TWIN BALLS

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
2.3 TESTING NOTE 6

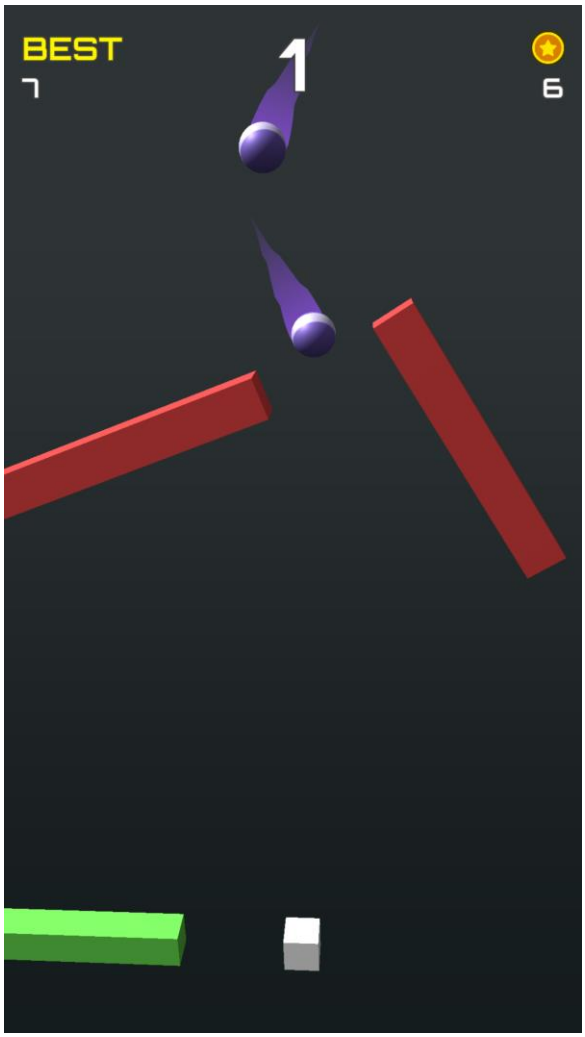
3 TEMPLATE CUSTOMIZATION 6

3.1 GAMEPLAY TWEAKING 6
3.2 PLAYER CONTROLLER 7
3.3 ADDING MORE CHARACTERS 8
3.4 CONTROL ANGULAR SPEED OF TWO BALLS 10
3.5 MANAGE OBSTACLES 10
3.6 CREATE NEW OBSTACLES AND IT'S INFORMATION 11
3.7 ROTATE OBSTACLE 13
3.8 MOVE OBSTACLE 14
3.9 MANAGE COIN 14
3.10 DAILY REWARD FEATURE 15
3.11 CHANGING BACKGROUND GRADIENT 16
3.12 CUSTOMIZING UI 16
3.13 SOUNDS 17

4 ENABLING PREMIUM FEATURES 19

4.1 BEFORE YOU BEGIN 19
4.2 ADVERTISING 20
4.2.1 Template-specific setup 20
4.2.2 Easy Mobile setup 21
4.3 IN-APP PURCHASING 23
4.3.1 Template-specific setup 23
4.3.2 Easy Mobile setup 25
4.3.3 Create the products for targeted stores 27
4.4 GAME SERVICE 27
4.4.1 Template-specific setup 27
4.4.2 Setup for your targeted stores 29
4.4.3 Easy Mobile setup 30
4.5 NATIVE SHARING 31
4.6 RATING REQUEST 32
4.7 PUSH NOTIFICATION 33

1 INTRODUCTION



Getting one ball through all the deadly obstacles can be fairly tough job already. What if there're two for you to juggle while getting them through? This simple yet challenging game will certainly give the users a lot of re-tries!

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

- Addictive one-touch gameplay
- Simple yet challenging
- Daily reward system for better retention
- 30 available characters.
- 20 obstacles difference from each other
- Optimized for mobile

Most importantly, when equipped with the **Easy Mobile** plugin, this template can form a truly full-featured game that is ready for release. Easy Mobile is a comprehensive, cross-platform package that provides most of desired features of mobile games:

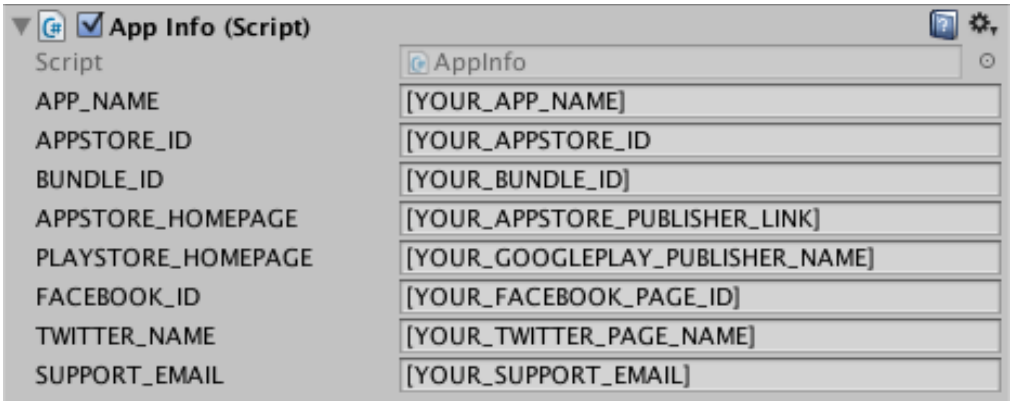
- Support for AdColony, AdMob, Chartboost, Heyzap and UnityAds
- In-app purchasing
- Support for Game Center (iOS) and Google Play Games Services (Android) for leaderboards and achievements
- Recording gameplay and exporting GIF images
- Sharing to social networks (PNG or GIF images)
- Push notification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

Native rating request popup (rate my app)

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.

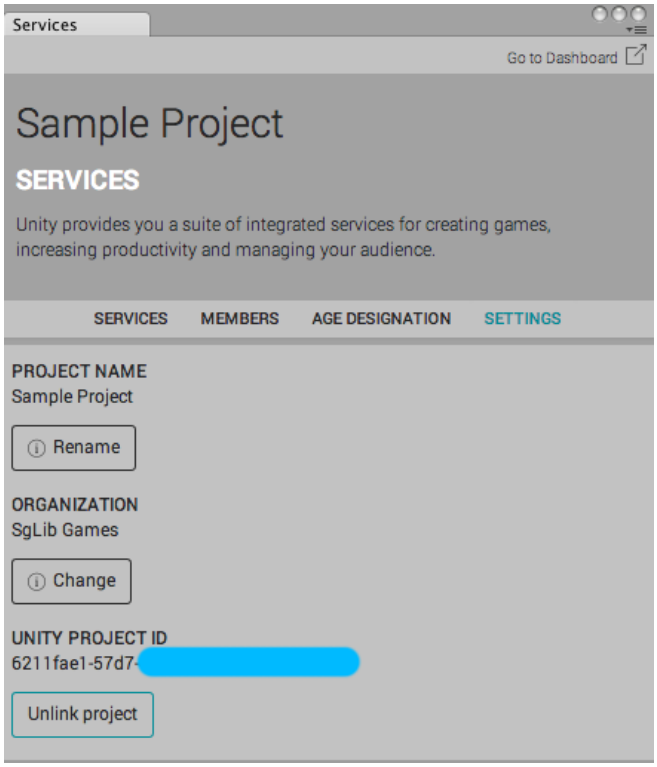


2.2 Link the game to your Unity project

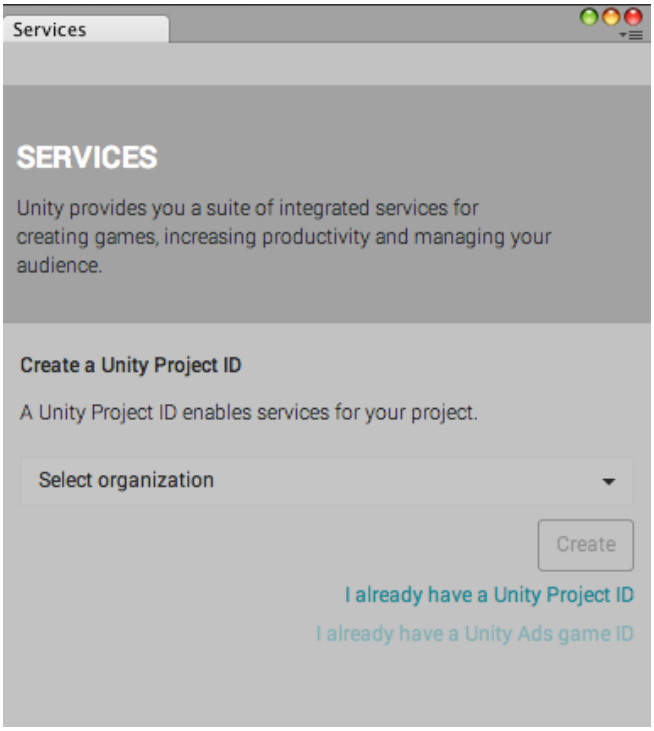
When developing this template, we normally need to link it to our own Unity project for testing, therefore you may need to unlink it from our project and link it to your own one, if you're going to use Unity services (e.g. if you want to enable premium features of this template, you'll need to use Unity 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 you game is linked to your own Unity project and is ready to use Unity services.

