



BRAIN 100 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!

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	4
3	TEMPLATE CUSTOMIZATION	6
3.1	GAMEPLAY TWEAKING	6
3.2	CHANGING BACKGROUND COLOR	7
3.3	CUSTOMIZING UI	8
3.4	SOUNDS	9
4	ENABLING PREMIUM FEATURES	11
4.1	BEFORE YOU BEGIN	12
4.1.1	<i>Easy Mobile setup</i>	<i>12</i>
4.2	IN-APP PURCHASING	14
4.2.1	<i>Template-specific setup</i>	<i>14</i>
4.2.2	<i>Easy Mobile setup</i>	<i>14</i>
4.2.3	<i>Create the products for targeted stores</i>	<i>16</i>
4.3	GAME SERVICE	16
4.3.1	<i>Template-specific setup</i>	<i>16</i>
4.3.2	<i>Setup for your targeted stores</i>	<i>17</i>
4.3.3	<i>Easy Mobile setup</i>	<i>18</i>
4.4	NATIVE SHARING	19
4.5	RATING REQUEST	19
4.6	PUSH NOTIFICATION	20

1 INTRODUCTION

BRAIN 100



Pre-integrated with



EASY MOBILE

sglib

Brain 100 is a little game that puts your memory to the ultimate test. Can you memorize all the tiles with the cute cat on a 3x3 board? Easy? How about a 4x4, 5x5 or 6x6 one? Well, maybe it's not as easy as it looks. Seriously, this is an addictive game that will keep its player entertained for hours.

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

- Addictive gameplay
- Clean, minimalist design
- Free-to-use assets (fonts, sounds, music, model, etc.)
- Optimized for mobile

Most importantly, this template includes **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 (with mediation) 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 notifications using OneSignal
- Native rating request popup (rate my app)

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.

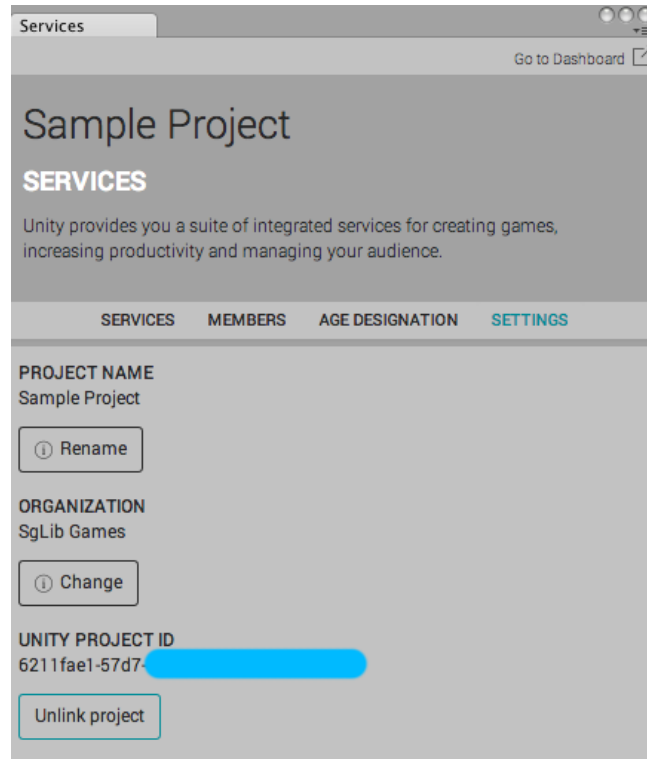


Script	AppInfo
APP_NAME	[YOUR_APP_NAME]
APPSTORE_ID	[YOUR_APPSTORE_ID]
BUNDLE_ID	[YOUR_BUNDLE_ID]
APPSTORE_HOMEPAGE	[YOUR_APPSTORE_PUBLISHER_LINK]
PLAYSTORE_HOMEPAGE	[YOUR_GOOGLEPLAY_PUBLISHER_NAME]
FACEBOOK_ID	[YOUR_FACEBOOK_PAGE_ID]
TWITTER_NAME	[YOUR_TWITTER_PAGE_NAME]
SUPPORT_EMAIL	[YOUR_SUPPORT_EMAIL]

