

# Advertisements

Documentation | 22-08-22





# Table of Contents

1. Introduction	3
2. Assembly Definition	4
3. Set-Up	5
Unity Ads	5
AdMob	5
4. Editor	6
5. API	7
MyClass	7
6. Known Limitations	8
7. Support and feedback	9



# 1. Introduction

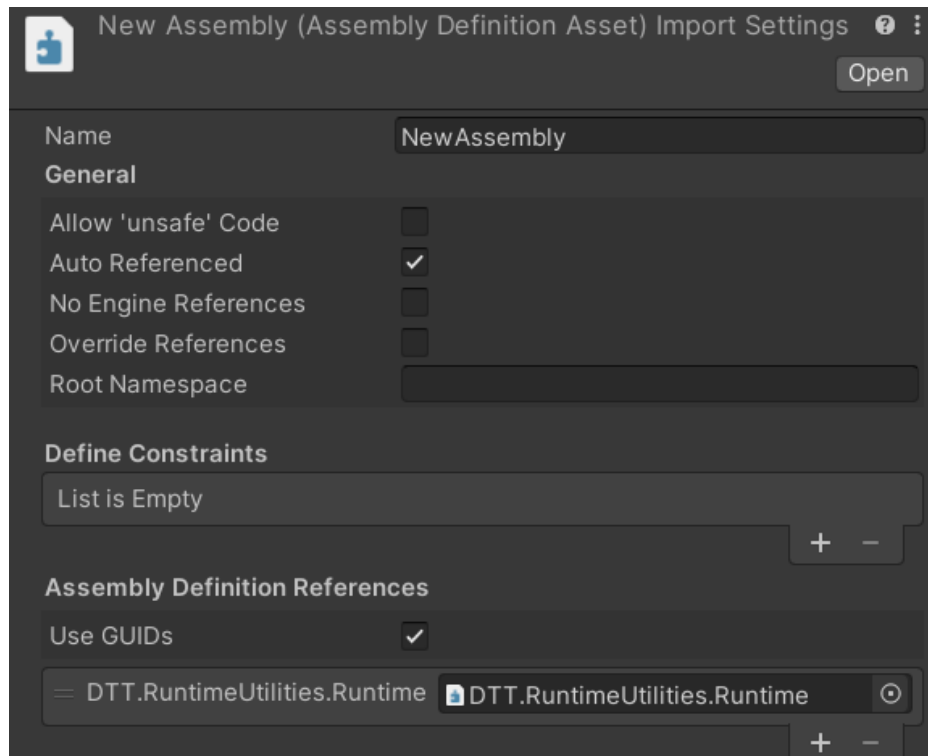
The minigame base package has basic implementations for both Unity Ads and AdMob. With the included components, you can implement ads into your app by simply dragging components into your scene. These components allow you to easily control when they should be opened, and what happens when they are clicked or closed.



## 2. Assembly Definition

Our assets use [assembly definitions](#) to organize the code into separate libraries. It helps with compiling, since code inside of assembly definitions don't have to be recompiled when changes are made to scripts in your project.

Each of our assets are separated in up to 3 different assemblies. These assembly definitions can be found in the Runtime, Editor, and Demo folder of the asset.



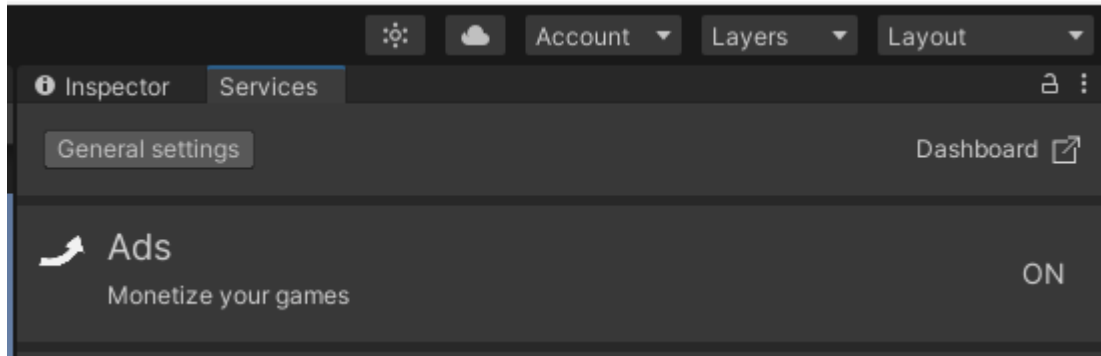
All of the scripts needed for the functionalities of our assets are contained in the Runtime assembly definition. If you are using assembly definitions in your project, make sure to add a reference to the assembly definition of the Runtime folder to use the asset in your code.



