



EMOJI DOWN THE HILL TEMPLATE

USER GUIDE

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

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.....	4
3	TEMPLATE CUSTOMIZATION	6
3.1	GAMEPLAY TWEAKING	6
3.2	ADDING MORE CHARACTERS	6
3.3	CUSTOMIZING UI	7
3.4	SOUNDS.....	8
4	ENABLING PREMIUM FEATURES.....	9
4.1	BEFORE YOU BEGIN	9
4.2	ADVERTISING.....	10
4.2.1	<i>Template-specific setup.....</i>	<i>10</i>
4.2.2	<i>Easy Mobile setup.....</i>	<i>10</i>
4.3	IN-APP PURCHASING.....	12
4.3.1	<i>Template-specific setup.....</i>	<i>12</i>
4.3.2	<i>Easy Mobile setup.....</i>	<i>13</i>
4.3.3	<i>Create the products for targeted stores</i>	<i>14</i>
4.4	GAME SERVICE.....	14
4.4.1	<i>Template-specific setup.....</i>	<i>14</i>
4.4.2	<i>Setup for your targeted stores.....</i>	<i>15</i>
4.4.3	<i>Easy Mobile setup.....</i>	<i>15</i>
4.5	NATIVE SHARING	17
4.6	PUSH NOTIFICATION.....	17

1 INTRODUCTION



Emoji Down The Hill is a simple and addictive one-touch mobile game. Tap to change direction of the bouncing emoji. Avoid obstacles (tree, holes) and collect items for special effects. One mistake and you die! Simple as that!

Emoji Down The Hill is ready for release out-of-the-box. Everything just works. It is also flexible and customizable. Some highlights:

- Addictive one-touch gameplay
- Free-to-use assets (fonts, sounds, music, model, etc.)
- Optimized for mobile

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 AdMob, Chartboost, Heyzap and UnityAds
- In-app purchasing
- Support for Game Center (iOS) and Google Play Games Services (Android) for leaderboards and achievements
- Sharing to social networks
- Push notification using OneSignal service

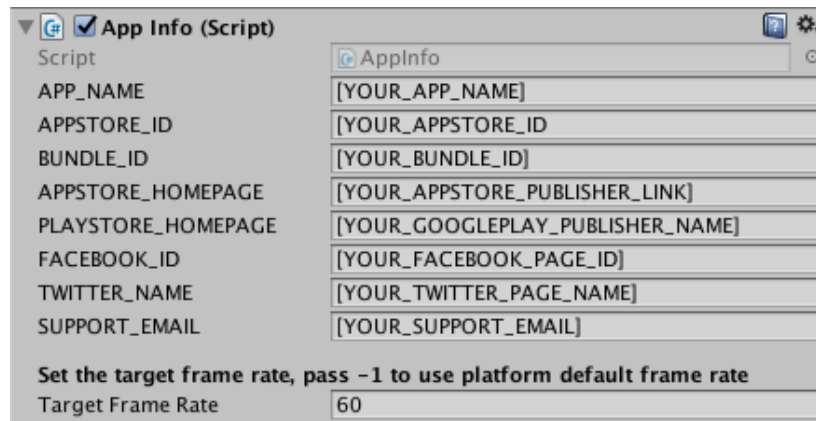
Being *pre-integrated* means this template is already configured to work with Easy Mobile. All you need is import Easy Mobile and do a few setup steps, and have all the above features readily implemented. You don't even have to write a single line of integration code!