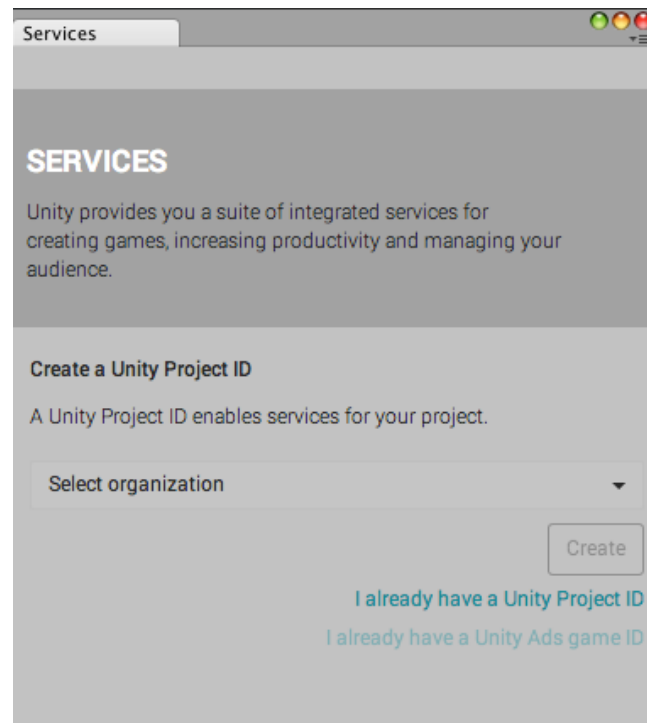
2.2 Link the game to your Unity project

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.



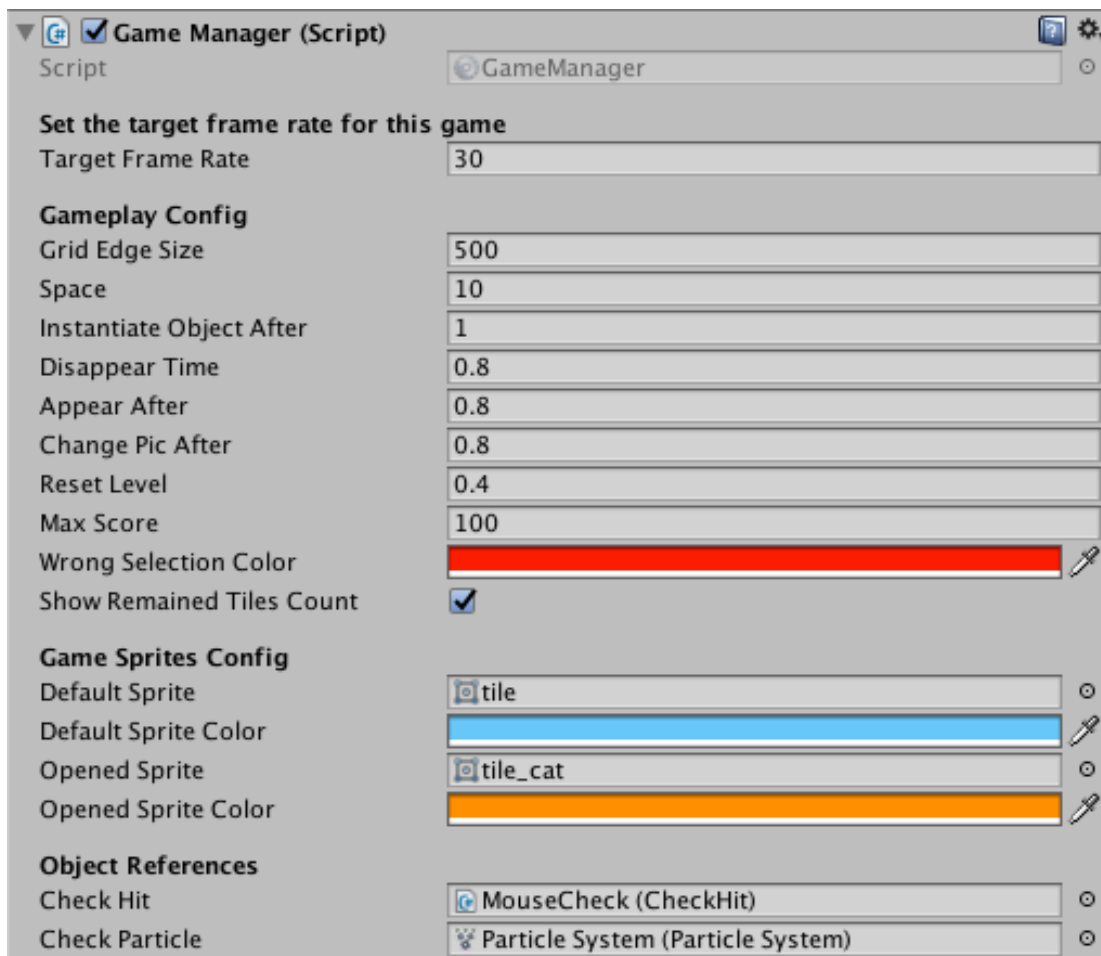
Now your game is linked to your own Unity project and is ready to use Unity

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.