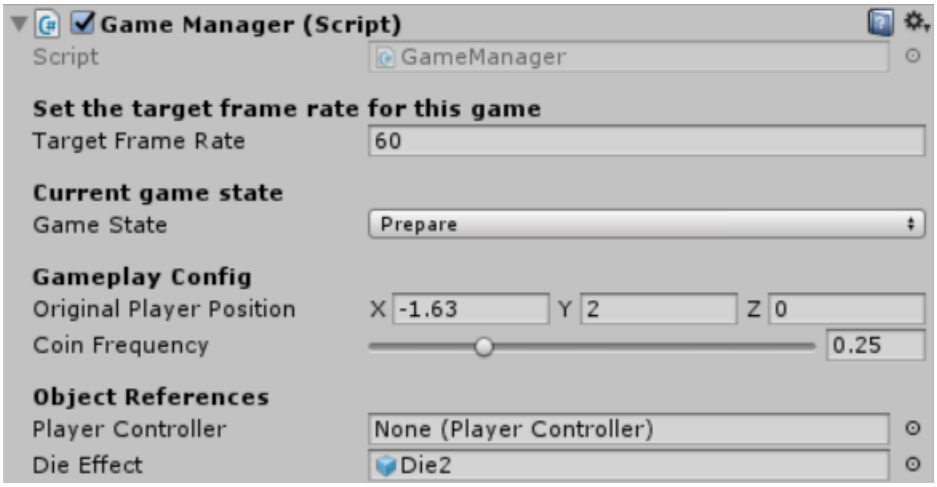
2.3 Testing Note

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.

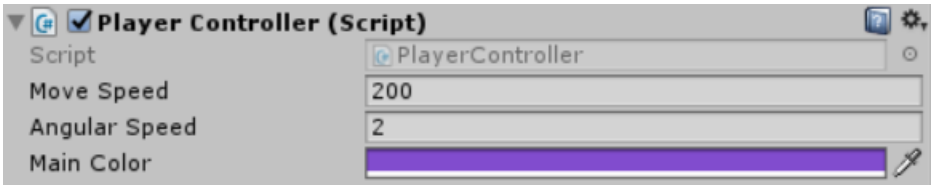


You can tweak the gameplay by modifying following variables:

- *TargetFrameRate*: the target frame rate for the game, which should be at least 60fps for games requiring smooth, fast motion.
- *CoinFrequency*: the appearance probability of “gold” (or coin).
- *OriginalPlayerPosition*: player position at start.
- *DieEffect*: dead effect of player.

3.2 Player controller

Each PlayerCharacter contains a PlayerController.



You can see a script has got name is “Player Controller”. It has got 3 properties which we can control:

- *MoveSpeed*: This properties affect to move speed of player. Fast or slow, if you adjust it.
- *AngularSpeed*: This attribute, you can adjust to two balls of player when they turn around player’s position. Still is fast or slow. They are make a circle when they turn around.
- *MainColor*: main color of player to set up color of trail and dieEffect.

3.3 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.

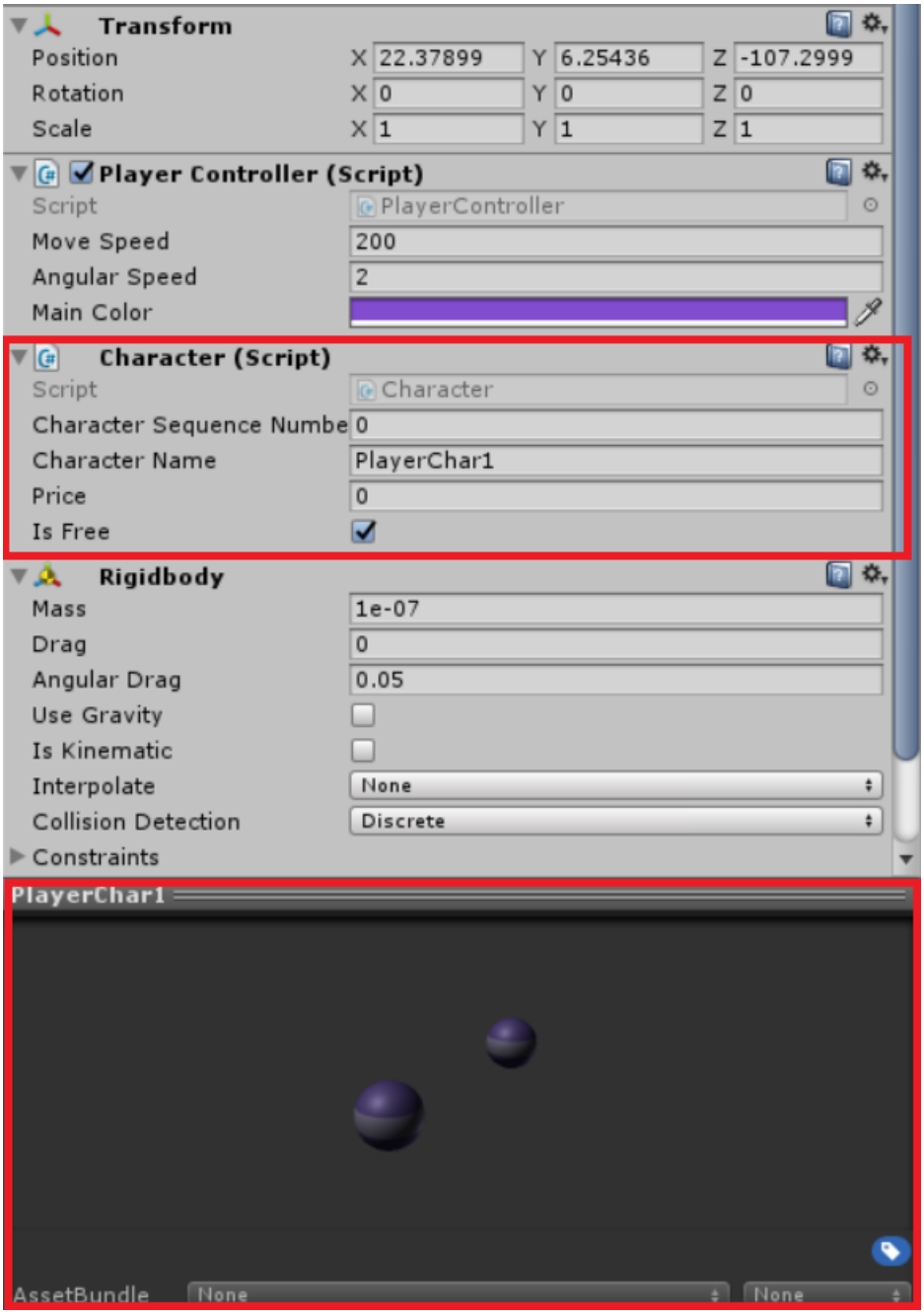
b. Navigate to *Assets/Prefabs/Game/Characters/CharacterPrefabs* and duplicate one of the available character prefabs.

c. Change the name of the prefab to a preferred one.

d. Each PlayerCharacter contains 2 balls. Replace the *Material* of balls in the *MeshRenderer* component with your new character material.

e. 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.*

f. Resize the character array in *CharacterManager* game object then drag the new character to it and hit Apply to save changes to its prefab.



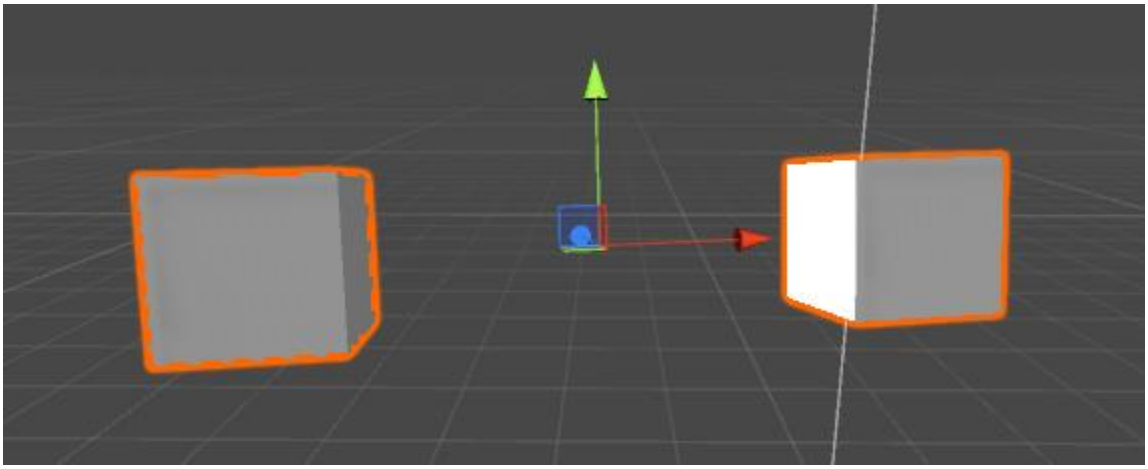
* IMPORTANT: **TRAIL EFFECT OF THE CHARACTER**

Every ball of character in this template have one Particle System component that is trail effect of character. Please make sure your character follows the rules.

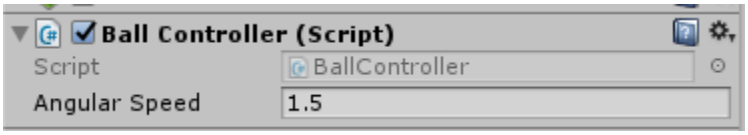
Now the new character has been added and ready to use in game! You will see it listed in the *CharacterSelelection* scene.

3.4 Control angular speed of two balls

You can control angular speed of them – two balls. At this game, player is the object which have got two ball and them always turn around a position.

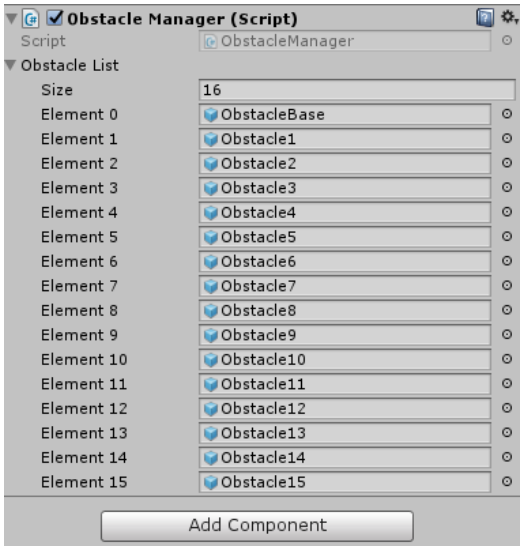


You can control it – angular speed. If you want to these balls turn around with fast speed, you can increase value of them. Otherwise, you can decrease value of them.