** It's worth noting that 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.



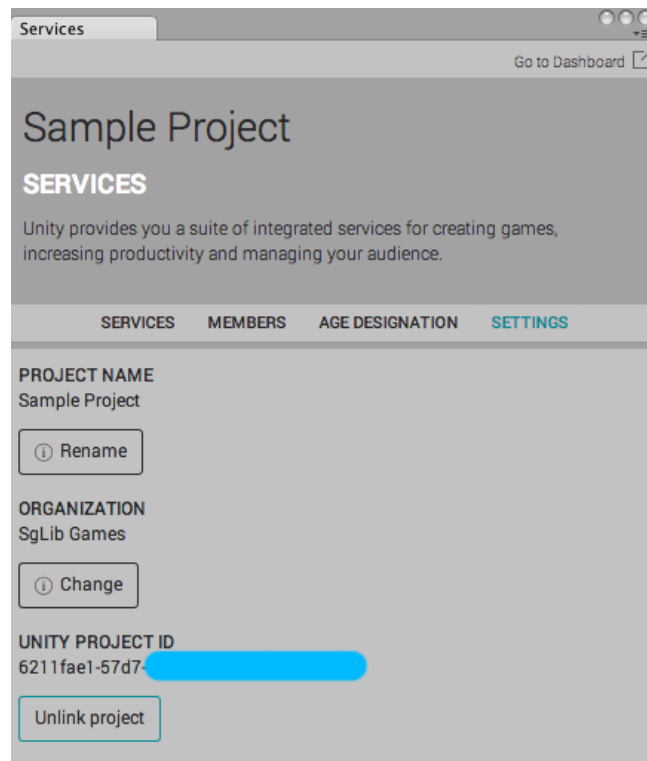
Field	Value
Script	AppInfo
APP_NAME	[YOUR_APP_NAME]
APPSTORE_ID	[YOUR_APPSTORE_ID]
BUNDLE_ID	[YOUR_BUNDLE_ID]
APPSTORE_HOMEPAGE	[YOUR_APPSTORE_PUBLISHER_LINK]
PLAYSTORE_HOMEPAGE	[YOUR_GOOGLEPLAY_PUBLISHER_NAME]
FACEBOOK_ID	[YOUR_FACEBOOK_PAGE_ID]
TWITTER_NAME	[YOUR_TWITTER_PAGE_NAME]
SUPPORT_EMAIL	[YOUR_SUPPORT_EMAIL]
Set the target frame rate, pass -1 to use platform default frame rate	
Target Frame Rate	60

Besides basic app info, there's one other important variable, *TargetFrameRate*, which sets the target frame rate for the game and should be at least 60fps for smooth motion.

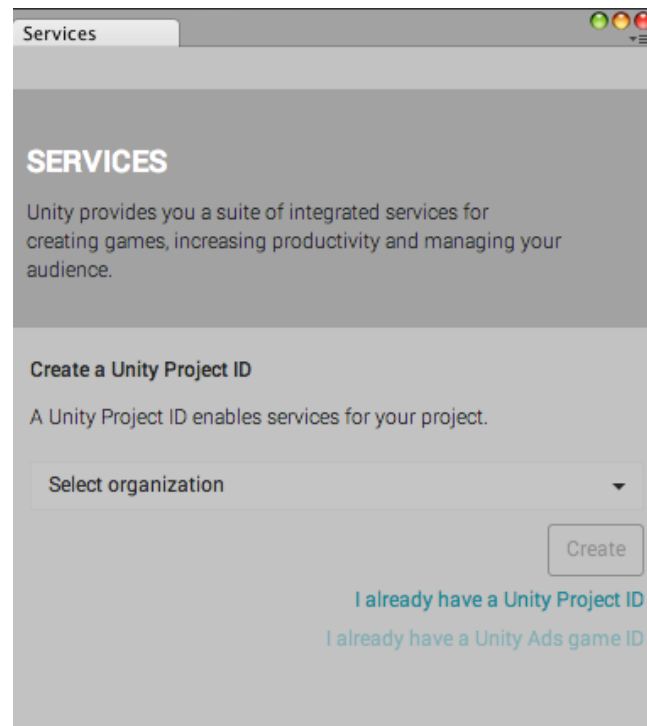
2.2 Link the game to your Unity project

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.

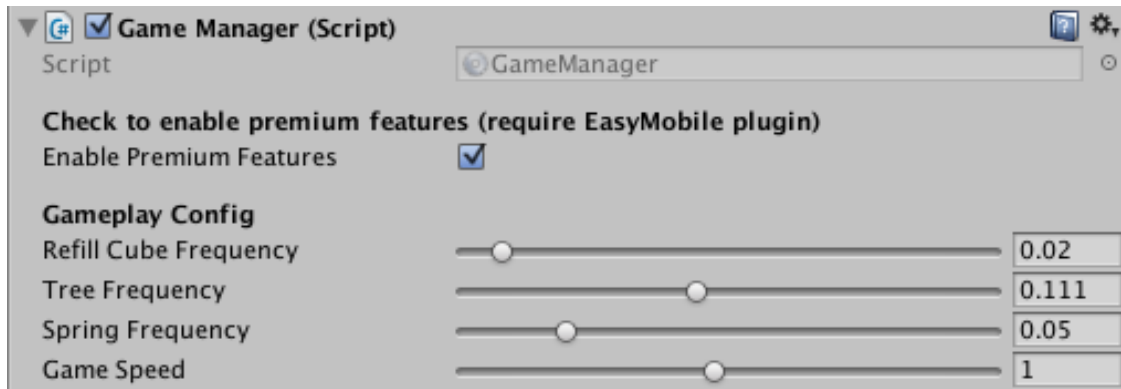


Now your game is linked to your own Unity project and is ready to use Unity services.

