



**WE MAKE IOS GAMES  
FOR YOU!**

Documentation to games from TapTap Studio.  
It's doc file have popular questions to games from TapTap Studio.

If you have questions write to this mail: [taptap.main@gmail.com](mailto:taptap.main@gmail.com)

### **Contents**

1. Set basic project settings.
2. Settings ad's.
3. Send game to itunes connect.
4. Edit launch screen.
5. Replace some image in project.
6. Change size or position some object in project.
7. Change speed or some another setting in game process.
8. Get random value in project.
9. Get UIColor from hex value.
10. Change sounds in project.
11. Change links to rate us, share and other.
12. Change In-App id's
13. Add more skins

# 1. Set basic project settings.

Open project in Xcode

Tap to blue project file in Xcode

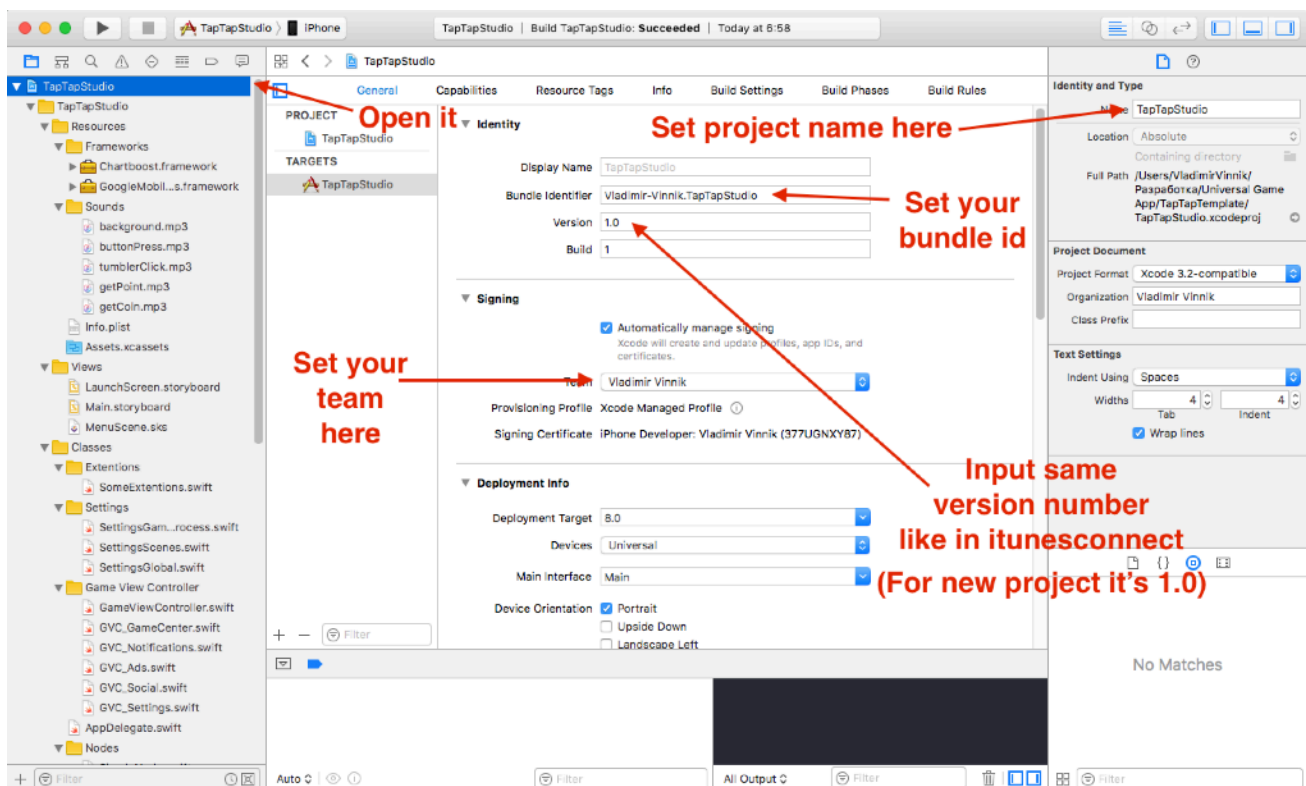
Here you can change:

- Bundle ID

- Name of project

- Project version

- Current build version



## 2. Settings ad's.

Go to folder with settings files. TapTapStudio -> Classes -> Settings

Open SettingsGlobal.swift file.

Find here «//Ads settings» line, after this line you can see ad's id's.

