

第 5 章

游戏主菜单设计

本章将从无到有一步步介绍如何使用 Swift 程序语言,设计出游戏的主菜单,如图 5-1 所示。



图 5-1 游戏主菜单设计

5.1 创建 Swift GameKit

1. 介绍

本章将介绍如何创建一个全新的游戏。

2. 步骤

STEP1：打开 Xcode

将 Xcode 开发工具打开。

STEP2：创建一个全新的程序

如图 5-2 所示,通过下拉式菜单选择 File→New→Project,创建一个全新的程序。

STEP3：设置式样

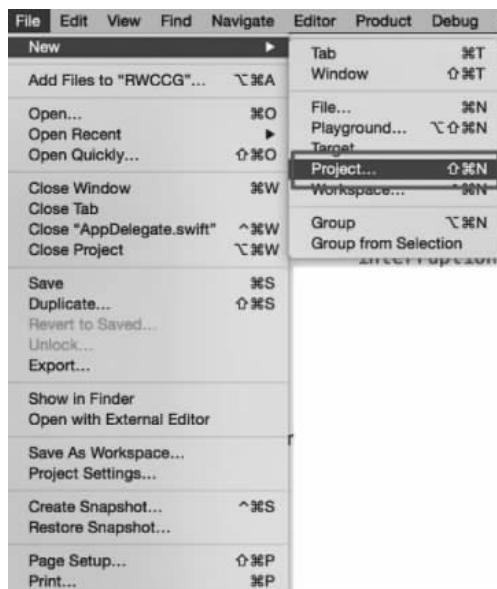


图 5-2 创建一个全新的程序

如图 5-3 所示,选择 iOS→Application 和 Game,之后单击 Next 按钮,进入下一步骤。

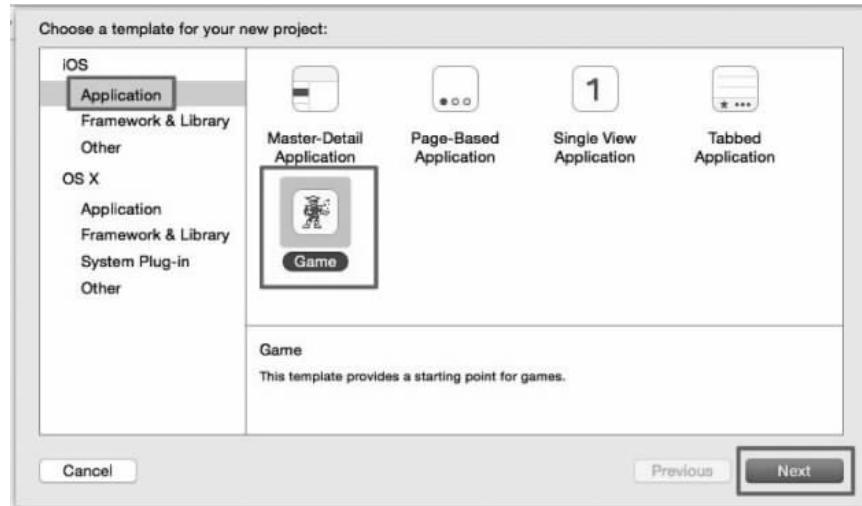


图 5-3 设置式样

STEP4: 设置名称

设置程序名称,按照表 5-1 的建议,选择和输入程序的设置,设置完成之后,单击 Next 按钮,进入下一步骤,如图 5-4 所示。

表 5-1 名称功能介绍

名 称	输入数据	注 释
Product Name	MyGame	名称
Organization Name	xxx	公司或学校的名字
Bundle Identifier	xxxx.com	唯一号码,一般都使用网站地址
Language	Swift	程序语言
Game Technology	SpriteKit	游戏技术
Devices	iPhone	开发后运行的设备

