

# Little Marino

Thank you for purchasing the “**Little Marino**” game template.



## Contents

Version History	<b>3</b>
Requirements	<b>3</b>
How to Open the Project	<b>4</b>
Build Settings	<b>5</b>
How to Play	<b>6</b>
How to Export	<b>7</b>
Prerequisites	7
Android and Java SDK	7
Export to Android	<b>8</b>
Export to XCode (iOS)	<b>10</b>
Admob	<b>12</b>
How to Create a New Level	<b>14</b>
Scripts	<b>17</b>
Music and SFX	<b>18</b>
Reskin	<b>19</b>

## Version History

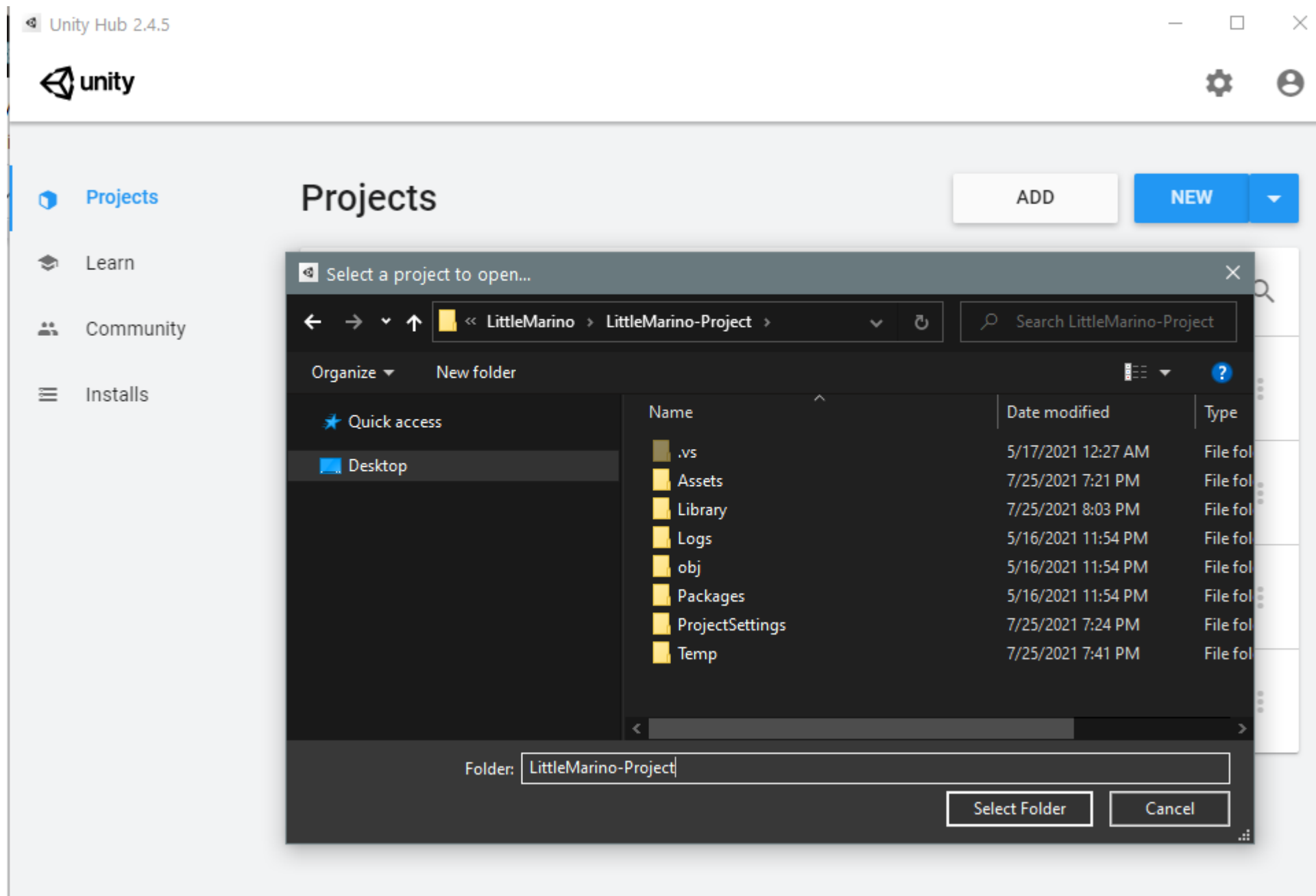
- Version 1.0
  - Initial release.

## Requirements

Unity 3D version 2019.3.9f1 or higher.