3 TEMPLATE CUSTOMIZATION

3.1 Gameplay tweaking

Most of important gameplay parameters can be configured in the *GameManager* component, which is attached to a game object also named *GameManager*.

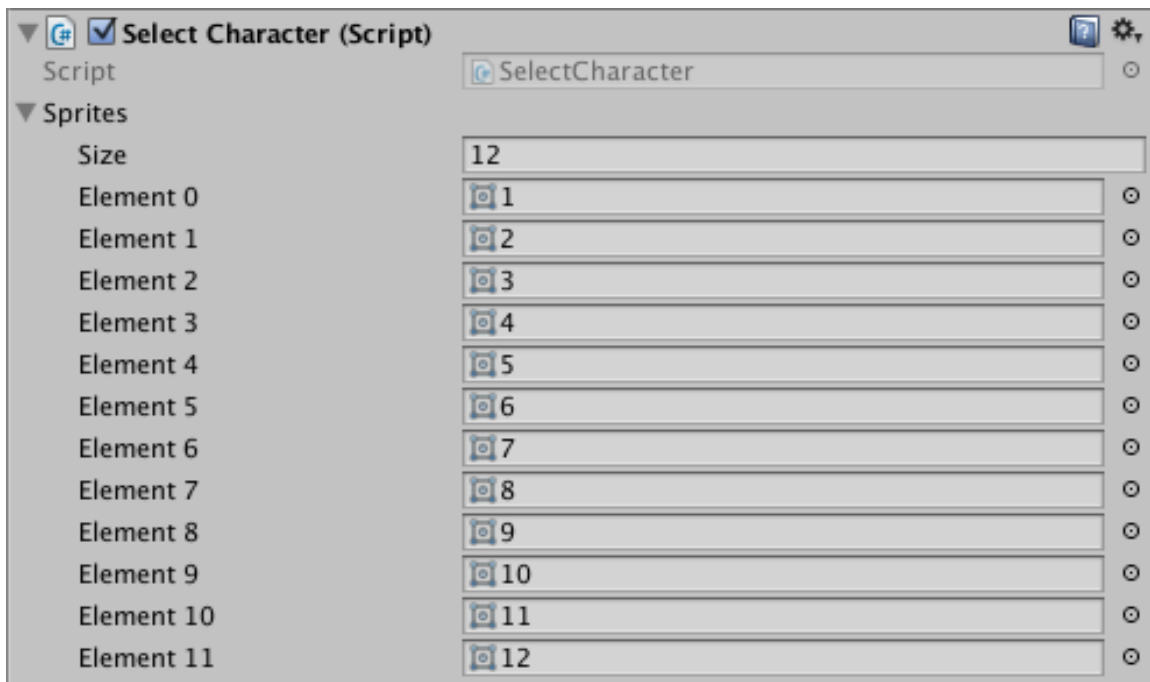


You can tweak the gameplay by modifying following variables:

- *Refill Cube Frequency*: how frequently the filling item appears
- *Tree Frequency*: how frequently the trees are generated, the more the harder.
- *Spring Frequency*: how frequently the springs are generated.
- *Game Speed*: change speed of the game by setting the timescale internally.

3.2 Adding more characters

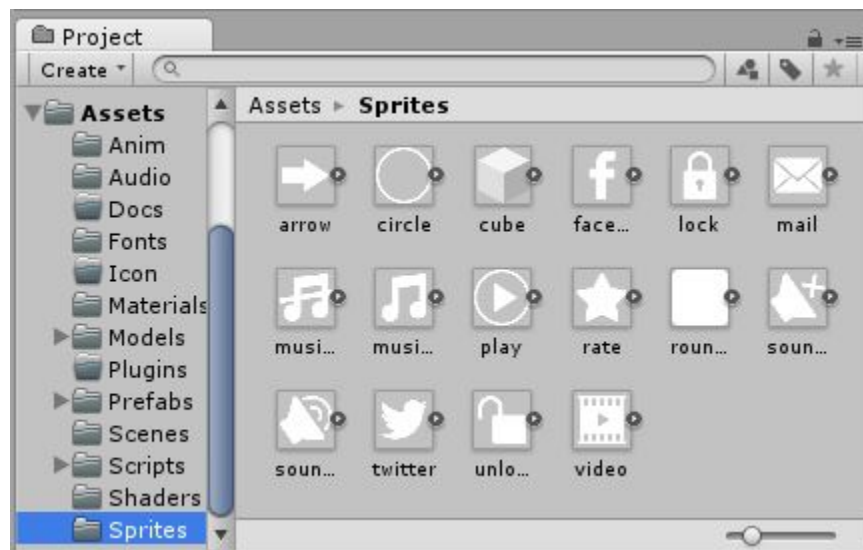
In the hierarchy, locate the object named *Player* and then its child object *PlayerSprite*, in this child object there's a component call *Select Character*. To add more emojis, simply drag their sprites into the *Sprites* array.