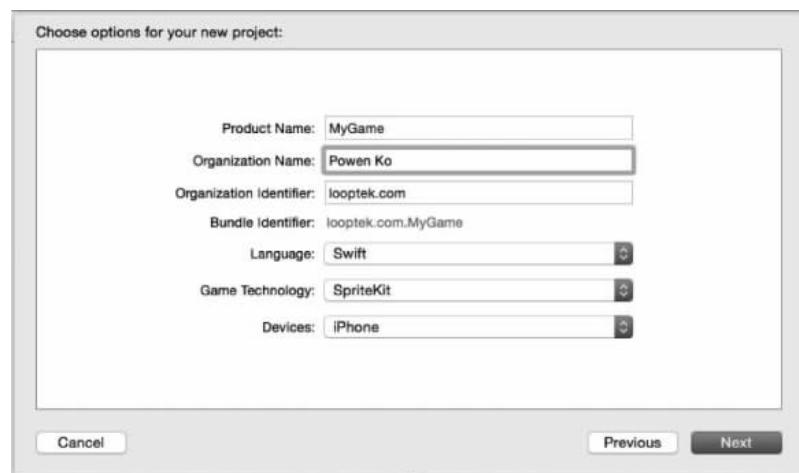


图 5-4 设置名称-1

Xcode 会询问存储位置,决定存储的位置之后,单击 OK 按钮确认,就会进入程序。

STEP5: 运行程序

接下来就会看到整个程序的工作环境,按照如图 5-5 所示的操作运行程序来测试程序。

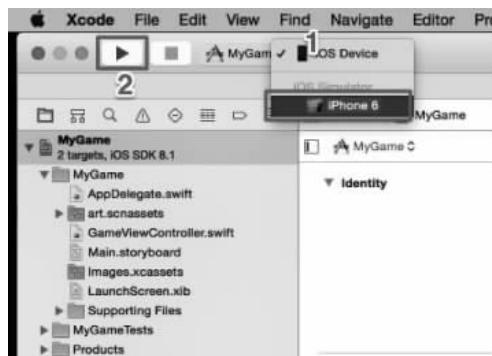


图 5-5 设置名称-2

(1) 选择信息设备的选项,把原本 iOS Device 修改为 iPhone 6 设备。

(2) 单击界面中的三角形按钮运行程序。

3. 结果

Swift APP 打开的时候,如图 5-6 所示,会有 Hello World 的字符出现在界面上。如果单击界面,每单击一次就会产生一只新的飞机,也可以通过下拉式菜单选取 Hardware→Rotate Left 调整模拟器的角度。



图 5-6 运行结果

4. 教学视频

完整的教学视频可以看 swift_05_GameKit。

5. 问题

错误排除

按照过去的经验,很多开发者运行 APP 时会发生问题,例如 Simulator error FBSSystemServiceDomain code 4,如图 5-7 所示。



图 5-7 错误别名 code 4

解决方法

在“iOS 仿真器”上,通过下拉式菜单选择 iOS Simulator → Reset Content and Settings,这样就可以了。或者把仿真器关掉再重新打开,这问题就顺利解决了。

5.2 主菜单后台——图片、指定位置

1. 目的

在 Swift 产生的 GameKit 样例程序语言中,官方的程序内容过于复杂,所以本节会改这个程序,并把它删除成简单版本,然后通过修改程序加上一个 JPG 的图片,作为游戏主菜单的后台图片。

2. 步骤

STEP1: 继续 5.1 节

延续 5.1 节的样例,继续往下。

STEP2: 添加一张图片

准备一张图片,当作游戏主菜单的后台图,推荐使用 JPG 或 PNG 格式的图片,推荐使用的文件尺寸不超过屏幕的尺寸,例如:

- (1) 旧 iPhone 屏幕分辨率为 320×480 pixels;
- (2) iPhone6 屏幕分辨率为 750×1334 pixels;
- (3) iPhone6 plus 屏幕分辨率为 1080×1920 pixels。

如图 5-8 所示,使用鼠标拖曳文件,把图片拖进 Xcode 的程序中。

STEP3: 设置 APP 的垂直画面或者水平画面

设置 APP 的纵向画面或者水平画面,如图 5-9 所示,单击程序名称,但是 Portrait 和 Upside Down 两个选项不要选中,这样设置的 App 是水平画面,即 Landscape Left 和 Landscape Right 是设定为水平画面。同理,反过来就是纵向画面。



