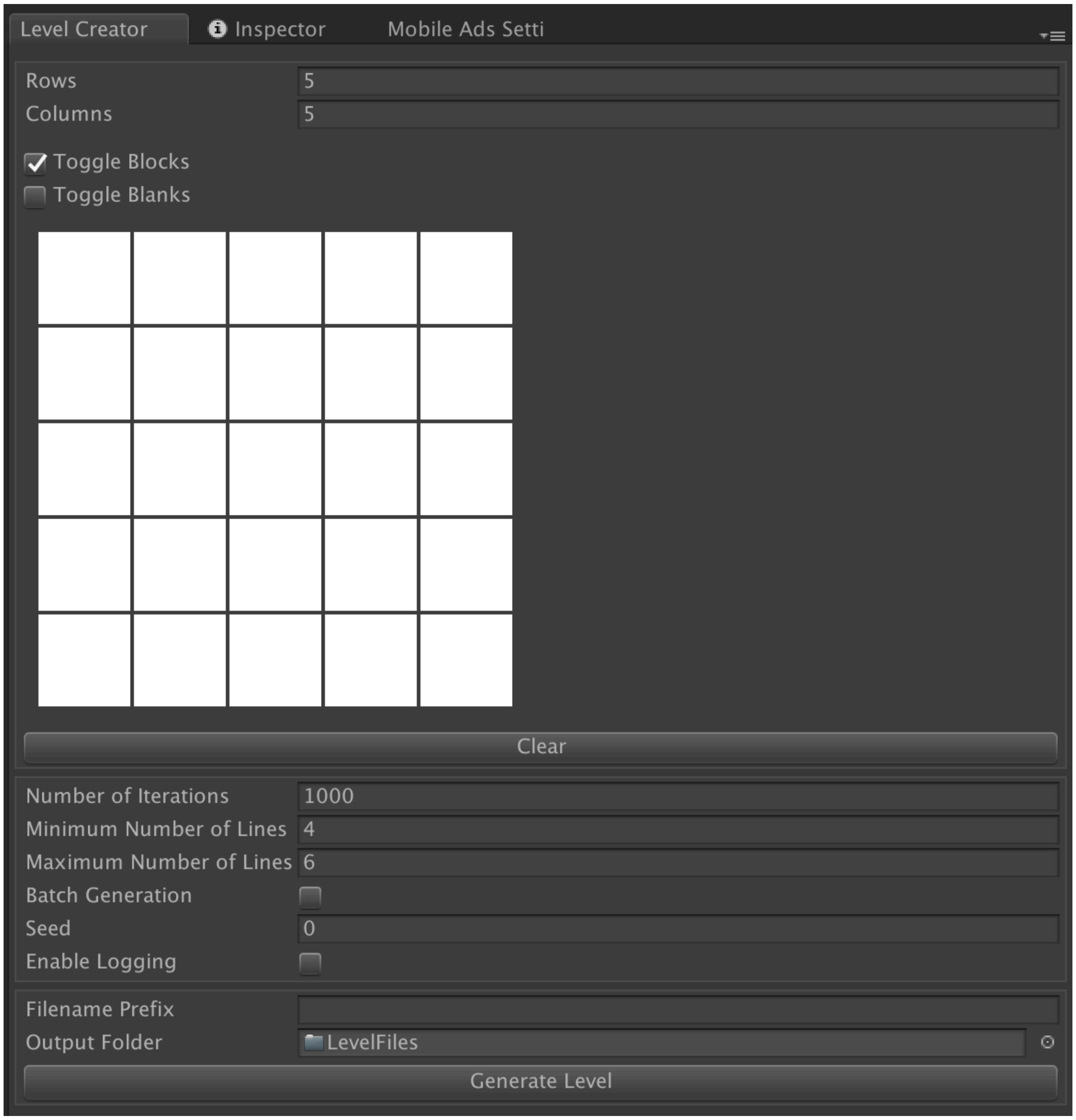
Dot Connect Documentation

Creating Levels

The asset comes with a level creator editor window which is used to create new levels in the game. To open the window selected the menu item **Window -> Dot Connect Level Creator**.



Rows 5
Columns 5
✓ Toggle Blocks
Toggle Blanks
Number of Iterations 1000
Minimum Number of Lines 4
Maximum Number of Lines 6
Batch Generation
Seed 0
Enable Logging
Filename Prefix
Output Folder LevelFiles

Rows - The number of rows in the level.

Columns - The number of columns in the level.

Toggle Blocks - If this is selected then clicking a cell on the window will toggle it as a “block” cell. When the game runs, the grid cell will display a grey block and lines will not be able to move into that cell.

Toggle Blanks - If this is selected then clicking a cell on the window will toggle it as a “blank” cell. When the game runs, the grid cell will be blank and lines will not be able to move into that cell.

Number of Iterations - The number of random moves the algorithm makes in order to randomize the lines in the level. The higher the number the more random the level will be. In most cases it can be left at 1000 since higher numbers don’t result in “better” levels.

Minimum Number of Lines - The minimum number of lines that should be in the level.

Maximum Number of Lines - The maximum number of lines that should be in the level.

NOTE: The minimum/maximum number of lines is only a preference that the level creator attempts to fulfill. In some cases levels will be created that have more/less than those values. For instance if you select a minimum number of 500 lines on a level that is only 3x3 then obviously it wont be able to fulfill that requirement.

Batch Generation - Select this if you want to create multiple levels at once.

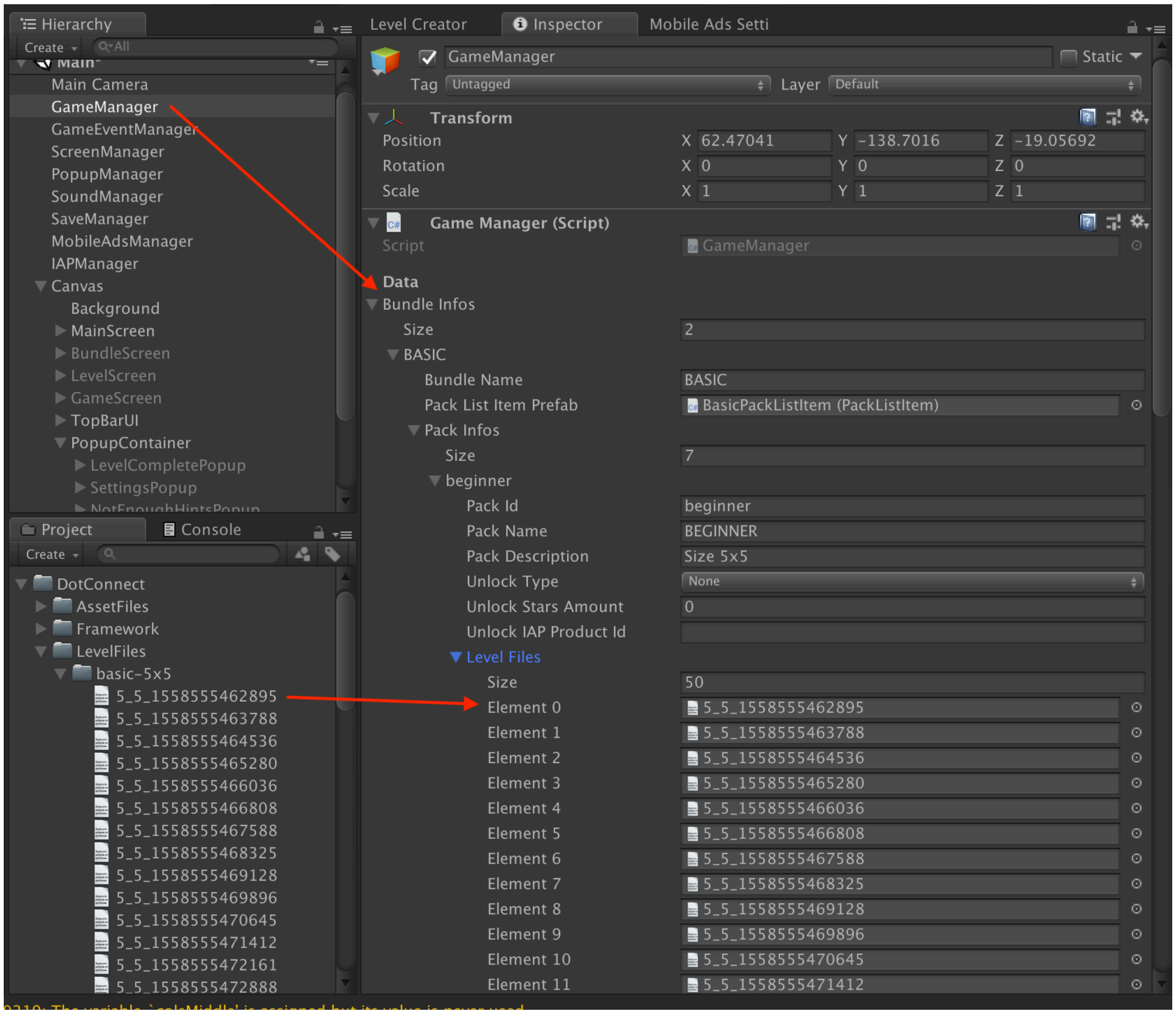
Seed - If Batch Generation is un-selected, then seed will appear and will be the seed used for the random number generator. (If seed is 0 then the system defined seed is used, ei no seed is specified)

Number of Levels - If Batch Generation is selected then this field will appear and specifies how many levels you want to create.

Filename Prefix - A prefix to append to the level file that is created.

