


# Adstronomic – Guide d'installation (iOS)

## 1 – Introduction

Adstronomic est une plateforme publicitaire vous permettant d'exploiter et de synthétiser les données de vos utilisateurs, afin de leur proposer la publicité la plus adaptée à leurs besoins. En nous appuyant sur les spécificités de chaque jeu et de ses utilisateurs, couplé à une IA révolutionnaire, nous parvenons à identifier les publicités les plus pertinentes, afin d'augmenter les revenus de votre jeu. Pour y parvenir, Adstronomic met à votre disposition trois outils clé :

- Une plateforme web sur laquelle vous pouvez paramétrer vos projets, et les publicités associées.
- Une API permettant d'interagir avec les données d'Adstronomic.
- Un SDK qui vous permet d'utiliser facilement et rapidement tout le potentiel d'Adstronomic

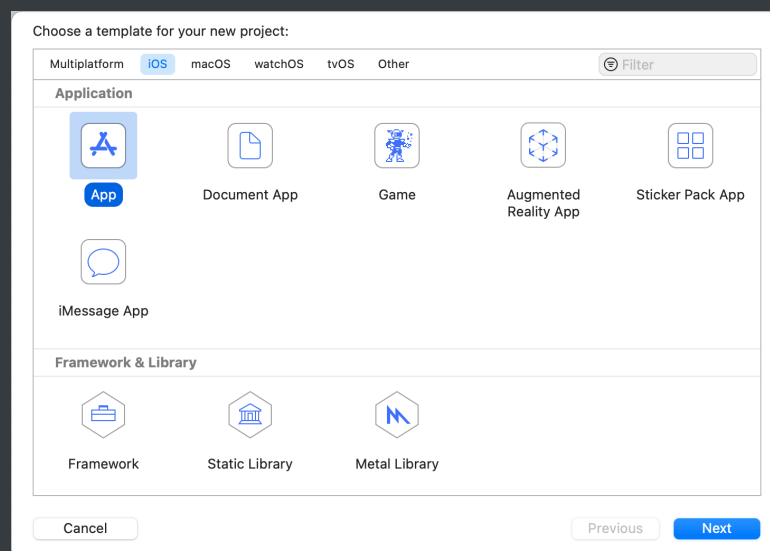
Chacun de ces outil est intuitif, afin de vous permettre de vous concentrer sur ce qui compte le plus pour vous : La réussite de votre projet. Dans ce guide, nous allons nous concentrer sur le troisième point : L'installation et l'utilisation du SDK, ici dans sa version iOS. 

Veillez noter que deux solutions s'offrent à vous : Si vous commencez un nouveau projet, nous vous invitons à clôner directement ce dépôt, et à l'utiliser comme base de travail. Vous pourrez ainsi sauter la section "2 - Création d'un nouveau projet". Toutefois, si votre projet est déjà bien avancé, vous préférerez sûrement installer le SDK à votre projet existant. Dans ce cas là, la section suivante est faite pour vous !

## 2 – Création d'un nouveau projet

En tant que développeur iOS, vous êtes sûrement familier avec Xcode, l'outil de développement par défaut d'Apple. Nous allons donc le lancer, et créer un nouveau projet.

Commencez par choisir "iOS" dans le menu supérieur, puis sélectionnez "App". Vous pouvez utiliser le nom et l'Organization Identifier que vous voulez. Par contre, assurez vous d'avoir sélectionné l'interface "Storyboard", le life cycle "UIKit App Delegate" et le langage "Swift".



Choose options for your new project:

Product Name:

Team:

Organization Identifier:

Bundle Identifier:

Interface:

Life Cycle:

Language:

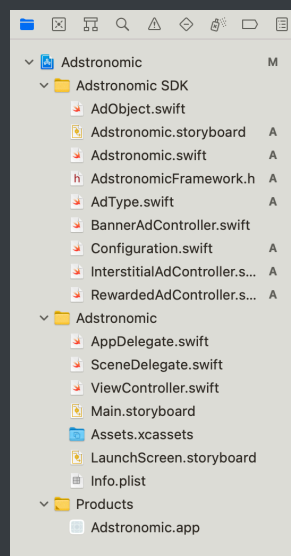
☐ Use Core Data

☐ Host in CloudKit

☐ Include Tests

Comme vous partez d'un projet vide, vous allez devoir ajouter manuellement le SDK. Pour cela, clônez ce projet, cherchez le dossier "Adstronomic SDK", et copiez son contenu, soit neuf fichiers. Retournez sur Xcode, et créez un nouveau groupe (J'ai gardé le nom "Adstronomic SDK", mais vous pouvez le renommer) dans votre projet, dans lequel vous collez ces neuf fichiers. Notez que le copier-coller peut ne pas marcher dans certains cas, et vous devrez alors le remplacer par un glisser-déposer.