Replace id's in this lines to your.

```
//Ads settings
struct Ad {
    static let on = true
    static let testModeNow = true
    static let testDeviceID = "18bee757358b39892368561bd2bc85b3"

    static let idApp = "ca-app-pub-3940256099942544~1458002511"
    static let idInterstitial = "ca-app-pub-3940256099942544/4411468911"
    static let idBanner = "ca-app-pub-3940256099942544/4411468910"

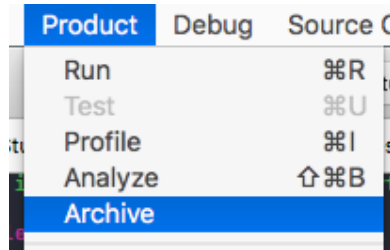
    struct Showing {
        struct Interstitial {
            static let game = false
            static let end = true
            static let menu = false
            static let settings = false
            static let store = false
            static let tutorial = false
        }

        struct Banner {
            static let game = false
            static let end = false
            static let menu = false
            static let settings = true
            static let store = false
            static let tutorial = false
        }
    }
}
```

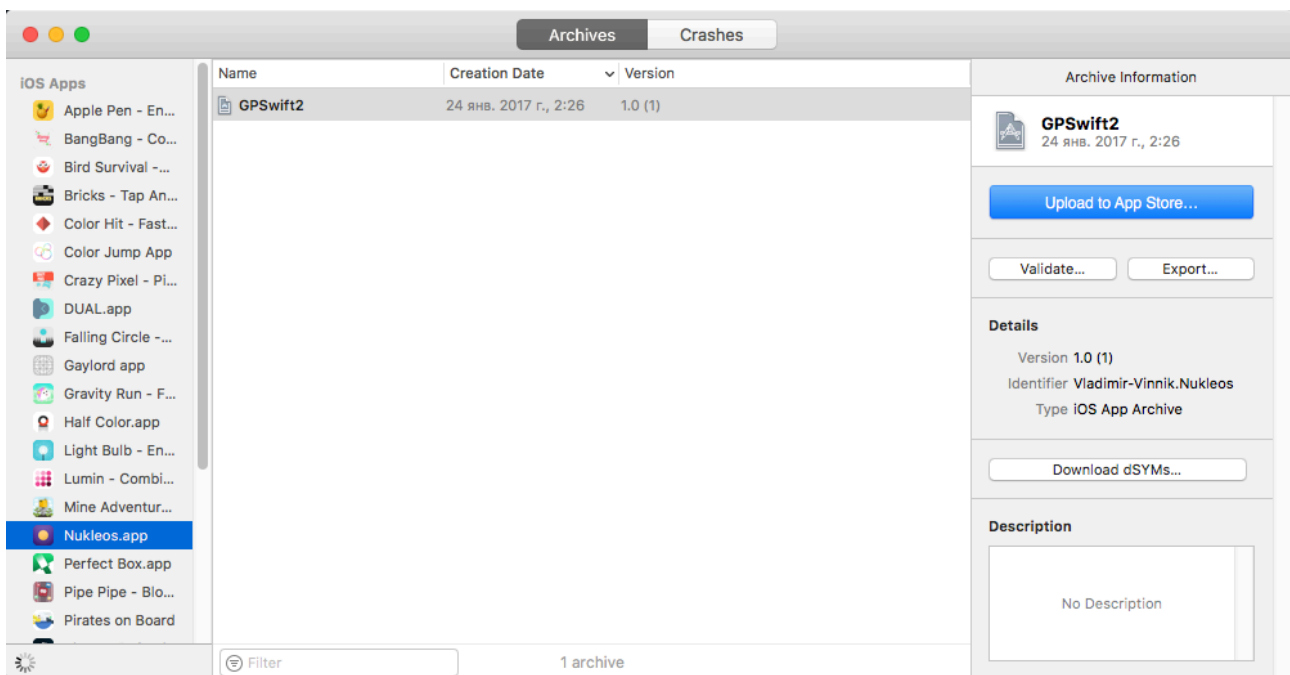
Change showing ad's settings

### 3. Send game to iTunes connect.

Open project in Xcode.  
Select Product -> Archive.  
Wait



Now you can see organizer.  
Select your project and press to button «Upload to App Store...»  
Wait



Your project in iTunes connect.  
(If you have some error, try to upload again)