You can tweak the gameplay by modifying following variables:

- *TargetFrameRate*: the target frame rate for the game.
- *GridEdgeSize*: is the sizes of the board that contains all the tiles (squares) game object
- *Space*: The space between 2 tiles on the board
- *InstantiateGameObjectAfter*: after you win several levels, the number of

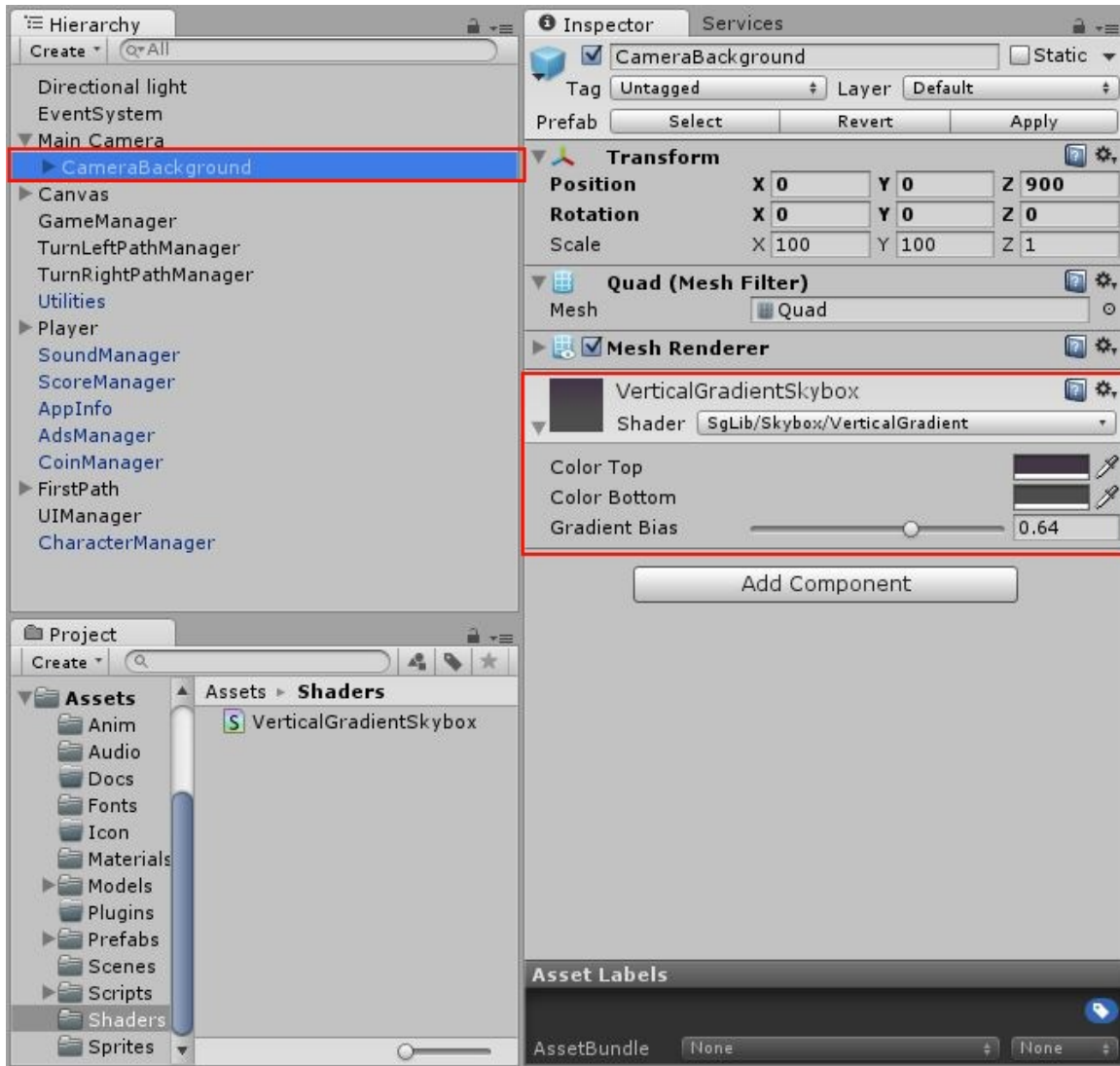
tiles (the sizes of the board) will increase (for example 3x3 -> 4x4). *InstantiateGameObjectAfter* is the amount of time the board will create more tiles after you won some rounds.