## 3. Set-Up

### Unity Ads

1. To get started, set up your Unity project to support Unity ads.



You can set up your project by clicking the cloud icon on the top right of your project and clicking on Ads in the services window. This will show you how to complete the setup.

2. After enabling Unity ads, you can go to the 'Runtime/Ads/UnityAds' folder and add the **UnityAdsInitializer** to your scene. This component allows you to initialize Unity ads for every play session. You will need to add your Android game ID and iOS game ID into the appropriate fields on this component. The ID's can be found on your project dashboard.
3. In the same folder as the initializer you can find components for the interstitial, banner and rewarded ads. You can use these ads by adding the components to your scene, and calling the **Load** and **Show** methods from these components.



## AdMob

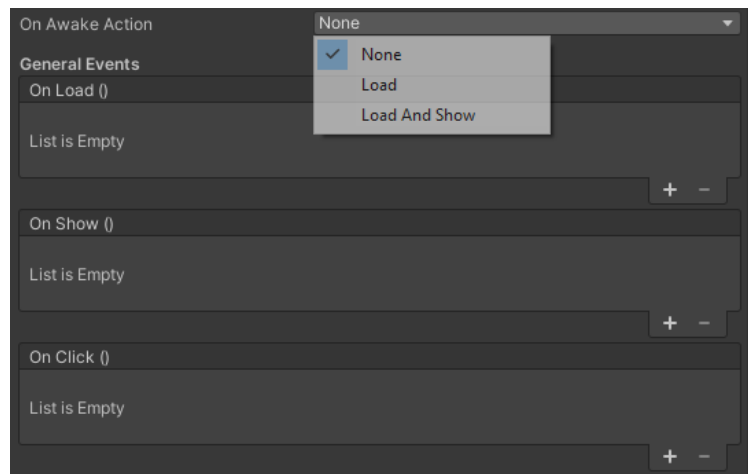
1. To get started, add the AdMob plugin to your project. You can follow [this page](#) to set up AdMob.
2. After setting up AdMob, go to 'Tools/DTT/Minigame Ads/Enable AdMob' to enable the components for AdMob.
3. You can then add the **AdMobInitializer** component to your scene. This component can be found under the 'Runtime/Ads/AdMob' folder.
4. In this folder you can also find all the ad types you can add to your project. These include the interstitial, banner, rewarded, interstitial rewarded and app start ads. You can use these ads by adding the components to your scene, and calling the **Load** and **Show** methods from these components.



## 4. Editor

### BaseAd

Every ad has some UnityEvents that allow you to easily call methods when ads are interacted with.



#### 1. On Awake Action

What should happen on awake. Load simply loads the ad, while Load And Show immediately shows the ad.

#### 2. OnLoad

Unity event that is invoked when the ad is loaded.

#### 3. OnShow

Unity event that is invoked when the ad is shown.

#### 4. OnClick

Unity event that is invoked when the ad is clicked.

### UnityAdsInitializer

#### 1. Initialize On Awake

Whether the initializer should initialize on Awake.

#### 2. Android Game Id

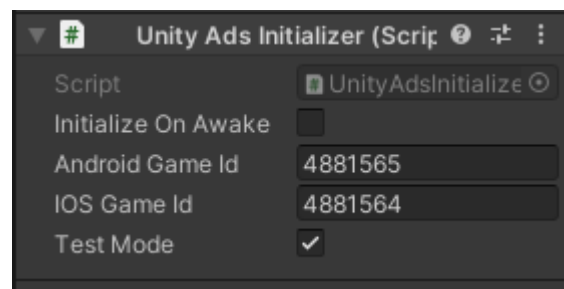
The game ID for Android. Can be found on your project dashboard.

#### 3. IOS Game Id

The game ID for iOS. Can be found on your project dashboard.