Vous devriez normalement avoir une arborescence comme ceci :



Félicitations ! Vous venez d'ajouter Adstronomic à votre projet ! 🎉 La prochaine étape est maintenant de le configurer !

## 3 – Configuration du SDK

Maintenant qu'Adstronomic est intégré à votre projet, nous allons voir comment le paramétrer pour qu'il récupère et envoie les bonnes données.

Pour rappel, nous sommes dans un storyboard, et le point d'entrée de notre projet est le fichier "ViewController.swift". Nous allons donc l'ouvrir. Vous devriez avoir le contenu suivant :

```
import UIKit

class ViewController: UIViewController {

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

}
```

Les commentaires sont enlevés des exemples que je donnerai, par soucis de clarté.

La première étape est d'initialiser Adstronomic. Pour cela, nous allons utiliser la fonction suivante, dans le constructeur :

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

Notez bien que le campaignId correspond à l'identifiant de votre campagne, tel qu'indiqué sur la plateforme web. Si vous utilisez cet identifiant d'exemple, aucune publicité ne sera chargée.

La suite est tout aussi simple. Une fois Adstronomic initialisé, vous devez charger la fonction suivante, qui va récupérer les métadonnées de vos publicités :

```
Adstronomic.loadAdsData()
```

Ensuite, nous allons avoir besoin de trois controllers, pour les trois types de publicités utilisables. Pour cela, nous allons rajouter, juste avant le constructeur, les lignes suivantes :

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

Une fois ces attributs définis, il nous reste juste à les initialiser dans le constructeur, avec :

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

Et voilà, vous avez terminé de configurer Adstronomic. Pour être sûrs que nous soyons au même point pour la suite, vous devriez normalement avoir sous les yeux le code suivant :

