Sudoku Documentation

To insure a quick response to any issues with the asset please send all support requests to the following e-mail address:

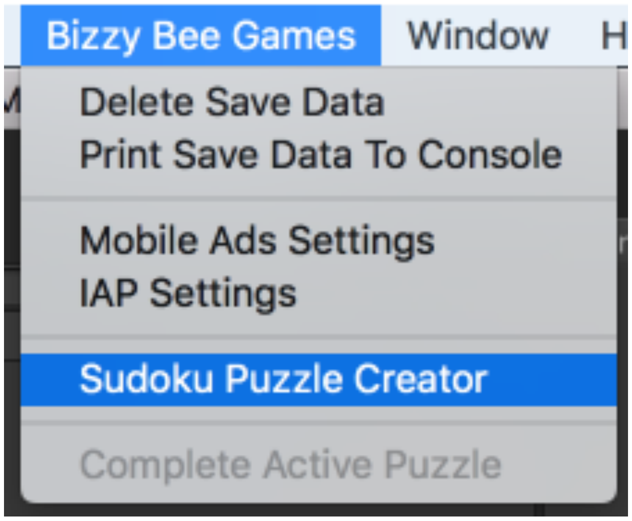
support@bizzybeegames.com

Please include the asset name and Unity version you are using. Thank you!

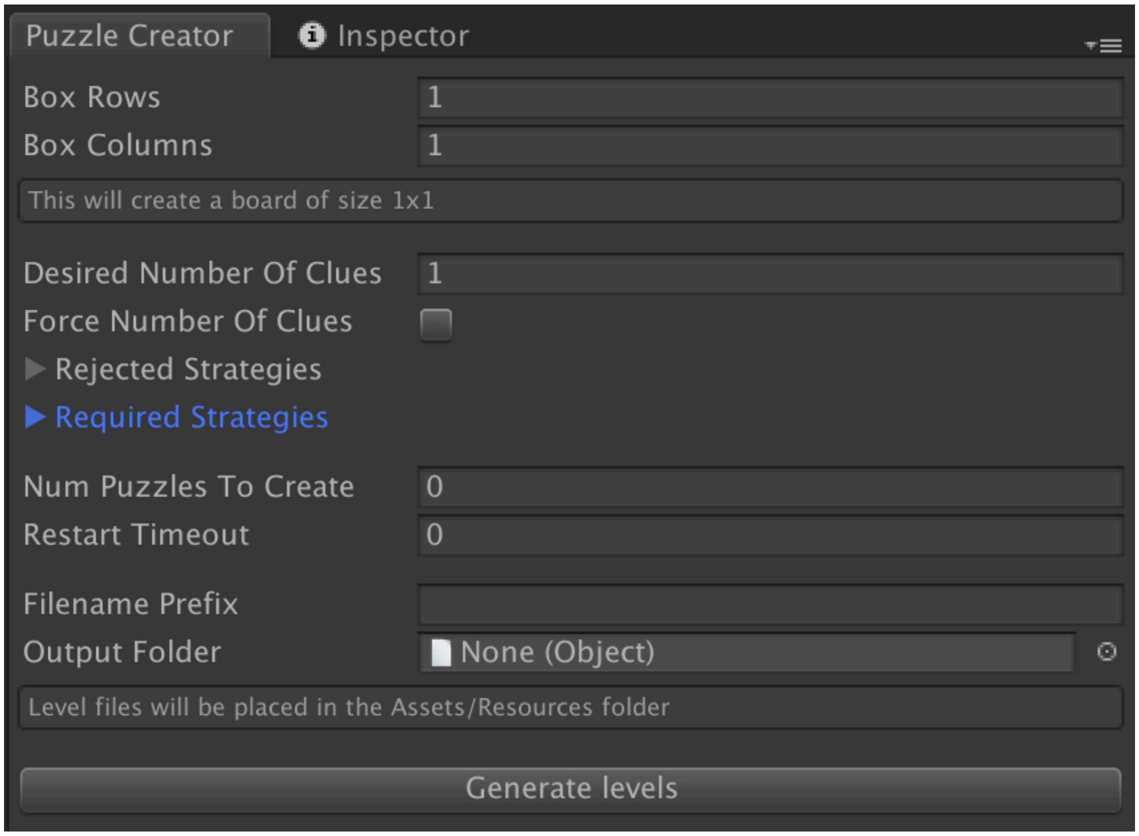
Table of Contents:

Creating Sudoku Puzzles

The game works by loading pre-generated puzzle files that are created with the Puzzle Creator editor window. To open the window select the menu item **Bizzy Bee Games -> Sudoku Puzzle Creator**:



Which open the Puzzle Creator window:



Box Rows	1
Box Columns	1

This will create a board of size 1x1

Desired Number Of Clues	1
Force Number Of Clues	☐

▶ Rejected Strategies

▶ Required Strategies

Num Puzzles To Create	0
Restart Timeout	0
Filename Prefix	
Output Folder	None (Object)

Level files will be placed in the Assets/Resources folder

Box Rows and **Box Cols** sets the size of puzzle that you want to create. These specify the number of rows/columns in one of the nine boxes on a Sudoku puzzle. So for instance setting them to 3 and 3 will create a classic 9x9 sudoku puzzle.

Desired Number Of Clues sets the number of “numbers” that appear on the puzzle when the player first starts the puzzle. The algorithm will try to get as close to this number as possible while retaining a valid sudoku puzzle (Only one solution to the puzzle). If **Force Number Of Clues** is selected then it will continue to try new random puzzles until it can create one with the exact number of desired clues.

NOTE: The generation process may take a long time if Force Number Of Clues is selected. It may also never complete if the desired number of clues is set so low that that there is no possible valid board. For instance, on a 9x9 sudoku puzzle, the lowest possible number of clues is 17 so setting Desired Number Of Clues to 16 will cause the the algorithm to run forever.

Rejected Strategies is all the sudoku solving strategies you do not want to be used in order to solve the sudoku puzzle. For instance, if you wanted to create an easy puzzle, select all except for Hidden singles. This will create puzzles where only naked/hidden single strategies are required to solve the board and will reject any puzzles where more advanced strategies are needed.

Required Strategies is all the sudoku solving strategies you want to be required to solve the puzzle. Only puzzles where one of the selected strategies is used to solve the puzzle will be generated. For instance, if you wanted to create a hard puzzle, select Hidden Pairs/Hidden Triples. This will create only puzzles that require one of those strategies in order for it to be solved.

Num Puzzles To Create sets the number of puzzle files you wish to generate and click the Generate Levels button. This will place your level files in the Resources folder.

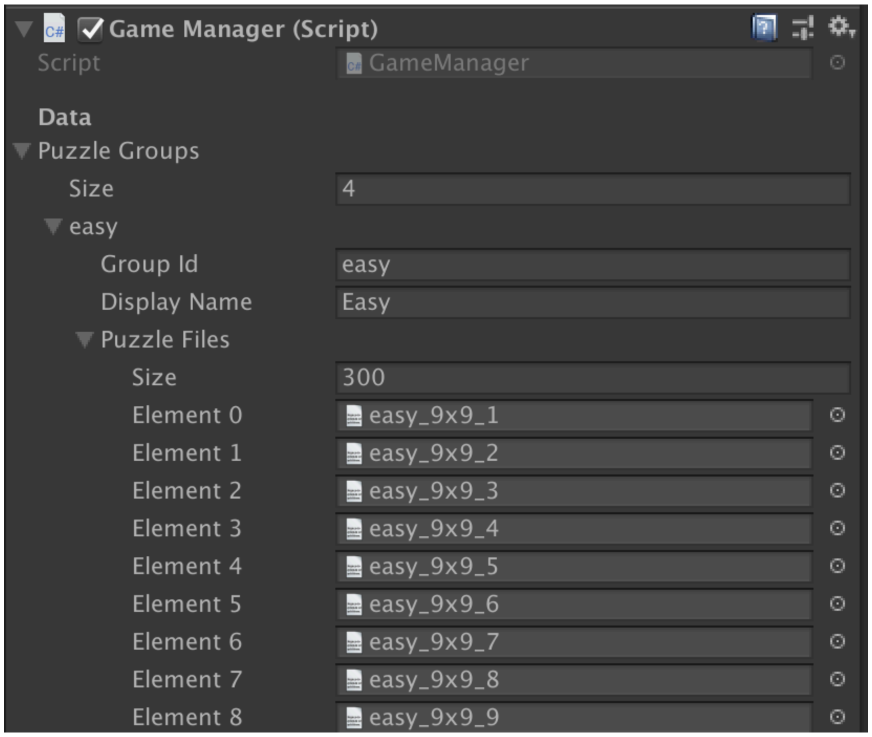
Restart Timeout is the amount of seconds before the algorithm gives up on the current puzzle, discards it, and tries again with a new random placement of numbers. This is required because sometimes on larger puzzles it will create a starting sequence of numbers that is very hard for the algorithm to solve so the best course of action is to discard it and try again with different starting numbers. Setting this to 0 will disable this and the algorithm will continue trying until the puzzle is solved.

Filename Prefix is the prefix to give all level files that are generated, the file name will be of the following format: prefix_#.txt

Output Folder is the folder to place generated puzzle files in, this is set by dragging a folder from the Project window.

Using Puzzle Files In Game

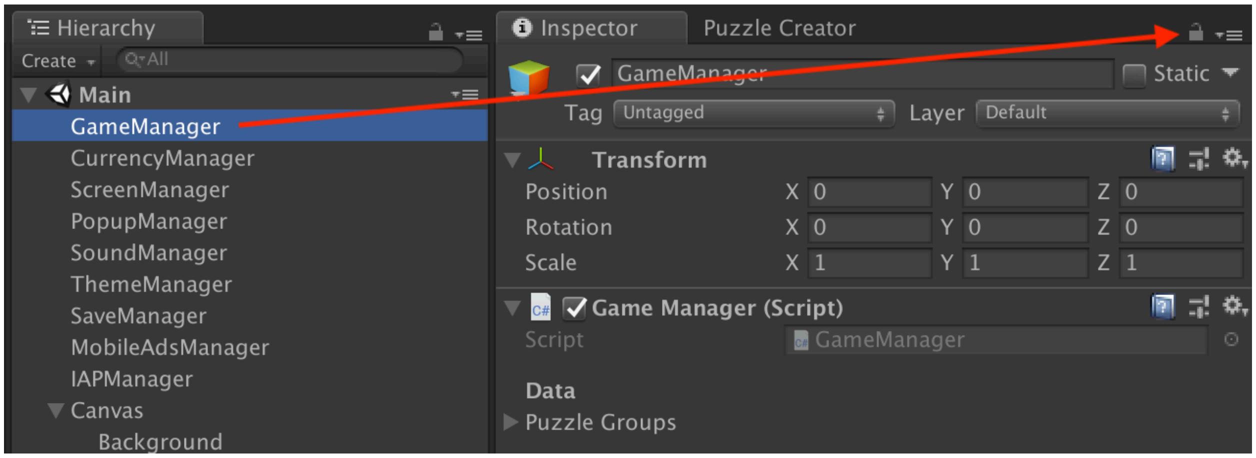
To add puzzle files generated using the Puzzle Creator window, select the **GameManager** in the Main scenes hierarchy and expand **Puzzle Groups**. Expand a puzzle group or create a new group and add the generated puzzle files to the **Puzzle Files** list:



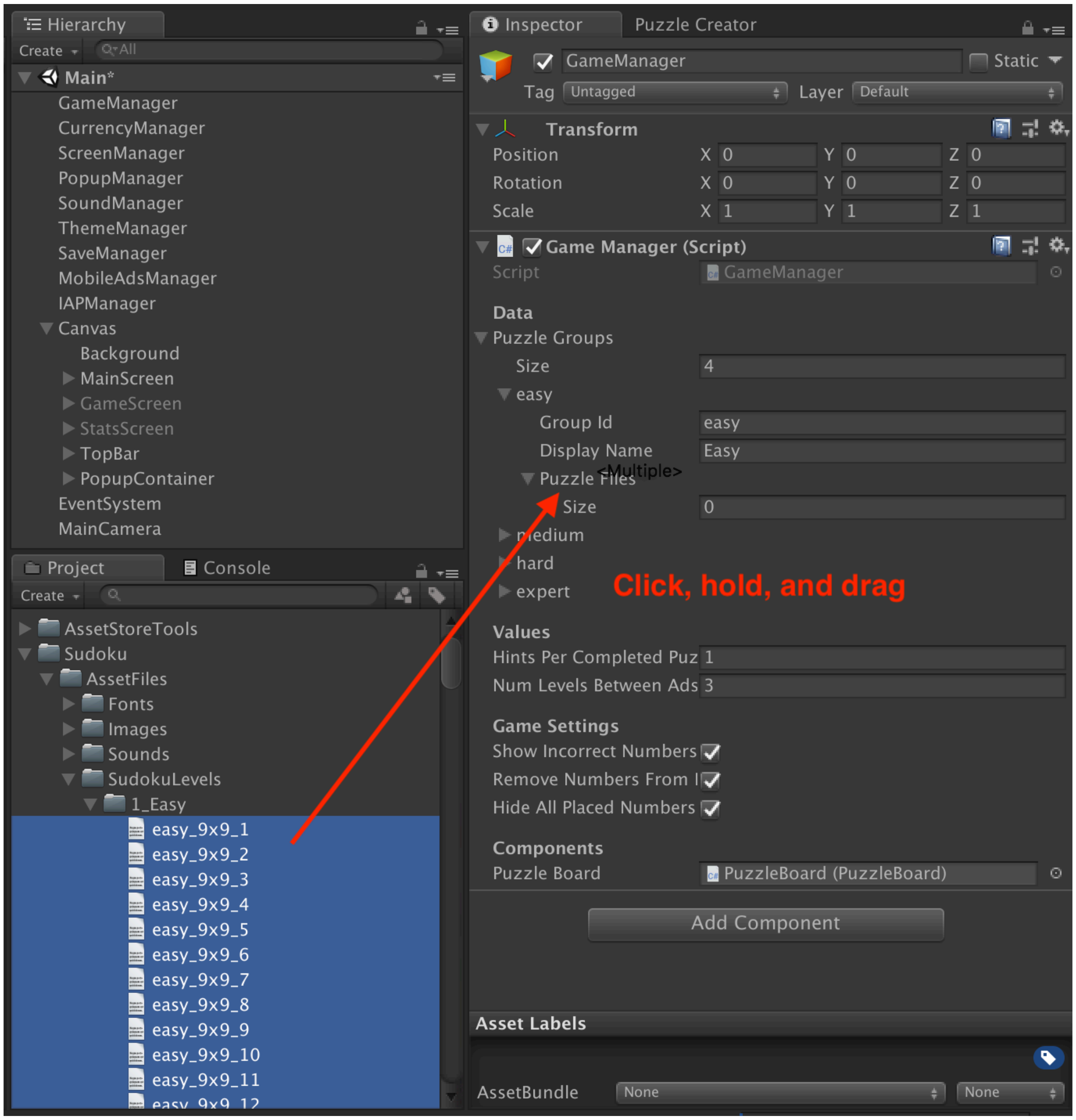
Game Manager (Script)