- *DisappearTime*: after you complete a board, the board will disappear for the amount of *DisappearTime*
- *AppearAfter*: at the start of a new round, the target tiles (the cat-tiles) will be revealed after the *AppearAfter* time.
- *ChangePicsAfter*: the revelation time of the target tiles, they will be hidden (changed back to normal tiles) after this time
- *ResetLevel*: After you complete a board, the new board will be reset after *ResetLevel* time
- *MaxScore*: the “target” score of this one, to fill the score circle when the game ends.
- *WrongSelectionColor*: color of the tile incorrectly selected
- *ShowRemainedTilesCount*: whether to show the number of remaining hidden tiles at the bottom of the board
- *DefaultSprite* & *DefaultSpriteColor*: the sprite and tint color of the normal tiles
- *OpenedSprite* & *OpendSpriteColor*: the sprite and tint color of the target (hidden) tiles

3.2 Changing background color

The background color in this game is implemented by a gradient shader call *VerticalGradientSkybox* located in the *Shaders* folder. It's an easy-to-use vertex/fragment shader. To modify the background gradient colors:

- Navigate to game object *CameraBackground* under *MainCamera*.
- Modify the gradient colors: *ColorTop*, *ColorBottom* and adjust the *GradientBias* until you're satisfied with the result.
- Hit *Apply* to save changes.

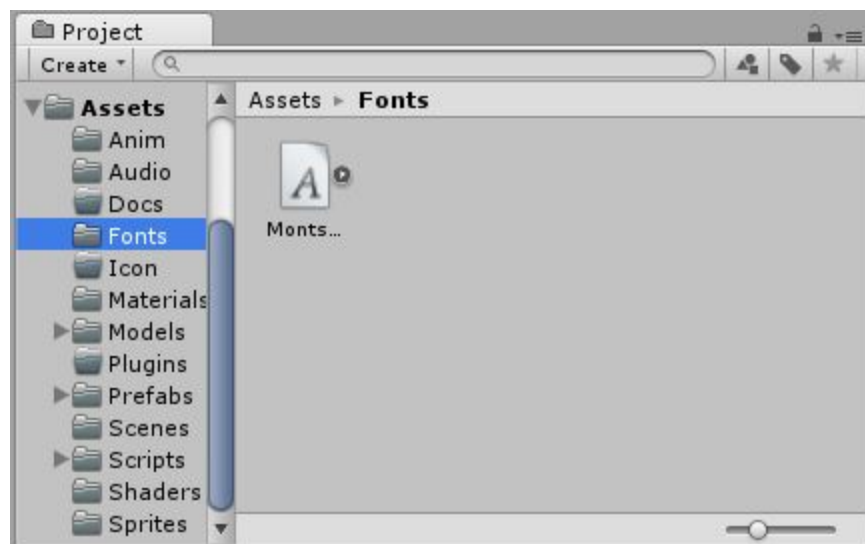


3.3 Customizing UI

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.

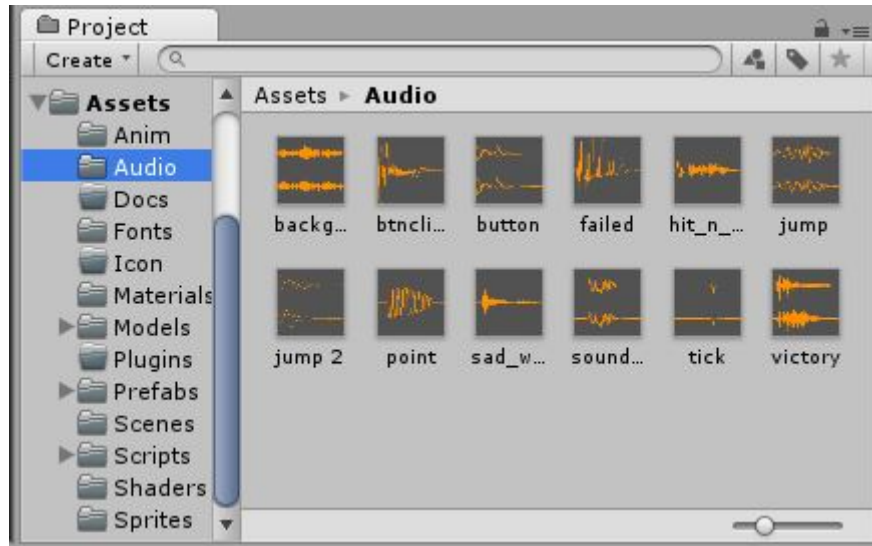


All fonts used in this game are free-to-use in commercial projects. Fonts are located under the *Fonts* folder together with appropriate license files.