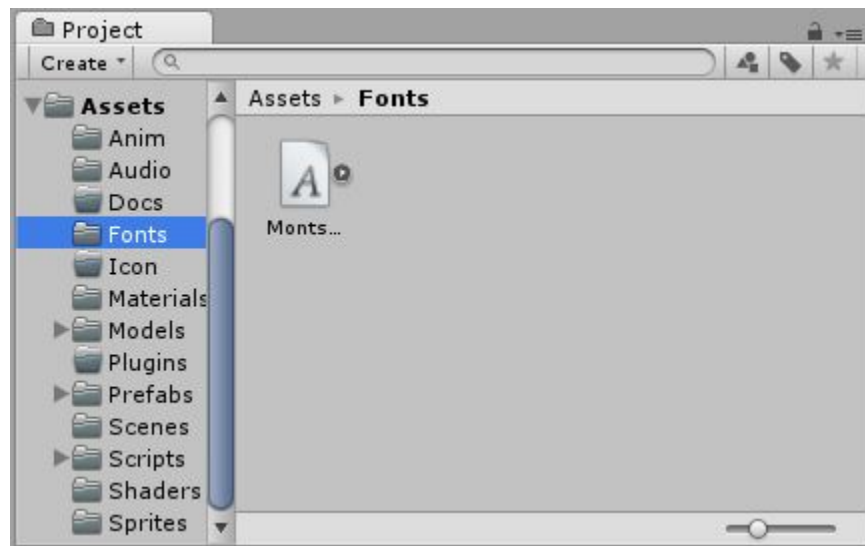
* **Acknowledgement:** The emoji used in this game are [Design by Freepik](#).

3.3 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.