Script	GameManager
Data	
Puzzle Groups	
Size	4
easy	
Group Id	easy
Display Name	Easy
Puzzle Files	
Size	300
Element 0	easy_9x9_1
Element 1	easy_9x9_2
Element 2	easy_9x9_3
Element 3	easy_9x9_4
Element 4	easy_9x9_5
Element 5	easy_9x9_6
Element 6	easy_9x9_7
Element 7	easy_9x9_8
Element 8	easy_9x9_9

TIP: To add a bunch of puzzle files at once, select the GameManager then click the lock button at the top of the Inspector window:



With the Inspector locked on the GameManager, you can then select other objects without the Inspector changing away from the GameManager object. Now you can select a bunch of puzzle files from the Project window and drag them to the Puzzle Files list in one of the Puzzle Groups to add them all at once:



Main*
GameManager
CurrencyManager
ScreenManager
PopupManager
SoundManager
ThemeManager
SaveManager
MobileAdsManager
IAPManager
Canvas
Background
MainScreen
GameScreen
StatsScreen
TopBar
PopupContainer
EventSystem
MainCamera

AssetStoreTools
Sudoku
AssetFiles
Fonts
Images
Sounds
SudokuLevels
1_Easy
easy_9x9_1
easy_9x9_2
easy_9x9_3
easy_9x9_4
easy_9x9_5
easy_9x9_6
easy_9x9_7
easy_9x9_8
easy_9x9_9
easy_9x9_10
easy_9x9_11
easy_9x9_12

GameManager ✓ Static
Tag Untagged Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

✓ Game Manager (Script)
Script GameManager

Data
Puzzle Groups
Size 4
easy
Group Id easy
Display Name Easy
Puzzle Files <Multiple>
Size 0
medium
hard
expert

Values
Hints Per Completed Puz 1
Num Levels Between Ads 3

Game Settings
Show Incorrect Numbers ✓
Remove Numbers From I ✓
Hide All Placed Numbers ✓

Components
Puzzle Board PuzzleBoard (PuzzleBoard)

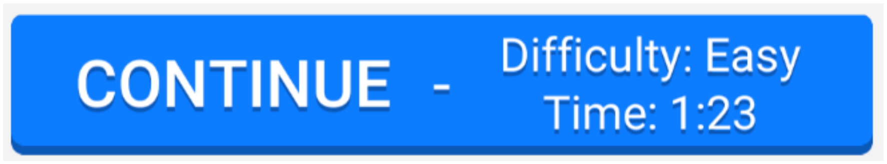
AssetBundle None None

Project

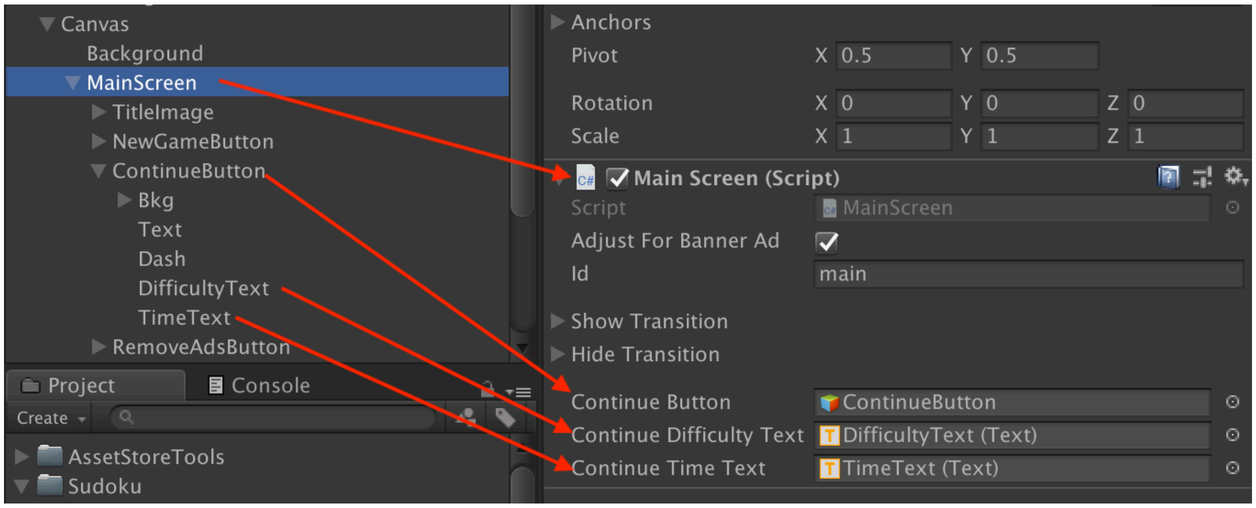
This section will show you where UI elements that are changed / instantiated by scripts are located and how to edit them.

NOTE: All Image and Text components colors are set by the ThemeGraphicBehaviour component when the game runs. If you want to change colors you will need to change them in the ThemeManager. Check the Themes section of this Documentation for more information.

Main Screen - Continue Button



The continue button on the main screen is controlled by the **MainScreen** script attached to the MainScreen GameObject.

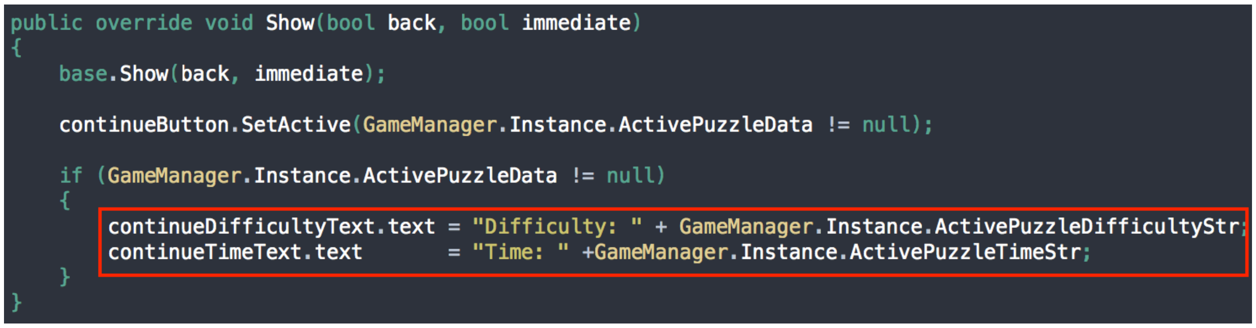


```
▼ Canvas
    Background
  ▼ MainScreen
    ▶ TitleImage
    ▶ NewGameButton
    ▼ ContinueButton
      ▶ Bkg
        Text
        Dash
        DifficultyText
        TimeText
    ▶ RemoveAdsButton
▶ AssetStoreTools
▼ Sudoku
```

```
▶ Anchors
  Pivot                     X 0.5   Y 0.5
  Rotation                  X 0     Y 0     Z 0
  Scale                     X 1     Y 1     Z 1
☑ Main Screen (Script)
  Script                    MainScreen
  Adjust For Banner Ad      ✓
  Id                        main
▶ Show Transition
▶ Hide Transition
  Continue Button           ContinueButton
  Continue Difficulty Text  DifficultyText (Text)
  Continue Time Text        TimeText (Text)
```

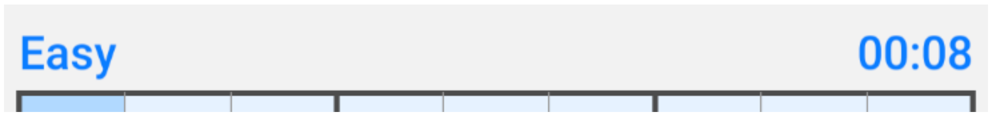
When there is a puzzle that can be continued the MainScreen will set it to active and if there is no puzzle to continue it is set to de-active.

The **DifficultyText** and **TimeText** are also set by the MainScreen script in the **Show** method:

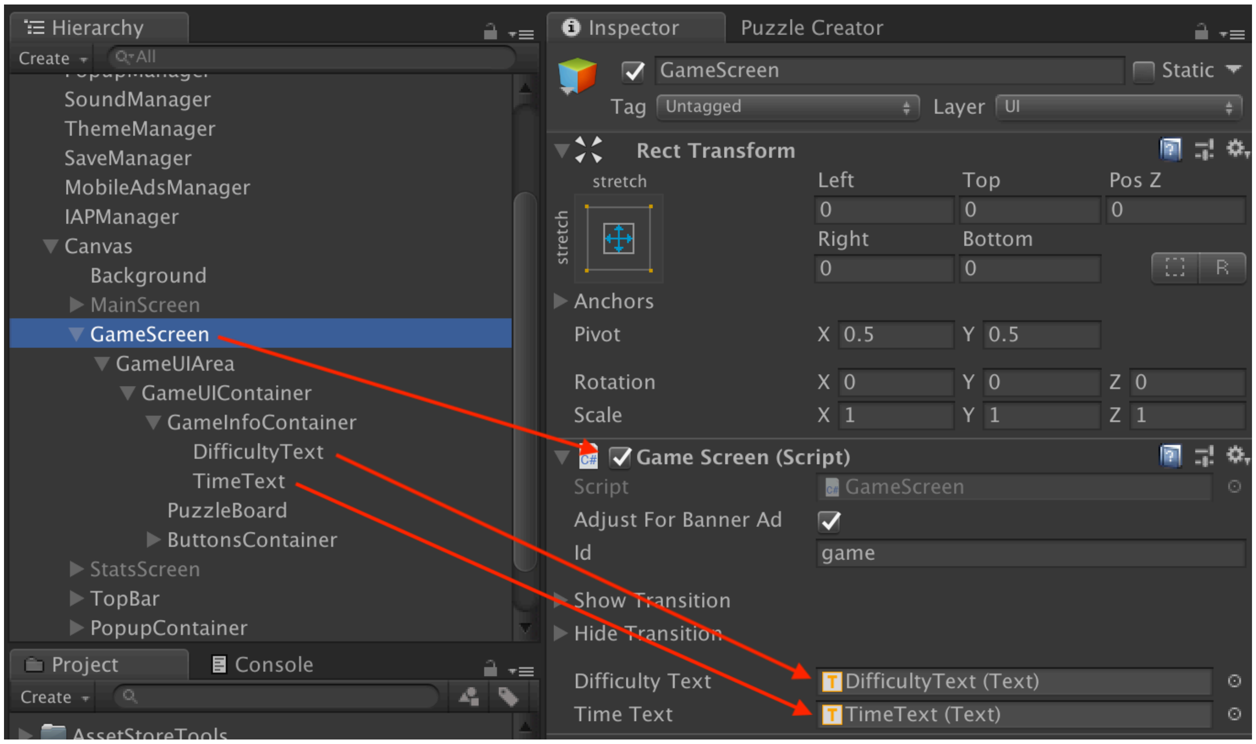


```
public override void Show(bool back, bool immediate)
{
    base.Show(back, immediate);

    continueButton.SetActive(GameManager.Instance.ActivePuzzleData != null);

    if (GameManager.Instance.ActivePuzzleData != null)
    {
        continueDifficultyText.text = "Difficulty: " + GameManager.Instance.ActivePuzzleDifficultyStr;
        continueTimeText.text       = "Time: " +GameManager.Instance.ActivePuzzleTimeStr;
    }
}
```

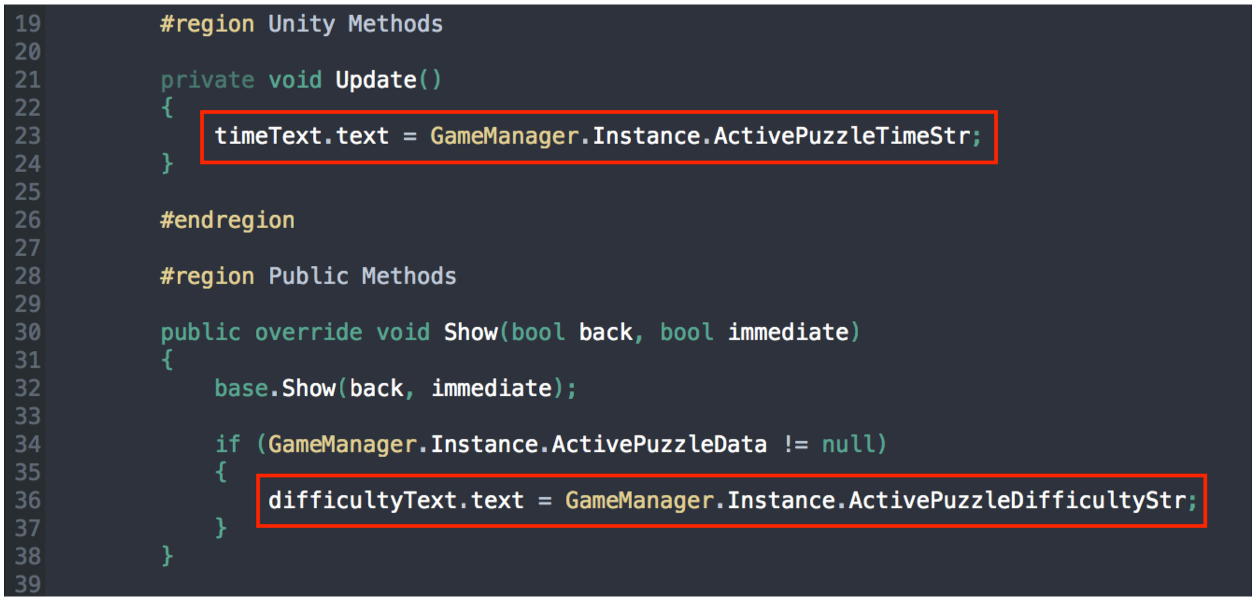
Game Screen - Difficulty / Time Text



The Difficulty and Time text that appear above the puzzle board on the GameScreen are set by the **GameScreen** script:

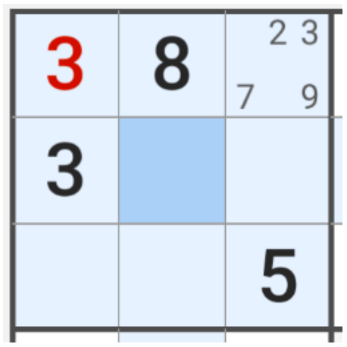


The **Time Text** is updated every frame in the Update loop and the **Difficulty Text** is set when the screen shows:

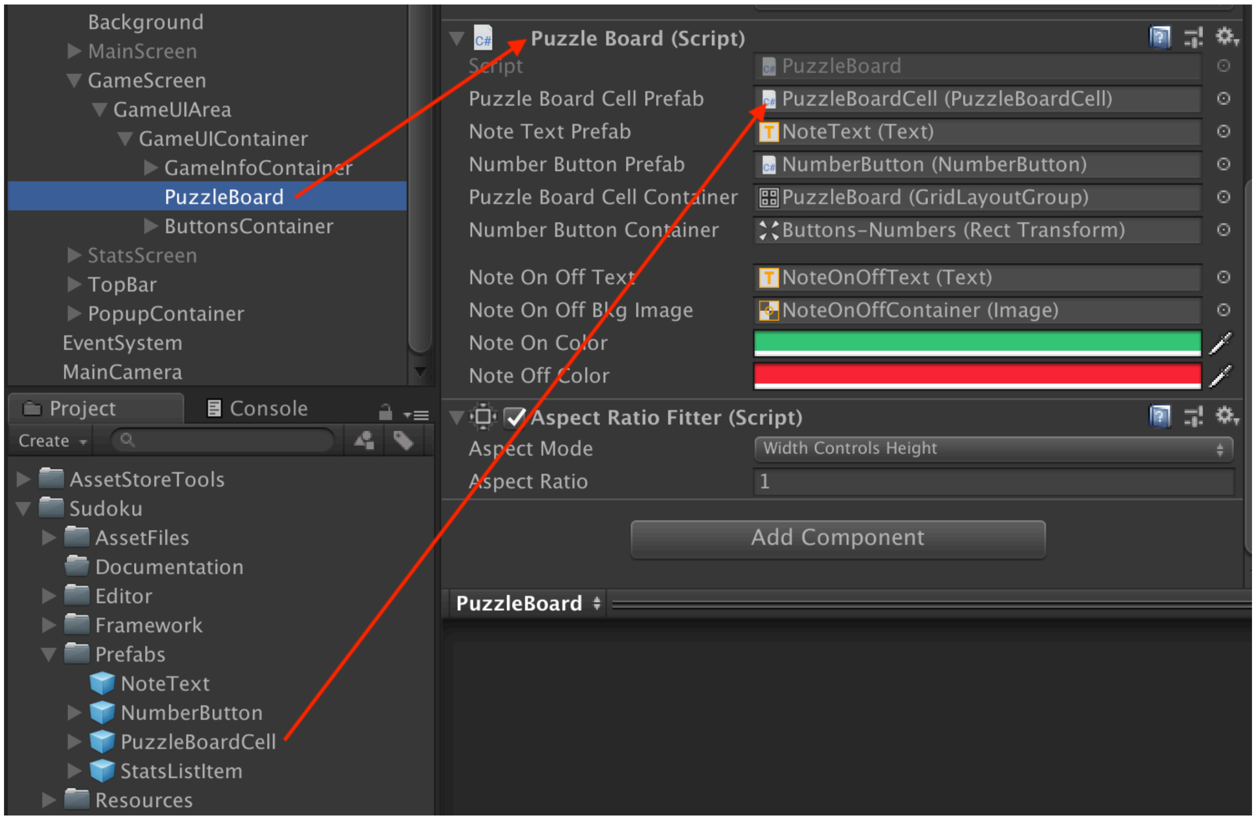


```
        #region Unity Methods

        private void Update()
        {
            timeText.text = GameManager.Instance.ActivePuzzleTimeStr;
        }

        #endregion

        #region Public Methods

        public override void Show(bool back, bool immediate)
        {
            base.Show(back, immediate);

            if (GameManager.Instance.ActivePuzzleData != null)
            {
                difficultyText.text = GameManager.Instance.ActivePuzzleDifficultyStr;
            }
        }

```

Puzzle Board - Cells



The **PuzzleBoard** script is responsible for creating and putting together the cells that make up the puzzle board. When a game starts a copy of the **Puzzle Board Cell Prefab** is instantiated for each cell in the puzzle (So for a 9x9 puzzle there will be 81 cells). The **PuzzleBoardCell** prefab located in the **Prefabs** folder is used in the asset.



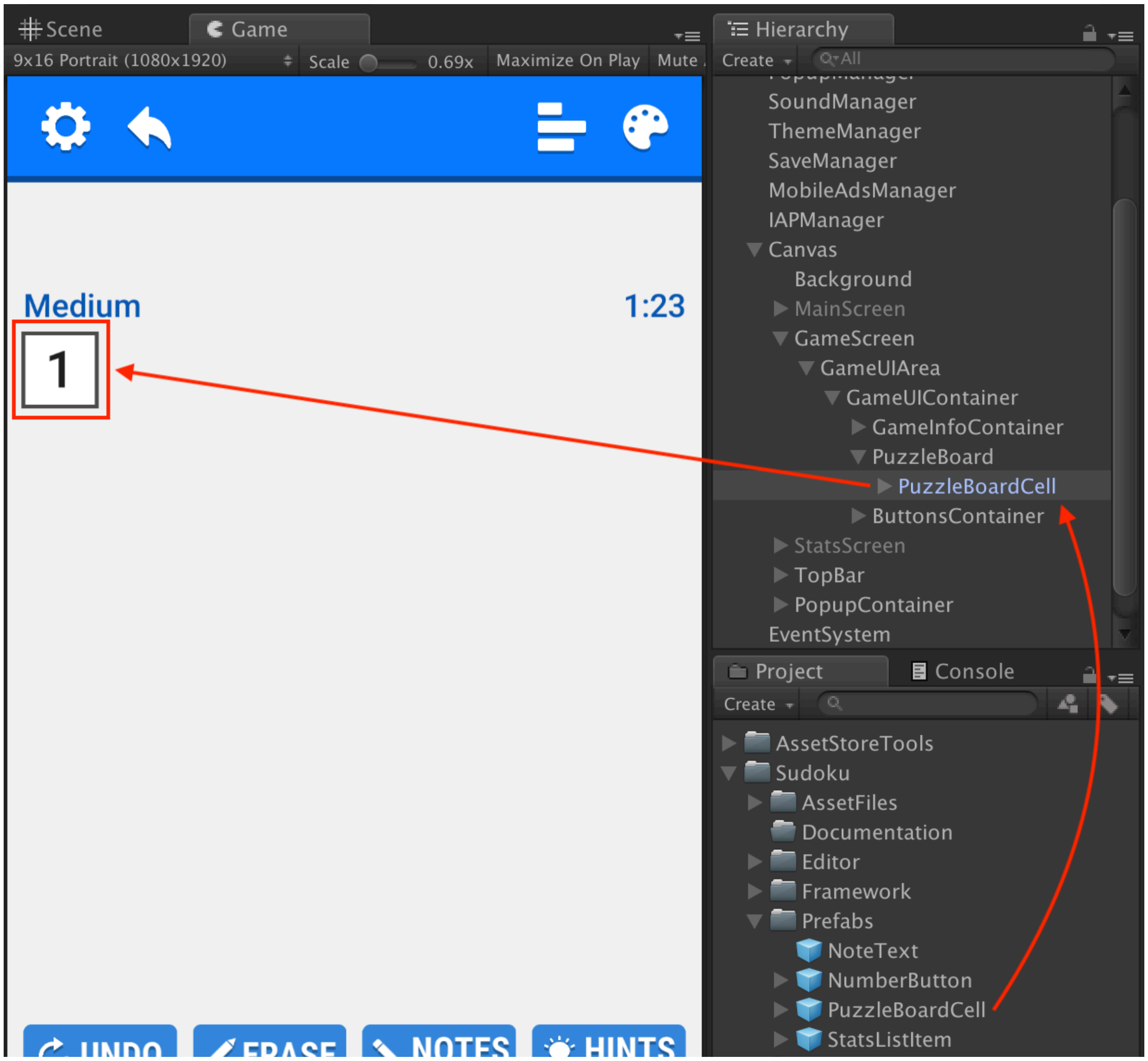
Puzzle Board (Script)

Script	PuzzleBoard
Puzzle Board Cell Prefab	PuzzleBoardCell (PuzzleBoardCell)
Note Text Prefab	NoteText (Text)
Number Button Prefab	NumberButton (NumberButton)
Puzzle Board Cell Container	PuzzleBoard (GridLayoutGroup)
Number Button Container	Buttons-Numbers (Rect Transform)
Note On Off Text	NoteOnOffText (Text)
Note On Off Bkg Image	NoteOnOffContainer (Image)
Note On Color	🟩
Note Off Color	🟥

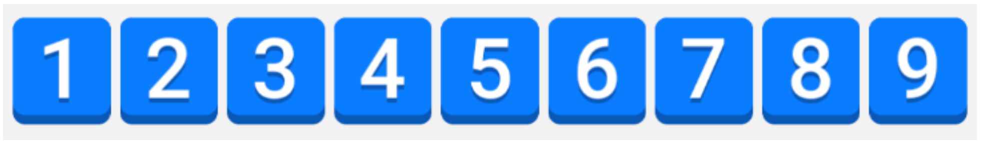
Aspect Ratio Fitter (Script)

Aspect Mode	Width Controls Height
Aspect Ratio	1

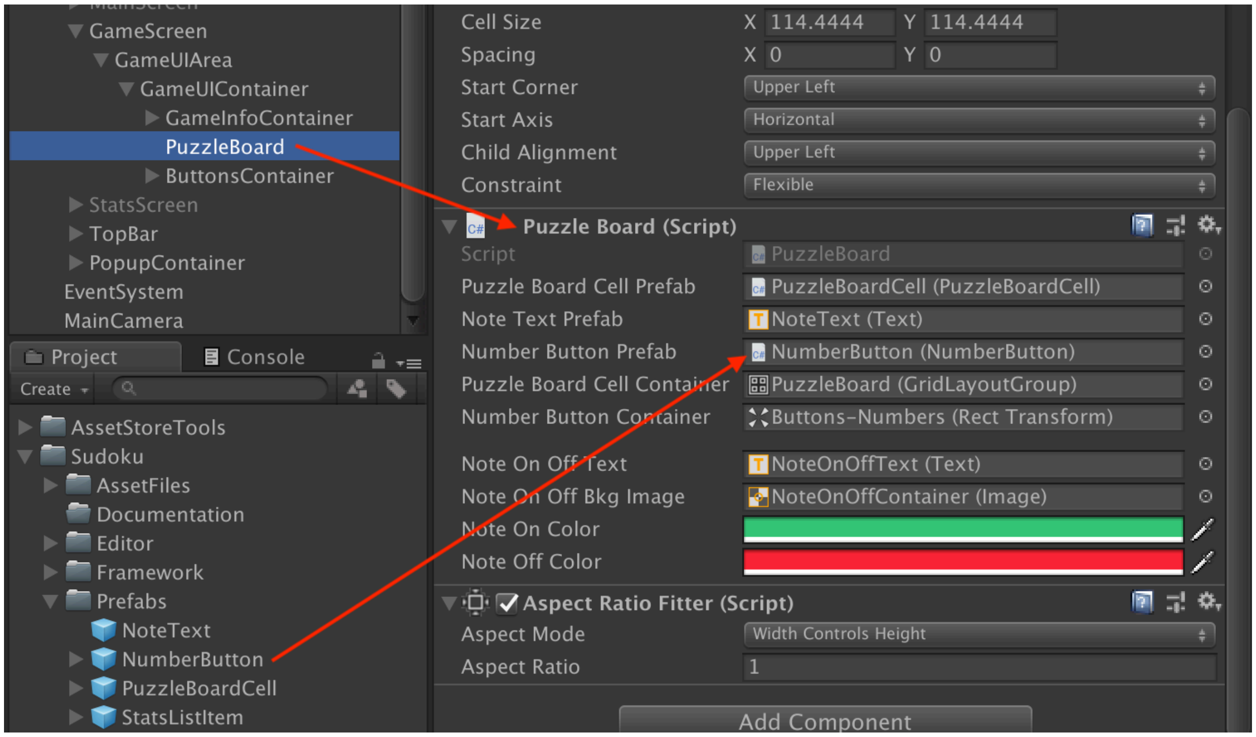
To edit the PuzzleBoardCell, drag it under the PuzzleBoard GameObject:



Game Screen - Numbers



The **PuzzleBoard** script is responsible for creating the number buttons that appear under the puzzle board. When a game starts a copy of the **Number Button Prefab** is instantiated for each number needed in the puzzle (So for a 9x9 puzzle there will be 9 buttons). The **NumberButton** prefab located in the **Prefabs** folder is used in the asset.