3.5 Manage Obstacles

This component has got name is ObstacleManager and it is attached to a game object also named ObstacleManager in the hierarchy.



Whenever you have got any obstacle which want to attach into game, then you can drag it and drop into ObstacleList properties of ObstacleManager component.

3.6 Create new obstacles and it's information

Besides, you can create new obstacles if you want.

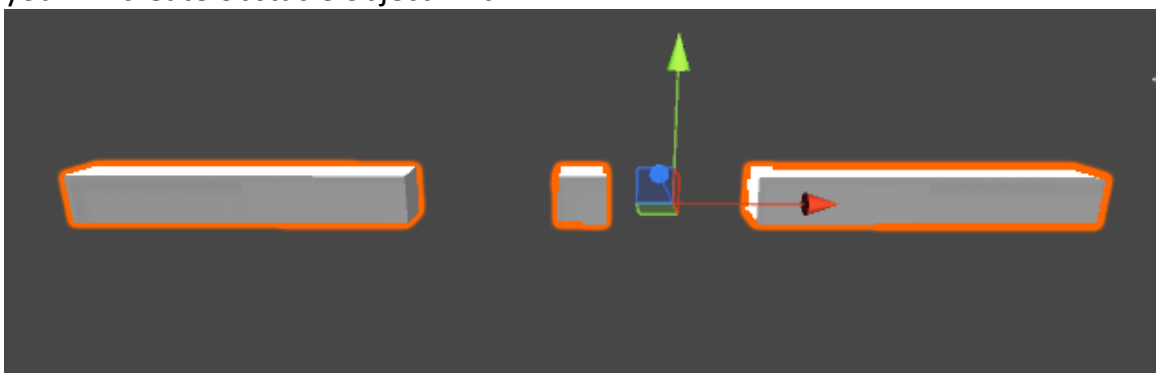
We were created some scripts to serve for this require. Include:

- Move Obstacle
- Rotate Obstacle
- Destroy Obstacle

And script "Obstacle Controller", also as every obstacle object which you created, it – Obstacle Controller, will assign to them – Obstacles are created by you.

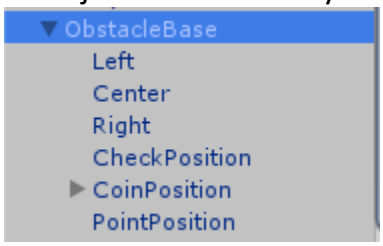
To create a new obstacle, you need:

- Create a new object in hierarchy tab with name "*Obstacle(number)".* Then, you will create obstacle object in it.

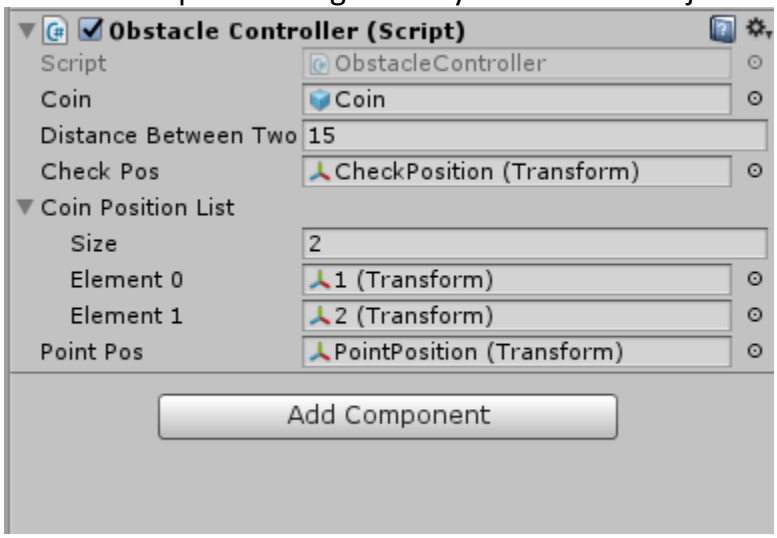


See above picture, and you will see three obstacle objects.

- Next, you will create some objects in hiearchy. Include:



  - *Left*: this object is left obstacle which we were created through above picture.
  - *Center*: obstacle object at center
  - *Right*: obstacle object at right
  - *CheckPosition*: with this object, you will need care it because you drag it into ObstacleController. It affect to create new obstacles automatically.
  - *CoinPosition*: This object contain some child objects or doesn't. Such as on, it is important object. And you must be drag it on ObstacleController. This object can be created new coin when player is passed the obstacle.
  - *PointPosition*: This object can be used to calculate point of player when he passed the obstacle. It automatically increase point for him.
- So, you was created all objects which are needed for create a new obstacle.
- Next, you will drag all objects which created by you, and drop them into obstacleController script was assigned on your obstacle object.



  - *Coin*: you will drag coin object in prefabs folder and drop it at here.

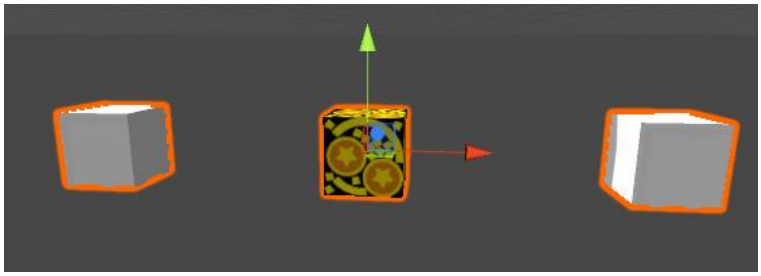
- *Distance Between two obstacles*: This property is use for calculating distance between old obstacle and new obstacle which spawned when player passed old obstacle.
- *CheckPos*: As mention in above, your check position object which created by you in obstacle object at hierarchy tab, you will drag it and drop into here.
- *PointPos*: The same as above property.
- *CoinPositionList*: you will drag all coin position which you was created in obstacle object and drop all of them into here.

- Finally, after you had completed all drags and drop work, you finished for create a new obstacle. Then, you will bring it and drop into prefab folder. Next, you will drag obstacle which you have just created and drop it into "ObstacleManger" in hierarchy tab, in its property "ObstacleList". You will drop your obstacle at here.

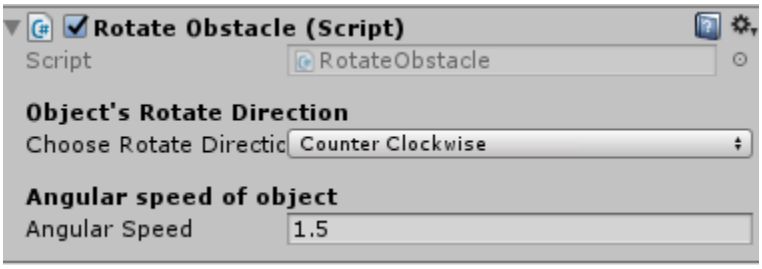
3.7 Rotate obstacle

After you had completed a new obstacle, but you want to custom it as rotate or move obstacle objects, you will care this script – Rotare obstacle or under script – Move obstacle.

This script is created and assigned for obstacle object when you want to need rotare it.



Above picture is mention about center obstacle. This object is assigned "Rotate Obstacle" because we need rotate it.

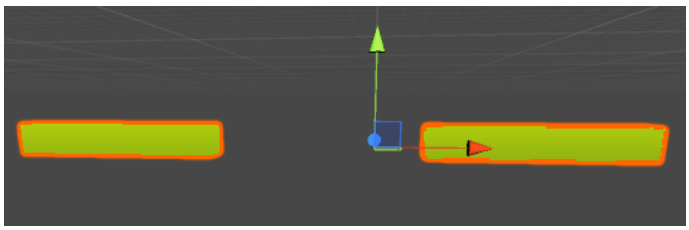


It has got:

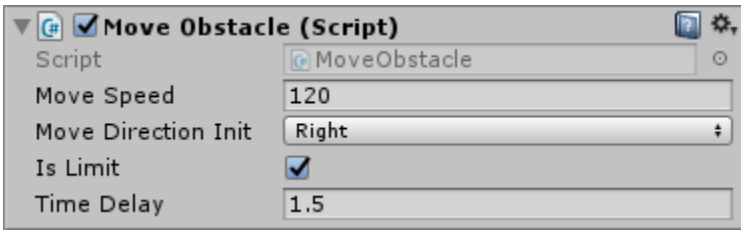
- *ChooseRotateDirection:* This property allow for you to choose rotate direction of obstacle. It has got two values **CounterClockwise** and **Clockwise**. Also it has got function same as it's name.
- *AngularSpeed:* This property affect to fast or slow speed of obstacle when it turn around.

3.8 Move Obstacle

Above mention, this script is served for move your obstacle.



Two obstacle objects in above picture are created and assigned "Move Obstacle".

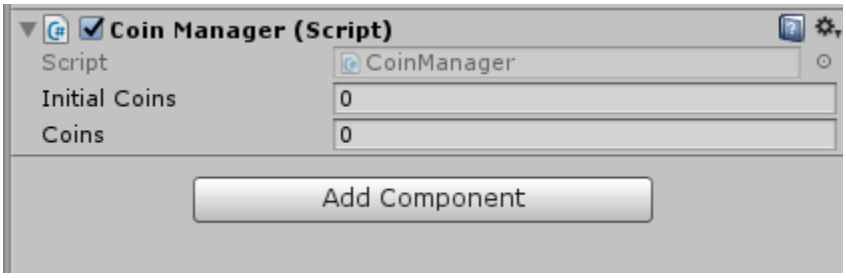


Above picture is "Move Obstacle" and it has got:

- *MoveSpeed:* Affect to fast or slow speed of obstacle object. 120 is normal value for move of obstacle.
- *MoveDirectionInit:* This property will decide move direction of obstacle when it begin to move. Right or Left, Up or Down.
- *IsLimit:* Your obstacle has got been limited space move? If you want to limit space move of it, you will check this here. Otherwise, you won't check.
- *TimeDelay:* time for delay when your obstacle is "IsLimit". It's mean you was checked at above property "IsLimit". If you was checked, then this property has just been used. It's value decide time amount whenever obstacle move and exhausted time, it will change direction to reverse.

