



# ENDLESS GOLF

## 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

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>INTRODUCTION.....</b>                             | <b>3</b>  |
| <b>2</b> | <b>GETTING STARTED.....</b>                          | <b>4</b>  |
| 2.1      | ENTER APP INFORMATION .....                          | 4         |
| 2.2      | LINK THE GAME TO YOUR UNITY PROJECT.....             | 4         |
| <b>3</b> | <b>TEMPLATE CUSTOMIZATION .....</b>                  | <b>6</b>  |
| 3.1      | GAMEPLAY TWEAKING .....                              | 6         |
| 3.2      | CHANGING BACKGROUND .....                            | 8         |
| 3.3      | CUSTOMIZING UI .....                                 | 10        |
| 3.4      | SOUNDS.....  | 10        |
| <b>4</b> | <b>ENABLING PREMIUM FEATURES.....</b>                | <b>12</b> |
| 4.1      | BEFORE YOU BEGIN .....                               | 12        |
| 4.2      | ADVERTISING.....                                     | 12        |
| 4.2.1    | <i>Template-specific setup.....</i>                  | <i>12</i> |
| 4.2.2    | <i>Easy Mobile setup.....</i>                        | <i>13</i> |
| 4.3      | IN-APP PURCHASING.....                               | 14        |
| 4.3.1    | <i>Template-specific setup.....</i>                  | <i>14</i> |
| 4.3.2    | <i>Easy Mobile setup.....</i>                        | <i>15</i> |
| 4.3.3    | <i>Create the products for targeted stores .....</i> | <i>17</i> |
| 4.4      | GAME SERVICE.....                                    | 17        |
| 4.4.1    | <i>Template-specific setup.....</i>                  | <i>17</i> |
| 4.4.2    | <i>Setup for your targeted stores.....</i>           | <i>18</i> |
| 4.4.3    | <i>Easy Mobile setup.....</i>                        | <i>19</i> |
| 4.5      | NATIVE SHARING .....                                 | 20        |
| 4.6      | RATING REQUEST .....                                 | 21        |
| 4.7      | PUSH NOTIFICATION.....                               | 22        |

## 1 INTRODUCTION



Aim and shoot the ball and put it into the hole with the least strokes possible! **Endless Golf** is a gorgeous, simple yet addictive game that is a perfect time-killer for everyone.

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

- Addictive aim-and-shoot 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 AdColony, AdMob, Chartboost, Heyzap and UnityAds
- In-app purchasing (remove ads)
- Support for Game Center (iOS) and Google Play Games Services (Android) for leaderboards and achievements
- Recording gameplay and exporting GIF images
- Sharing to social networks (PNG or GIF images)
- 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!

\* 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

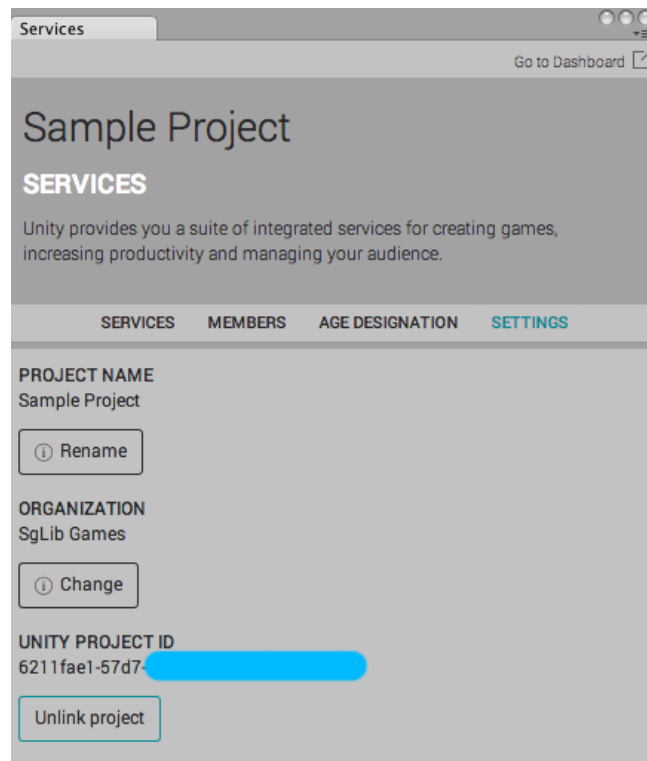
The project contains a game object called AppInfo where you can fill in important app-related metadata like AppStore Id and Bundle Id. These values will be used for features like Rate Us button and opening Facebook or Twitter page.