GameScreen
- GameUIArea
  - GameUIContainer
    - GameInfoContainer
    - PuzzleBoard
    - ButtonsContainer
- StatsScreen
- TopBar
- PopupContainer
- EventSystem
- MainCamera

Project | Console

- AssetStoreTools
- Sudoku
  - AssetFiles
  - Documentation
  - Editor
  - Framework
  - Prefabs
    - NoteText
    - NumberButton
    - PuzzleBoardCell
    - StatsListItem

Cell Size	X 114.4444 Y 114.4444
Spacing	X 0 Y 0
Start Corner	Upper Left
Start Axis	Horizontal
Child Alignment	Upper Left
Constraint	Flexible

Puzzle Board (Script)

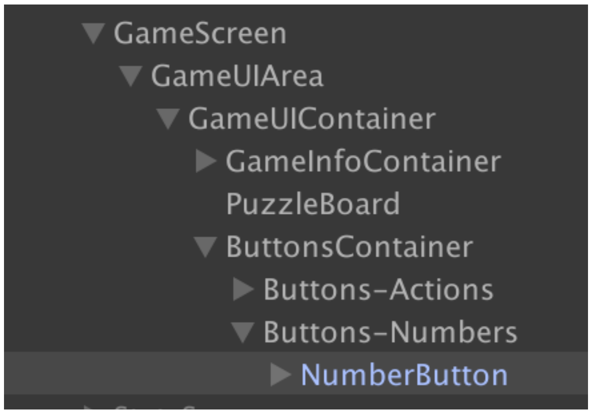
Script	PuzzleBoard
Puzzle Board Cell Prefab	PuzzleBoardCell (PuzzleBoardCell)
Note Text Prefab	NoteText (Text)
Number Button Prefab	NumberButton (NumberButton)
Puzzle Board Cell Container	PuzzleBoard (GridLayoutGroup)
Number Button Container	Buttons-Numbers (Rect Transform)
Note On Off Text	NoteOnOffText (Text)
Note On Off Bkg Image	NoteOnOffContainer (Image)
Note On Color	🟩
Note Off Color	🟥

Aspect Ratio Fitter (Script)

Aspect Mode	Width Controls Height
Aspect Ratio	1

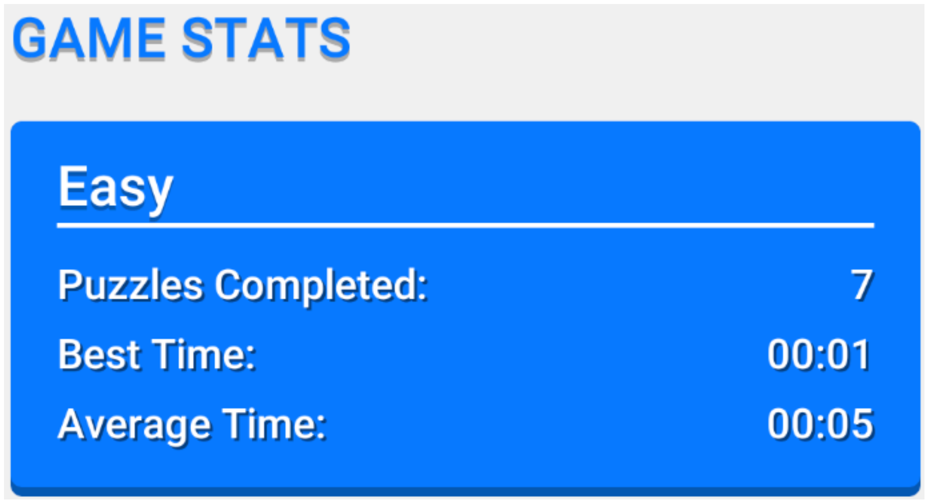
Add Component

To edit the NumberButton, drag it into the **Buttons-Numbers** GameObject



- GameScreen
  - GameUIArea
    - GameUIContainer
      - GameInfoContainer
      - PuzzleBoard
      - ButtonsContainer
        - Buttons-Actions
        - Buttons-Numbers
          - NumberButton

Stats Screen - Game Stats List Item



GAME STATS

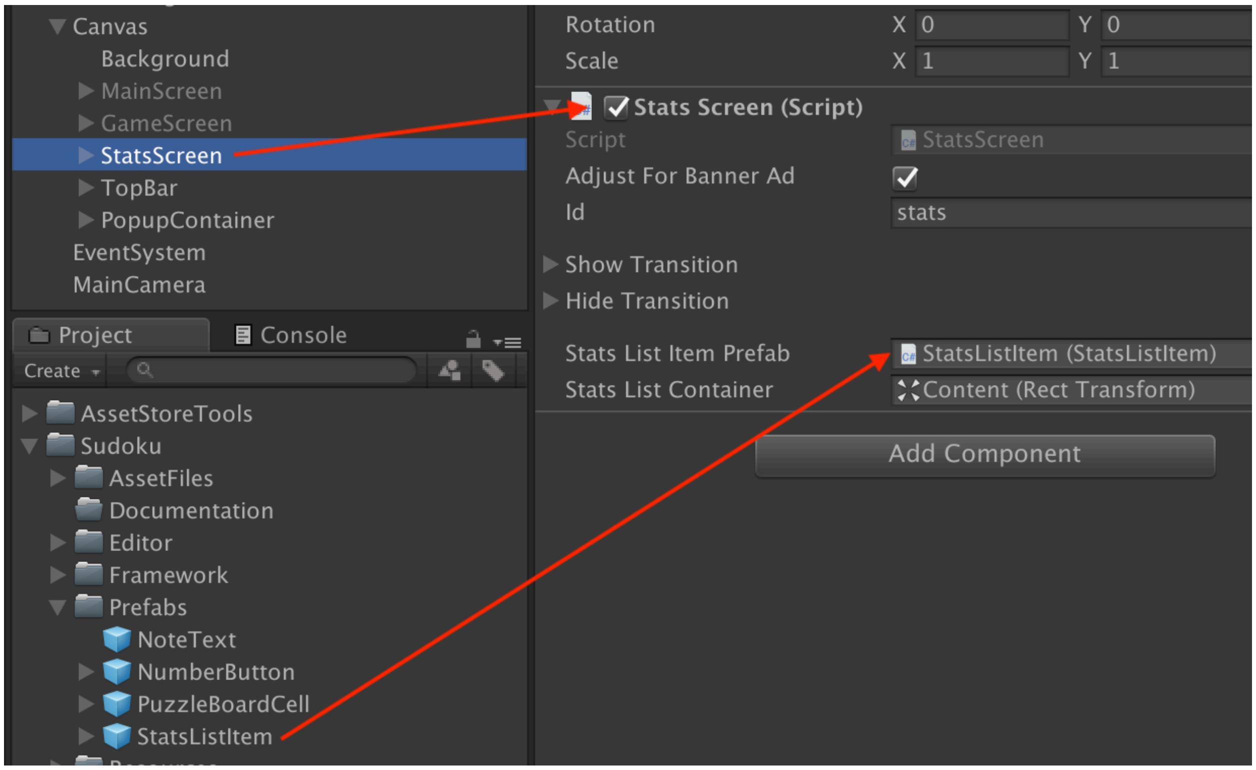
Easy

Puzzles Completed: 7

Best Time: 00:01

Average Time: 00:05

The StatsScreen script attached to the StatsScreen GameObject is responsible for instantiating copies of the **Stats List Item Prefab** for each Puzzle Group that exists in the GameManager Puzzle Groups list. The **StatsListItem** prefab located in the **Prefabs** folder is used in the asset:



Stats Screen (Script)

Script	StatsScreen
Adjust For Banner Ad	✓
Id	stats
Stats List Item Prefab	StatsListItem (StatsListItem)
Stats List Container	Content (Rect Transform)

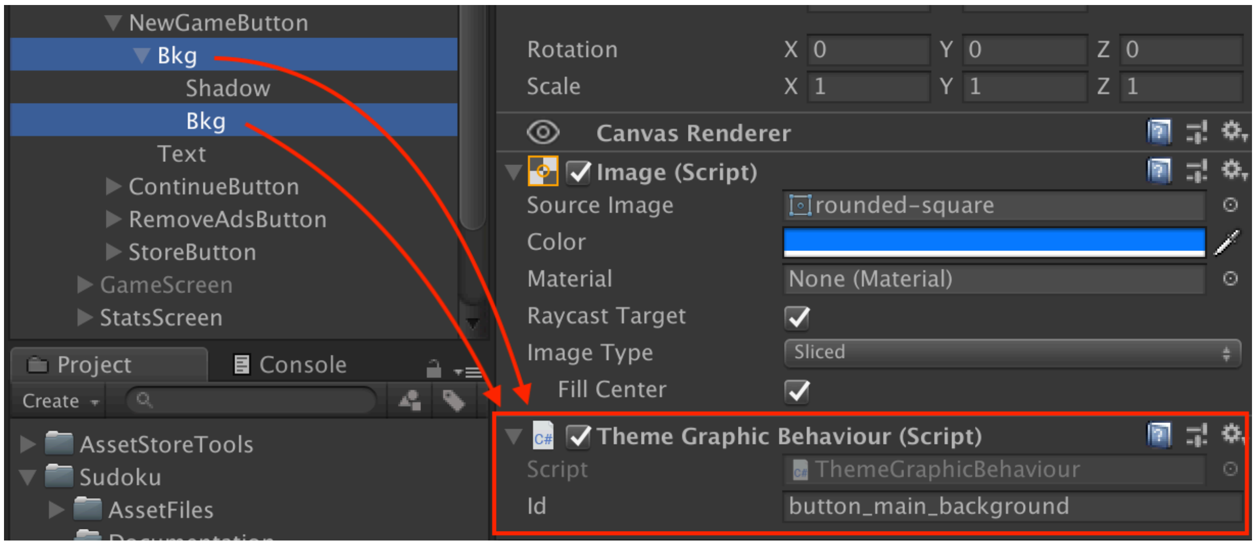
Themes

The **ThemeManager** is responsible for editing / adding new color themes.

Changing Colors In The Asset

All Image and Text components colors are controlled using the ThemeManager and ThemeGraphicBehaviour component. If you would like to change the color of certain Image/ Text components you will have to change the color on the ThemeManager. To know what color to change check the **Id** that is set on the **ThemeGraphicBehaviour** component.

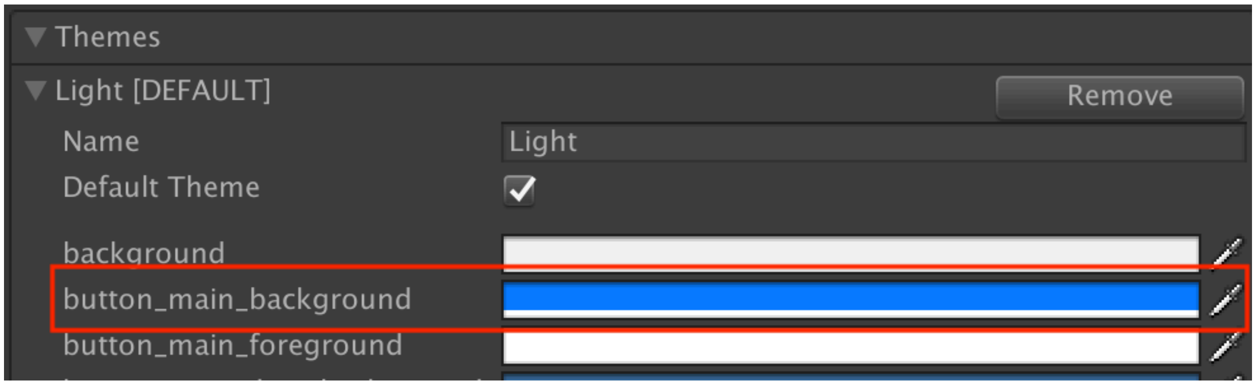
For example, if we wanted to changed the color of the “NEW GAME” button on the main screen, select the **Bkg** GameObjects that have the Image components on them and we see that the Id on the Theme Graphic Behaviour is set to **button_main_background**:



```
▼ NewGameButton
    ▼ Bkg
        Shadow
        Bkg
      Text
  ▶ ContinueButton
  ▶ RemoveAdsButton
  ▶ StoreButton
▶ GameScreen
▶ StatsScreen
```

```
Rotation          X 0    Y 0    Z 0
Scale             X 1    Y 1    Z 1
Canvas Renderer
Image (Script)
  Source Image    rounded-square
  Color
  Material        None (Material)
  Raycast Target  ✓
  Image Type      Sliced
    Fill Center   ✓
Theme Graphic Behaviour (Script)
  Script          ThemeGraphicBehaviour
  Id              button_main_background
```

Then over on the **ThemeManager**, expand the Themes box and locate the button_main_background ids to change the background color of the button:

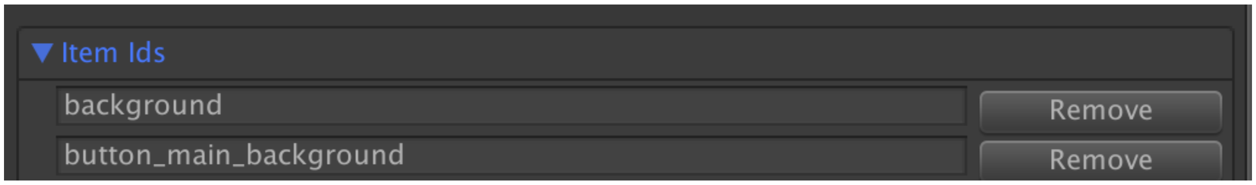


```
▼ Themes
▼ Light [DEFAULT]                      Remove
    Name                     Light
    Default Theme            ✓

    background
    button_main_background
    button_main_foreground
```

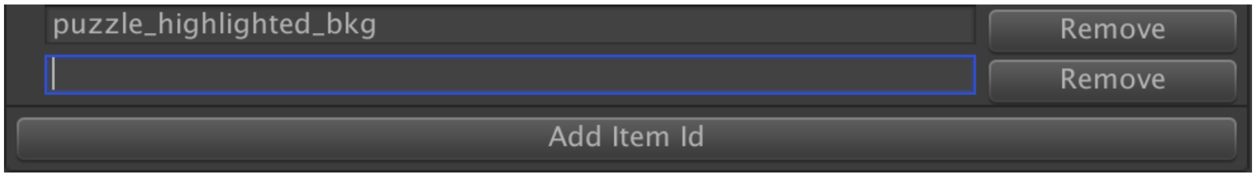
Adding New Theme Color Items

All the button colors are controlled by the button_main_background id. So if you wanted to change the color of just the “NEW GAME” button on the main screen and no other buttons you would have to create a new color id.

