



UNITY GAME TEMPLATE JUMPY SKY

USER GUIDE

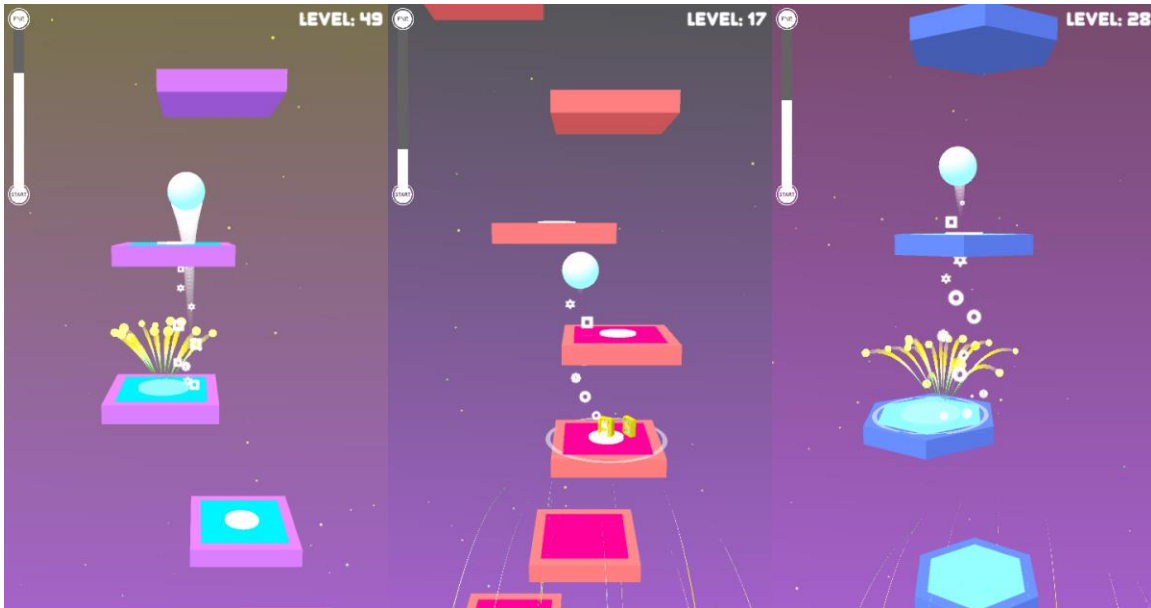


VERSION 1.0
CLAWBEAR GAMES
clawbeargames@gmail.com

Table of Contents

I. TEMPLATE INTRODUCTION	3
II. TEMPLATE SETUP.....	4
III. REQUIREMENT PACKAGES	7
IV. GAMEPLAY CUSTOMIZATION	8
1. INGAME MANAGER	8
2. PLAYER CONTROLLER.....	8
3. ITEM CONTROLLER	8
V. LEVEL CUSTOMIZATION.....	9
VI. SERVICES CONFIGURATION.....	13
1. SOUND MANAGER.....	13
2. COIN MANAGER	14
3. DAILY REWARD MANAGER.....	14
4. AD MANAGER	16
5. UNITY ADS CONTROLLER	18
6. ADMOB CONTROLLER.....	19
7. SHARE MANAGER	20
8. LEADERBOARD MANAGER	22
9. CUSTOMIZING UI	24
VII. CHARACTER MANAGERMENT	27

I. TEMPLATE INTRODUCTION



Jumpy Sky is an exciting endless level-base game in which you hold your finger on the screen and swipe left or right to control the ball. The goal is to make the player jump over higher platforms and reach the highest platform to complete the level, collect coins to unlocks new cool characters. The game is ready to release straight out of the box, and it can also be easily customized to make it even more engaging to your players. Supports for PC/Mac, iOS, Android, etc...!

This template is made with Unity C# and optimized for mobile devices. This template provides you a full-featured, ready-for-release game source code that you can customize and build your own game in no time!

Highlight features:

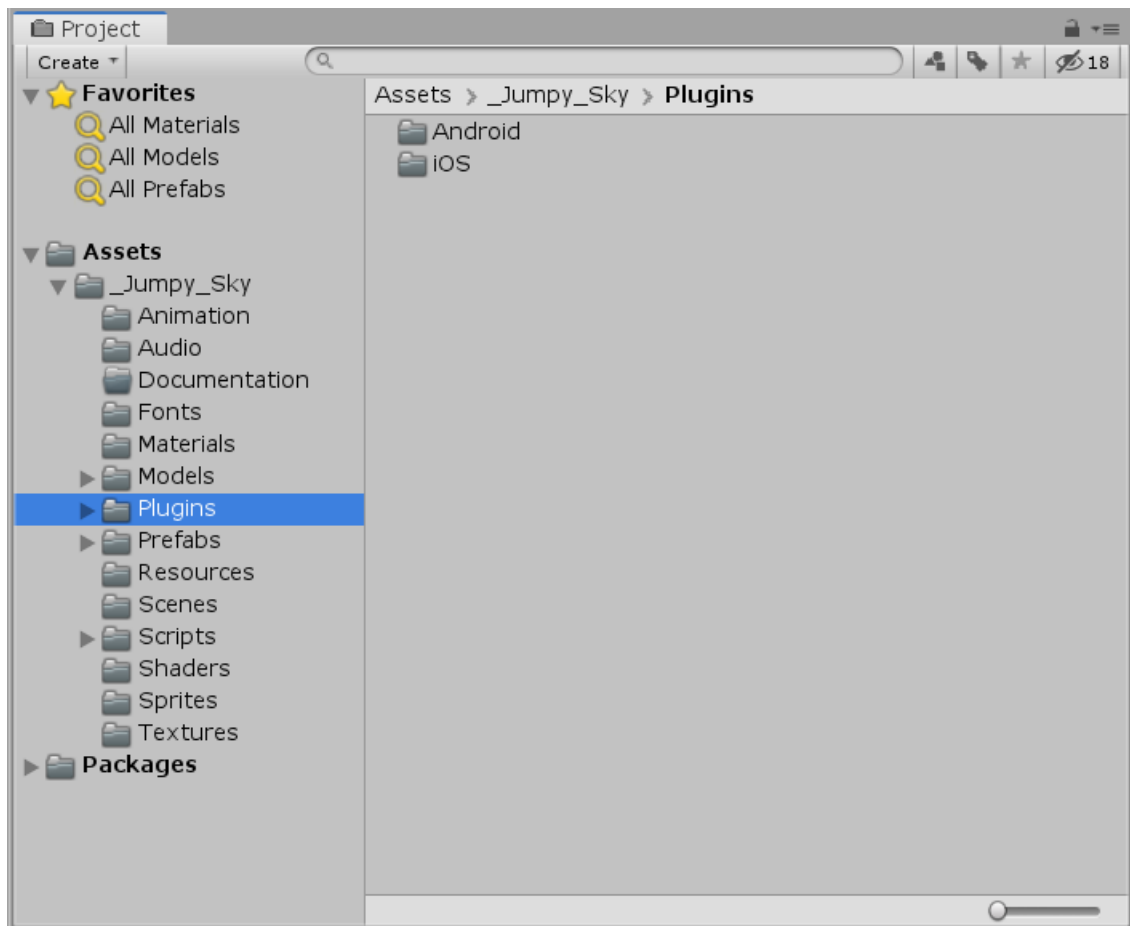
- ❖ Addictive one-touch gameplay
- ❖ Smooth control, eye-catching graphics, 60 FPS gameplay
- ❖ 100++ levels
- ❖ Very easy to add new levels with detailed instruction
- ❖ 9 days daily reward system for better retention
- ❖ Watch ad for coins
- ❖ 20+ characters ready to unlock

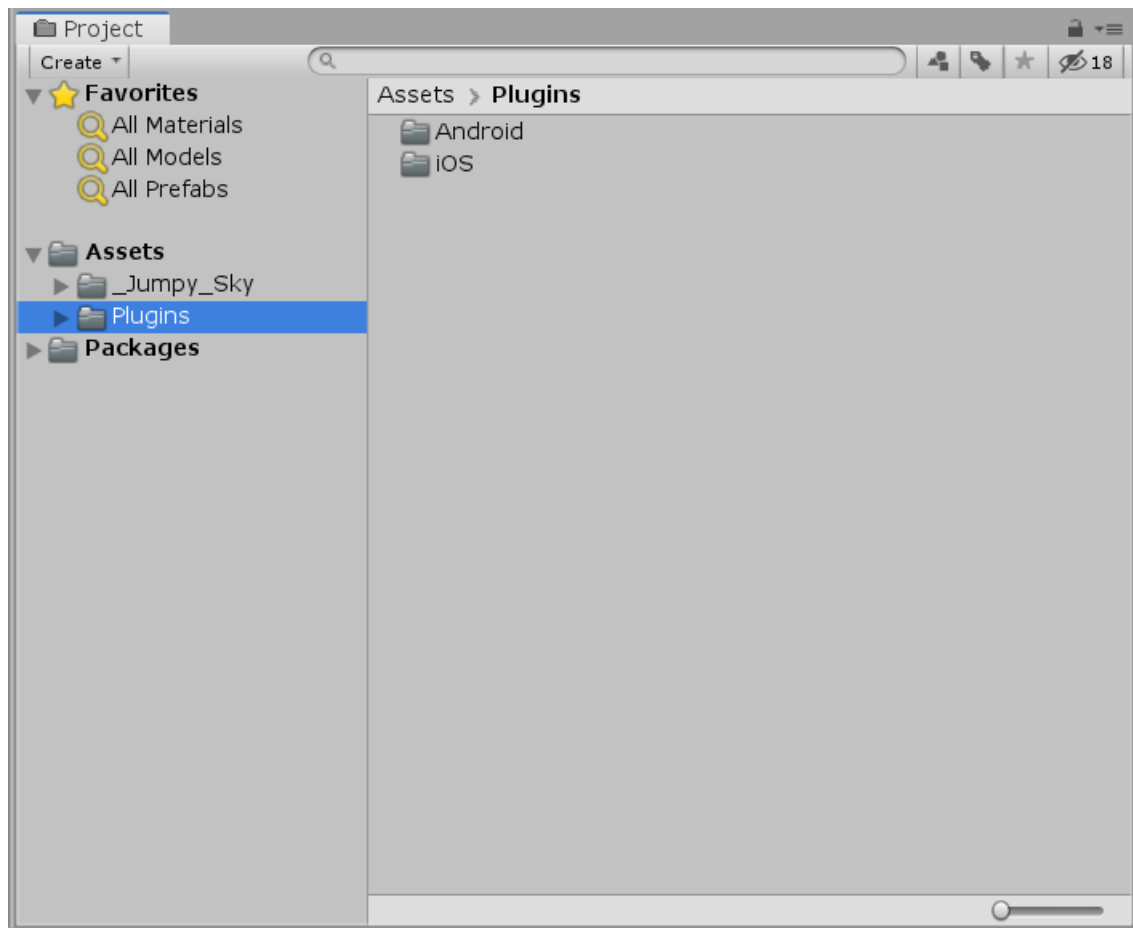
- ❖ Unlock new characters using coins
- ❖ Multiple ad networks: Admob and Unity Ads ready to use (banner, interstitial and rewarded video).
- ❖ Native share Android/iOS
- ❖ Facebook/Twitter share
- ❖ Commented C# code with detailed documentation
- ❖ Optimized for mobile
- ❖ Free-to-use assets (fonts, sounds, music, models, etc.)
- ❖ Ready to publish out-of-the-box

II. TEMPLATE SETUP

This template was designed for mobile (Android, iOS, Windows Phone...) so after imported the package to unity, you need to switch to Android or iOS, or Window Phone.