图 5-8 拖曳图片进入 Xcode 程序中



图 5-9 设置 APP 的垂直画面或者是水平画面

STEP4：修改程序

程序到目前为止，画面上有两个东西显示，一个是 FPS，另一个是 Node。

- (1) FPS：每秒所显示的画面数。
- (2) Node：画面上有几个点。

3. 样例

1) 样例程序：ch05\5-2\GameScene.swift

```

1.  class GameViewController: UIViewController {
2.
3.      override func viewDidLoad() {
4.          super.viewDidLoad()
5.
6.          if let scene = GameScene.unarchiveFromFile("GameScene") as? GameScene {
7.              //画面设置
8.              let skView = self.view as SKView
9.              //skView.showsFPS = true           //显示 FPS 信息
10.             //skView.showsNodeCount = true    //显示 Node 信息
11.
12.             /* 设置使用画面的优化效果来提高渲染性能 */
13.             skView.ignoresSiblingOrder = true
14.
15.             /* 调整场景尺寸缩放比例来符合手机画面 */
16.             scene.scaleMode = .AspectFill
17.
18.             skView.presentScene(scene)
19.         }
20.     ...
21.     }
22.

```

程序解说

第 9 行：为了画面的简洁所以把第 9 行显示的 FPS 信息拿掉，如图 5-10 所示。

第 10 行：为了画面的简洁所以把第 10 行显示的 Node 信息拿掉。



图 5-10 删除第 9、10 行，这个画面就会消失

2) 样例程序：ch05\5-2\GameScene.swift

```

1.  class GameScene: SKScene {
2.      override func didMoveToView(view: SKView) {
3.          /* 设置游戏的场景 */

```

```
4.         let myLabel = SKLabelNode(fontNamed:"Chalkduster")
5.         myLabel.text = "copyright 2015 PowenKo.com, All Rights Reserved.";
6.         myLabel.fontSize = 32;
7.         myLabel.position = CGPoint(x:CGRectGetMidX(self.frame), y: 100);
8.         self.addChild(myLabel)
9.
10.        ///////////////////////////////////////////////////////////////////
11.        var bgImage = SKSpriteNode(imageNamed: "bg.jpg")      //定义变量
12.        self.addChild(bgImage)                                //把图片加上
13.
14.        bgImage.position = CGPoint(x:CGRectGetMidX(self.frame), y:CGRectGetMidY(self.
frame));                                         //设置位置
15.
16.    }
17. ...
18. }
```

程序解说

第5行：指定文字的内容。

第6行：改变字体尺寸。

第7行：调整显示文字的位置。

第14行：调整显示图片的位置。

结果

程序样例运行结果如图 5-11 所示。



图 5-11 运行结果

4. 教学视频

完整的教学视频可以看 swift_05_2_AddImage_Text。

5. 函数

1) SKScene.size 取得显示器的实际长宽

开发程序时,需要知道实际设备画面尺寸,可以通过以下程序实现。

示例

```
class GameScene: SKScene {
    override func didMoveToView(view: SKView) {
        var y1 = self.size.height //768
        var x1 = self.size.width //1024
    }
}
```

2) CGRectGetMidY 取得显示器的中间位置

取得实际设备画面正中间的位置,可以通过以下程序实现。

示例

```
class GameScene: SKScene {
    override func didMoveToView(view: SKView) {
        var y2 = CGRectGetMidY(self.frame) //384
        var x2 = CGRectGetMidX(self.frame) //512
    }
}
```

3) position 设置位置

position 用来设置位置,CGPoint(x:0, y:0);分别指定 X 和 Y 的位置。

示例

```
class GameScene: SKScene {
    override func didMoveToView(view: SKView) {
        var bgImage = SKSpriteNode(imageNamed: "bg.jpg") //定义变量放图
        self.addChild(bgImage)
        bgImage.position = CGPoint(x:0, y:0);
    }
}
```