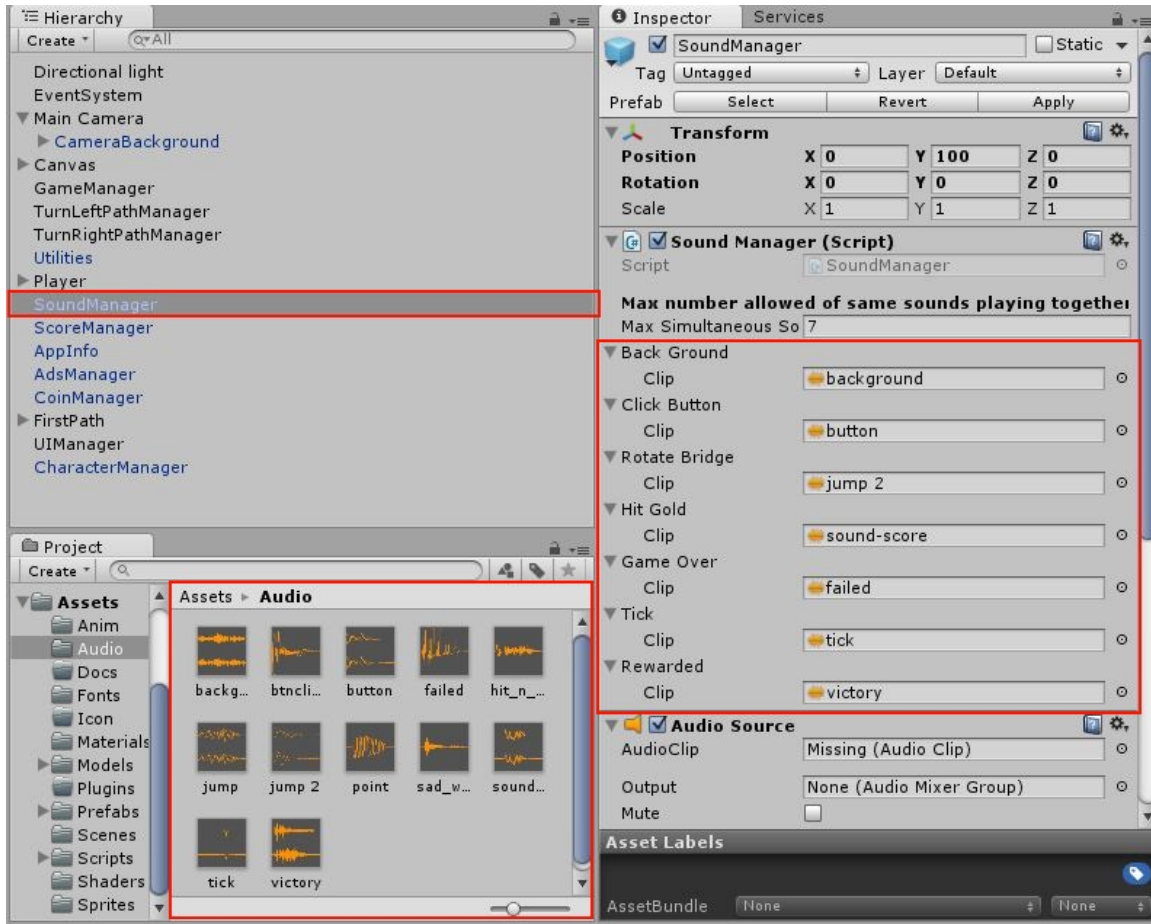


3.4 Sounds

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

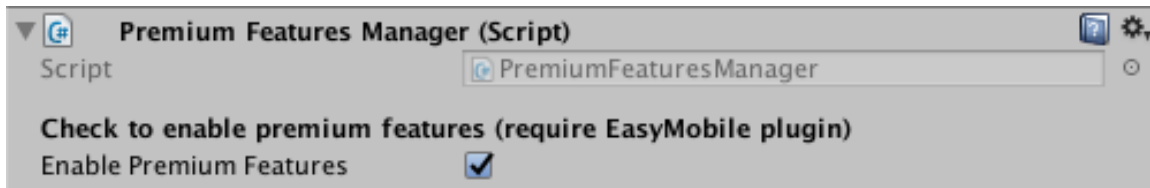
Premium features include:

- Advertising
- In-app purchasing
- Leaderboards & achievements
- Sharing
- Push notifications
- Rating request

This section provides a guide on configuring each feature for your game. As the features are implemented using EasyMobile plugin, this guide is intended to be used alongside EasyMobile's official user guide. Therefore, it is strongly recommended that you also go through that 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 *PremiumFeaturesManager* object is checked.



- Make sure to add an instance of 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).

4.1.1 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.