Output Folder - The project folder to place the created level files (Drag from your Project window)

Once you created your level files you can add them to the game by selecting the **GameManager** and dragging them into one of the **Bundle/Pack Infos**:



Hierarchy
Create All
Main
Main Camera
GameManager
GameEventManager
ScreenManager
PopupManager
SoundManager
SaveManager
MobileAdsManager
IAPManager
Canvas
Background
MainScreen
BundleScreen
LevelScreen
GameScreen
TopBarUI
PopupContainer
LevelCompletePopup
SettingsPopup
NotEnoughHintsPopup

Project Console
Create
DotConnect
AssetFiles
Framework
LevelFiles
basic-5x5
5_5_1558555462895
5_5_1558555463788
5_5_1558555464536
5_5_1558555465280
5_5_1558555466036
5_5_1558555466808
5_5_1558555467588
5_5_1558555468325
5_5_1558555469128
5_5_1558555469896
5_5_1558555470645
5_5_1558555471412
5_5_1558555472161
5_5_1558555472888

Level Creator Inspector Mobile Ads Setti

GameManager Static
Tag Untagged Layer Default

Transform
Position X 62.47041 Y -138.7016 Z -19.05692
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Game Manager (Script)
Script GameManager

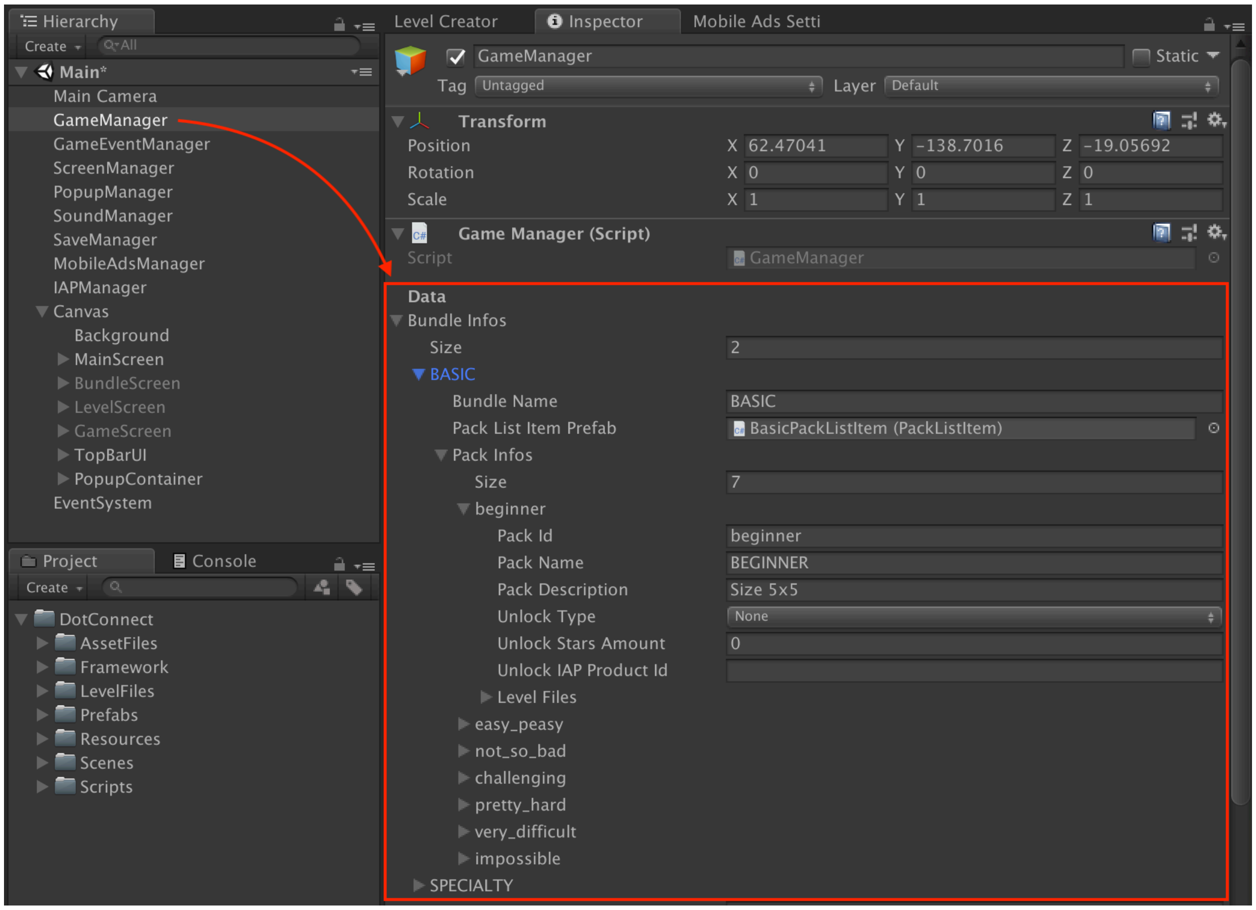
Data
Bundle Infos
Size 2
BASIC
Bundle Name BASIC
Pack List Item Prefab BasicPackListItem (PackListItem)
Pack Infos
Size 7
beginner
Pack Id beginner
Pack Name BEGINNER
Pack Description Size 5x5
Unlock Type None
Unlock Stars Amount 0
Unlock IAP Product Id
Level Files
Size 50
Element 0 5_5_1558555462895
Element 1 5_5_1558555463788
Element 2 5_5_1558555464536
Element 3 5_5_1558555465280
Element 4 5_5_1558555466036
Element 5 5_5_1558555466808
Element 6 5_5_1558555467588
Element 7 5_5_1558555468325
Element 8 5_5_1558555469128
Element 9 5_5_1558555469896
Element 10 5_5_1558555470645
Element 11 5_5_1558555471412

0219: The variable `calcMiddle' is assigned but its value is never used

Project

Bundles / Packs

The game is organized into Bundles and Packs. Bundles contain a list of Packs and each Pack contains a number of Levels. The **GameManager** is used to create/edit the Bundles and Packs in the game.



```
Main*
  Main Camera
  GameManager
  GameEventManager
  ScreenManager
  PopupManager
  SoundManager
  SaveManager
  MobileAdsManager
  IAPManager
  Canvas
    Background
    MainScreen
    BundleScreen
    LevelScreen
    GameScreen
    TopBarUI
    PopupContainer
  EventSystem

DotConnect
  AssetFiles
  Framework
  LevelFiles
  Prefabs
  Resources
  Scenes
  Scripts
```

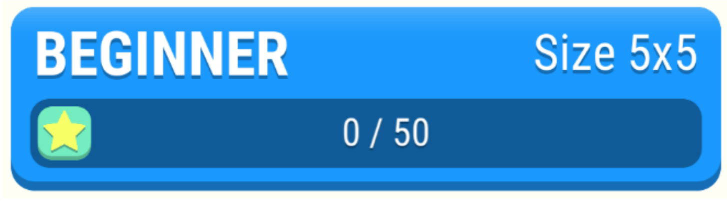
```
GameManager                                   Static
Tag Untagged                 Layer Default
Transform
  Position     X 62.47041   Y -138.7016   Z -19.05692
  Rotation     X 0          Y 0           Z 0
  Scale        X 1          Y 1           Z 1
Game Manager (Script)
  Script                     GameManager
Data
Bundle Infos
  Size                       2
  BASIC
    Bundle Name              BASIC
    Pack List Item Prefab    BasicPackListItem (PackListItem)
    Pack Infos
      Size                   7
      beginner
        Pack Id              beginner
        Pack Name            BEGINNER
        Pack Description     Size 5x5
        Unlock Type          None
        Unlock Stars Amount  0
        Unlock IAP Product Id
        Level Files
      easy_peasy
      not_so_bad
      challenging
      pretty_hard
      very_difficult
      impossible
  SPECIALTY
```

Bundle Infos:
Bundle Name - The name that is display in the game when the player is on the bundles screen.
Pack List Item - The Prefab which has a PackListItem Component attached to it. This prefab will be instantiated in the game to be used as the list item for the of the packs in the bundle. Each bundle can use a different PackListItem prefab.
Pack Infos - The list of Pack Infos in this bundle.

Pack Infos:
Pack Id - Should be unique for each Pack Info in the game (Across all bundles). This is used in the save data to save the progress of each pack.
Pack Name - The name that s displayed in the game when this pack is selected.

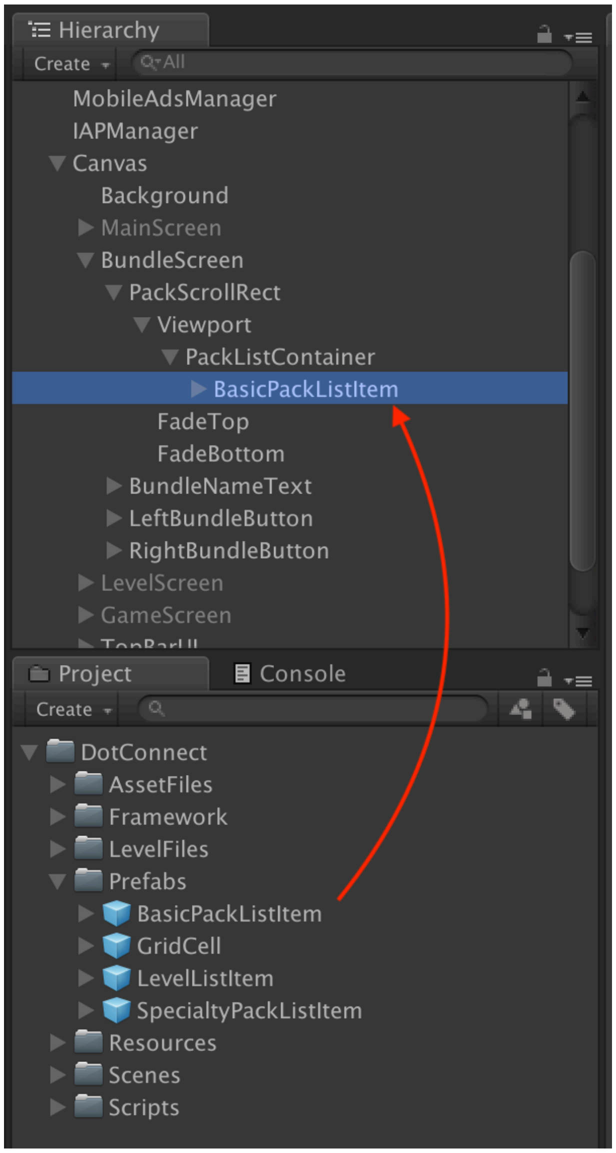
Pack Description - The text that is assigned to the PackListItems Description Text field in the game.
Unlock Type - Sets if this pack is locked, packs can be locked for Stars or IAP.
Unlock Stars Amount - If Unlock Type is Stars, then this field specifies the number of stars the player must acquire before the pack is unlocked.
Unlock IAP Product Id - If Unlock Type is IAP, then this is the product id that must be used when purchasing the pack. (This product id must also be added to the IAPSettings window, see the IAP section of this documentation for more information on setting up IAP)
Level Files - The list of level text files which were created using the level creator.

Pack List Items

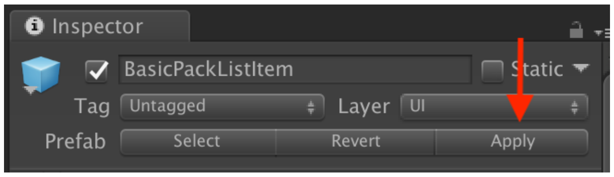


The Pack list items that appear on the Bundles screen are instantiated by the BundleScreen at run time. The prefabs that are used in the asset are located in the **Prefabs** folder. The **BasicPackListItem** is used for packs in the BASIC bundle and the **SpecialtyPackListItem** is used for packs in the SPECIALTY bundle. They are functionally the same, just have slight visual differences.