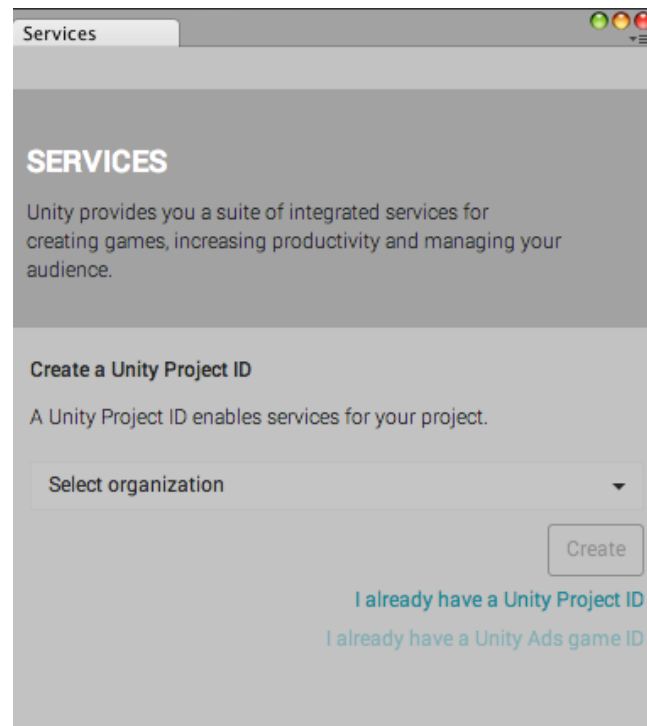
### 2.2 Link the game to your Unity project

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.