After that, go to Assets/_Jumpy_Sky and move the folder Plugins out of _Jumpy_Sky folder.





The template contains 4 scenes: Home, Loading, Ingame, Character under the path Assets/_ Jumpy_Sky/Scenes. You need to start from Home scene.

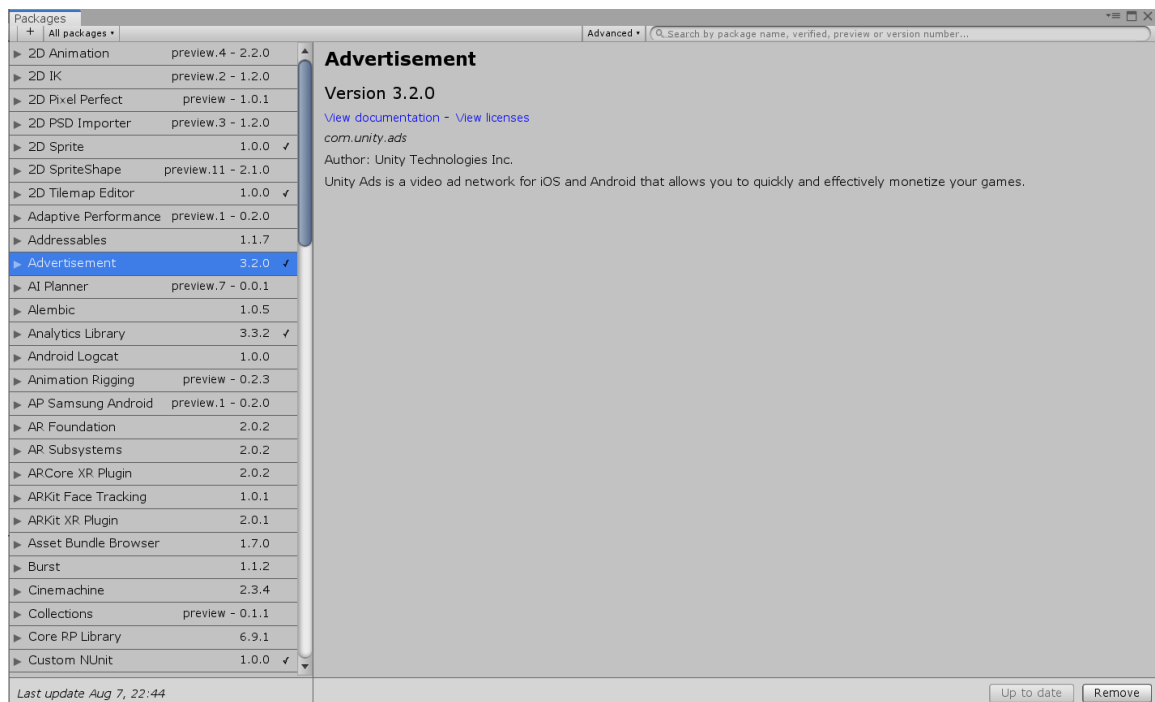
Scenes In Build	
<input checked="" type="checkbox"/> Snaky_Run_Run/Scenes/Home	0
<input checked="" type="checkbox"/> Snaky_Run_Run/Scenes/Character	1
<input checked="" type="checkbox"/> Snaky_Run_Run/Scenes/Loading	2
<input checked="" type="checkbox"/> Snaky_Run_Run/Scenes/InGame	3

Add Open Scenes

III. REQUIREMENT PACKAGES

When you open this template, at first you will see some errors in Console window, that's because this template requires some packages to run. You have to install these packages below to have this template run smoothly. Please follow these instructions:

- Open Package Manager by going to Window/Package Manager. If you don't see Package Manager button, then you have to re-install your Unity completely because it's the problem that Unity Technologies still not able to fix.



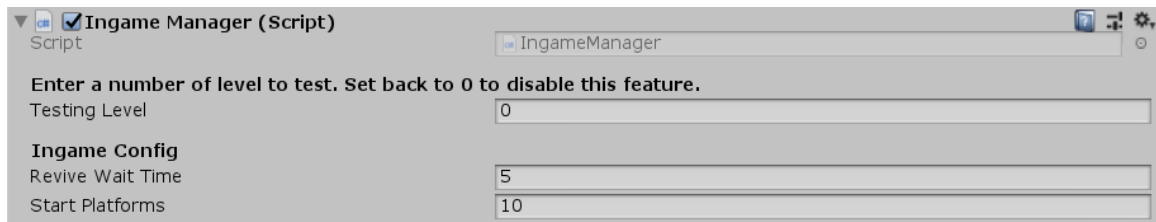
- These are packages you need to install:
 - ❖ Advertisement version 3.4.2 or higher.
 - ❖ Analytics Library version 3.2.2 or higher.
 - ❖ Text Mesh Pro version 1.3.0 or higher.

After you installed these packages, there's maybe some errors in Console window. If you see errors in Console window, go to File - > Build Setting -> Player Setting and change the Scripting Run Time Version to .Net 4.x Equivalent.

IV. GAMEPLAY CUSTOMIZATION

1. Ingame Manager

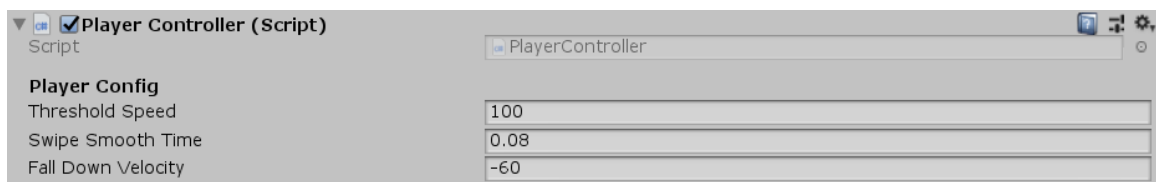
Most of important gameplay parameters can be configured within the IngameManager component which is attached to a game object also named IngameManager in the hierarchy. You can find IngameManager object in InGame scene.



- ❖ *Testing Level*: the level you need to test before building the game. If you set it to 0, the script will not load the testing level.
- ❖ *Revive Wait Time*: the delay time for revive feature last.
- ❖ *Start Platforms*: the first platforms will be created when the game start.

2. Player Controller

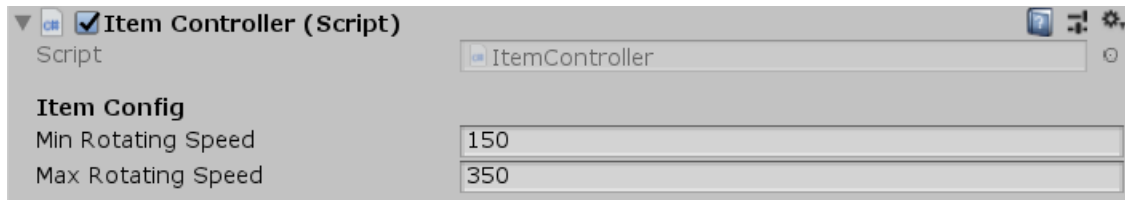
Most of important parameters of player can be configured within the PlayerController component which is attached to a game object named Player in the hierarchy.



- ❖ *Threshold Speed*: how fast the player moves on horizontal. The higher value, the faster player moving around.
- ❖ *Swipe Smooth Time*: the smooth time when swiping.
- ❖ *Fall Down Velocity*: the velocity when player fall down. The lower value, the faster player falls down.

3. Item Controller

Most of important parameters of item can be configured within the ItemController component. You can find all items under the path Assets/_Jumpy_Sky/Prefabs/Items.



- ❖ *Min Rotating Speed & Max Rotating Speed*: minimum and maximum rotating speed of the item.

V. LEVEL CUSTOMIZATION

This is a level-base game template and the goal of the game is you have to complete a level and start a new level, try to collect coins and avoid traps and obstacles. All the parameters of the level were shown on the inspector for you to adjust and modify. They are stored in IngameManager script and you can adjust all the levels in the inspector of IngameManager object which you can find it in Ingame scene.



As you can see, there's a list of 20 configuration parameters of levels, these parameters are already designed and ready to use. If you want to add new configs, just resize the list and hit Enter. Now I will show you the parameters of each level.

Levels Config

- ▼ List Level Config
 - Size: 21
 - ▼ Element 0
 - Level Number Configuration**
 - Min Level: 1
 - Max Level: 5
 - Background Colors Configuration**
 - Background Top Color: [Cyan color bar]
 - Background Bottom Color: [Blue color bar]
 - Background Music Configuration**
 - ▼ Music Clip
 - Audio Clip: [background_1]
 - Platform Colors Configuration**
 - Jump Steps: 12
 - List Platform Colors Config
 - List Platform Configuration**
 - List Platform Config

Level Number Configuration: this section is about the configuration of the level number.

Level Number Configuration

- Min Level: 1
- Max Level: 5

- ❖ *Min Level & Max Level:* the minimum and maximum level of this config. All the level between this range will use these parameters below to create the level.

Background Colors Configuration: this section is about the configuration of the top and bottom colors of the background.

Background Colors Configuration

- Background Top Color: [Cyan color bar]
- Background Bottom Color: [Blue color bar]

- ❖ *Background Top Color:* the top color of the background.
- ❖ *Background Bottom Color:* the bottom color of the background.

The color of the background will be blended from top bottom color to top color.

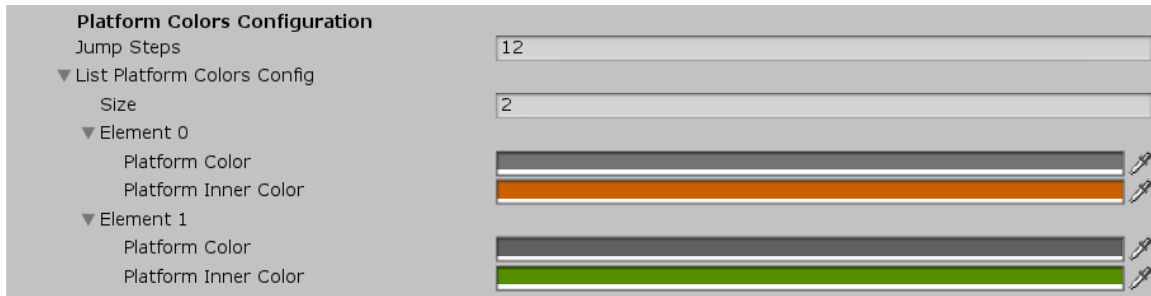
Background Music Configuration: this section is about the configuration of the background music.

Background Music Configuration

- ▼ Music Clip
 - Audio Clip: [background_1]

❖ *Audio Clip*: the audio clip to play as background music.

Platform Colors Configuration: this section is about the configuration of the colors of each platform.



- ❖ *Jump Steps*: how many jump steps to change the colors of all platforms.
- ❖ *List Platform Colors Config*: the list of configured colors for all platforms.
 - *Platform Color*: the main color of the platform.
 - *Platform Inner Color*: the color of inner side of the platform.