To edit the prefabs, drag them into the PackListContainer object located in the BundleScreen:

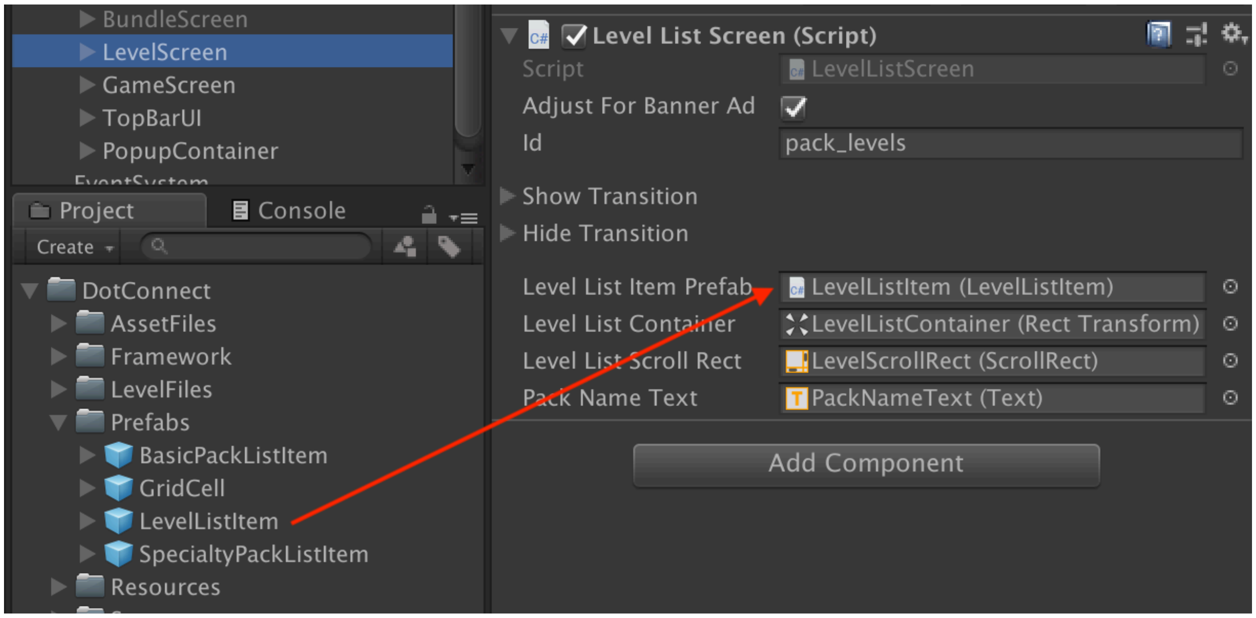


After you finish editing it click the Apply button at the top of the Inspector then delete it from the scene:



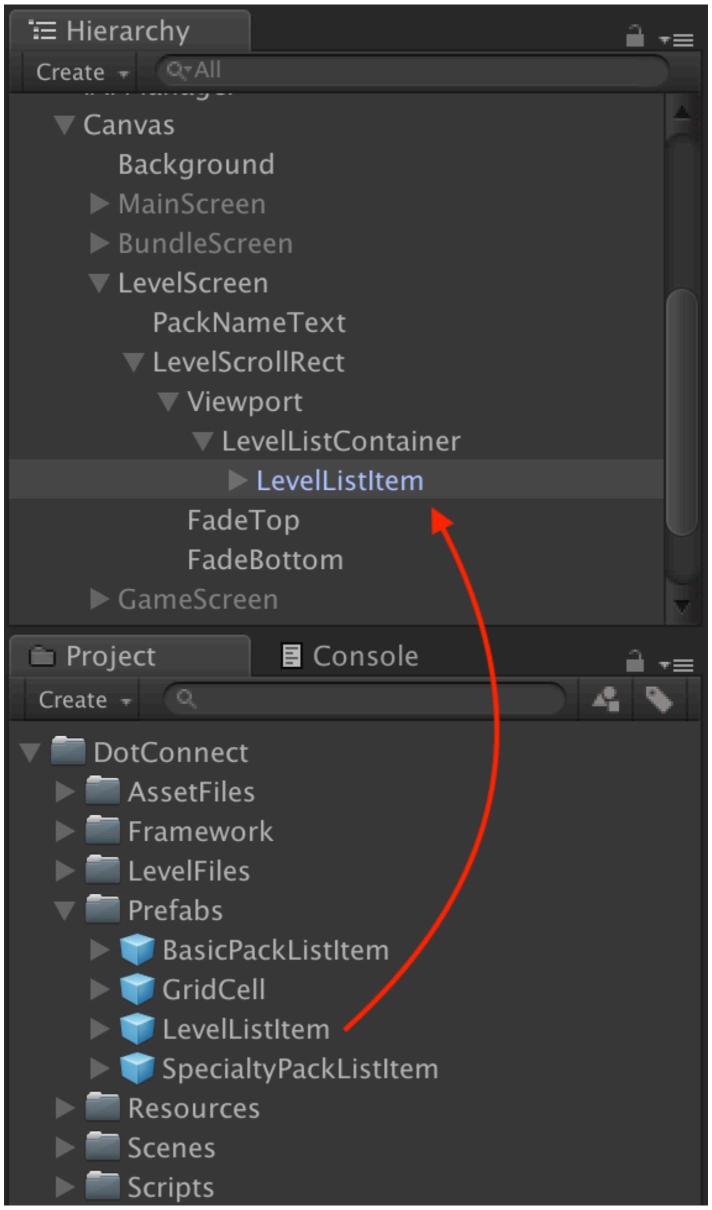
Level List Items

The level list items that appear on the level screen are instantiated by the LevelScreen at run time. The prefab that is used in the asset is called **LevelListItem** and is located in the **Prefabs** folder.

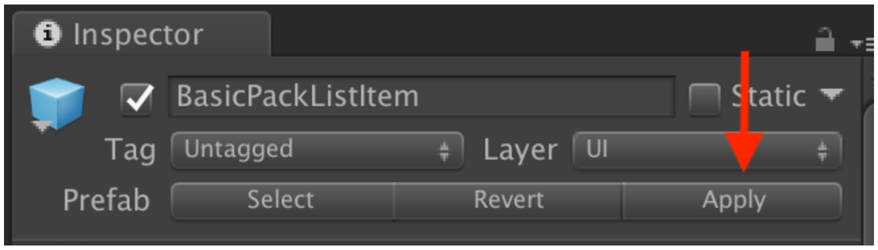


Level List Screen (Script)
Script: LevelListScreen
Adjust For Banner Ad: ✓
Id: pack_levels
Show Transition
Hide Transition
Level List Item Prefab: LevelListItem (LevelListItem)
Level List Container: LevelListContainer (Rect Transform)
Level List Scroll Rect: LevelScrollRect (ScrollRect)
Pack Name Text: PackNameText (Text)

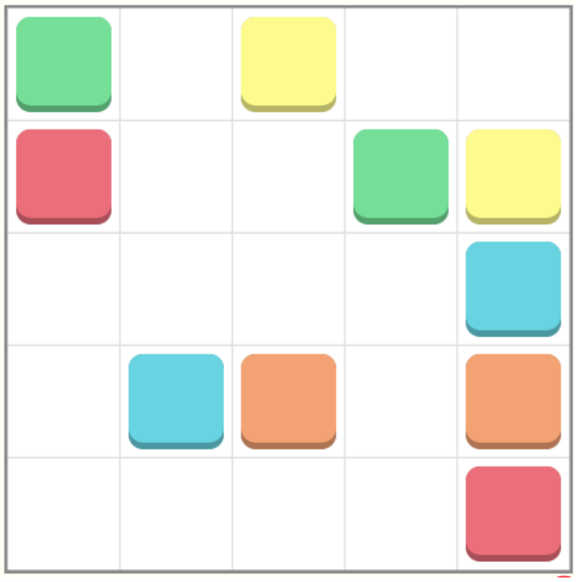
To edit the prefabs, drag them into the LevelListContainer object located in the LevelScreen:



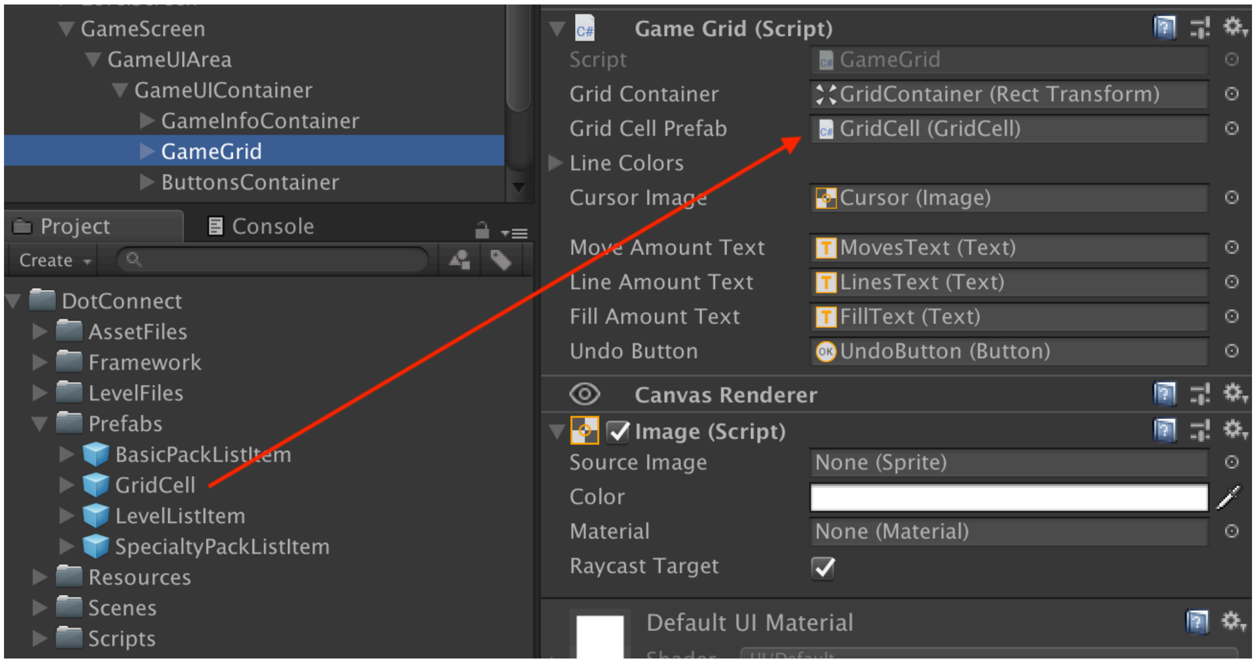
After you finish editing it click the Apply button at the top of the Inspector then delete it from the scene:



Grid Cells



The Grid Cells control how the individual cells look when they are displayed in the game. The **GridCell** prefab is instantiated for each cell in the grid. It is located in the **Prefabs** folder and is used in the game by assigning it to the **Grid Cell Prefab** field on the **GameGrid** object:



GameScreen
GameUIArea
GameUIContainer
GameInfoContainer
GameGrid
ButtonsContainer

Project | Console

DotConnect
AssetFiles
Framework
LevelFiles
Prefabs
BasicPackListItem
GridCell
LevelListItem
SpecialtyPackListItem
Resources
Scenes
Scripts

Game Grid (Script)

Script	GameGrid
Grid Container	GridContainer (Rect Transform)
Grid Cell Prefab	GridCell (GridCell)
Line Colors	
Cursor Image	Cursor (Image)
Move Amount Text	MovesText (Text)
Line Amount Text	LinesText (Text)
Fill Amount Text	FillText (Text)
Undo Button	UndoButton (Button)

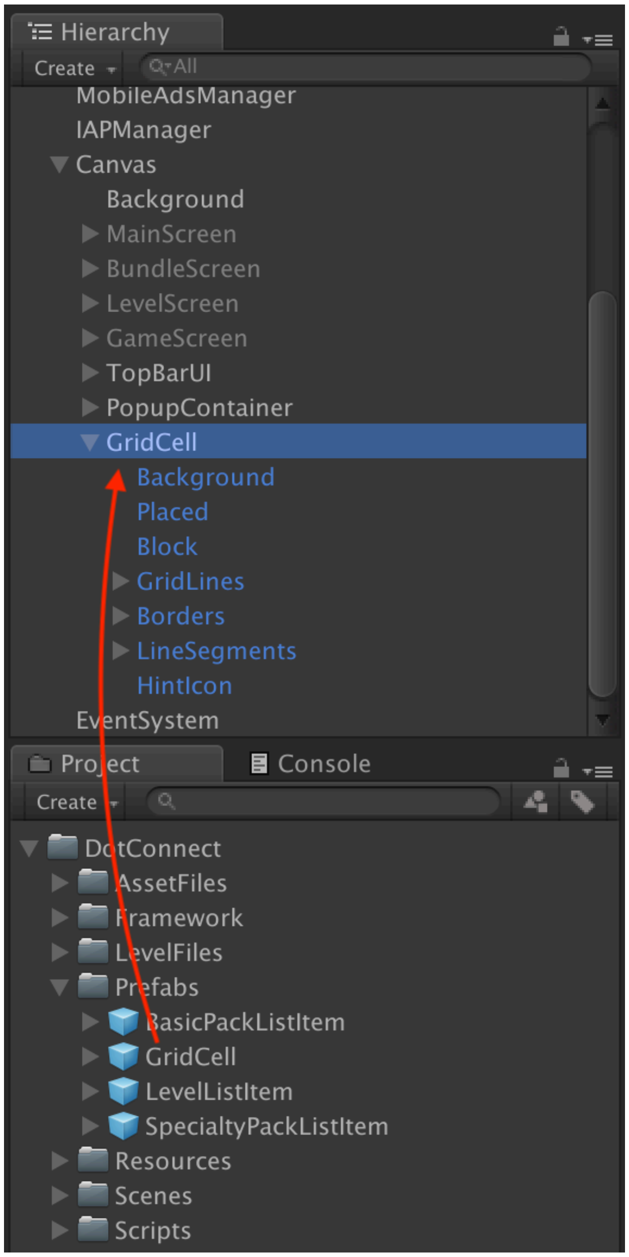
Canvas Renderer

Image (Script)

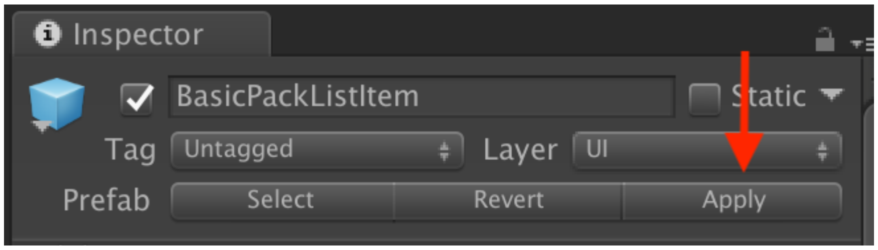
Source Image	None (Sprite)
Color	
Material	None (Material)
Raycast Target	✓

Default UI Material

To edit the GridCell drag it into the scene under the Canvas object:

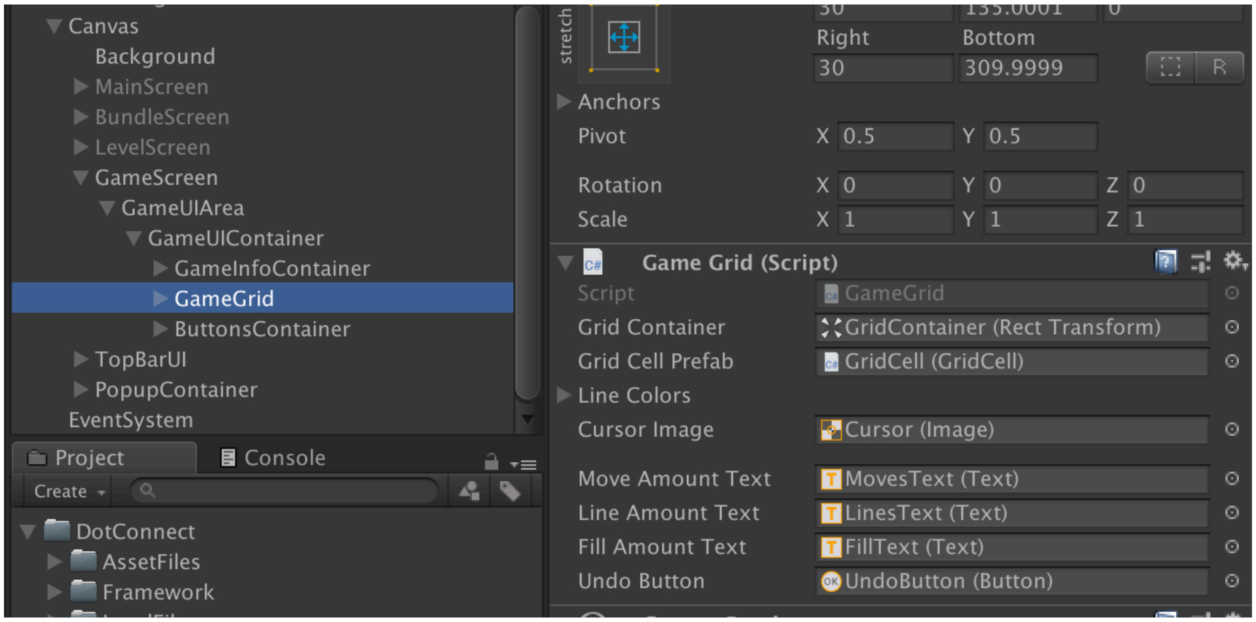


After you finish editing it click the Apply button at the top of the Inspector then delete it from the scene:



GameGrid / Line Colors

The GameGrid handles setting up the game board by instantiating all the GridCells, scaling and placing them in the correct positions, and setting the info texts (Moves, Lines, and Fill).



```
▼ Canvas
    Background
  ▶ MainScreen
  ▶ BundleScreen
  ▶ LevelScreen
  ▼ GameScreen
    ▼ GameUIArea
      ▼ GameUIContainer
        ▶ GameInfoContainer
        ▶ GameGrid
        ▶ ButtonsContainer
  ▶ TopBarUI
  ▶ PopupContainer
  EventSystem

Project   Console
Create
▼ DotConnect
  ▶ AssetFiles
  ▶ Framework
```

```
stretch            30          135.0001     0
                   Right       Bottom
                   30          309.9999
▶ Anchors
Pivot              X 0.5       Y 0.5
Rotation           X 0         Y 0          Z 0
Scale              X 1         Y 1          Z 1

▼ Game Grid (Script)
Script             GameGrid
Grid Container     GridContainer (Rect Transform)
Grid Cell Prefab   GridCell (GridCell)
▶ Line Colors
Cursor Image       Cursor (Image)
Move Amount Text   MovesText (Text)
Line Amount Text   LinesText (Text)
Fill Amount Text   FillText (Text)
Undo Button        UndoButton (Button)
```

It is also where the color of the lines are set. The **Line Colors** list is the list of colors that are used in the game. The GameGrid will assign the colors sequentially (So for instance, if there is only 4 lines in the level then the first 4 colors will be used). If there are more lines in a level then colors in the list then it will start re-using the colors.

Sounds

Sounds in the game are controlled using the SoundManager. On the SoundManager's inspector you will find a number of Sounds Infos already created and used in the game.

Sound Info fields

Id - The Id used to play the sound in the game.
Audio Clip - The sound file from your project.
Type - The type of sound (Sound Effect or Music), this is used to turn on/off all sounds of a particular type.
Play And Loop On Start - If selected the Audio Clip will play when the game starts and will loop forever unless it is stopped.
Clip Volume - Sets the volume of the sound when it is played.

Playing Sounds

Sounds can be played by calling the **Play** method on the SoundManager like so:

```
SoundManager.Instance.Play(string id);
```

You can easily play a sound when a Button is clicked by adding the **ButtonSound** component to a GameObject with a **Button** component. The ButtonSound will play the sound with the specified Id every time the button is clicked.