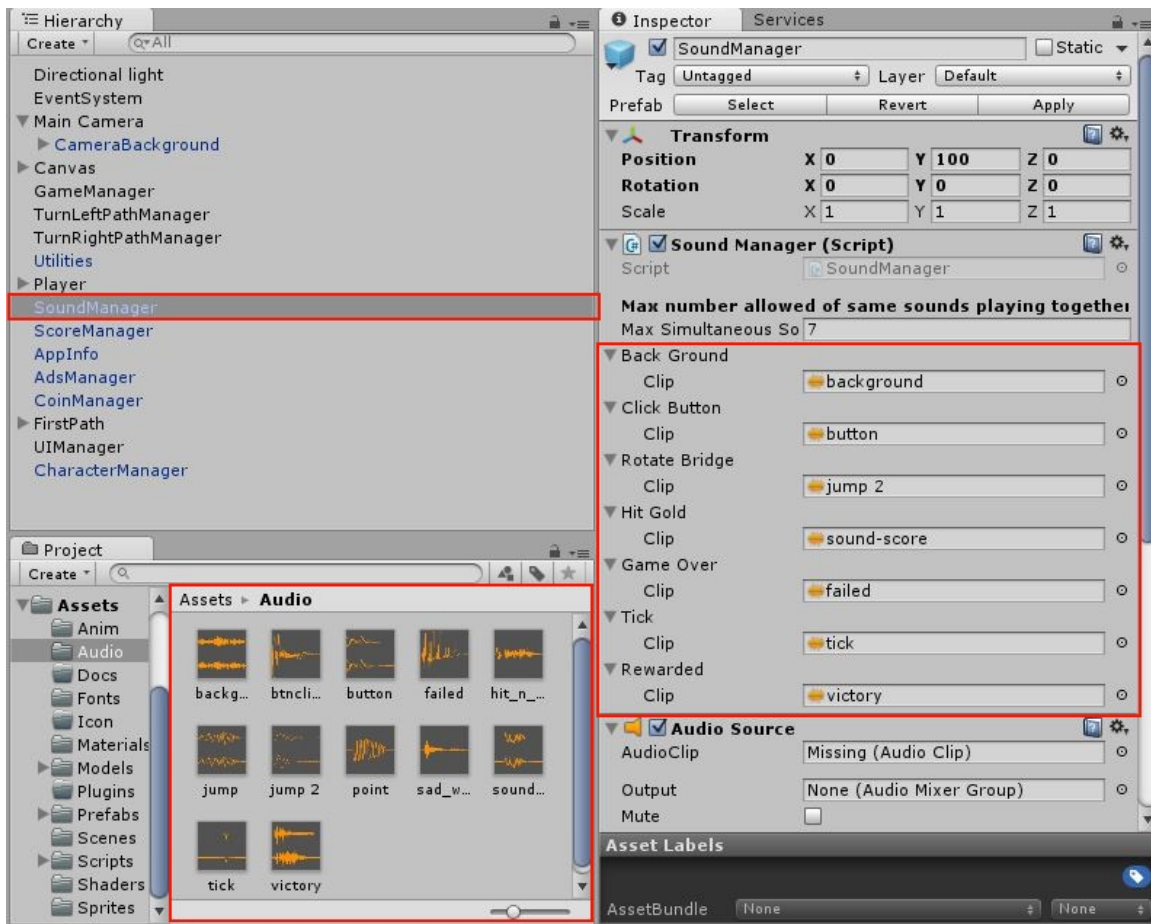


3.4 Sounds

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.



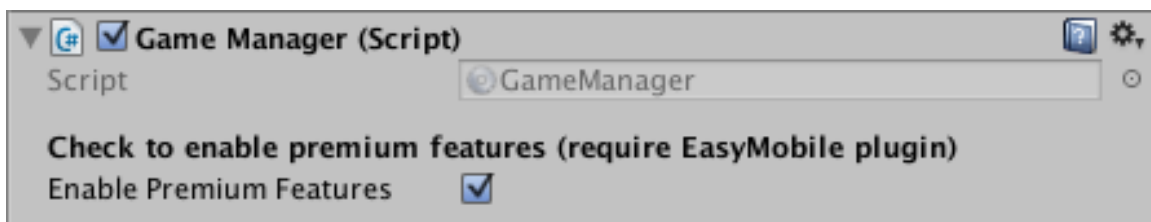
4 ENABLING PREMIUM FEATURES

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

- Make sure the *EnablePremiumFeatures* option in the *GameManager* object is checked.



- Make sure to add the EasyMobile prefab to the Main scene, you can find

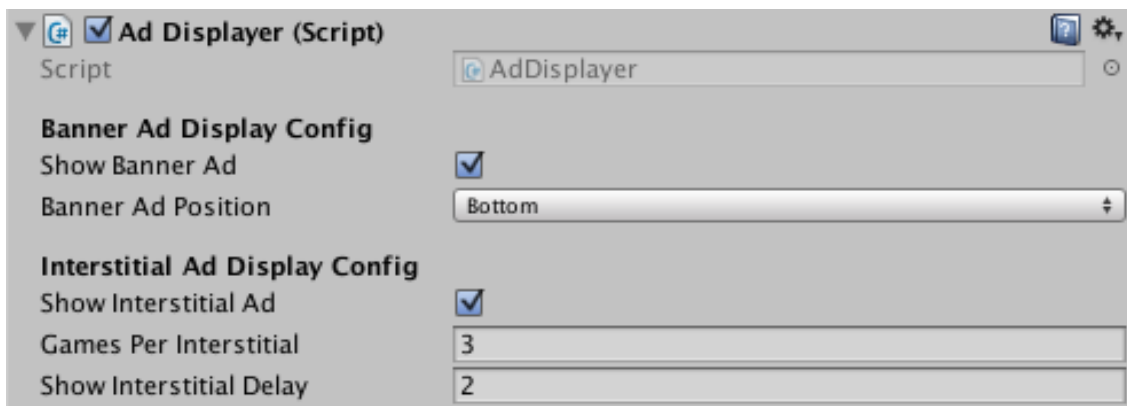
the prefab at folder *Assets/EasyMobile*. It is necessary for the plugin to function properly.

- 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.
- 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.
- 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


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 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

Ad Loading Interval

DEFAULT AD NETWORKS

▼ [iOS] Default Ad Networks

Banner Ad Network

Interstitial Ad Network

Rewarded Ad Network

▼ [Android] Default Ad Networks

Banner Ad Network

Interstitial Ad Network

Rewarded Ad Network

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.