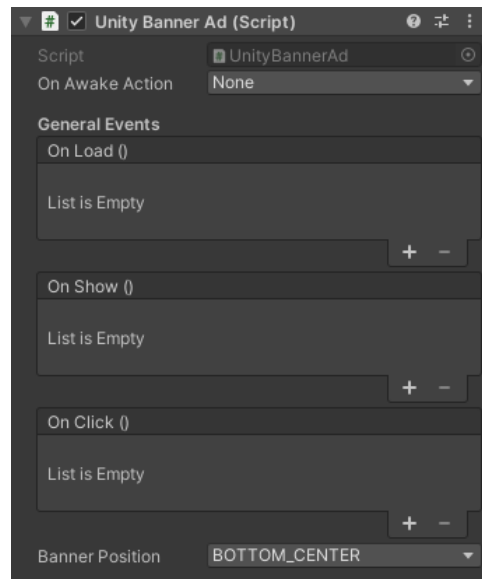
#### 4. Test Mode

Whether the ads should be initialized in test mode. Allows you to test your integration without serving live ads.





## UnityBannerAd

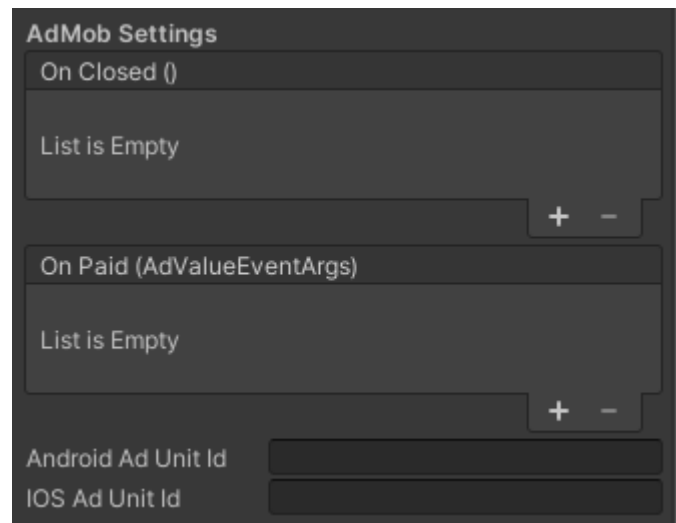


1. **Banner Position**  
determines the position the banner should be shown.

## AdMobAd

Every AdMob ad has the following fields in their editor. All AdMob ads require an ad unit ID. The test ID's for each ad type can be found [here](#).

1. **OnClosed**  
Invoked when the ad is closed.
2. **OnPaid**  
Invoked when the ad returned a paid event.
3. **Android Ad Unit Id**  
Ad unit ID for Android.
4. **IOS Ad Unit Id**  
Ad unit ID for iOS.







## AdMobBannerAd

1. **Banner Position**  
Determines at which position on screen the banner should be placed.
2. **Use Adaptive Banner**  
Whether an adaptive banner should be used. Adaptive banners allow you to display banners that fill the whole width of the screen at all times.
3. **Use Full Width**  
When using Adaptive banners, this will allow the banner to take up the full width of the screen.
4. **Custom Width**  
When full width is not being used, the custom width will be used instead.

**AdMob Settings**

On Closed ()

List is Empty

+ -

On Paid (AdValueEventArgs)

List is Empty

+ -

Android Ad Unit Id

IOS Ad Unit Id

**Banner Settings**

Banner Position: Bottom

Use Adaptive Banner: ☐

Use Full Width: ☒

Custom Width: 200



## 5. API

### BaseAd

*Base class that all ads inherit from. The API from this class can be applied to all ads.*

Property Name	Type	Description
onLoad	UnityEvent	Invoked when the ad loaded successfully.
onShow	UnityEvent	Invoked when the ad is shown on screen.
onClick	UnityEvent	Invoked when the ad is clicked.

Method name	Return Type	Parameters	Description
LoadAd	void	bool showOnLoad	Loads the ad. Set showOnLoad to immediately show the ad once it's loaded.
ShowAd	void	-	Shows the ad. Make sure the ad is loaded before calling this.



## UnityAdsInitializer

*Handles initializing Unity Advertisements. Make sure to initialize the ads before showing them.*

Property Name	Type	Description
IsInitialized	static bool	Whether Unity ads have been initialized or not.
Initialized	static Action	Invoked when Unity ads have been successfully initialized.

Method name	Return Type	Parameters	Description
Initialize	void	-	Call to start initializing Unity ads.

## UnityInterstitialAd

*Handles showing a full screen ad. You can override from the Unity ad methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onComplete	UnityEvent<UnityAdsShowCompletionState>	Invoked when the interstitial ad is completed.