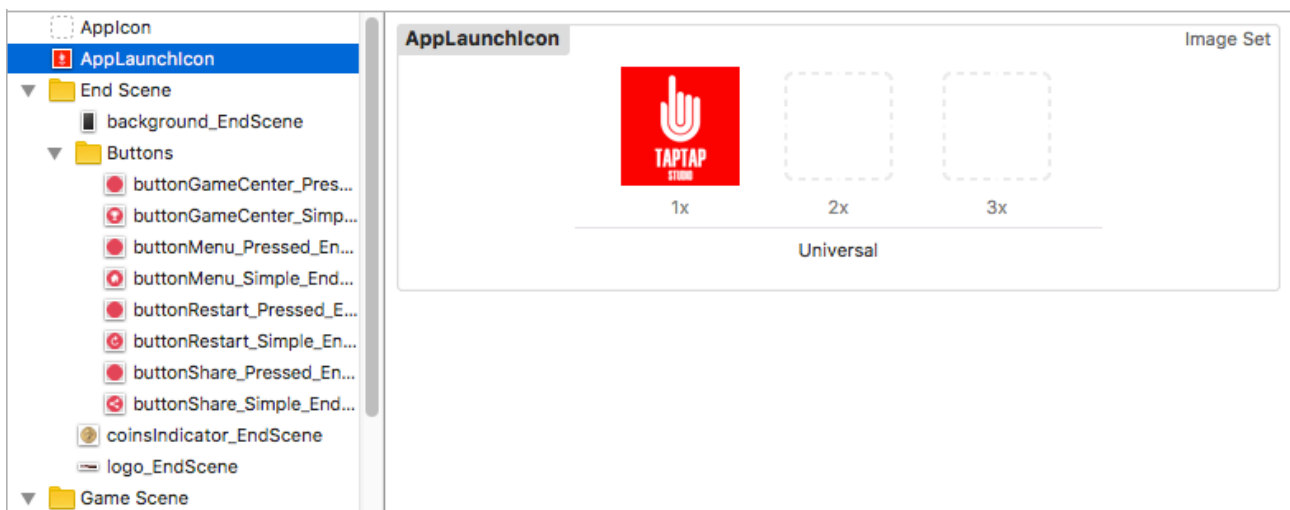
## 4. Edit launch screen.

### Change copyright text

Go to folder with views TapTapStudio -> Views.  
Open LaunchScreen.storyboard file.  
Here you can edit text field with copyright.

### Change logo image

Go to folder with resources TapTapStudio -> Resources.  
Open Assets.xcassets file.  
Here you can see AppLaunchIcon set.  
Replace image in this set.  
(If images is not changes, wait. Xcode have bug, it need time for showing new logo)