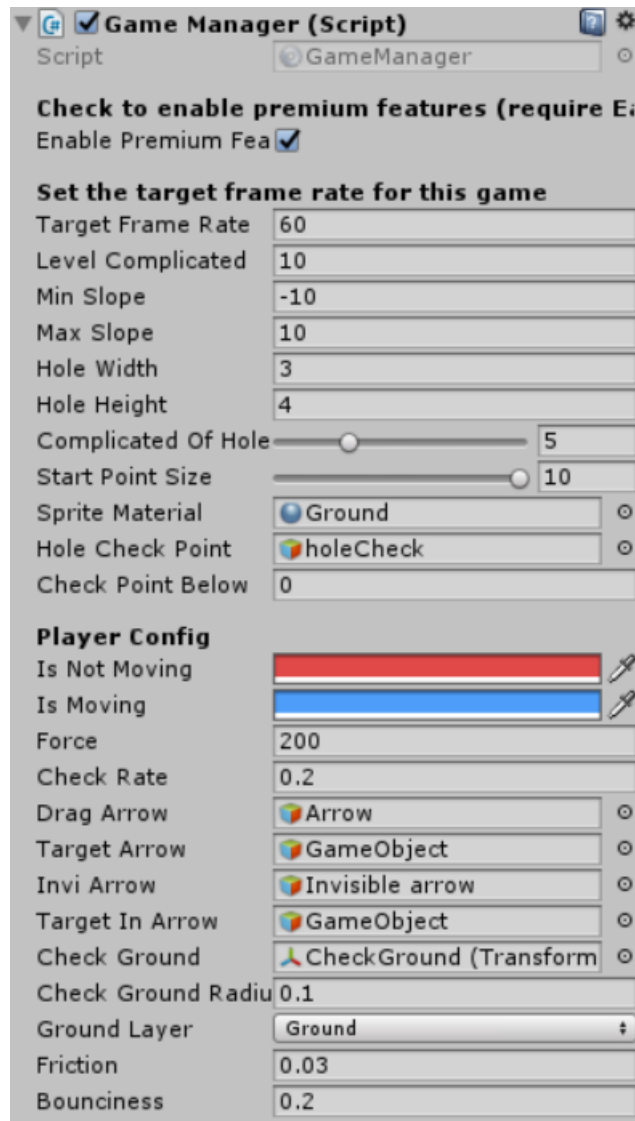


Now your game is linked to your own Unity project and is ready to use Unity services.

## 3 TEMPLATE CUSTOMIZATION

### 3.1 Gameplay tweaking

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



**Gameplay Config**

|                  |                           |
|------------------|---------------------------|
| Number Of Stroke | 5                         |
| Stroke Add       | 2                         |
| Min Wind Force   | 3                         |
| Max Wind Force   | 15                        |
| Body Xscale      | 0.1                       |
| Body Yscale      | 0.1                       |
| Coin Frequency   | <input type="range"/> 0.1 |

**Camera Config**

|                    |    |
|--------------------|----|
| Smooth Time        | 2  |
| Distance Botto Cam | 15 |

**Falling Effect**

|                      |     |
|----------------------|-----|
| Min Falling Speed    | -20 |
| Max Falling Speed    | 5   |
| Ratio With Wind Forc | 1   |

**Object References**

|                   |  |
|-------------------|--|
| Player Controller | <input checked="" type="checkbox"/> Player (PlayerController) <input type="button" value="x"/>     |
| Falling Effect    | <input checked="" type="checkbox"/> FallingEffect (Particle Syst) <input type="button" value="x"/> |

You can tweak the gameplay by modifying following variables:

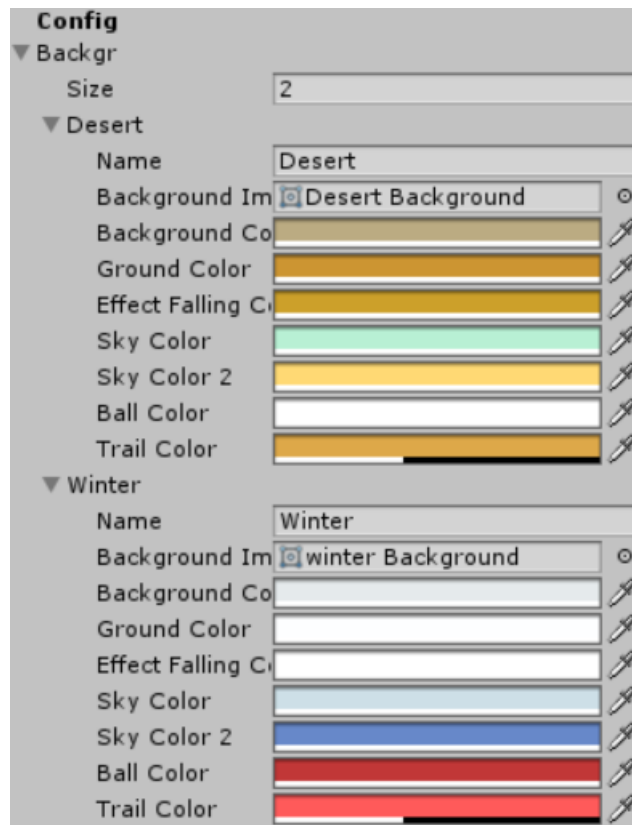
- *TargetFrameRate*: set the target frame rate for the game, which should be at least 60fps for games that requires smooth, fast motion.
- *LevelComplicated*: How complicated the slopes on the ground you want to be.
- *MinSlope,MaxSlope*: The slope of ground. It will be random in these 2 numbers.
- *HoleWidth,HoleHeight*: The width and height of the hole.
- *ComplicatedOfHole*: How complicated you want your hole will be.
- *StartPointSize*: The distance between the first two vertices of ground. The player will places in the middle.
- *SpriteMaterial*: material of ground.
- *HoleCheckPoint,CheckPointBelow*: The GameObject to determine where the hole is. It will be placed below the hole the amount of *CheckPointBelow*.
- *IsNotMoving*: The color of the drag arrow if player can't be hit.
- *IsMoving*: the color of the drag arrow if player can be hit.
- *Force*: force applied on the player when hit.
- *CheckRate*: the position of player will be check continuously between *CheckRate* time. It is used for checking if player is moving or not.