3.9 Manage Coin

You can configured coin stat for game within the CoinManager component which is attached to a game object also named CoinManager in the hierarchy.

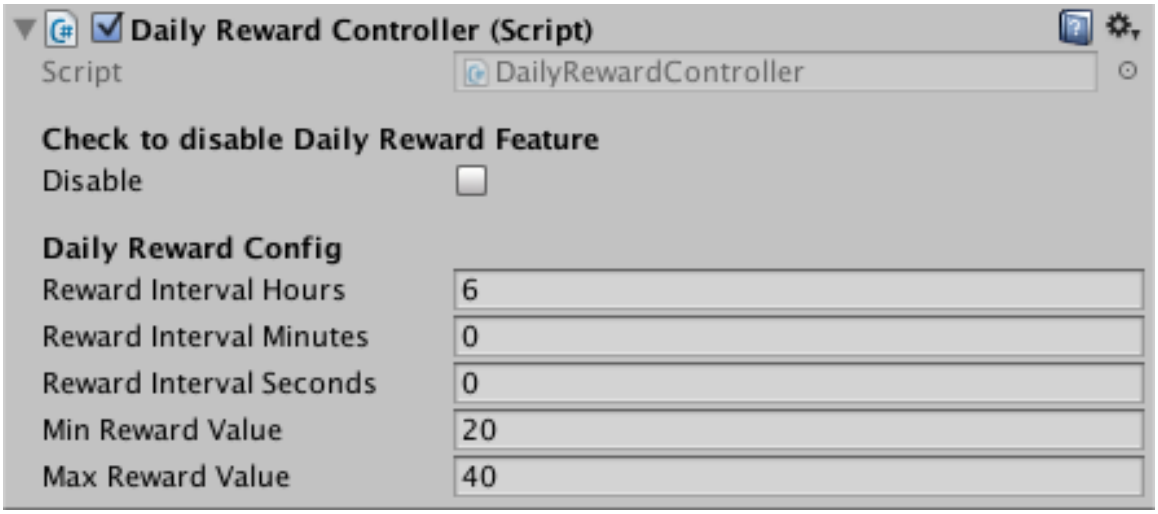


You can tweak stats:

- *InitialCoins*: This is a coin stat which is used for assign coin value when in game start the first time.
- *Coins*: This is current value of coin. If game start the first time, it will be assign with *InitialCoins* value. Then, if player earn new coin, it will be incredible value.

3.10 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.

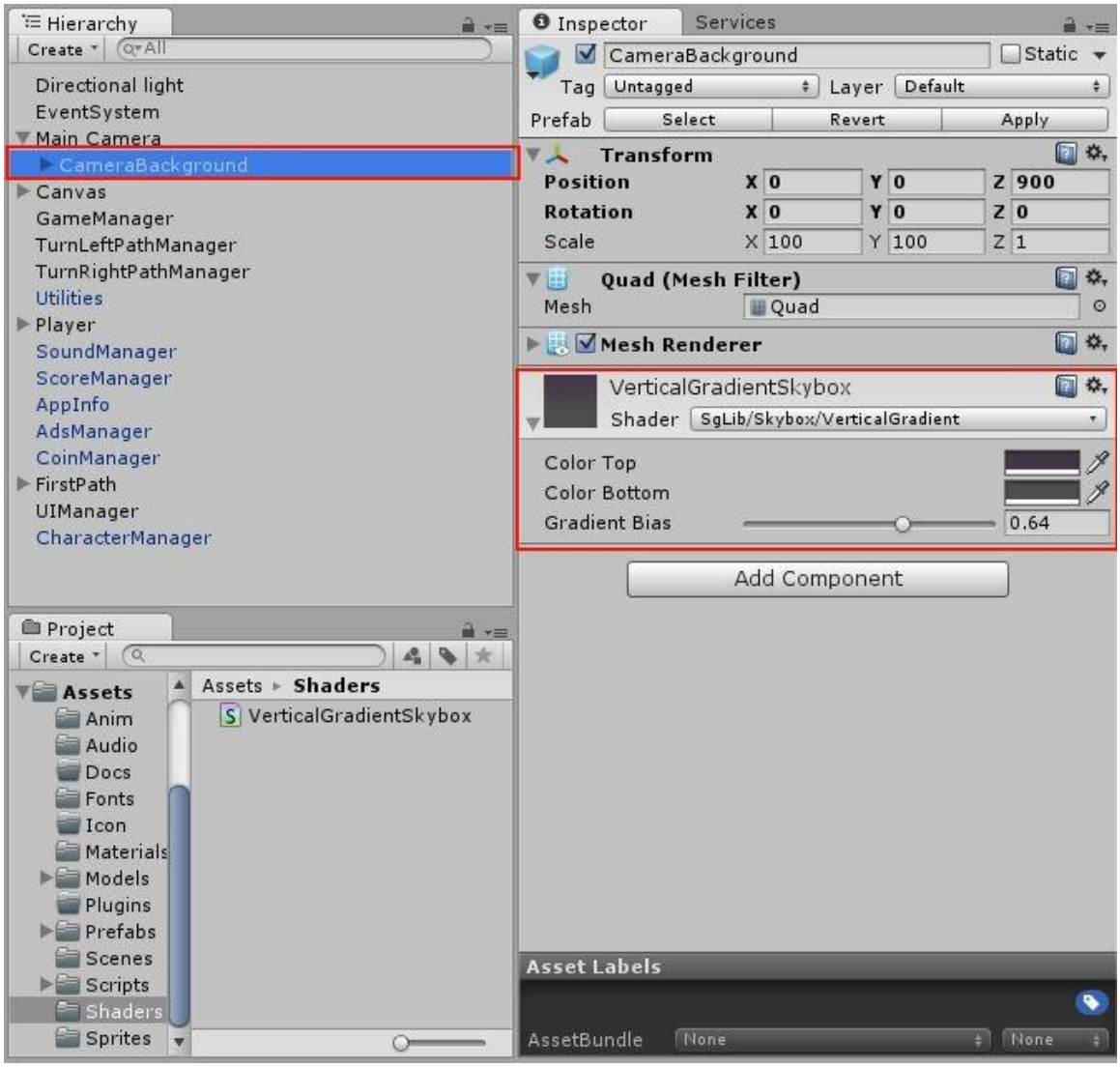


- *Disable*: check to disable this feature
- *Reward Interval Hours, Minutes and Seconds*: the amount of time until the next reward
- *Min Reward Value* & *Max Reward Value*: the actual rewarded coins will be randomized between these two values

3.11 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 vertext/fragment shader. To modify the background gradient colors:

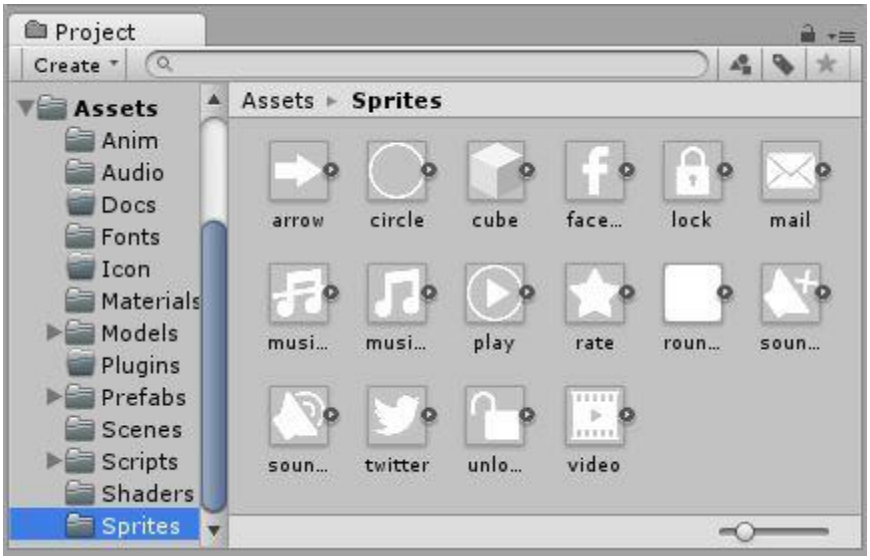
- Navigate to game object *CameraBackground* under *MainCamera*.
- Modify the gradient colors: *ColorTop*, *ColorBottom* and adjust the *GradientBias* until you're satisfied with the result.
- Hit *Apply* to save changes.



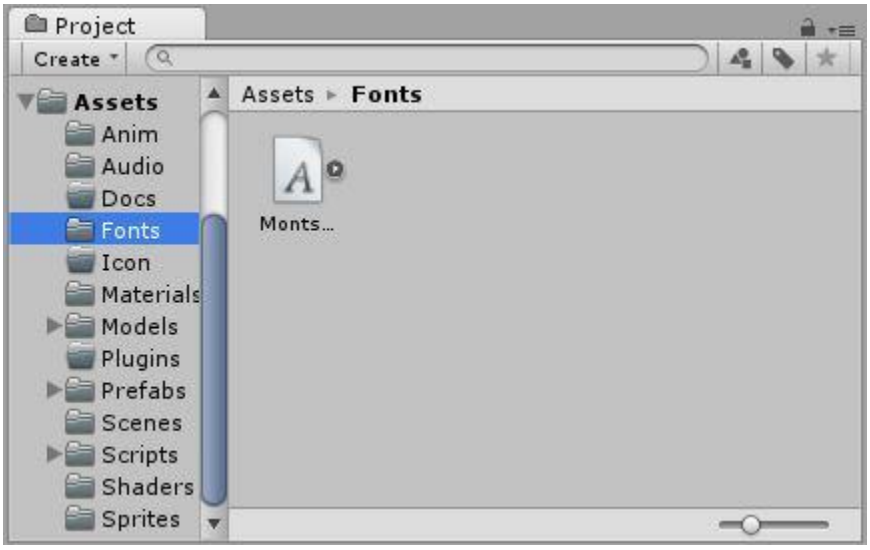
3.12 Customizing UI

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.