Each platform has 3 part, the main object, the inner object and the center object. The color of the center object always white, *Platform Color* is the color of the main object, *Platform Inner Color* is the color of the inner object.

The color of all platform will change from the current color to the next color in the list every time player reached *Jump Steps*.

Platform Colors Configuration: this section is about the configuration of the number of platform will be created in this level.

List Platform Configuration

▼ List Platform Config

Size

▼ Element 0

Platform Type And Size Configuration

Platform Type

Platform Size

Platform Number Configuration

Min Platform

Max Platform

Platform Distance Configuration

Platform Distance

XAxis Deviation Configuration

Min X Axis Deviation

Max X Axis Deviation

Moving Parameters Configuration

Moving Platform Frequency

Moving Left Distance

Moving Right Distance

Moving Speed

Lerp Type

Coin Number Configuration

Coin Frequency

Min Coin Number

Max Coin Number

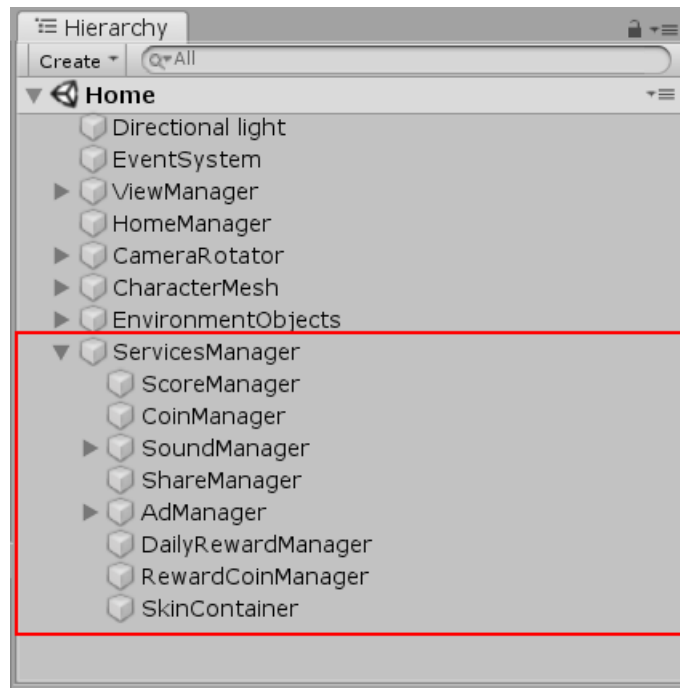
- ❖ *Platform Type*: the type of the platform. Currently, this template has 5 platform's type: SQUARE, HEXAGON, HEPTAGON, OCTAGON and CIRCLE.
- ❖ *Platform Size*: the size of the platform. Currently, this template has 3 platform's size: BIG, NORMAL and SMALL.
- ❖ *Min Platform & Max Platform*: minimum and maximum platform number. The actual number will be randomized between these two values.
- ❖ *Platform Distance*: the y distance between 2 platforms.
- ❖ *Min X Axis Deviation & Max X Axis Deviation*: minimum and maximum x axis deviation when creating platforms. The x axis of each platform will be randomized between these two values.
- ❖ *Moving Platform Frequency*: the frequency to play moving action of each platform.
- ❖ *Moving Left Distance & Moving Right Distance*: the left and right distance of each platform. Base on this value and the x axis deviation, the platform will generate the limit left and limit right, then moving from left to right and right to left, repeatedly.
- ❖ *Moving Speed*: moving speed of the platform.
- ❖ *Lerp Type*: the lerp type when moving the platform. You can see all the lerp type [here](#).
- ❖ *Coin Frequency*: the frequency to create coin on each platform.
- ❖ *Min Coin Number & Max Coin Number*: minimum and maximum coin

will be created on each platform. The actual value will be randomized between these two values.

Currently, *List Platform Config* has only 1 element, but if you want to add more, just increase the size of the list and config the parameters as you want.

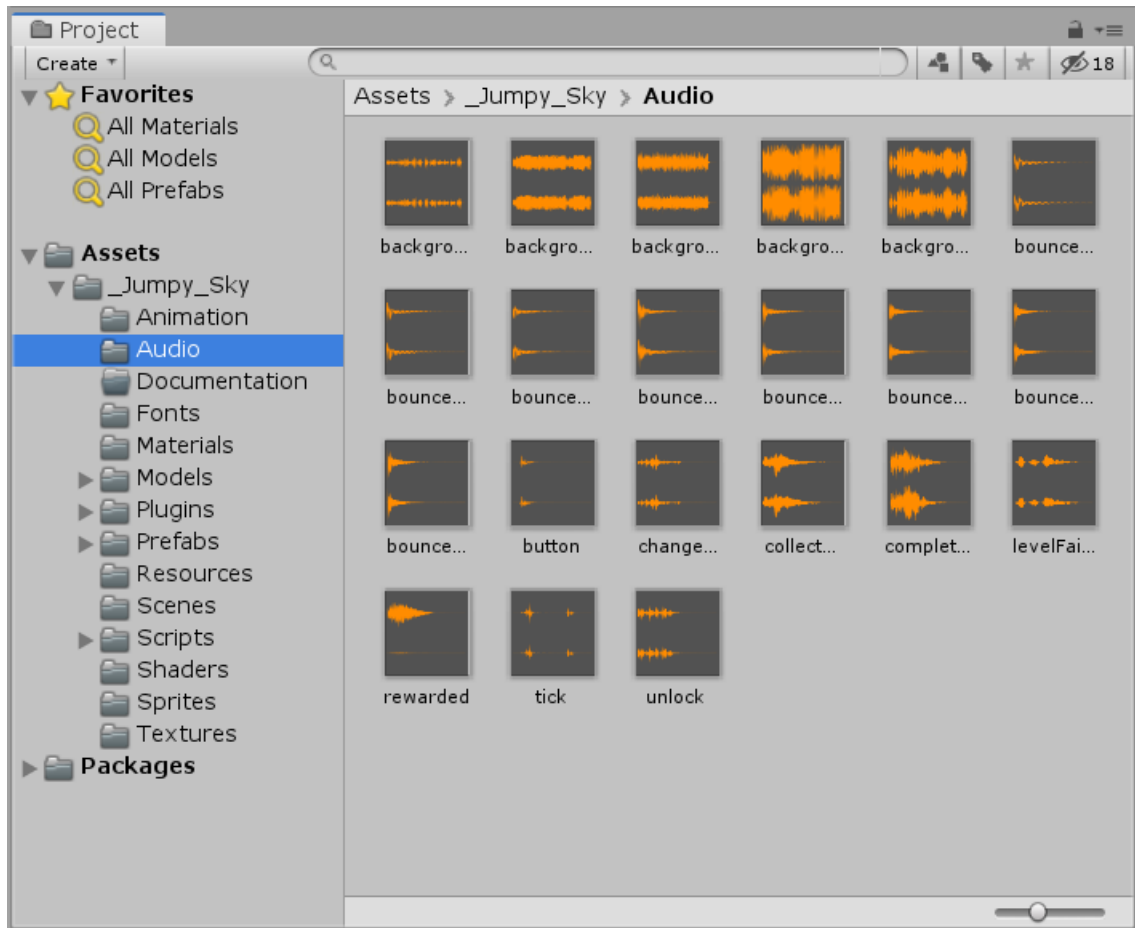
VI. SERVICES CONFIGURATION

This template comes with many features like high score system, collected coins and total coins, sounds, sharing, multiple ads and daily reward. You can find all the scripts for these features under *ServicesManager* object in Home scene.



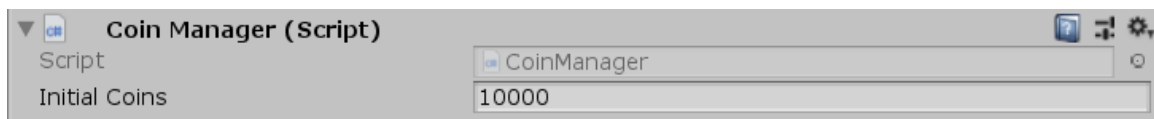
1. Sound Manager

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



2. Coin Manager

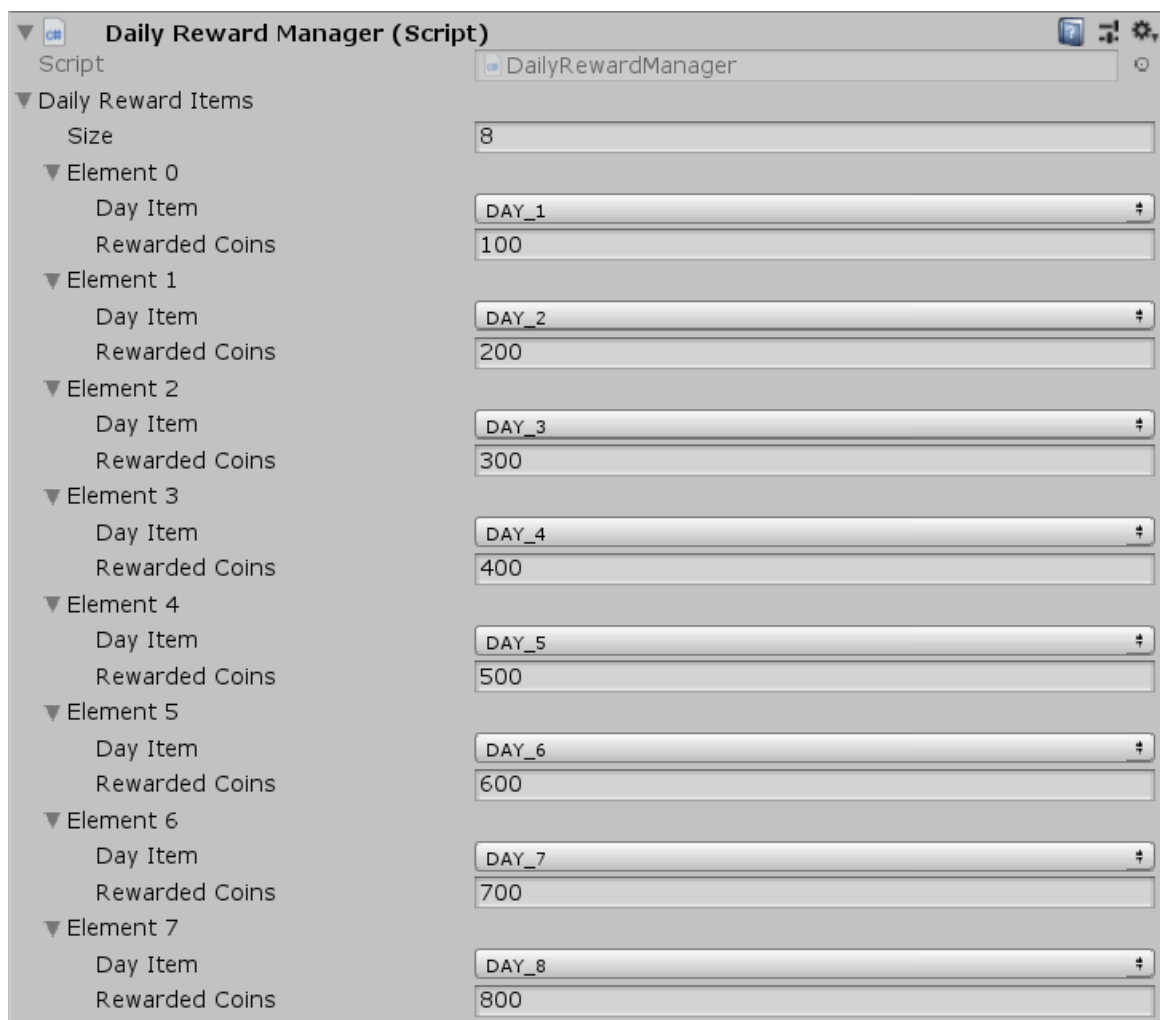
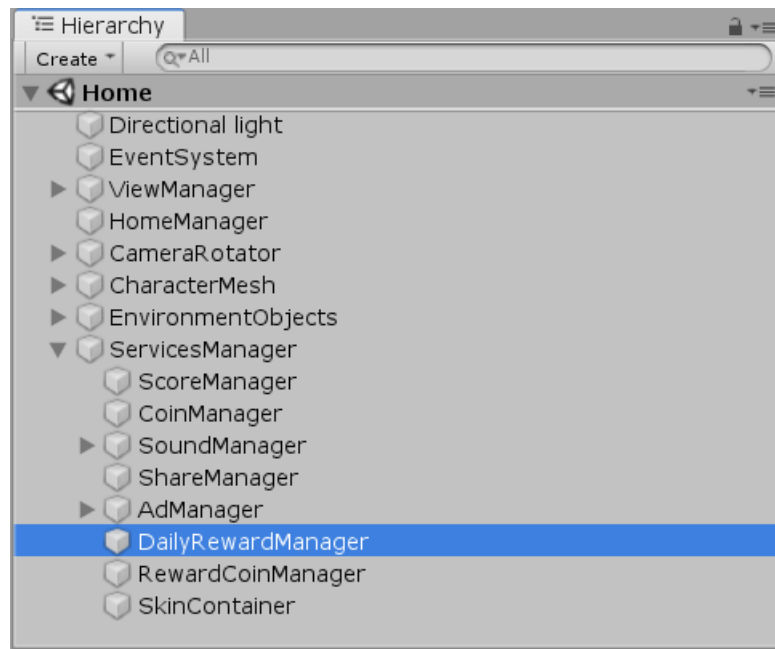
Coin is the currency in the game that player collect to unlock new characters. Coin is manage by CoinManager component.