5.3 加上菜单——显示按钮

1. 目的

游戏中总会需要有选项,实现用户单击的功能。一般游戏设计中,不可能使用 iOS 按键组件,例如 UIButton,会使用图片的显示方法加上按键的处理,来达到目的。

2. 步骤

STEP1: 继续 5.2 节

延续 5.1 节的样例,继续往下。

STEP2: 添加两张图片

准备两张图片,分别当作按钮和按下后的图,使用鼠标拖曳文件,把图片拖进 Xcode 的

程序中,如图 5-12 所示。

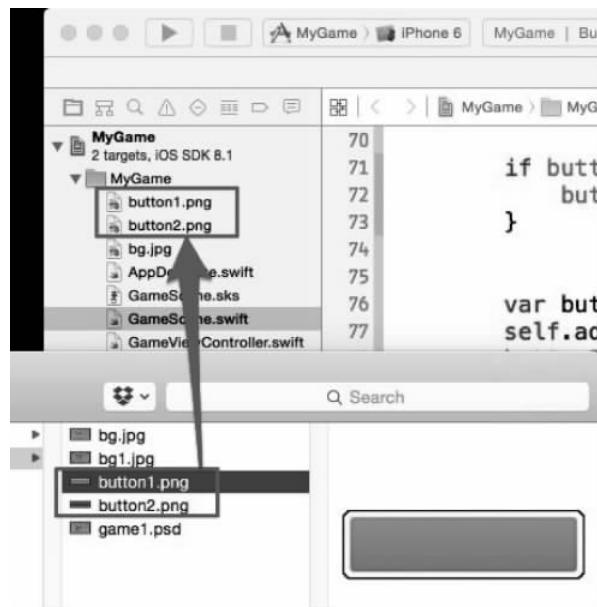


图 5-12 拖曳图片进入 Xcode 的程序中

STEP3：修改程序

在这里使用增加图片 SKSpriteNode 和增加文字技巧 SKLabelNode,重叠在一起让它看起来像是按钮,并且做两个按钮。

3. 样例

样例程序：ch05\5-2\GameScene.swift

```

1. import SpriteKit
2.
3. class GameScene: SKScene {
4.     override func didMoveToView(view: SKView) {
5.         /* 设置场景 */
6.         let myLabel = SKLabelNode(fontNamed:"Chalkduster")
7.         myLabel.text = "copyright 2015 PowenKo.com, All Rights Reserved."
8.         myLabel.fontSize = 32;
9.         myLabel.position = CGPointMake(x:CGRectGetMidX(self.frame),y: 100);
10.        self.addChild(myLabel)
11.
12.        //////////////////////////////// 后台
13.        var bgImage = SKSpriteNode(imageNamed: "bg.jpg")           //定义变量
14.        self.addChild(bgImage)
15.        bgImage.zPosition = -1                                     //把后台图放后面

```

```

16.
17.     bgImage.position = CGPointMake(x:CGRectGetMidX(self.frame), y:CGRectGetMidY(self.
18.                                     frame));
19.     ////////////////////////////// 菜单按钮
20.     var button1 = SKSpriteNode(imageNamed: "button1.png") //放图
21.     self.addChild(button1)
22.     button1.position = CGPointMake(x: 300, y:200 );           //设置位置
23.     let myLabel2 = SKLabelNode(fontNamed:"Chalkduster")      //字形
24.     myLabel2.text = "About Us";                                //字
25.     myLabel2.fontSize = 28;                                    //字尺寸
26.     myLabel2.fontColor = UIColor(red: 255/255, green: 255, blue: 255/255, alpha: 200/255)
27.     myLabel2.position = CGPointMake(x: 300, y:200 - 12 );    //设置位置
28.     self.addChild(myLabel2)
29.
30.     var button2 = SKSpriteNode(imageNamed: "button1.png") //放图
31.     self.addChild(button2)
32.     button2.position = CGPointMake(x: 300, y:300 );           //设置位置
33.     let myLabel3 = SKLabelNode(fontNamed:"Chalkduster")      //字形
34.     myLabel3.text = "Start";                                 //字
35.     myLabel3.fontSize = 28;                                //字尺寸
36.     myLabel3.position = CGPointMake(x: 300, y:300 - 12 );    //设置位置
37.     self.addChild(myLabel3)
38. ...

```

程序解说

第 18~25 行：显示 About 按钮。

第 27~34 行：显示 Start 按钮

注意第 15 行，一般如果没有特别指定 zPosition，所有图片都会放在 0 的位置，出现图片的上下关系，就按照 addChild 加入的前后顺序，但是有的时候两张图片会重叠在一起甚至会消失。各位可试试看，如果把 zPosition 拿掉，使用鼠标单击画面，飞机不见得会全部出现在画面上。

结果

在游戏画面上会产生两个按钮，并且使用鼠标单击画面，飞机就会出现在游戏上，如图 5-13 所示。

4. 教学视频

完整的教学视频可以看 swift_05_3_Button。

5. 函数

1) SKLabelNode, fontColor 设置字体颜色

使用 SKLabelNode 怎么设置文字尺寸和颜色呢？请看以下示例。



图 5-13 运行结果

示例

```
class GameScene: SKScene {
    override func didMoveToView(view: SKView) {
        let myLabel2 = SKLabelNode(fontNamed:"Chalkduster") //字形
        myLabel2.fontSize = 28; //字尺寸
        myLabel2.fontColor = UIColor(red: 255/255, green: 255, blue: 255/255, alpha: 200/255)
        self.addChild(myLabel3)
    }
}
```