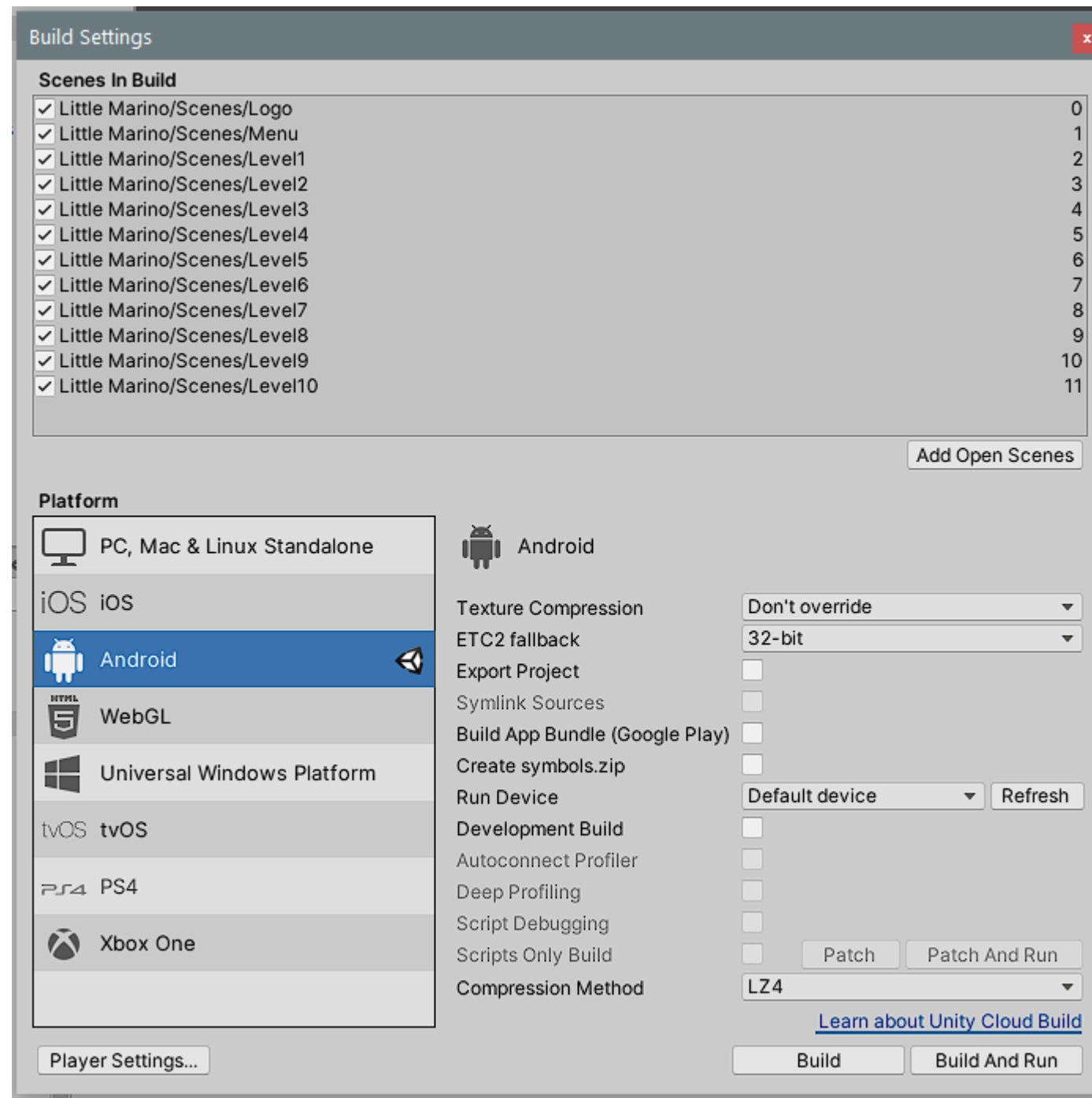
# How to Open the Project

Install [Unity Hub](#) and add the project in Unity Hub like the image below. Then click on the project name to open the project.



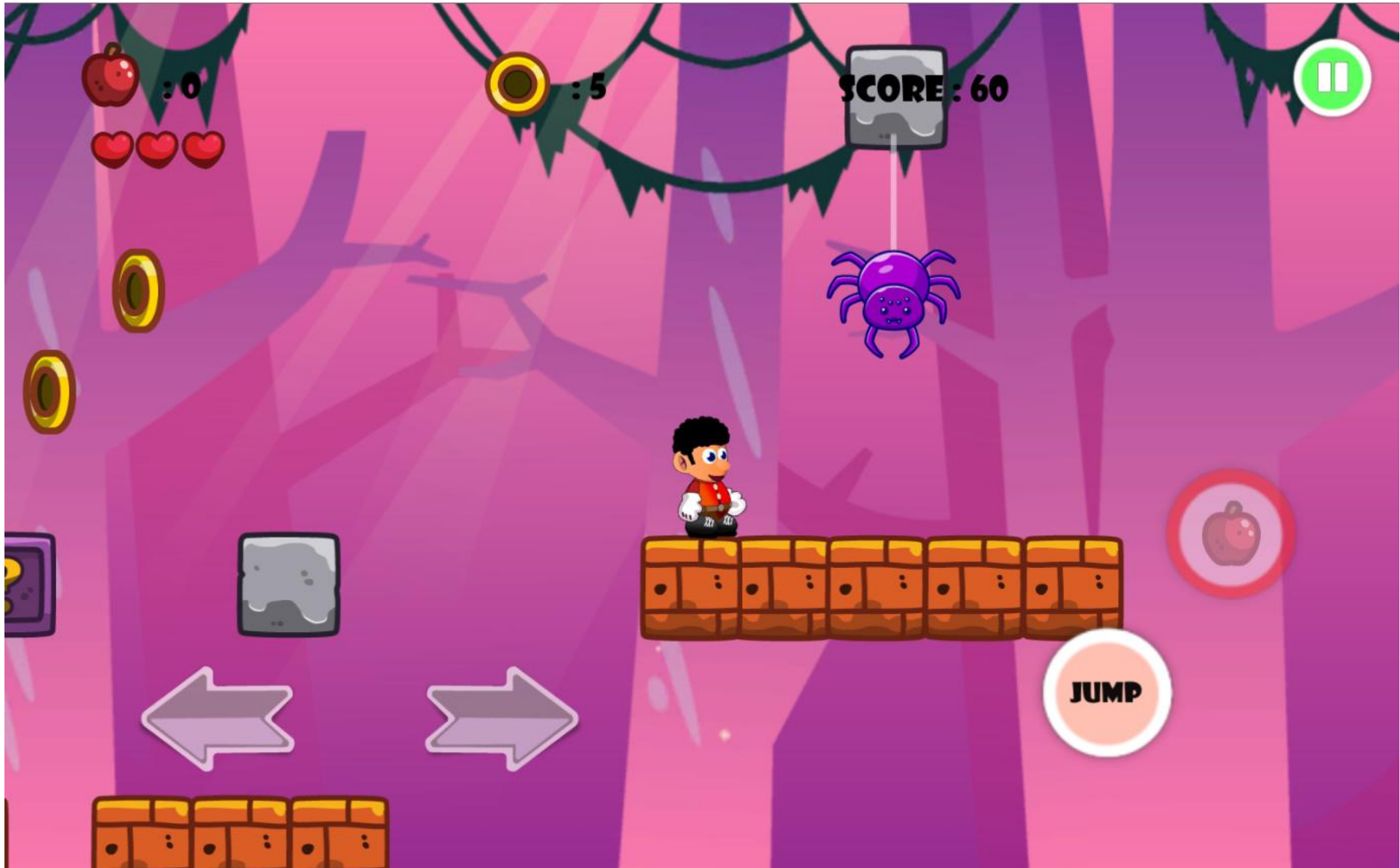
# Build Settings

Open build settings and make sure you have added all scenes to it.



