


Adstronomic – Installation Guide (iOS)

1 – Introduction

Adstronomic is an advertising platform that allows you to leverage and synthesize your users' data to deliver the most relevant advertising to them. By taking into account the specificities of each game and its users, coupled with a revolutionary AI, we can identify the most relevant ads to increase your game's revenues. To achieve this, Adstronomic offers three key tools :

- A web platform where you can set up your projects and associated ads.
- An API that allows you to interact with Adstronomic data.
- An SDK that allows you to quickly and easily use the full potential of Adstronomic

Each of these tools is intuitive, so you can focus on what matters most to you : The success of your project. In this guide, we will focus on the third point : Installing and using the SDK, here in its iOS version. 

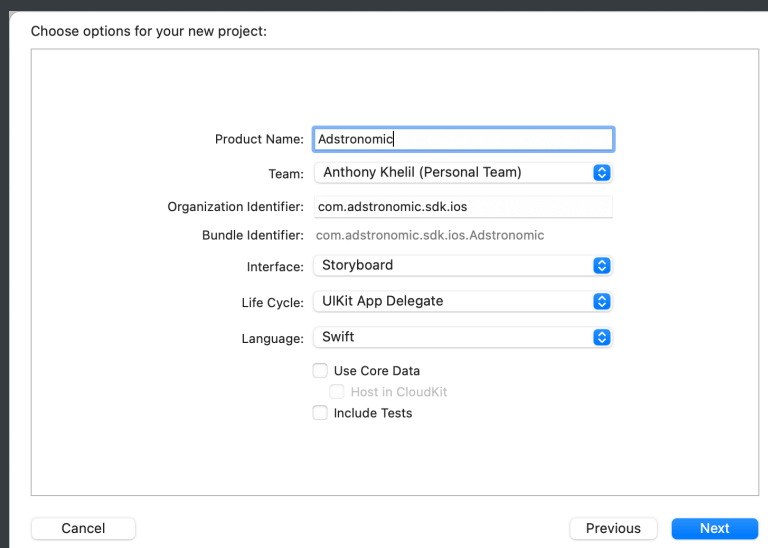
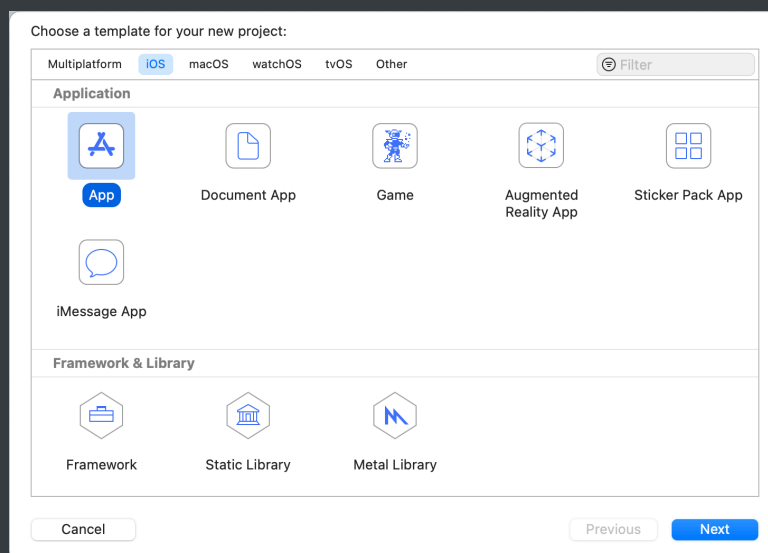
Please note that there are two solutions available to you : If you start a new project, we invite you to clone this repository directly, and use it as a working base. This will allow you to skip the section "2 - Creating a new project". However, if your project is already well advanced, you will probably prefer to install the SDK to your existing project. In this case,

the next section is for you !

2 – Creating a new project

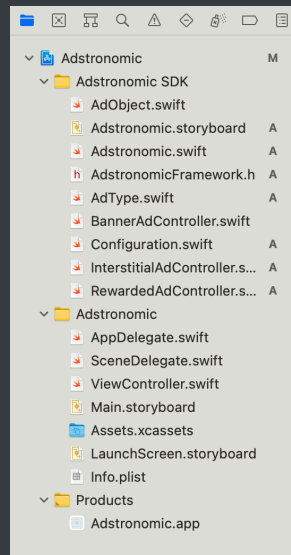
As an iOS developer, you are probably familiar with Xcode, Apple's default development tool. So let's launch it, and create a new project.