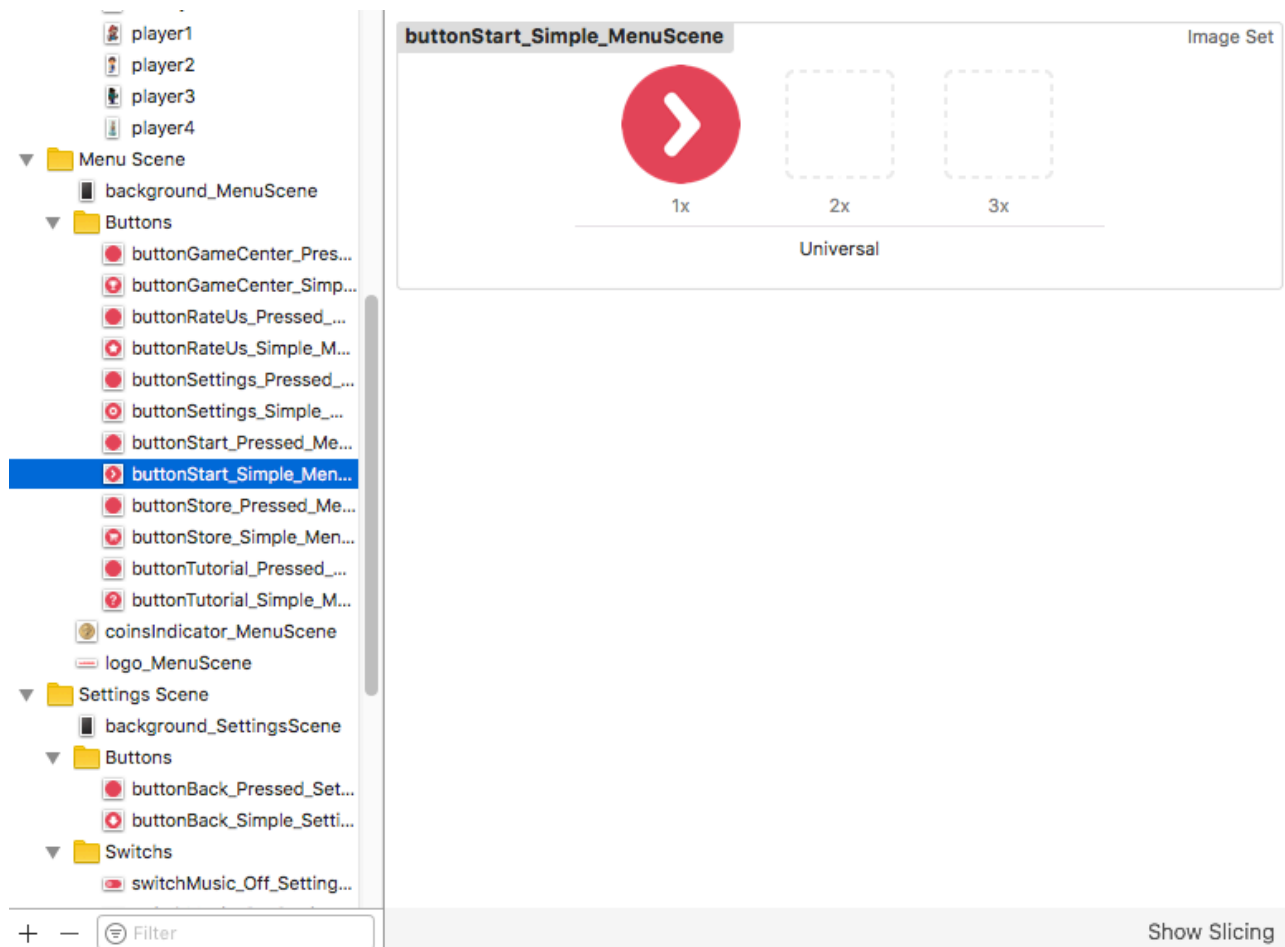
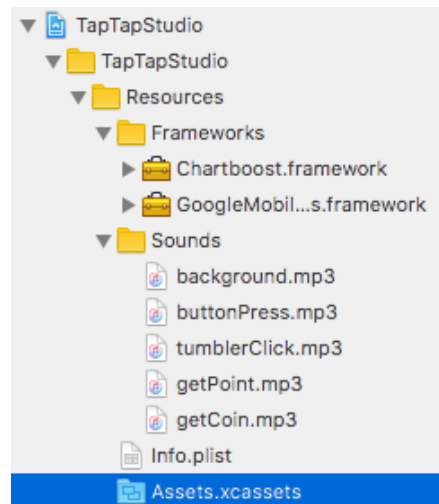
## 5. Replace some image in project.

Go to folder with resources TapTapStudio -> Resources.

Open Assets.xcassets file.

Here you can see all graphics in project.

Select what you want and change to you image.



## 6. Change size or position some object in project.

Go to settings folder TapTapStudio->Classes->Settings

Open file SettingsScenes.swift if SettingsScenesIPad.swift if you want to change settings of object in iPad.

Change what you want

```
123
124 struct size {
125     static let background = CGSize.withPercent(100, height: 100)
126     static let logo = CGSize.withPercentScaled(80, height: 30)
127
128     static let coinsIndicator = CGSize.withPercentScaled(4, height: 4)
129
130     static let buttonStart = CGSize.withPercentScaled(roundByWidth: 32)
131     static let buttonRateUs = CGSize.withPercentScaled(roundByWidth: 11)
132     static let buttonStore = CGSize.withPercentScaled(roundByWidth: 11)
133     static let buttonTutorial = CGSize.withPercentScaled(roundByWidth: 11)
134     static let buttonGameCenter = CGSize.withPercentScaled(roundByWidth: 11)
135     static let buttonSettings = CGSize.withPercentScaled(roundByWidth: 11)
136 }
137
138 struct position {
139     static let background = CGPoint.withPercent(50, y: 50)
140     static let logo = CGPoint.withPercent(50, y: 70)
141
142     static let coinsIndicator = CGPoint.withPercent(46, y: 10)
143
144     static let buttonStart = CGPoint.withPercent(50, y: 39)
145     static let buttonRateUs = CGPoint.withPercent(26, y: 25)
146     static let buttonStore = CGPoint.withPercent(38, y: 25)
147     static let buttonTutorial = CGPoint.withPercent(50, y: 25)
148     static let buttonGameCenter = CGPoint.withPercent(62, y: 25)
149     static let buttonSettings = CGPoint.withPercent(74, y: 25)
150
151     static let labelBestScore = CGPoint.withPercent(50, y: 14.5)
152     static let labelCoins = CGPoint.withPercent(50, y: 10)
153 }
154
155 struct zPosition {
156     static let background: CGFloat = 0
157     static let logo: CGFloat = 1
158
159     static let coinsIndicator: CGFloat = 1
160
161     static let buttonStart: CGFloat = 2
162     static let buttonRateUs: CGFloat = 2
163     static let buttonStore: CGFloat = 2
```