# How to Play

In this platformer game, you have to help a man who is lost in the forest to find his house and in this way, you have to cross various obstacles.

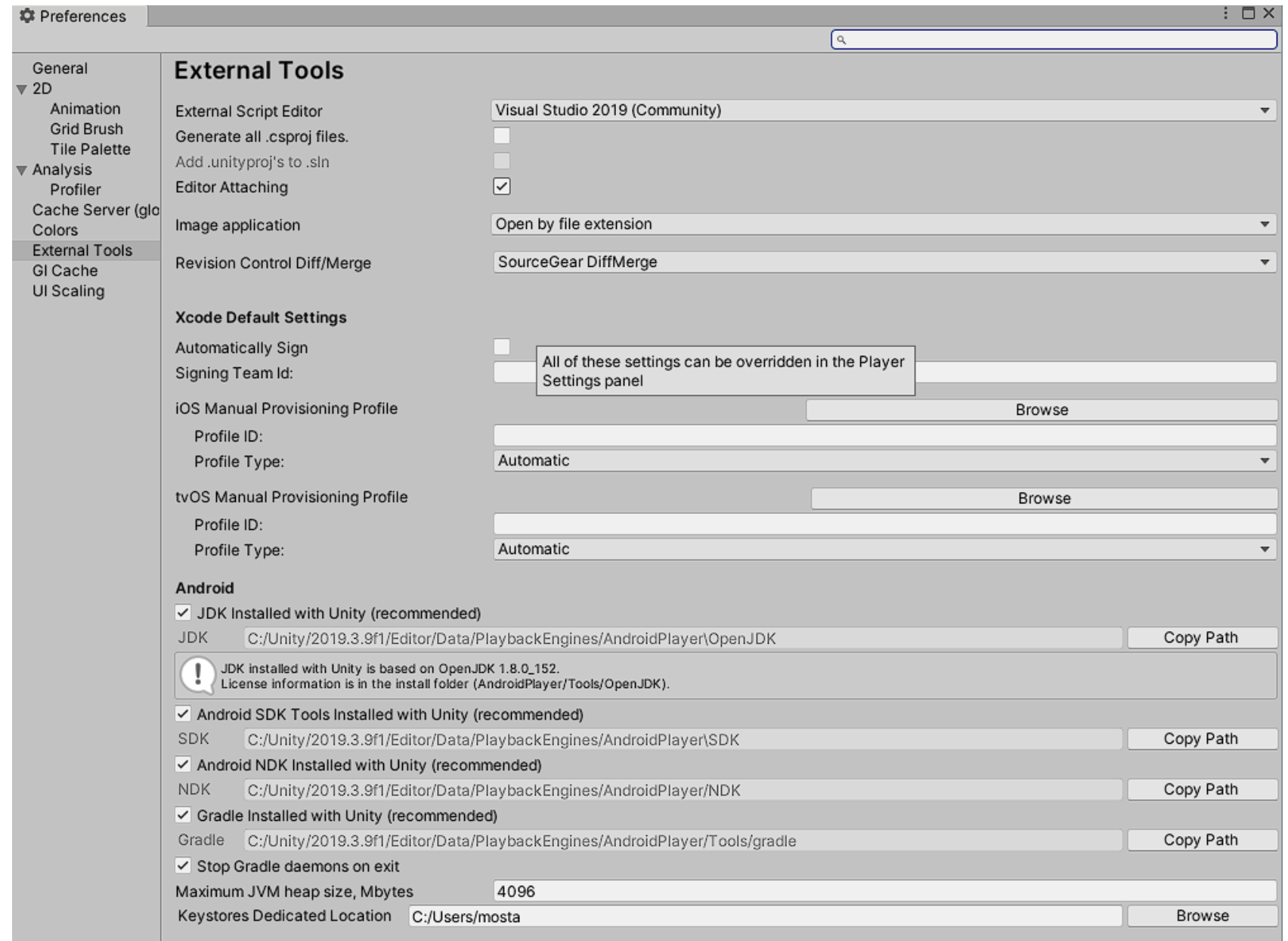


# How to Export

## Prerequisites

### Android and Java SDK

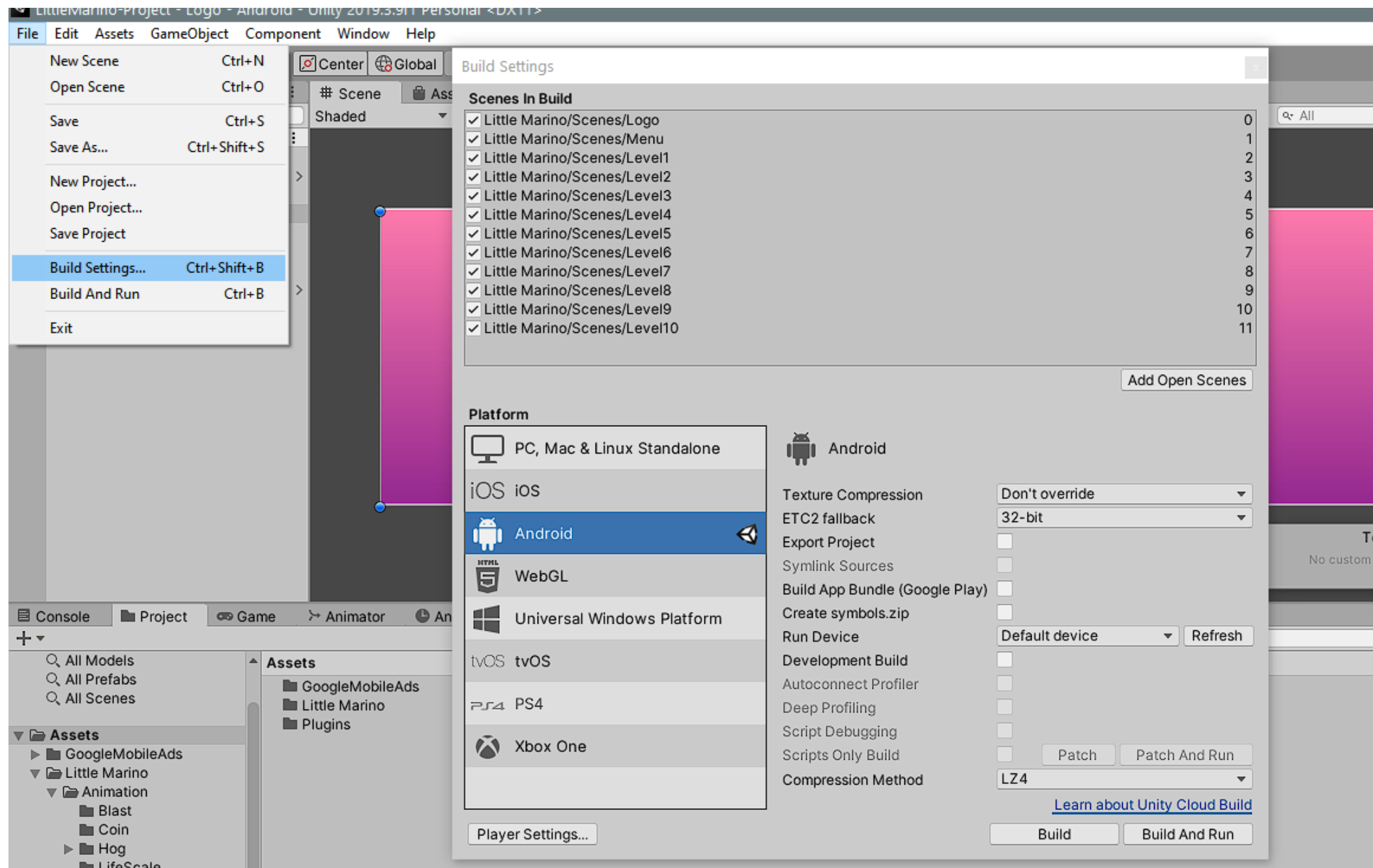
Like the image, click on Preferences from the Edit menu in Unity so that Unity Preferences window is shown and then enter the installation folder address of Android SDK, Java SDK and NDK (ARM64) in their places (you need to have Android SDK and Java SDK installed on your system beforehand).