Start by choosing "iOS" in the top menu, then select "App". You can use any name and Organization Identifier you want. However, make sure you have selected the "Storyboard" interface, the "UIKit App Delegate" life cycle and the "Swift" language.



As you are starting from an empty project, you will have to add the SDK manually. To do this, clone this project, find the "Adstronomic SDK" folder, and copy its contents, which are nine files. Go back to Xcode, and create a new group (I kept the name "Adstronomic SDK", but you can rename it) in your project, into which you paste these nine files. Note that copy and paste may not work in some cases, and you will have to replace it with drag and drop.

You should normally have a tree structure like this :



Congratulations ! You've just added Adstronomic to your project ! 🎉 The next step is now to configure it !

3 – SDK configuration

Now that Adstronomic is integrated into your project, let's see how to set it up so that it retrieves and sends the right data.

As a reminder, we are in a storyboard, and the entry point of our project is the "ViewController.swift" file. So we will open it. You should have the following content :

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
    }

}
```

Comments are removed from the examples I will give, for the sake of clarity.

The first step is to initialize Adstronomic. To do this, we will use the following function in the constructor :

```
Adstronomic.initialize(campaignId: "01234567-89AB-CDEF-0123-456789ABCDEF")
```

Please note that the campaignId is your campaign ID, as indicated on the web platform. If you use this example ID, no ads will be loaded.

The rest is just as simple. Once Adstronomic is initialized, you need to load the following function, which will retrieve the metadata of your ads :

```
Adstronomic.loadAdsData()
```

Then, we will need three controllers, for the three types of usable ads. To do this, we will add, just before the constructor, the following lines :

```
var bannerAdController: BannerAdController?
var interstitialAdController: InterstitialAdController?
var rewardedAdController: RewardedAdController?
```

Once these attributes are defined, we just have to initialize them in the constructor, with :

```
bannerAdController = BannerAdController.initializeObject()
interstitialAdController = InterstitialAdController.initializeObject()
rewardedAdController = RewardedAdController.initializeObject()
```

And that's it, you're done configuring Adstronomic. To make sure we're on the same page for the next part, you should have the following code in front of you :

```
import UIKit

class ViewController: UIViewController {

    var bannerAdController: BannerAdController?
    var interstitialAdController: InterstitialAdController?
    var rewardedAdController: RewardedAdController?

    override func viewDidLoad() {
        super.viewDidLoad()

        Adstronomic.initialize(campaignId: "01234567-89AB-CDEF-0123-456789ABCDEF")
        Adstronomic.loadAdsData()

