



SKY JUMP TEMPLATE

USER GUIDE

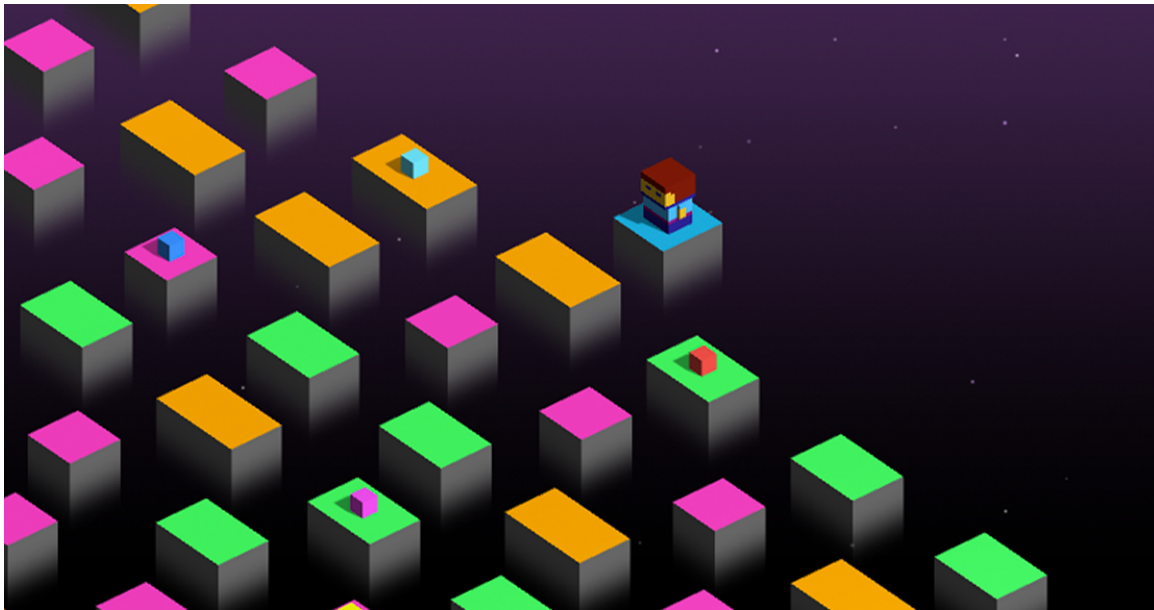
We strive to provide the best service as we can, if you have any questions or suggestions, please contact us!
Thank you!

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.....	5
2.3	TESTING NOTE	6
3	TEMPLATE CUSTOMIZATION	6
3.1	GAMEPLAY TWEAKING	6
3.2	CHANGING PLANE COLOR	8
3.3	DAILY REWARD FEATURE	9
3.4	ADDING MORE CHARACTERS	10
3.5	CHANGING BACKGROUND GRADIENT	11
3.6	CUSTOMIZING UI	12
3.7	SOUNDS.....	13
4	ENABLING PREMIUM FEATURES.....	15
4.1	BEFORE YOU BEGIN	15
4.2	ADVERTISING.....	15
4.2.1	<i>Template-specific setup.....</i>	<i>15</i>
4.2.2	<i>Easy Mobile setup.....</i>	<i>16</i>
4.3	IN-APP PURCHASING.....	18
4.3.1	<i>Template-specific setup.....</i>	<i>18</i>
4.3.2	<i>Easy Mobile setup.....</i>	<i>20</i>
4.3.3	<i>Create the products for targeted stores</i>	<i>22</i>
4.4	GAME SERVICE.....	22
4.4.1	<i>Template-specific setup.....</i>	<i>22</i>
4.4.2	<i>Setup for your targeted stores.....</i>	<i>23</i>
4.4.3	<i>Easy Mobile setup.....</i>	<i>24</i>
4.5	NATIVE SHARING	25
4.6	PUSH NOTIFICATION.....	25

1 INTRODUCTION



Tap to jump on moving platforms and travel the sky! Don't fall, it hurts! Seriously, **Sky Jump** is a gorgeous, addictive game that will keep the player entertained for hours.

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

- Addictive one-touch gameplay
- Daily reward system for better retention
- 30 built-in unlock-able characters with cute blocky style
- 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

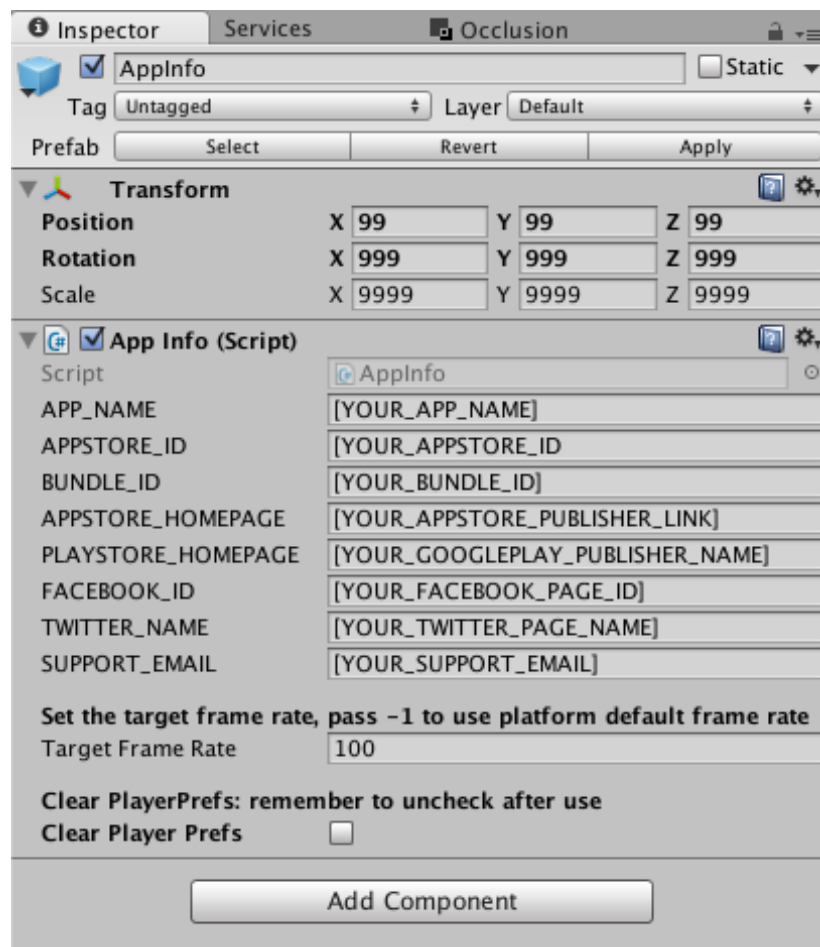
of integration code!

