



MERGE CHAIN

USER GUIDE

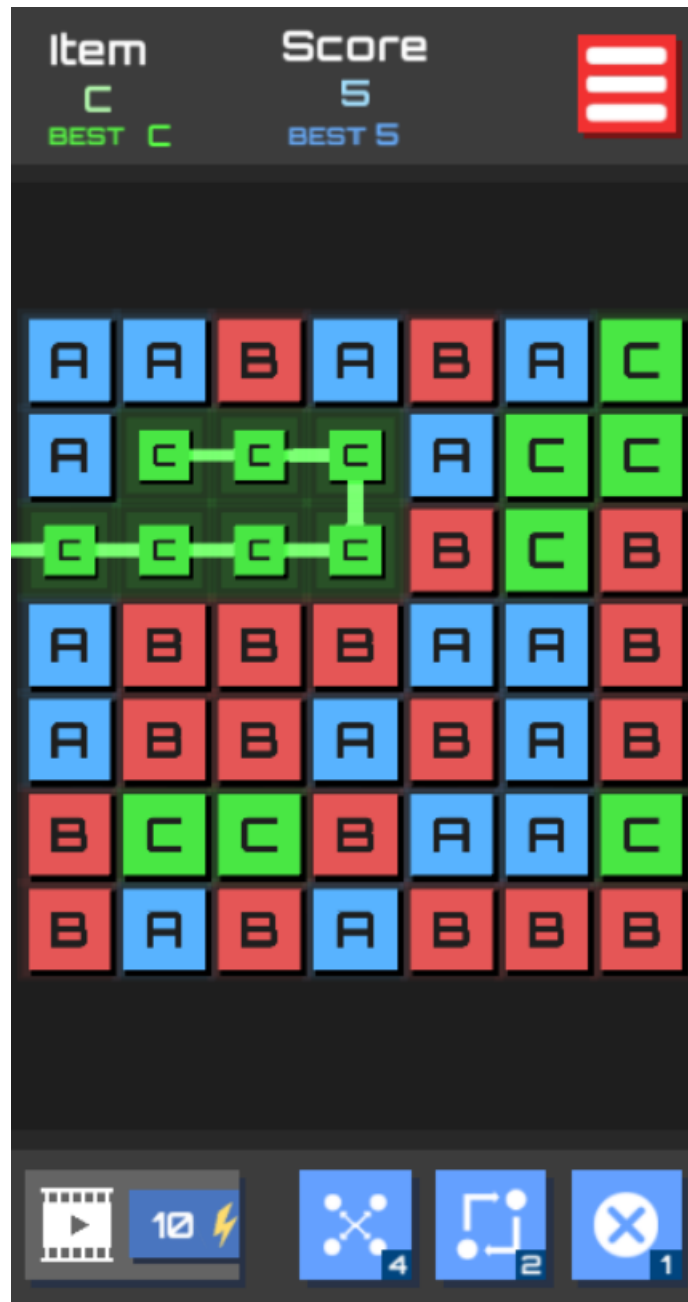
We strive to provide the best service as we can, if you have any questions or suggestions, please contact us!
Thank you!

SgLib Games

Table of Contents

1	INTRODUCTION	3
2	GETTING STARTED	4
2.1	ENTER APP INFORMATION	4
2.2	LINK THE GAME TO YOUR UNITY PROJECT	5
2.3	TESTING NOTE	6
3	TEMPLATE CUSTOMIZATION	6
3.1	ITEMDATAMANAGER	6
3.2	GRID ITEM MANAGER	8
3.3	POWERUPMANAGER	11
3.4	CUSTOMIZING UI	12
3.5	SOUNDS.....	12
4	ENABLING PREMIUM FEATURES	14
4.1	BEFORE YOU BEGIN	14
4.2	ADVERTISING.....	14
4.2.1	<i>Template-specific setup</i>	<i>14</i>
4.2.2	<i>Easy Mobile setup.....</i>	<i>15</i>
4.3	IN-APP PURCHASING.....	17
4.3.1	<i>Template-specific setup</i>	<i>17</i>
4.3.2	<i>Easy Mobile setup.....</i>	<i>18</i>
4.3.3	<i>Create the products for targeted stores</i>	<i>19</i>
4.4	GAME SERVICE.....	19
4.4.1	<i>Template-specific setup</i>	<i>19</i>
4.4.2	<i>Setup for your targeted stores.....</i>	<i>20</i>
4.4.3	<i>Easy Mobile setup.....</i>	<i>21</i>
4.5	NATIVE SHARING	22
4.6	RATING REQUEST	24
4.7	PUSH NOTIFICATION.....	25

1 INTRODUCTION



In **Merge Chain**, you drag and merge chains of same letters to combine them and get the next letter. What letter can you reach?

This game is ready for release out-of-the-box. Everything just works. It is also flexible and customizable. Some highlights:

- A creative variant of the ubiquitous Match-3 gameplay

- Grid size is easy to custom (default to matrix 7x7, change it if you want).
- The item colors are fully customized.
- Score number on every item is customizable.
- 3 built-in power-ups: board shuffling, swap tiles and delete tiles.
- Free-to-use assets (fonts, sounds, music, model, etc.)
- Optimized for iOS and Android

Most importantly, this template is pre-integrated with **Easy Mobile** plugin, making it a truly fully-featured game that is release-ready. Easy Mobile is a comprehensive, cross-platform package that provides most of desired features of mobile games:

- Support for AdColony, AdMob, Chartboost, Heyzap and UnityAds
- In-app purchasing (remove ads)
- Leaderboards and achievements
- Sharing to social networks
- Push notification using OneSignal service
- Native rating request popup (rate my app)

** Being pre-integrated means this template is already configured to work with Easy Mobile. All you need is import Easy Mobile and do a few setup steps, and have all the above features readily implemented. You don't even have to write a single line of integration code!*

** This template DOES NOT include Easy Mobile.*

** The use of Easy Mobile is totally optional: as long as it's not imported, all the integration code will automatically be excluded from compilation, so that no impact will be made on the game, which is fully functioning on its own.*

2 GETTING STARTED

2.1 Enter app information

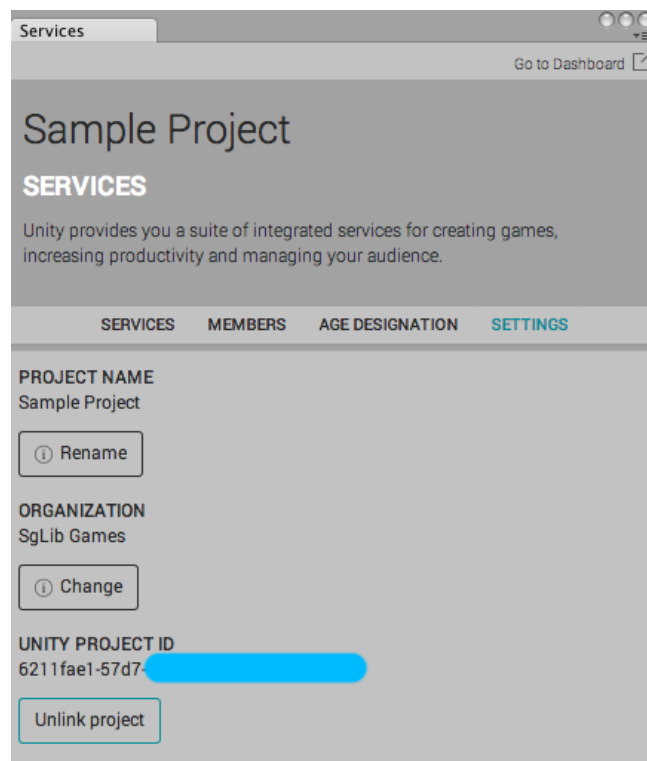
The project contains a game object called AppInfo where you can fill in important app-related metadata like AppStore Id and Bundle Id. These values will be used for features like Rate Us button and opening Facebook or Twitter page.



