



SLIDING SANTA 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
3.1.1	<i>GameManager</i>	6
3.1.2	<i>PlayerController</i>	7
3.2	ADDING MORE CHARACTERS	8
3.3	ENVIRONMENT MANAGER	10
3.4	DAILY REWARD FEATURE	12
3.5	CUSTOMIZING UI	13
3.6	SOUNDS	14
4	ENABLING PREMIUM FEATURES	15
4.1	BEFORE YOU BEGIN	15
4.2	ADVERTISING	16
4.2.1	<i>Template-specific setup</i>	16
4.2.2	<i>Easy Mobile setup</i>	17
4.3	IN-APP PURCHASING	19
4.3.1	<i>Template-specific setup</i>	19
4.3.2	<i>Easy Mobile setup</i>	21
4.3.3	<i>Create the products for targeted stores</i>	23
4.4	GAME SERVICE	23
4.4.1	<i>Template-specific setup</i>	23
4.4.2	<i>Setup for your targeted stores</i>	25
4.4.3	<i>Easy Mobile setup</i>	26
4.5	NATIVE SHARING	27
4.6	RATING REQUEST	28
4.7	PUSH NOTIFICATION	29

1 INTRODUCTION



Help Mr. Santa slide through a slippery, zig-zag, snowy path amid a beautiful pine forest to collect presents for all the children this Christmas! Tap and hold to turn left, release to turn right. Avoid the obstacles and collect the gifts. It couldn't be simpler! Seriously, **Sliding Santa** is the perfect game for this Christmas season, but it's not only that: use gifts to unlock dozens of cool new vehicles and characters, of whom some special ones even change the environment to a completely different world!

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

- a. Perfect game for Christmas with simple and addictive one-touch gameplay
- b. Daily reward system for better retention
- c. Lots of built-in vehicles characters with cute voxel style, ready to be unlocked using in-game currency
- d. Free-to-use assets (fonts, sounds, music, model, etc.)
- e. Optimized for iOS and Android

Most importantly, this template is compatible 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: buy coins and remove ads
- Support for Game Center (iOS) and Google Play Games Services (Android) for leaderboards and achievements

- Recording gameplay and exporting GIF images for sharing
- Sharing GIF or PNG images to social networks
- Push notification using OneSignal service
- Native rating prompt (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.



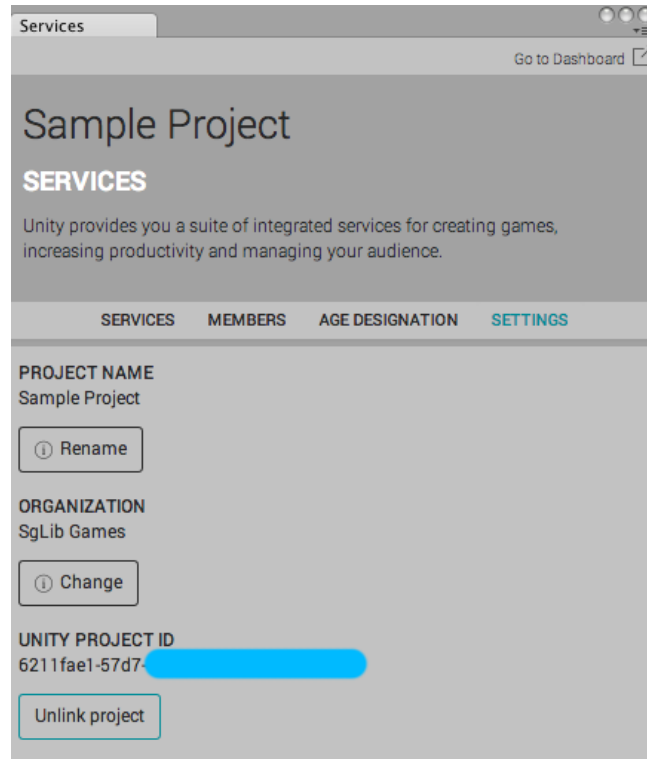
The screenshot shows the Unity Inspector window for a game object named 'AppInfo'. The script 'App Info (Script)' is selected, and the following fields are visible:

Field Name	Value
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]

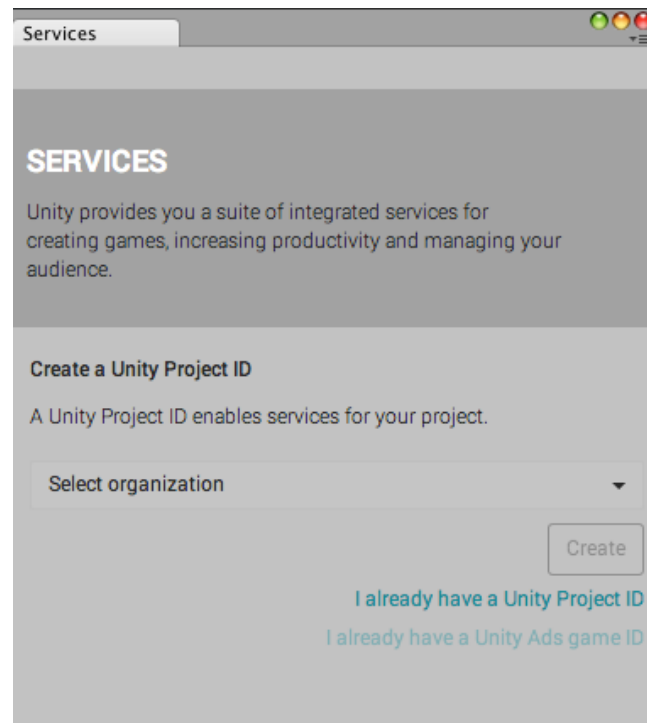
2.2 Link the game to your Unity project

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

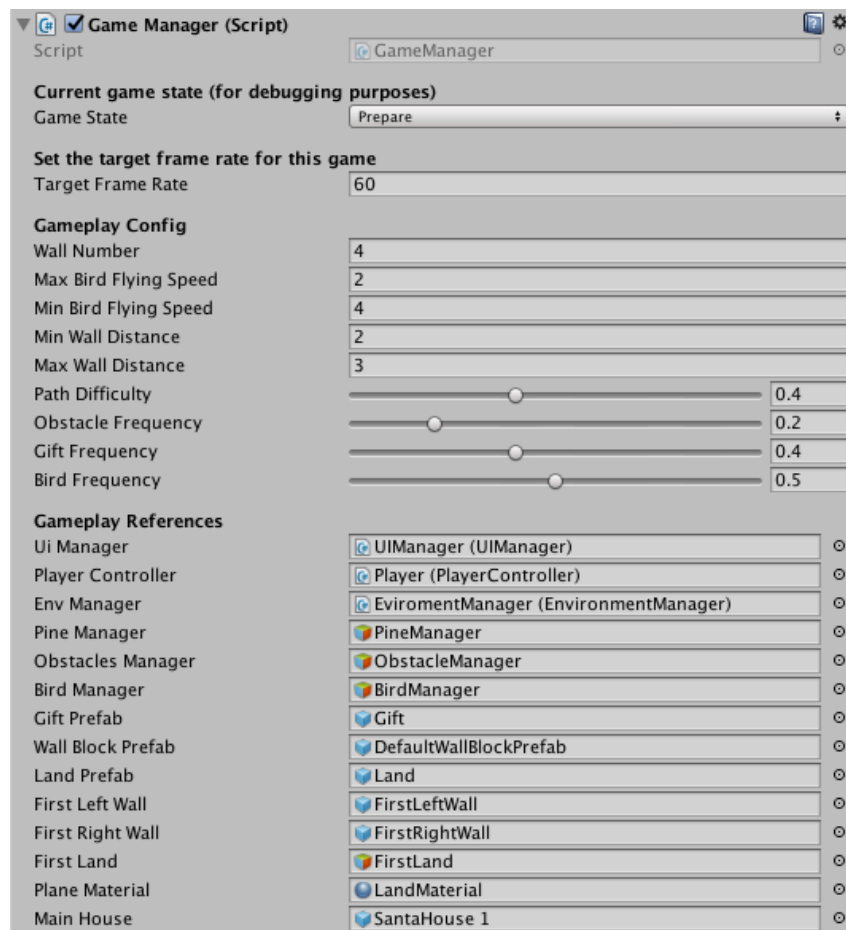
services.