- *DragArrow*: The arrow for dragging.
- *TargetArrow*: The arrow is divided into two parts: Head and body. This is the place to connect them.
- *InviArrow*: The invisible arrow. It could be anything that not *DragArrow* and *WindArrow*. It is used for hitting the player in the right direction.
- *TargetInArrow*: The connection place between Head and Body of *InviArrow*.
- *CheckGround*: The place to check if player is on ground or not.
- *CheckGroundRadius*: The radius to check ground, where the center is *CheckGround* position.
- *GroundLayer*: the ground layer.
- *Friction*: The friction of player.
- *Bounciness*: The bounciness of player.
- *NumberOfStroke*: How many strokes in the beginning.
- *StrokeAdd*: a number of strokes will be added if winning condition is met.
- *MinWindForce,MaxWindForce*: The force in X direction of the wind. It will be randomly picked between these two numbers.
- *BodyXScale,BodyYScale*: The scale in X and Y of the body of *WindArrow*.
- *SmoothTime*: When winning, the camera will move to the new ground in *SmoothTime* time.
- *DistanceBotToCam*: distance from camera to lowest point of current ground.
- *MinFallingSpeed*: min falling speed of effect.
- *MaxFallingSpeed*: max falling speed of effect.
- *RatioWithWindForce*: ratio between horizontal moving speed of effect and winforce.

### 3.2 Changing background

Most of important graphic parameters can be configured within the *BackGroundGenerate* component which is attached to a game object also named *BackGroundHolder* in the hierarchy.





You can tweak the graphic by modifying following variables:

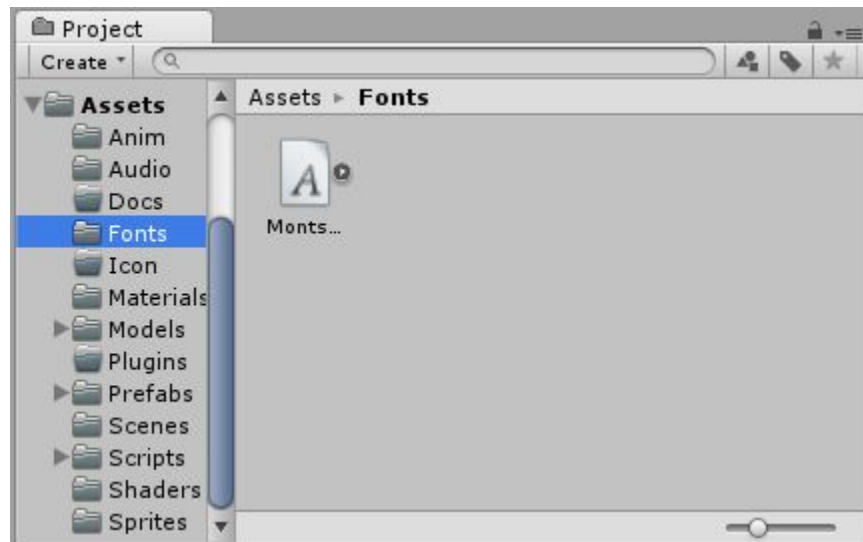
- Name: background's name.
- BackgroundImage: the image of the background in *Sprite* type. The image should be 1280x800 pixel.
- BackGroundColor: color of the bottom background.
- GroundColor: color of the ground that you want to appear with the background
- EffectFallingColor: color of falling effect.
- SkyColor & Skycolor2: two colors that make the gradient effect of the sky.
- BallColor: color of the ball that go with the background.
- TrailColor: color of ball' trail.