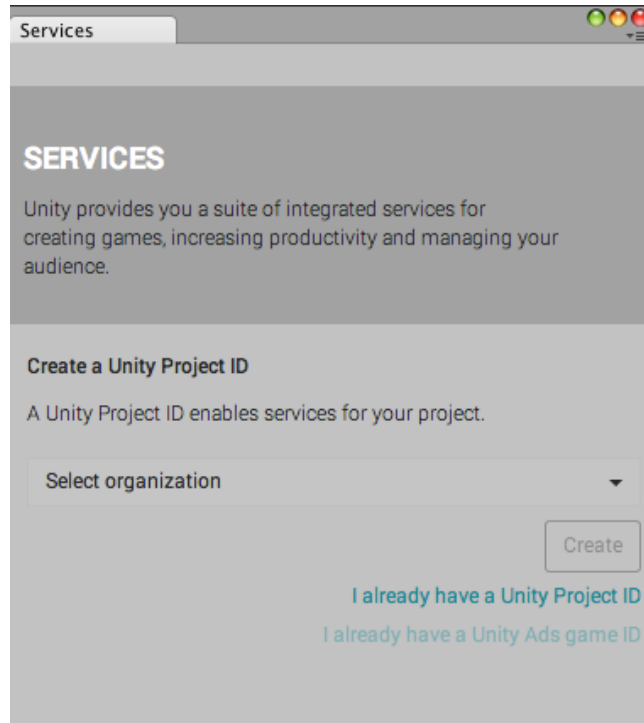
2.2 Link the game to your Unity project

When developing this template, we normally need to link it to our own Unity project for testing, therefore you may need to unlink it from our project and link it to your own one, if you're going to use Unity services (e.g. if you want to enable premium features of this template, you'll need to use Unity IAP service). To unlink the project:

- Select Window -> Unity Services
- Select SETTINGS tab
- Click Unlink Project button



Now you can create a new project for the game.



Now your game is linked to your own Unity project and is ready to use Unity services.

2.3 Testing Note

There are 2 scenes in this game, it should be run from scene *Main*.

3 TEMPLATE CUSTOMIZATION

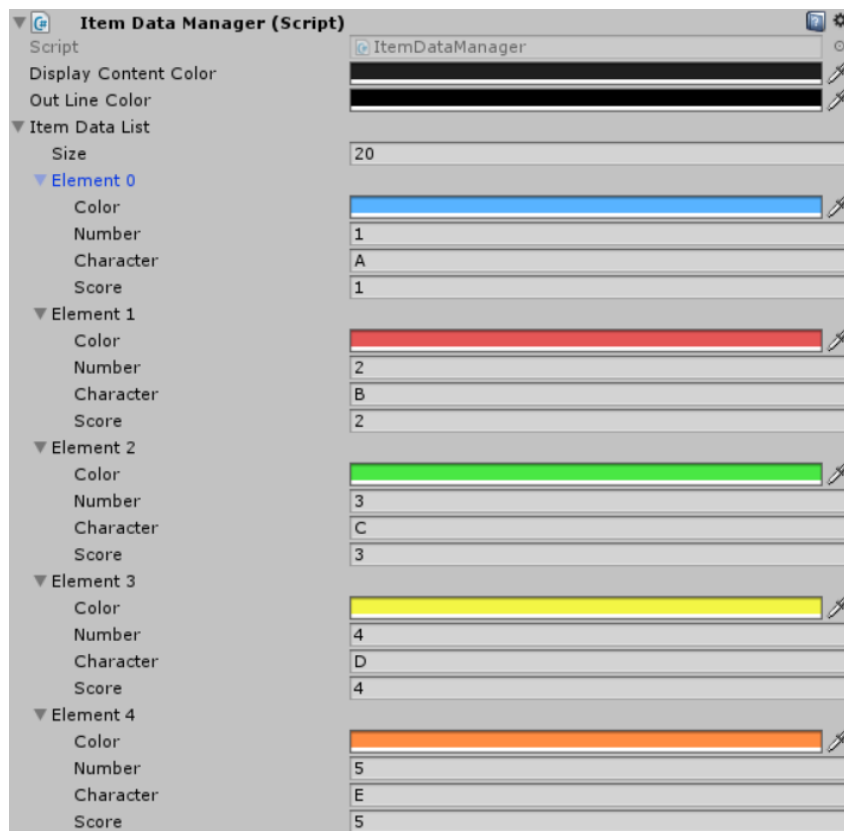
3.1 ItemDataManager

You also can tweak color, number and character of all items which you create in game. Besides, you can create the large number item if you want.

Only you access into hierarchy TAB, and click to ItemDataManager object:



Next, let you open ItemDataManager component of it, you will see:



That are properties of ItemDataManager script which attach into ItemDataManage object:

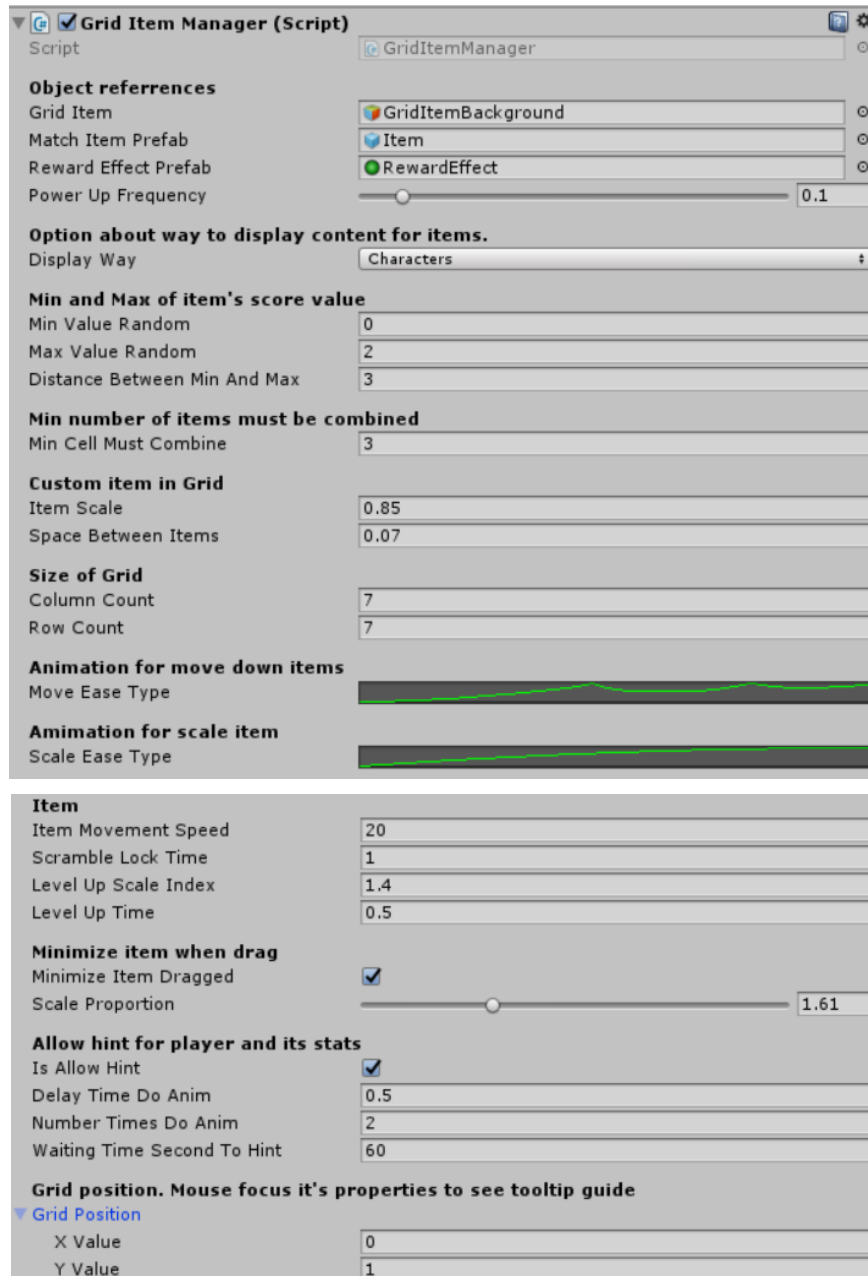
- **OutlineColor:** If you want to custom outline color of every item in game, you need only custom here, at this property.
- **ScoreNumberColor:** Also you want to custom score number on every item, you need custom this property.
- **ItemDataList:** It contains all ItemData that is struct to save Information of Item object.
 - **ItemData:**
 - Color: Color of this item.
 - Number: value will display when display way is “Number”.
 - Character: value will display when display way is “Character”.
 - Score: player will earn this score when destroy item.

3.2 Grid Item Manager

All items in game are contained in GridItemManager object.



You will see component which assigned for it:



This is the Grid Item Manager scripts which created to manager all items in game.

You can tweak all properties it, follow by:

- **DisplayWay:** With this option, you can choose way to display alphabet or number in every item. At this time, it has got two value: Number and Characters. If you choose “Numbers”, content of items will display numbers. Otherwise, if you choose “Characters”, they will display characters.
- **Min and Max of item's score value:** You can tweak score number for initial

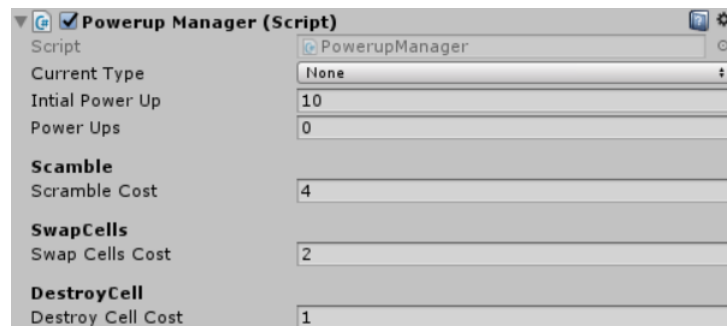
items or all items which spawned random whenever grid has got empty cell in it. Besides, whenever you do tweak with every property in this region, it will automatically do action as increment `maxValueRandom` when `minValueRandom` greater than `maxValueRandom`, ...

- ***MinValueRandom***: This is min value to spawn new random with min score in it.
- ***MaxValueRandom***: Opposite with above property, this is the max value for items has spawned whenever grid has been empty cell.
- ***DistanceBetweenMinMax***: This property use for control distance between min value and max value. Ex: if you input 3, then $\text{maxValue} - \text{minValue} = 3$. It mean `maxValueRandom` can't greater than `minValueRandom` 3 unit value.
- ***MinCellMustCombine***: This value decide for number of items which player must combine to destroy them, and create new increment item.
- ***Custom item in grid***: This region properties, you can tweak about size of every item with ***ItemScale*** property. It will change item scale whenever you are changed it.
- ***Size of Grid***: This region allow you to custom size of matrix contain all items.
 - ***ColumnCount***: The number columns of grid
 - ***RowCount***: The number rows of grid
- ***ControlEffectItem***: Would you like to instantiate effect on every item? If you want to them, you will check into this property.
- ***MoveEaseType***: Whenever player match success greater than 3 items, game will spawn new items and will move down them. At that time, it also does this animation.
- ***ScaleEaseType***: If you had checked is allow hint for player, it would have been do this animation for scale items which had found by hint of game.
- ***ItemMovingSpeed***: moving speed of item.
- ***Scramble Lock Time***: time out to scramble all items completely.
- ***LevelUpScaleIndex***: ratio of scale of item when level up.
- ***LevelUpTime***: time out to end level up animation.
- ***Minimize Item when drag***:

- ***MinimizeItemDragged:*** Would you like to minimum items when player dragging them? If you want to that, check into it.
- ***ScaleProportion:*** It will decide to scale for items when player dragging them. This value dependence on above properties. If you have checked *minimizeItemDragged*, this property just do it.
- ***Allow hint for player and its stats***
 - ***IsAllowHint:*** This property decide to hint for player if you check it. The under properties will be do if you check into this value. Otherwise, they don't work if you uncheck this value.
 - ***DelayTimeDoAnim:*** Whenever game do hint for player and you have checked *MinimizeItemDragged*, then items will do animation is scale to minimum their own. At this time, game will appear time delay to scale items.
 - ***NumberTimesDoAnim:*** This value decide to number of times do scale animation for the items were dragged by player.
 - ***WaitingTimeSecond:*** Whenever player don't play game in time is time equal this value. Game will hint for player.
- ***GridPosition:*** This value decide to matrix game field where local.
 - ***xValue:*** if it have value equal 0, matrix game field is local center follow x axis. With negative value, it move to left direction. Otherwise, with positive value, it move to right direction.
 - ***yValue:*** similar to above property, if have value equal 0, matrix game field is local center follow y axis ...