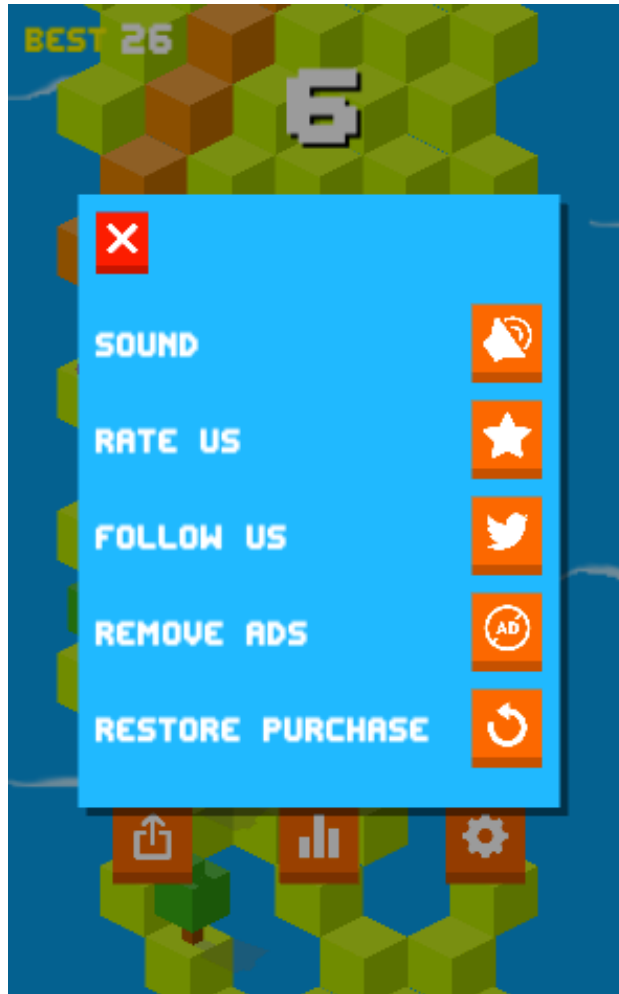
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads 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!

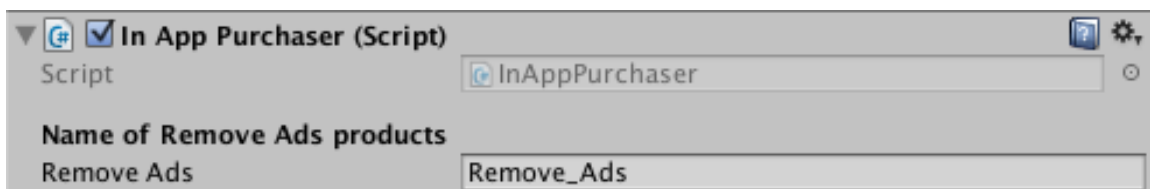
4.3 In-App Purchasing

4.3.1 Template-specific setup

The built-in in-app purchases of this template include a *Remove Ads* button, and a *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.



