touch.location(in:self)
26             let arrObject = self.nodes(at:location)
27             for p in arrObject {
28                 let point = p as? EPoint
29                 if point != nil && point!.type != pointtype.red {
30                     pcontainer.onNextPoint(point!.index)
31                 }
32             }
33         }
34     }
35 }
```

當觸碰到灰點時，呼叫
onNextPoint函式來判斷輸贏
或處理Cat的移動
Index為你觸碰點之index

此函式處理Cat下一步要怎麼走
Index為目前所點到的node's index

isEdge為true且為灰點



PointContainer.swift

```
114 func onGetNextPoint(_ index: Int){  
115     onSetRed(index)  
116     for point in arrPoint[currIndex].aroundPoint {  
117         if arrPoint[point].isEdge && arrPoint[point].type == pointtype.gray {  
118             gameReset("You loss !")  
119         }  
120     }  
121     onResetStep()  
122     let currPoint = arrPoint[currIndex]  
123     currPoint.step = 0  
124  
125     arrNext.append(currIndex)  
126     onFind(arrNext)  
127  
128  
129     if !isFind {  
130         gameReset("You win !")  
131     }  
132 }
```

判斷Cat 下一步是否為
出口

尋找路徑的資料還原
並設定Cat所在點的
step

尋找路徑開始

尋找路徑結束後,根據
isFind判斷輸贏

此函式用來重置路徑狀態與step

PointContainer.swift

```
133     func onResetStep (){
134         arrNext = [Int]()
135         isFind = false
136
137         for p in arrPoint {
138             p.step = 99
139             p.prePointIndex = -1
140         }
141     }
142     func onFind(_ arrNext:[Int]) {
143         if !isFind {
144             let arrNext = onGetNexts(arrNext)
145             if arrNext.count != 0 {
146                 onFind(arrNext)
147             }
148         }
149     }
```

如果找到邊緣點，
就停止遞迴

onFind -> onGetNexts -> onFind -> onGetNexts
直到isFind為true或沒有外圈點時跳出遞迴

PointContainer.swift

```
151 func onGetNexts(_ arrNext:[Int])->[Int]{  
152     let step = arrPoint[arrNext[0]].step + 1  
153     var tempPoints = [Int]()  
154     for nextP in arrNext {  
155         if isFind {  
156             break  
157         }  
158         for p in arrPoint[nextP].aroundPoint{  
159             if arrPoint[p].isEdge && arrPoint[p].type == pointtype.gray {  
160                 isFind = true  
161                 onGetPrePoint(arrPoint[p])  
162                 break  
163             }  
164             else if ( arrPoint[p].type == pointtype.gray ) && ( arrPoint[p].step > arrPoint[nextP].step ) {  
165                 arrPoint[p].step = step  
166                 arrPoint[p].prePointIndex = arrPoint[nextP].index  
167                 if arrPoint[p].index != arrPoint[nextP].prePointIndex {  
168                     tempPoints.append(p)  
169                 }  
170             }  
171         }  
172     }  
173     return tempPoints  
174 }  
175 }
```

此函式用於尋找路徑

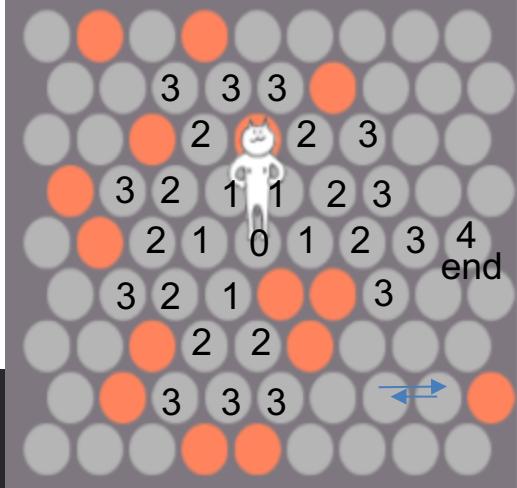
設定外圍點的step值

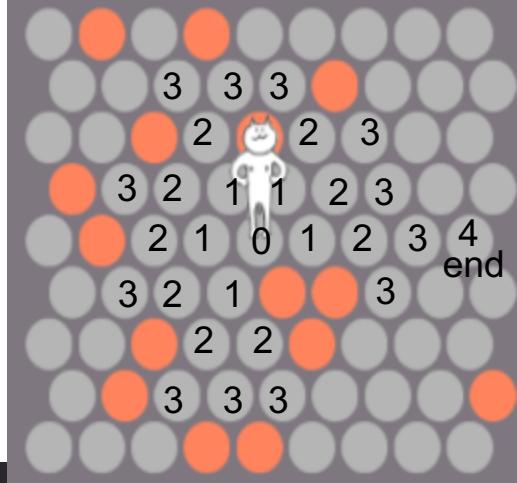
如果找到邊緣點
則跳出loop

如果找到邊緣點時
利用onGetPrePoint
來找Cat所在點

避免外圈相
鄰兩點互相
無限搜索

周圍點step大於
目前點的step,表
沒有
搜索過,因為沒
搜索過的點step為99





PointContainer.swift