3.3 PowerUpManager

PowerUpManager contain a component is PowerupManager.



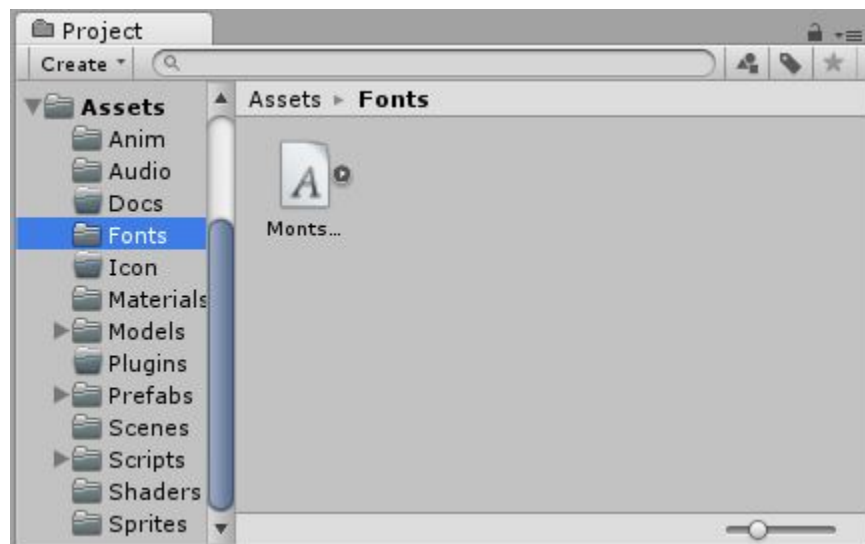
You can customize cost of each type's powerup.

3.4 Customizing UI

All sprites used in this game (for buttons and other UI components) are located under the *Sprites* folder. You can replace them with your own sprites to modify the UI as you like.



All fonts used in this game are free-to-use in commercial projects. Fonts are located under the *Fonts* folder together with appropriate license files.



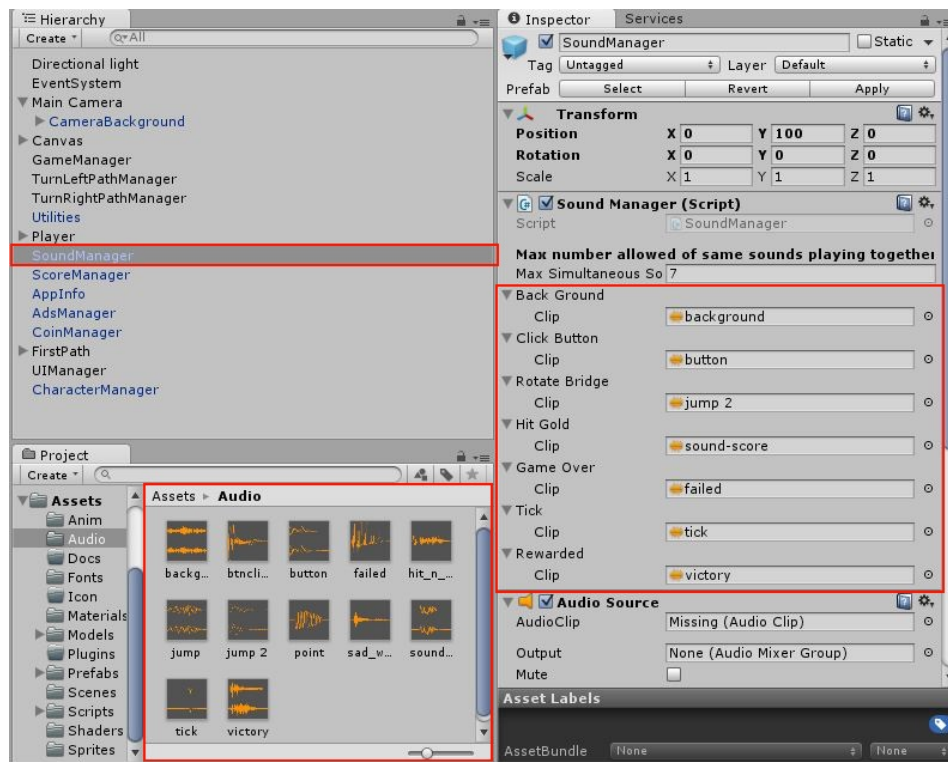
3.5 Sounds

All sounds included in this game are free-to-use in commercial projects and are

located under the *Audio* folder.



This game features a *SoundManager* class to manage activities in game like playing music or mute/unmute sounds. If you want to replace sounds in this game, simply drag and drop new sounds to appropriate slots in the *SoundManager* component.



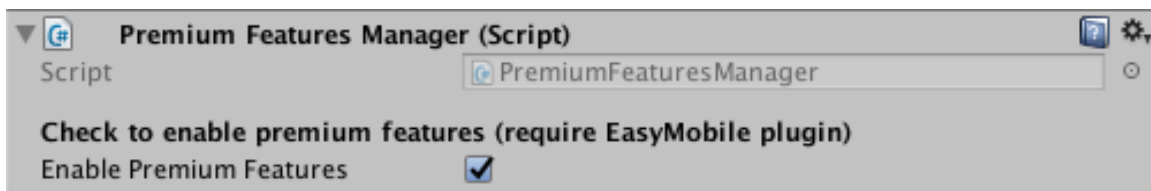
4 ENABLING PREMIUM FEATURES

To enable premium features of this template, you need to download and import Easy Mobile plugin from <http://u3d.as/Dd2>.

This section provides a guide on configuring each feature for your game. If you're not familiar with using Easy Mobile, it is strongly recommended that you read through its user guide to familiarize yourself with the plugin.

4.1 Before You Begin

- In the Main scene's hierarchy, there's an object named *PremiumFeaturesManager* which contains all the relevant components from which you can configure how premium features behave in your game.
- Make sure the *EnablePremiumFeatures* option in the *PremiumFeaturesController* object is checked.