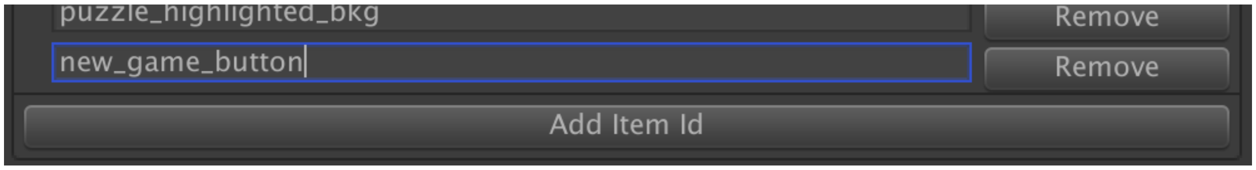
To do so, first create a new Id in the **Item Ids** list on the **ThemeManager**. Expand the Item Ids list:



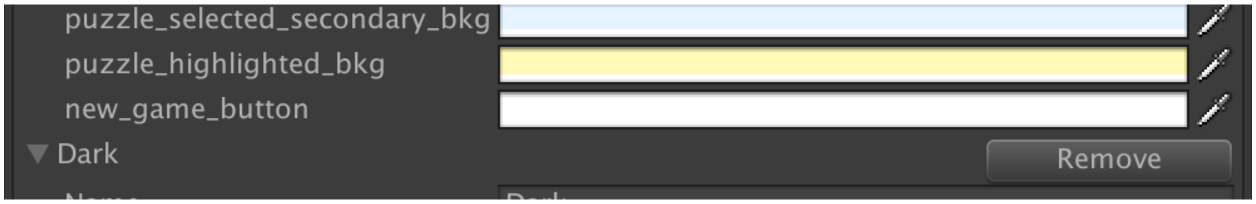
Then click the **Add Item Id** button at the bottom of the list:



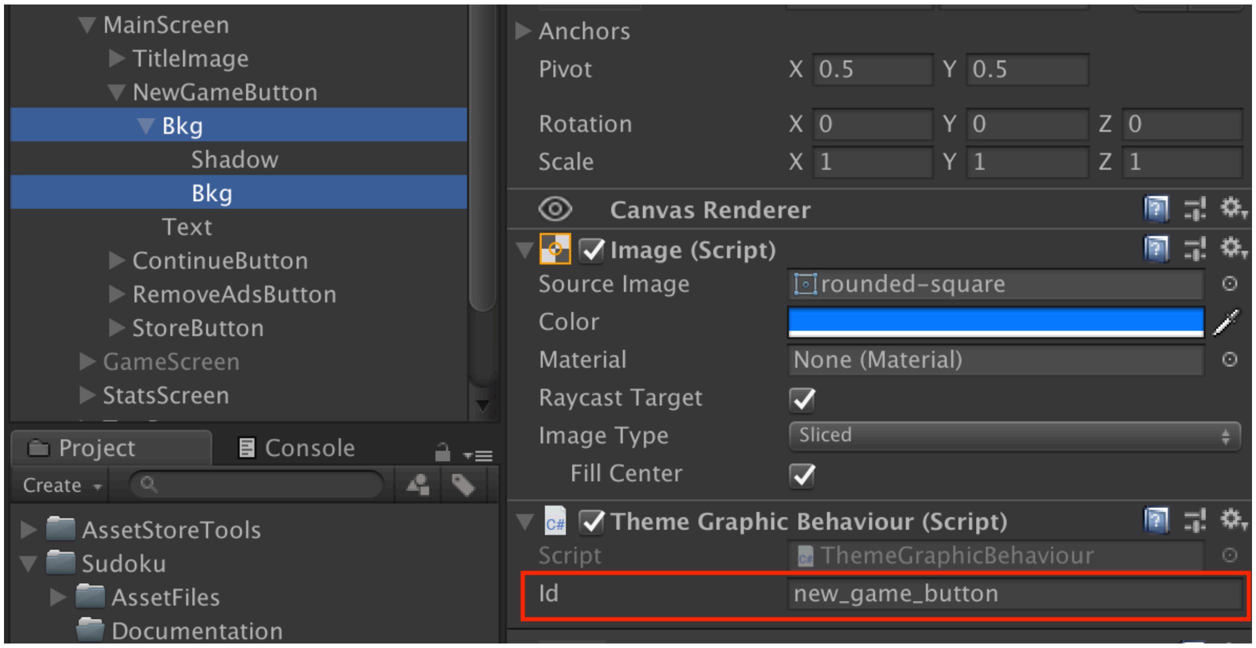
We will set the new id to **new_game_button**:



A new color item will appear in each theme with the new_game_button id:



Set the color of the new_game_button item for each theme. Then select the **Bkg** GameObjects in the MainScreen under the **NewGameButton** object and change the **Id** on the ThemeBehaviourComponent to the new_game_button id we just created:



MainScreen
TitleImage
NewGameButton
Bkg
Shadow
Bkg
Text
ContinueButton
RemoveAdsButton
StoreButton
GameScreen
StatsScreen

Project Console
Create

AssetStoreTools
Sudoku
AssetFiles
Documentation

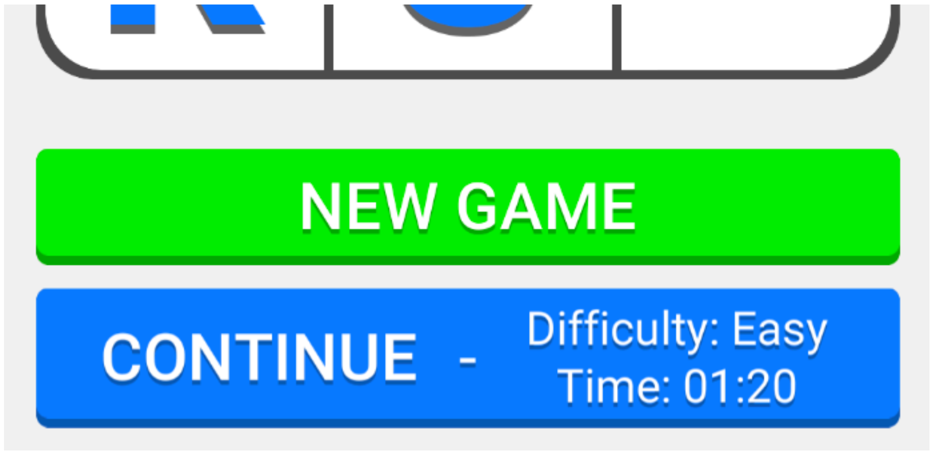
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer

Image (Script)
Source Image rounded-square
Color
Material None (Material)
Raycast Target ✓
Image Type Sliced
Fill Center ✓

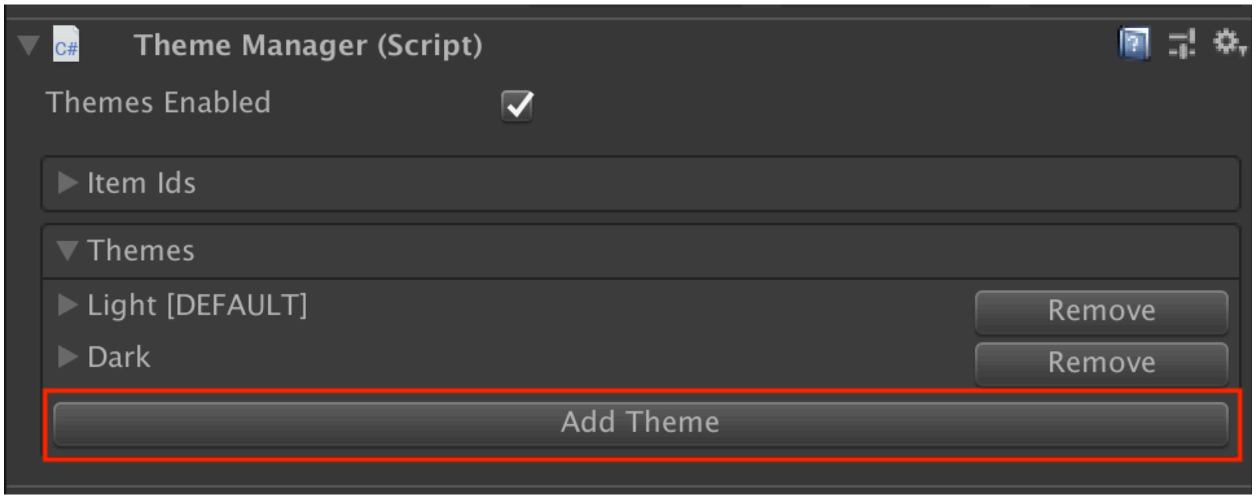
Theme Graphic Behaviour (Script)
Script ThemeGraphicBehaviour
Id new_game_button

Now when you run the game the button will be set to the new color:

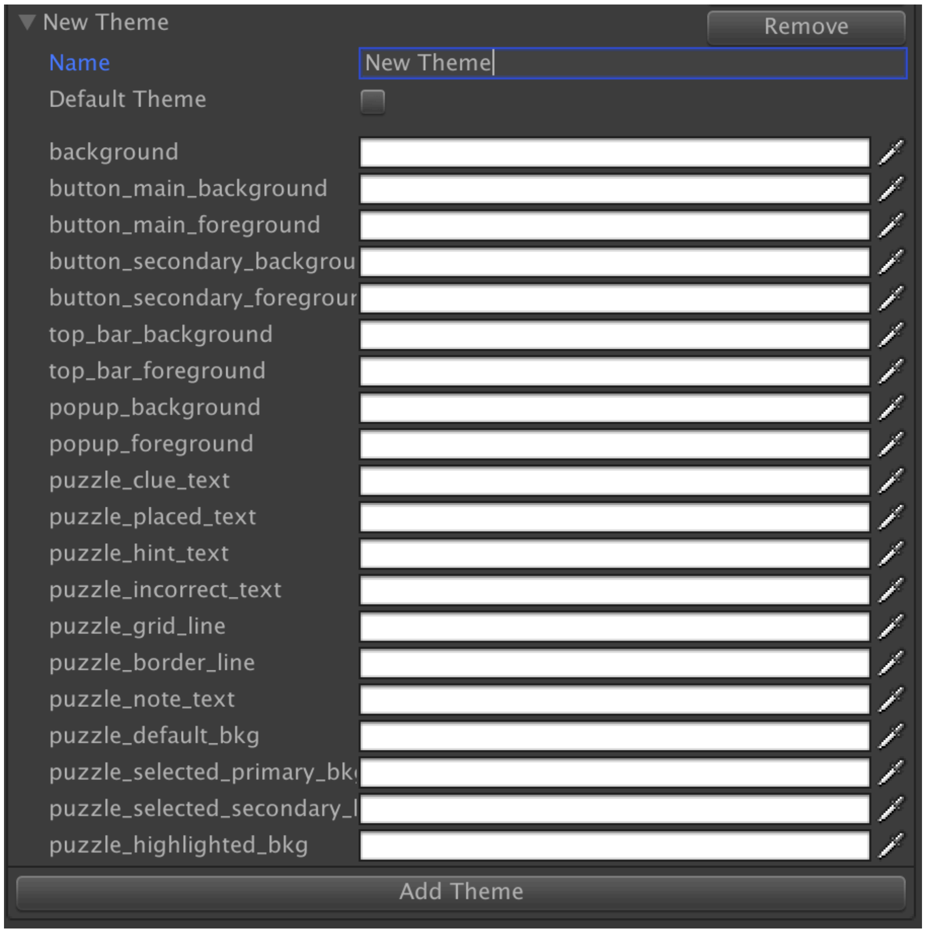


Adding New Themes

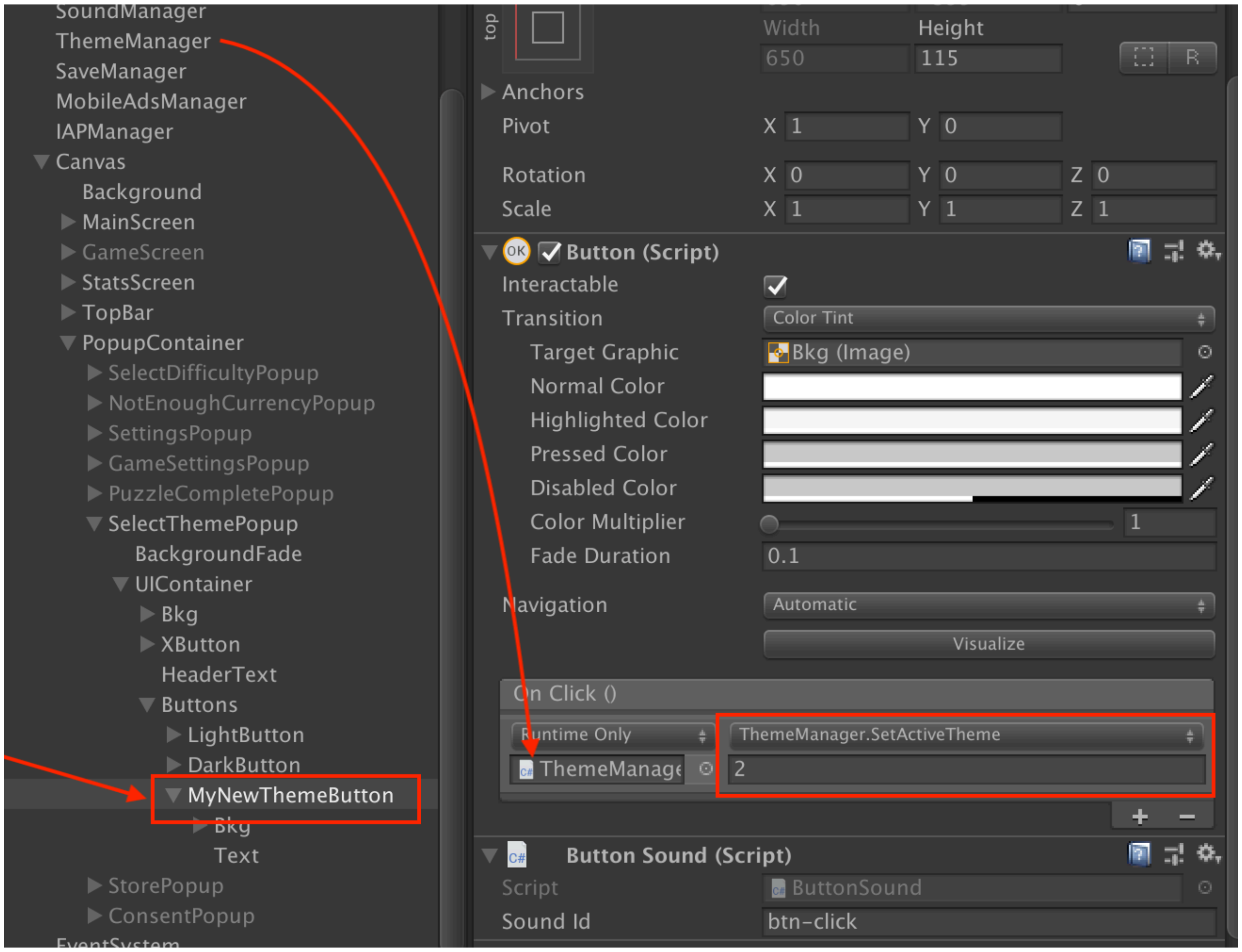
To create a new theme simply select the **Add Theme** button on the **ThemeManager:**



You will then need to set the color of all the existing theme color items or everything will be set to white when using that theme.



Once you have set all the colors you will need a way for the player to select the theme. Add a new button to the **SelectThemePopup.** You can then have button call the **SetActiveTheme** method on the **ThemeManager** and pass it the index of the theme you want to set active when the button is clicked:



```
SoundManager
ThemeManager
SaveManager
MobileAdsManager
IAPManager
▼ Canvas
    Background
  ▶ MainScreen
  ▶ GameScreen
  ▶ StatsScreen
  ▶ TopBar
  ▼ PopupContainer
    ▶ SelectDifficultyPopup
    ▶ NotEnoughCurrencyPopup
    ▶ SettingsPopup
    ▶ GameSettingsPopup
    ▶ PuzzleCompletePopup
    ▼ SelectThemePopup
        BackgroundFade
      ▼ UIContainer
        ▶ Bkg
        ▶ XButton
          HeaderText
        ▼ Buttons
          ▶ LightButton
          ▶ DarkButton
          ▼ MyNewThemeButton
            ▶ Bkg
              Text
    ▶ StorePopup
    ▶ ConsentPopup
EventSystem
```

Width	650
Height	115
Anchors	
Pivot	X 1 Y 0
Rotation	X 0 Y 0 Z 0
Scale	X 1 Y 1 Z 1

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 ()

Runtime Only	ThemeManager.SetActiveTheme
ThemeManage	2

Button Sound (Script)

Script	ButtonSound
Sound Id	btn-click

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

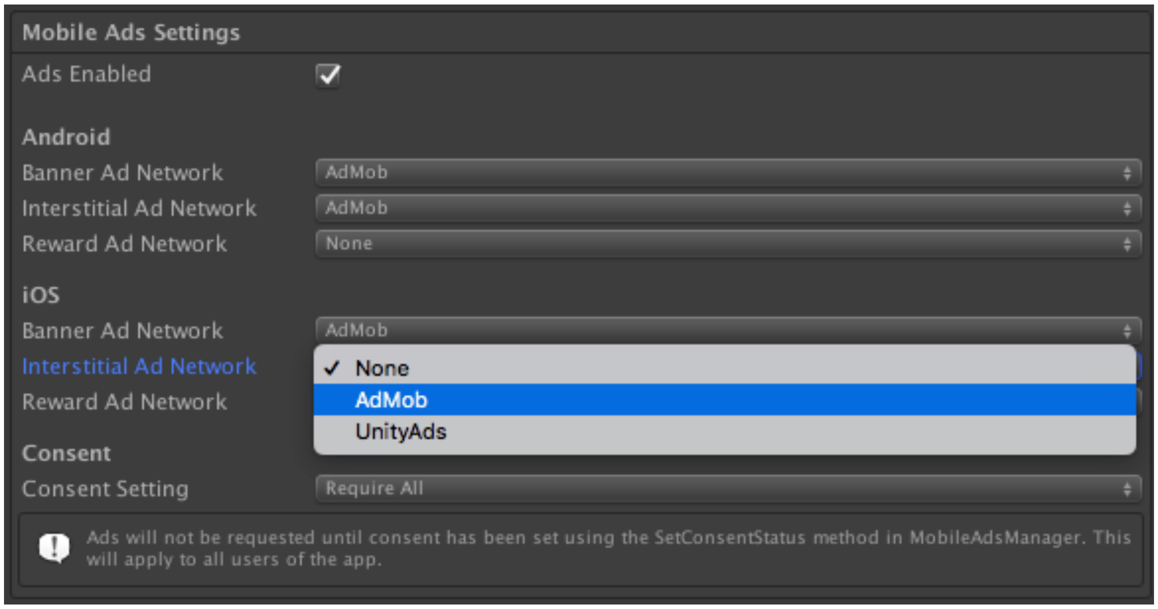
Ads are setup using the **Mobile Ads Settings** window which can be opened by selecting the menu item **Window -> Mobile Ads Settings** (Or clicking the button on the MobileAdsManager inspector).

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.

NO FILL ERROR - If you have setup AdMob and you are getting the **no fill** error message when trying to load ads it means the AdMob server does not have any available ad inventory to give your application. This usually happens with a new AdMob account / app. After a couple days your app should start receiving ads. This error is **not** an implementation error and **cannot** be solved with any code changes. If after 7 days your app is still not receiving ads you will have to contact AdMob support to rectify the issue.

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

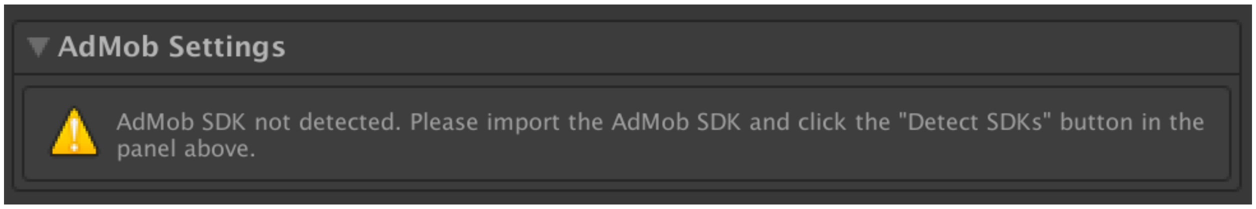
Reward Ad Network

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:

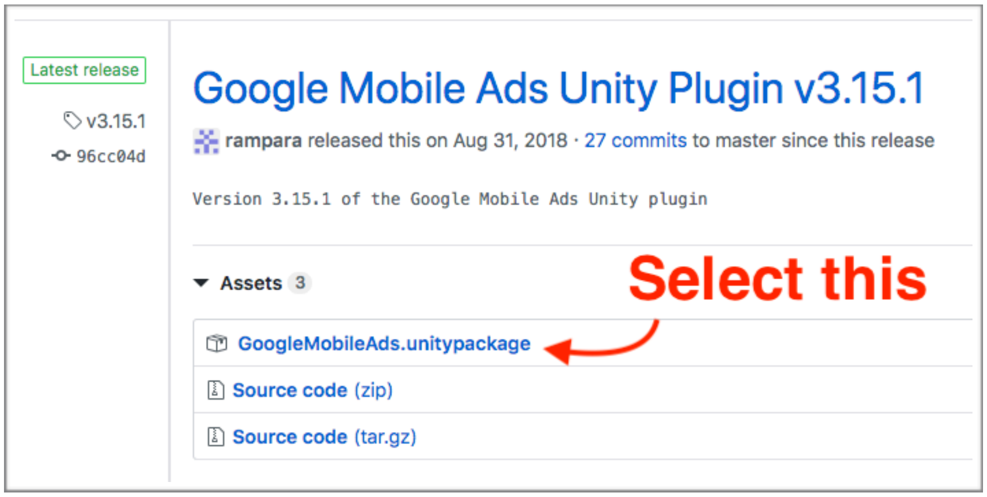


AdMob Settings

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:

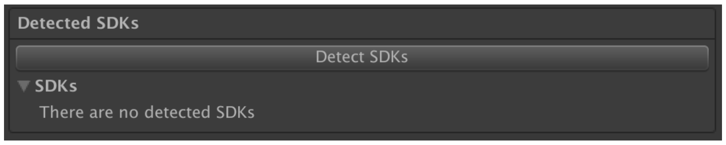


Google Mobile Ads Unity Plugin v3.15.1

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

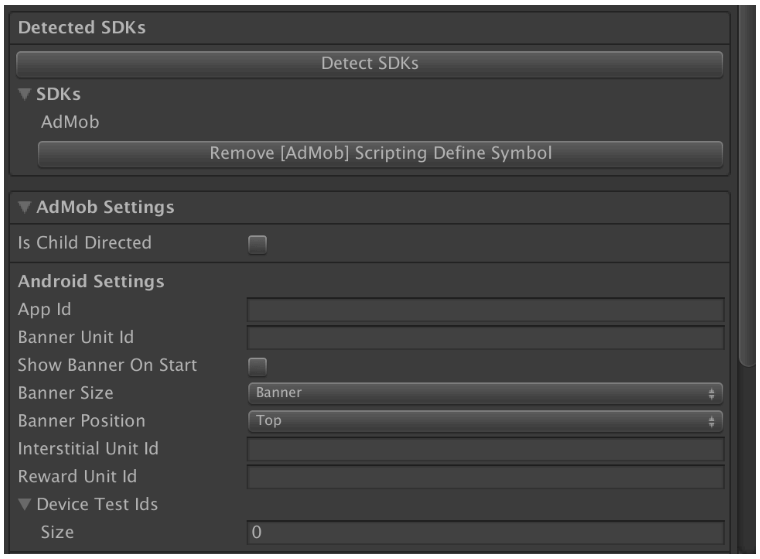
```
Version 3.15.1 of the Google Mobile Ads Unity plugin
```

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



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



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

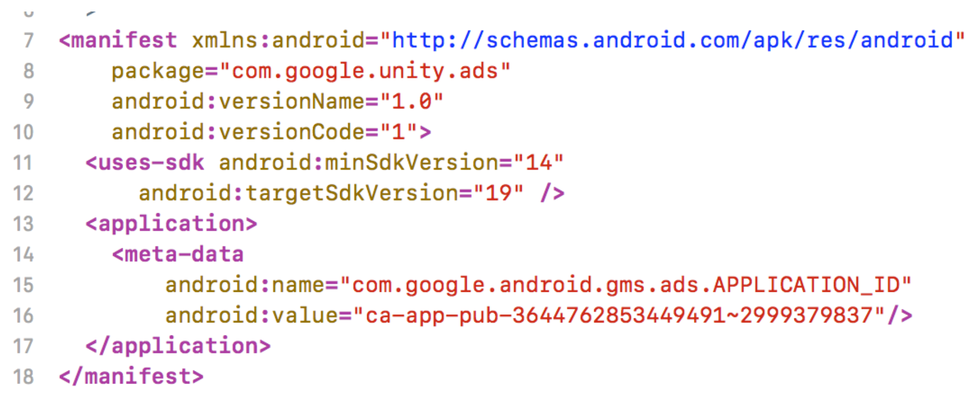
Step5 [Android Only]. Make sure the play services resolver that comes with the GoogleMobileAds plugin has executed by selecting the menu item **Assets -> Play Services Resolver -> Android Resolver -> Resolve**.