2) SKSpriteNode.zPosition 设置图片放置的前后

使用 zPosition 设置图片放置的前后关系,数字越大代表越上层。

示例

```
class GameScene: SKScene {
    override func didMoveToView(view: SKView) {
        var button3 = SKSpriteNode(imageNamed: "button1.png")
        self.addChild(button3)
        button3.position = CGPointMake(x: 850,y:200 );
        button3.zPosition = 2
        var button4 = SKSpriteNode(imageNamed: "button2.png")
        self.addChild(button4)
        button4.position = CGPointMake(x: 800,y:200 );
        button4.zPosition = 1
    }
}
```

当加上这个 zPosition,就会强制指定图片的上下关系,如图 5-14 所示。



图 5-14 zPosition 的效果

5.4 单击菜单——按键处理

1. 目的

紧接游戏菜单，要如何检测用户按下选项呢？在本节加上按键的处理，来达到视觉上的按键效果。也就是说本节，将学习使用鼠标和触控显示器的处理技巧，并判断是否单击到按键菜单。

2. 步驟

STEP1：继续 5.3 节

延续 5.3 节的样例，继续往下。

STEP2：修改程序

接下来修改程序，并且处理按键的操作。

3. 样例

样例程序：ch05\5-4\GameScene.swift

```
1. ...
2. import SpriteKit
3.
4. class GameScene: SKScene {
5.     var button1 = SKSpriteNode(imageNamed: "button1.png") //菜单图
6.     var button2 = SKSpriteNode(imageNamed: "button1.png") //菜单图
7.
8.     override func didMoveToView(view: SKView) {
9.         /* 设置场景 */
10.        ...
11.        ///////////////////////////////
12.        //var button1 = SKSpriteNode(imageNamed: "button1.png")
13.        self.addChild(button1)
14.        button1.position = CGPointMake(x: 300, y:200 );
15.        ...
16.        //var button2 = SKSpriteNode(imageNamed: "button1.png")
17.        self.addChild(button2)
18.        button2.position = CGPointMake(x: 300, y:300 );
19.        ...
20.
21.    }
22. } //手指头移动的触发事件
```

```

23.     override func touchesMoved(touches: NSSet, withEvent event: UIEvent) {
24.     }
25.                                         //手指头放开的触发事件
26.     override func touchesEnded(touches: NSSet, withEvent event: UIEvent) {
27.         button1.texture = SKTexture(imageNamed: "button1.png") //改变图
28.         button2.texture = SKTexture(imageNamed: "button1.png") //改变图
29.     }
30.                                         //点下去的触发事件
31.     override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
32.     }
33.     var touch: UITouch = touches.allObjects[0] as UITouch
34.     var location: CGPoint = touch.locationInNode(self)      //取得按下的位置
35.     if button2.containsPoint(location) {                      //判断是否点到菜单
36.         button2.texture = SKTexture(imageNamed: "button2.png") //改变图
37.     }
38.
39.     if button1.containsPoint(location) {                      //判断是否点到菜单
40.         button1.texture = SKTexture(imageNamed: "button2.png")
41.     }
42.
43.     var buttonx = SKSpriteNode(imageNamed: "button1.png")
44.     self.addChild(buttonx)
45.     button2.position = location
46.
47.     /*
48.     for touch: AnyObject in touches {
49.         ...
50.         let sprite = SKSpriteNode(imageNamed:"Spaceship")
51.         self.addChild(sprite)
52.     } */
53. }
54.                                         //画面更新的触发事件
55.     override func update(currentTime: CFTimeInterval) {
56.     }
57. }
58.
59.