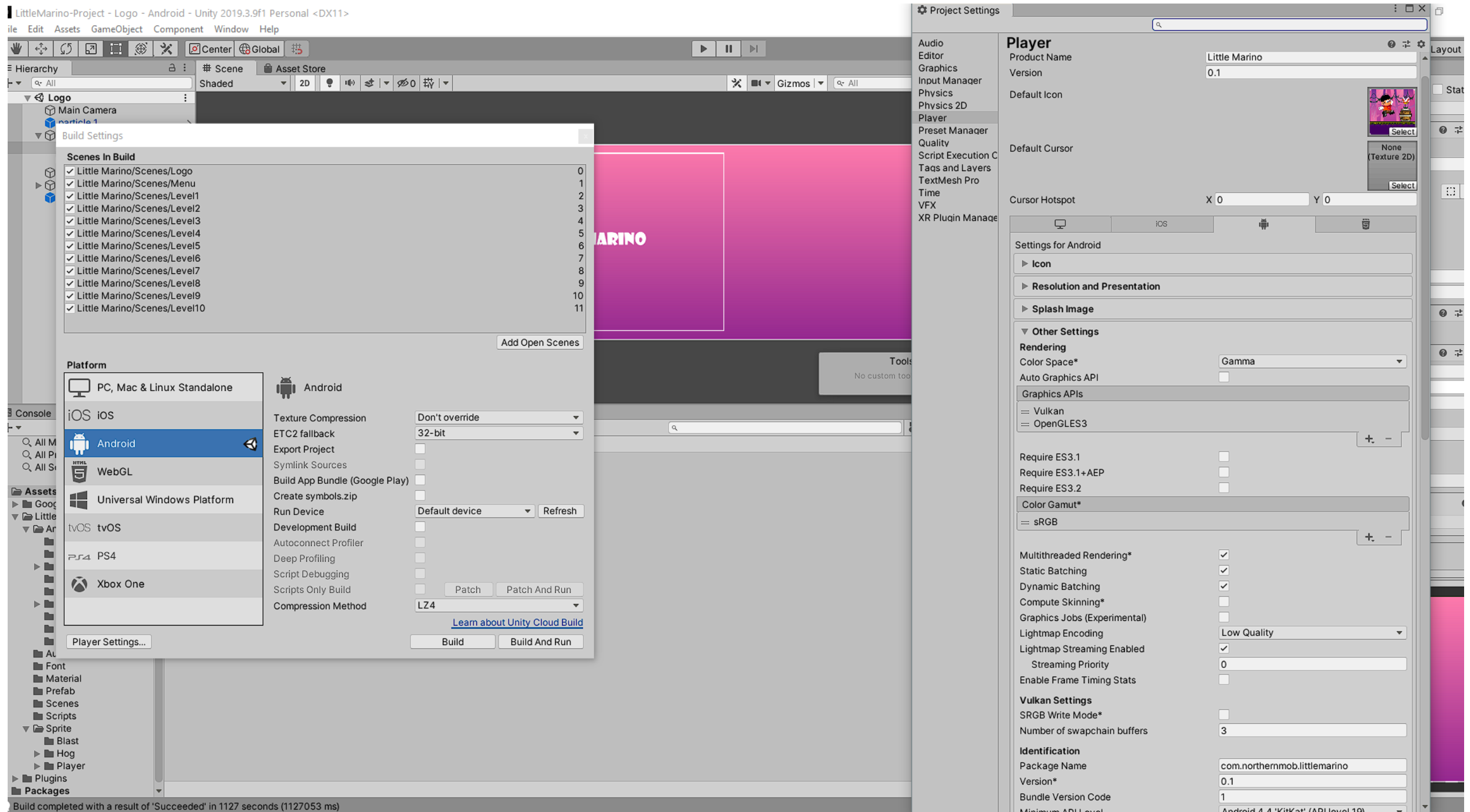
# Export to Android

After the project is opened, click on Build Settings in the File menu like the image below. Then in the newly opened window, click on Android platform and then on Player settings.

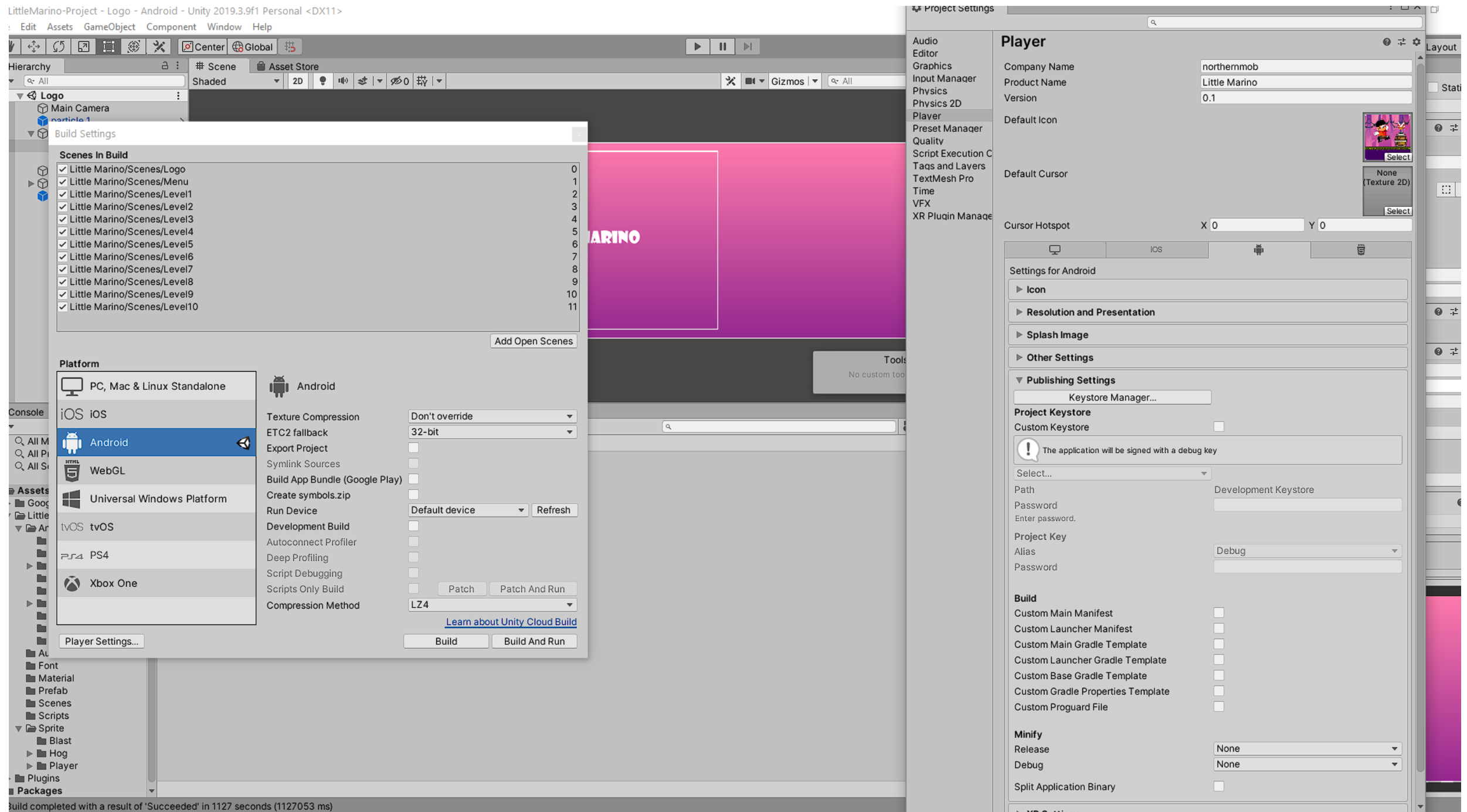


Then like the image below, select the desired name for the game and its package name in the Inspector section (if you like).