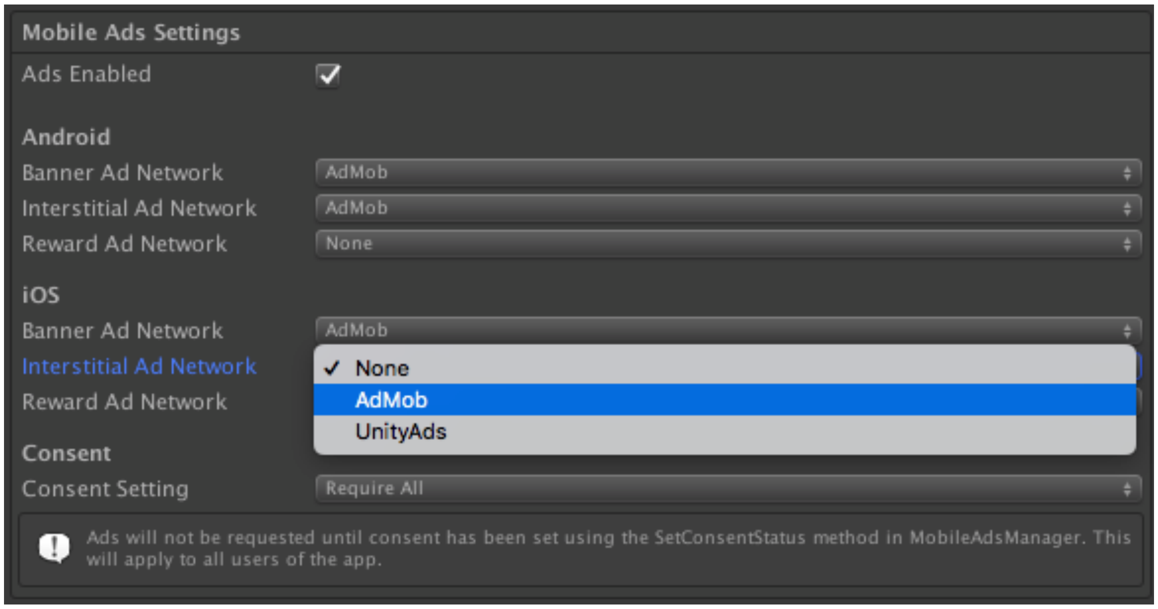
Ads

Ads are setup using the **Mobile Ads Settings** window which can be opened by selecting the menu item **Window -> Mobile Ads Settings** (Or clicking the button on the MobileAdsManager inspector).

On the Mobile Ads Settings window you can select either **AdMob** or **Unity Ads** to be used for Banner, Interstitial, and/or Reward ads for both Android and iOS platforms. Selecting **None** will disable ads for that platform / ad type.

AdMob Setup

Step1. Select AdMob in one or more of the drop downs.


Mobile Ads Settings
Ads Enabled ✓

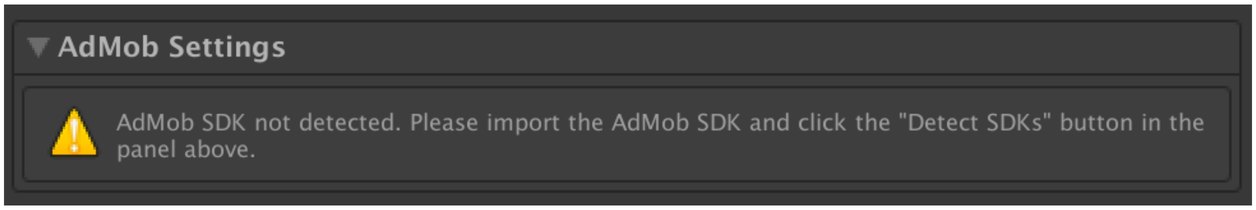
Android
Banner Ad Network: AdMob
Interstitial Ad Network: AdMob
Reward Ad Network: None

iOS
Banner Ad Network: AdMob
Interstitial Ad Network: ✓ None / AdMob / UnityAds
Reward Ad Network

Consent
Consent Setting: Require All

Ads will not be requested until consent has been set using the SetConsentStatus method in MobileAdsManager. This will apply to all users of the app.

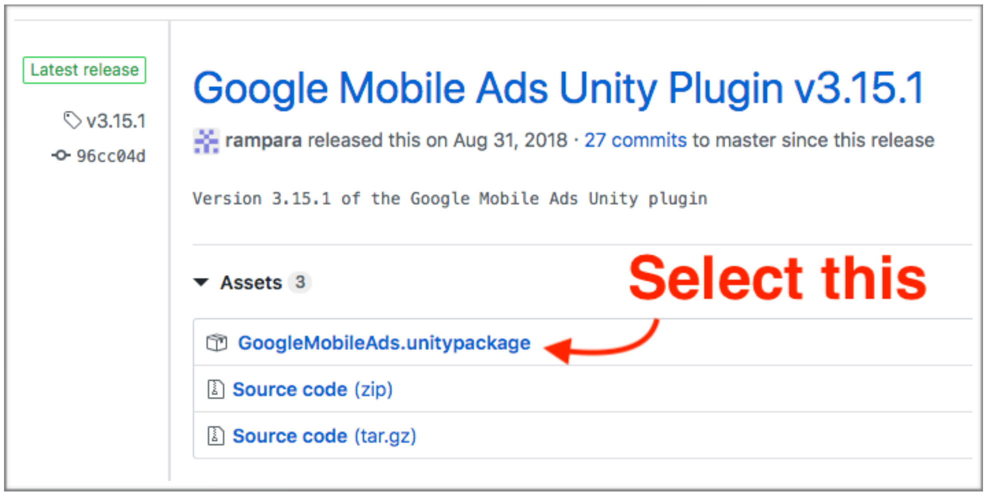
A new section will appear at the bottom of the window called **AdMob Settings**. Expanding it now will display the following warning:


AdMob Settings

AdMob SDK not detected. Please import the AdMob SDK and click the "Detect SDKs" button in the panel above.

Step2. Download and import the AdMob Unity SDK by clicking on this link **https://github.com/googleads/googleads-mobile-unity/releases** and clicking the GoogleMobileAds.unitypackage

Download the GoogleMobileAds.unitypackage:


Latest release

v3.15.1

96cc04d

Google Mobile Ads Unity Plugin v3.15.1

rampara released this on Aug 31, 2018 · 27 commits to master since this release

```
Version 3.15.1 of the Google Mobile Ads Unity plugin
```

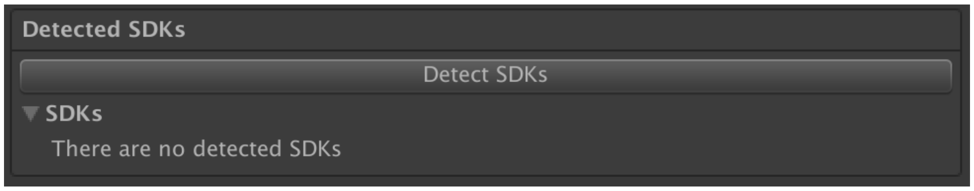
Assets 3

GoogleMobileAds.unitypackage

Source code (zip)

Source code (tar.gz)

Step3. Once the GoogleMobileAds.unitypackge has finished importing into Unity click the Detect SDKs button on the Mobile Ads Settings window:

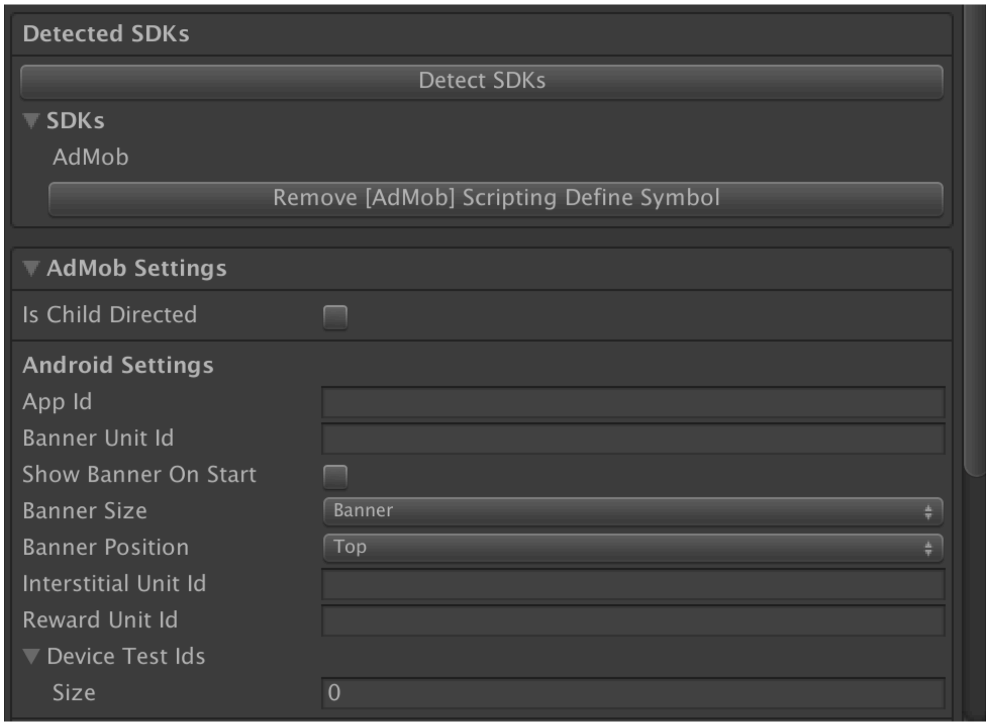

Detected SDKs

Detect SDKs

SDKs

There are no detected SDKs

After Unity finishes compiling AdMob should appear under the SDKs list and the AdMob fields should appear under AdMob Settings


Detected SDKs

Detect SDKs

SDKs

AdMob

Remove [AdMob] Scripting Define Symbol

AdMob Settings

Is Child Directed

Android Settings

App Id	
Banner Unit Id	
Show Banner On Start	
Banner Size	Banner
Banner Position	Top
Interstitial Unit Id	
Reward Unit Id	

Device Test Ids

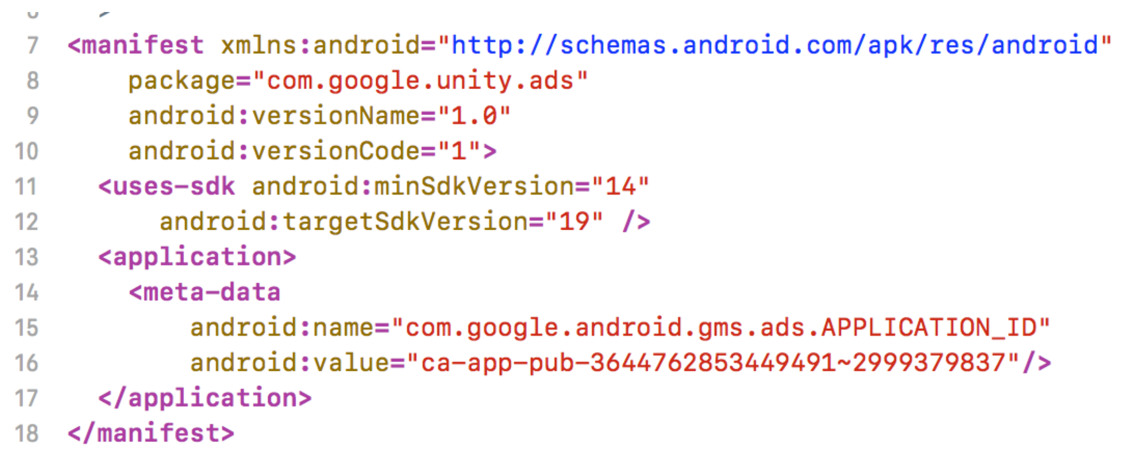
Size	0

Step4. Add you App Id and Unit Ids to the fields under AdMob Settings

Step5 [Android only]. Open the AndroidManifest located in the folder Plugins/Android/GoogleMobileAdsPlugin and add the following lines in the **application** element, replace ADMOB_APP_ID with your App Id:

```
<meta-data
        android:name="com.google.android.gms.ads.APPLICATION_ID"
        android:value="ADMOB_APP_ID"/>
```