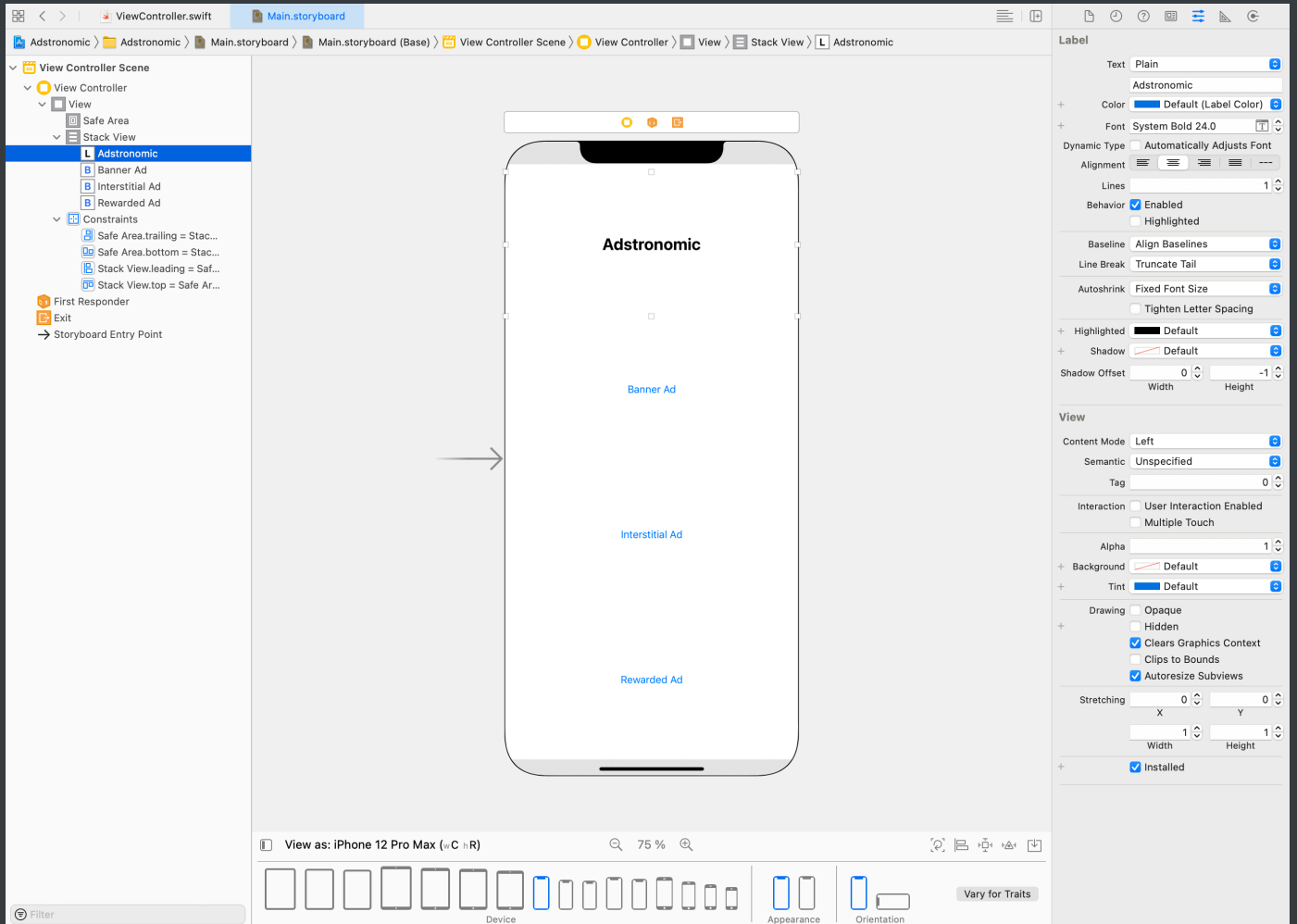
```
import UIKit  
  
class ViewController: UIViewController {  
  
    var bannerAdController: BannerAdController?  
    var interstitialAdController: InterstitialAdController?  
    var rewardedAdController: RewardedAdController?  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        Adstronomic.initialize(campaignId: "01234567-89AB-CDEF-0123-  
456789ABCDEF")  
        Adstronomic.loadAdsData()  
  
        bannerAdController = BannerAdController.initializeObject()
```

```
        interstitialAdController =  
        InterstitialAdController.initializeObject()  
        rewardedAdController = RewardedAdController.initializeObject()  
    }  
  
}
```