3 TEMPLATE CUSTOMIZATION

3.1 Gameplay tweaking

3.1.1 *GameManager*

Most of important game parameters are located in **GameManager** object in the hierarchy



You can tweak the gameplay by modifying the variables within the **Gameplay Config** section:

WallNumber: how many wall (left and right blocks) to be created when the game starts.

MaxBirdFlyingSpeed: max speed of the birds.

MinBirdFlyingSpeed: min speed of the birds.

MinWallDistance: min distance between 2 walls.

MaxWallDistance: max distance between 2 walls.

PathDifficulty: the larger values will generate harder paths.

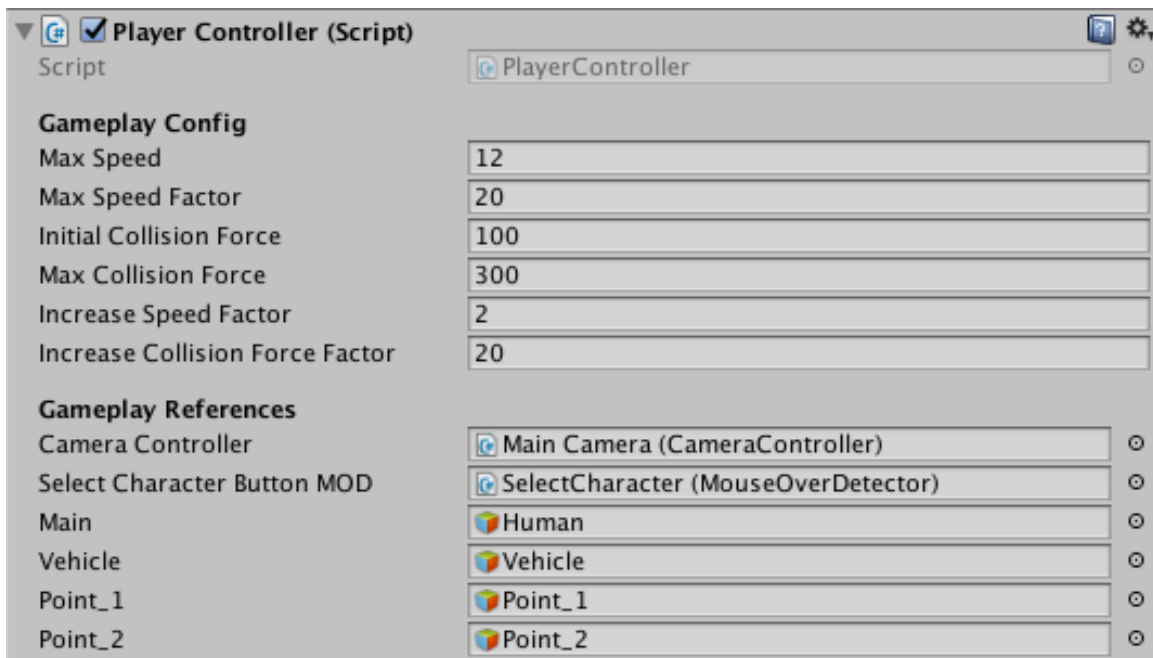
ObstacleFrequency: the larger values will generate more obstacles.

GiftFrequency: the larger values will generate more gift boxes.

BirdFrequency: the larger values will generate more birds.

3.1.2 PlayerController

You can tweak the player's behavior from the **PlayerController** component of the **Player** object in the hierarchy.



MaxSpeed: how fast player turn right (max value)

MaxSpeedFactor: how fast player turn left

InitialCollisionForce: the collision force when the game starts

MaxCollisionForce: max value of collision force

IncreaseSpeedFactor: how fast player reaches max speed

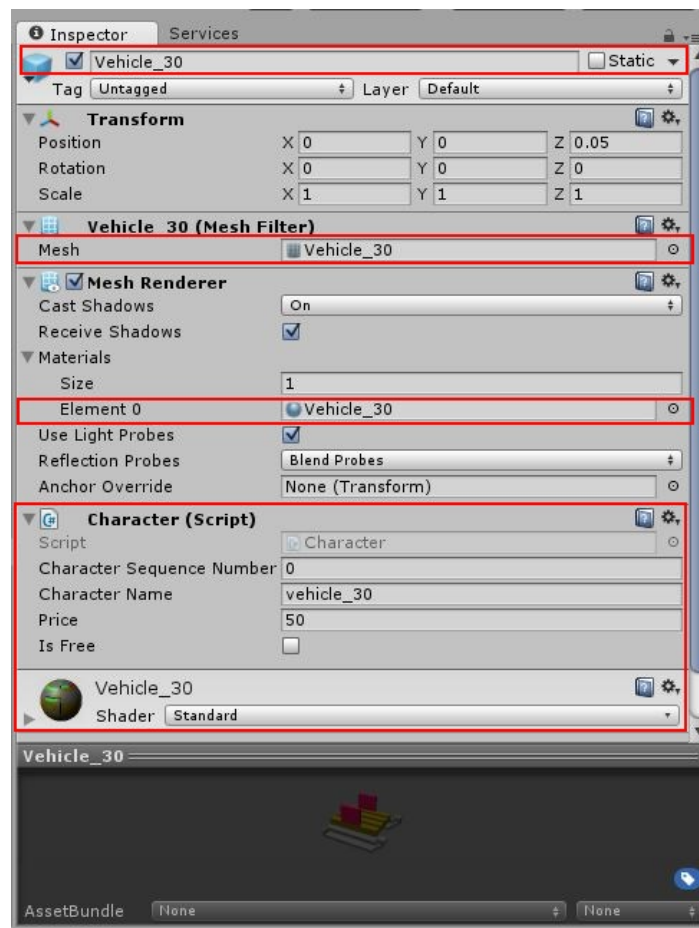
IncreaseCollisionForceFactor: how fast player reaches max collision force