** 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.



Besides basic app info, there're 2 other important variables:

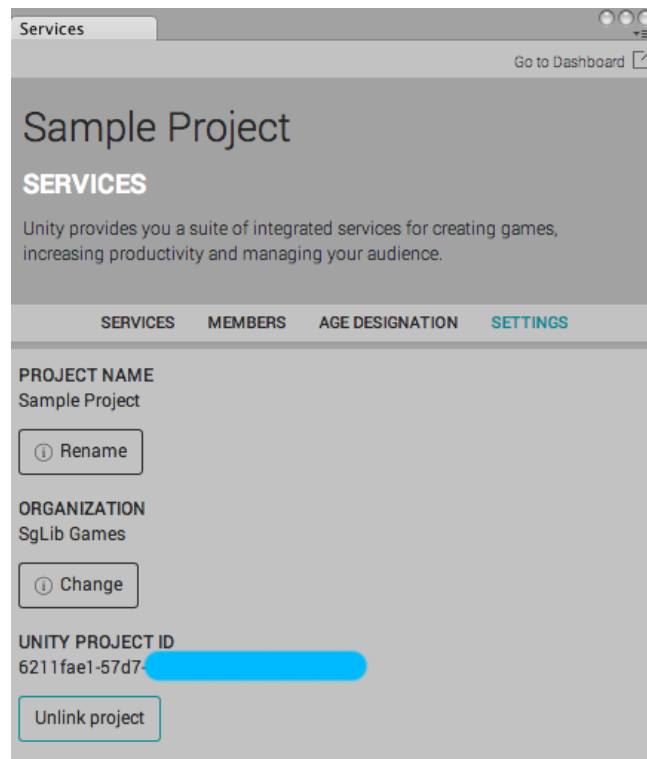
- *TargetFrameRate*: set the target frame rate for the game, which should be at least 60fps for smooth motion.
- *ClearPlayerPrefs*: a convenient variable used for debugging purpose, check this and run the game will clear the *PlayerPrefs* (using

PlayerPrefs.DeleteAll() method). It only works in the editor but it's worth remembering to uncheck the variable after use to avoid later confusion.

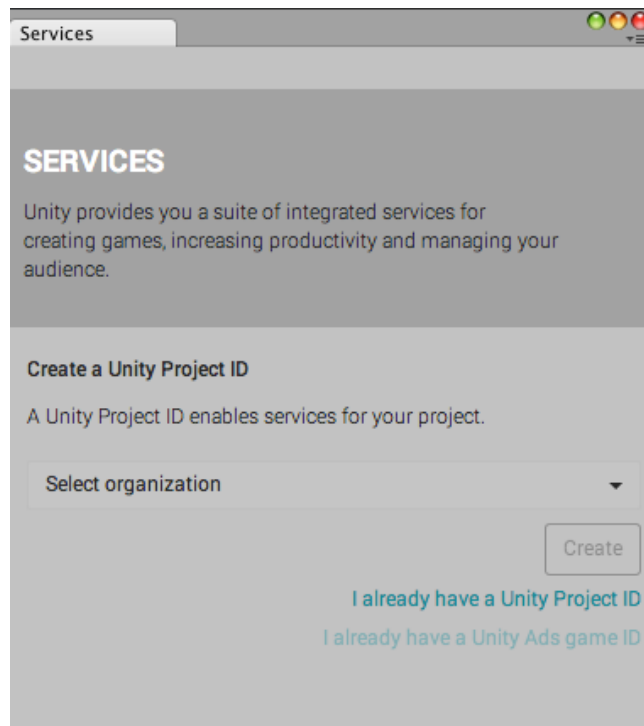
2.2 Link the game to your Unity project

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.

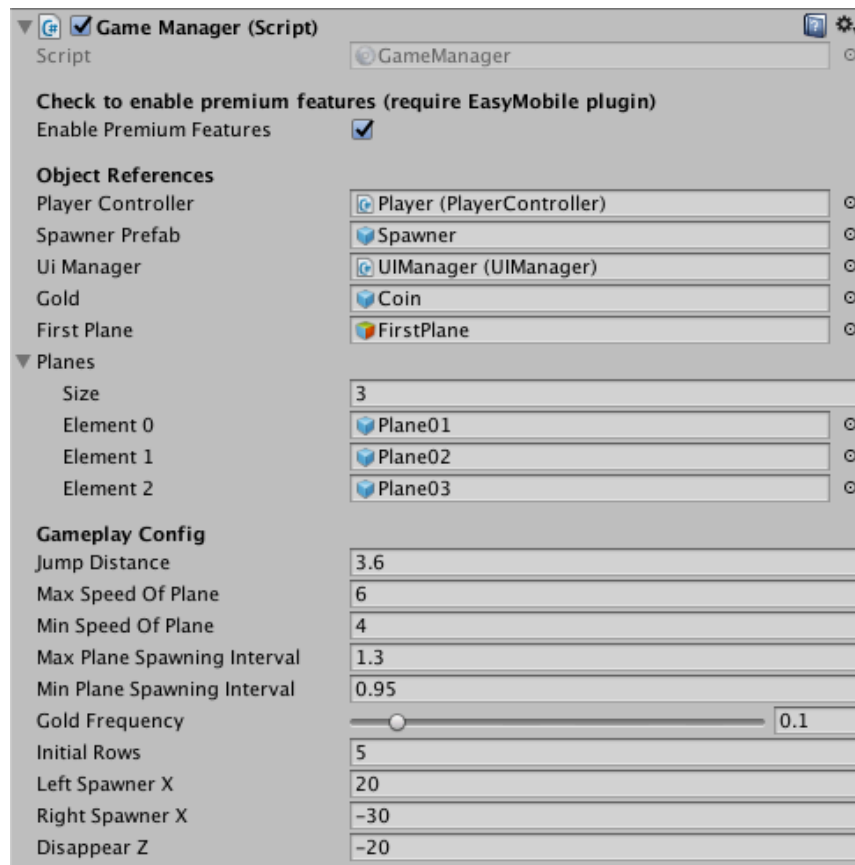
2.3 Testing Note

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

3 TEMPLATE CUSTOMIZATION

3.1 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:

- *JumpDistance*: the distance between 2 rows of planes that the character jumps through.
- *MaxSpeedOfPlane* & *MinSpeedOfPlane*: maximum & minimum moving speed of planes, the actual speeds of planes are randomized between these two values.
- *MaxPlaneSpawningInterval* & *MinPlaneSpawningInterval*: maximum & minimum time between the spawning of 2 consecutive planes on a row, these two values can be set through experiments to better sync with plane speeds, in order to have reasonable gaps between planes.
- *GoldFrequency*: the appearance probability of “gold” (or coin).