### 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.

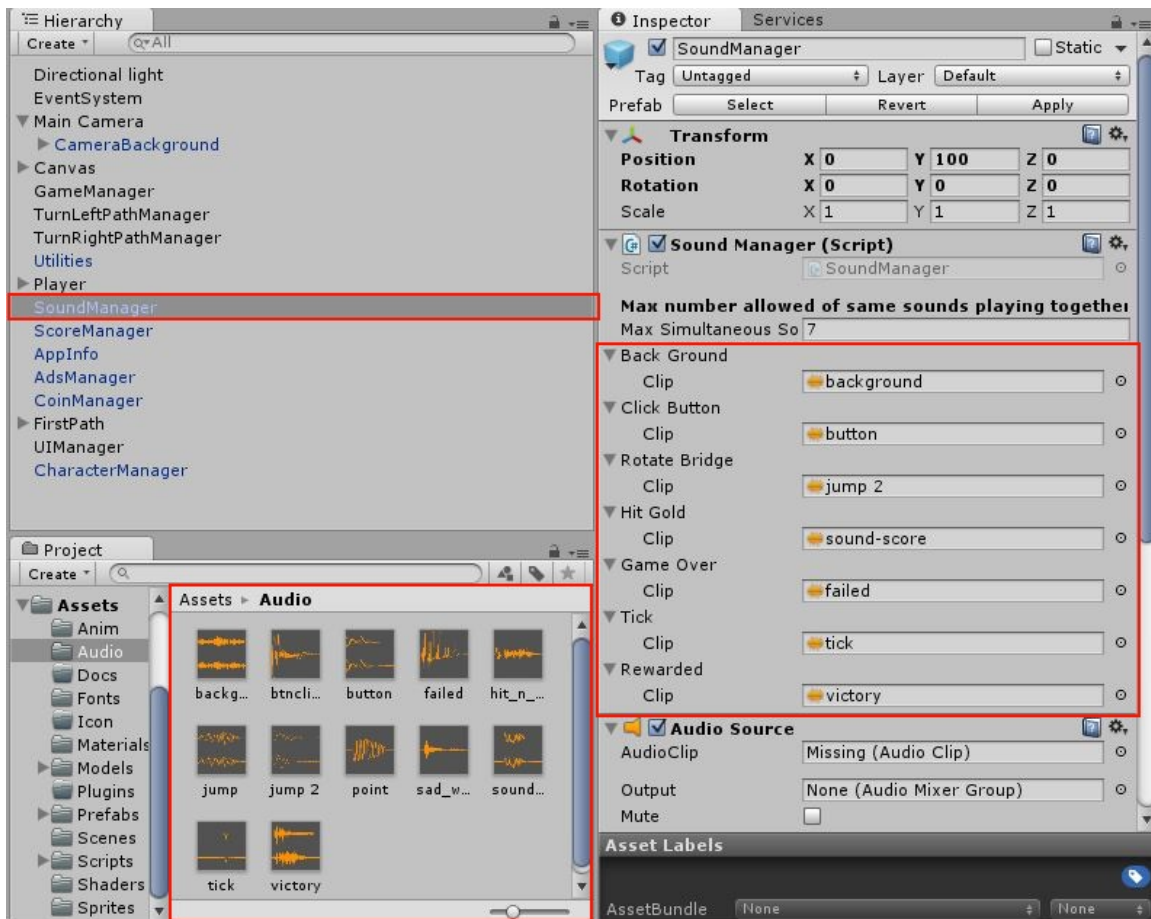


### 3.4 Sounds

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.