***** If using AdMob SDK version 3.17 or below *****

Step6 [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"/>
```

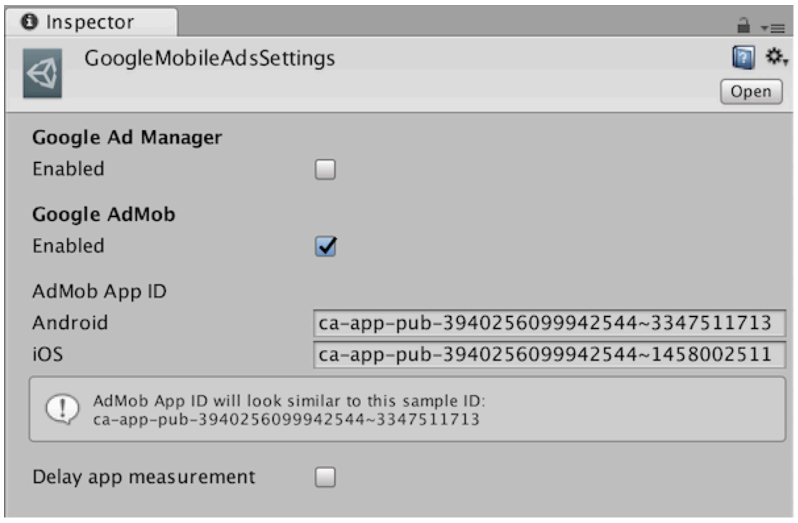
Your AndroidManifest should look something like this:



```
<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>
```

***** If using AdMob SDK version 3.18 or above *****

Step6 [Android only]. Select **Assets > Google Mobile Ads > Settings** from the menu. Enable AdMob by clicking Enabled checkbox under Google AdMob section. Then enter your Android and iOS AdMob app ID in each field.



Google Ad Manager
Enabled ☐

Google AdMob
Enabled ✓

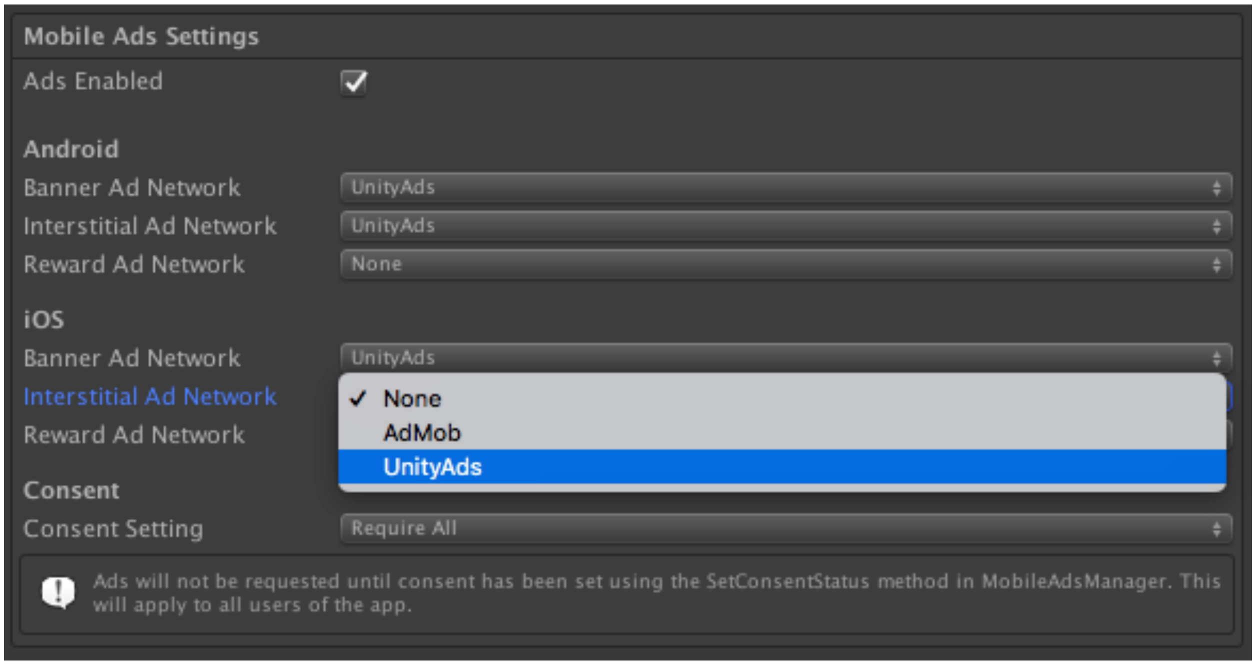
AdMob App ID
Android ca-app-pub-3940256099942544~3347511713
iOS ca-app-pub-3940256099942544~1458002511

AdMob App ID will look similar to this sample ID:
ca-app-pub-3940256099942544~3347511713

Delay app measurement ☐

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

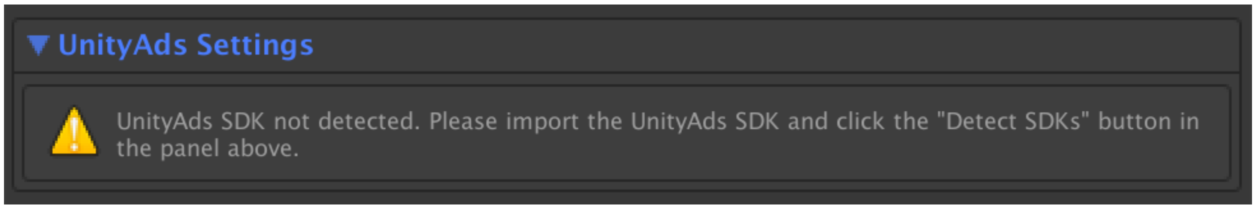
Reward Ad Network

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.

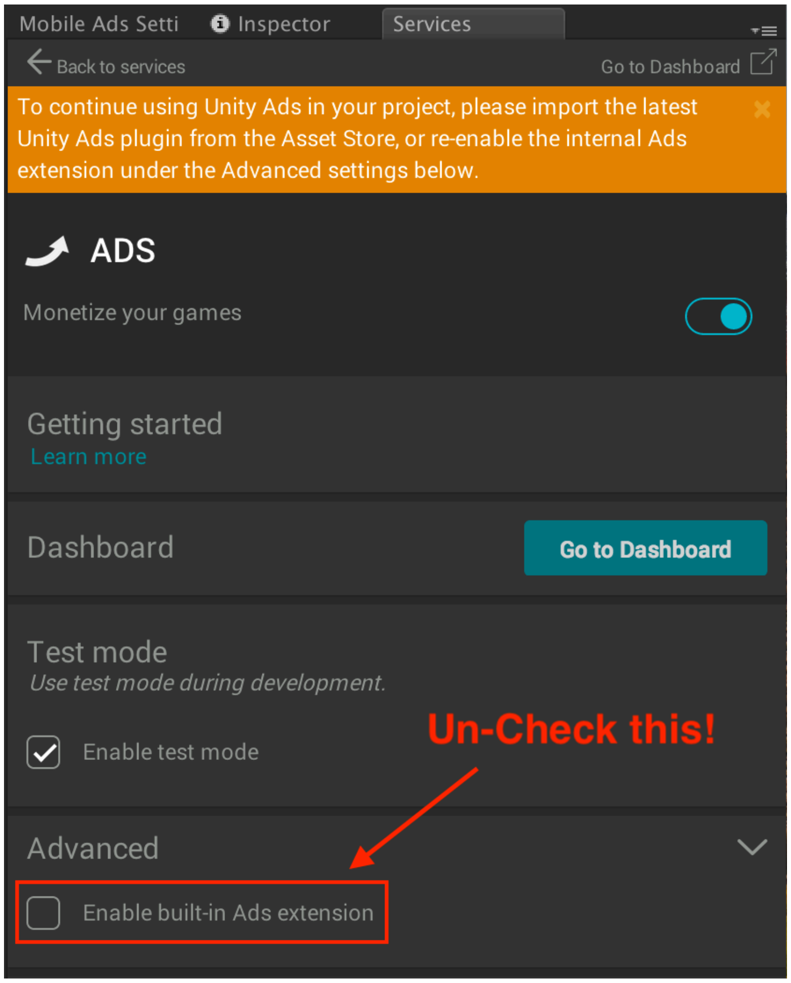
A new section will appear at the bottom of the window called **UnityAds Settings**. Expanding it now will display the following warning:


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


Mobile Ads Setti | Inspector | Services

Back to services | Go to Dashboard

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.

ADS

Monetize your games

Getting started
Learn more

Dashboard | Go to Dashboard

Test mode
Use test mode during development.

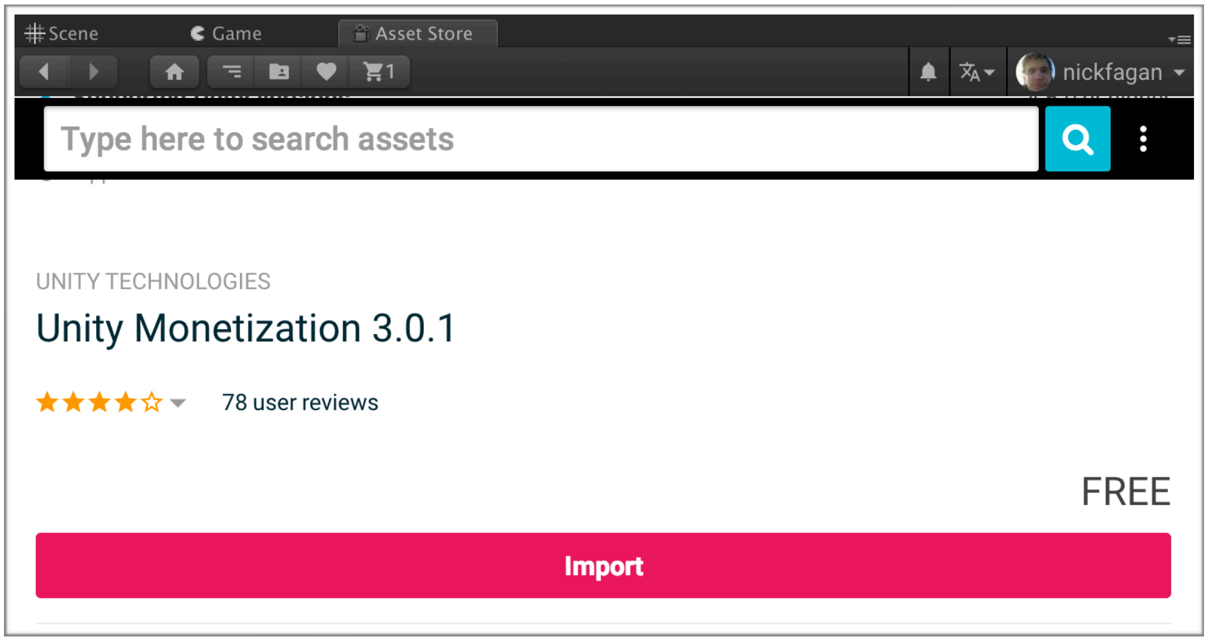
☑ Enable test mode

Un-Check this!

Advanced

☐ Enable built-in Ads extension

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


Scene | Game | Asset Store

nickfagan

Type here to search assets

UNITY TECHNOLOGIES

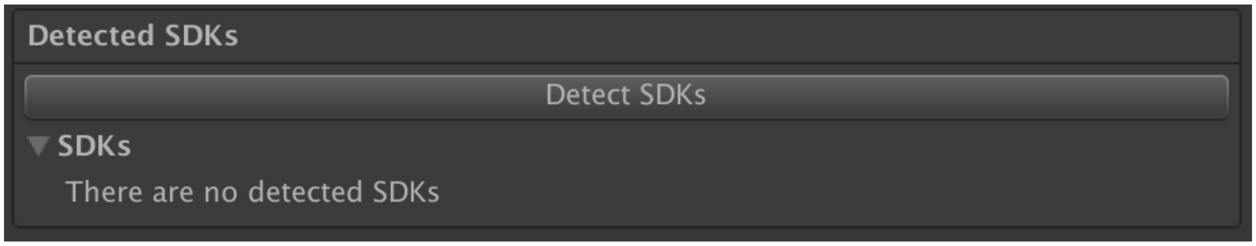
Unity Monetization 3.0.1

★★★★☆ 78 user reviews

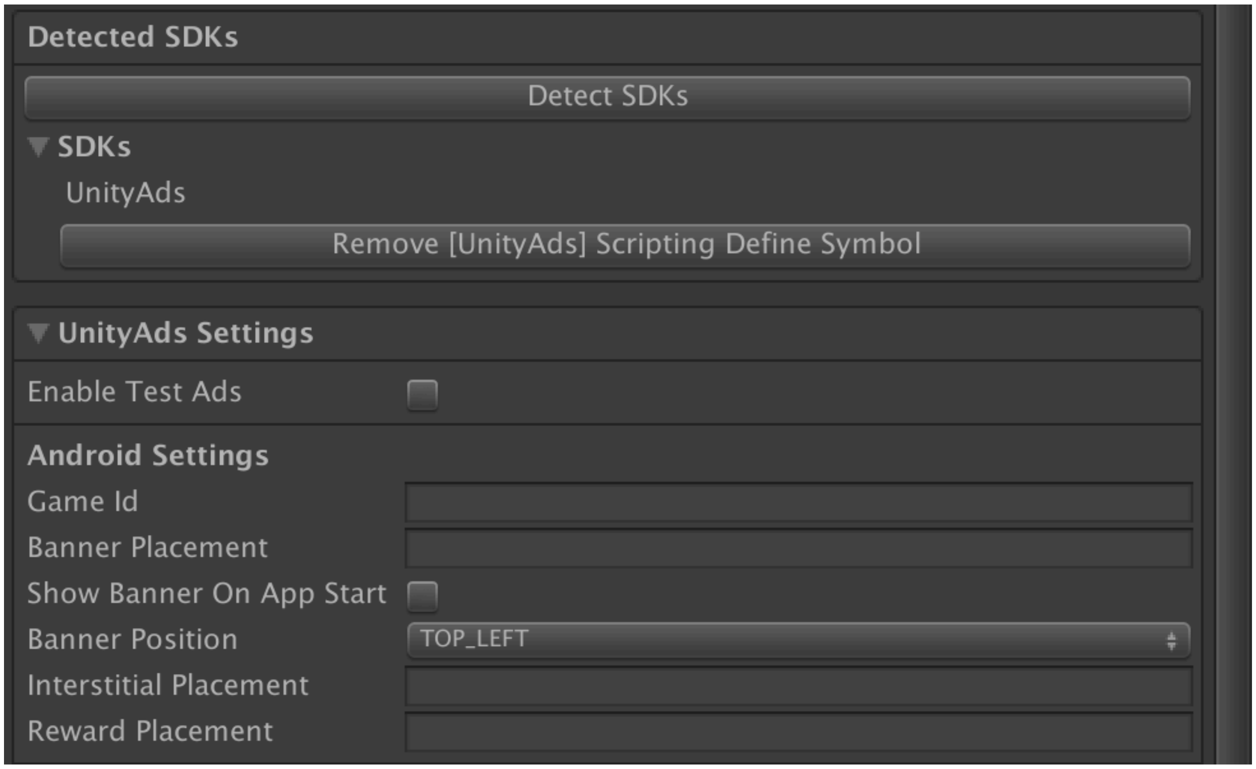
FREE

Import

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



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



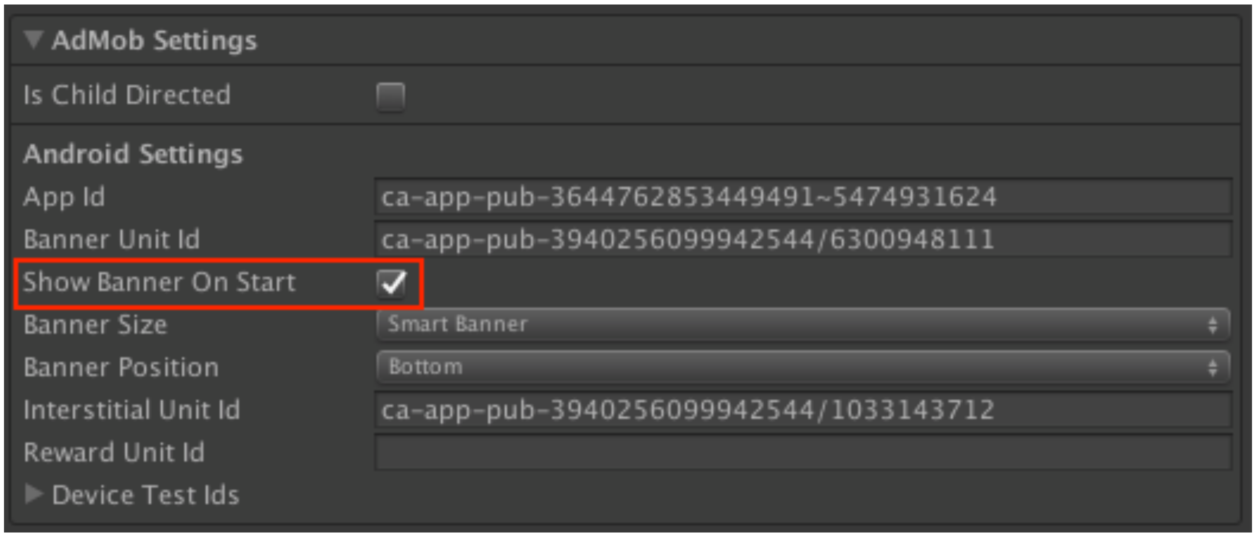
Step4. Add you Game Id and Placement Ids to the fields under UnityAds Settings.

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

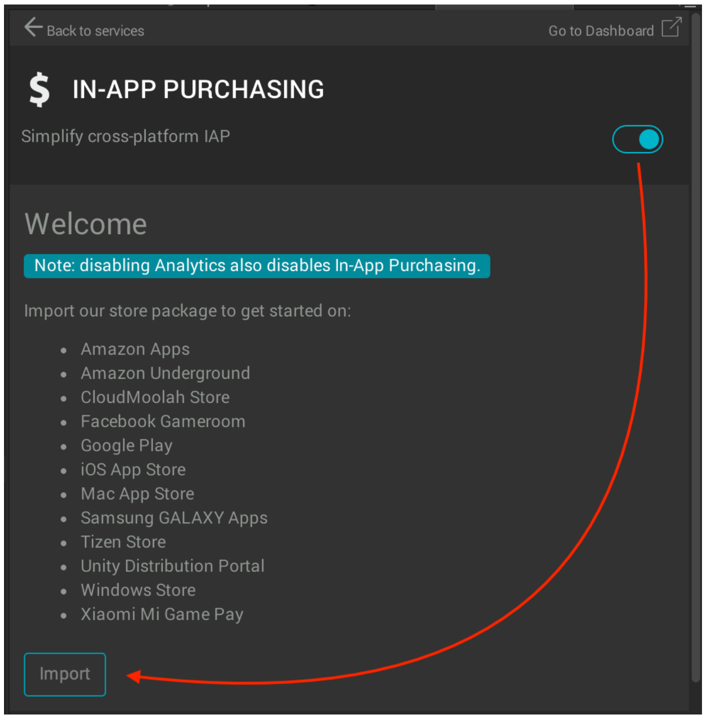
You should use the TestScene located in the Scenes folder to test if you have setup your ad networks properly. The test scene can be used to show ads and will print out log messages if there are any errors.

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:

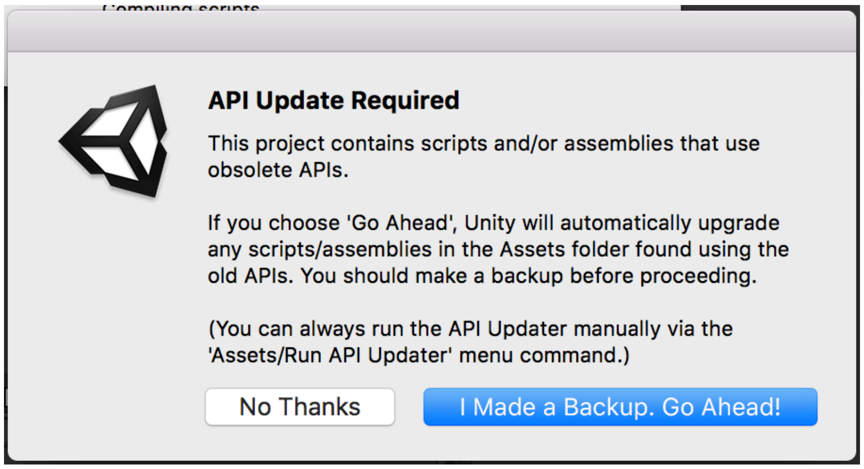
- Amazon Apps
- Amazon Underground
- CloudMoolah Store
- Facebook Gameroom
- Google Play
- iOS App Store
- Mac App Store
- Samsung GALAXY Apps
- Tizen Store
- Unity Distribution Portal
- Windows Store
- Xiaomi Mi Game Pay

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.

IMPORTANT!!!

If you get this dialog box:

You MUST click the **I Made a Backup. Go Ahead!** button. If you don't you will not be able to enable IAP in IAP Settings window.



API Update Required

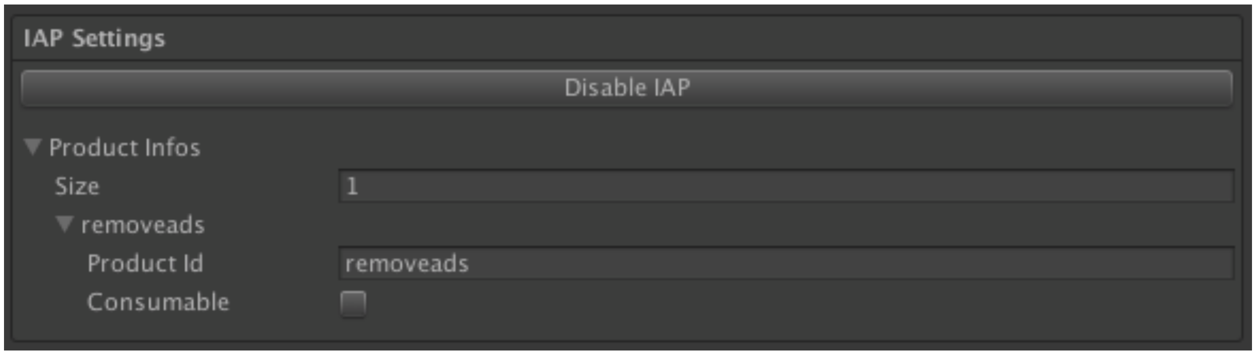
This project contains scripts and/or assemblies that use obsolete APIs.

If you choose 'Go Ahead', Unity will automatically upgrade any scripts/assemblies in the Assets folder found using the old APIs. You should make a backup before proceeding.

(You can always run the API Updater manually via the 'Assets/Run API Updater' menu command.)

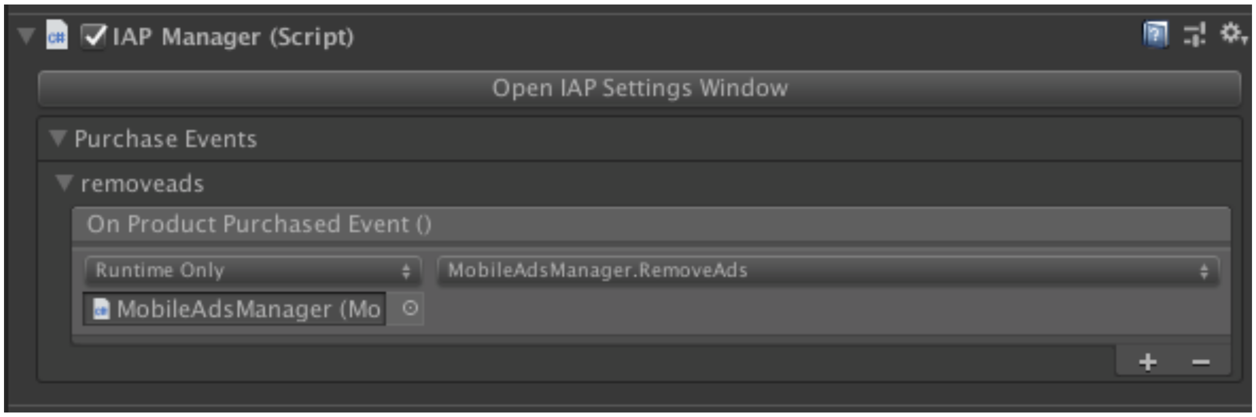
Add Product Ids

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



Size: 1
removeads
Product Id: removeads
Consumable: ☐

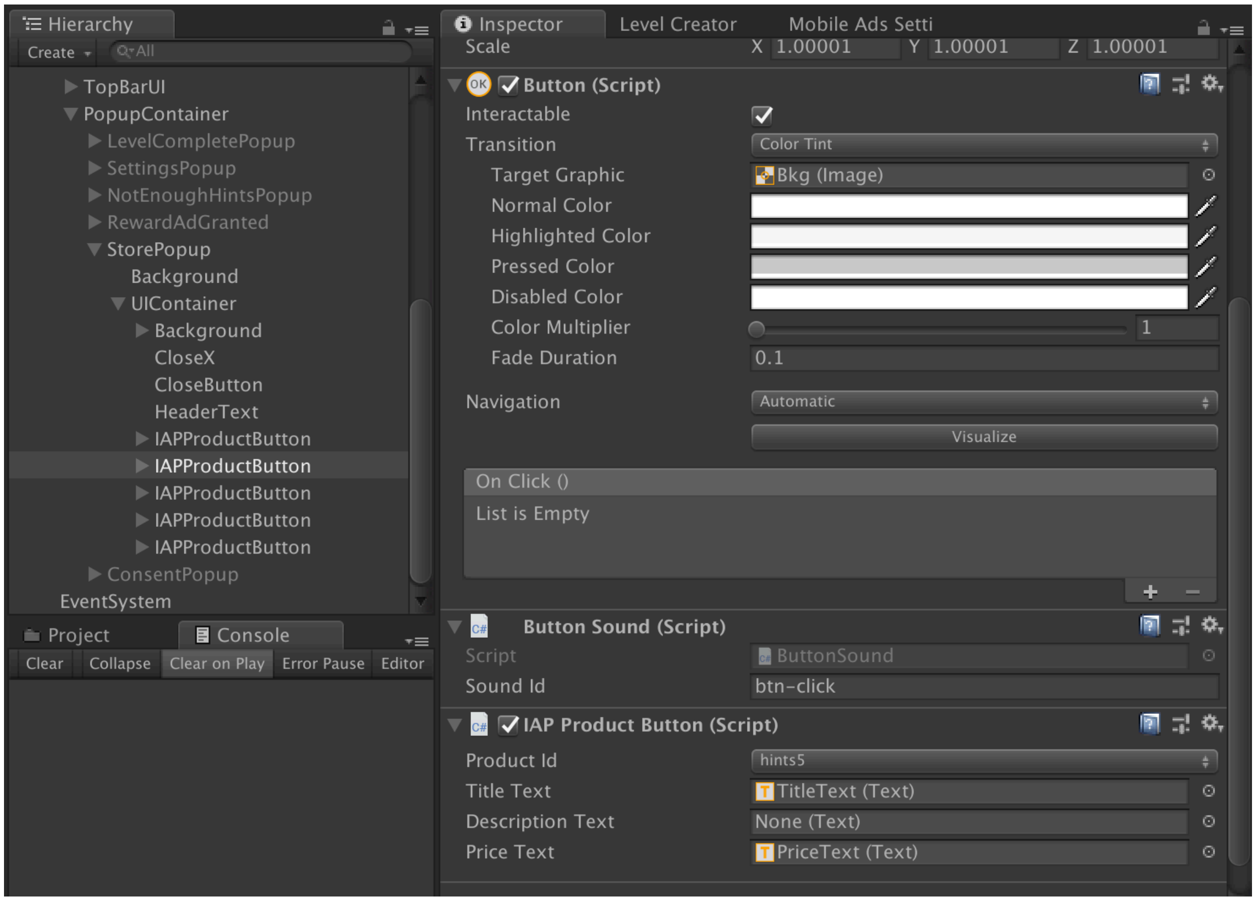
Once you add a new product an entry will appear on the IAPManager's inspector. Here you can add events for when the product is purchased:



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

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.