## 7. Change speed or some another setting in game process.

Go to settings folder TapTapStudio->Classes->Settings.  
Open here file with name SettingsGameProcess.swift.  
Change what you want.

```
28 struct SettingsGameProcess {
29
30     //Player speeds
31     static let enemySpeed: Double = 1.0
32     static let enemySpeedMax: Double = 0.5
33     static let enemySpeedChangeByPercent: Double = 2.0
34     static let enemyTimeToGenerate: Double = 2.2
35     static let enemyTimeToGenerateMax: Double = 1.5
36     static let enemyTimeToGenerateChangeByPercent: Double = 2.0
37
38     static let playerSpeed: Double = 4.0
39     static let playerSpeedMax: Double = 2.0
40     static let playerSpeedChangeByPercent: Double = 2.0
41
42     static let timeToChangeSceneIfLose: Double = 0.75
43
44     //Global process
45
46     static let speedAnimationChangeLabel: Double = 0.2
47
48     /**
49      Store category for collisions, use in physic body on nodes.
50      */
51     struct CollisionCategory {
52         static let playerCategory: UInt32 = 0x1 << 0
53         static let enemyCategory: UInt32 = 0x1 << 1
54         static let coinCategory: UInt32 = 0x1 << 2
55     }
56
57     /**
58      Store value fo price all skins in game
59      */
60     struct SkinPrice {
61         static let skin2: Int = 50
62         static let skin3: Int = 50
63         static let skin4: Int = 50
64     }
```



## 8. Get random value in project.

In this game we have extensions for simple setting random values.  
You can use it everywhere in project.

Int(from: 0, to: 1) -> Return random int value, replace 0 and 1 to your values  
Double(from: 0, to: 1)  
Float(from: 0, to: 1)  
CGFloat(from: 0, to: 1)

## 9. Get UIColor from hex value.

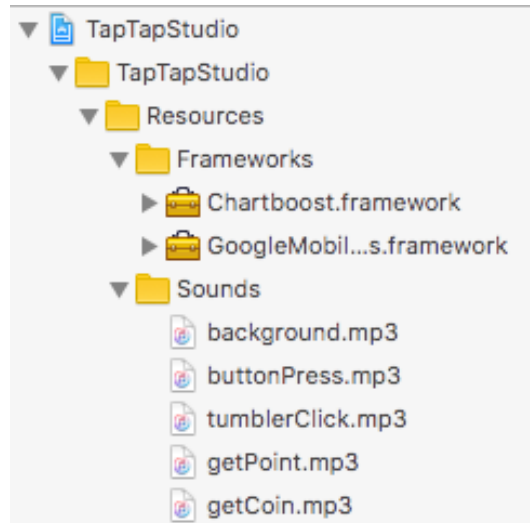
In this game we have extension for easy creating UIColor from hex value.

For use it just write.

`UIColor(hexString: «#000000»)`      *replase #000000 to your color hex code.*

## 10. Change sounds in project

Go to sounds folder TapTapStudio->Resources->Sounds.  
Replace what you want. (You need use .mp3 files).



## 11. Change links to rate us, share and other.

Go to settings folder TapTapStudio->Classes->Settings

Open SettingsGlobal.swift folder

Here you can find struct «SomeSocialStats», edit it.

```
/**
    Some social info.
 */
struct SomeSocialStats {
    static let shareText = "Try to beat my score!"
    static let linkToRateUs = "http://google.com/"
}

/**
```

