SIMPLE PINBALL TEMPLATE

USER GUIDE

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

SgLib Games
sglib.games@gmail.com

Table of Contents

1   INTRODUCTION

In **Simple Pinball**, the player needs to aim the ball accurately using the two flickers in order to destroy the targets in a short amount of time. Hopefully this template can serve as a base for your own awesome pinball game.

Some highlights of this template

- Using pure Unity physics
- Free-to-use assets (fonts, sounds, music, model, etc.)
- Optimized for iOS and Android

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

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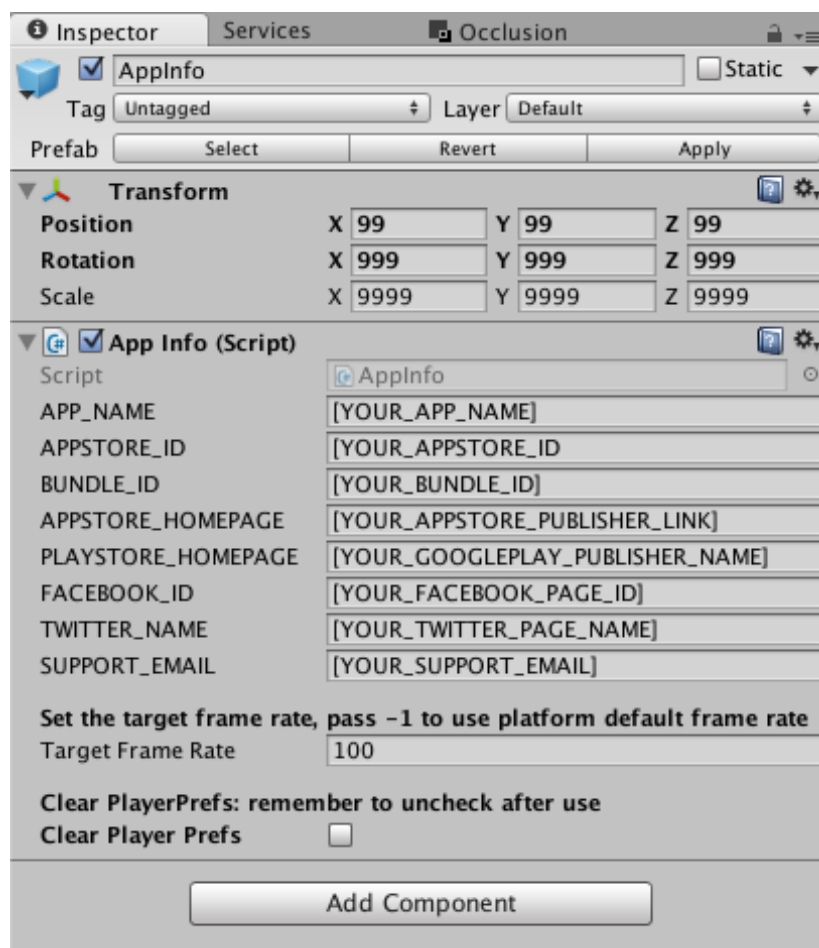
Besides basic app info, there're 2 other important variables:

- *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 *PlayerPrefes.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
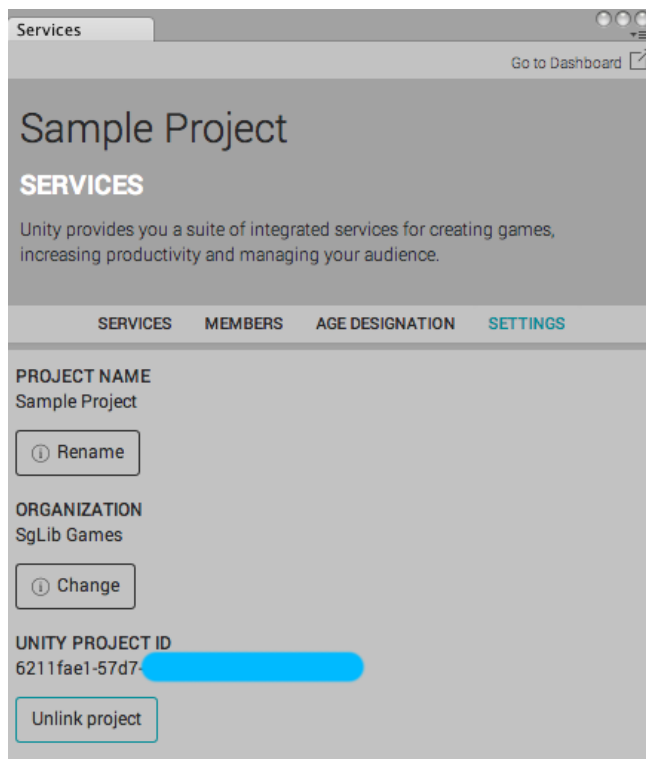
Now you can create a new project for the game.
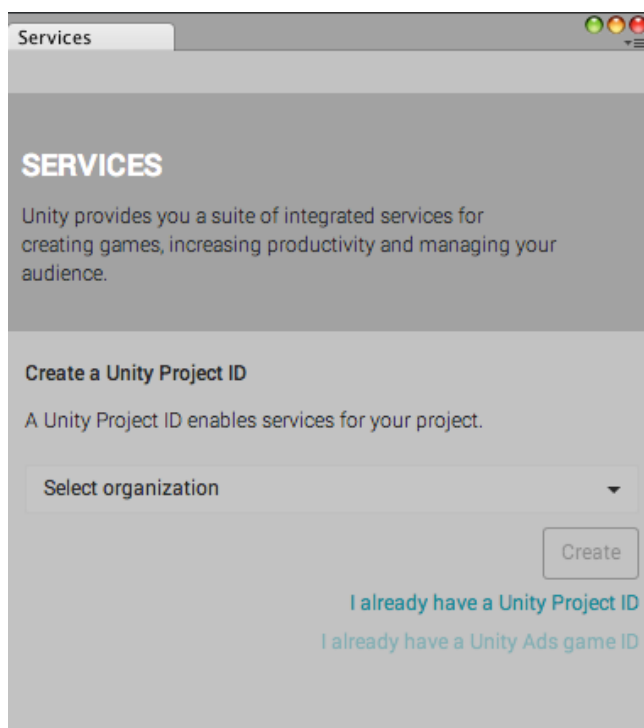
Now you game is linked to your own Unity project and is ready to use Unity services.

2.3 Testing Note

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

3 TEMPLATE CUSTOMIZATION

3.1 Toggling premium features UI

In the hierarchy there's an object named *GameManager* which manages most of the important aspects of the game. One of the important options is enabling or disable premium features via the *EnablePremiumFeatures* toggle. Enabling these features will make the relevant buttons (leaderboard, in-app purchase, sharing, etc.) visible in the game UI. Otherwise, these UI elements will be hidden. Note that this option only displays or hides the UI buttons, for these buttons to actually function you need to import the Easy Mobile plugin as mentioned above. Please see the **Enabling Premium Features** section below for more information.
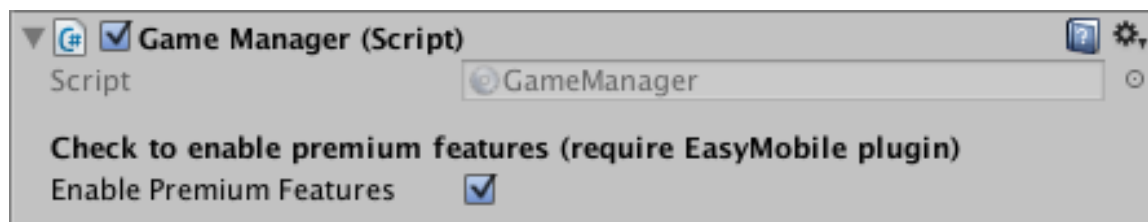
3.2 Gameplay tweaking