If you want to intervene deeply into the operation of the game, you can modify following variables:

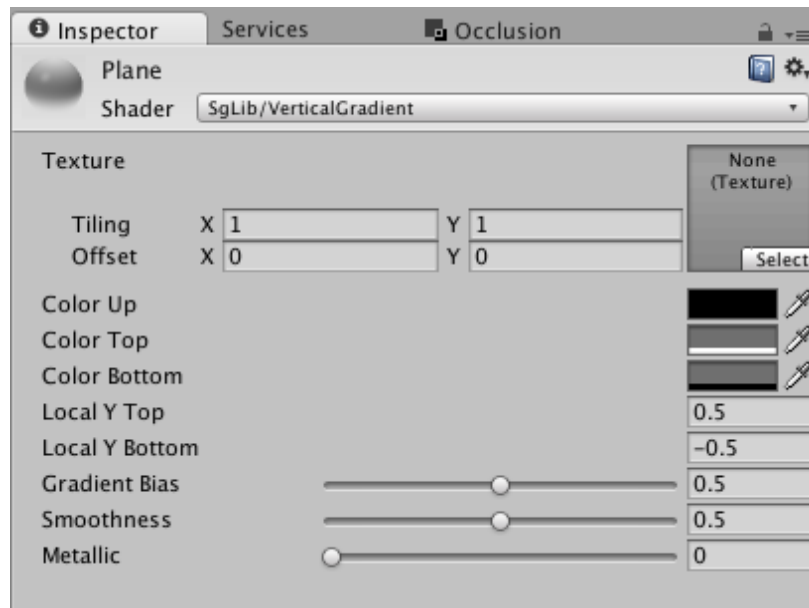
- *InitialRows*: the number of pairs of plane rows generated at startup
- *LeftSpawnerX* & *RightSpawnerX*: the left and right position of the spawner, which should be out of the visible area on the screen.
- *DisappearZ*: the z position at which a plane is not visible from the screen.

3.2 Changing plane color

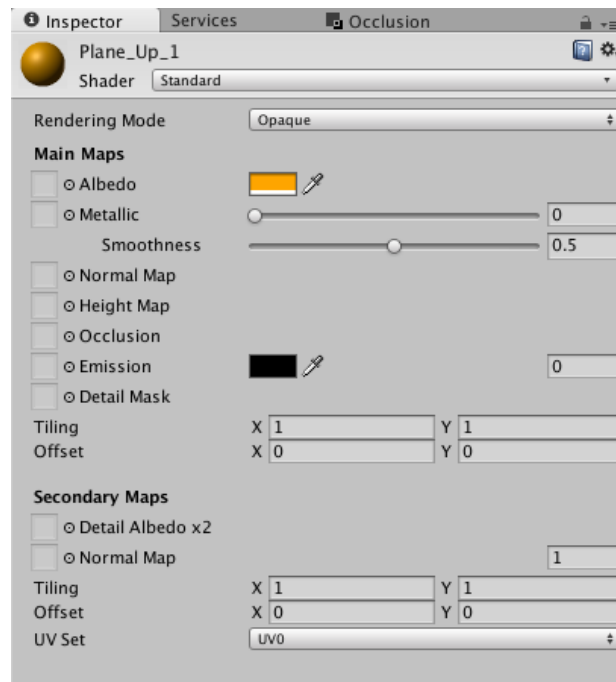
The planes (platforms) in this game have nice gradient colors with transparency to create fading effect. That is achieved thanks to our easy-to-use shader called *VerticalGradient* located in *Assets/_SkyJump/Shaders* folder. All planes use a material equipped with this shader. These materials are located in folder *Assets/_SkyJump/Materials*.

To modify the planes' gradient colors, select the *Plane* material and change the following parameters:

- *ColorUp*: the color of the upper side, you should leave this unchanged or set it to any color with alpha = 0.
- *ColorTop*: the top color of the gradient
- *ColorBottom*: the bottom color of the gradient, this one should have alpha = 0 to create fading effect.
- *LocalYTop* & *LocalYBottom*: top and bottom y position of the mesh vertices, for primitive cubes these values are 0.5 and -0.5 respectively.
- *GradientBias*: the color bias of the gradient.



The upper side colors of the planes are determined by the materials *First_Plane_Up*, *Plane_Up_1*, *Plane_Up_2* and *Plane_Up_3* corresponding to each type of plane. These materials use Standard shader, you can change the planes' upper side colors by changing the *Albedo* color of their material.



3.3 Daily reward feature

This template has a built-in daily reward system in which the user will be rewarded with coins every predefined interval of time. This is an effective way to increase user engagement and retention for your game. You can configure this feature from the *DailyRewardController* object in the hierarchy.