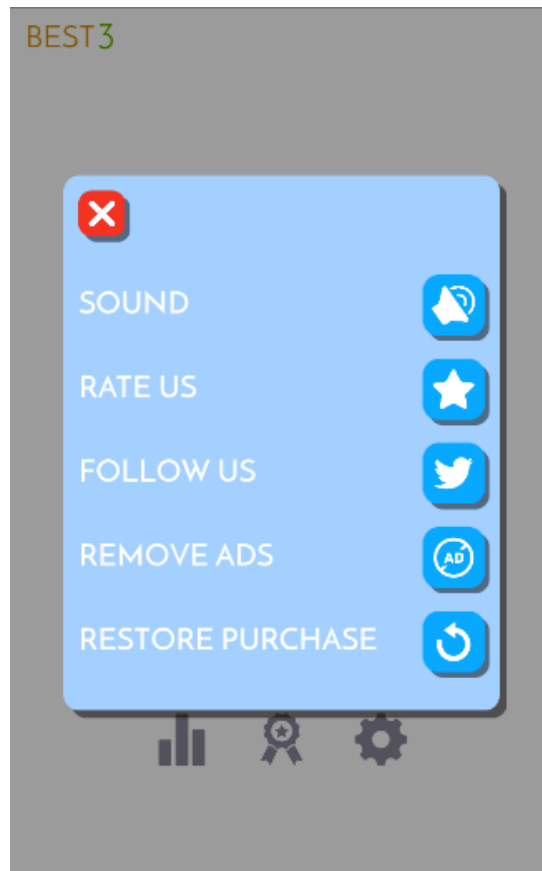
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads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

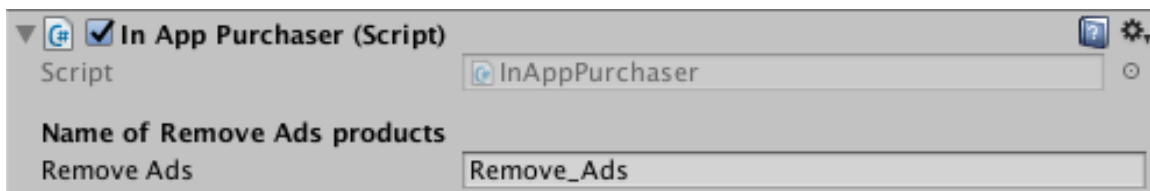
4.2 In-App Purchasing

4.2.1 Template-specific setup

This template has a *Remove Ads* button for the user to pay to remove in-app advertisements. It also has 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.



You can change the `Remove_Ads` name if you wish, just make sure this name matches the one declared in Easy Mobile settings.

4.2.2 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

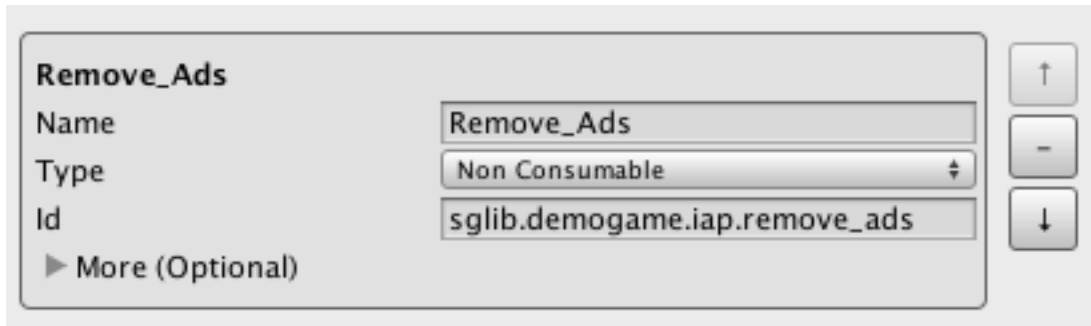
Add New Product

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* must be a non-consumable product.



4.2.3 Create the products for targeted stores

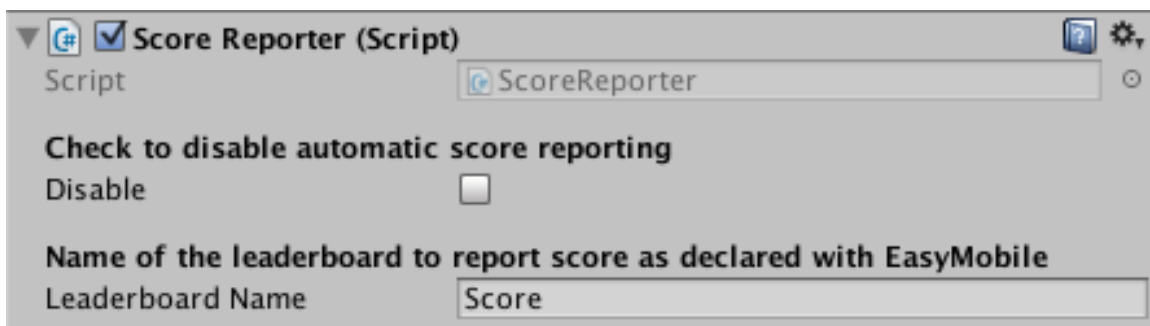
That last step in configuring the in-app purchasing feature is to create products for your targeted stores (e.g. Google Play and Apple App Store). Make sure the product ID, product type and price match the ones you have in your game.