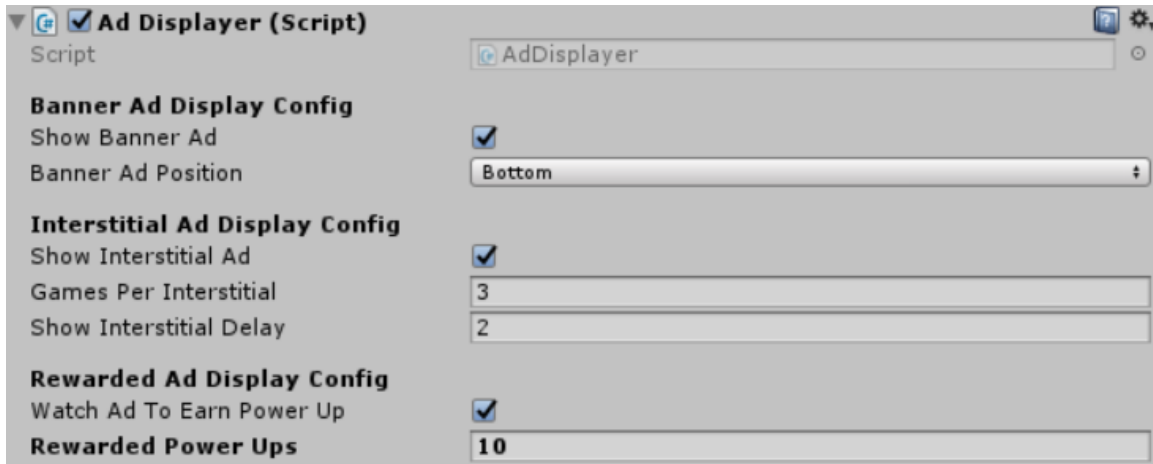


- Make sure to add the EasyMobile prefab to the Main scene, you can find the prefab at folder *Assets/EasyMobile*. It is necessary for the plugin to function properly.
- The settings interface of Easy Mobile can be opened via menu *Window > Easy Mobile > Settings*, this is the only place to go to configure this plugin.
- Note that you won't need to write a single line of integration code for Easy Mobile to work, as the integration was done beforehand, you only need to configure the plugin in the editor (that means you can ignore all the Scripting sections in Easy Mobile user guide if you wish to).

4.2 Advertising

4.2.1 Template-specific setup

The PremiumFeatureManager object contains a component named *AdDisplayer* which is responsible for all ads displaying activities in the game. There you can configure how ads should be served in your game.



Banner ads are configured in the **Banner Ad Display Config** section.

- *Show Banner Ad*: whether to show a banner ad in game
- *Banner Ad Position*: which position the banner should be placed

Interstitial ads are configured in the **Interstitial Ad Display Config** section.

- *Show interstitial ad*: whether to show interstitial ads when game over
- *Games Per Interstitial*: how many games to be played before showing ad
- *Show Interstitial Delay*: how many seconds after game over that ad is shown

Rewarded ads are configured in the **Rewarded Ad Display Config** section.

- *Watch Ad To Earn Power Up*: whether to allow the user to watch an ad to earn extra powerups.
- *Rewarded Coins*: how many powerups should be awarded after watching an ad

4.2.2 Easy Mobile setup

Open Easy Mobile's settings interface to start configuring its Advertising module (see its user guide for more information). With Easy Mobile you'll have support for AdColony, AdMob, Chartboost, Heyzap (with mediation) and Unity Ads. You can use multiple ad networks at once and have different configurations for iOS and Android. Below is the settings interface of the Advertising module.

ADVERTISING

ADMOB SETUP

Google Mobile Ads (AdMob) plugin was imported.

Reimport Google Mobile Ads Plugin

► [iOS] AdMob Ids
► [Android] AdMob Ids

CHARTBOOST SETUP

Chartboost plugin not found. Please download and import it to show ads from Chartboost.

Download Chartboost Plugin

HEYZAP SETUP

Heyzap plugin not found. Please download and import it to show ads from Heyzap.

Download Heyzap Plugin

UNITY ADS SETUP

Unity Ads service is enabled.

AUTO AD-LOADING CONFIG

Auto-Load Default Ads ☒

Ad Checking Interval 10

Ad Loading Interval 20

DEFAULT AD NETWORKS

▼ [iOS] Default Ad Networks

Banner Ad Network Ad Mob

Interstitial Ad Network Ad Mob

Rewarded Ad Network Unity Ads

▼ [Android] Default Ad Networks

Banner Ad Network Ad Mob

Interstitial Ad Network Ad Mob

Rewarded Ad Network Unity Ads

You can setup the module in just a few steps as below. Please see the Advertising section in Easy Mobile’s user guide for detailed instructions on each step.

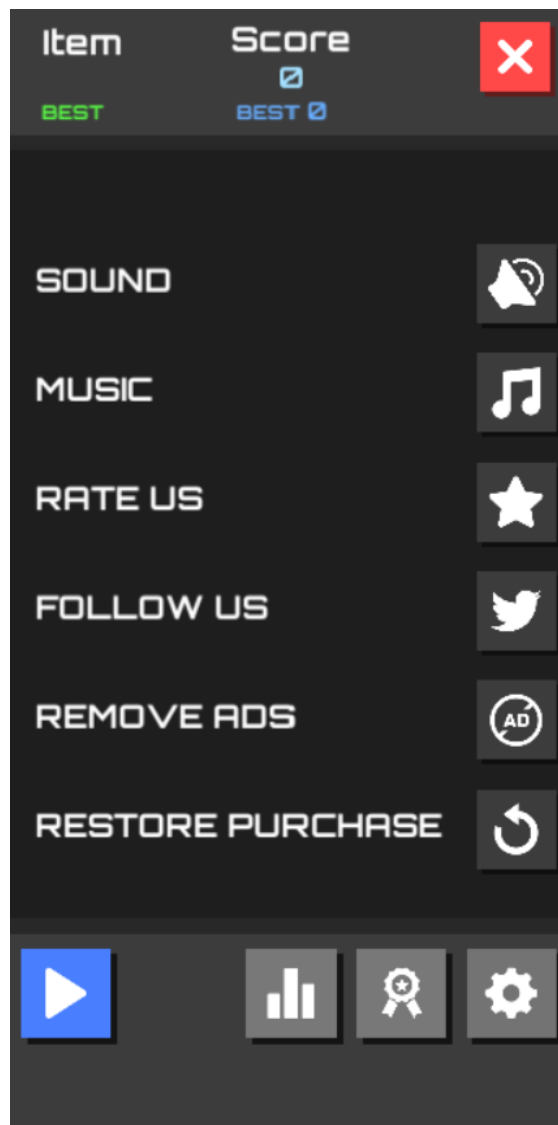
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads in the background
- Select default ad networks for each platform: choose your preferred network for each type of ad on each platform

That’s it! Now your game is ready for showing ads!

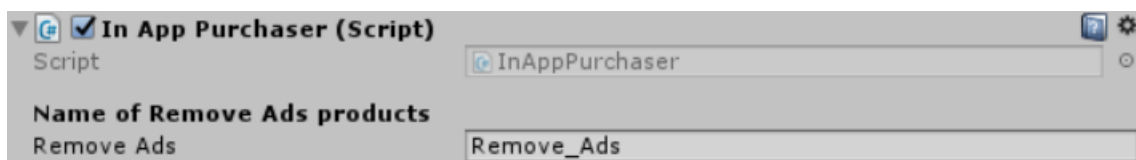
4.3 In-App Purchasing

4.3.1 Template-specific setup

The built-in in-app purchases of this template include a *Remove Ads* button. There's also one *Restore Purchase* button as required on iOS.



The `PremiumFeaturesManager` object contains a component named `InAppPurchaser` which manages all the in-app purchasing activities in this game.