```

程序解说

第 31 行：按下的反应处理。

第 35 行：判断是否单击到菜单。

第 47~52 行：删除原本按下后会显示飞机的程序。

结果

当用户单击游戏的菜单时，按键的颜色就会变得不一样，如图 5-15 所示。

4. 教学视频

完整的教学视频可以看 swift_05_4_PressButton。



图 5-15 运行结果

5. 函数

1) touchesBegan(touches: NSSet, withEvent event: UIEvent) 单击屏幕触发事件
取得单击的反应时,可以通过 touchesBegan 取得触发事件。

示例

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
    var touch: UITouch = touches.allObjects[0] as UITouch
    var location: CGPoint = touch.locationInNode(self)
    var buttonx = SKSpriteNode(imageNamed: "button1.png")
    self.addChild(buttonx)
    button2.position = location
}
```

2) touchesEnded(touches: NSSet, withEvent event: UIEvent) 单击放开触发事件
取得单击放开的反应时,可以通过 touchesEnded 取得触发事件。

示例

```
override func touchesEnded(touches: NSSet, withEvent event: UIEvent) {
    var touch: UITouch = touches.allObjects[0] as UITouch
    var location: CGPoint = touch.locationInNode(self)
    var buttonx = SKSpriteNode(imageNamed: "button1.png")
    self.addChild(buttonx)
    button2.position = location
}
```

3) touchesMoved(touches: NSSet, withEvent event: UIEvent) 移动的触发事件
取得移动的反应时,可以通过 touchesMoved 取得触发事件。

示例

```
override func touchesMoved(touches: NSSet, withEvent event: UIEvent) {
    var touch: UITouch = touches.allObjects[0] as UITouch
    var location: CGPoint = touch.locationInNode(self)
    var button1 = SKSpriteNode(imageNamed: "button1.png")
    self.addChild(button1)
    button1.position = location
}
```

5.5 主角显示——图层处理

1. 目的

Swift 产生的 GameKit 样例程序语言中,有个很重要的观念,就是每一个图层都会有上下关系。设置图层是通过 zPosition 的功能指定,本节将会再加上两个图层让游戏主菜单看起来更加丰富,如图 5-16 所示,会添加一个第 0 层的烟花效果和第 2 层的主角图片在游戏主菜单上。

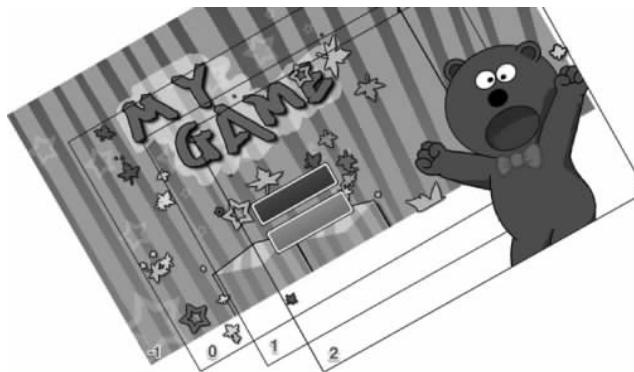


图 5-16 zPosition 图层

2. 步骤

STEP1: 继续 5.4 节

延续 5.4 节的样例,继续往下。

STEP2: 添加图片

添加第 0 层的烟花效果图和第 2 层的主角图片,这两张图片通过文件夹 Finder,把图片加入到程序中,如图 5-17 所示。

STEP3: 修改程序

接下来修改程序,添加图片 SKSpriteNode 和设置每一张图片的图层上下关系 zPosition。

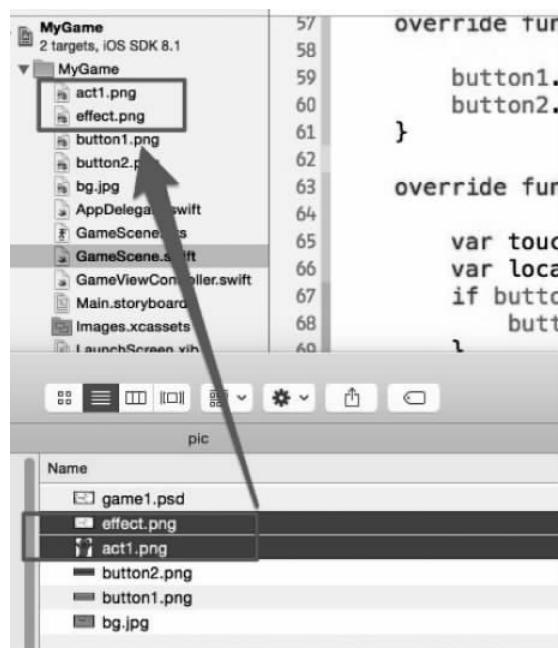


图 5-17 添加图片

3. 样例

样例程序：ch05\5-5\GameScene.swift