## UnityRewardAd

*Handles showing a full screen ad that returns a reward. You can override from the Unity ad methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onComplete	UnityEvent<UnityAdsShowCompletionState>	Invoked when the interstitial ad is completed.
onRewardComplete	UnityEvent	Invoked when the reward ad was successful.
onRewardFailed	UnityEvent	Invoked when the reward ad failed to complete.

## UnityBannerAd

*Handles showing a banner ad. You can override from the banner ad methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onComplete	UnityEvent<UnityAdsShowCompletionState>	Invoked when the interstitial ad is completed.

Method name	Return Type	Parameters	Description
HideAd	void	bool destroy	Call to hide the banner.



## AdMobAdsInitializer

*Handles initializing AdMob advertisements. Make sure to initialize the ads before showing them.*

Property Name	Type	Description
IsInitialized	static bool	Whether AdMob has been initialized or not.
Initialized	static Action	Invoked when AdMob has successfully been initialized.

Method name	Return Type	Parameters	Description
Initialize	void	-	Call to start initializing AdMob.

## AdMobInterstitialAd

*Handles showing a full screen ad. You can override from the AdMob methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onClosed	UnityEvent	Invoked when the ad is closed.
onPaid	UnityEvent<AdValueEventArgs>	Invoked when the ad returned a paid event.
InterstitialAd	InterstitialAd	Reference to the ad instance.

Method name	Return Type	Parameters	Description
RequestAd	void	AdRequest request, bool showOnLoad	Loads in an ad with the given request. You can call this instead of LoadAd if you want to customize your AdRequest.



## AdMobRewardedAd

*Handles showing a rewarded ad. You can override from the AdMob methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onClosed	UnityEvent	Invoked when the ad is closed.
onPaid	UnityEvent<AdValueEventArgs>	Invoked when the ad returned a paid event.
onRewarded	UnityEvent<Reward>	Invoked when the ad returned a reward.
RewardedAd	RewardedAd	Reference to the ad instance.

Method name	Return Type	Parameters	Description
RequestAd	void	AdRequest request, bool showOnLoad	Loads in an ad with the given request. You can call this instead of LoadAd if you want to customize your AdRequest.



## AdMobRewardedInterstitialAd

*Handles showing a full screen rewarded ad. You can override from the AdMob methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onClosed	UnityEvent	Invoked when the ad is closed.
onPaid	UnityEvent<AdValueEventArgs>	Invoked when the ad returned a paid event.
onRewarded	UnityEvent<Reward>	Invoked when the ad returned a reward.
RewardedInterstitialAd	RewardedInterstitialAd	Reference to the ad instance.

Method name	Return Type	Parameters	Description
RequestAd	void	AdRequest request, bool showOnLoad	Loads in an ad with the given request. You can call this instead of LoadAd if you want to customize your AdRequest.



## AdMobBannerAd

*Handles showing a banner ad. You can override from the AdMob methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onClosed	UnityEvent	Invoked when the ad is closed.
onPaid	UnityEvent<AdValueEventArgs>	Invoked when the ad returned a paid event.
BannerView	BannerView	Reference to the ad instance.

Method name	Return Type	Parameters	Description
RequestAd	void	AdRequest request, bool showOnLoad	Loads in an ad with the given request. You can call this instead of LoadAd if you want to customize your AdRequest.
HideAd	void	-	Hides the banner.
RequestAd	void	BannerView banner, AdRequest request, bool showOnLoad	Loads in the passed in banner ad.





## AdMobAppOpenAd

*Handles showing an ad when the app opens. You can override from the AdMob methods to implement your own behaviours for this ad.*

Property Name	Type	Description
AdUnitId	string	The ad unit ID used for this ad.
onClosed	UnityEvent	Invoked when the ad is closed.
onPaid	UnityEvent<AdValueEventArgs>	Invoked when the ad returned a paid event.
AppOpenAd	AppOpenAd	Reference to the ad instance.

Method name	Return Type	Parameters	Description
RequestAd	void	AdRequest request, bool showOnLoad	Loads in an ad with the given request. You can call this instead of LoadAd if you want to customize your AdRequest.



## 6. Known Limitations

This asset currently does not have any known limitations.



## 7. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out.

Always feel free to contact us at:

[unity-support@d-tt.nl](mailto:unity-support@d-tt.nl)

*(We typically respond within 1-2 business days)*

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

### **DTT stands for Doing Things Together**

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

<https://www.d-tt.nl/en/>