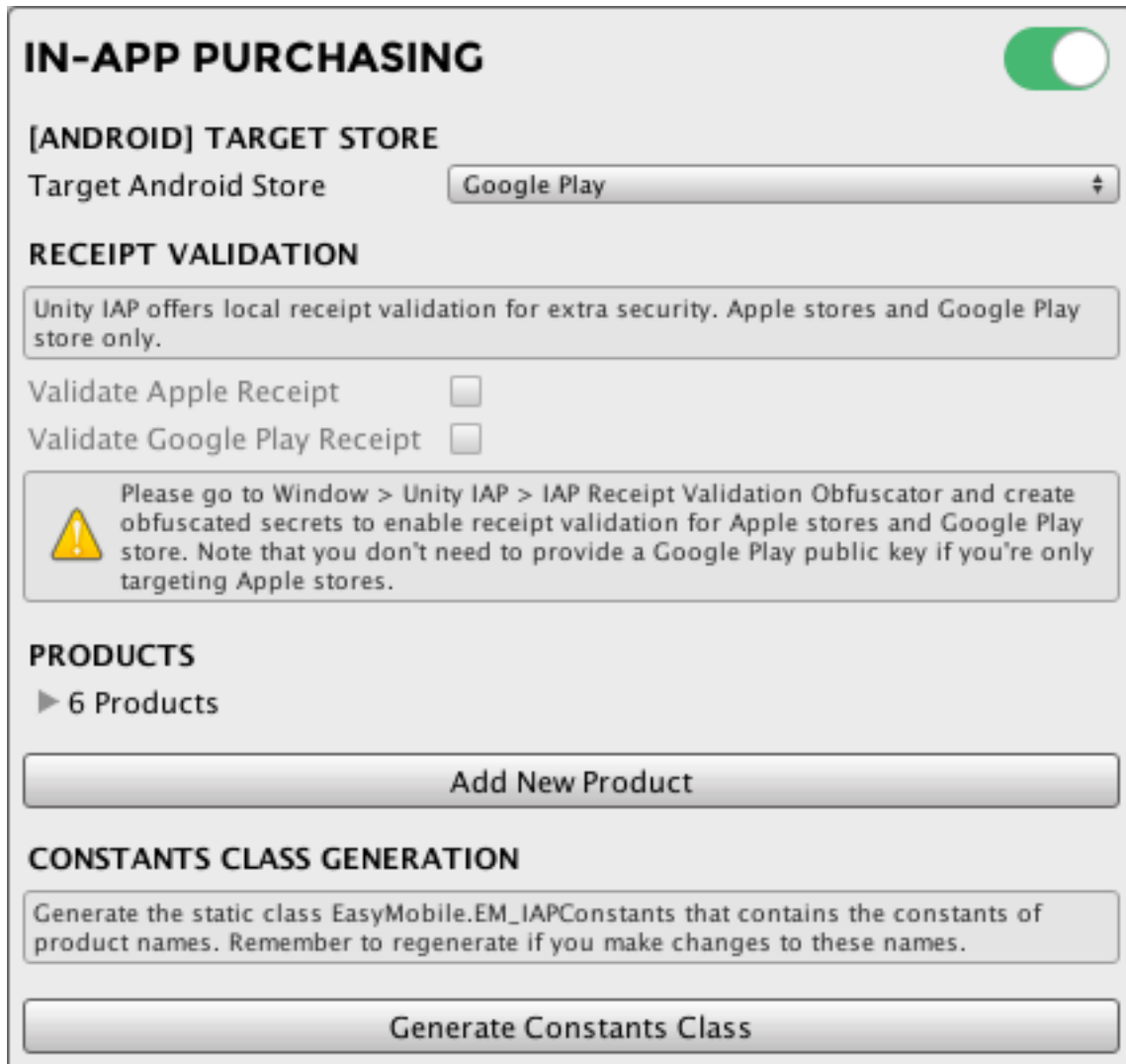
Here you can change the name of the `Remove_Ads` product if you want.

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.



The screenshot shows the 'IN-APP PURCHASING' settings panel. At the top right is a green toggle switch. Below it is the '[ANDROID] TARGET STORE' section with a dropdown menu set to 'Google Play'. The 'RECEIPT VALIDATION' section contains a text box explaining that Unity IAP offers local receipt validation for extra security on Apple and Google Play stores. Below this are two unchecked checkboxes: 'Validate Apple Receipt' and 'Validate Google Play Receipt'. A yellow warning icon is next to a text box instructing the user to go to 'Window > Unity IAP > IAP Receipt Validation Obfuscator' to create obfuscated secrets. The 'PRODUCTS' section shows '6 Products' and an 'Add New Product' button. The 'CONSTANTS CLASS GENERATION' section has a text box explaining that it generates the static class 'EasyMobile.EM_IAPConstants' and a 'Generate Constants Class' button.

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 *RemoveAds* product name declared with Easy Mobile must match with the one you have in the aforementioned *InAppPurchaser* object. Also note that *Remove Ads* should be a non-consumable product.

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.

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

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

GAME SERVICE [Toggle On]