3.2 Adding more characters

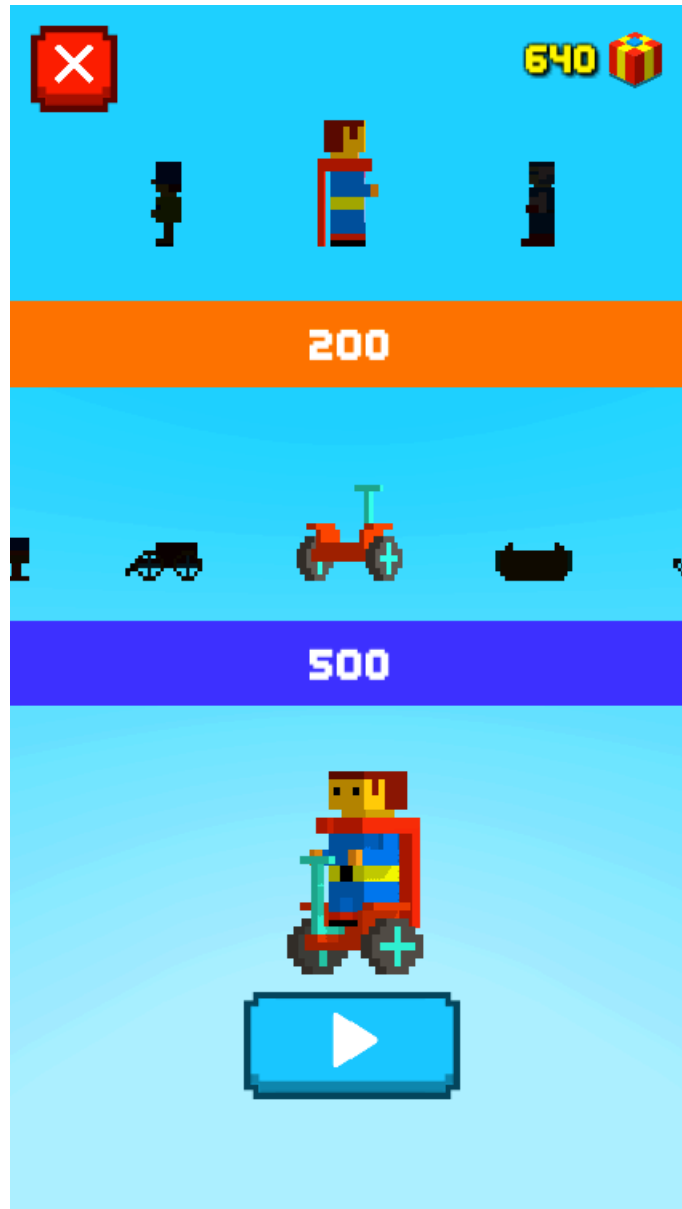
Out-of-the-box, this game is already packed with dozens of characters and vehicles, cute and ready to use!

Adding vehicles and characters are similar, below are the steps to add a new vehicle:

- Create a vehicle model.
- Navigate to `_SlidingSanta/Prefabs/Characters/VehiclePrefabs` and duplicate one of the available vehicle prefabs.
- Change the name of the prefab to a preferred one.
- Replace the Mesh in the MeshFilter component with your new model mesh.
- Replace the Material in the MeshRenderer component with your new character material.
- Enter the vehicle name and price to the Character component. Check the *isFree* box if you want to give out this vehicle for free (it will be automatically unlocked).
- Resize the array in the **VehicleManager** game object then drag the new character to it and hit Apply to save changes to its prefab.



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



You can add more characters similarly by creating new characters in the `_SlidingSanta/Prefabs/Characters/HumanPrefabs` folder and add it to the **HumanManager** game object.

***** IMPORTANT NOTE:** *new character/vehicles must have a different name than all existing ones.*

3.3 Environment Manager

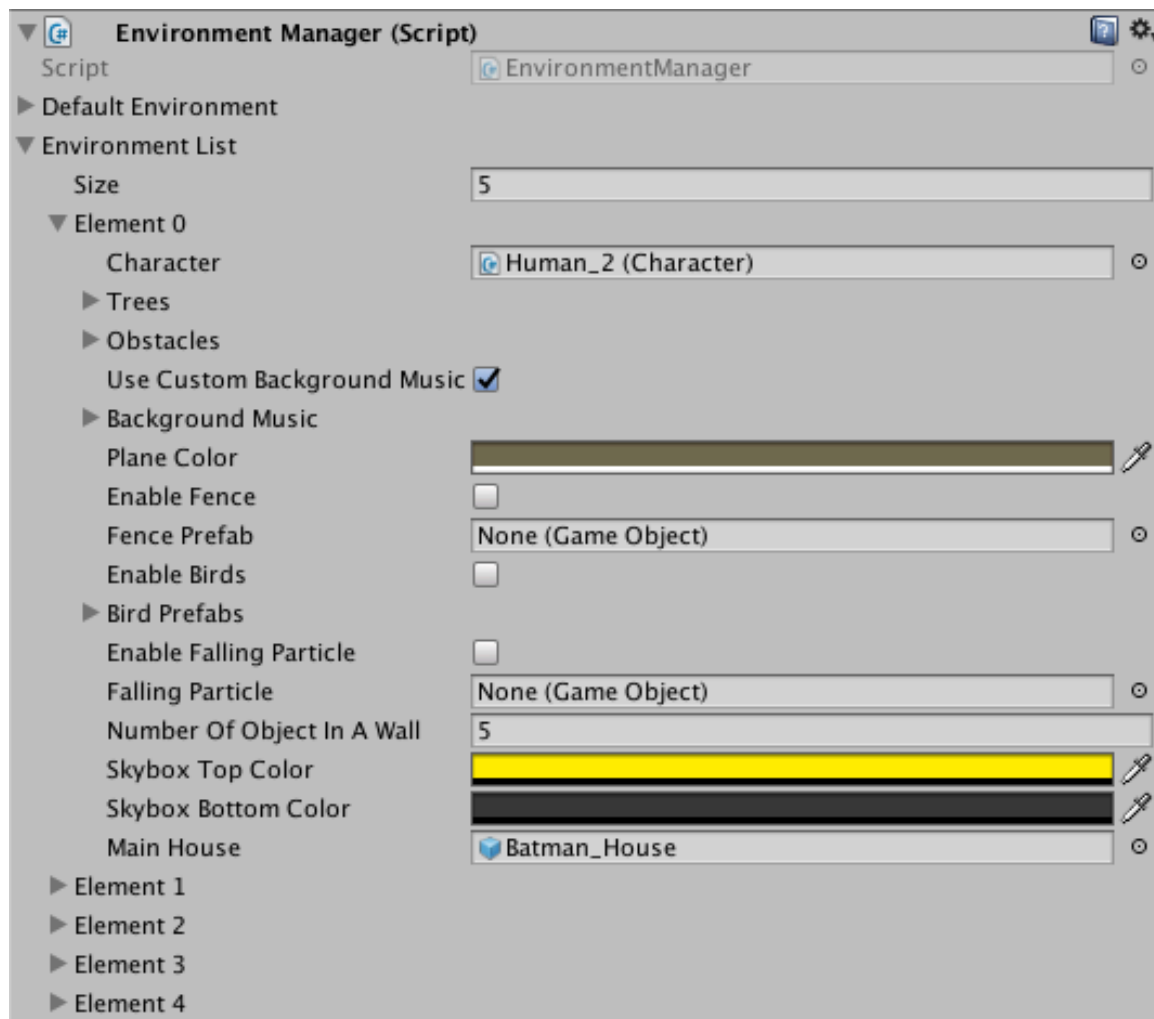
Some built-in characters in this template has a special effect that they switch the whole in-game world to a completely new environment! You can give your new

character that special effect too!

To add new environment for your character you will need several prefabs to construct the in-game world: the player's house, the trees, the obstacles, the fences, the falling particles and even the birds if you want. So you should prepare your own models and prefabs for these elements.

The folder `_SlidingSanta/Prefabs/Game/SpecialObject` contains the world-element prefabs for the built-in special characters. You can duplicate one of its subfolders and update the contained prefabs with your new models (replace the current mesh in the Mesh Filter component with your model).

Then locate the **EnvironmentManager** object in the hierarchy.