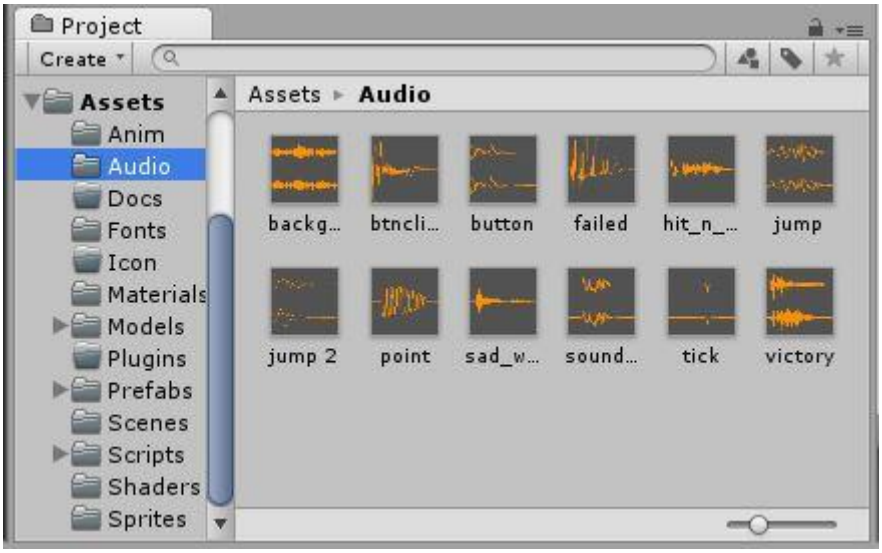


All fonts used in this game are free-to-use in commercial projects. Fonts are located under the *Fonts* folder together with appropriate license files.

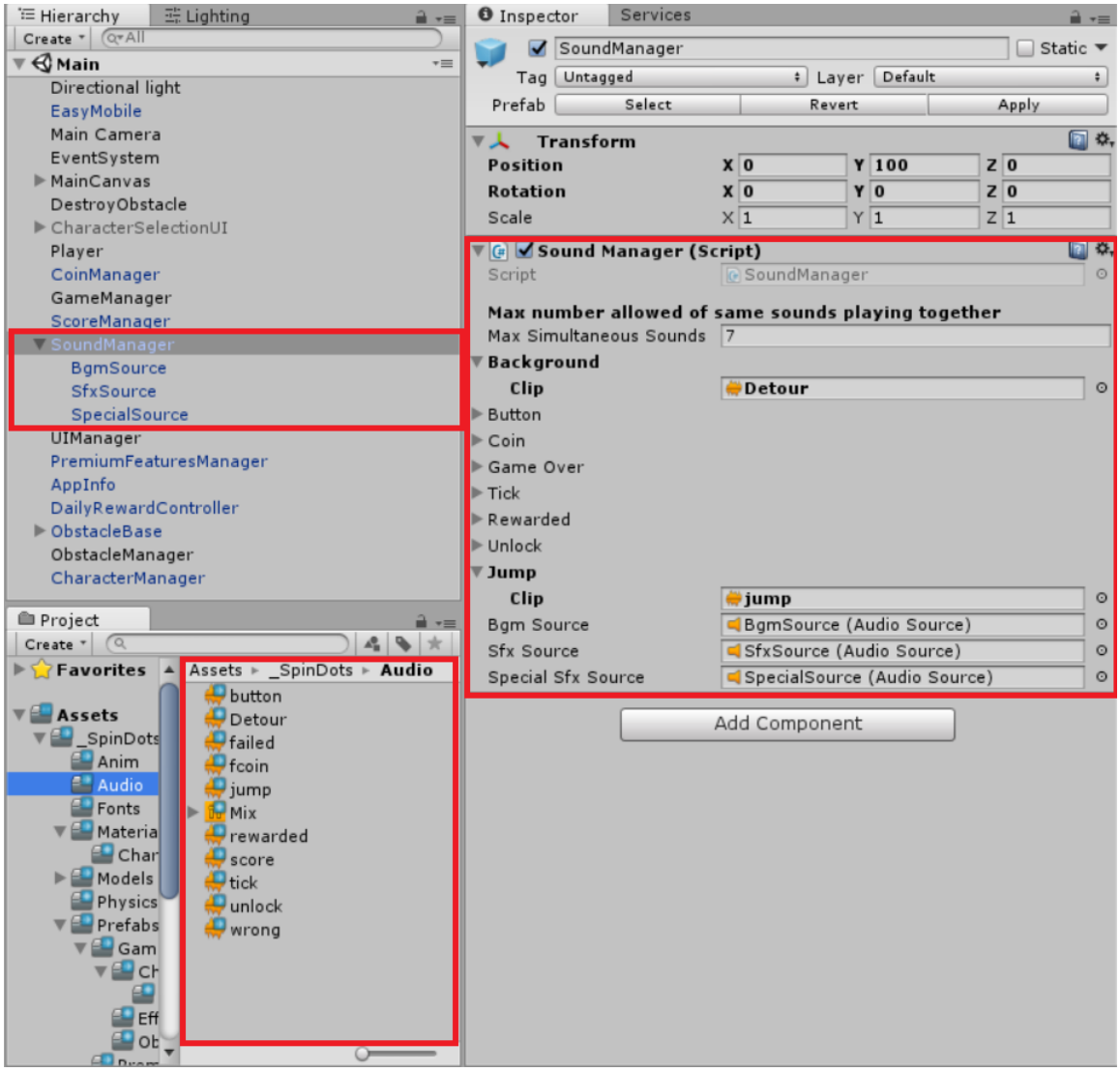


3.13 Sounds

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

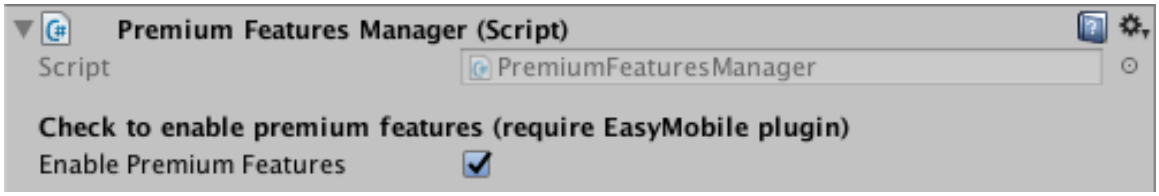
This section provides a guide on configuring premium features for your game. As these features are implemented using Easy Mobile plugin, this guide is intended to be used alongside Easy Mobile's official user guide. Therefore, it is strongly recommended that you also read through that guide to familiarize yourself with the plugin. You can access the online user guide from menu *Window > Easy Mobile > Online Documentation* (after Easy Mobile is imported to the project).

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.

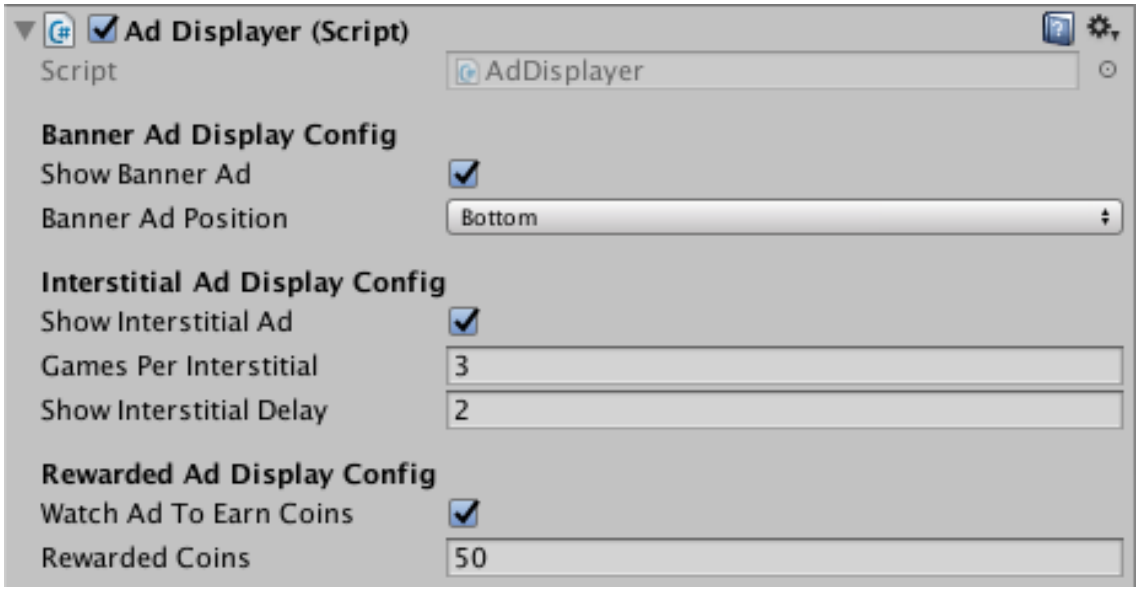


- The settings interface of Easy Mobile can be opened via menu *Window > Easy Mobile > Settings*, this is the only place to go to configure this plugin.
- Note that you won't need to write a single line of integration code for Easy Mobile to work, as the integration was done beforehand, you only need 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.