❖ *Initial Coins*: the initial coins that player will have when the game start at the first time.

3. Daily Reward Manager

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 *DailyRewardManager* object in the hierarchy.

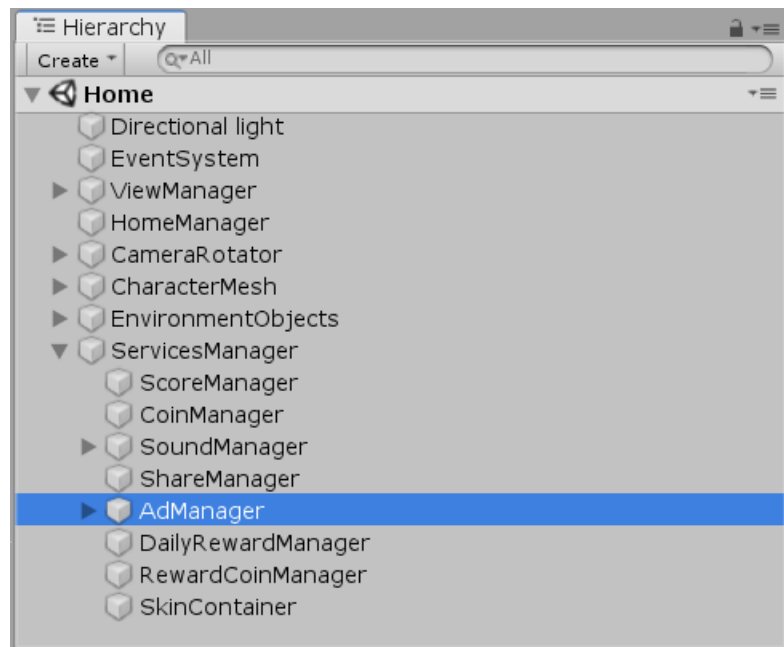


- ❖ *Daily Reward Items*: this array allows you config the amount of rewarded coins for each day.
 - *Day Item*: the day of the reward.
 - *Rewarded Coins*: how many coins player will get at that day.

***IMPORTANT:** Currently, the daily reward system was designed for only 8 days, after player get all the reward (claimed 8 rewards), the daily reward system will reset to day 1. DO NOT ADD MORE ELEMENT TO THE ARRAY, IT MAY CAUSE UNEXPECTED ERRORS.

4. Ad Manager

The AdManager object in hierarchy of scene Home contains AdManager component, in which you can customize which type of ads you want to use and how you want to show the ads.



▼ **Ad Manager (Script)**

Script:

Banner Ad config

Banner Ad Type:

Showing Banner Ad Delay:

Interstitial Ad Config

▼ List Show Interstitial Ad Config

Size:

▼ Element 0

Game State For Showing Ad:

Game State Count For Showing Ad:

Show Ad Delay:

▼ List Interstitial Ad Type

Size:

Element 0:

Element 1:

▼ Element 1

Game State For Showing Ad:

Game State Count For Showing Ad:

Show Ad Delay:

▼ List Interstitial Ad Type

Size:

Element 0:

Element 1:

Rewarded Video Ad Config

Showing Rewarded Video Ad Delay:

▼ List Rewarded Ad Type

Size:

Element 0:

Rewarded Coins Config

Min Rewarded Coins:

Max Rewarded Coins:

Reward Delay:

❖ **Banner Ad Config:** this is the section where you can control which type of banner ad you want to show. Currently, the template support for 2 banner ad type: Admob ad Unity.

- **Banner Ad Type:** the type of banner ad you want to show. (Unity Ads just released Unity Monetization 3.0 which included Banner Ad, but seems like its still unstable, so I recommend using Admob for banner ad).
- **Showing Banner Ad Delay:** delay time to show banner ad.

❖ **Interstitial Ad Config:** this is the section where you can control which type of interstitial ad you want to show and how you want to

how it. Currently, the template support for 2 interstitial ad type: Admob and Unity Ads.

List Show Interstitial Ad Config: this is the list contains all the config parameters of showing interstitial ad.

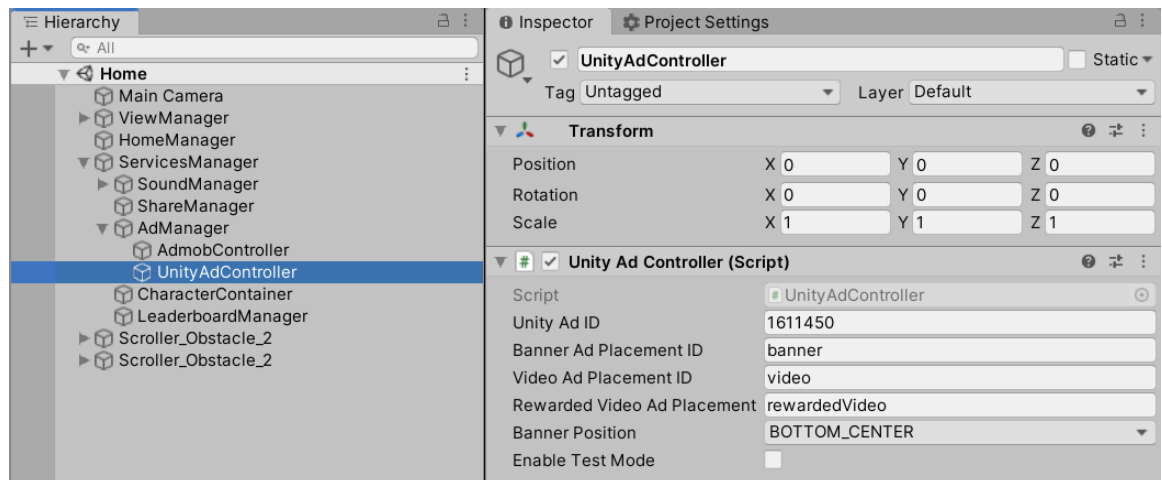
- *Game State For Showing Ad*: the game state you want to show the ad.
- *Game State Count For Showing Ad*: the number of game state that the game go through to show ad. Example: if the value is 2 and *Game State For Show Ad* is Game Over, that mean the ad will show up after 2 times of game over.
- *Show Ad Delay*: the delay for showing the ad.
- *List Interstitial Ad Type*: the list of interstitial ad type. The first item of this list will be the first priority of interstitial ad. Example: in the image you can see 2 items: ADMOB and UNITY. ADMOB is the first item so, *Ad Manager* will load rewarded video ad of Admob first, if there's no Admob's rewarded video to load, then *Ad Manger* will continue to load Unity interstitial ad.

❖ *Rewarded Video Ad Config*: this is the section where you can control which type of rewarded video ad you want to show and how you want to how it. Currently, the template support for 2 rewarded video ad type: Admob and Unity Ads.

- *Showing Rewarded Video Ad Delay*: the delay time for showing rewarded video ad.
- *List Rewarded Ad Type*: the list of rewarded video ad type. The first item of this list will be the first priority of interstitial ad. Example: in the image you can see 2 items: ADMOB and UNITY. ADMOB is the first item so, *Ad Manager* will load rewarded video ad of Admob first, if there's no Admob's rewarded video to load, then *Ad Manger* will continue to load Unity rewarded video ad.

5. Unity Ads Controller

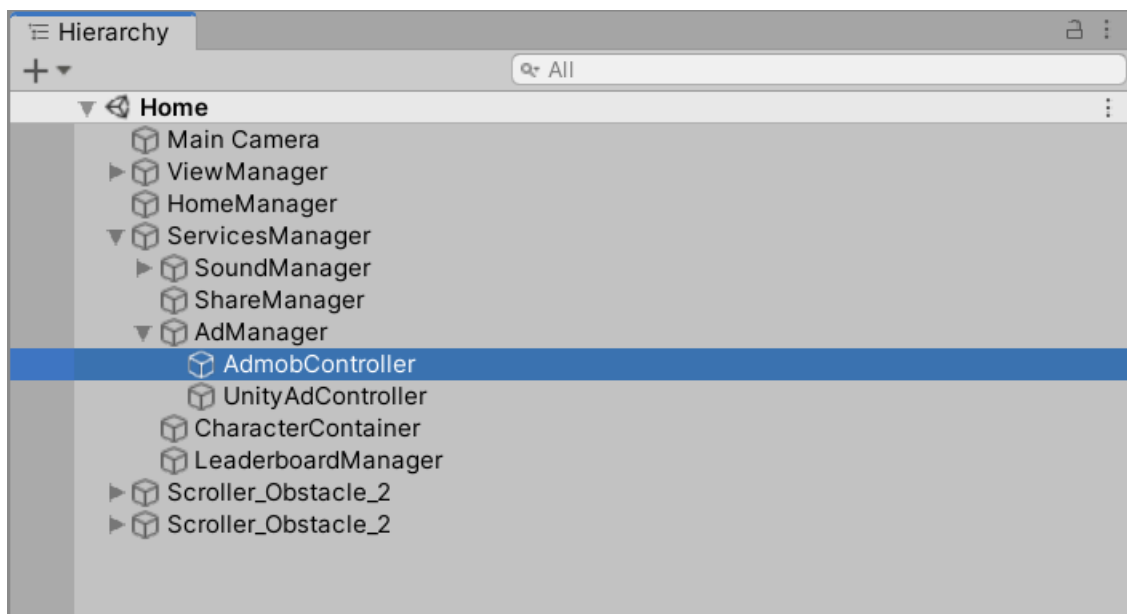
The template already had unity ads sdk included, so you don't need to import anything else to use unity ads, just put your project id and your placement ids and you will good to go.