4.3.2 Easy Mobile setup

Setting up the In-App Purchasing module of Easy Mobile includes the following steps. Please see the In-App Purchasing section in Easy Mobile's user guide for detailed instructions on each step.

- a. Enable Unity In-App Purchasing service
- b. Select target store if you're on Android
- c. Enable receipt validation if you wish
- d. Declare the products

Below is the settings interface of the In-App Purchasing module of Easy Mobile.

IN-APP PURCHASING

[ANDROID] TARGET STORE


Target Android Store Google Play

RECEIPT VALIDATION

Unity IAP offers local receipt validation for extra security. Apple stores and Google Play store only.

Validate Apple Receipt ☐

Validate Google Play Receipt ☐



Please go to Window > Unity IAP > IAP Receipt Validation Obfuscator and create obfuscated secrets to enable receipt validation for Apple stores and Google Play store. Note that you don't need to provide a Google Play public key if you're only targeting Apple stores.

PRODUCTS

► 6 Products

Add New Product

CONSTANTS CLASS GENERATION

Generate the static class EasyMobile.EM_IAPConstants that contains the constants of product names. Remember to regenerate if you make changes to these names.

Generate Constants Class

Note that the products declared with Easy Mobile must have names that match with the ones you have in the aforementioned *InAppPurchaser* component. Also note that *Remove Ads* is a non-consumable product, while the coin packs must be consumable.

Remove_Ads

Name: Remove_Ads

Type: Non Consumable

Id: sglib.demogame.iap.remove_ads

► More (Optional)

4.3.3 Create the products for targeted stores

That last step in configuring the in-app purchasing feature is to create products for your targeted stores (e.g. Google Play and Apple App Store). Make sure the product ID, product type and price match the ones you have in your game.

4.4 Game Service

4.4.1 Template-specific setup

This template has a built-in leaderboard for ranking users' scores, and many achievements. It works with Game Center (iOS) and Google Play Game Services (Android).

User's score will be submitted automatically when game over by a component named *ScoreReporter*, which is also attached to *PremiumFeaturesManager* object. There you can change the leaderboard name or even disable automatic score reporting altogether.

Score Reporter (Script)

Script: ScoreReporter

Check to disable automatic score reporting

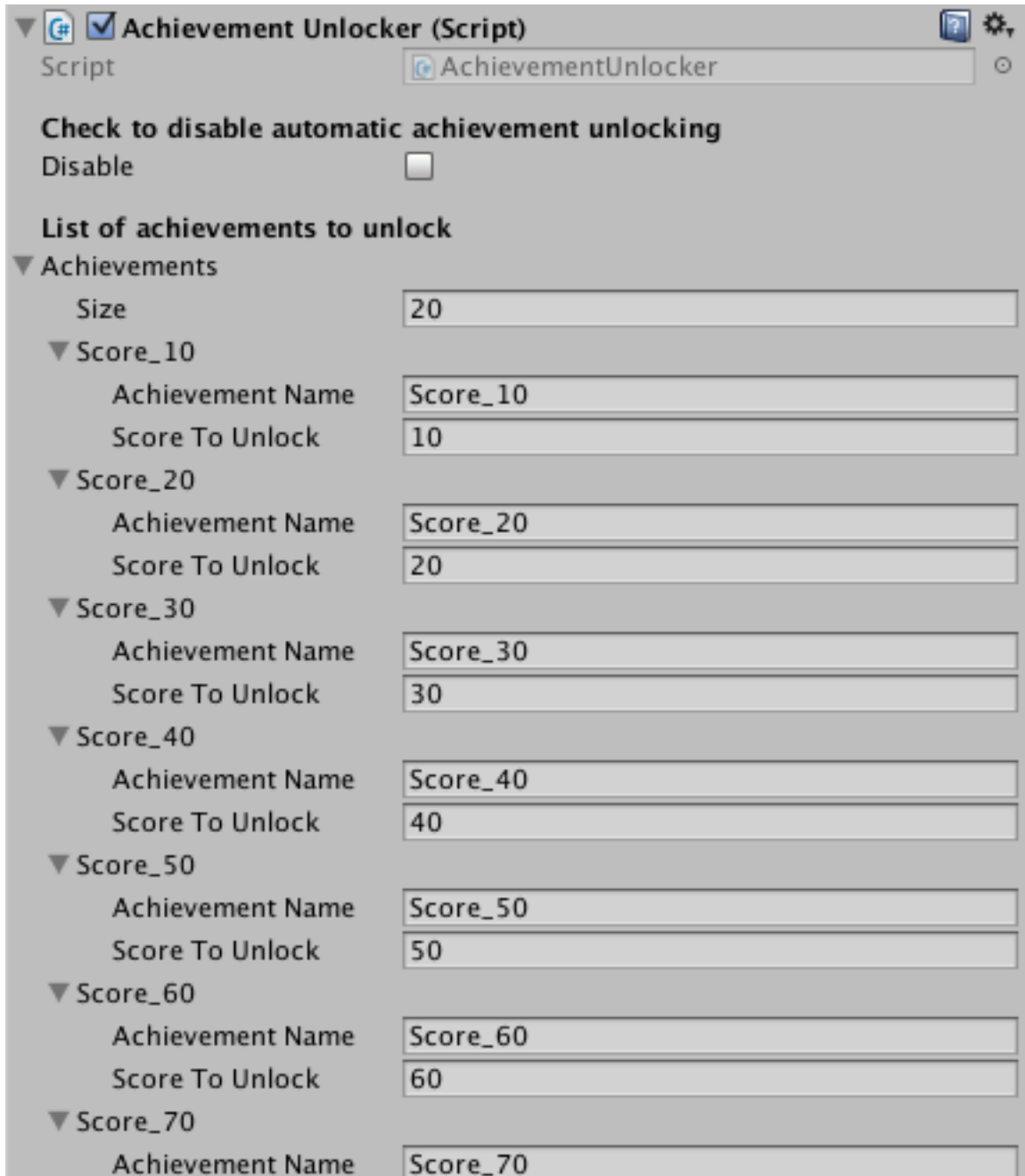
Disable: ☐




Name of the leaderboard to report score as declared with EasyMobile

Leaderboard Name: Score

Achievements will be unlocked automatically when the user reaches a certain score. The achievement unlocking is handled by the component named

AutoAchievementUnlocker. In this component, you can modify existing achievements and add or remove achievements. You can also disable the automatic achievement unlocking feature if you wish.



▼  **Achievement Unlocker (Script)**  

Script

Check to disable automatic achievement unlocking

Disable ☐

List of achievements to unlock

▼ Achievements

Size

▼ Score_10

Achievement Name

Score To Unlock

▼ Score_20

Achievement Name

Score To Unlock

▼ Score_30

Achievement Name

Score To Unlock

▼ Score_40

Achievement Name

Score To Unlock

▼ Score_50

Achievement Name

Score To Unlock

▼ Score_60

Achievement Name

Score To Unlock

▼ Score_70

Achievement Name

4.4.2 Setup for your targeted stores

The next step is to create the required leaderboard and achievements for your targeted stores (i.e. in iTunes Connect for App Store and the Developer Console for

Google Play). Take note of their IDs for use in the next step.