3.2.1 *GameManager*

The *GameManager* object also holds important parameters which you can adjust to tweak the gameplay.
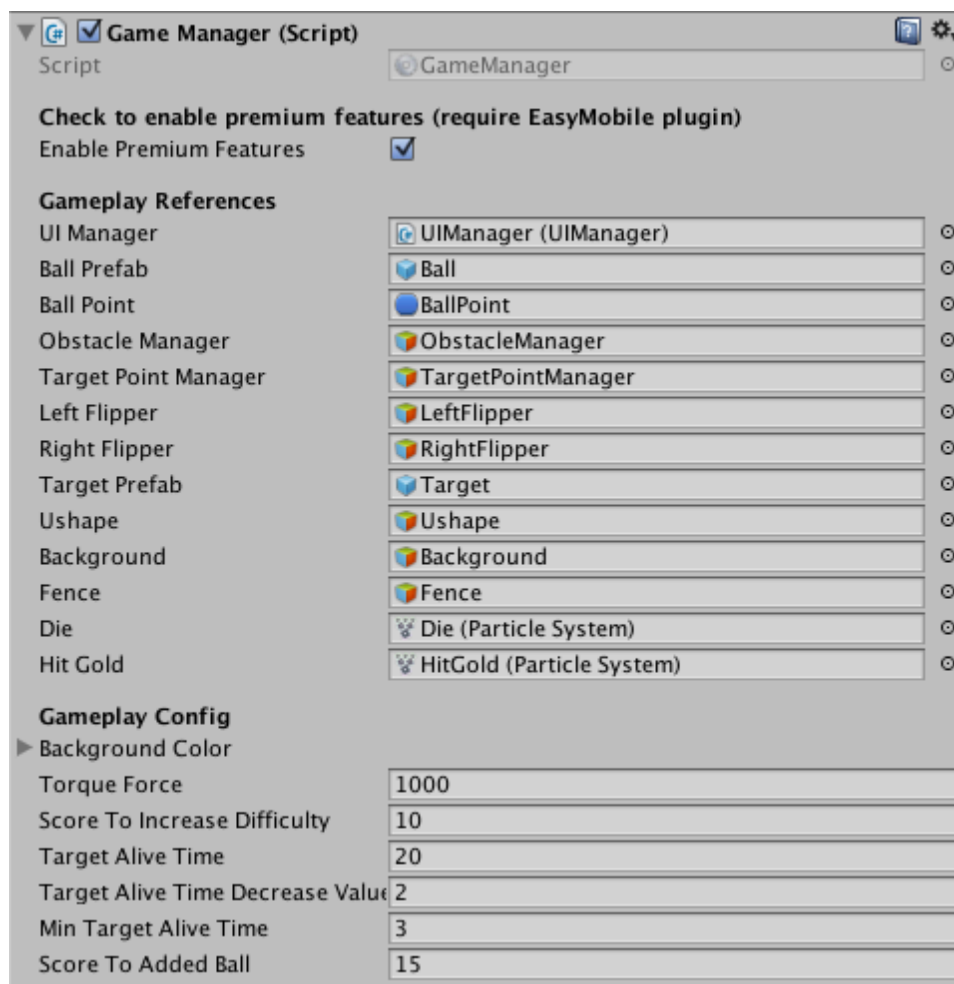
You can tweak the gameplay by modifying following variables:

- *BackgroundColor*: the color of background will be randomly generated by this array.

- *TorqueForce*: the force use to rotate the flipper

- *ScoreToIncreaseDifficulty*: the score to update difficulty values (see below variables)

- *TargetAliveTime*: the time allowed to destroy a target.

- *TargetAliveTimeDecreaseValue*: the *TargetAliveTime* will be decreased by this value every time the score reaches a multiple of

ScoreToIncreaseDifficulty.

- *MinTargetAliveTime*: min limit of *TargetAliveTime*.
- *ScoreToAddedBall*: score to added more balls.