```
177     func onGetPrePoint(_ point:EPoint){  
178         var p2 = point.aroundPoint[0]  
179         for p in point.aroundPoint {  
180             if arrPoint[p].step < arrPoint[p2].step{  
181                 p2 = p  
182             }  
183         }  
184         if arrPoint[p2].step == 0 {  
185             cat.position = onGetPosition(point.index)  
186             self.currtIndex = point.index  
187         }  
188         else {  
189             onGetPrePoint(arrPoint[p2])  
190         }  
191     }  
192 }
```

找到自己周圍點的step小於目前點step的點

4->3->2->1

代表Cat要走的點
因為Cat所在的step為0

如果還沒有到Cat的周圍,則繼續遞迴

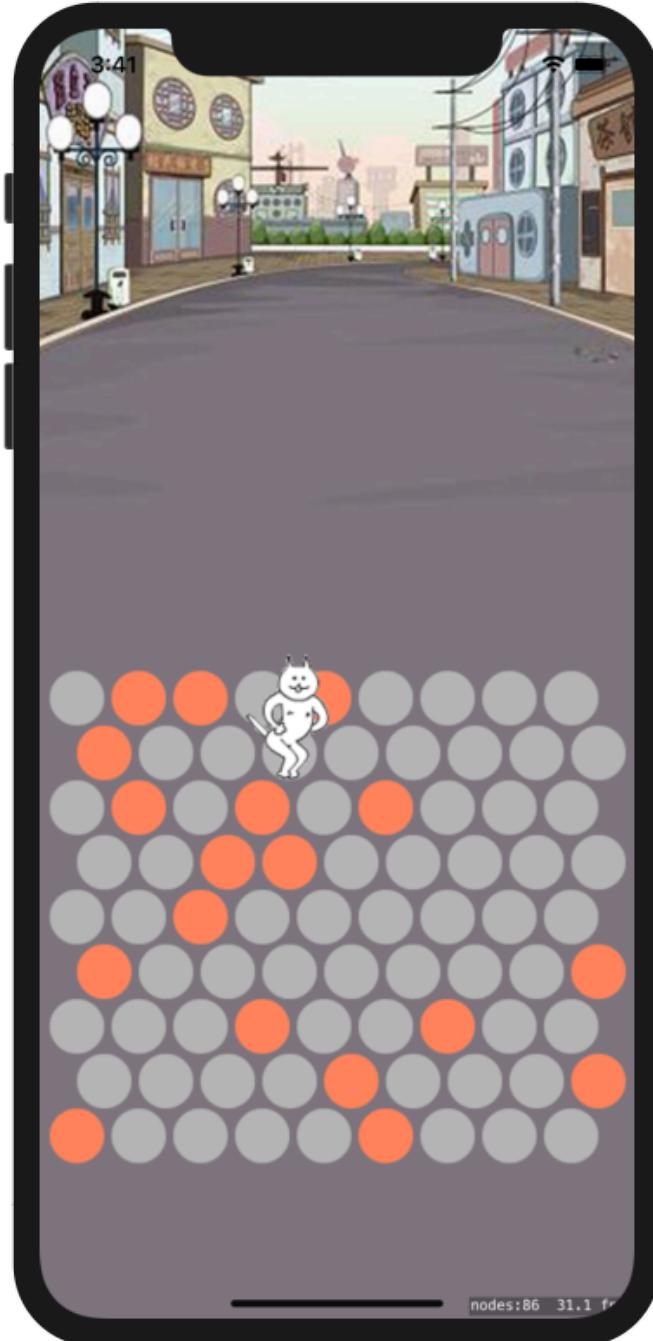
PointCotainer.swift