4.4.3 Easy Mobile setup

Setting up the Game Service module of Easy Mobile includes the following steps. Please see the Game Service section in Easy Mobile's user guide for detailed instructions on each step.

- a. Import Google Play Games plugin for Unity and setup it if you're targeting Android
- b. Enable the automatic initialization feature: just leave everything under the **AUTO-INIT CONFIG** section as default
- c. Declare the leaderboards and achievements

Below is the settings interface of the Game Service module of Easy Mobile.

GAME SERVICE

Google Play Games plugin is imported and ready to use.

Reimport Google Play Games Plugin

[ANDROID] GOOGLE PLAY GAMES SETUP

GPGS Debug Log ☐

Paste in the Android XML Resources from the Play Console and hit the Setup button.

Android XML Resources

```
<?xml version="1.0" encoding="utf-8"?>
<!--
Google Play game services IDs.
Save this file as res/values/games-ids.xml in your project.
-->
<resources>
<string name="app_id">104</string>
<string name="package_name">com.sglib.demogame</string>
<string name="achievement_score_10">Cgkl3tzAhK8eEAIQEG</string>
<string name="achievement_score_20">Cgkl3tzAhK8eEAIQEQ</string>
<string name="achievement_score_30">Cgkl3tzAhK8eEAIQEw</string>
</resources>
```

Setup Google Play Games

AUTO-INIT CONFIG

Auto Init ☒

Auto Init Delay 0

[Android] Max Login Request 3

LEADERBOARD SETUP

1 Leaderboards

Add New Leaderboard

ACHIEVEMENT SETUP

20 Achievements

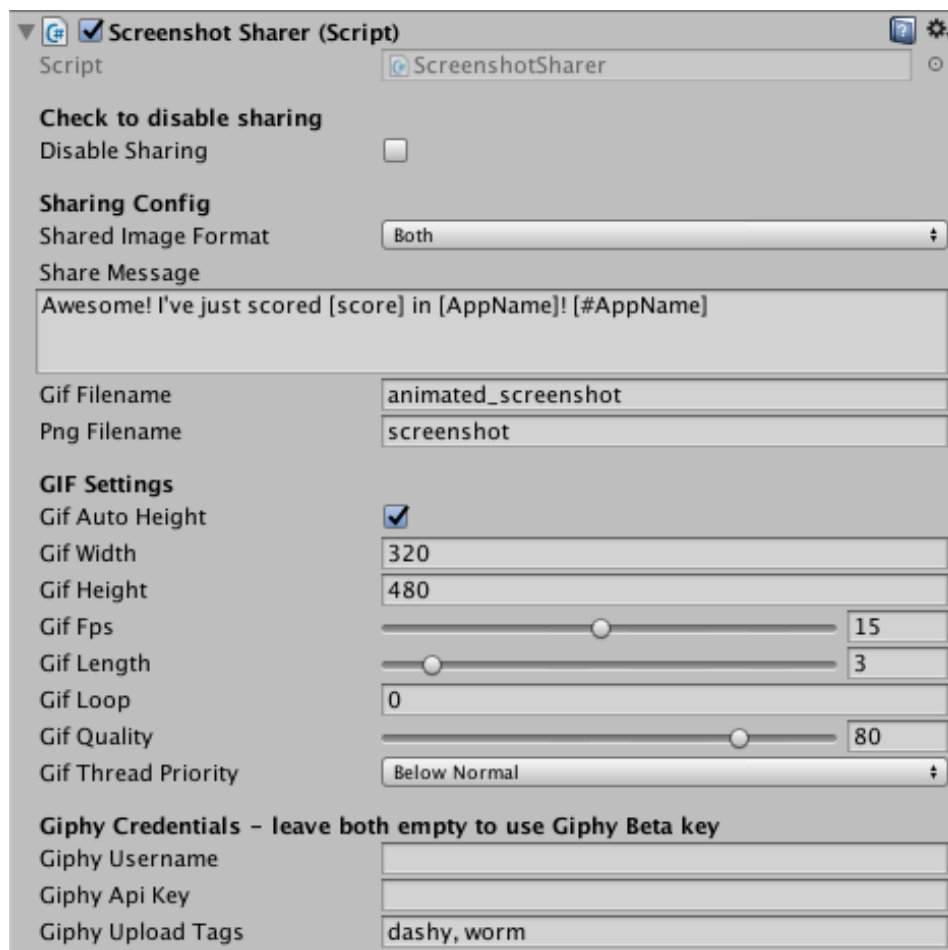
Add New Achievement

Note that you must declare the leaderboard and achievements with the same names as the ones you have in the *AutoScoreReporter* and *AutoAchievementUnlocker* components. Also their IDs must match the ones you created in iTunes Connect and Google Play Developer Console.

4.5 Native Sharing

This template has a Share button that allows the user to share a screenshot of the gameplay (in animated GIF or static PNG format) to social networks using the native sharing functionality. This activity is managed by a component named *ScreenshotSharer*, which is also attached to the *PremiumFeaturesManager* object. When a new game starts, this component will setup and start a recorder to record

the screen content, and stop the recorder when the game ends. The recorder automatically stores a few last seconds of the recording, and discards the rest. The recorded clip will then be exported to a GIF image, then uploaded to [Giphy](#), and finally its URL can be shared and played automatically on major social networks including Facebook and Twitter.



Here you can configure the sharing feature.

- *Disable Sharing*: disable this feature
- *Share Image Format*: you can share GIF or PNG image, or both
- *Share Message*: the default sharing message, note that [score] will be automatically replaced by actual score, and [AppName] will be replaced by the app name declared in AppInfo
- *Gif Filename*: filename to store the generated GIF image
- *Png Filename*: filename to store the captured PNG image

In the **GIF Settings** section, you can configure the generation of the GIF image.