3.3 Customizing UI

All sprites used in this game (for buttons and other UI components) are located under the *Sprites* folder. You can replace them with your own sprites to modify the UI as you like.
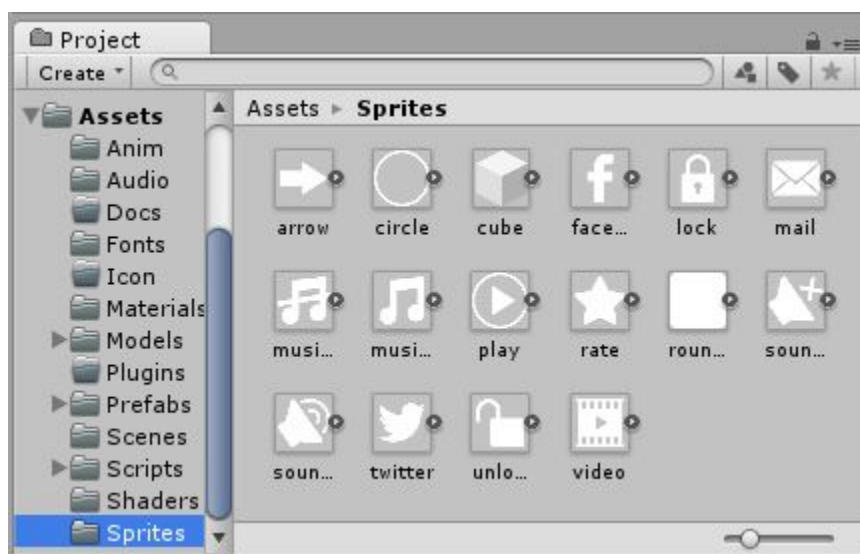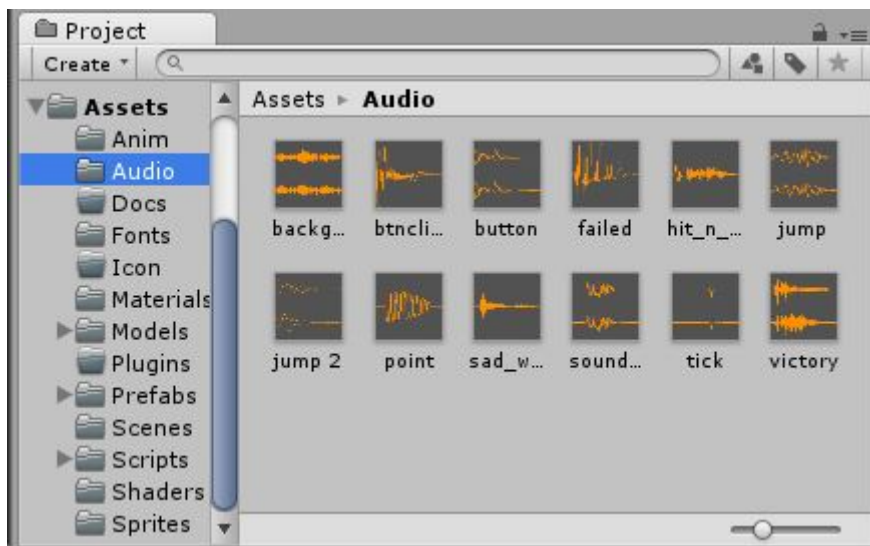

All fonts used in this game are free-to-use in commercial projects. Fonts are located under the *Fonts* folder together with appropriate license files.
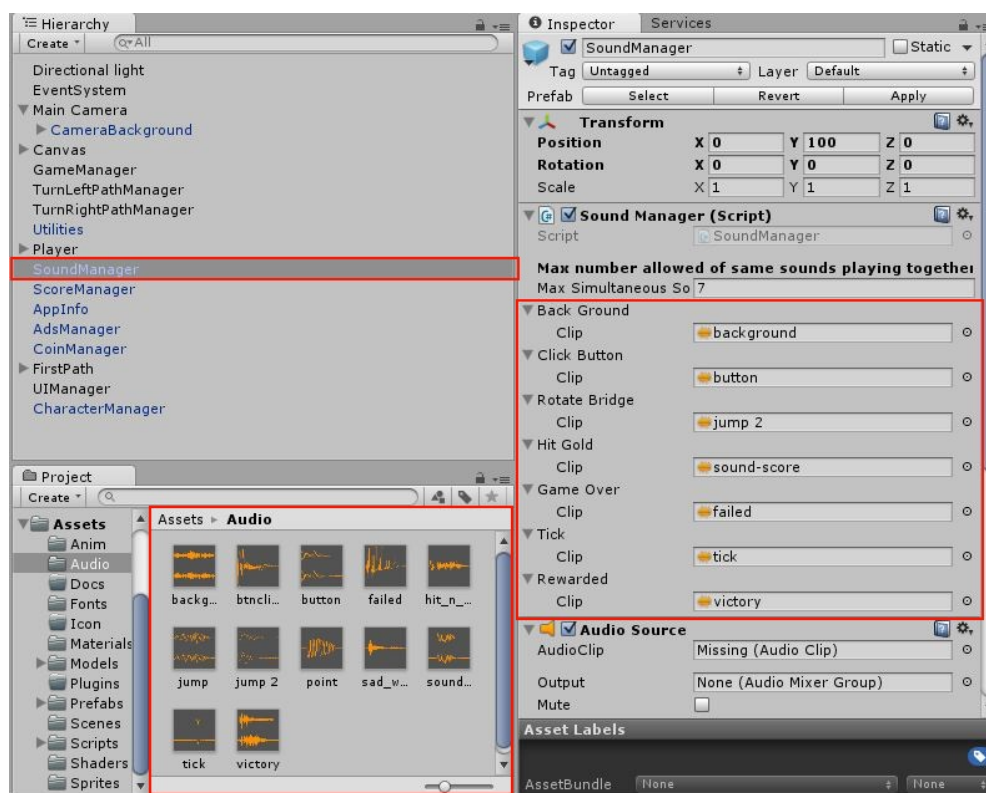
3.4 Sounds

All sounds included in this game are free-to-use in commercial projects and are located under the *Audio* folder.

