

## ■ Xcode 7.1 への対応方法

本書は、Xcode 6.1 を対応として書いていますので、現在の最新版 Xcode 7.1 では Xcode の仕様が変更エラーが出るようになりました。

Xcode 7.1 に対応させるには、書き方を変えたり、「!」記号をつけたりする必要がありますので、その修正ヶ所についてリストアップいたしましたので、こちらをご覧ください。

また、サンプルファイルも Xcode 7.1 に対応したものを用意いたしましたので、こちらをご利用ください。

---

### Xcode 7.1 への修正方法

---

#### ■ 全サンプル共通

---

[GameViewController.swift]

【誤】

```
let skView = self.view as SKView
```

【正】

```
let skView = self.view as! SKView
```

【誤】

```
override func supportedInterfaceOrientations() -> Int {
    if UIDevice.currentDevice().userInterfaceIdiom == .Phone {
        return Int(UIInterfaceOrientationMask.AllButUpsideDown.rawValue)
    } else {
        return Int(UIInterfaceOrientationMask.All.rawValue)
    }
}
```

【正】

```
override func supportedInterfaceOrientations() -> UIInterfaceOrientationMask {
    if UIDevice.currentDevice().userInterfaceIdiom == .Phone {
        return .AllButUpsideDown
    } else {
        return .All
    }
}
```

---

#### ■ Chapter 2: spriteKitTest

---

[GameScene.swift]

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

### ■ Chapter 3: omikuji

---

[GameScene.swift]

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

【誤】

```
func showOmikuji() {  
    var omikuji = ["大吉", "中吉", "吉", "凶"]  
    var r = Int(arc4random_uniform(4))  
    myLabel.text = omikuji[r]  
}
```

【正】

```
func showOmikuji() {  
    var omikuji = ["大吉", "中吉", "吉", "凶"]  
    let r = Int(arc4random_uniform(4))  
    myLabel.text = omikuji[r]  
}
```

---

### ■ Chapter 4

---

[TitleScene.swift]

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameOverScene.swift]

【誤】

```
let gameSKView = self.view as GameSKView
```

【正】

```
let gameSKView = self.view as! GameSKView
```

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameScene.swift]

【誤】

```
let skView = self.view as GameSKView
```

【正】

```
let skView = self.view as! GameSKView
```

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

## ■ Chapter 5

---

[TitleScene.swift]

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameOverScene.swift]

【誤】

```
let gameSKView = self.view as GameSKView
```

【正】

```
let gameSKView = self.view as! GameSKView
```

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameScene.swift]

【誤】

```
// ロケット噴射を用意する  
var rocketspk = SKEmitterNode(fileName: "mySpark")
```

【正】

```
// ロケット噴射を用意する  
var rocketspk = SKEmitterNode(fileName: "mySpark")!
```

【誤】

```
// 背景の星を作る  
func initStar() {  
    for i in 0...starCount { // 0～starCount まで繰り返す  
        // 星を作る  
        let star = SKSpriteNode(imageNamed: "star.png")  
        // 画面にランダムに表示する
```

```
var wx = Int(arc4random_uniform(750))
var wy = Int(arc4random_uniform(1334))
```

【正】

```
// 背景の星を作る
func initStar() {
    for _ in 0...starCount { // 0～starCount まで繰り返す
        // 星を作る
        let star = SKSpriteNode(imageNamed: "star.png")
        // 画面にランダムに表示する
        let wx = Int(arc4random_uniform(750))
        let wy = Int(arc4random_uniform(1334))
    }
}
```

【誤】

```
// 隕石を作る
func initMeteo() {
    for i in 0...meteoCount { // 0～meteoCount まで繰り返す

```

【正】

```
// 隕石を作る
func initMeteo() {
    for _ in 0...meteoCount { // 0～meteoCount まで繰り返す

```

【誤】

```
override func touchesMoved(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesMoved(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

【誤】

```
if distance < 90 {
    var spk = SKEmitterNode(fileName: "mySpark")
}
```

【正】

```
if distance < 90 {
    let spk = SKEmitterNode(fileName: "mySpark")!
}
```

【誤】

```
// カスタマイズした GameSKView にスコアを渡して
let skView = self.view as GameSKView
```

【正】

```
// カスタマイズした GameSKView にスコアを渡して
let skView = self.view as! GameSKView
```

---

## ■ Chapter 6

---

[TitleScene.swift]

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameOverScene.swift]

【誤】

```
let gameSKView = self.view as GameSKView
```

【正】

```
let gameSKView = self.view as! GameSKView
```

【誤】

```
override func touchesBegan(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

---

[GameScene.swift]

【誤】

```
// マシュマロが登場したときに、衝撃を与えて、斜め上方向に飛び出させる  
let vec = CGVector(dx: arc4random_uniform(200) - 100, dy: 100)
```

【正】

```
// マシュマロが登場したときに、衝撃を与えて、斜め上方向に飛び出させる  
let vec = CGVector(dx: Int(arc4random_uniform(200)) - 100, dy: 100)
```

【誤】

```
override func touchesMoved(touches: NSSet, withEvent event: UIEvent) {
```

【正】

```
override func touchesMoved(touches: Set<UITouch>, withEvent event: UIEvent?) {
```

【誤】

```
let skView = self.view as GameSKView
```

【正】

```
let skView = self.view as! GameSKView
```

---

<本書サポートサイト>

<http://www.shuwasystem.co.jp/support/7980html/4349.html>

<秀和システム>

<http://www.shuwasystem.co.jp/>