4.3 Game Service

4.3.1 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.



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.

The screenshot shows a software interface titled "Achievement Unlocker (Script)". It has a "Script" tab and a text field containing "AchievementUnlocker". Below this, there is a section "Check to disable automatic achievement unlocking" with a "Disable" checkbox that is currently unchecked. The main section is "List of achievements to unlock", which is expanded to show a list of achievements. Each achievement has a "Size" field and a "Score To Unlock" field. The achievements listed are Score_10, Score_20, Score_30, Score_40, Score_50, Score_60, and Score_70. The "Score To Unlock" values are 10, 20, 30, 40, 50, 60, and 70 respectively. The "Achievement Name" field for each achievement is also filled with the same name.

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

4.3.2 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.

4.3.3 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.

The screenshot shows the 'GAME SERVICE' settings window. At the top, there's a toggle switch for 'GAME SERVICE' which is turned on. Below it, a message states 'Google Play Games plugin is imported and ready to use.' with a 'Reimport Google Play Games Plugin' button. The '[ANDROID] GOOGLE PLAY GAMES SETUP' section includes a 'GPGS Debug Log' checkbox and a text area for pasting Android XML Resources. The XML content is as follows:

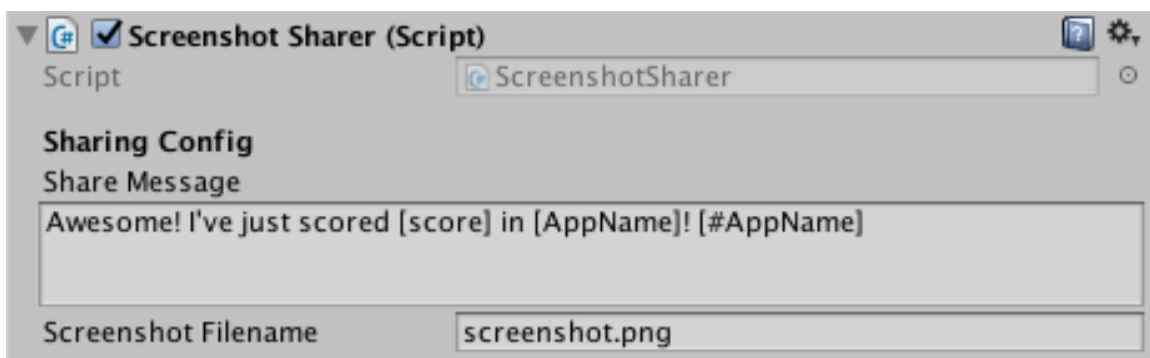
```
<?xml version="1.0" encoding="utf-8"?>
<!--
Google Play game services IDs.
Save this file as res/values/games-ids.xml in your project.
-->
<resources>
  <string name="app_id">104[REDACTED]</string>
  <string name="package_name">com.sglib.demogame</string>
  <string name="achievement_score_10">Cgkl3tzAhK8eEAIQEg</string>
  <string name="achievement_score_20">Cgkl3tzAhK8eEAIQAQ</string>
  <string name="achievement_score_30">Cgkl3tzAhK8eEAIQEW</string>
</resources>
```

Below the XML editor is a 'Setup Google Play Games' button. The 'AUTO-INIT CONFIG' section has 'Auto Init' checked, 'Auto Init Delay' set to 0, and '[Android] Max Login Requests' set to 3. The 'LEADERBOARD SETUP' section shows '1 Leaderboards' and an 'Add New Leaderboard' button. The 'ACHIEVEMENT SETUP' section shows '20 Achievements' and an 'Add New Achievement' button.

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.