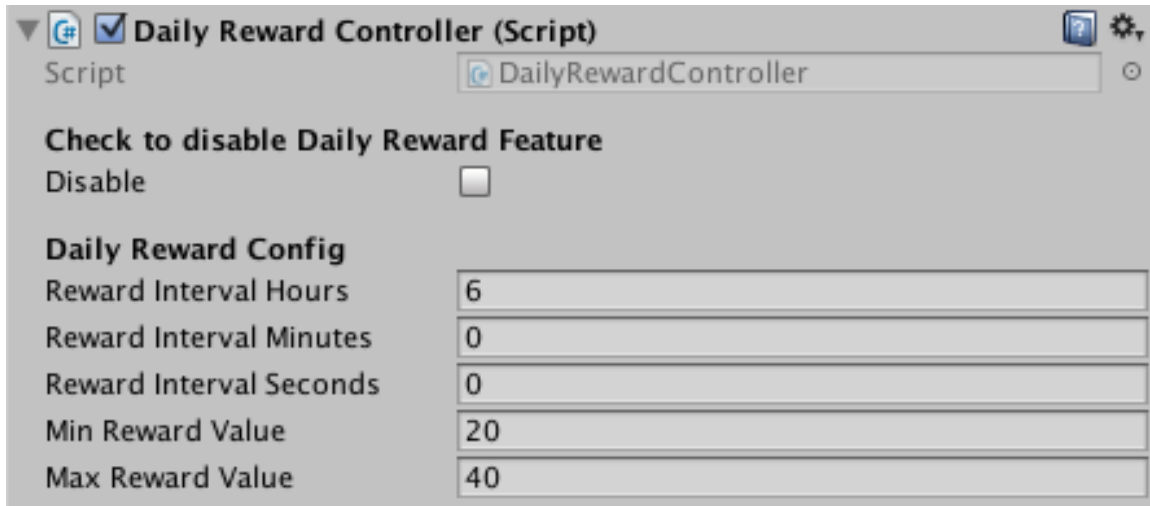


Increase the size of the *EnvironmentList* array to accommodate your new environment settings.

- *Character*: The character you want to have a special environment. Drag your character prefab to this field.
- *Trees*: your new tree prefabs
- *Obstacles*: your new obstacle prefabs
- *UseCustomBackgroundMusic* : check this to use a custom background song for your character
- *BackgroundMusic*: drag the custom music file to this field if the above option is checked
- *PlaneColor*: the ground color of this environment
- *EnableFence*: check this if you want to have fences along the path
- *FencePrefab*: drag your custom fence prefab here if you ticked the above box
- *EnableBirds*: check this if you want to have birds flying in this world
- *BirdPrefabs*: drag your custom bird prefabs here if you ticked the above box
- *EnableFallingParticle*: check this to have particles falling in this world
- *FallingParticle*: drag your custom falling particle prefab to this field if you checked the above box
- *NumberOfObjectInAWall*: the number of object generated inside a block (that forms left and right walls)
- *SkyboxTopColor* & *SkyBoxBottomColor*: the top & bottom color of the gradient skybox used in the character selection screen when the current special character is in focus
- *MainHouse*: the house where the character starts

3.4 Daily reward feature

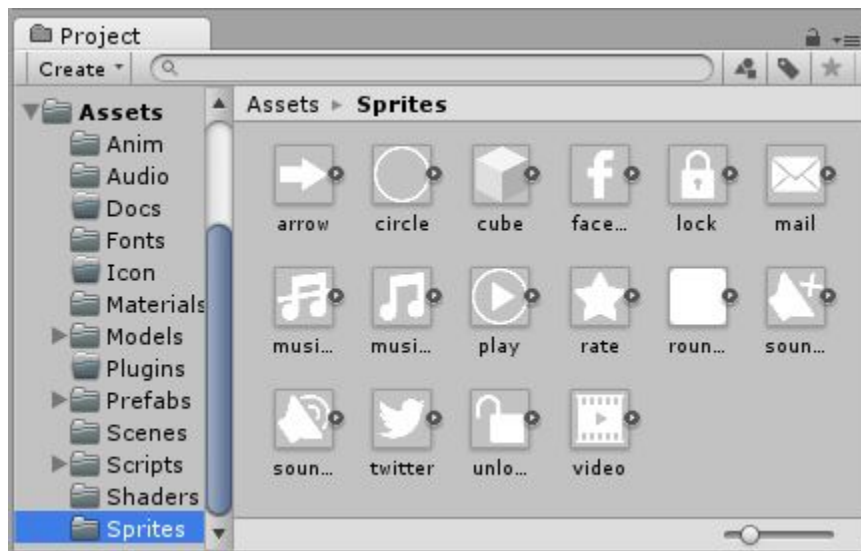
This template has a built-in daily reward system in which the user will be rewarded with coins every predefined interval of time. This is an effective way to increase user engagement and retention for your game. You can configure this feature from the *DailyRewardController* object in the hierarchy.



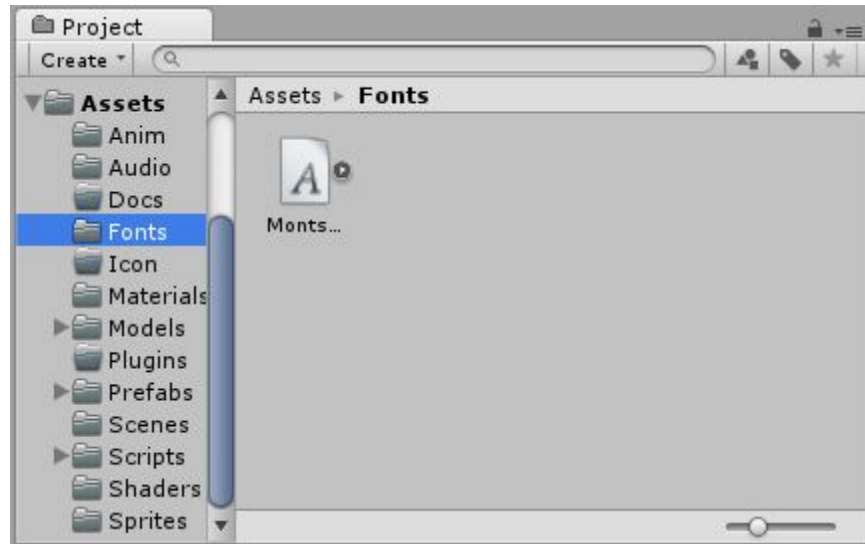
- *Disable*: check to disable this feature
- *Reward Interval Hours, Minutes and Seconds*: the amount of time until the next reward
- *Min Reward Value & Max Reward Value*: the actual rewarded coins will be randomized between these two values

3.5 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.

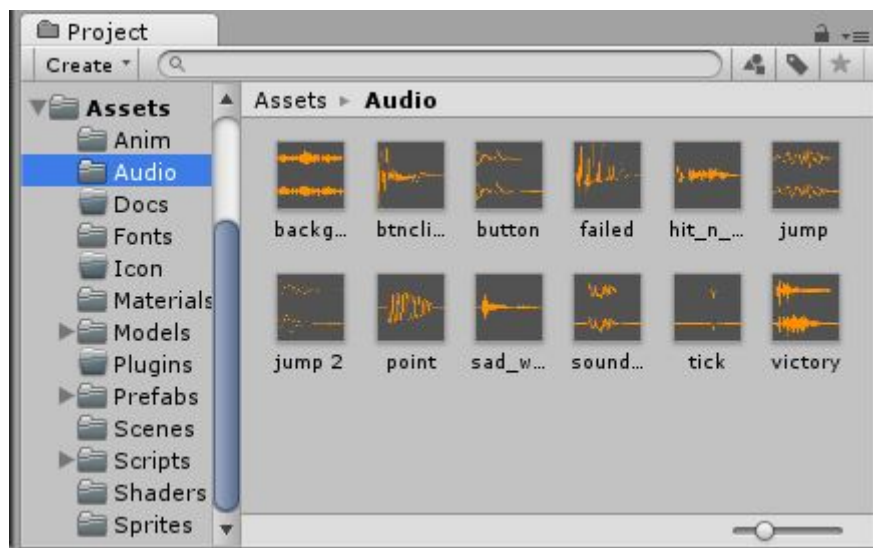


All fonts used in this game are free-to-use in commercial projects. Fonts are located under the *Fonts* folder together with appropriate license files.