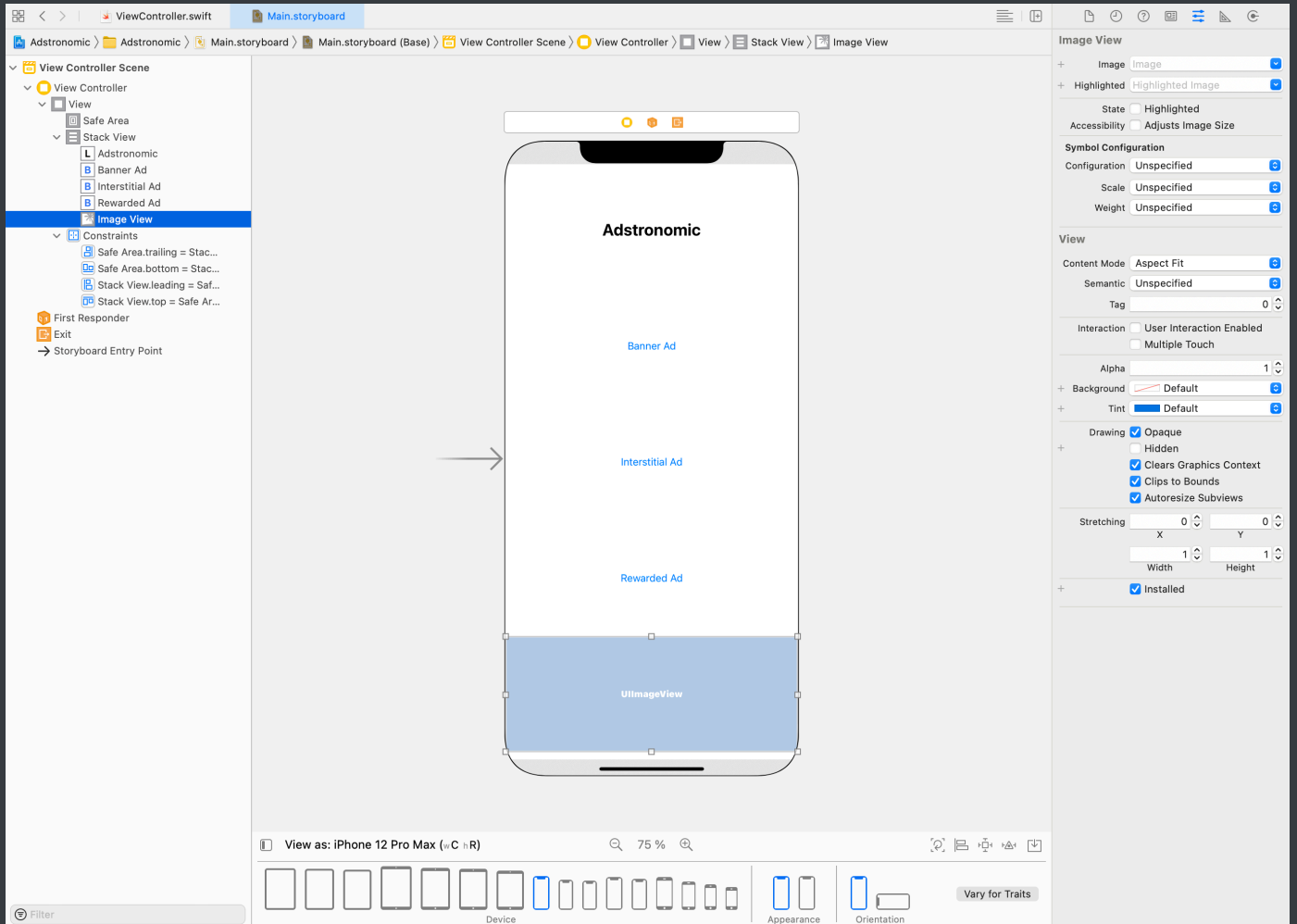
## 4 – Chargement d'une Banner Ad

Une BannerAd est tout simplement une publicité sous forme d'image, habituellement affichée au bas de l'écran pendant une partie. L'intérêt de ce type d'affichage est qu'elle ne bloque pas le reste de l'écran, et peut donc être affichée pendant toute la partie.

Dans Adstronomic, ces publicités sont représentées par des images classiques, de type UIImageView. Nous allons donc manuellement en ajouter une à notre projet. Pour cela, ouvrez le fichier "Main.storyboard".



Si vous ouvrez ce fichier pour la première fois, vous devriez normalement voir un écran blanc. C'est normal ! J'ai ajouté sur mon projet une StackView contenant un Label, et trois Button, mais qui sont facultatifs à cet étape. Nous allons avoir seulement besoin d'ajouter un UIImageView, comme ceci :



Maintenant que nous avons notre UIImageView sur notre scène, il nous reste juste à la récupérer dans le fichier ViewController.swift, en la faisant glisser. Une fois cela fait, votre publicité est prête à être chargée.

La seule étape pour cela est d'appeler la fonction :

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

Cette fonction prends en paramètre une UIImageView, qui est celle que nous avons récupérée à l'instant.

Où appeler cette fonction ? Et bien, c'est à vous de voir. Vous pouvez la charger dès le début, afin que votre publicité apparaisse dès le lancement, ou attendre une action spécifique. Comme nous sommes ici sur un projet de test, j'ai fait en sorte de l'appeler au clic sur un bouton, présent dans le storyboard que nous venons de voir. Mais vous pouvez l'appeler à n'importe quel événement de votre 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()