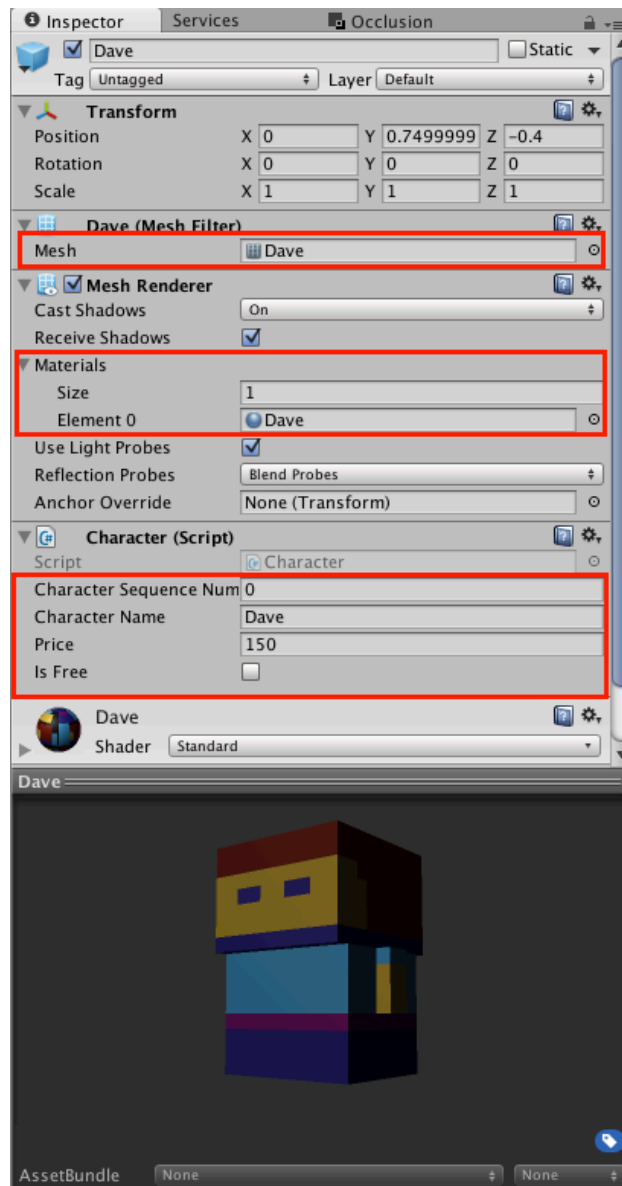


- *Disable*: check to disable this feature
- *Reward Interval Hours, Minutes and Seconds*: the amount of time until the next reward
- *Min Reward Value & Max Reward Value*: the actual rewarded coins will be randomized between these two values

3.4 Adding more characters

Out-of-the-box, this game is already packed with 30 characters, cute and ready to use! If you want to add more, follow these simple steps:

- a. Create a character model with the pivot at the bottom center.
- b. Navigate to *Assets/Prefabs/Characters/CharacterPrefabs* and duplicate one of the available character prefabs.
- c. Change the name of the prefab to a preferred one.
- d. Replace the *Mesh* in the *MeshFilter* component with your new model mesh.
- e. Replace the *Material* in the *MeshRenderer* component with your new character material.
- f. Enter the character name and price to the *Character* component. Check the *isFree* option if you want to give out this character for free (it will be automatically unlocked). **Important:** *the new character's name must not repeat any existing character name.*
- g. Resize the character array in *CharacterManager* game object then drag the new character to it and hit Apply to save changes to its prefab.



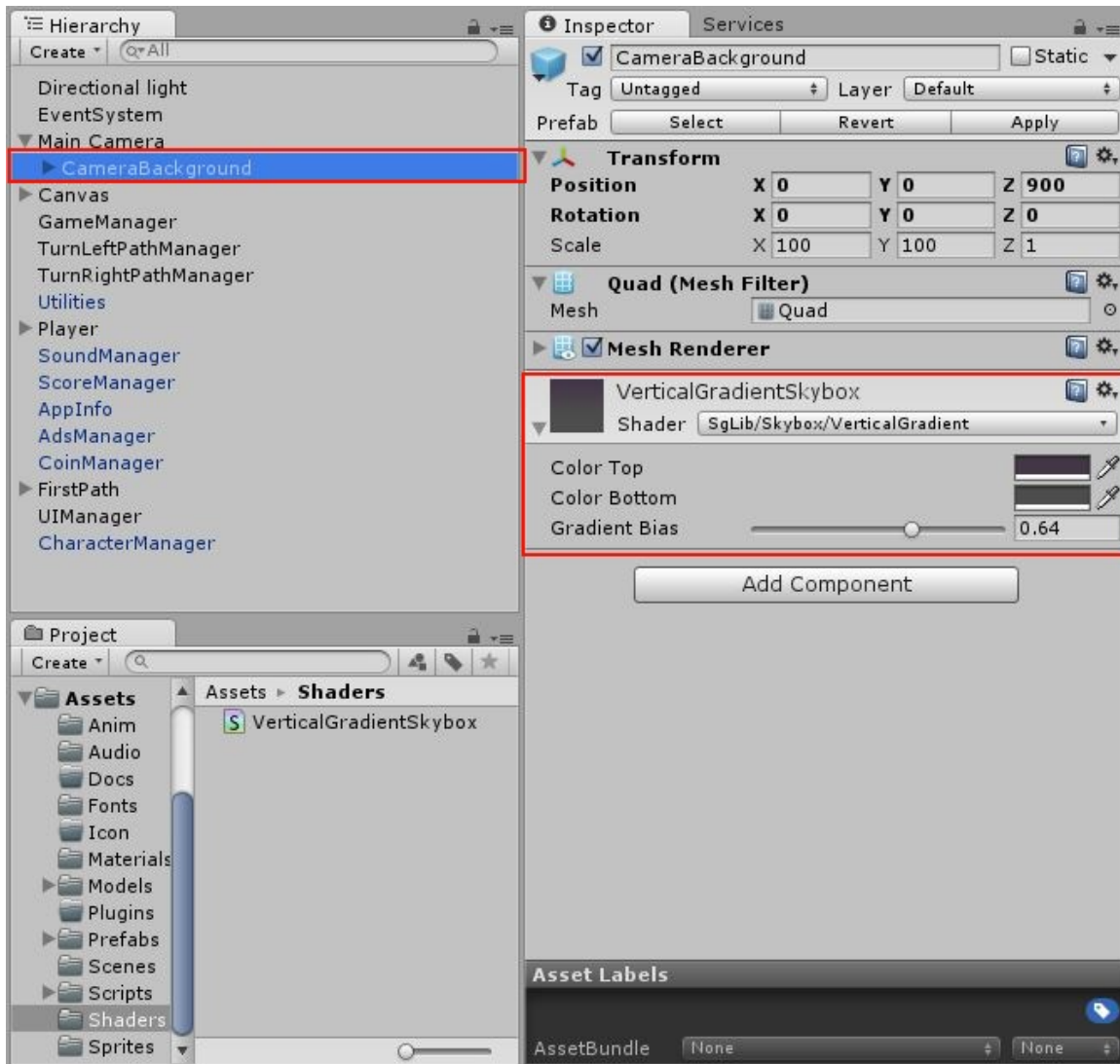
Now the new character has been added and ready to use in game! You will see it listed in the *CharacterSelection* scene.

3.5 Changing background gradient

The gradient sky background 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.