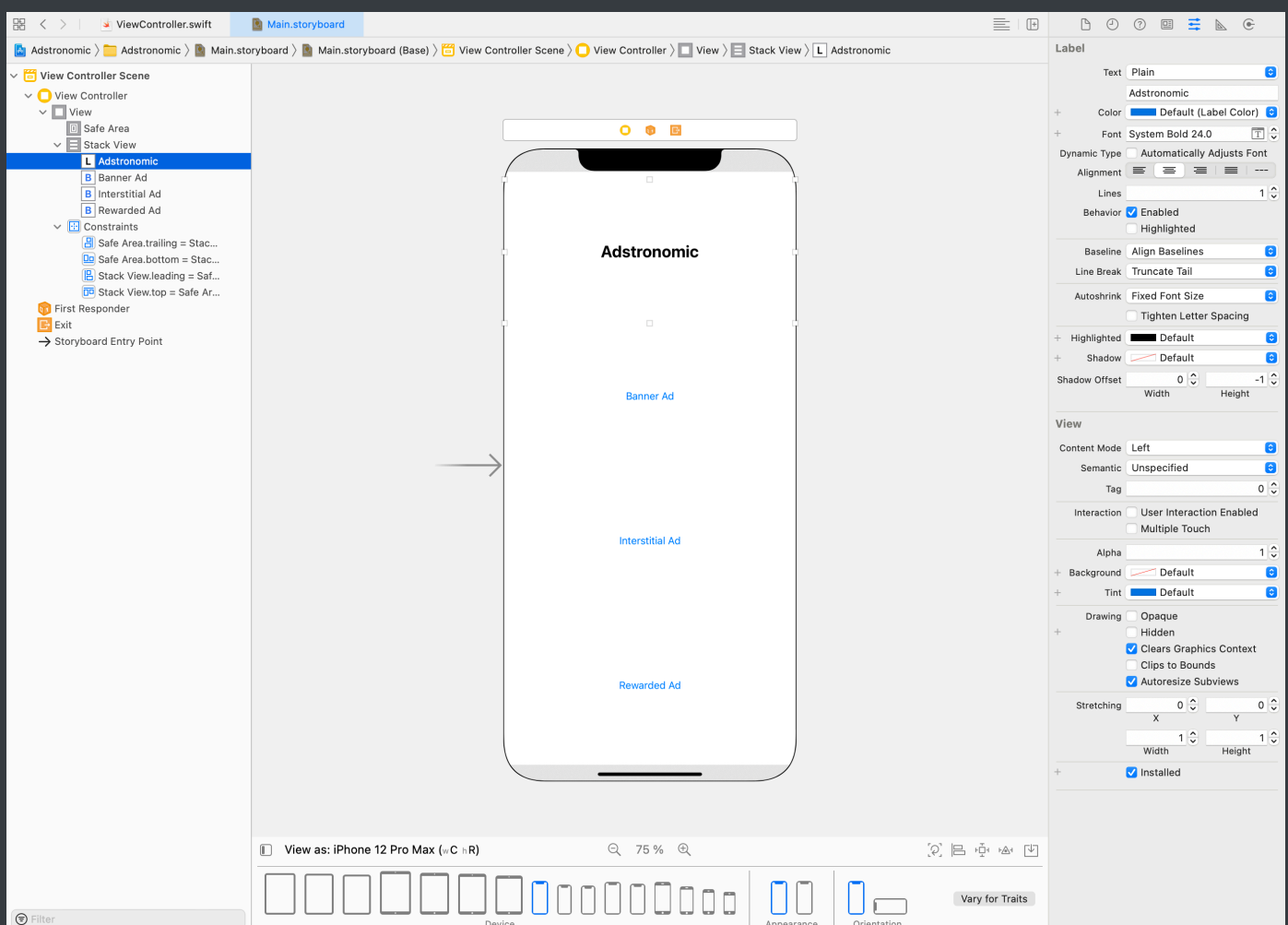
        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }

}
```

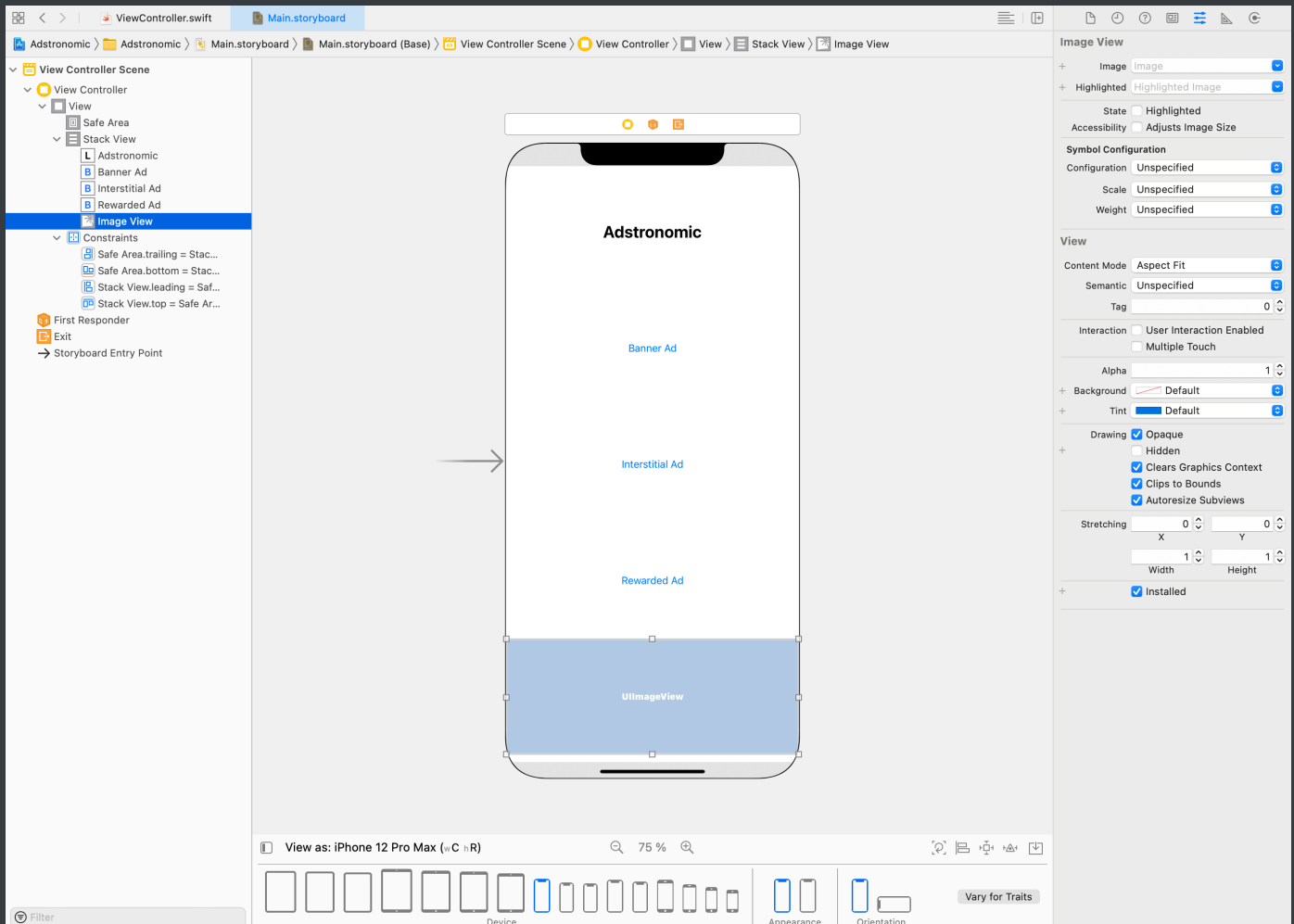
4 – Loading a Banner Ad

A BannerAd is simply an advertisement in the form of an image, usually displayed at the bottom of the screen during a game. The advantage of this type of display is that it does not block the rest of the screen, and can therefore be displayed throughout the game.

In Adstronomic, these ads are represented by classic images, of type UIImageView. So we will manually add one to our project. To do this, open the "Main.storyboard" file.



If you open this file for the first time, you should normally see a blank screen. This is normal ! I have added a StackView to my project containing a Label, and three Buttons, but they are optional at this stage. We will only need to add a UIImageView, like this:



Now that we have our UIImageView on our stage, we just need to retrieve it from the ViewController.swift file, by dragging it. Once this is done, your ad is ready to be loaded.

The only step is to call the function :

```
bannerAdController?.showInComponent(uiImageView: bannerAdComponent)
```

This function takes as parameter a UIImageView, which is the one we have just retrieved.

Where to call this function ? Well, that's up to you. You can load it from the beginning, so that your ad appears as soon as it is launched, or wait for a specific action. As we are on a test project here, I made sure to call it when a button is clicked, which is present in the storyboard we just saw. But you can call it at any event of your application.

```
import UIKit
```

```

class ViewController: UIViewController {

    var bannerAdController: BannerAdController?
    var interstitialAdController: InterstitialAdController?
    var rewardedAdController: RewardedAdController?

    @IBOutlet weak var bannerAdResult: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

        Adstronomic.initialize(campaignId: "DB901CE6-07C7-499D-B4DA-
4CD34CD82C45")
        Adstronomic.loadAdsData()