Your AndroidManifest should look something like this:



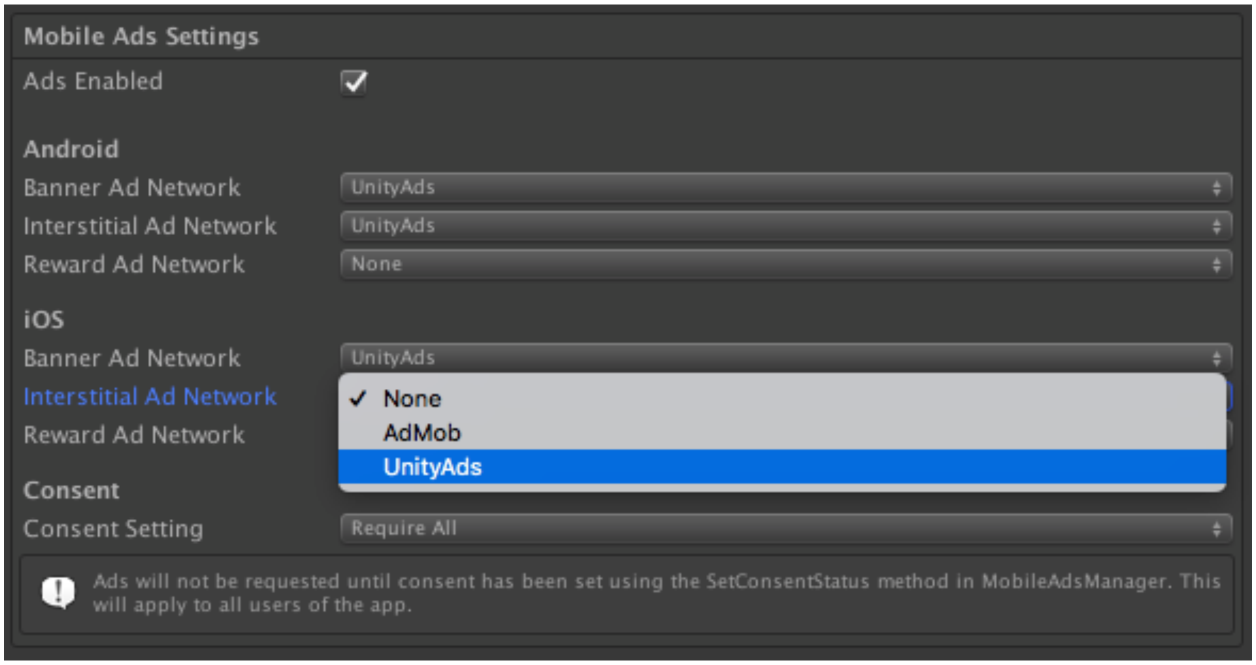
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.google.unity.ads"
    android:versionName="1.0"
    android:versionCode="1">
  <uses-sdk android:minSdkVersion="14"
      android:targetSdkVersion="19" />
  <application>
    <meta-data
        android:name="com.google.android.gms.ads.APPLICATION_ID"
        android:value="ca-app-pub-3644762853449491~2999379837"/>
  </application>
</manifest>
```

Step6 [Android Only]. Make sure the play services resolver that comes with the GoogleMobileAds plugin has executed by selecting the menu item **Assets -> Play Services Resolver -> Android Resolver -> Resolve**.

Thats it! AdMob ads should appear in the game.

Unity Ads Setup

Step1. Select Unity Ads in one or more of the drop downs.



Mobile Ads Settings

Ads Enabled ✓

Android

Banner Ad Network: UnityAds

Interstitial Ad Network: UnityAds

Reward Ad Network: None

iOS

Banner Ad Network: UnityAds

Interstitial Ad Network: ✓ None / AdMob / UnityAds

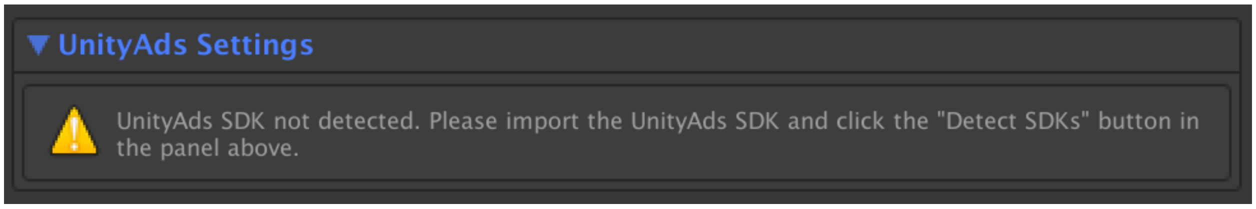
Reward Ad Network

Consent

Consent Setting: Require All

Ads will not be requested until consent has been set using the SetConsentStatus method in MobileAdsManager. This will apply to all users of the app.

A new section will appear at the bottom of the window called **UnityAds Settings**. Expanding it now will display the following warning:

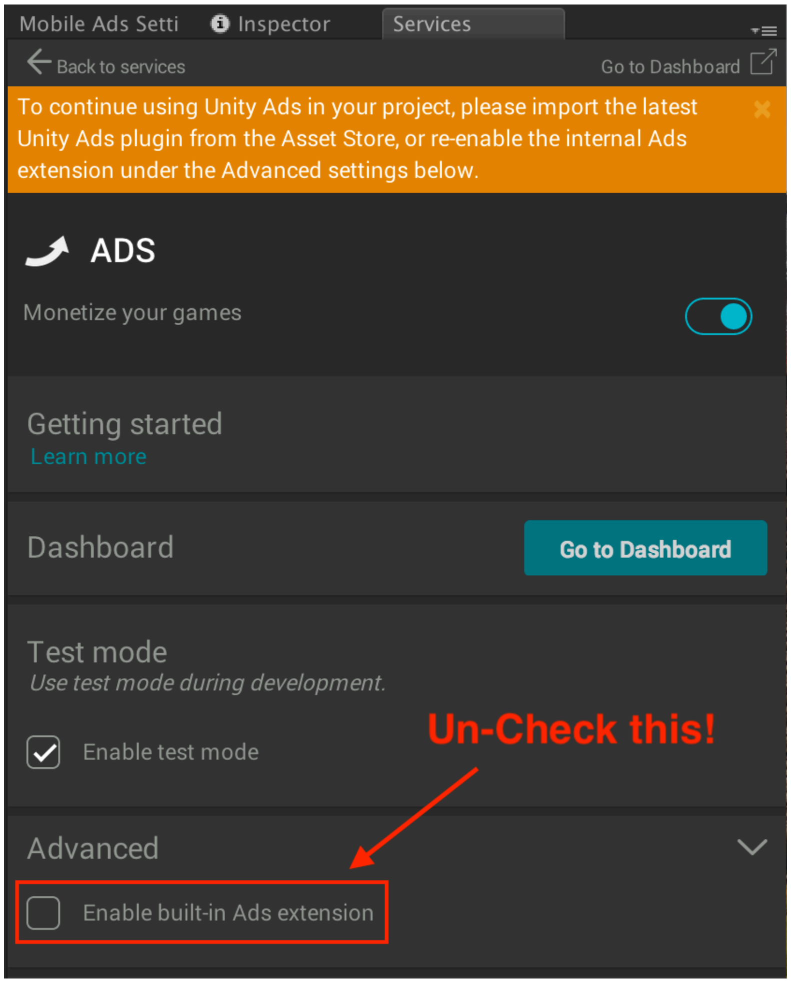


UnityAds Settings

UnityAds SDK not detected. Please import the UnityAds SDK and click the "Detect SDKs" button in the panel above.

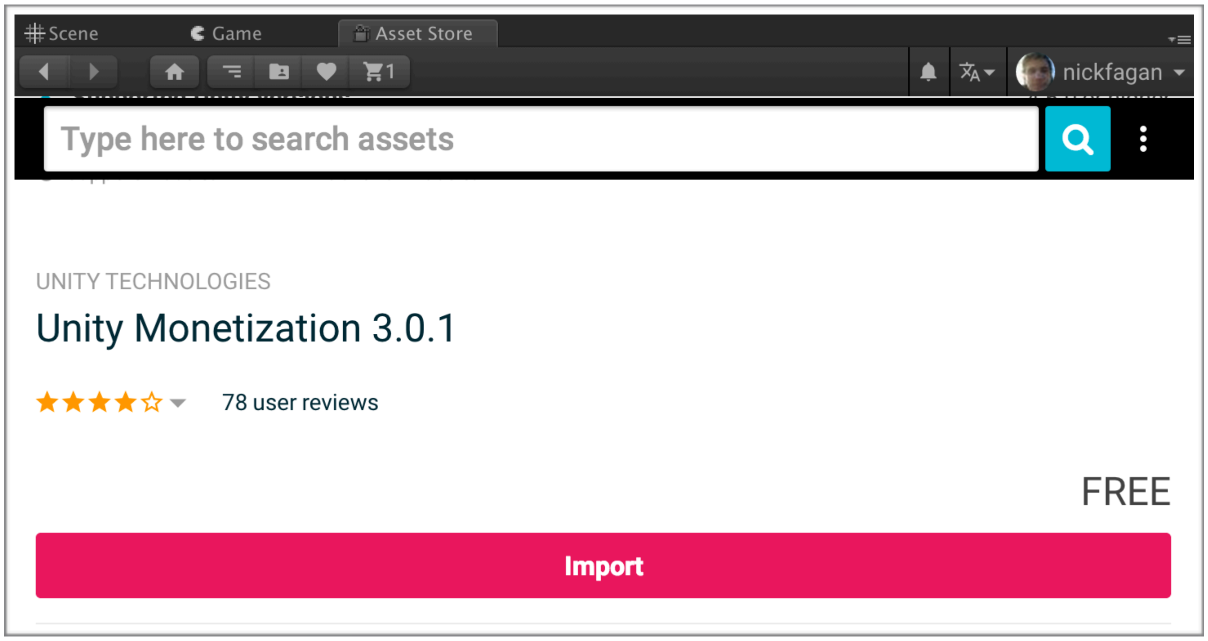
Step2. Enable Ads in the Services window:

*** **IMPORTANT** *** Make sure “Enable built-in Ads Extension” is disabled or it will collide with the Monetization plugin you will import in the next step. To do so expand the **Advanced** section and un-check the field if it is checked

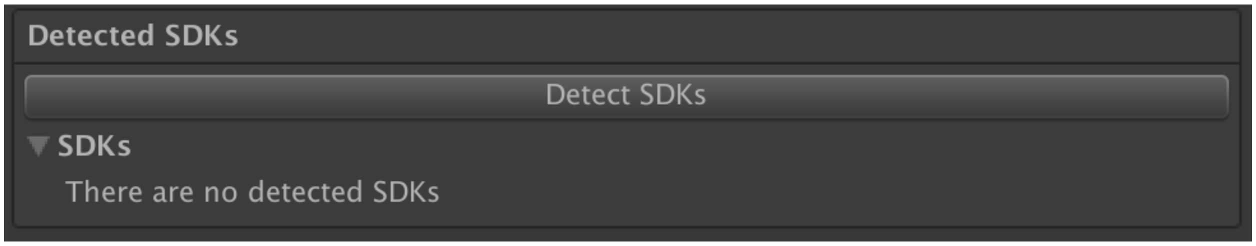


To continue using Unity Ads in your project, please import the latest Unity Ads plugin from the Asset Store, or re-enable the internal Ads extension under the Advanced settings below.

Step2. Open the Asset Store window and Download/Import the Unity Monetization asset:



Step3. Click the **Detect SDKs** button on the Mobile Ads Settings window:

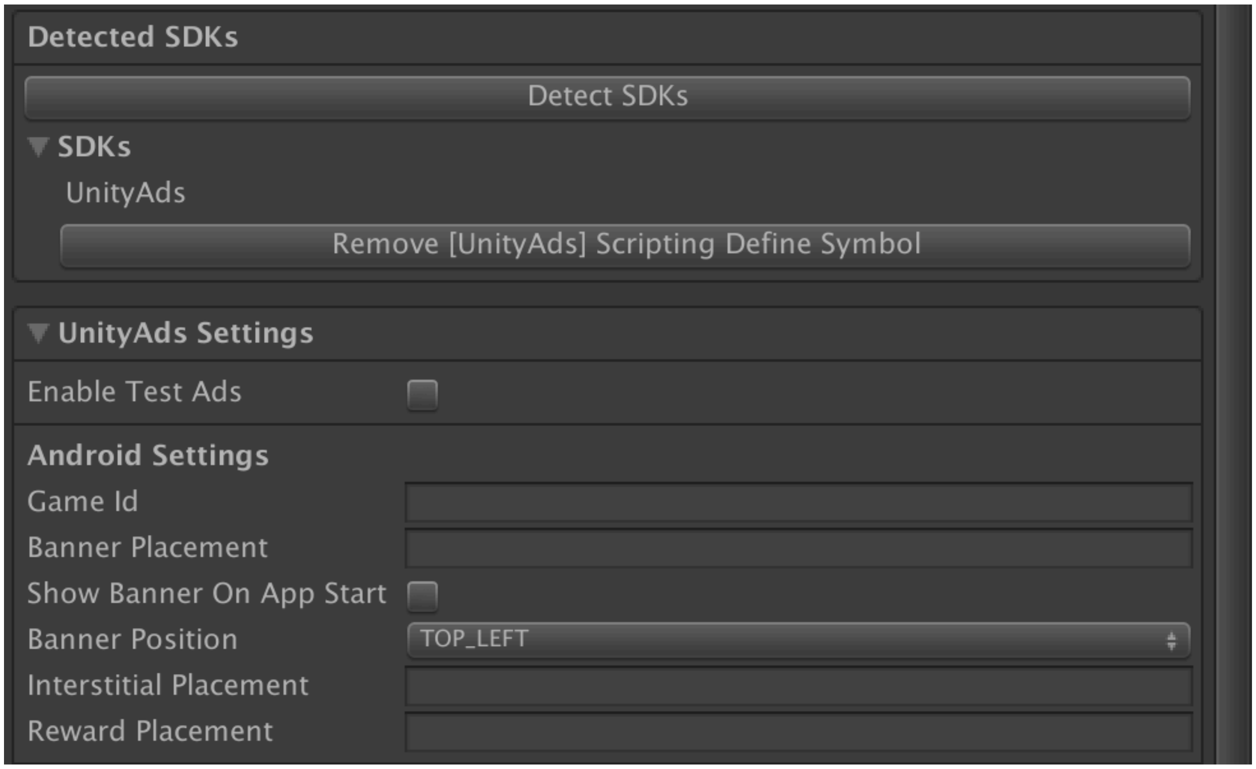