```

1. import SpriteKit
2.
3. class GameScene: SKScene {
4.     var button1 = SKSpriteNode(imageNamed: "button1.png")
5.     var button2 = SKSpriteNode(imageNamed: "button1.png")
6.
7.     override func didMoveToView(view: SKView) {
8.         /* 设置场景 */
9.         let myLabel = SKLabelNode(fontNamed:"Chalkduster")
10.        myLabel.text = "copyright 2015 PowenKo.com, All Rights Reserved.";
11.        myLabel.fontSize = 32;
12.        myLabel.position = CGPointMake(x:CGRectGetMidX(self.frame),y: 100);
13.        self.addChild(myLabel)
14.        myLabel.zPosition = 3
15.
16.        ////////////////////////////////后台
17.        var bgImage = SKSpriteNode(imageNamed: "bg.jpg")
18.        self.addChild(bgImage)
19.        bgImage.zPosition = -1           //图层上下关系
20.        bgImage.position = CGPointMake(x:CGRectGetMidX(self.frame), y:CGRectGetMidY(self.

```

```

        frame))
21.     //////////////////////////////////////////////////////////////////菜单
22.
23.     self.addChild(button1)
24.     button1.position = CGPointMake(x: 300,y:200 );
25.     button1.zPosition = 1           //图层上下关系
26.     let myLabel2 = SKLabelNode(fontNamed:"Chalkduster")
27.     myLabel2.text = "About Us"
28.     myLabel2.fontSize = 28
29.     myLabel2.fontColor = UIColor(red: 255/255, green: 255, blue: 255/255, alpha: 200/255)
30.     myLabel2.position = CGPointMake(x: 300,y:200 - 12 );
31.     myLabel2.zPosition = 1           //图层上下关系
32.     self.addChild(myLabel2)
33.
34.     self.addChild(button2)
35.     button2.position = CGPointMake(x: 300,y:300 );
36.     button2.zPosition = 1           //图层上下关系
37.     let myLabel3 = SKLabelNode(fontNamed:"Chalkduster")
38.     myLabel3.text = "Start"
39.     myLabel3.fontSize = 28
40.     myLabel3.fontColor = UIColor(red: 255/255, green: 255, blue: 255/255, alpha: 200/255)
41.     myLabel3.position = CGPointMake(x: 300,y:300 - 12 )
42.     myLabel3.zPosition = 1           //图层上下关系
43.     self.addChild(myLabel3)
44.
45.     ////////////////////////////////////////////////////////////////// 烟花效果图
46.     var effect = SKSpriteNode(imageNamed: "effect.png")
47.     self.addChild(effect)
48.     effect.zPosition = 0           //图层上下关系
49.     effect.position = CGPointMake(x:CGRectGetMidX(self.frame), y:CGRectGetMidY(self.
frame))
50.     ////////////////////////////////////////////////////////////////// 主角图片
51.     var act1 = SKSpriteNode(imageNamed: "act1.png")
52.     self.addChild(act1)
53.     act1.zPosition = 2           //图层上下关系
54.     act1.position = CGPointMake(x:CGRectGetMidX(self.frame) + 250, y:CGRectGetMidY
(self.frame) - 150)
55.   }
56. ...
57. }

```

程序解说

第 21~43 行：加上烟花效果图并且设置图层为 0。

第 45~49 行：加上烟花效果图并且设置图层为 0。

第 50~54 行：加上主角的图片并且设置图层为 2。

注意图层 zPosition 的数字，数字越大，代表图片位置越近；数字越小，代表图片位置

越远。

结果

执行之后的效果如图 5-18 所示,因为有图片前后的关系,并且加上更多的图片重叠在一起,强化游戏的菜单,看起来更专业和漂亮。



图 5-18 运行结果

4. 教学视频

完整的教学视频可以看 swift_05_5_zPosition。

5.6 习题

1. 问答题

- (1) SKLabelNode 和 SKSpriteNode 的用途?
- (2) SKScene 的用途?
- (3) SKSpriteNode 可以显示哪几种图片文件格式? 这些图片文件格式的用途和差异性是什么?

2. 实践题

- (1) 修改本章的程序,并且在画面的上下左右四个角落,摆放文字,通过此 APP 知道屏幕 X 轴和 Y 轴的关系。
- (2) 按照本章所学的技巧,做出一款你心目中的游戏 APP 主菜单画面。