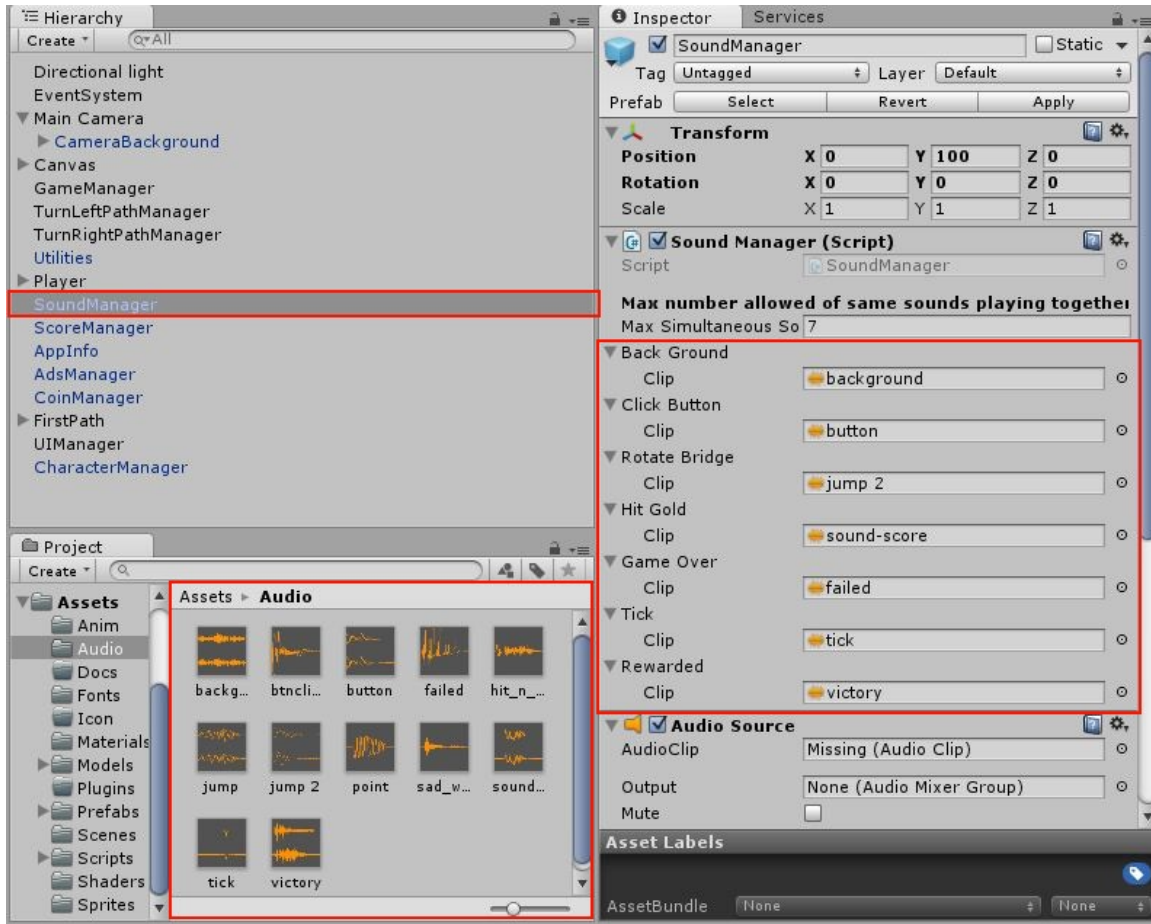


3.6 Sounds

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.

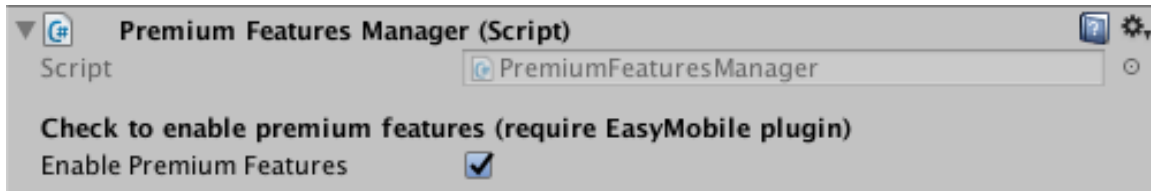


4 ENABLING PREMIUM FEATURES

This section provides a guide on configuring premium features for your game. As these features are implemented using [EasyMobile](#) plugin, this guide is intended to be used alongside EasyMobile's official user guide. The PDF user guide can be found inside the Assets/EasyMobile folder. Its online version can be accessed from menu *Window > Easy Mobile > Online Documentation*.

4.1 Before You Begin

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

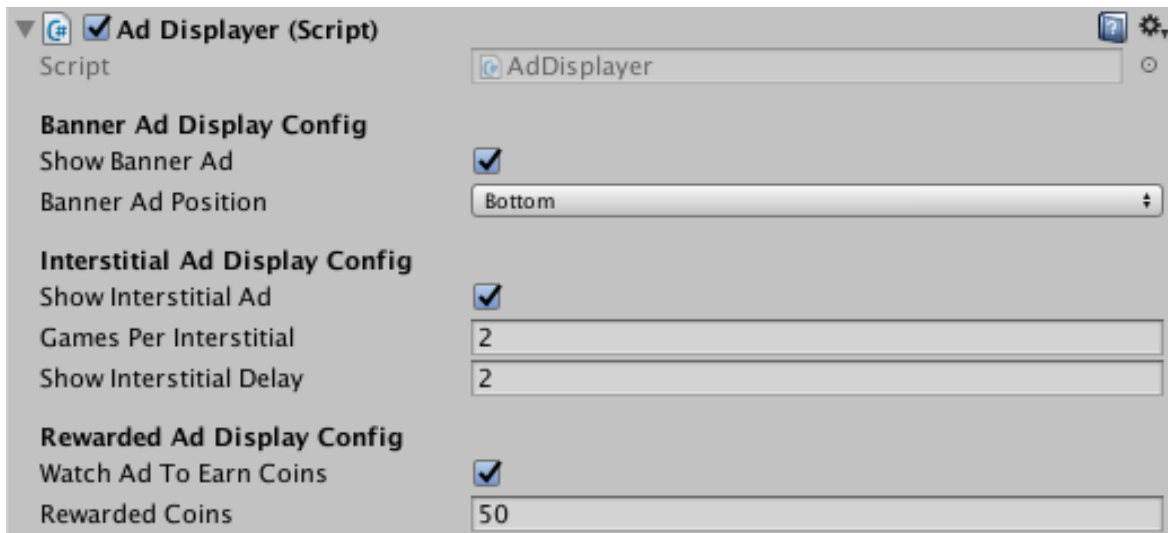


- Make sure to add the EasyMobile prefab to the Main scene, you can find the prefab at folder *Assets/EasyMobile*. It is necessary for the plugin to function properly.
- The settings interface of Easy Mobile can be opened via menu *Window > Easy Mobile > Settings*, this is the only place to go to configure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

4.2 Advertising

4.2.1 Template-specific setup

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



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

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

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

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

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

- *Watch Ad To Earn 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 AdColony, AdMob, Chartboost, Heyzap (with mediation) and Unity Ads. You can use multiple ad networks at once and have different configurations for iOS and Android. Below is the settings interface of the Advertising module.