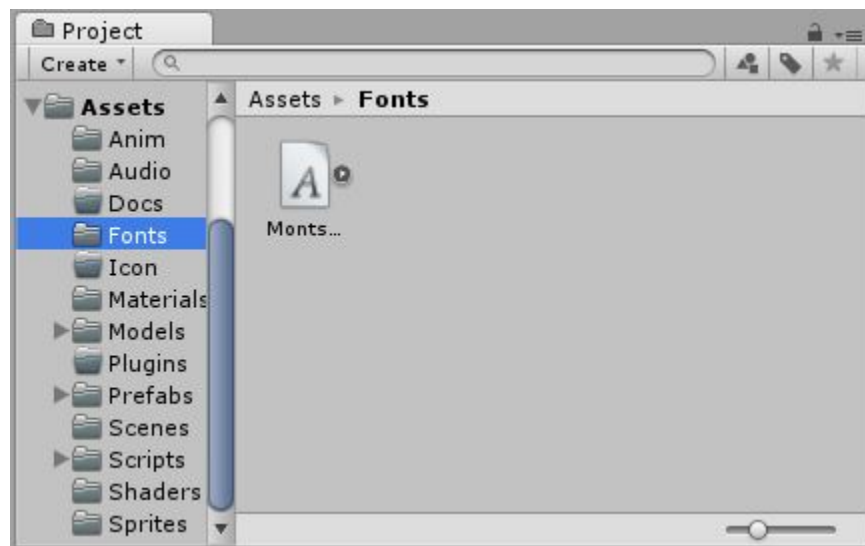


3.6 Customizing UI

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.

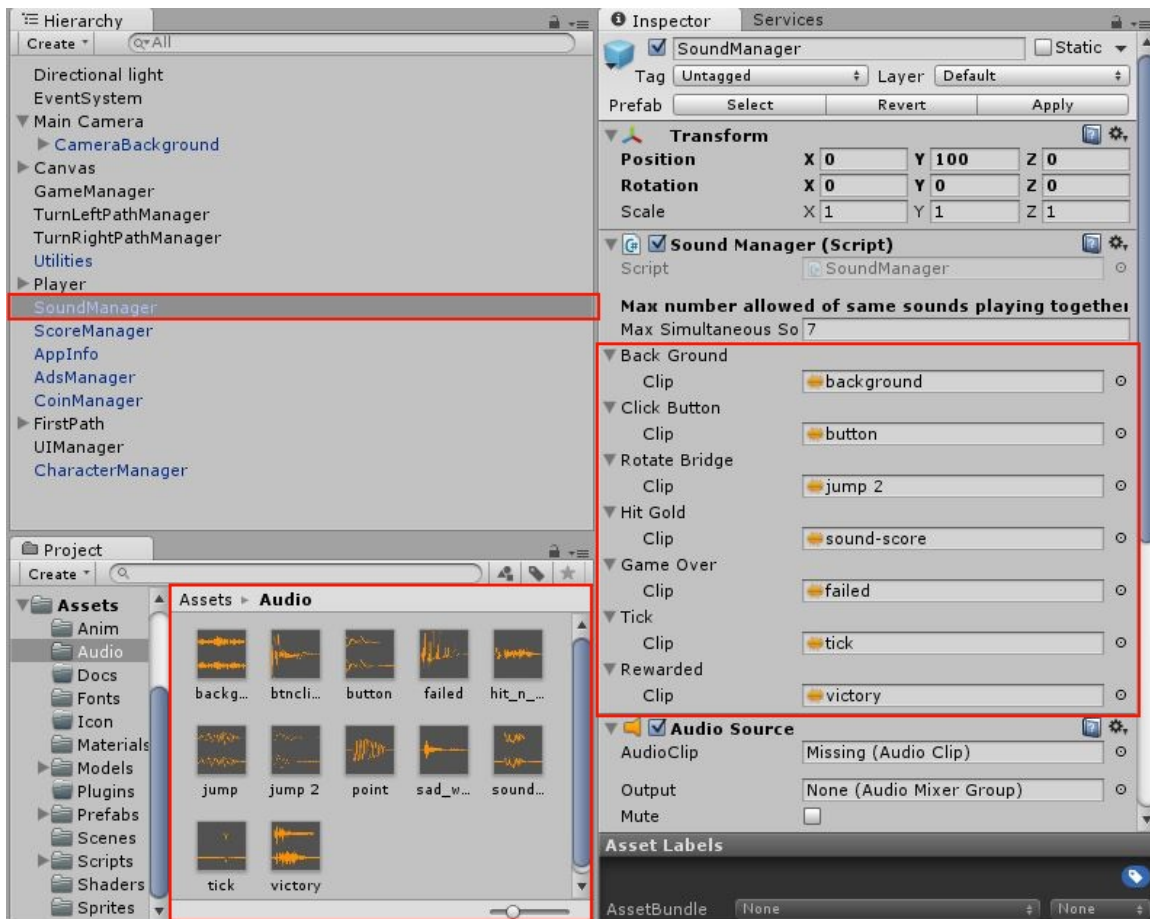


3.7 Sounds

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.



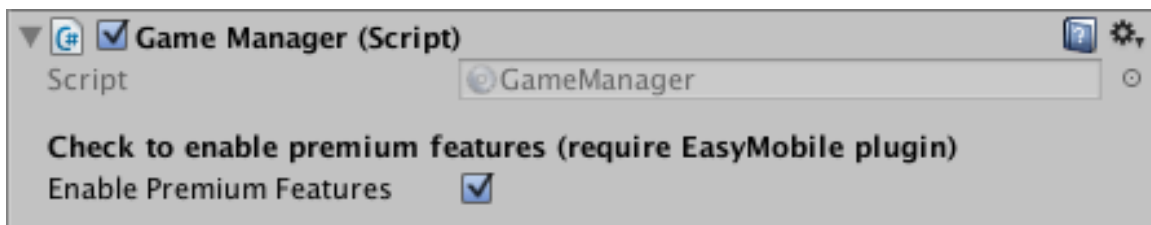
4 ENABLING PREMIUM FEATURES

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.