- *Gif Auto Height*: automatically calculate the image height based on the specified width and the screen aspect ratio
- *Gif Width*: the image width
- *Gif Height*: the image height, will be overwritten if *GifAutoHeight* is enabled
- *Gif Fps*: frame per second of the GIF image
- *Gif Length*: the length of the GIF in seconds, as mentioned earlier, the recorder only keeps this many seconds of the recording, and discards old content
- *Gif Loop*: looping mode of the GIF; 0 means loop indefinitely, -1 means no loop, > 0 means loop a set number of times
- *Gif Quality*: quality setting value in range [1,100], bigger values mean better quality but slightly longer generation time; 80 is generally a good value in terms of quality-time balance
- *Gif Thread Priority*: the priority of the GIF generation thread

You can also control the Giphy uploading activity with the following parameters:

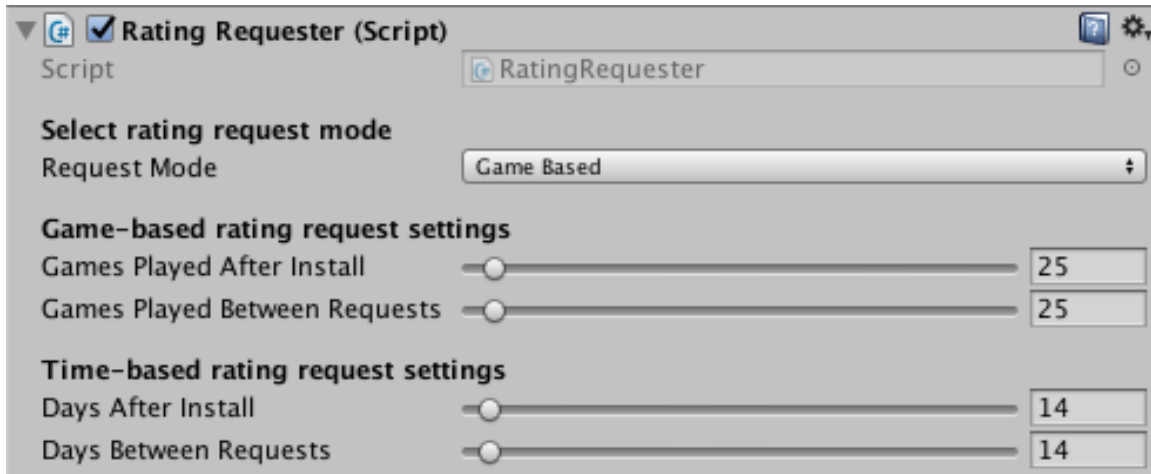
- *Giphy Username & Giphy Api Key*: provide these values if you want to upload the GIF image to your own Giphy channel; otherwise leave them empty to use the [Giphy beta key](#)
- *Giphy Upload Tags*: comma-delimited list of tags of the uploaded image

Note that you need to enable the *External Write Permission* for this feature to function properly on Android. Please see the Native Sharing section in Easy Mobile user guide for detailed instructions on doing that.

4.6 Rating Request

This template employs the Rating Request feature of Easy Mobile, to show a rate-my-app popup when game over, if some certain conditions are met. The Rating Request feature of Easy Mobile allows us to show the built-in rating prompt on iOS (10.3+) and a native rating popup on Android. Please see the Rating Request section in Easy Mobile user guide for instructions on configuring the appearance and behavior of this popup.

You can set the conditions to show this rating popup using the *RatingRequester* component of the *PremiumFeaturesManager* object.



- *Request Mode*: whether to show the rating popup based on the number of games played (Game Based mode), or based on the time since the installation of the app (Time Based mode)

If you select *Game Based* mode, pay attention to these two variables:

- *Games Played After Install*: how many games should be played since the installation before a rating popup is shown
- *Game Played Between Requests*: how many games should be played since the last time a rating popup is shown (in case it was dismissed by the user) that a new popup can be shown

If you select *Time Based* mode, adjust these two variables:

- *Days After Install*: how many days after the installation that a rating popup is shown
- *Days Between Requests*: how many days since the last time a rating popup is shown that a new one can be shown

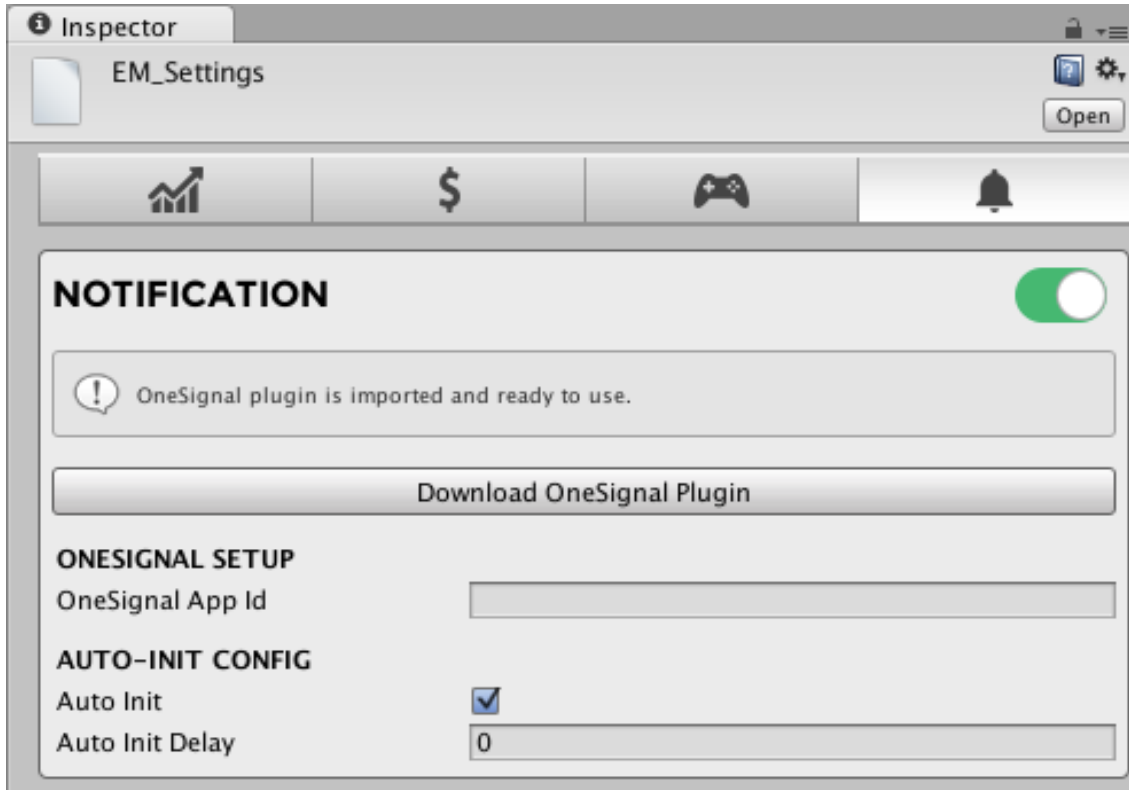
4.7 Push notification

Enabling push notification for your app using OneSignal service includes following steps. Please see the Notification section in Easy Mobile user guide for detailed instructions on each step.

- Open the Notification tab in Easy Mobile's settings interface
- Import OneSignal plugin
- Prepare your app for push notifications, e.g. enable the Push Notification capability for the provisioning profile on iOS (please see Easy Mobile user guide as well as OneSignal documentation for detailed instructions).
- Add your app to OneSignal dashboard

- Enter your app ID to Easy Mobile settings in Unity

Below is the settings interface of the Notification module of Easy Mobile after importing OneSignal plugin and you can enter your app ID.



That's it! You've just finished implemented premium features for your game!

THANK YOU AND GOOD LUCK WITH YOUR GAMES!