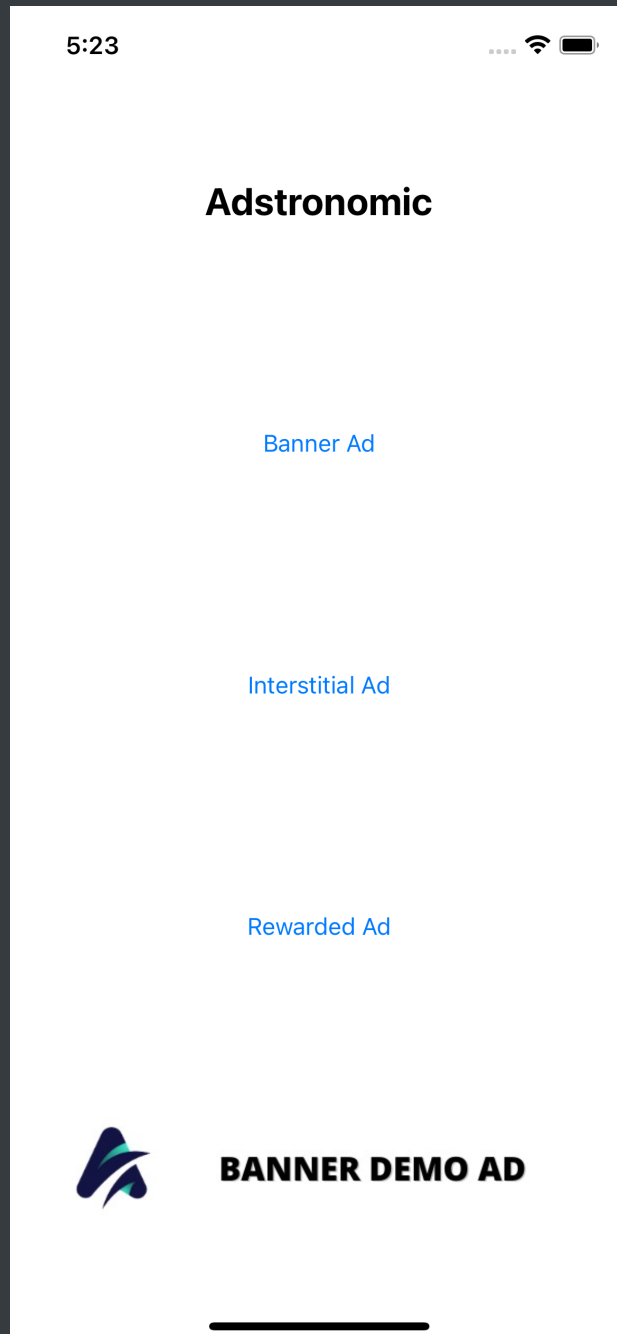
        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }

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

}

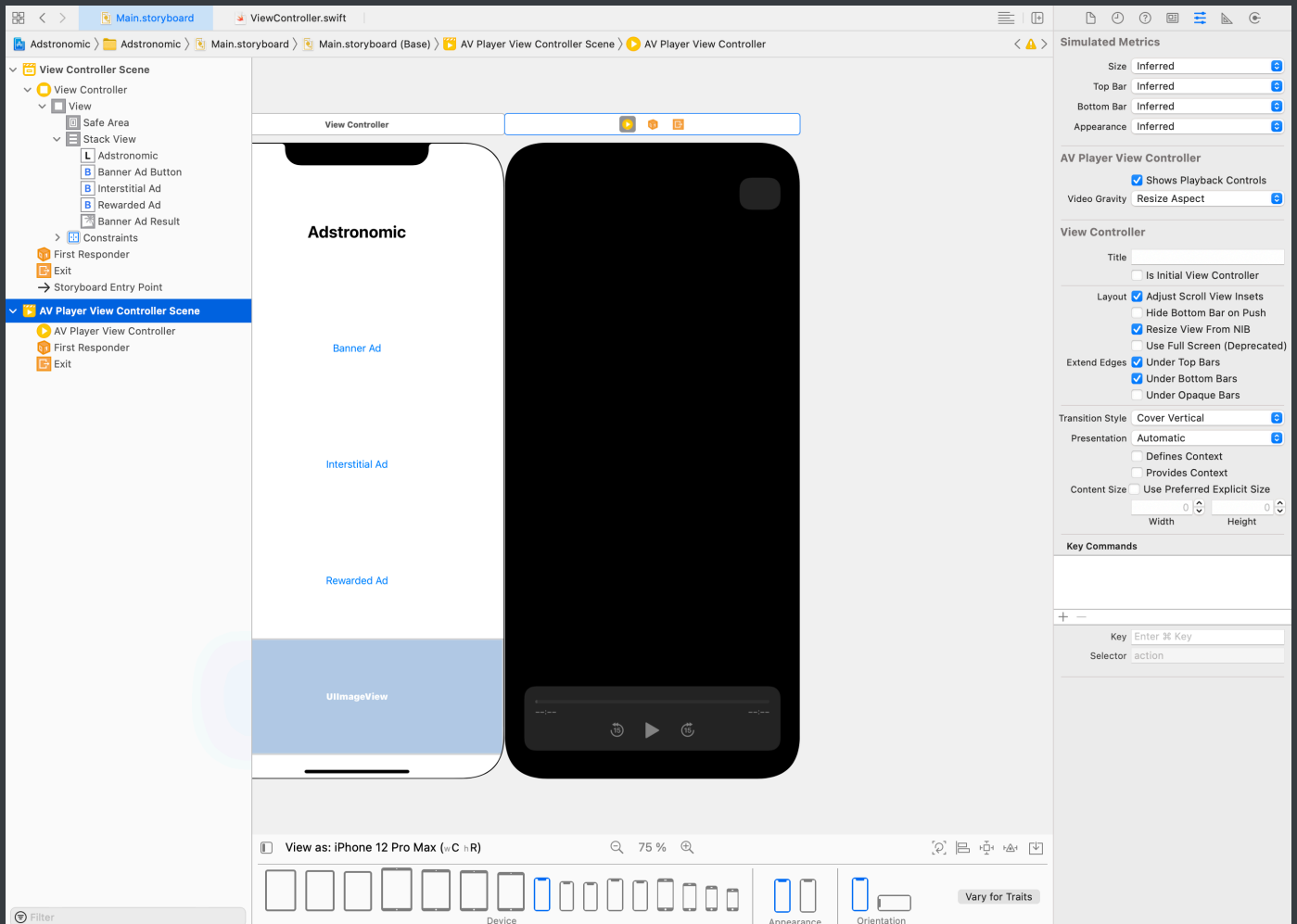
```

Nous pouvons maintenant lancer notre application, et voir le résultat. Au clic sur le bouton, la publicité est chargée, et s'affiche en bas de notre application.



## 5 – Chargement d'une Interstitial Ad

Si vous avez réussi à afficher une Banner Ad, vous ne devriez pas avoir de difficulté pour l'affichage des Interstitial Ad et des Rewarded Ad, car elles s'utilisent presque de la même façon. Retournons dans le Main.storyboard, et ajoutons-y cette fois un AVKit Player View Controller. Il s'agit de l'équivalent d'une UIImageView, pour les vidéos.



Notez qu'il s'agit d'un composant un peu particulier, car il prendra tout la place de votre scène, et devra donc être mis à côté de votre scène, pour ne pas la cacher.

La différence avec la Banner Ad est qu'une fois notre AVKit Player View Controller ajouté, nous n'avons plus à nous en occuper. En effet, le SDK Adstronomic le récupèrera automatiquement au moment de charger notre publicité vidéo.

Et justement, comment faire ce chargement ? En ajoutant les quelques lignes suivantes :

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

La première ligne se charge de récupérer le contexte de la vidéo, préchargé par Adstronomic. Les deux lignes suivantes permettent le lancement de la vidéo, et son affichage en plein écran.

Deux remarques par rapport à ce code. Tout d'abord, il est possible de le séparer en deux, entre la condition et l'appel de la fonction present, afin de retarder le lancement de la vidéo. C'est utile si vous souhaitez par exemple précharger votre publicité vidéo en début de partie, pour la lancer sans aucune latence par la suite.

Deuxième point, comme pour la Banner Ad, vous pouvez lancer la vidéo à n'importe quel moment. Je l'ai fait ici lors du clic sur un nouveau bouton, mais vous pouvez le faire dès le lancement de l'application, ou à n'importe quel autre évènement.