4.4 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

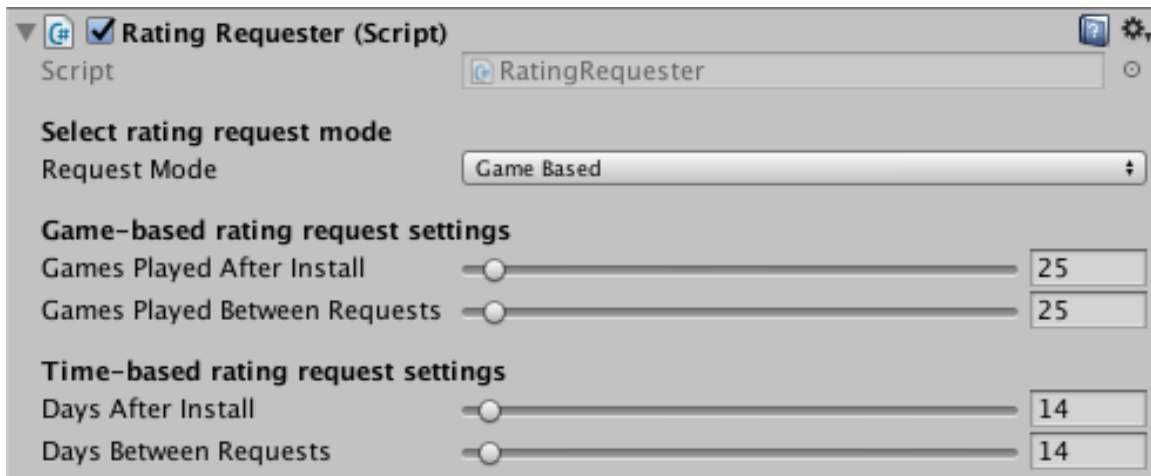
Note that you need to enable the *external write permission* for this feature to function properly on Android. Please see the Native Sharing section in Easy Mobile user guide for detailed instructions on doing that.

4.5 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

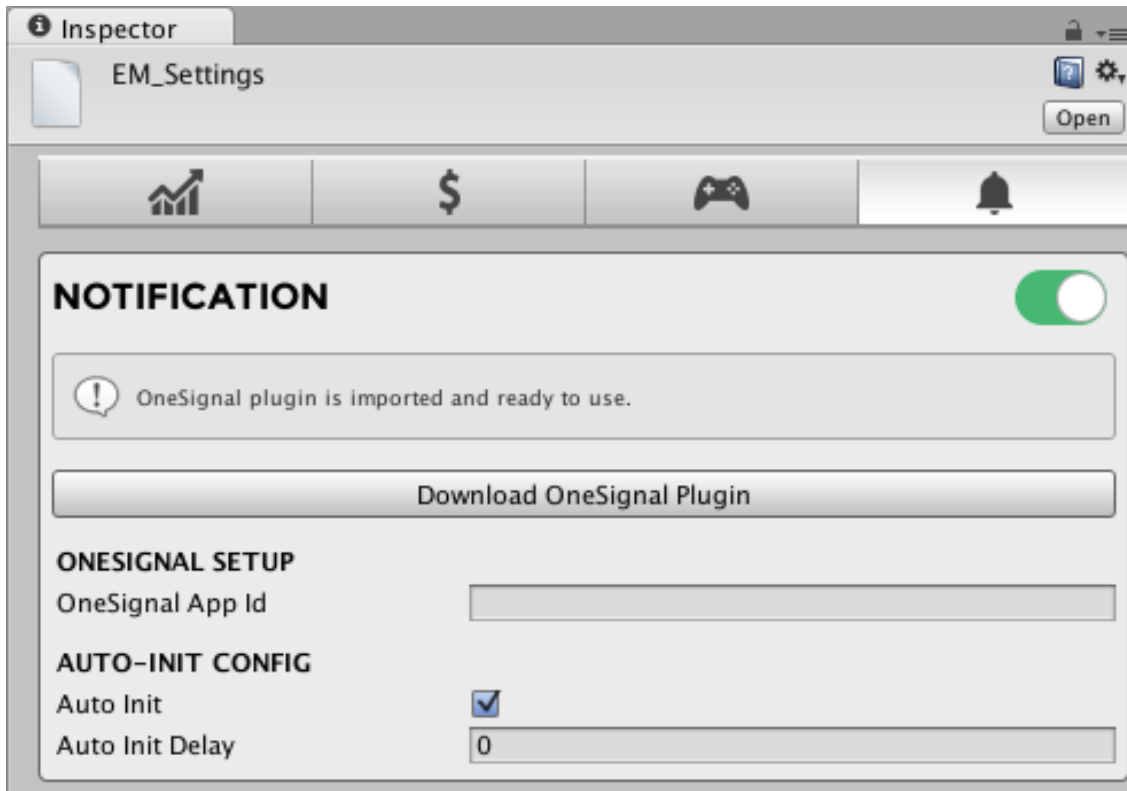
4.6 Push notification

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!