ADVERTISING

ADMOB SETUP

Google Mobile Ads (AdMob) plugin was imported.

Reimport Google Mobile Ads Plugin

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

CHARTBOOST SETUP

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

Download Chartboost Plugin

HEYZAP SETUP

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

Download Heyzap Plugin

UNITY ADS SETUP

Unity Ads service is enabled.

AUTO AD-LOADING CONFIG

Auto-Load Default Ads ☒

Ad Checking Interval 10

Ad Loading Interval 20

DEFAULT AD NETWORKS

▼ [iOS] Default Ad Networks

Banner Ad Network Ad Mob

Interstitial Ad Network Ad Mob

Rewarded Ad Network Unity Ads

▼ [Android] Default Ad Networks

Banner Ad Network Ad Mob

Interstitial Ad Network Ad Mob

Rewarded Ad Network Unity Ads

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

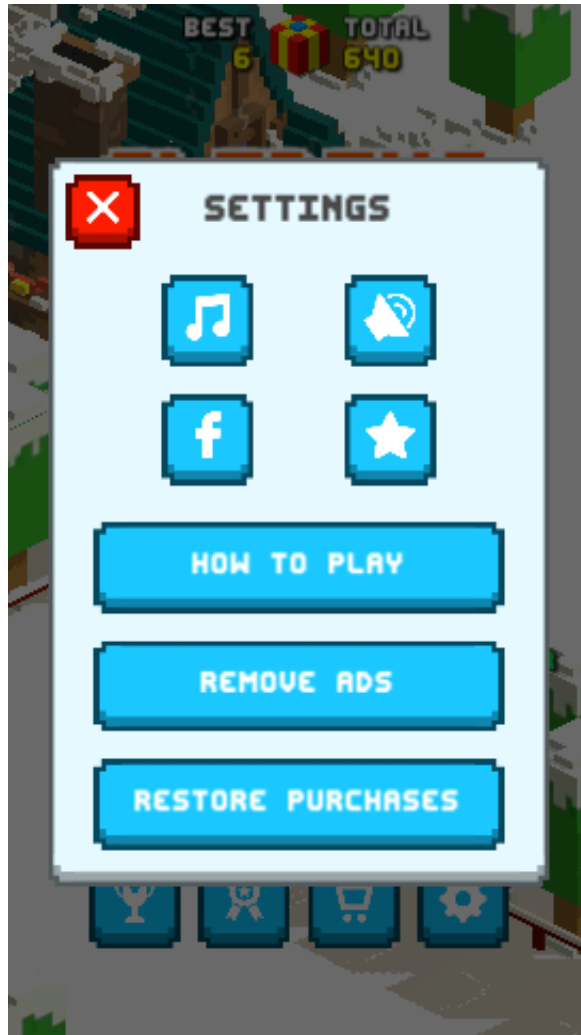
- Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
- Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load ads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

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

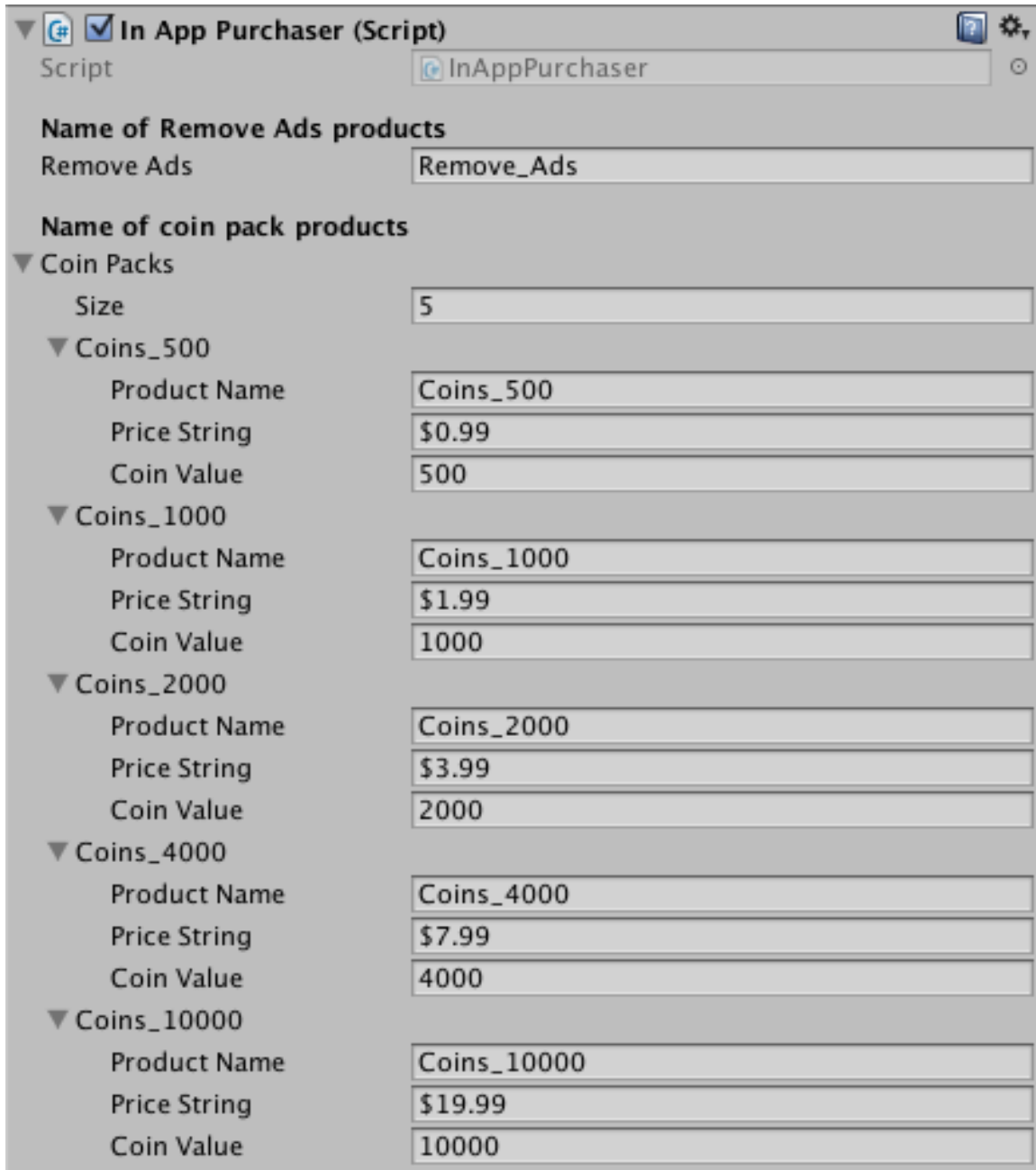
4.3 In-App Purchasing

4.3.1 Template-specific setup

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



In App Purchaser (Script)