```
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()

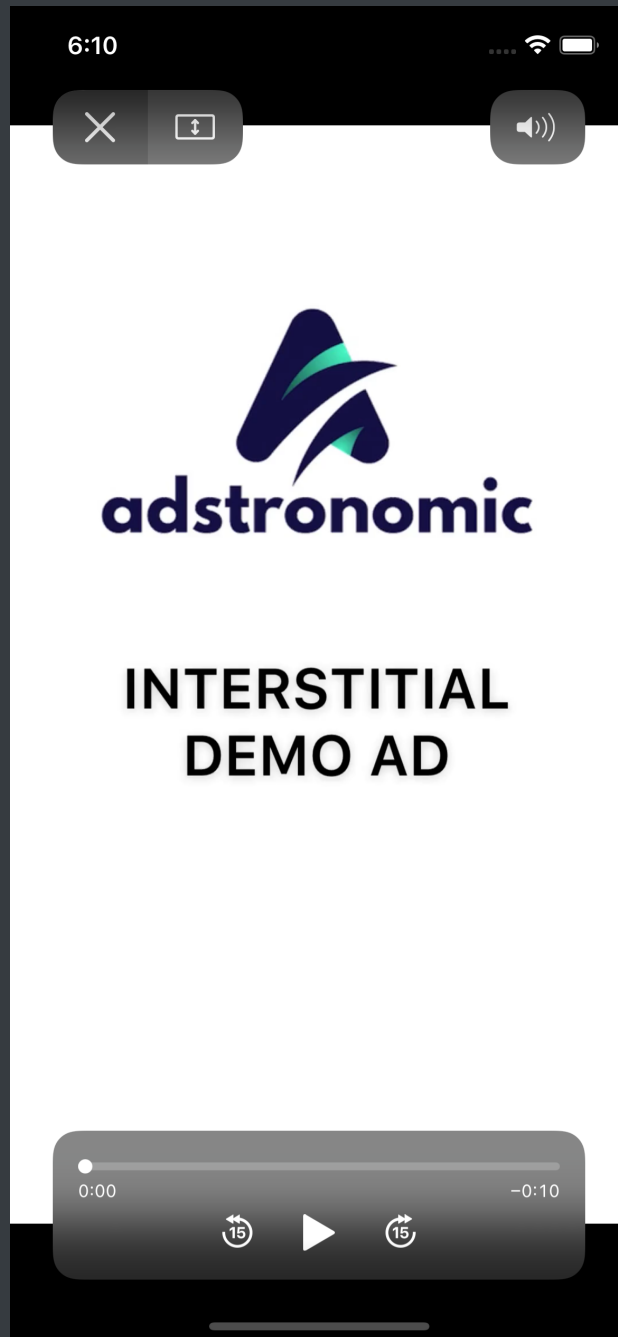
        bannerAdController = BannerAdController.initializeObject()
        interstitialAdController =
        InterstitialAdController.initializeObject()
        rewardedAdController = RewardedAdController.initializeObject()
    }
}
```

```
@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()
        }
    }
}

}
```

Nous utilisons le second bouton présent dans le storyboard, et interceptons son clic afin d'exécuter les quelques lignes que j'ai indiquées juste avant. En lançant l'application, et en cliquant sur le bouton correspondant, notre publicité vidéo devrait apparaître en plein écran.



## 6 – Chargement d'une Rewarded Ad

Prêts pour la dernière étape ?

Les Rewarded Ad sont des publicités vidéo semblables aux Interstitial Ad. La différence est que celles-ci ont un but marketing différent, et visent plus à récompenser une action spécifique. Techniquement parlant, leur fonctionnement est identique, sauf qu'il faut créer la vidéo depuis rewardedAdController et non interstitialAdController.

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

Comme pour les deux premières publicités, ce code peut être exécuté à n'importe quel moment de votre application, et le chargement des Rewarded Ad peut être fait en amont du lancement de la vidéo, comme expliqué pour les Interstitials Ads.

Voici le code complet qui charge et lance une Rewarded Ad dès que l'on clique sur le troisième et dernier bouton du storyboard :

```
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()
            }
        }
    }
}

```

Et voilà ! Vous êtes maintenant un pro d'Adstronomic ! 😎

Il ne vous reste plus qu'à créer une campagne sur notre plateforme web, y ajouter vos publicités, et indiquez votre campagneId au chargement de l'application.

Et si vous avez besoin de nous contacter, vous pouvez le faire ici !