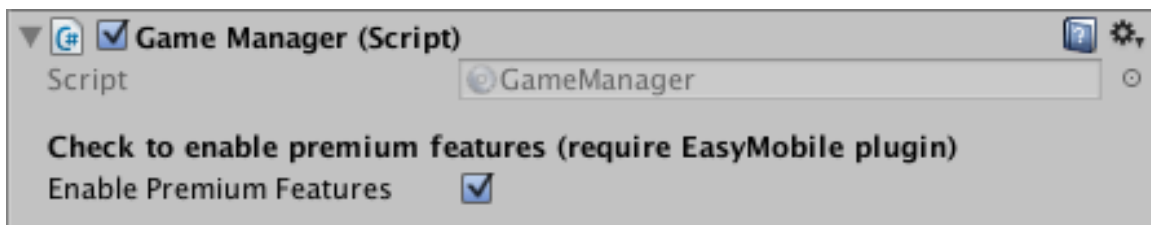
## 4 ENABLING PREMIUM FEATURES

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.

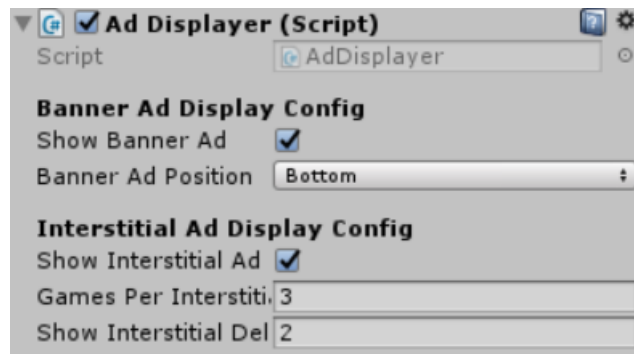


- Make sure to add the EasyMobile prefab to the Main scene, you can find the prefab at folder *Assets/EasyMobile*. It is necessary for the plugin to function properly.
- The settings interface of Easy Mobile can be opened via menu *Window > Easy Mobile > Settings*, this is the only place to go to configure 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**

**ADCOLONY SETUP**

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

Download AdColony Plugin

**ADMOB SETUP**

! Google Mobile Ads (AdMob) plugin was imported.

Download Google Mobile Ads Plugin

**AdMob IDs**

▶ iOS

▶ Android

**Ad Targeting**

Gender: Unspecified

Tag For Child Directed Treatment: Unspecified

Extras: 0

**Test Mode**

Enable Test Mode: ☐

**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.

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.