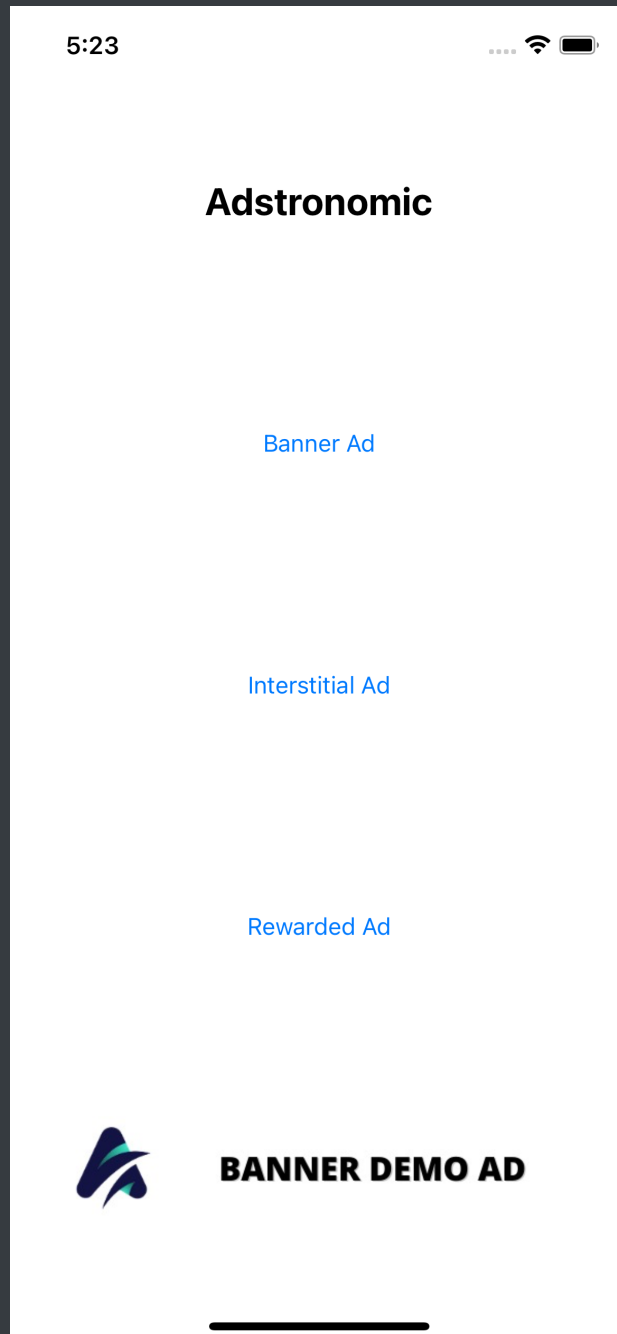
        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }

    @IBAction func startBannerAd(_ sender: Any) {
        bannerAdController?.showInComponent(uiImageView: bannerAdResult)
    }

}

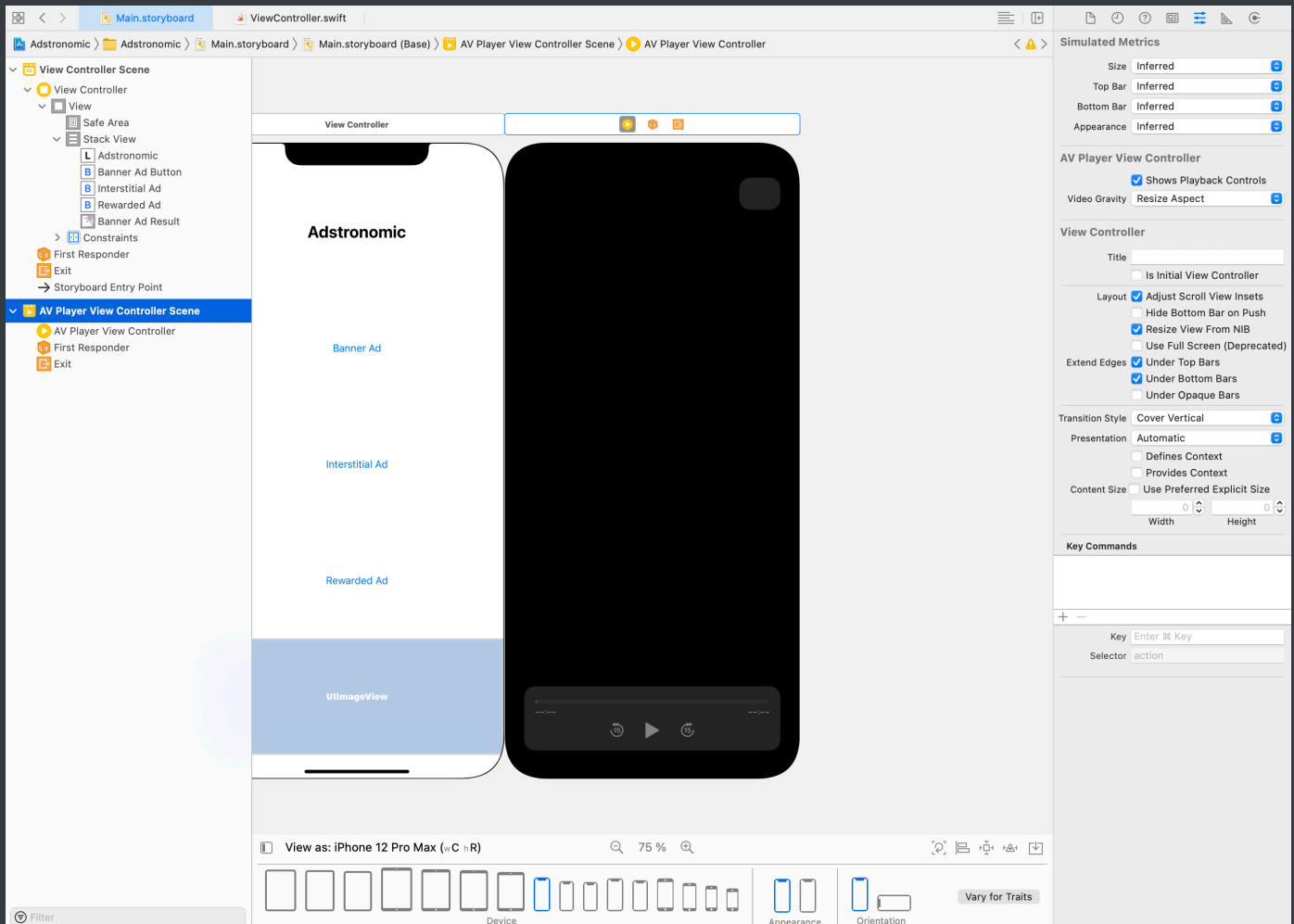
```

We can now launch our application, and see the result. When we click on the button, the ad is loaded, and is displayed at the bottom of our application.



5 – Loading an Interstitial Ad

If you have successfully displayed a Banner Ad, you should have no difficulty in displaying Interstitial Ad and Rewarded Ad, as they are used almost the same way. Let's go back to the Main.storyboard, and this time add an AVKit Player View Controller. This is the equivalent of a UIImageView, for videos.



Note that this is a bit of a special component, as it will take up all the space of your scene, and should therefore be placed next to your scene, so as not to hide it.

The difference with the Banner Ad is that once our AVKit Player View Controller is added, we don't have to take care of it anymore. In fact, the Adstronomic SDK will automatically pick it up when we load our video ad.

So how do we do this ? By adding the following lines :