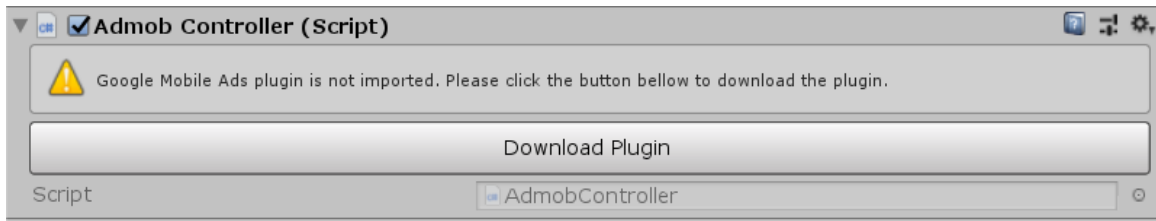


- ❖ *Unity Ad ID*: the id of your unity ad project. You can find all of your ad projects [here](#).
- ❖ *Video Ad Placement ID*: the video ad placement id of your ad project.
- ❖ *Rewarded Video Ad Placement ID*: the rewarded video ad placement id of your ad project.

6. Admob Controller

The AdmobManager object in hierarchy of scene Home contains AdmobController component, in which you can customize parameters like admob id, ad units...





As you can see in the image, there's no option to change ads units because the template did not have google mobile ads plugin included, so if you want to use Admob, please click to the Download Plugin button, download the latest version of google mobile ads and import it into the project.

After imported the plugin, select AdManager object in the hierarchy, wait for few seconds for the project rebuild and you will see this.

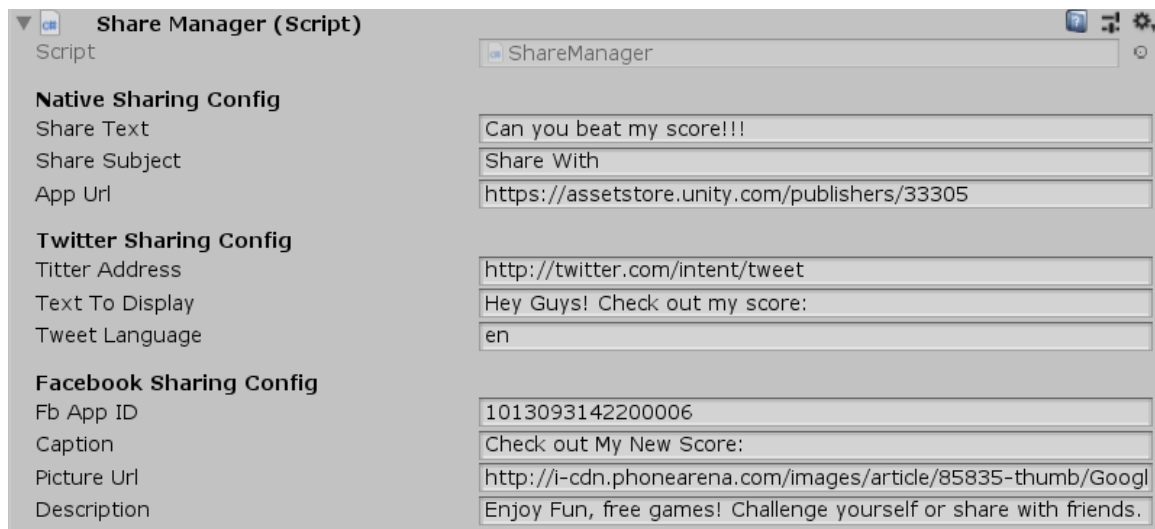
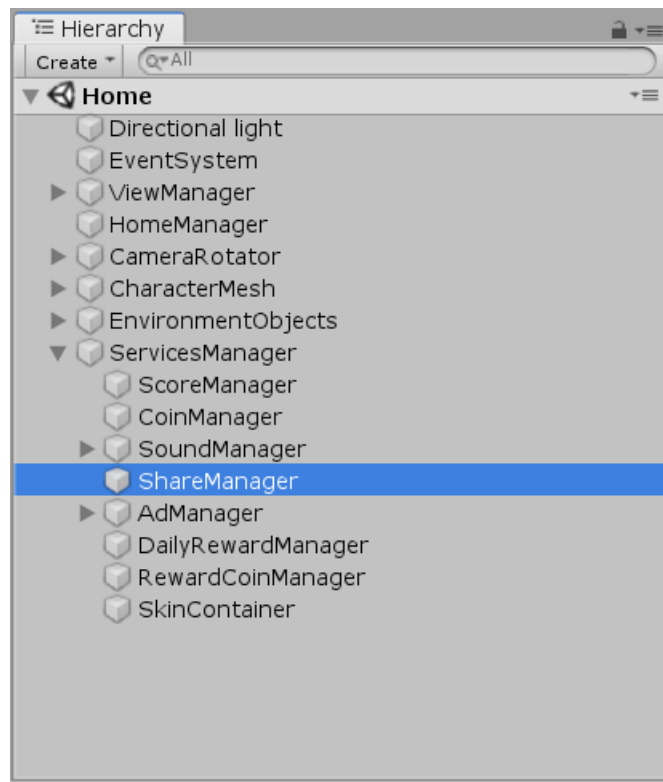


Now you can config admob id and all the ad units as you want. Currently, the platform using is iOS, that why all the ids in the image showing for iOS, of course it will show the ids for Android when you switch the build platform to Android.

- ❖ *IOS Admob App Id*: the android id of your admob app.
- ❖ *IOS Banner Id*: the banner ad unit of your admob app.
- ❖ *Banner Position*: the position of the banner ad.
- ❖ *IOS Interstitial Id*: the interstitial ad unit of your admob app.
- ❖ *IOS Rewarded Base Video Id*: the rewarded ad unit of your admob account.

7. Share Manager

All information for sharing feature can be config in ShareManager game object. It contains the information likes shreenshot's name, text, subject and url... You can config these features from the ShareManager object in the hierarchy.



Native Sharing Config:

- ❖ *Screenshot Name*: the name of screenshot for sharing feature.
- ❖ *Share Text*: the text for sharing feature.
- ❖ *Share Subject*: the subject for sharing feature.
- ❖ *App Url*: the url of the app (Google Play on Android and App Store on IOS).

Twitter Sharing Config:

- ❖ *Twitter Address*: address of your twitter account.

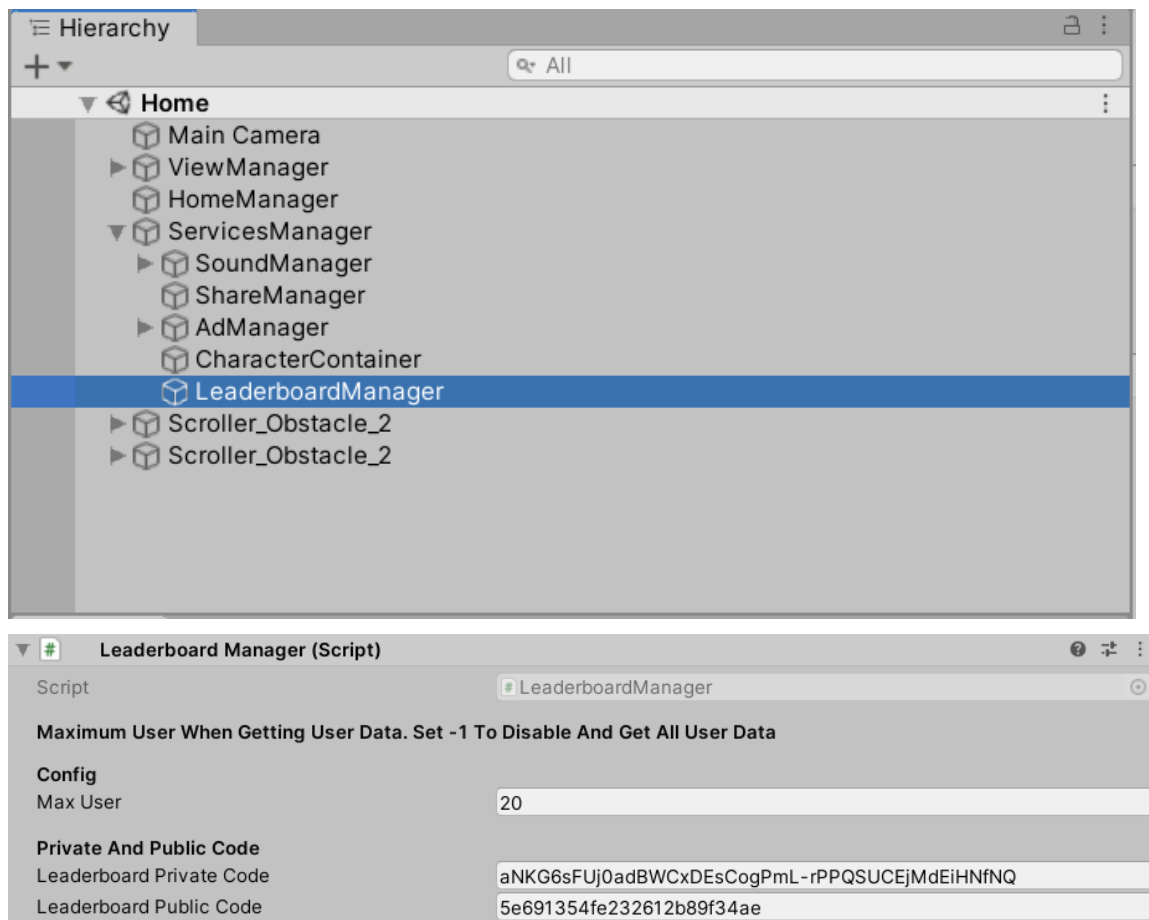
- ❖ *Text To Display*: the text you want to display in the tweet.
- ❖ *Tweet Language*: language of the tweet you want to share.

Facebook Sharing Config:

- ❖ *Fb App ID*: the id of your facebook app.
- ❖ *Caption*: the caption in your status.
- ❖ *Picture Url*: url of the picture you want to share. If you don't want to share pictures or just don't have any picture's url to share, leave this field empty.
- ❖ *Description*: the description you want to share.

8. Leaderboard Manager

All information for leaderboard feature can be config in LeaderboardManager game object. It contains 2 parameters for you to config the leaderboard. Currently, we used Dreamlo to handle leaderboard feature.