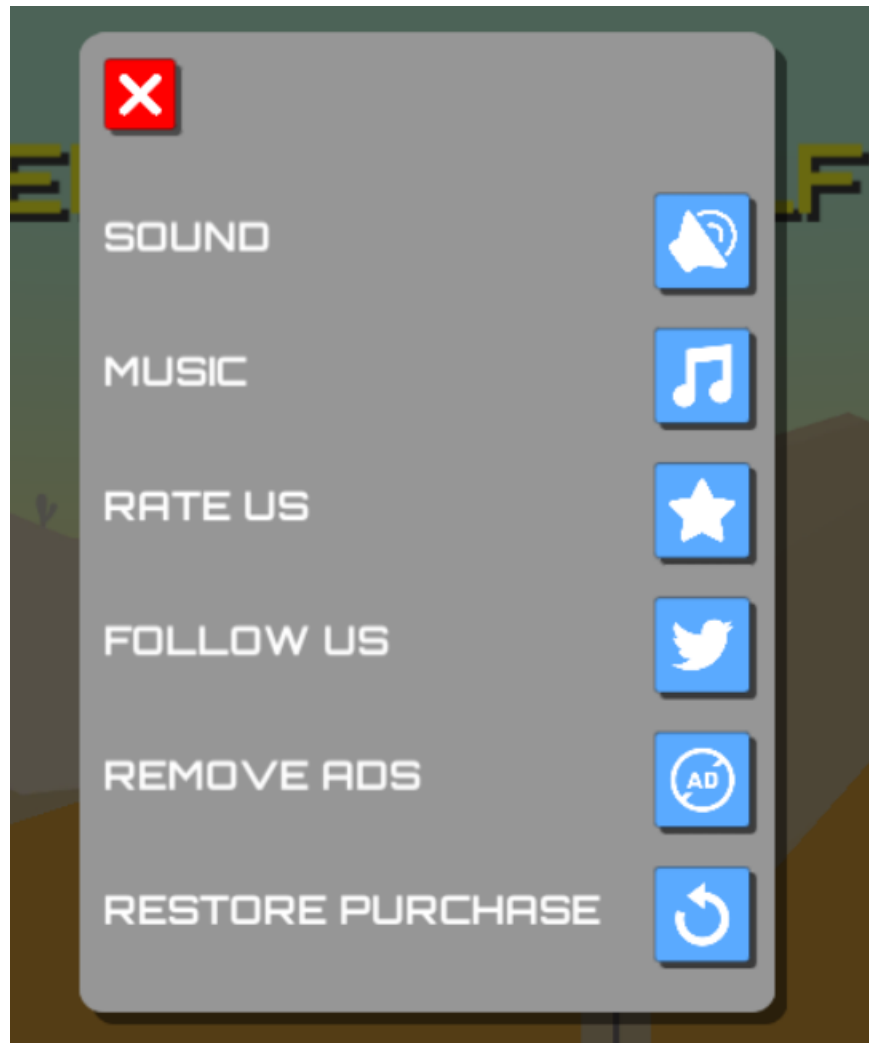
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads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



The screenshot shows a configuration window for in-app purchases. It contains two main sections, each with a title, a 'Name' field, a 'Type' dropdown, an 'Id' field, and a 'More (Optional)' link. To the right of each section are three buttons: an up arrow, a minus sign, and a down arrow.

| Product Name | Type           | Id                            |
|--------------|----------------|-------------------------------|
| Remove_Ads   | Non Consumable | sglib.demogame.iap.remove_ads |
| Coins_500    | Consumable     | sglib.demogame.iap.coins_500  |

### 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.

The screenshot shows a configuration window for the 'Score Reporter (Script)' component. It has a 'Script' dropdown set to 'ScoreReporter'. Below this is a checkbox labeled 'Check to disable automatic score reporting' which is currently unchecked. At the bottom, there is a text field for 'Name of the leaderboard to report score as declared with EasyMobile' with the value 'Score'.

| Property  | Value                    |
|---|--------------------------|
| Script  | ScoreReporter            |
| Check to disable automatic score reporting                          | <input type="checkbox"/> |
| Name of the leaderboard to report score as declared with EasyMobile | 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: AchievementUnlocker

**Check to disable automatic achievement unlocking**  
 Disable ☐

**List of achievements to unlock**

| Achievements     | Size     |
|------------------|----------|
| ▼ Score_10       |          |
| Achievement Name | Score_10 |
| Score To Unlock  | 10       |
| ▼ Score_20       |          |
| Achievement Name | Score_20 |
| Score To Unlock  | 20       |
| ▼ Score_30       |          |
| Achievement Name | Score_30 |
| Score To Unlock  | 30       |
| ▼ Score_40       |          |
| Achievement Name | Score_40 |
| Score To Unlock  | 40       |
| ▼ Score_50       |          |
| Achievement Name | Score_50 |
| Score To Unlock  | 50       |
| ▼ Score_60       |          |
| Achievement Name | Score_60 |
| Score To Unlock  | 60       |
| ▼ Score_70       |          |
| Achievement Name | Score_70 |

#### 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**

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">Cgkl3tzAhK8eEAIQAQ</string>
  <string name="achievement_score_30">Cgkl3tzAhK8eEAIQEW</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 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.

The image shows a configuration window for a script named "Screenshot Sharer (Script)". The window has a title bar with a blue icon and a settings gear icon. Below the title bar, there's a "Script" section with a dropdown menu showing "ScreenshotSharer".