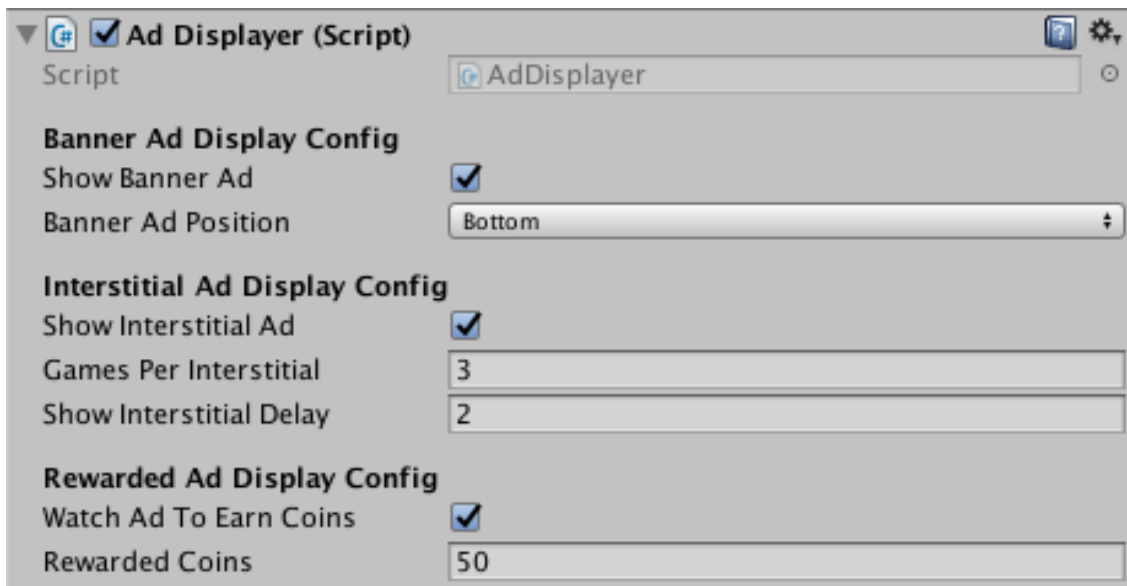


- Make sure to add the EasyMobile prefab to the Main scene, you can find the prefab at folder *Assets/EasyMobile*. It is necessary for the plugin to function properly.
- The settings interface of Easy Mobile can be opened via menu *Window > Easy Mobile > Settings*, this is the only place to go to configure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

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

- *Watch Ad To Earn Coins*: whether to allow the user to watch an ad to earn extra coins
- *Rewarded Coins*: how many coins 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 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

!

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.

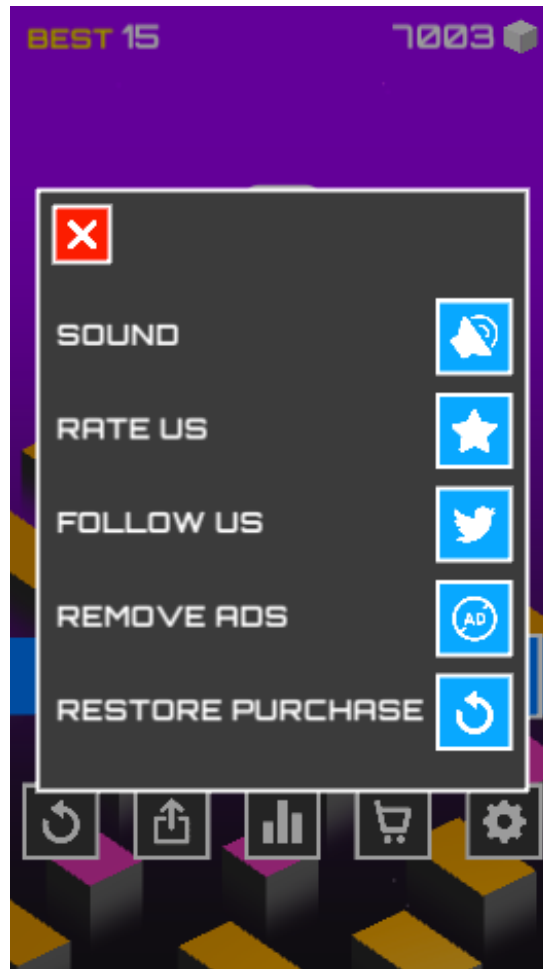
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads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