This game features a *SoundManager* class to manage activities in game like playing music or mute/unmute sounds. If you want to replace sounds in this game, simply drag and drop new sounds to appropriate slots in the *SoundManager* component.

4 ENABLING PREMIUM FEATURES

To enable premium features of this template, you need to download and import Easy Mobile plugin from http://u3d.as/Dd2.

This section provides a guide on configuring each feature for your game. If you're not familiar with using Easy Mobile, it is strongly recommended that you read through its user guide to familiarize yourself with the plugin.

4.1 Before You Begin

- Make sure the *EnablePremiumFeatures* option in the *GameManager* object is checked.
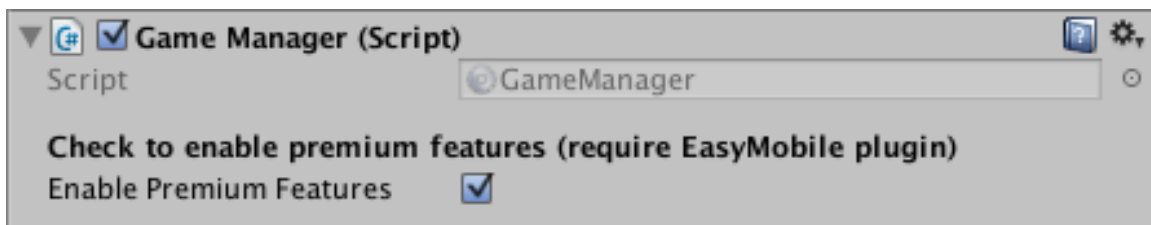
- Also make sure you add the EasyMobile prefab at folder *Assets/EasyMobile* to the *Main* scene. It is necessary for the plugin to function properly.

4.2 Advertising

4.2.1 Template-specific setup

This template contains a game object named *AdDisplayer* which is responsible for all ads displaying activities in the game. There you can configure how ads should be served in your game.
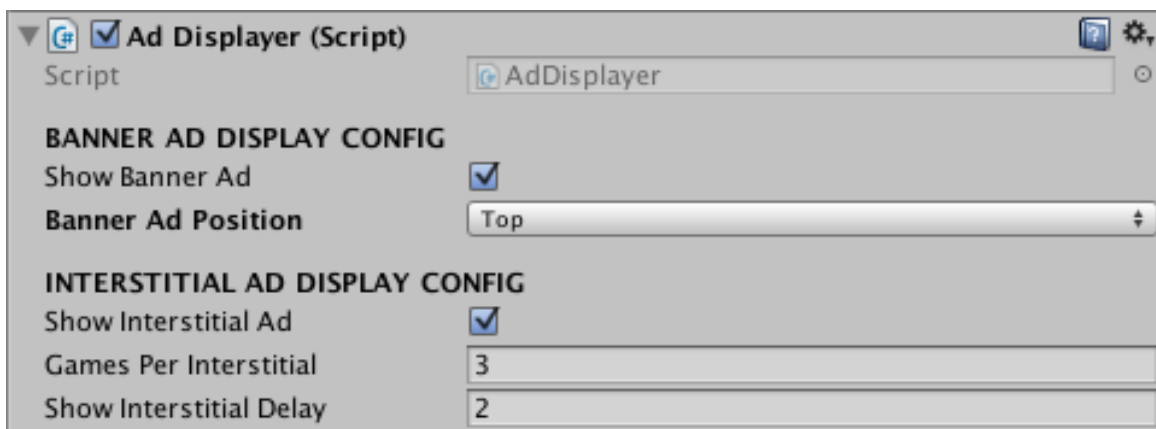
Banner ads are configured in the **BANNER AD DISPLAY CONFIG** section.

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

Interstitial ads are configured in the **INTERSTITIAL AD DISPLAY CONFIG**

section.

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

4.2.2 Easy Mobile setup

Open Easy Mobile's settings interface to start configuring its Advertising module (see its user guide for more information). With Easy Mobile you'll have support for AdMob, Chartboost, Heyzap (with mediation) and Unity Ads. You can use multiple ad networks at once and have different configurations for iOS and Android. Below is the settings interface of the Advertising module.

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 a *Restore Purchase* button as required on iOS.