```
if let interstitialAdResult = interstitialAdController?.returnPlayer() {
    present(interstitialAdResult, animated: true) {
        interstitialAdResult.player?.play()
    }
}
```

The first line retrieves the video's context, preloaded by Adstronomic. The next two lines launch the video, and display it in full screen.

Two remarks about this code. First of all, it is possible to split it in two, between the condition and the present function call, in order to delay the video launch. This is useful if you want to preload your video ad at the beginning of the game, for example, to launch it without any latency afterwards.

Second point, as for the Banner Ad, you can launch the video at any time. I did it here when I clicked on a new button, but you can do it as soon as the application is launched, or at any other event.

```
import UIKit

class ViewController: UIViewController {

    var bannerAdController: BannerAdController?
    var interstitialAdController: InterstitialAdController?
    var rewardedAdController: RewardedAdController?

    @IBOutlet weak var bannerAdResult: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

        Adstronomic.initialize(campaignId: "DB901CE6-07C7-499D-B4DA-4CD34CD82C45")
        Adstronomic.loadAdsData()

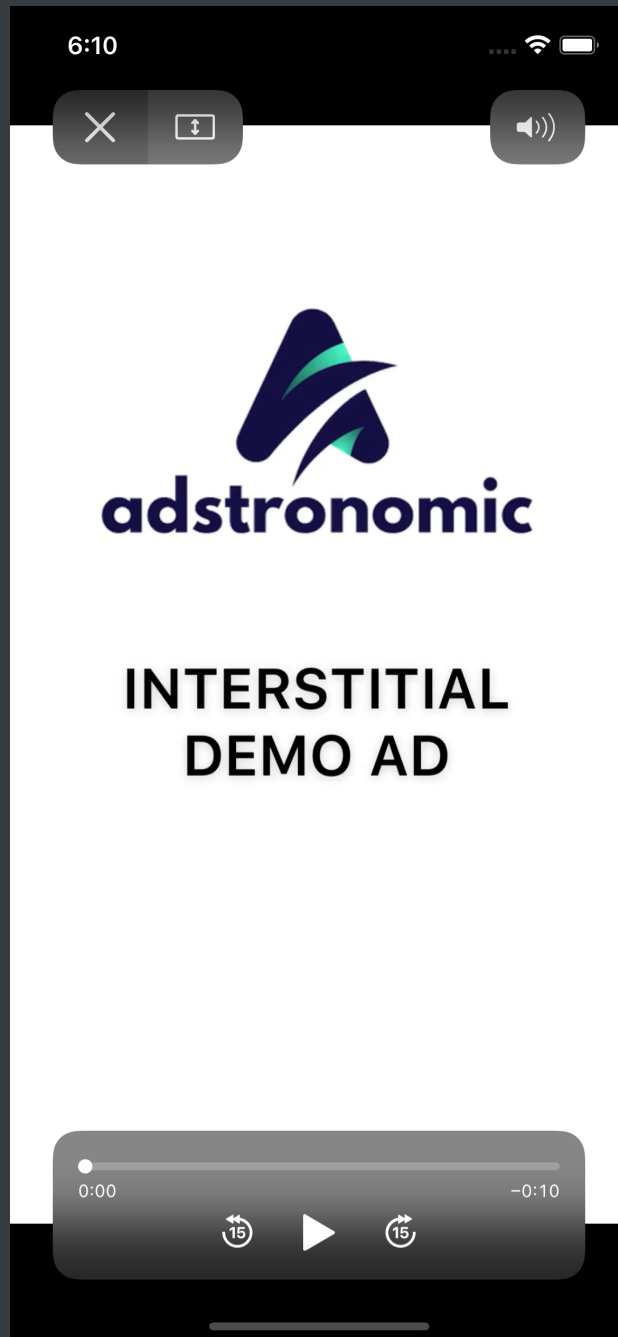
        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
        InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }

    @IBAction func startBannerAd(_ sender: Any) {
```

```
        bannerAdController?.showInComponent(uiImageView: bannerAdResult)
    }

    @IBAction func startInterstitialAd(_ sender: Any) {
        if let interstitialAdResult =
        interstitialAdController?.returnPlayer() {
            present(interstitialAdResult, animated: true) {
                interstitialAdResult.player?.play()
            }
        }
    }
}
```

We use the second button in the storyboard, and intercept its click in order to execute the few lines I indicated just before. By launching the application, and clicking on the corresponding button, our video ad should appear in full screen.



6 – Loading a Rewarded Ad

Ready for the final step ?

Rewarded Ads are video ads similar to Interstitial Ads. The difference is that they have a different marketing goal, and are more about rewarding a specific action. Technically speaking, they work the same way, except that you have to create the video from `rewardedAdController` and not `interstitialAdController`.

```
if let rewardedAdResult = rewardedAdController?.returnPlayer() {
    present(rewardedAdResult, animated: true) {
        rewardedAdResult.player?.play()
    }
}
```

As for the first two ads, this code can be executed at any time in your application, and the loading of the Rewarded Ad can be done before the launch of the video, as explained for the Interstitials Ads.

Here is the complete code that loads and launches a Rewarded Ad as soon as the third and last button of the storyboard is clicked :

```
import UIKit

class ViewController: UIViewController {

    var bannerAdController: BannerAdController?
    var interstitialAdController: InterstitialAdController?
    var rewardedAdController: RewardedAdController?

    @IBOutlet weak var bannerAdResult: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

        Adstronomic.initialize(campaignId: "DB901CE6-07C7-499D-B4DA-4CD34CD82C45")
        Adstronomic.loadAdsData()
```

```

        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }

    @IBAction func startBannerAd(_ sender: Any) {
        bannerAdController?.showInComponent(uiImageView: bannerAdResult)
    }

    @IBAction func startInterstitialAd(_ sender: Any) {
        if let interstitialAdResult =
interstitialAdController?.returnPlayer() {
            present(interstitialAdResult, animated: true) {
                interstitialAdResult.player?.play()
            }
        }
    }

    @IBAction func startRewardedAd(_ sender: Any) {
        if let rewardedAdResult = rewardedAdController?.returnPlayer() {
            present(rewardedAdResult, animated: true) {
                rewardedAdResult.player?.play()
            }
        }
    }
}

```

That's it ! You are now an Adstronomic pro ! 😎

All that's left to do is create a campaign [on our web platform](#), add your ads, and specify your campaignId when the app loads.

And if you need to contact us, you can do so [here](#) !