## 12. Change In-App id.

Go to settings folder TapTapStudio->Classes->Settings

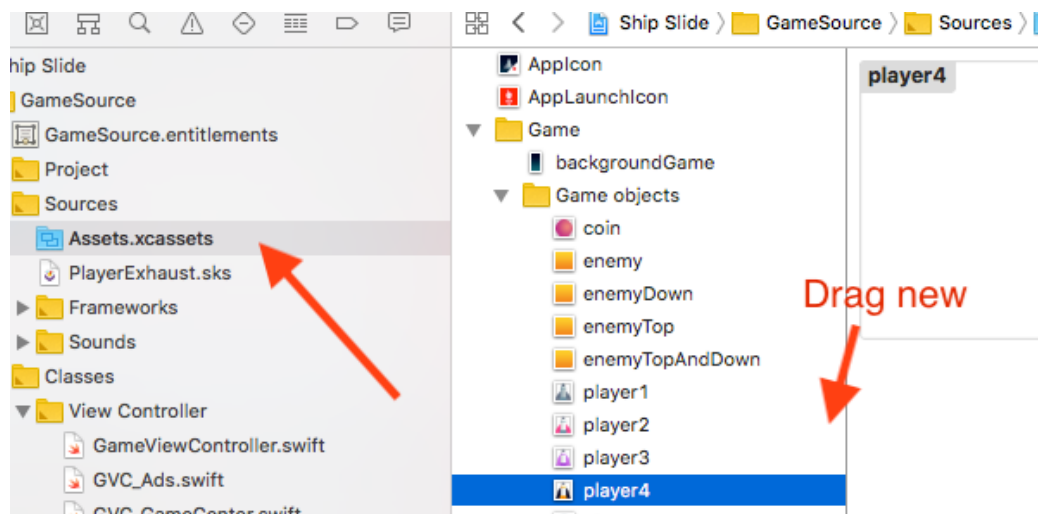
Open SettingsGlobal.swift folder

Here you can find struct «InApp», edit it.

```
//In App
struct InApp {
    static let on = true
    static let removingAds = "removeAdsIDs"
}
```

### 13. Add more skins.

## Add your skin image to Assets



Open GraphicPrevloads.swift file

Create new texture in GraphicPreloadsGame class for your image like this

```
internal let player1 = SKTexture(consideringFiltrationSettingsInProject: "player1")
internal let player3 = SKTexture(consideringFiltrationSettingsInProject: "player3")
internal let player4 = SKTexture(consideringFiltrationSettingsInProject: "player4")
internal let enemy = SKTexture(consideringFiltrationSettingsInProject: "enemy")
```

And this in GraphicPreloadInterface

```
internal let skin3 = SKTexture(consideringFiltrationSettingsInProject: "player3")
internal let skin4 = SKTexture(consideringFiltrationSettingsInProject: "player4")
```

Open SkinValue.swift file

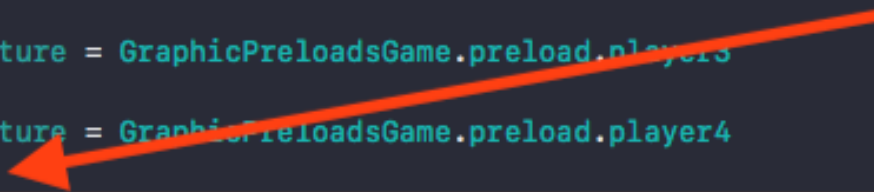
Here toy can attach new texture

Draw size of your skin (for store view)

And add price.

```
0 class SkinValues {  
1  
2     //MARK: - Values  
3  
4  
5     static var textures = [GraphicPreloadsInterface.preload.skin1,  
6                             GraphicPreloadsInterface.preload.skin2,  
7                             GraphicPreloadsInterface.preload.skin3,  
8                             GraphicPreloadsInterface.preload.skin4]  
9  
10    static var size = [CGSize.withPercentScaled(20, height: 20),  
11                        CGSize.withPercentScaled(20, height: 20),  
12                        CGSize.withPercentScaled(20, height: 20),  
13                        CGSize.withPercentScaled(20, height: 20)]  
14  
15    static var prices = [0,  
16                          50,  
17                          50,  
18                          50]
```

Now go to PlayerNode.swift  
And in function loadSkin() add new case for your new skin  
Like this.

```
fileprivate func loadSkin() {  
    if SettingsGlobal.Skins.on {  
        switch UserDefaults.standard.integer(forKey: "CurrentSkin") {  
            case 2:  
                texture = GraphicPreloadsGame.preload.player2  
            case 3:  
                texture = GraphicPreloadsGame.preload.player3  
            case 4:  
                texture = GraphicPreloadsGame.preload.player4  
            case 5:   
                texture = GraphicPreloadsGame.preload.player5  
            default:  
                texture = GraphicPreloadsGame.preload.player1  
        }  
    }  
}
```

Final! Launch your project!.