In the hierarchy, there's a game object named *InAppPurchaser* which manages all the in-app purchasing activities in this game.
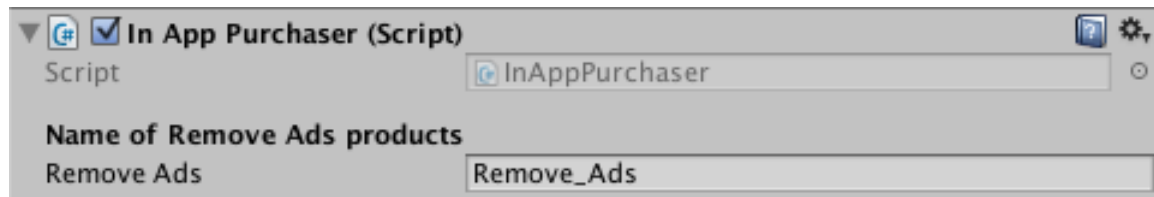
Here you can modify the name of the RemoveAds product if you want.

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 *RemoveAds* product name declared with Easy Mobile must match with the one you have in the aforementioned *InAppPurchaser* object. Also note that *Remove Ads* should be a non-consumable product.

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 an object named *AutoScoreReporter*. There you can change the leaderboard name or even disable automatic score reporting altogether.
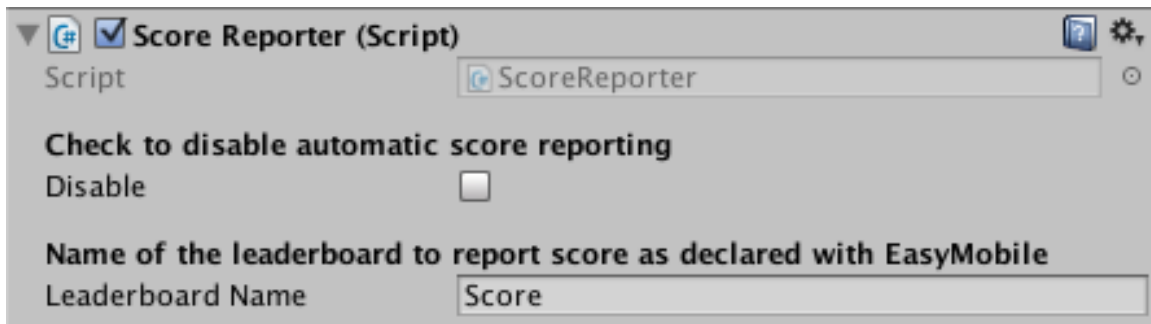
Achievements will be unlocked automatically when the user reaches a certain score. The achievement unlocking is handled by an object named *AutoAchievementUnlocker*. From this object, you can modify existing achievements and add or remove achievements. You can also disable the automatic achievement unlocking 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* objects. 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 the object named *ScreenshotSharer*.
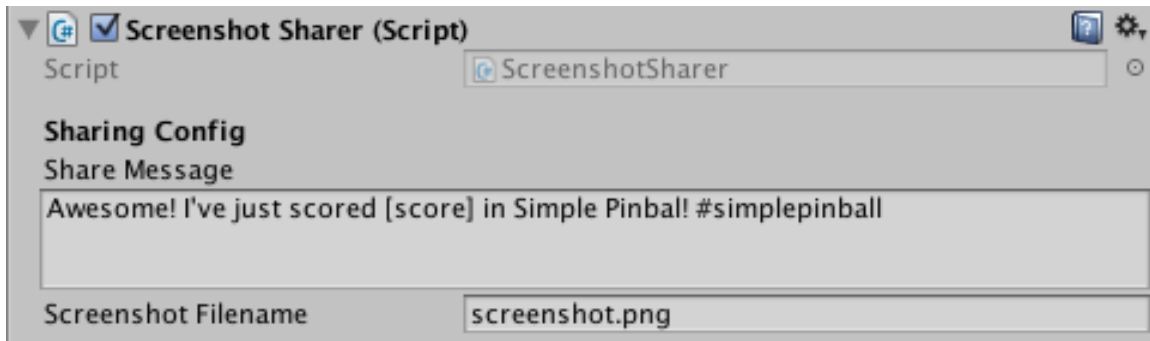
Here you can configure the sharing feature.

- *Share Message*: the default sharing message, note that [score] will be automatically replaced by actual score
- *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.

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

THANK YOU AND GOOD LUCK WITH YOUR GAMES!