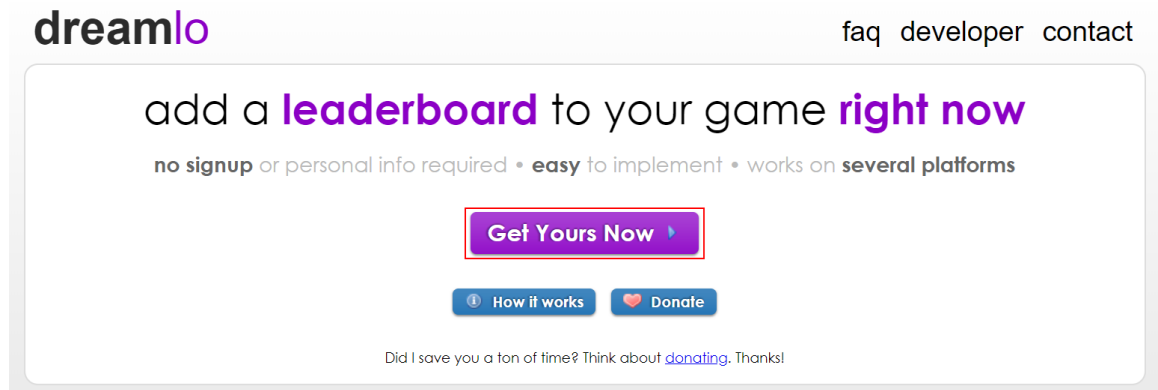
- ❖ *Max User*: the maximum user show in the leaderboard. If you set it to -1, it will show all the user that played the game and report to leaderboard table. We suggest you to limit it to 50 or 100, because

if your game have 1000 downloads, which mean you have 1000 user in the leaderboard and when you load it all to the leaderboard view, the game will become laggy.

- ❖ *Leaderboard Private Code*: the leaderboard private code of Dreamlo.
- ❖ *Leaderboard Public Code*: the leaderboard public code of Dreamlo.

Now we will show you how to get the private and public code from Dreamlo.

First, you need to go to [Dreamlo.com](https://dreamlo.com), at the top of the front page, click to Get Yours Now button.

The screenshot shows the top section of the Dreamlo website. At the top left is the 'dreamlo' logo in a bold, sans-serif font. To its right are three links: 'faq', 'developer', and 'contact'. Below the logo is a large white banner with rounded corners. Inside the banner, the text 'add a **leaderboard** to your game **right now**' is displayed in a large, bold font. Below this, in a smaller font, it says 'no signup or personal info required • easy to implement • works on several platforms'. In the center of the banner is a large purple button with the text 'Get Yours Now' and a small right-pointing arrow. Below the button are two smaller blue buttons: 'How it works' with an information icon and 'Donate' with a heart icon. At the bottom of the banner, in a very small font, is the text 'Did I save you a ton of time? Think about donating. Thanks!'.

dreamlo [faq](#) [developer](#) [contact](#)

add a **leaderboard** to your game **right now**

no signup or personal info required • easy to implement • works on several platforms

Get Yours Now

[How it works](#) [Donate](#)

Did I save you a ton of time? Think about [donating](#). Thanks!

After that, you will be direct to a page that you can get your private and public code.

Here is your **private** url. Copy and paste this somewhere.

Do not tell anyone about this link.

http://dreamlo.com/lb/2_qIYeHSOkCtoGpQU2ENegN6JOfe5uUUCf46IhWjJ3eQ

WebGL builds hosted at itch.io and other services as well as some versions of **Android** may need SSL to work!

Want to use **https (SSL)**? [Donate \\$5 or more](#) and let me know.

Want to store **more than 1000 scores**? [Contact](#) me.

(If you have a limit of 1000 scores and another score comes in, the lowest will get bumped out.)

You copy and pasted that somewhere and are never going to tell anyone right?

You can not have an asterisk * character in your URL, scores, usernames, etc.

Your Leaderboard

Highest 1000 scores.
Your leaderboard is empty.

[Remove All Scores](#)

Codes for Unity Example

Here are just your public and private code so that you can cut and paste them into the sample code for Unity.

Private Code (It's long, get all of it!)

2_qIYeHSOkCtoGpQU2ENegN6JOfe

Public Code

5e7c0020fe232612b8e01d54

Adding and deleting scores

Changes and updates to your leaderboard are made through simple [http get requests](#) using your **private** url.

A player named **Carmine** got a score of **100**. If the same name is added twice, we use the higher score.

http://dreamlo.com/lb/2_qIYeHSOkCtoGpQU2ENegN6JOfe5uUUCf46IhWjJ3eQ/add/Carmine/100

A player named **Carmine** got a score of **1000** in **90 seconds**.

http://dreamlo.com/lb/2_qIYeHSOkCtoGpQU2ENegN6JOfe5uUUCf46IhWjJ3eQ/add/Carmine/1000/90

A player named **Carmine** got a score of **1000** in **90 seconds** and is **Awesome**.

http://dreamlo.com/lb/2_qIYeHSOkCtoGpQU2ENegN6JOfe5uUUCf46IhWjJ3eQ/add/Carmine/1000/90/Awesc

Delete **Carmine's** score

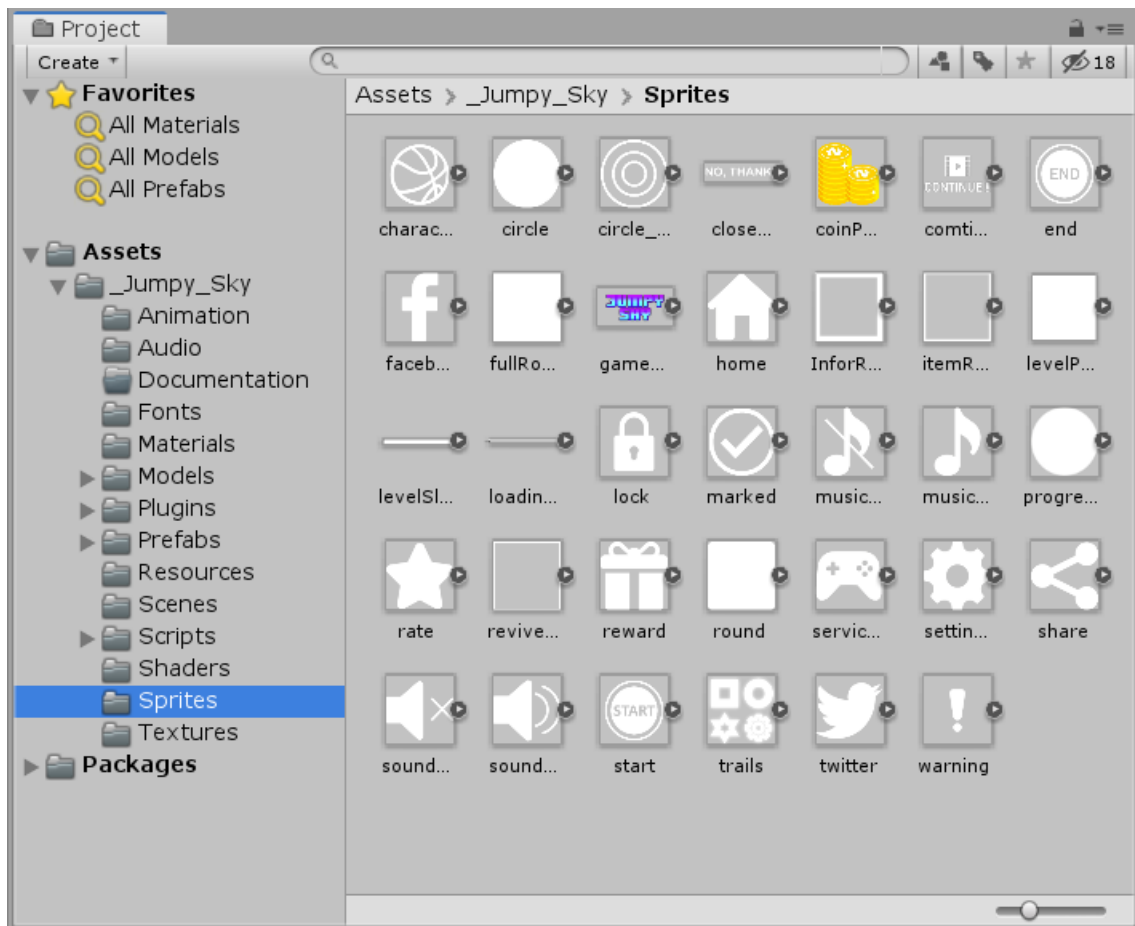
http://dreamlo.com/lb/2_qIYeHSOkCtoGpQU2ENegN6JOfe5uUUCf46IhWjJ3eQ/delete/Carmine

Clear **all** scores

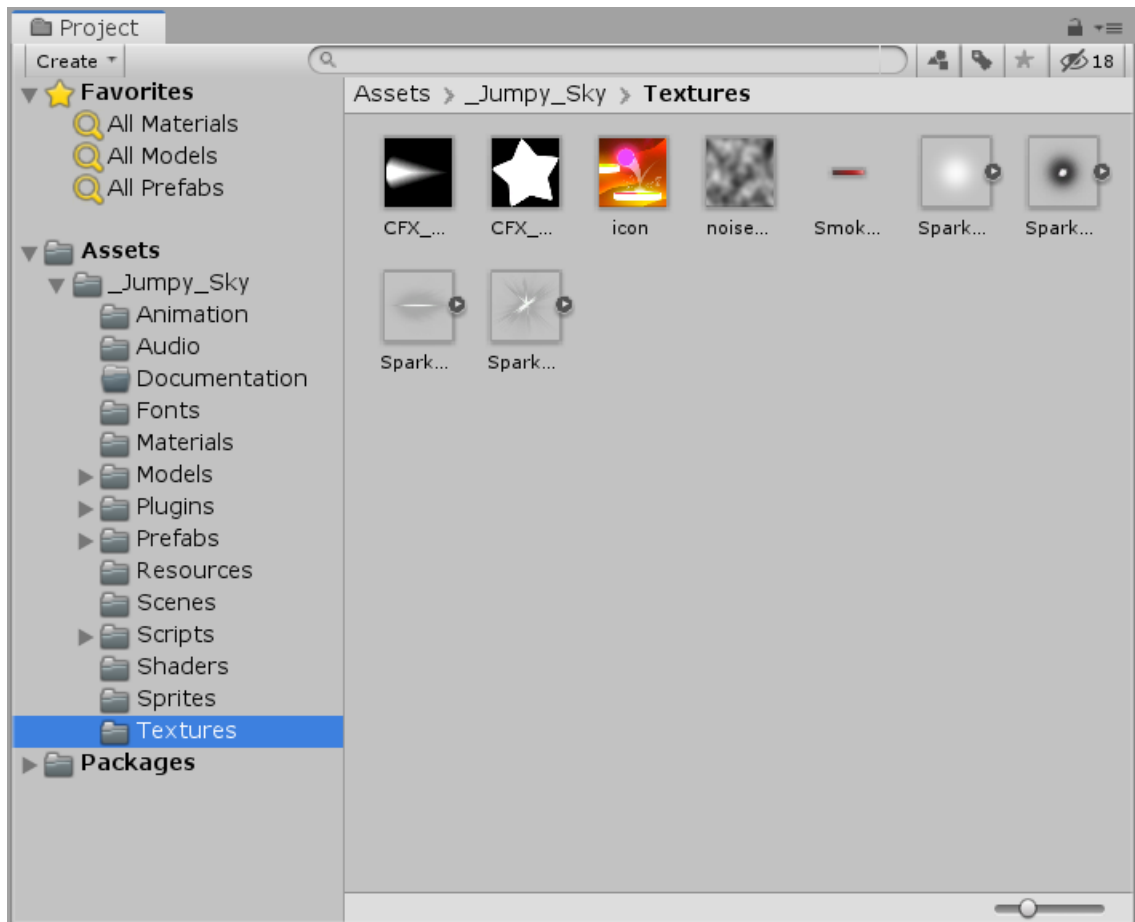
Copy 2 codes and paste it on LeaderboardManager component.
That's it.