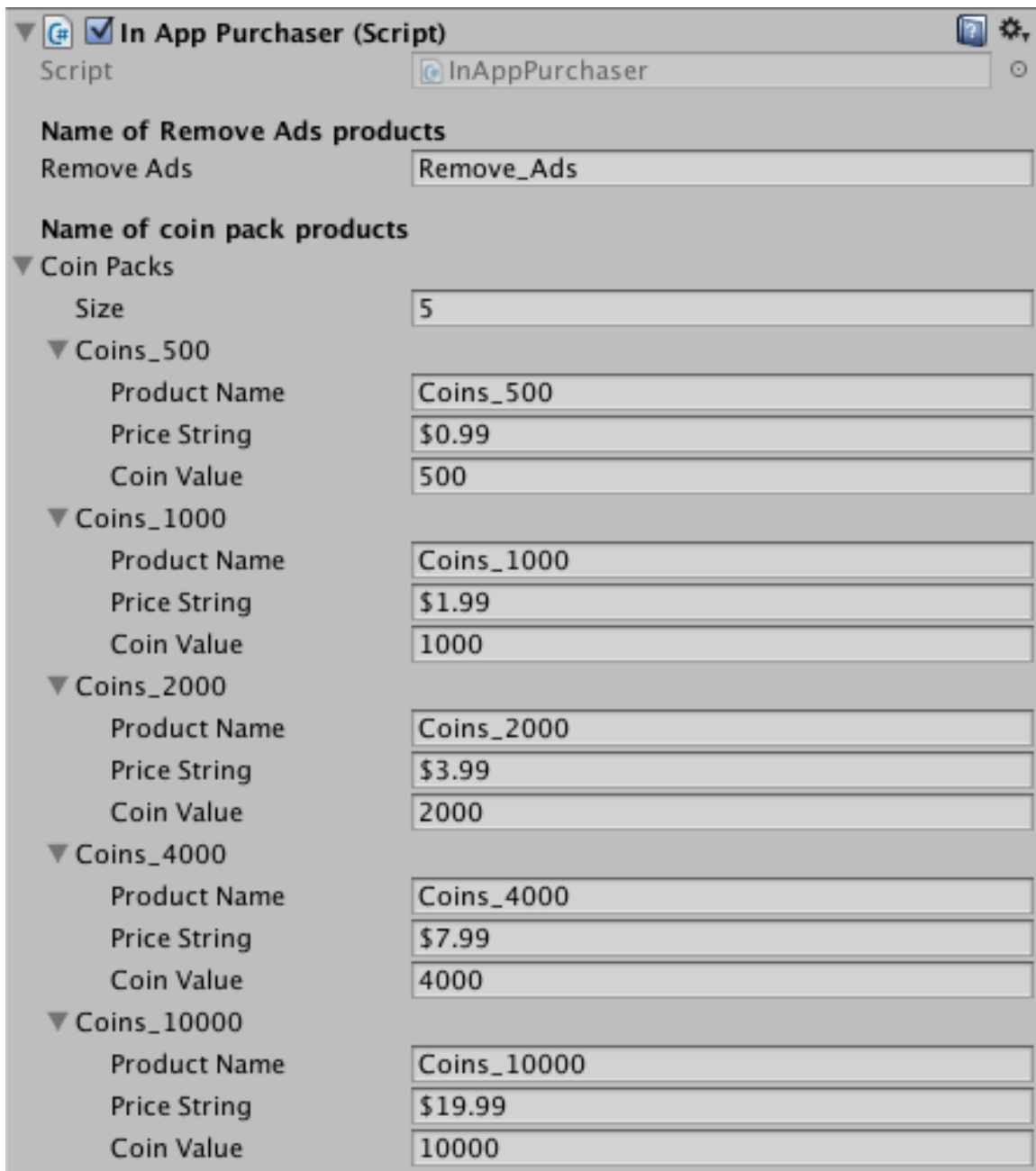
4.3 In-App Purchasing

4.3.1 Template-specific setup

The built-in in-app purchases of this template include a *Remove Ads* button, and several coin packs. You can modify existing products and add more coin packs if you like. There's also one *Restore Purchase* button as required on iOS.



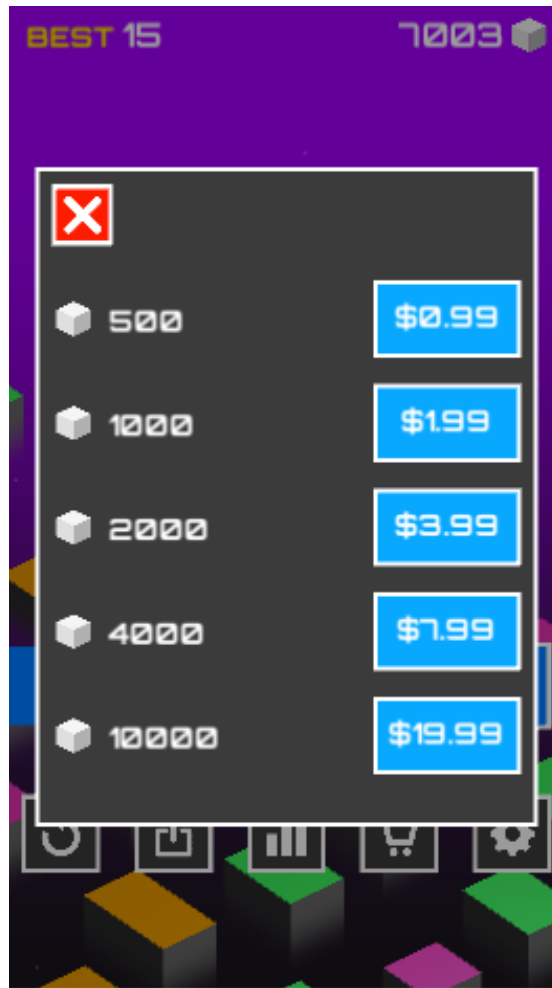
The `PremiumFeaturesManager` object contains a component named `InAppPurchaser` which manages all the in-app purchasing activities in this game.



Name of Remove Ads products	
Remove Ads	Remove_Ads

Name of coin pack products	
▼ Coin Packs	
Size	5
▼ Coins_500	
Product Name	Coins_500
Price String	\$0.99
Coin Value	500
▼ Coins_1000	
Product Name	Coins_1000
Price String	\$1.99
Coin Value	1000
▼ Coins_2000	
Product Name	Coins_2000
Price String	\$3.99
Coin Value	2000
▼ Coins_4000	
Product Name	Coins_4000
Price String	\$7.99
Coin Value	4000
▼ Coins_10000	
Product Name	Coins_10000
Price String	\$19.99
Coin Value	10000

Here you can modify the product definitions including the displayed name, price or coin value of the coin packs. To add more coin packs, simply increase the *CoinPacks* array size and enter necessary information for your new packs. The built-in store UI will automatically update to your changes in the product list without you having to do anything.



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.

- Enable Unity In-App Purchasing service
- Select target store if you're on Android
- Enable receipt validation if you wish
- 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

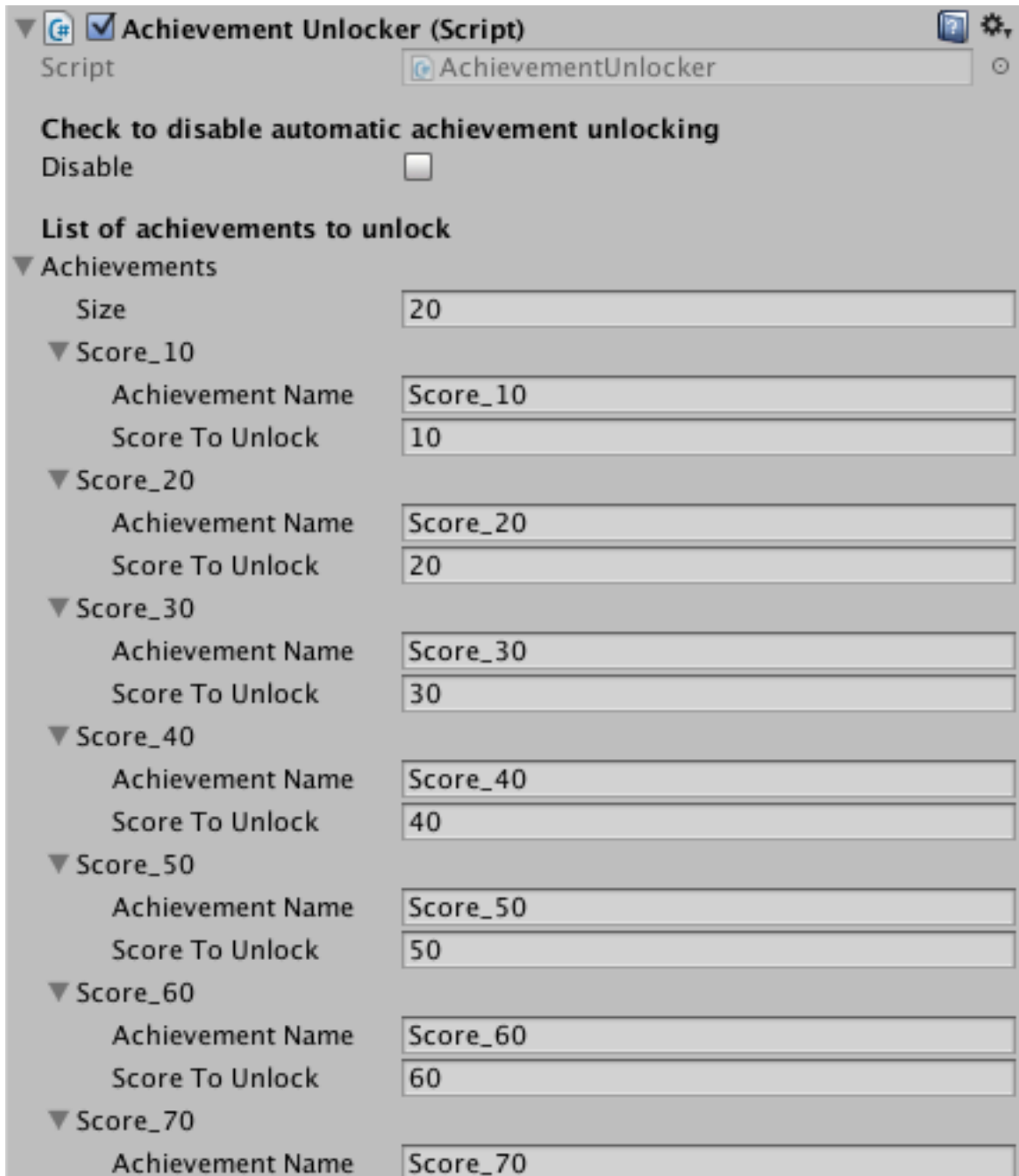
This template has a built-in leaderboard for ranking users' scores, and many achievements. It works with Game Center (iOS) and Google Play Game Services (Android).