The main configuration area is divided into several sections:

- Check to disable sharing:** A checkbox labeled "Disable Sharing" is currently unchecked.
- Sharing Config:**
  - Shared Image Format:** A dropdown menu set to "Both".
  - Share Message:** A text field containing the message: "Awesome! I've just scored [score] in [AppName]! [#AppName]".
- Gif Filename:** A text field containing "animated\_screenshot".
- Png Filename:** A text field containing "screenshot".
- GIF Settings:**
  - Gif Auto Height:** A checked checkbox.
  - Gif Width:** A text field containing "320".
  - Gif Height:** A text field containing "480".
  - Gif Fps:** A slider control ranging from 0 to 15, currently set at 15.
  - Gif Length:** A slider control ranging from 0 to 3, currently set at 3.
  - Gif Loop:** A text field containing "0".
  - Gif Quality:** A slider control ranging from 0 to 80, currently set at 80.
  - Gif Thread Priority:** A dropdown menu set to "Below Normal".
  - Enable Easy Mobile Attribution Mark:** A checked checkbox.
- Giphy Credentials - leave both empty to use Giphy Beta key:**
  - Giphy Username:** An empty text field.
  - Giphy Api Key:** An empty text field.
  - Giphy Upload Tags:** A text field containing "dashy, worm".

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
- *Enable Easy Mobile Attribution Mark*: whether to display Easy Mobile attribution when generating GIF images

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

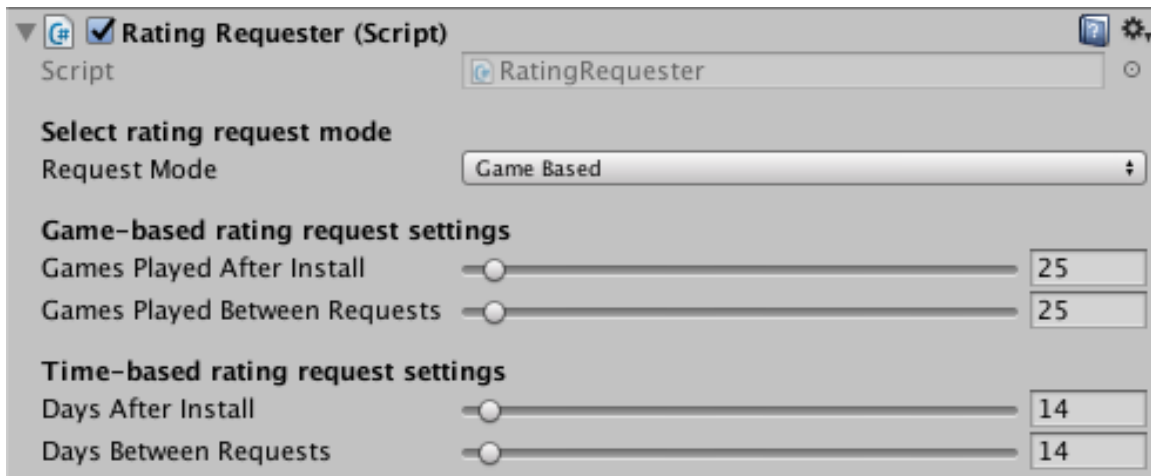
- *Giphy Username & Giphy Api Key*: provide these values if you want to upload the GIF image to your own Giphy channel; otherwise leave them empty to use the [Giphy beta key](#)
- *Giphy Upload Tags*: comma-delimited list of tags of the uploaded image

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 Notifications 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.

**NOTIFICATIONS**

**AUTO-INIT CONFIG**

Auto Init ☒

Auto Init Delay 0

**REMOTE NOTIFICATION SETUP**

Push Notification Service

None

✓ One Signal

! OneSignal plugin is imported and ready to use.

Download OneSignal Plugin

OneSignal App Id

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

**THANK YOU AND GOOD LUCK WITH YOUR GAMES!**