9. Customizing UI

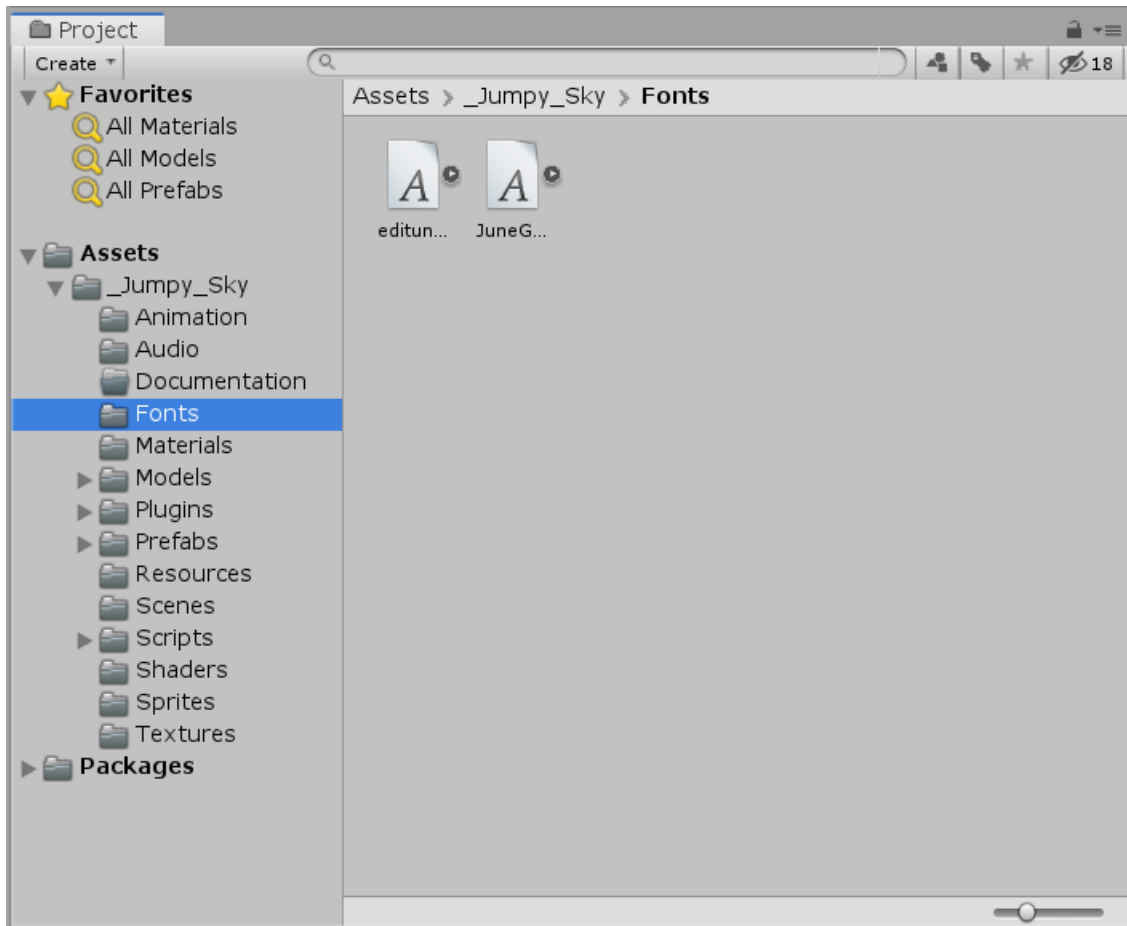
All sprites used in this game (for buttons and other UI components) are located under the path *Assets/_Jumpy_Sky/Sprites* folder. You can replace them with your own sprites to modify the images as you like.



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



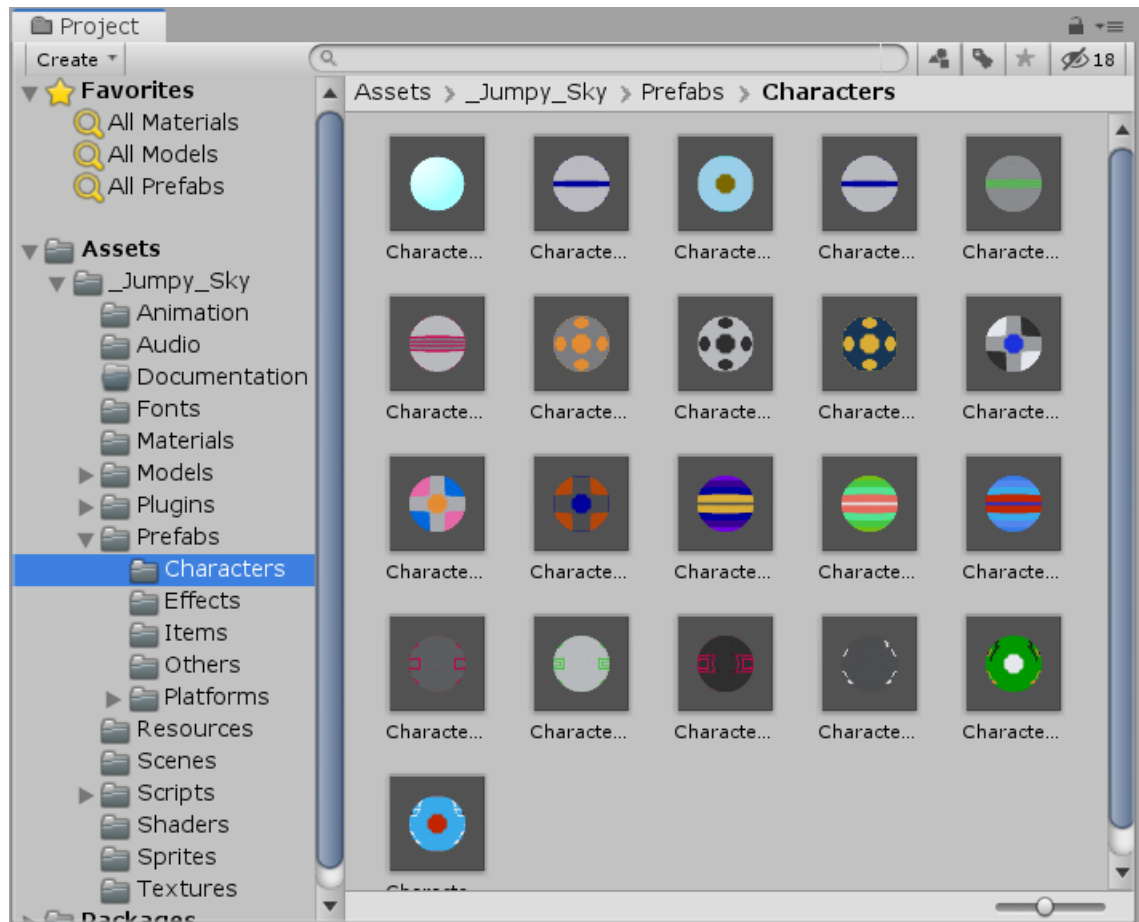
All fonts used in this game are free-to-use in commercial projects. Fonts are located under the path *Assets/_Jumpy_Sky/Fonts* folder together with appropriate license files.



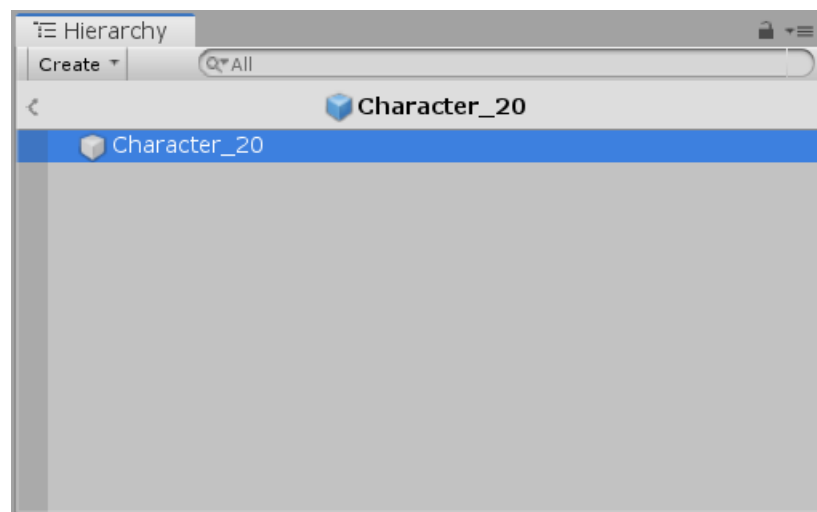
VII. CHARACTER MANAGERMENT

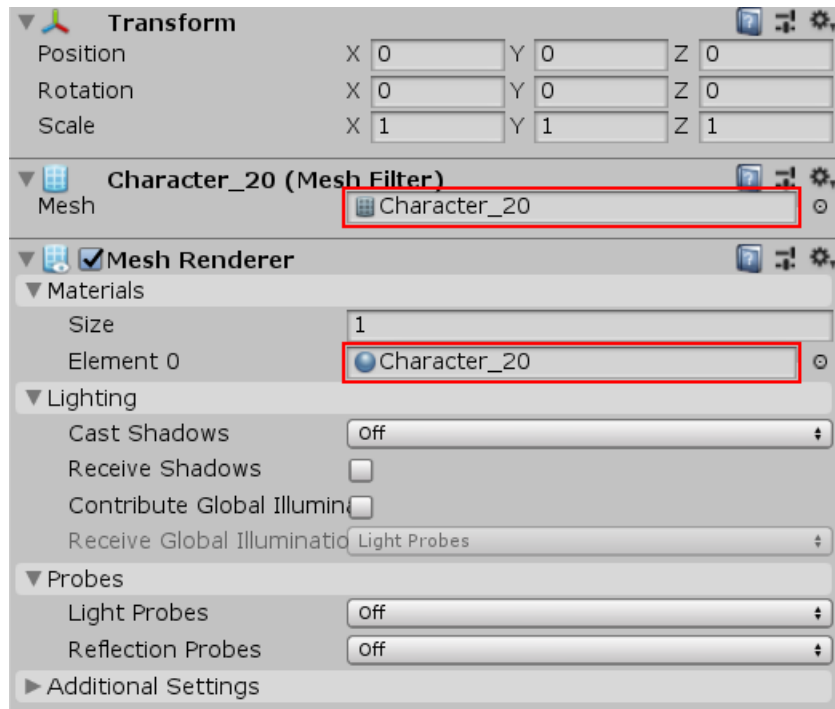
This template comes with built-in 20 characters ready to unlock, each character has only one mesh. You can find all the skins under the path *Assets/_Jumpy_Sky/Prefabs/Characters*. Here's the step to add more characters.