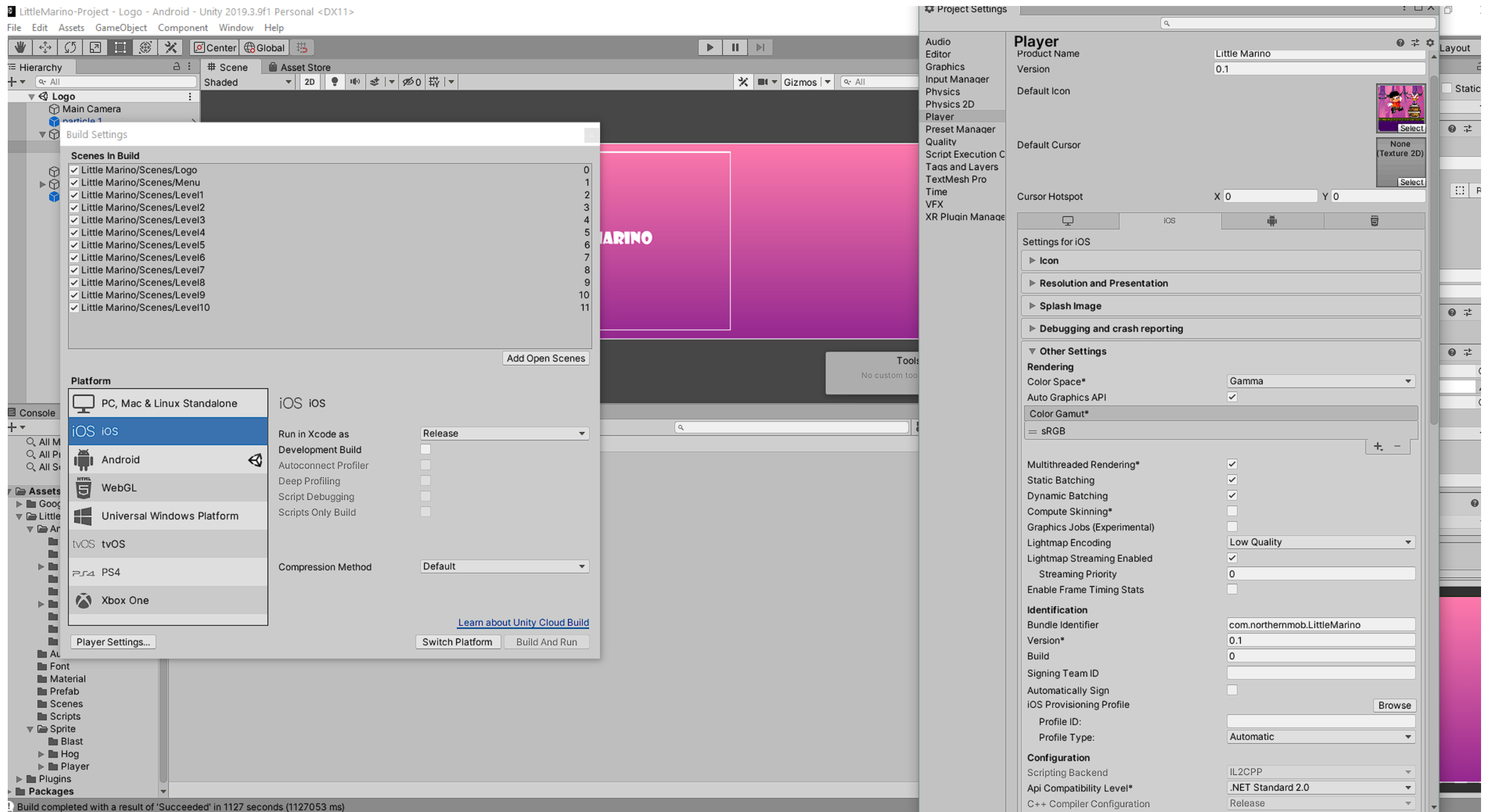
And finally, you have to sign the exported version with your key store like the image below and then click on the Build button so that Unity starts making the exported version for Android.



Here is a guide from Unity website about making Android version: <https://learn.unity.com/tutorial/how-to-publish-to-android>

## Export to XCode (iOS)

Click on Build Settings from the File menu in Unity.



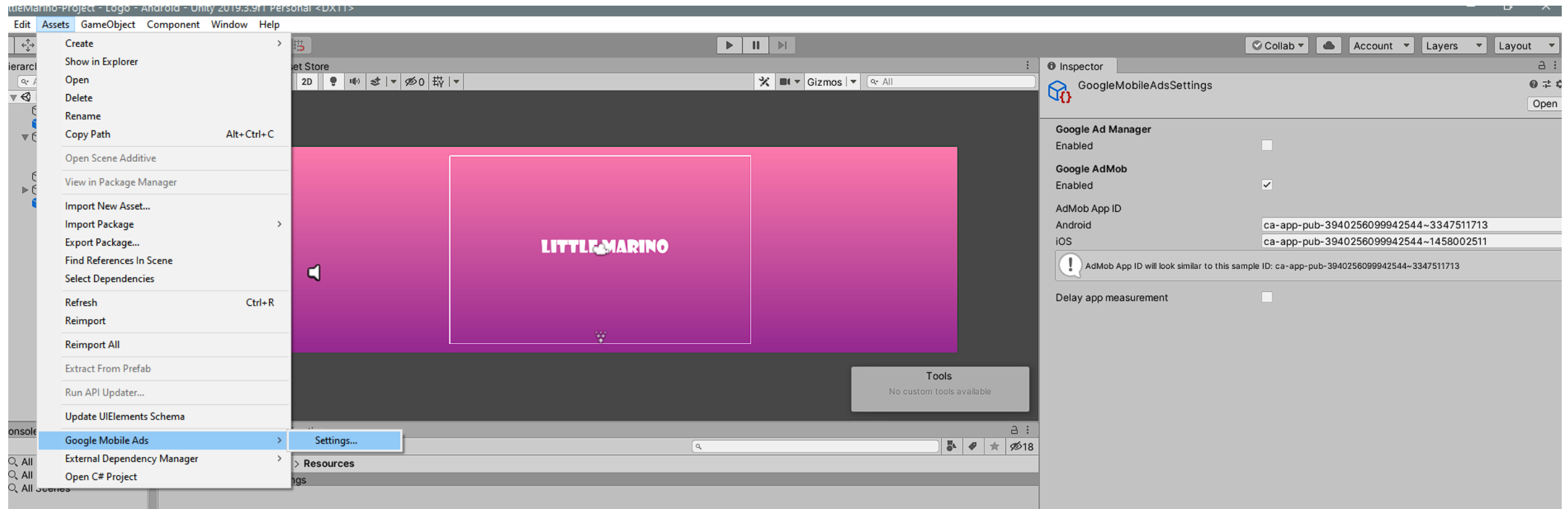
Finally to make the exported version for XCode, click on the Build button in the Build Setting window.