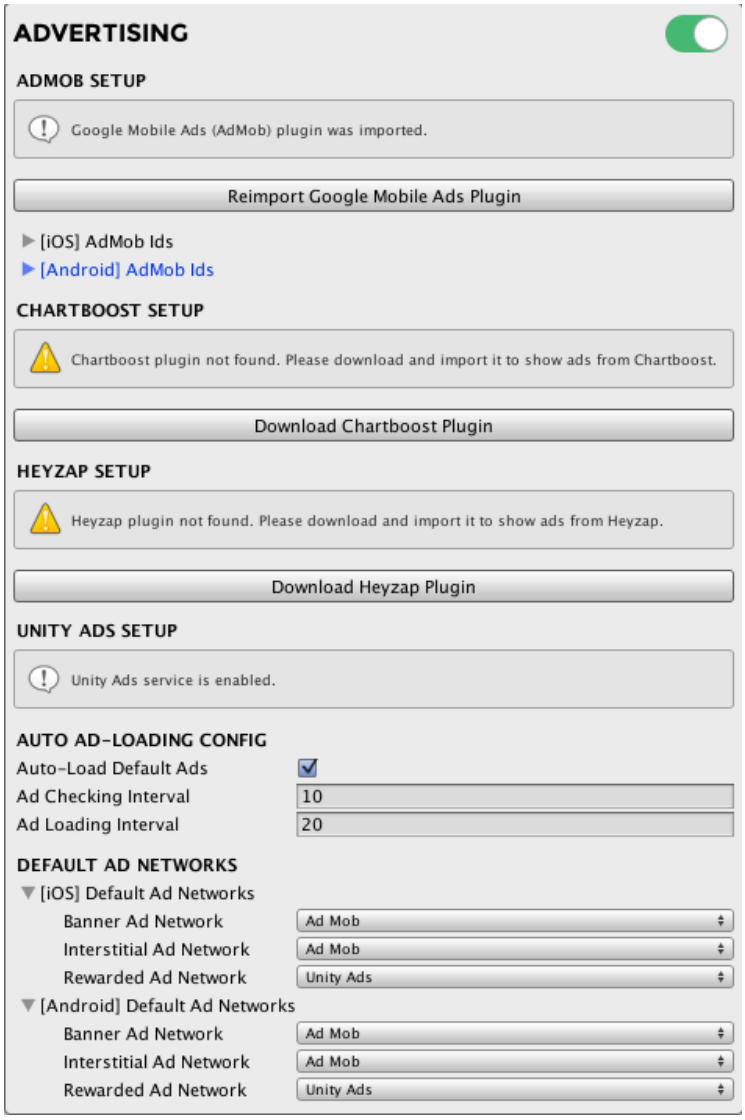
- *Show interstitial ad*: whether to show interstitial ads when game over
- *Games Per Interstitial*: how many games to be played before showing ad
- *Show Interstitial Delay*: how many seconds after game over that ad is shown

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.



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.

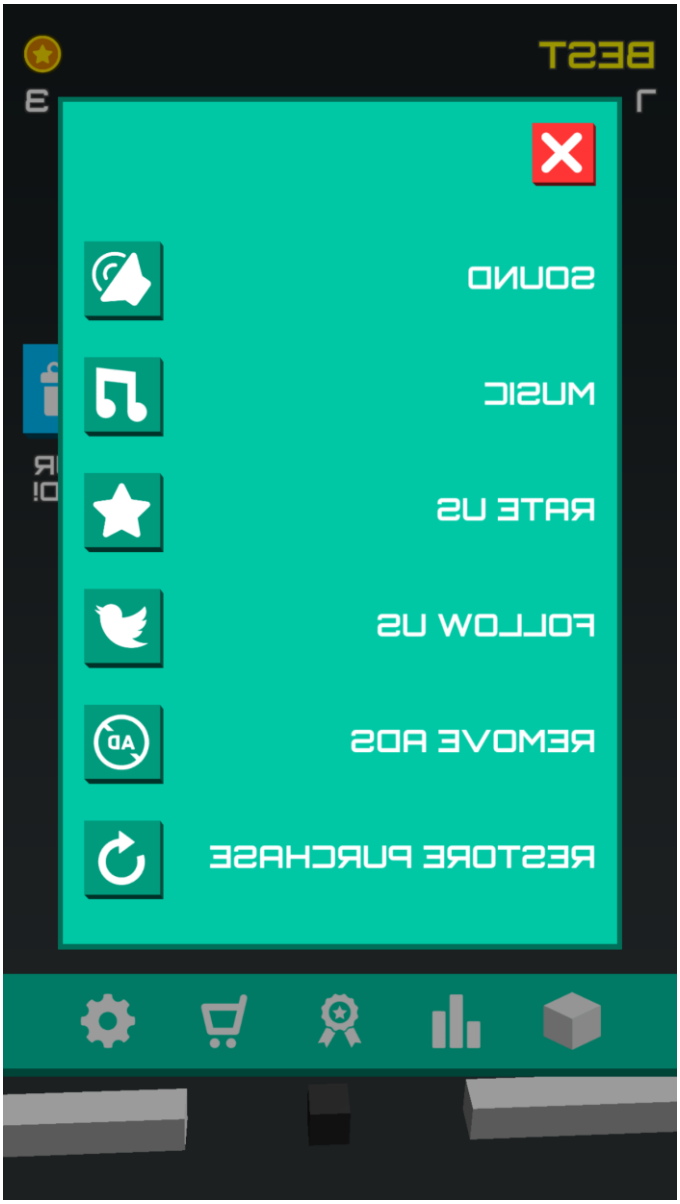
a. Setup the ad networks you want to use, including importing the required plugins, please see Easy Mobile user guide for more information
b. Enable auto ad-loading feature: simply leave the *Auto-Load Default Ads* option as checked and other parameters as default, the plugin will automatically load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