Detected SDKs

Detect SDKs

SDKs

There are no detected SDKs

After Unity finishes compiling, UnityAds should appear under the SDKs list and the UnityAds fields should appear under UnityAds Settings:


Detected SDKs

Detect SDKs

SDKs

UnityAds

Remove [UnityAds] Scripting Define Symbol

UnityAds Settings

Enable Test Ads

Android Settings

Game Id

Banner Placement

Show Banner On App Start

Banner Position: TOP_LEFT

Interstitial Placement

Reward Placement

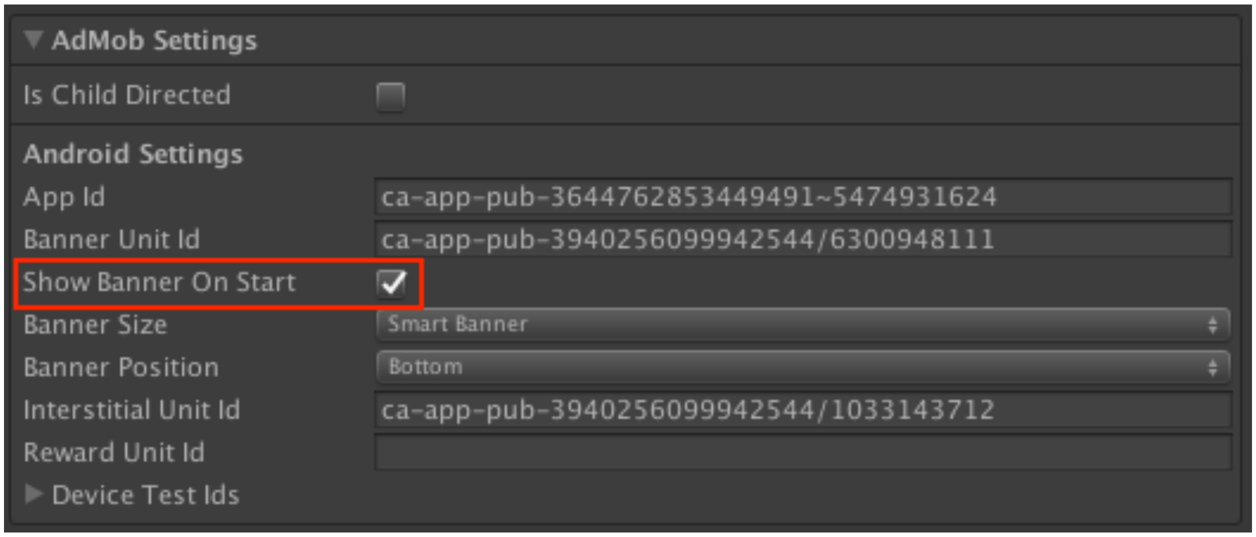
Step4. Add you Game Id and Placement Ids to the fields under UnityAds Settings.

Thats it! Unity Ads will now appear in your game.

Showing Ads

The game is already setup to show Interstitial ads every couple of levels. You can set how many levels the player can play before seeing another interstitial ad by setting the **Num Levels Between Ads** field on the **GameManager** inspector.

For banner ads you must select the **Show Banner Ads On Start** field on the Mobile Ads Settings window:



AdMob Settings

Is Child Directed	☐

Android Settings

App Id	ca-app-pub-3644762853449491~5474931624
Banner Unit Id	ca-app-pub-3940256099942544/6300948111
Show Banner On Start	✓
Banner Size	Smart Banner
Banner Position	Bottom
Interstitial Unit Id	ca-app-pub-3940256099942544/1033143712
Reward Unit Id	

Device Test Ids

Consent

Consent can be required before any ads are loaded by setting the **Consent Setting** on the Mobile Ads Manager. There are three types you can set the consent setting to:

Not Required - Consent is not required for ads to be loaded.

Required Only In EEA - Consent is only required for users in the European Economic Area. When the app starts it first attempts to determine if the user is located in the EEA and if so ads will not be loaded until the consent status has been set to either Personalized or Non-Personalized. If the user location can not be determined for any reason then it errs on the side of caution and requires consent before ads are loaded.

Require All - Consent is required for all users before ads are loaded.

Setting the Consent Status

If consent is required before ads are loaded then the **SetConsentStatus** method must be called on the MobileAdsManager to set the consent status to either Personalized or Non-Personalized ads.

To set the consent simply call the method like so:

```
MobileAdsManager.Instance.SetConsentStatus(consentStatus);
```

The consentStatus parameter is an integer value:

1 - Indicates the user has consented to receive personalized ads
0 - Indicates the user should only be shown non-personalized ads.

Testing

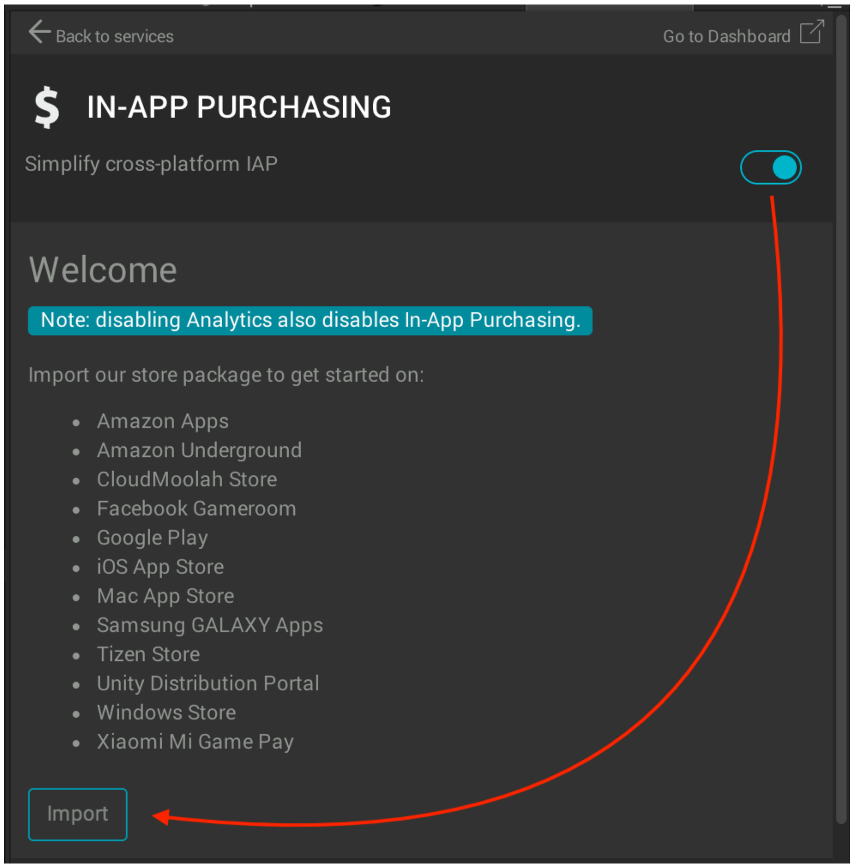
You should use the TestScene located in the Scenes folder to test if you have setup your ad networks properly. The test scene can be used to show ads and will print out log messages if there are any errors.

IAP

IAP is setup using the **IAP Settings** window which can be opened by selecting the menu item **Window -> IAP Settings** (Or clicking the button on the IAPManagers inspector).

Enable IAP

To enable IAP first you need to import the Unity plugin from the Services window. Open the Services window and turn on IAP then click the Import button:



Welcome

Note: disabling Analytics also disables In-App Purchasing.

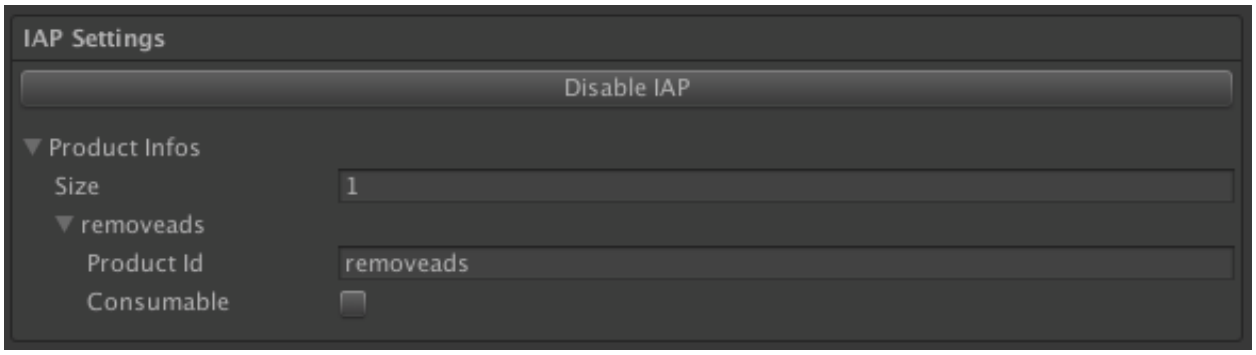
Import our store package to get started on:

- Amazon Apps
- Amazon Underground
- CloudMoolah Store
- Facebook Gameroom
- Google Play
- iOS App Store
- Mac App Store
- Samsung GALAXY Apps
- Tizen Store
- Unity Distribution Portal
- Windows Store
- Xiaomi Mi Game Pay

Once it has finished importing you can open the IAP Settings window and click the Enable IAP button which will enable the code in the project.

Add Product Ids

To add products open the IAP Settings window and add a Product Info for each product in your game.



IAP Settings

Disable IAP

Product Infos

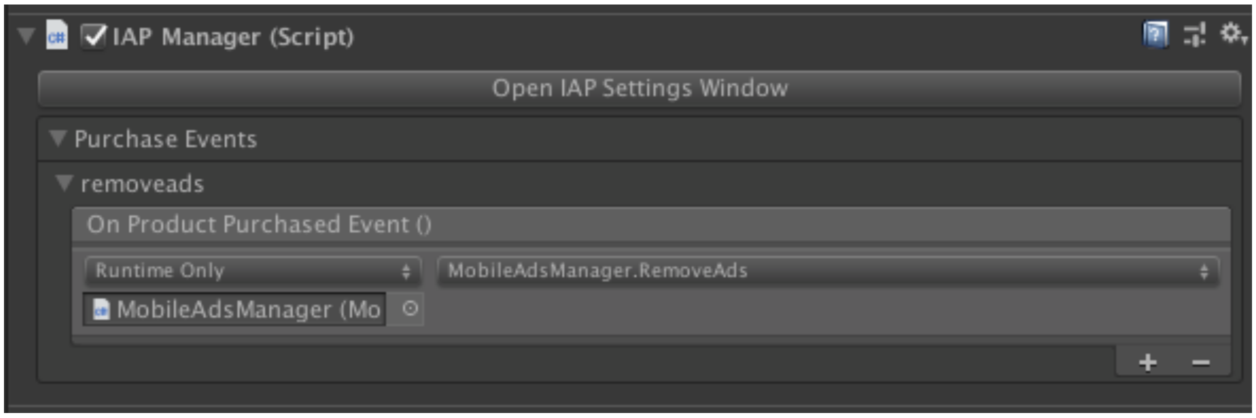
Size: 1

removeads

Product Id: removeads

Consumable: ☐

Once you add a new product an entry will appear on the IAPManager's inspector. Here you can add events for when the product is purchased:



IAP Manager (Script)

Open IAP Settings Window

Purchase Events

removeads

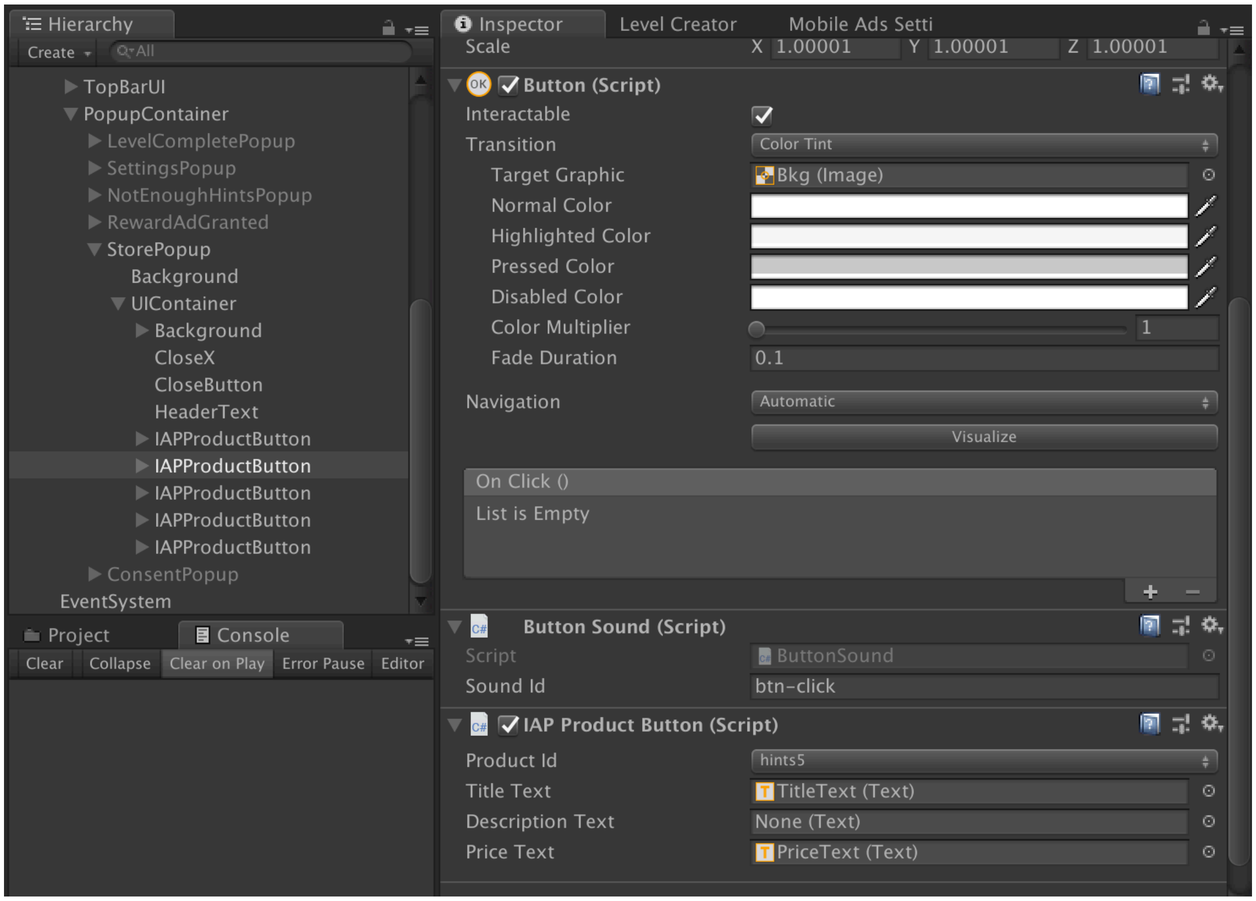
On Product Purchased Event ()

Runtime Only | MobileAdsManager.RemoveAds

MobileAdsManager (Mo

Purchase A Product

To invoke the purchasing of a product you can use the **IAPProductButton** component.



Scale X 1.00001 Y 1.00001 Z 1.00001

Button (Script)
Interactable ✓
Transition: Color Tint
Target Graphic: Bkg (Image)
Normal Color
Highlighted Color
Pressed Color
Disabled Color
Color Multiplier: 1
Fade Duration: 0.1
Navigation: Automatic
Visualize
On Click ()
List is Empty

Button Sound (Script)
Script: ButtonSound
Sound Id: btn-click

IAP Product Button (Script)
Product Id: hints5
Title Text: TitleText (Text)
Description Text: None (Text)
Price Text: PriceText (Text)

The Product Id dropdown will contain all the products you created in the IAPSettings. When the button is clicked in the game the IAPManager's BuyProduct method will be called with selected Product Id. You can also call the BuyProduct method manually like so:

IAPManager.Instance.BuyProduct(string productId);

You can also listen for successful product purchases with the OnProductPurchases action like so:

IAPManager.Instance.OnProductPurchases += YourMethod;

Testing

You should use the TestScene located in the Scenes folder to test if you have setup IAP correctly. The scene will create a button for each product in the IAPSettings window which you can click to purchase the product. Logs will output on the screen to show and errors that may occur.