Script: InAppPurchaser

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

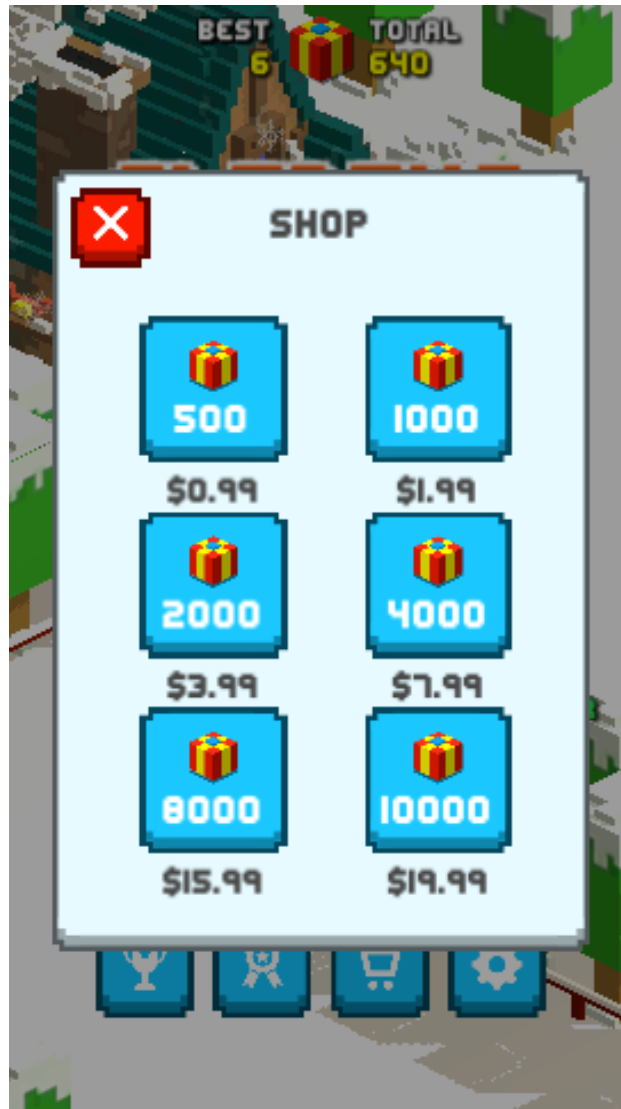
▼ Coins_10000

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 displays a configuration interface for in-app purchases. It features two main sections, each with a title, input fields for Name, Type, and Id, and a 'More (Optional)' link. To the right of each section are three vertically stacked buttons: an up arrow, a minus sign, and a down arrow.

Remove_Ads

Name: Remove_Ads

Type: Non Consumable

Id: sglib.demogame.iap.remove_ads

► More (Optional)

Coins_500

Name: Coins_500

Type: Consumable

Id: sglib.demogame.iap.coins_500

► More (Optional)

4.3.3 Create the products for targeted stores

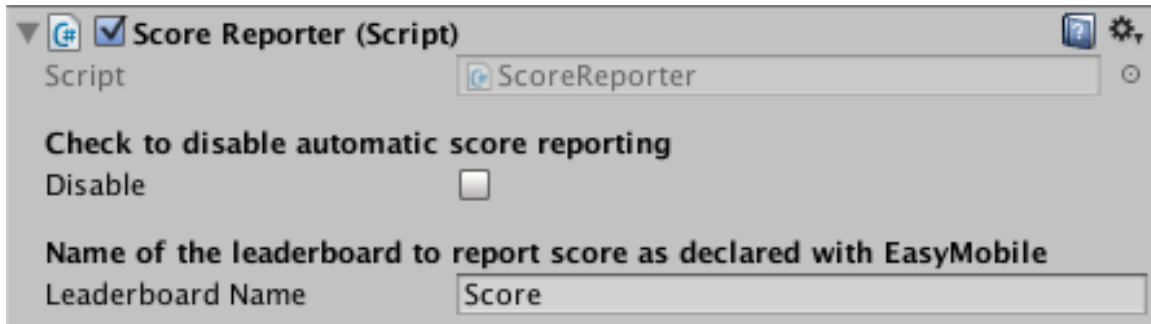
That last step in configuring the in-app purchasing feature is to create products for your targeted stores (e.g. Google Play and Apple App Store). Make sure the product ID, product type and price match the ones you have in your game.

4.4 Game Service

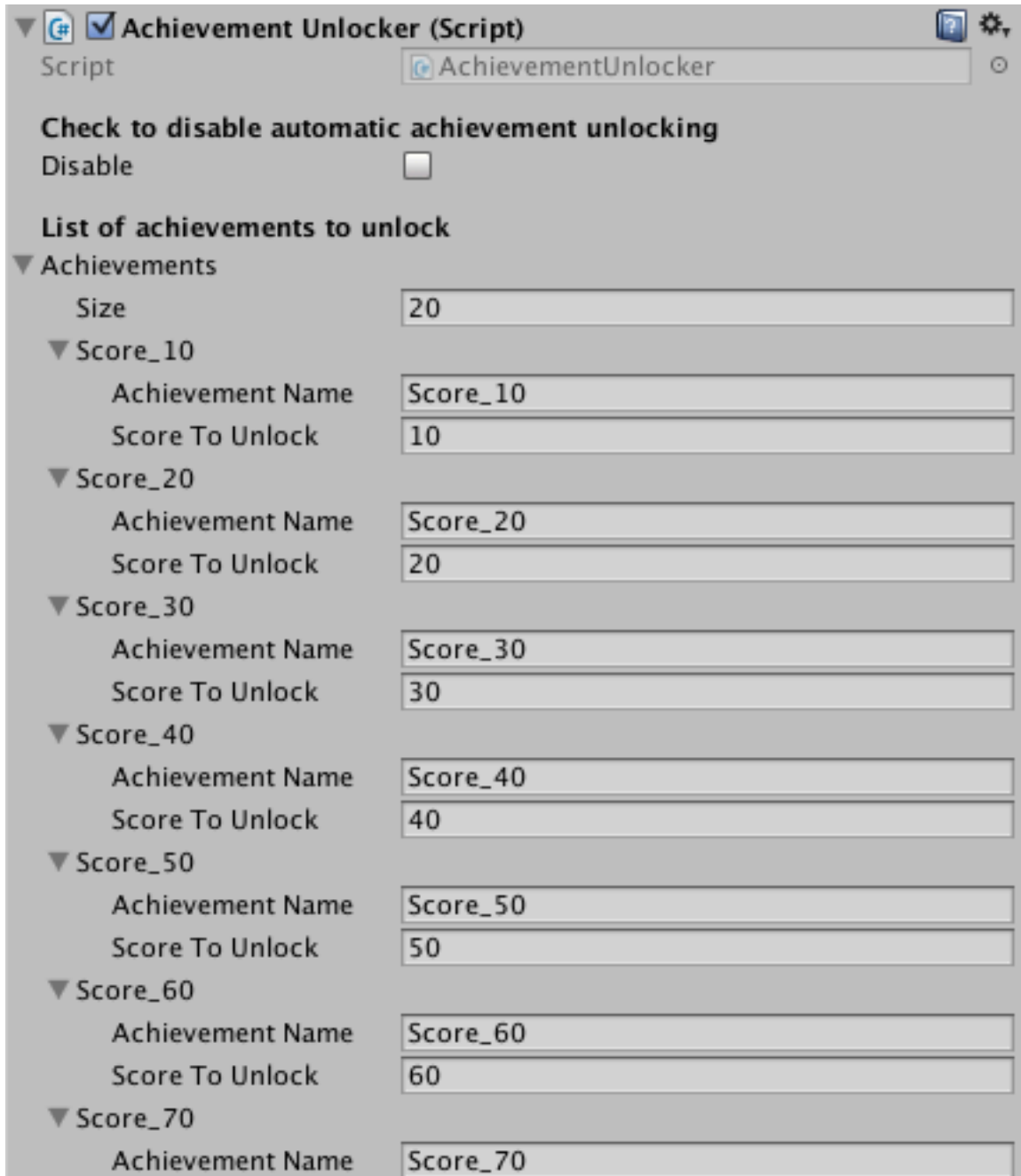
4.4.1 Template-specific setup




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.