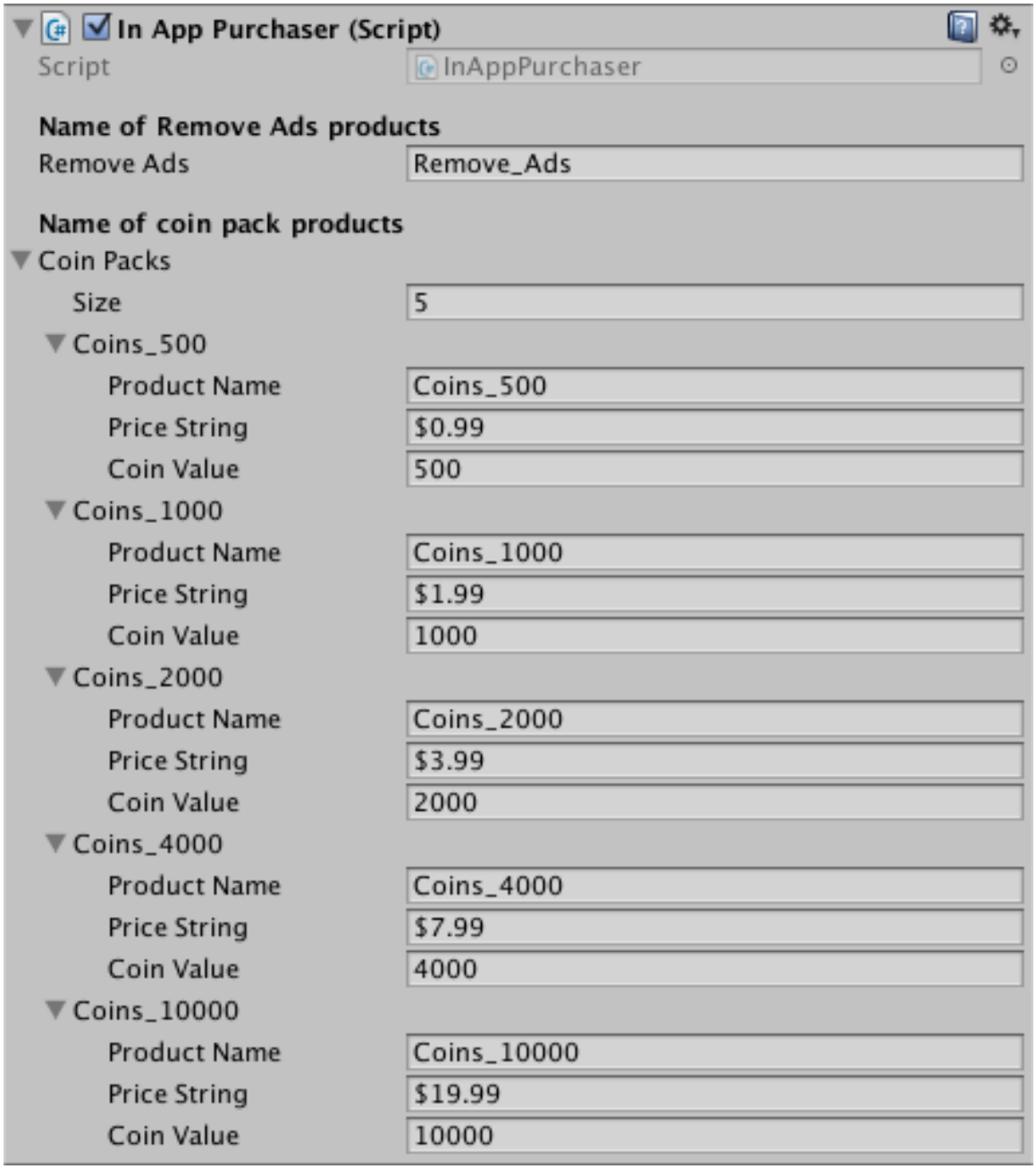
4.3 In-App Purchasing

4.3.1 Template-specific setup

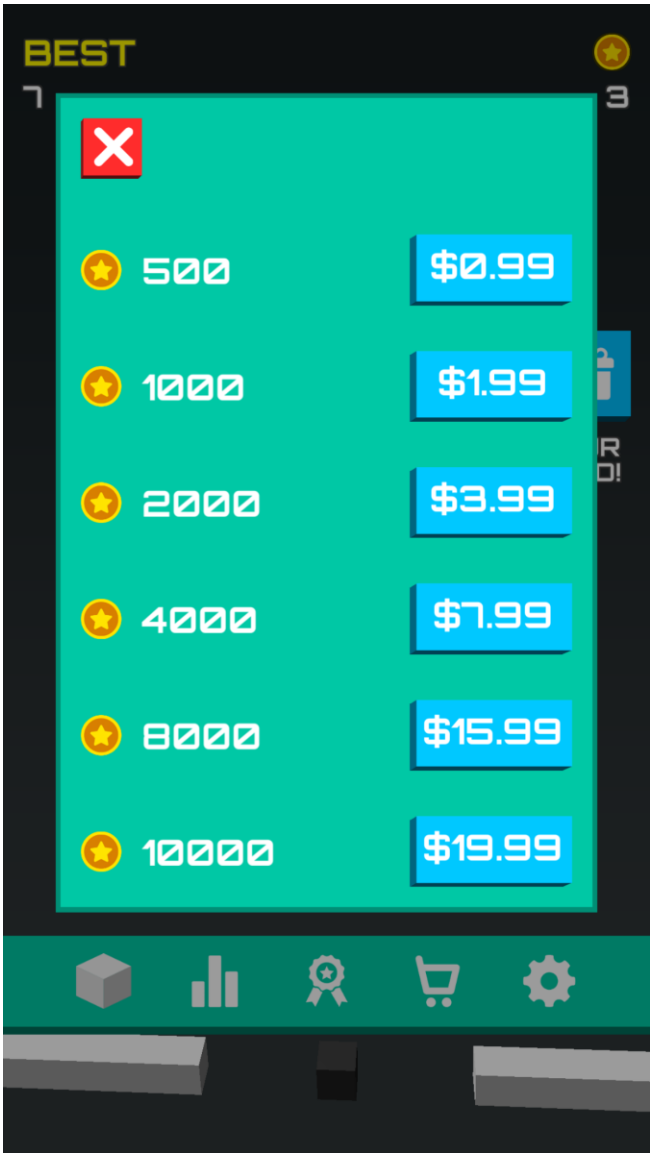
The built-in in-app purchases of this template include a *Remove Ads* button, and several coin packs. You can modify existing products and add more coin packs if you like. There's also one *Restore Purchase* button as required on iOS.



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



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.

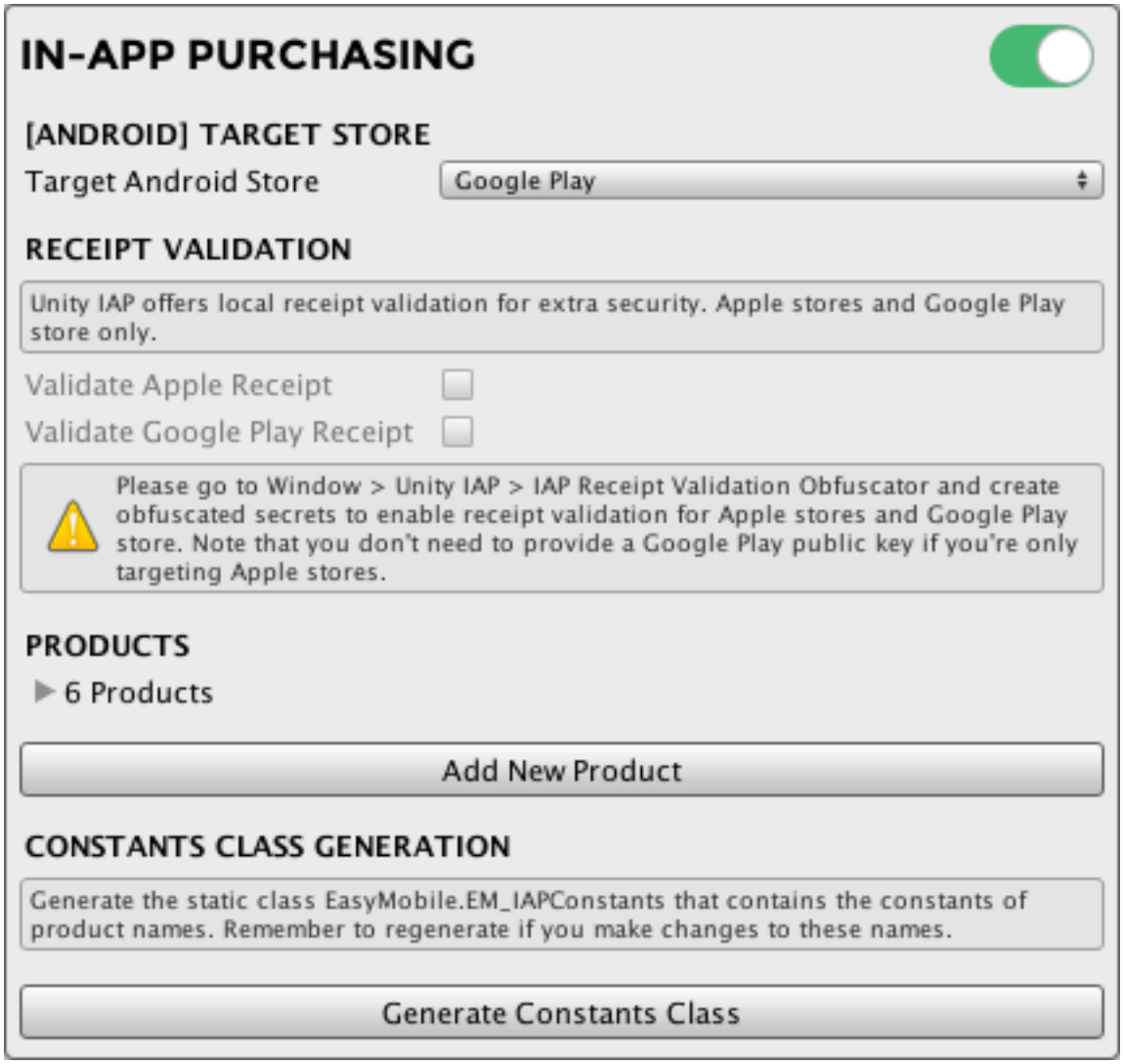


4.3.2 Easy Mobile setup

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.

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.



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.

4.3.3 Create the products for targeted stores

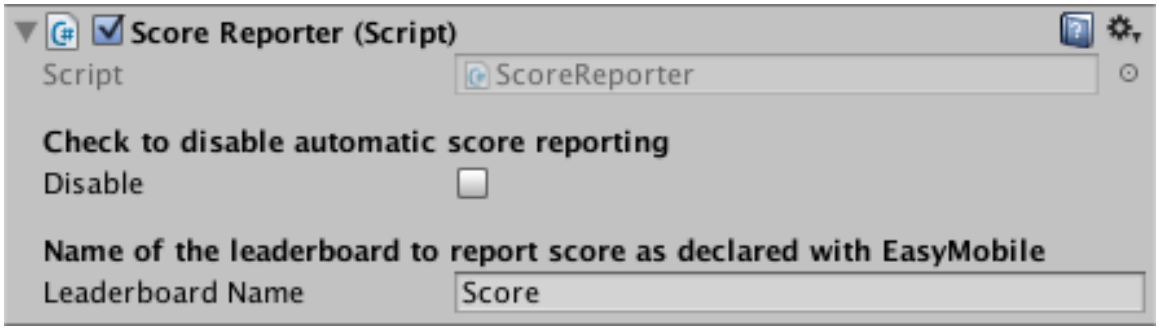
That last step in configuring the in-app purchasing feature is to create products for your targeted stores (e.g. Google Play and Apple App Store). Make sure the product ID, product type and price match the ones you have in your game.

4.4 Game Service

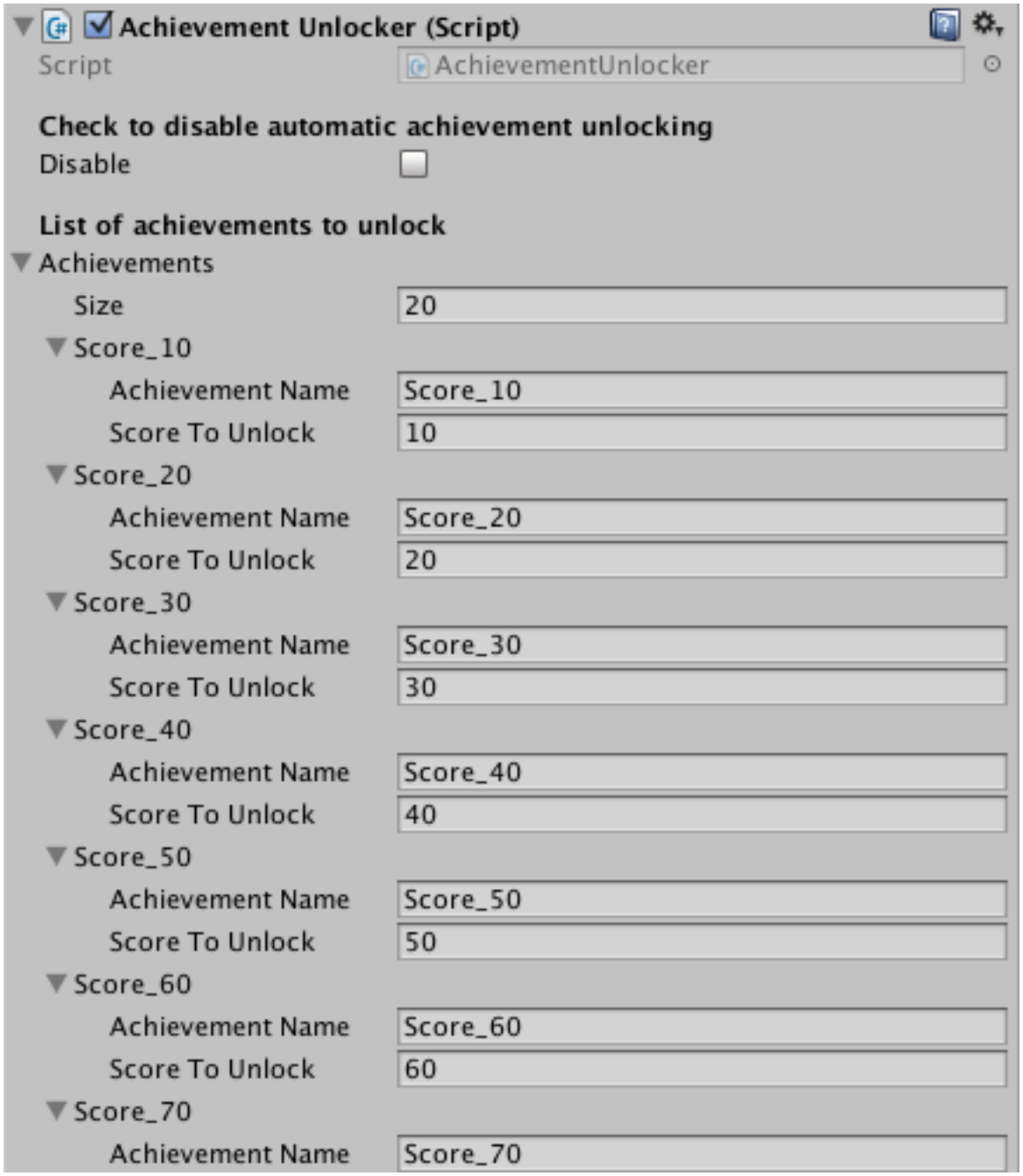
4.4.1 Template-specific setup

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.