! 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[REDACTED]</string>
<string name="package_name">com.sglib.demogame</string>
<string name="achievement_score_10">CgkI3tzAhK8eEAIQEg</string>
<string name="achievement_score_20">CgkI3tzAhK8eEAIQAA</string>
<string name="achievement_score_30">CgkI3tzAhK8eEAIQEW</string>
</resources>
```

Setup Google Play Games

AUTO-INIT CONFIG

Auto Init ☒

Auto Init Delay

[Android] Max Login Request

LEADERBOARD SETUP

▶ 1 Leaderboards

Add New Leaderboard

ACHIEVEMENT SETUP

▶ 20 Achievements

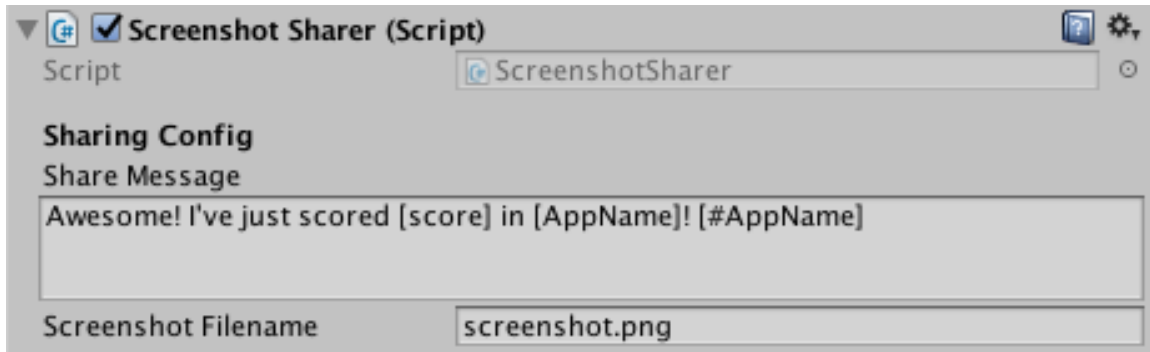
Add New Achievement

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 the game's screenshot 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.



Here you can configure the sharing feature.

- *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
- *Screenshot Filename*: filename to store the screenshot in the device storage

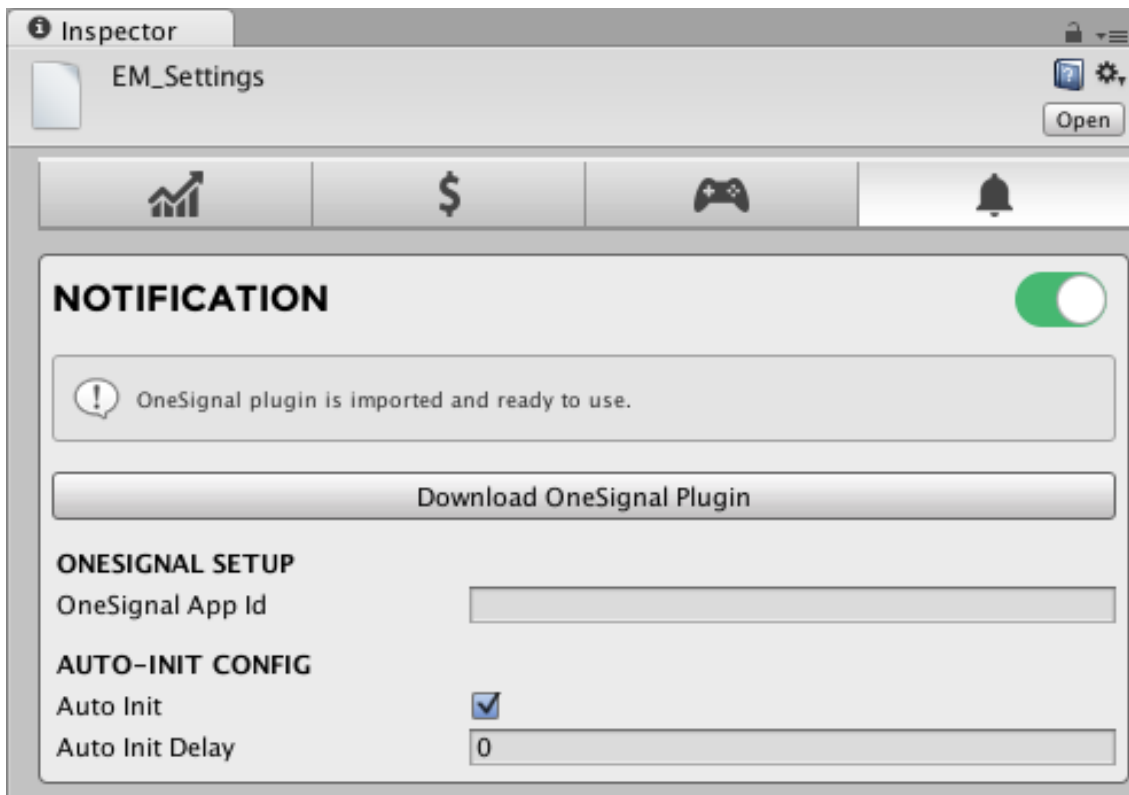
Note that you need to enable the *external write permission* for this feature to function properly on Android. Please see the Native Sharing section in Easy Mobile user guide for detailed instructions on doing that.

4.6 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!