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 titled 'Score Reporter (Script)'. 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 labeled 'Name of the leaderboard to report score as declared with EasyMobile' with the value '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: 20

▼ 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[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>
```

[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

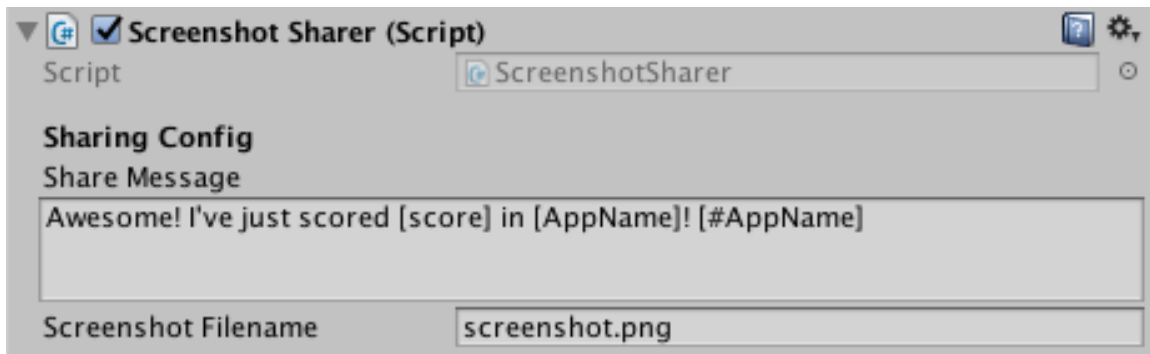
▶ 20 Achievements

[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

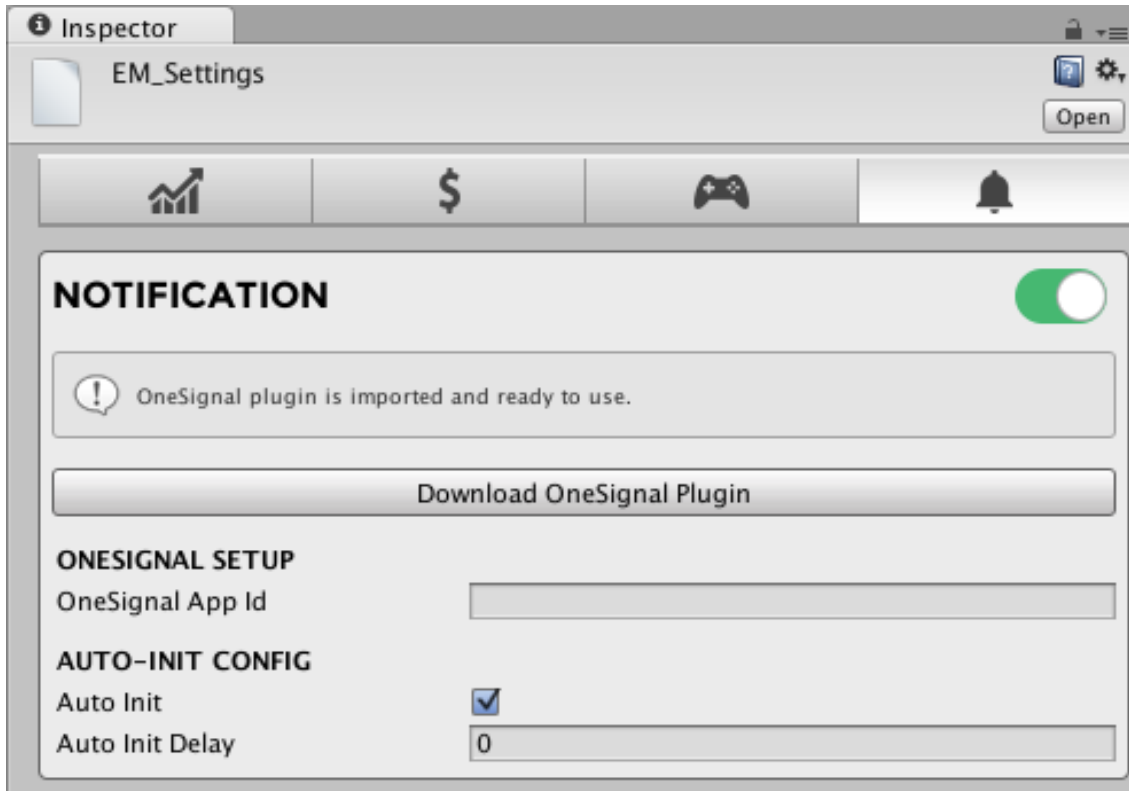
Note that you need to enable the *external write permission* for this feature to function properly on Android. Please see the Native Sharing section in Easy Mobile user guide for detailed instructions on doing that.

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!