4.4.2 Setup for your targeted stores

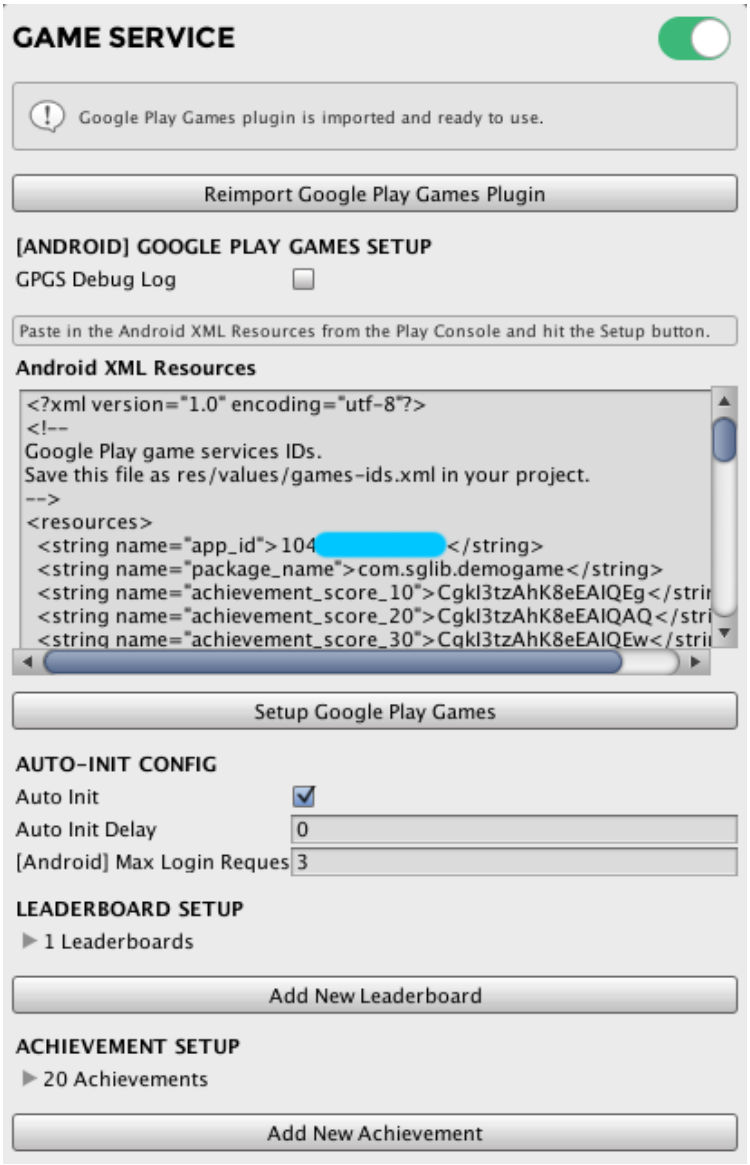
The next step is to create the required leaderboard and achievements for your targeted stores (i.e. in iTunes Connect for App Store and the Developer Console for Google Play). Take note of their IDs for use in the next step.

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.

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

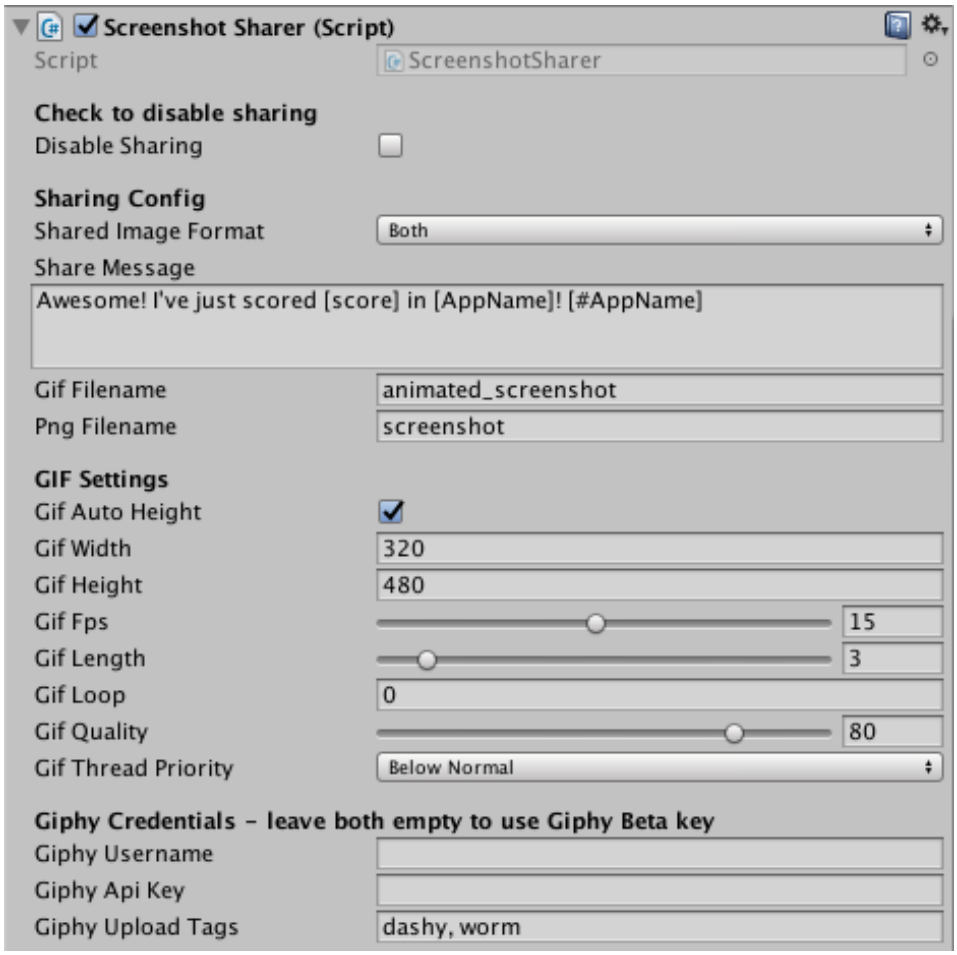
Below is the settings interface of the Game Service module of Easy Mobile.



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.



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:

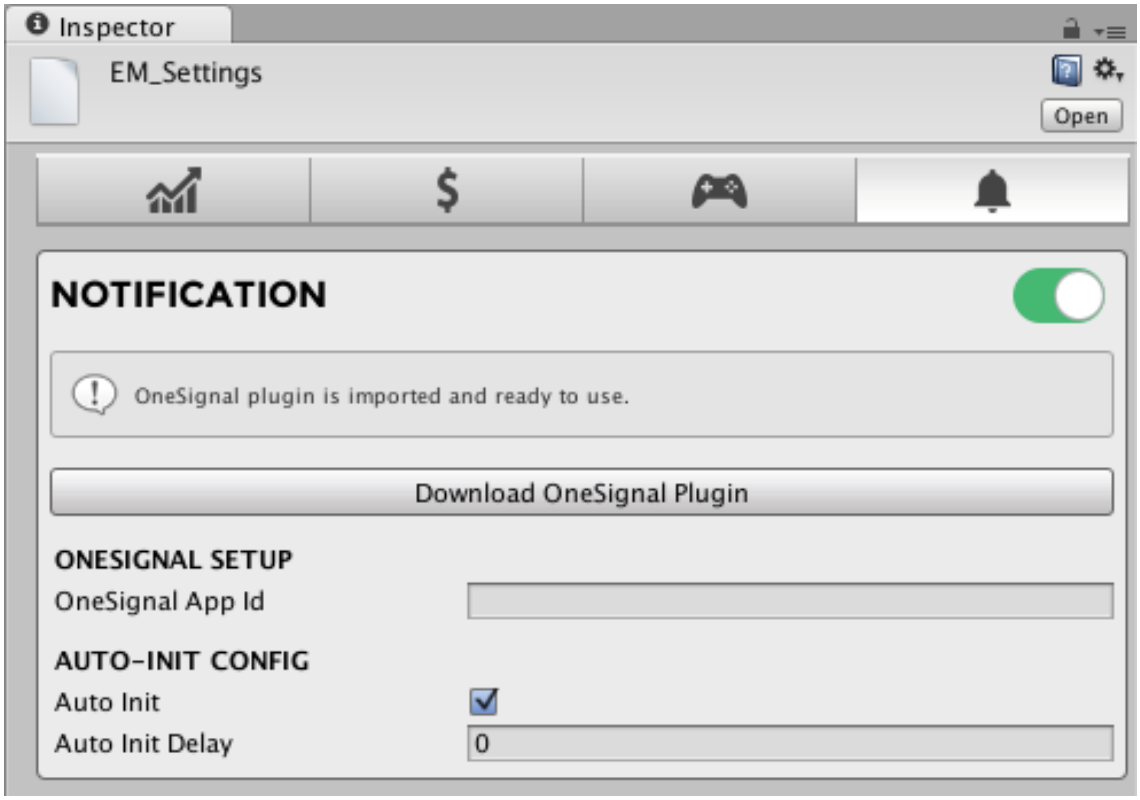
- *Days After Install*: how many days after the installation that a rating popup is shown
- *Days Between Requests*: how many days since the last time a rating popup is shown that a new one can be shown

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.

- 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!