```
196     func gameReset(_ result: String) {
197         let alert = UIAlertController(title: result, message: nil, preferredStyle: .alert)
198         let alertAction = UIAlertAction(title: "Again", style: .default, handler: nil)
199         alert.addAction(alertAction)
200         self.scene?.view?.window?.rootViewController?.present(alert, animated: true, completion: nil)
201
202         for point in arrPoint {
203             point.type = pointtype.gray
204             point.texture = textPoint1
205         }
206         cat.position = onGetPosition(startIndex)
207         currIndex = startIndex
208         onCreateRed()
209         onResetStep()
210     }
211 }
```

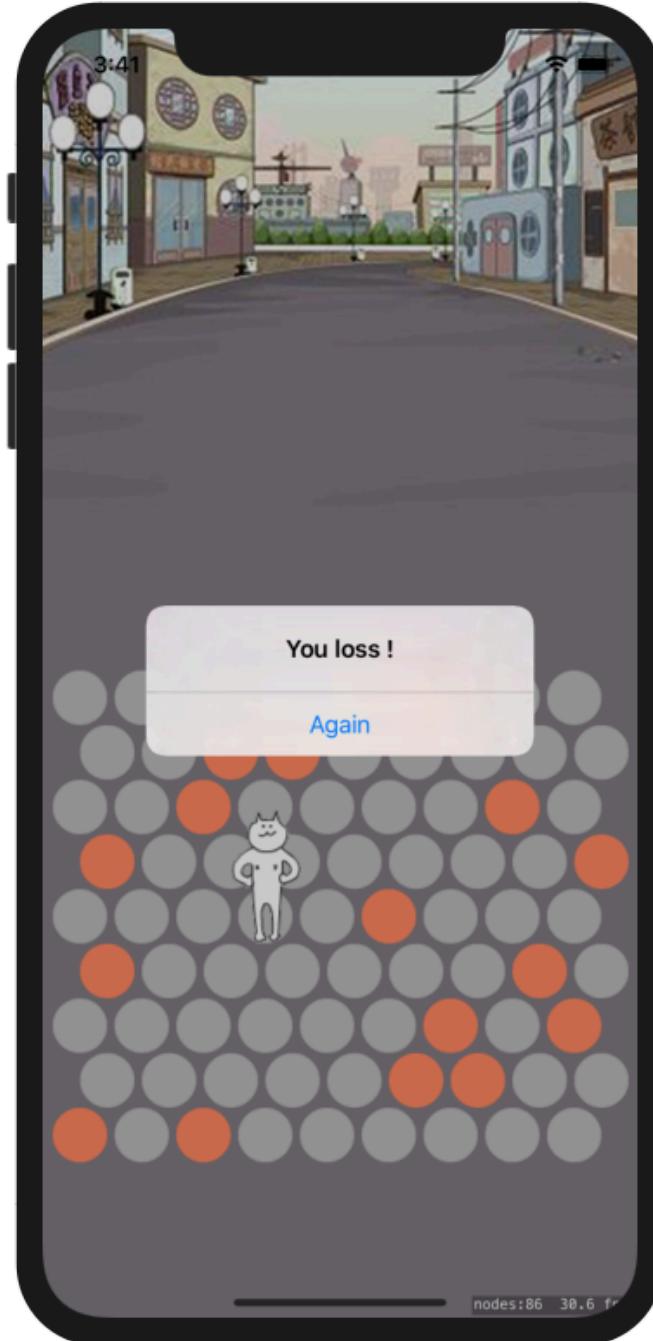
加入一個UIAlertController

還原到初始狀態

Run it



iPhone 11 — 13.3



iPhone 11 — 13.3

Physical collision

三者皆為UInt32型態 ($0x1 = 0...01$) , ($0x1 \ll 1 = 0..010$) ，使用上皆需設定在 SKPhysicsBody() 之後

categoryBitMask : 用於對 nodes 的碰撞分類, 可分為至多 32 個分類, 預設為 1.....1(32bits)

contactTestBitMask : 用於決定可被哪些分類碰撞(一定會碰撞)且取得 contact , contact 包含 didBegin() 與 didEnd() 函式, 預設為 0...0(32bits)

若不止一類需要 contact, 可使用 or 運算 :

```
ship.physicsBody?.contactTestBitMask = 0x1 << 2 | 0x1 << 3 | 0x1 << 4
```

collisionBitMask : 用於決定可被哪些分類產生碰撞影響, 不會取得 contact, 預設為 1.....1(32bits)

Ex: A node 之category為 $0x1 << 1$, collision為 $0x1 << 3$
B node 之category為 $0x1 << 3$, collision為 $0x1 << 4$

- 由於A之collision設為 $0x1 << 3$ 與B之category 執行AND運算後為非零值, 則A與B碰撞後,A將產生碰撞影響(A可能被撞飛等效果)
- 但B之collision為 $0x1 << 4$,所以B將不受與A碰撞產生的影響(可能直接穿過A),B只有與 $0x1 << 4$ 碰撞,B才會被撞飛等效果