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.

The screenshot shows the 'GAME SERVICE' settings panel. At the top, there's a toggle switch for 'GAME SERVICE' which is turned on. Below it, a message box states 'Google Play Games plugin is imported and ready to use.' with a 'Reimport Google Play Games Plugin' button. The next section is '[ANDROID] GOOGLE PLAY GAMES SETUP', which includes a 'GPGS Debug Log' checkbox (unchecked) and a text box for pasting Android XML Resources. Below this is a text area containing XML code for game services IDs. A 'Setup Google Play Games' button is located below the XML text area. The 'AUTO-INIT CONFIG' section has 'Auto Init' checked, 'Auto Init Delay' set to 0, and '[Android] Max Login Request' 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.

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">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 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 settings panel for a script named "Screenshot Sharer". The panel is organized into several sections:

- Script:** A dropdown menu showing "ScreenshotSharer".
- Check to disable sharing:** A checkbox labeled "Disable Sharing" which is currently unchecked.
- Sharing Config:**
 - Shared Image Format:** A dropdown menu set to "Both".
 - Share Message:** A text field containing the template "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 set to "15".
 - Gif Length:** A slider set to "3".
 - Gif Loop:** A text field containing "0".
 - Gif Quality:** A slider set to "80".
 - Gif Thread Priority:** A dropdown menu set to "Below Normal".
- 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
- You can also control the Giphy uploading activity with the following parameters:
 - *Giphy Username & Giphy Api Key*: provide these values if you want to upload the GIF image to your own Giphy channel; otherwise leave them empty to use the [Giphy beta key](#)
 - *Giphy Upload Tags*: comma-delimited list of tags of the uploaded image

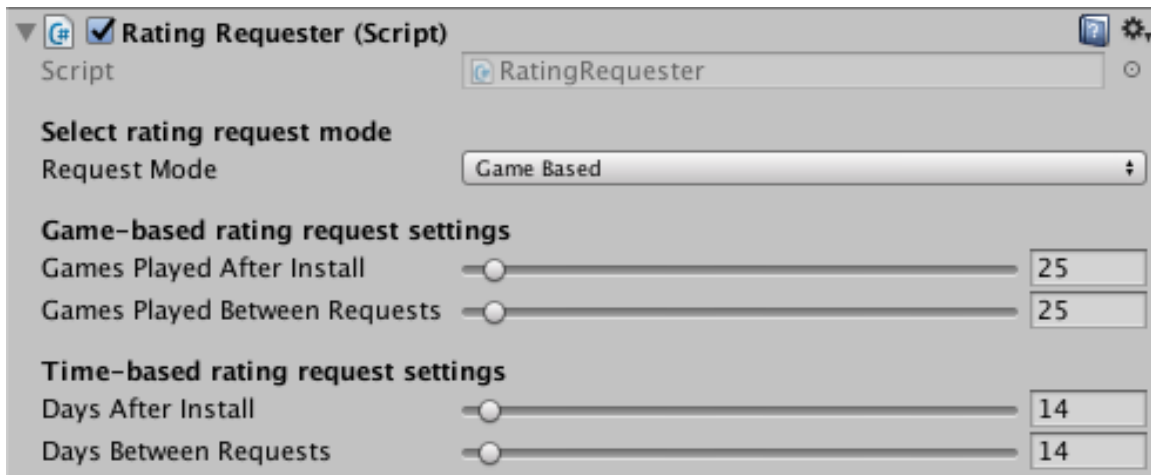
Note that you need to enable the *External Write Permission* for this feature to function properly on Android. Please see the Native Sharing section in Easy Mobile user guide for detailed instructions on doing that.

4.6 Rating Request

This template employs the Rating Request feature of Easy Mobile, to show a rate-

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

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



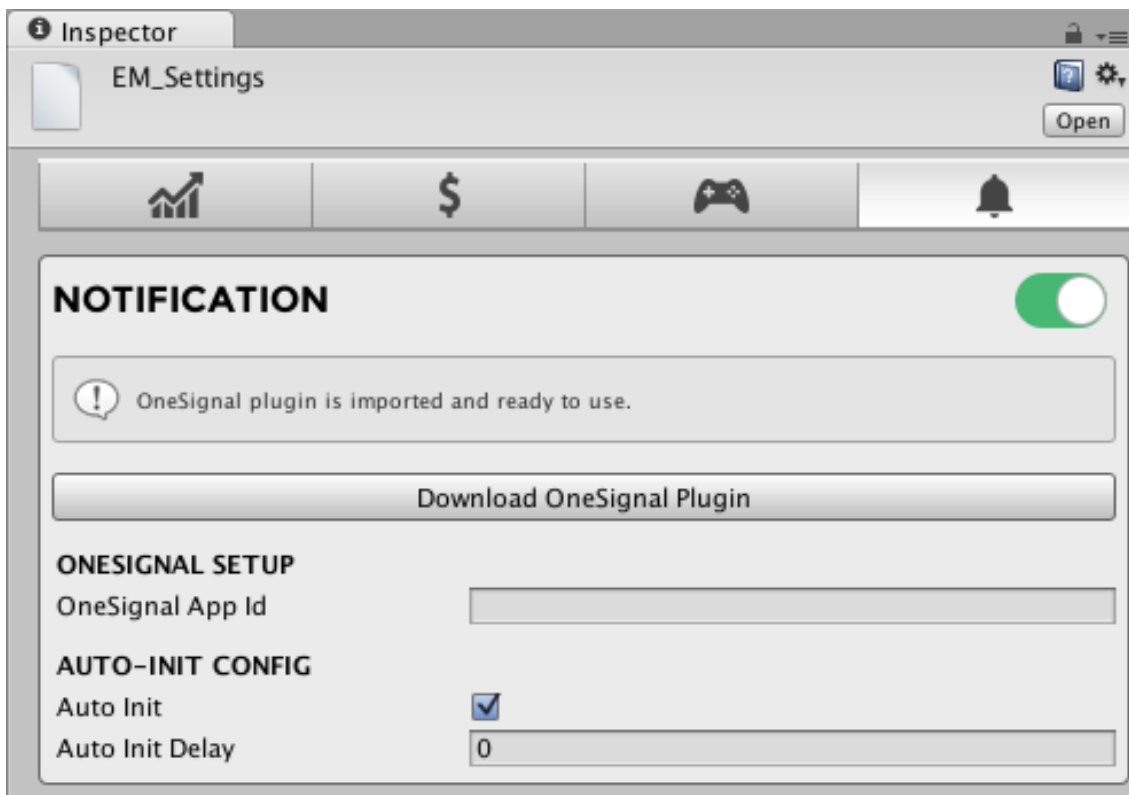
- *Request Mode*: whether to show the rating popup based on the number of games played (Game Based mode), or based on the time since the installation of the app (Time Based mode)
- If you select *Game Based* mode, pay attention to these two variables:
- *Games Played After Install*: how many games should be played since the installation before a rating popup is shown
- *Game Played Between Requests*: how many games should be played since the last time a rating popup is shown (in case it was dismissed by the user) that a new popup can be shown
- If you select *Time Based* mode, adjust these two variables:
- *Days After Install*: how many days after the installation that a rating popup is shown
- *Days Between Requests*: how many days since the last time a rating popup is shown that a new one can be shown

4.7 Push notification

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

- i. Open the Notification tab in Easy Mobile's settings interface
- ii. Import OneSignal plugin
- iii. 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).
- iv. Add your app to OneSignal dashboard
- v. 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!