- Create a model, change the pivot of it to center (x: center, y: center, z: center). You can use any software out there like Maya, Zbrush, Blender, Magical Voxel...The software I used to create these models is Qubicle.
- Navigate to *Assets/_Jumpy_Sky/Prefabs/Characters* and duplicate one of the available car prefabs.

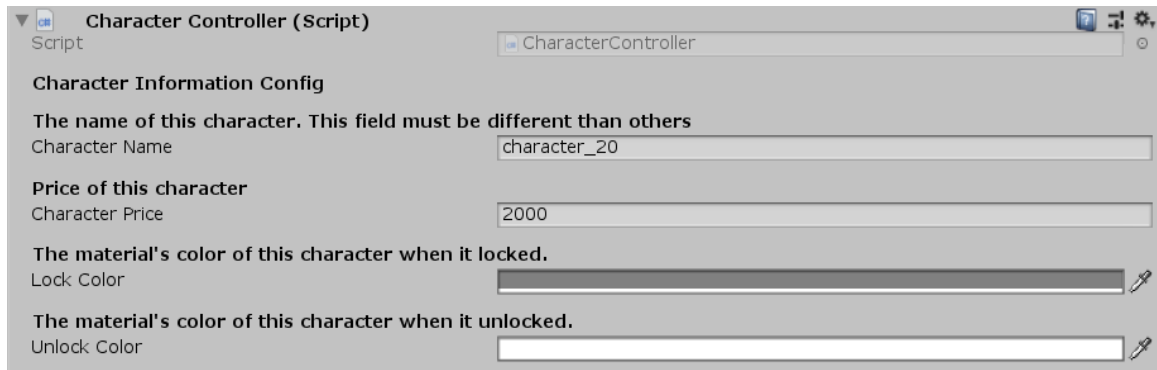


- Change the name of the prefab to a preferred one.
- Replace the *Mesh* in the *MeshFilter* component with your new mesh.
Replace the *Material* in the *MeshRenderer* component with your new material.

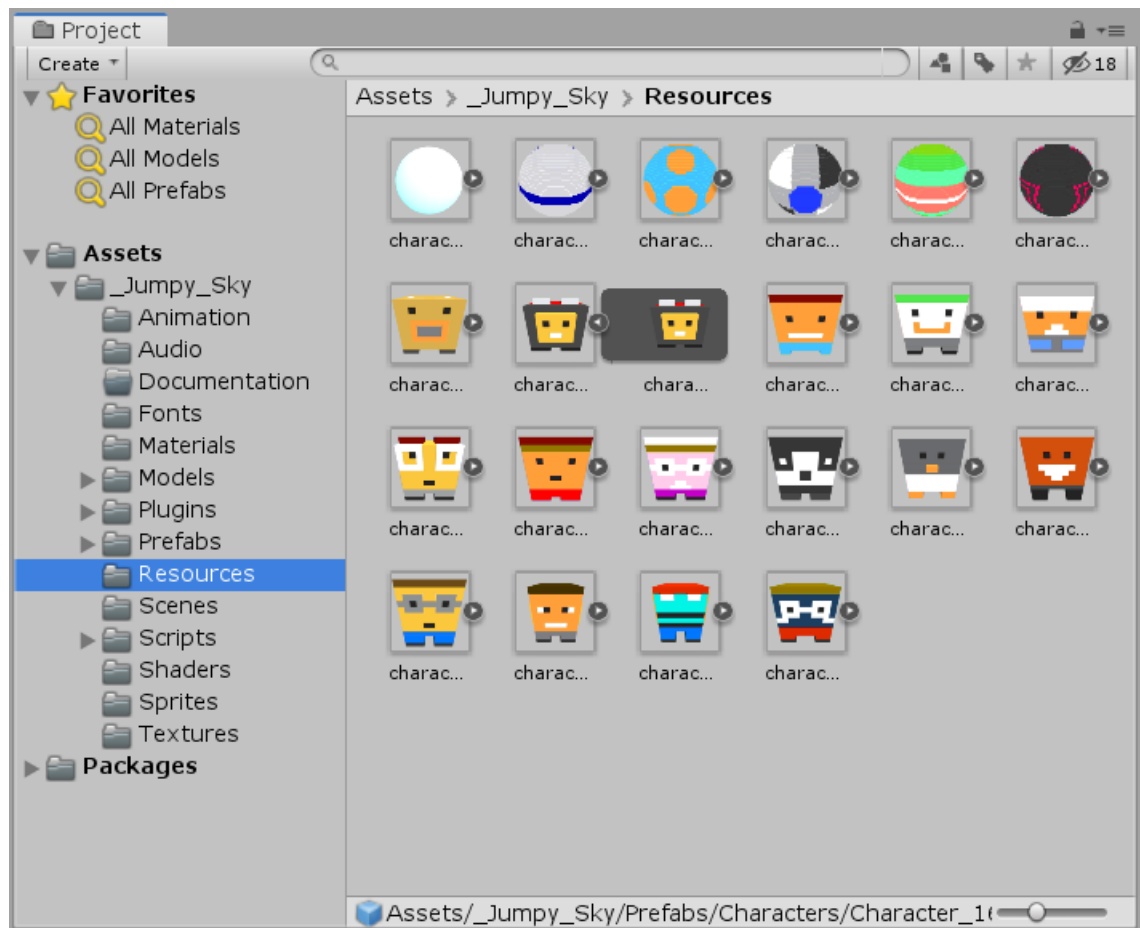




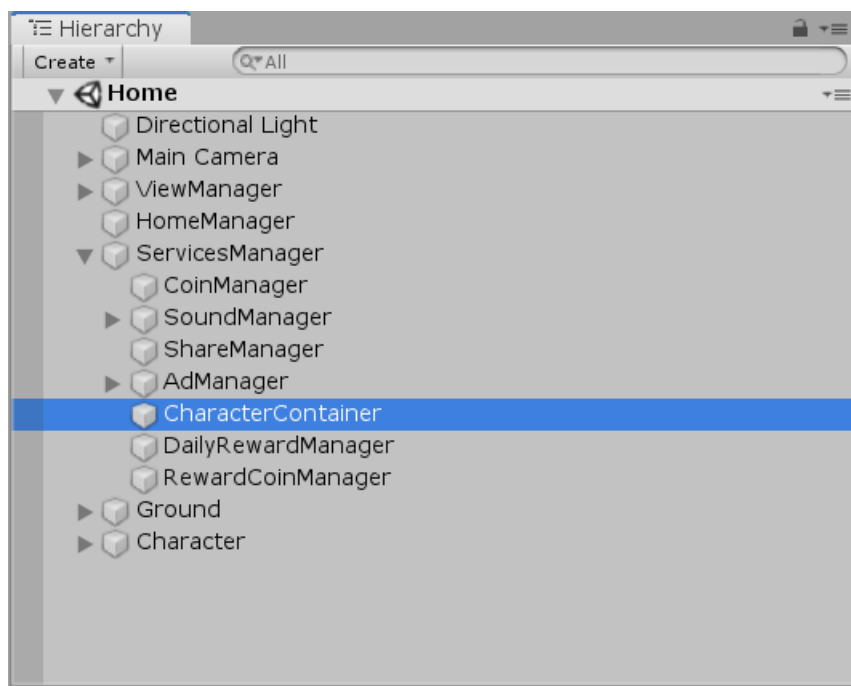
- Enter the character name and character price to the *CharacterController* component. If you want to give out this character for free, leave the *Character Price* to be 0 (it will be automatically unlocked). *Locked Color* is the color of the skin when it still not un lock. *Unlocked Color* is the color when the skin already unlocked. You can change these two colors as you want.

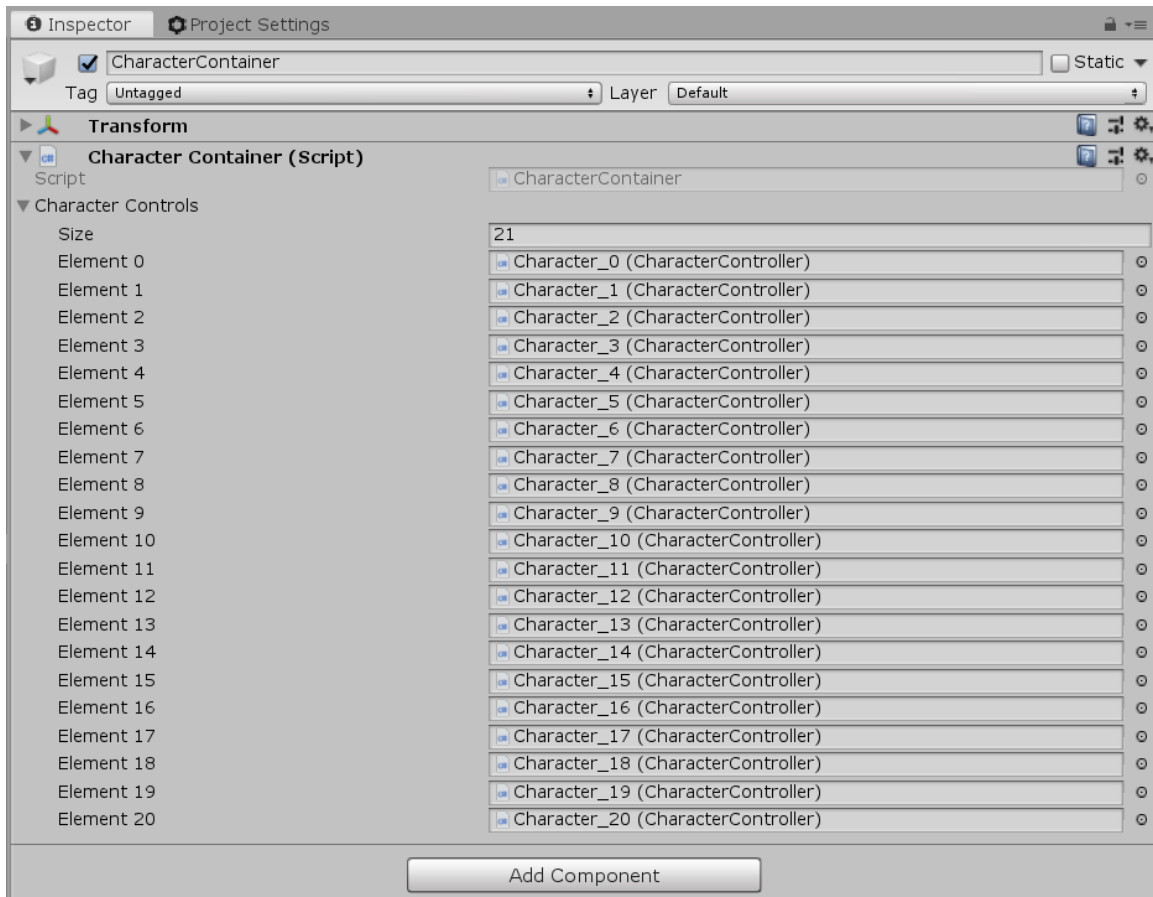


- Create an image of the character that just created with the size: 256x256, the background must be transparent. Save it with the name must exactly like the *Character Name* in Resources folder.



- Resize the character array in *CharacterContainer* game object then drag the new character to it and hit *Ctrl + Z* to save changes.





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

* **IMPORTANT:** the new character's name must not repeat any existing character name.

THANK YOU AND GOOD LUCK WITH YOUR GAMES!