And here is a guide for making Android and iOS versions from the Unity website that you can read to get some extra details and information:

<https://learn.unity.com/tutorial/building-for-mobile>

# Admob

Like the below image, click on settings from assets menu so that Google Mobile Ads configurations are shown up and then enter the App IDs that you have created in your admob panel in Android and iOS sections.

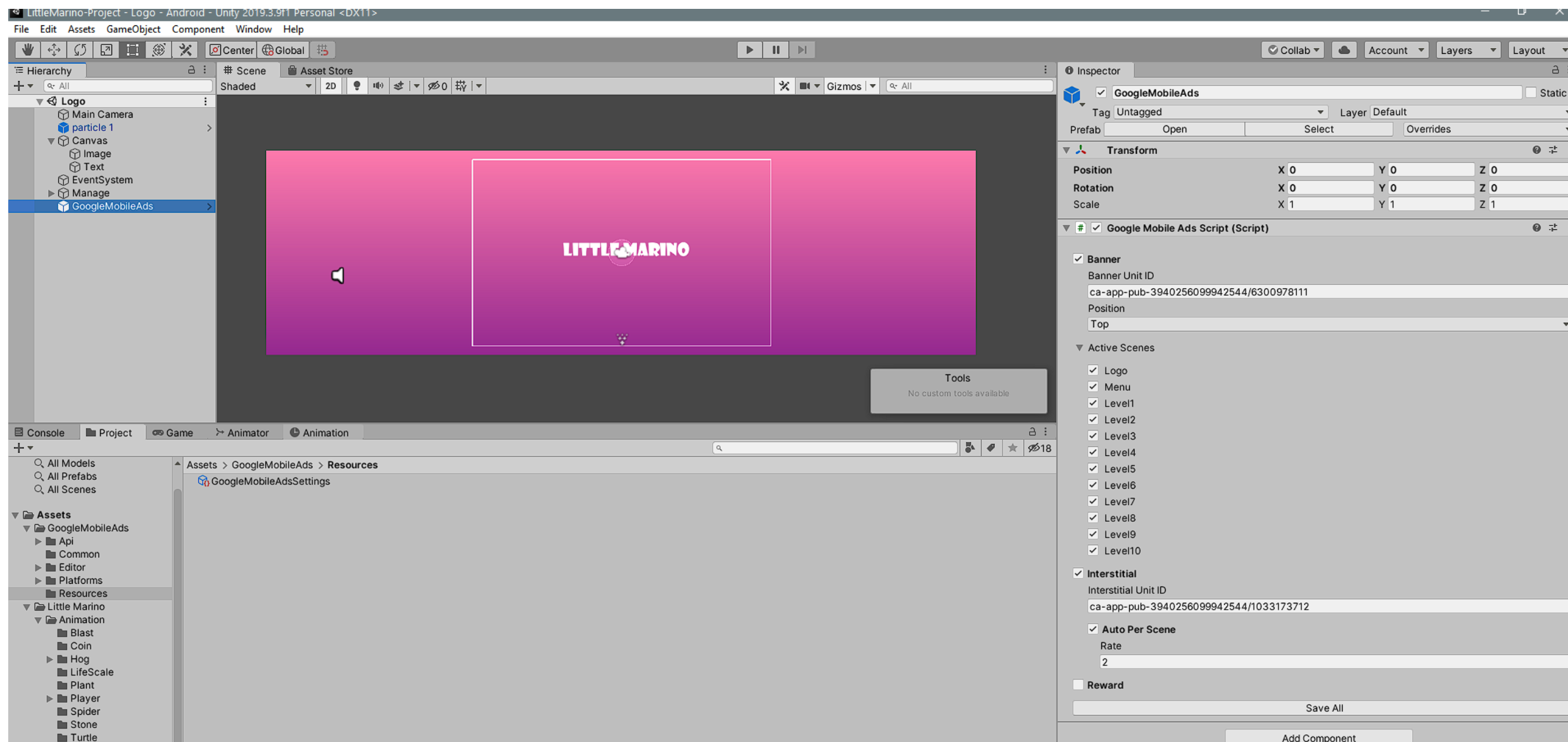


Then in the "**Logo**" scene, select GoogleMobileAds object and then in the inspector, Admob settings are shown. If you want to use banners, you can activate the banner section and enter the corresponding unitid. You can also choose the banner's position and where in the game it is shown.

To activate full screen ads, you have to mark interstitial choice and enter the related unit id in it like the below image. If you activate Auto per scene, ads will be shown based on the number entered in the Rate field when scenes are changing and you don't have to load the ads manually.

If you want to use reward ads, then you need to activate the Reward section and enter its unit id.

**Note:** After implementing the changes, make sure to press the Save All button to save your changes.



To show the ads manually, use these codes:

Showing Banner ads:

```
GoogleMobileAdsScript.USE.ShowBanner();
```

Removing and not showing banner ads:

```
GoogleMobileAdsScript.USE.HideBanner();
```

Showing full screen ads:

```
GoogleMobileAdsScript.USE.ShowInterstitial();
```

Showing reward ads:

```
GoogleMobileAdsScript.USE.ShowRewardAd();
```

Rewarding the player after reward ads is shown successfully:

For this, you need to edit line 73 of GoogleMobileAdsScript like below:

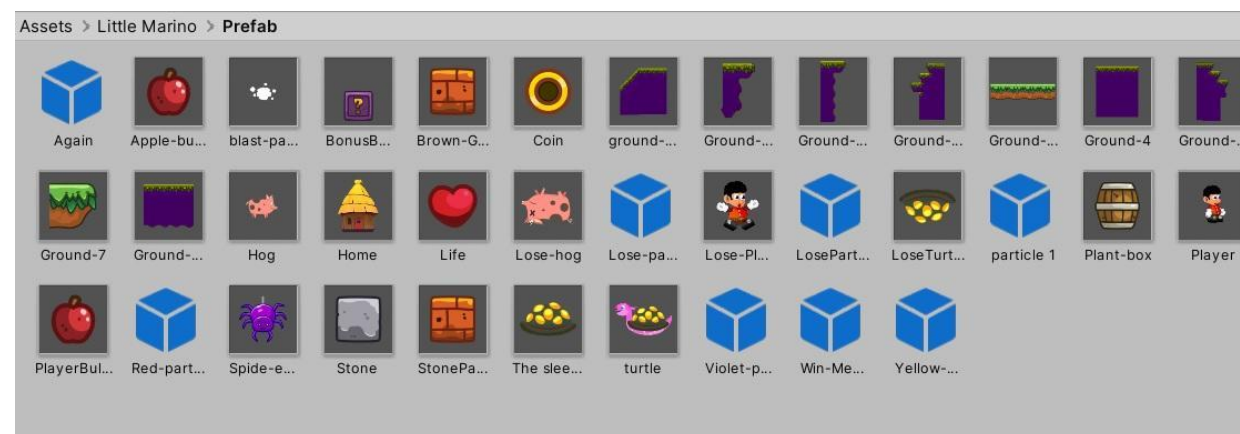
```
71     public void RewardVideo(string type, double amount)
72     {
73         // put ur rewarded video here
74     }
75 }
```

## How to Create a New Level

Save one of the scenes that its first name starts with Level and place it in the Scenes folder.

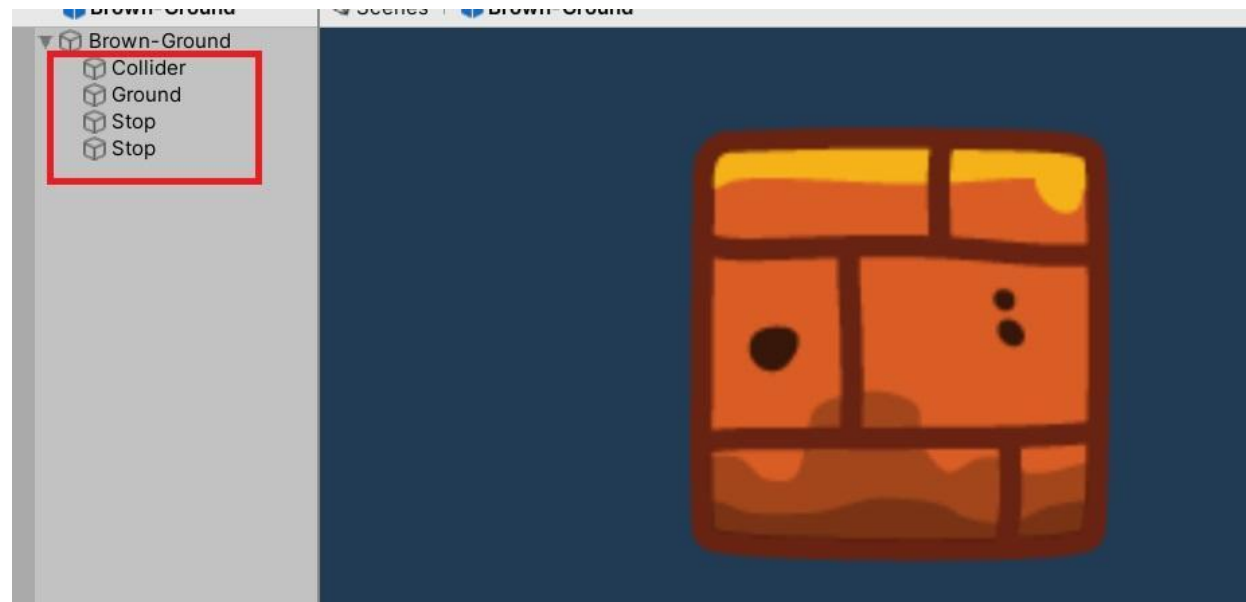


You can design this Scene as you wish and you can also use the Prefab folder for your design.

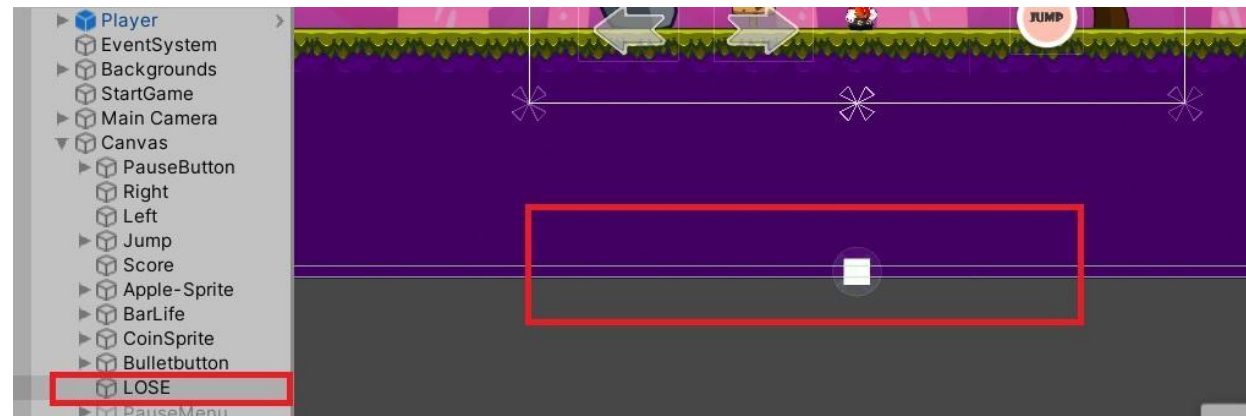




There are a few things to keep in mind when designing a playground, tiles contain colliders that each have separate tags.

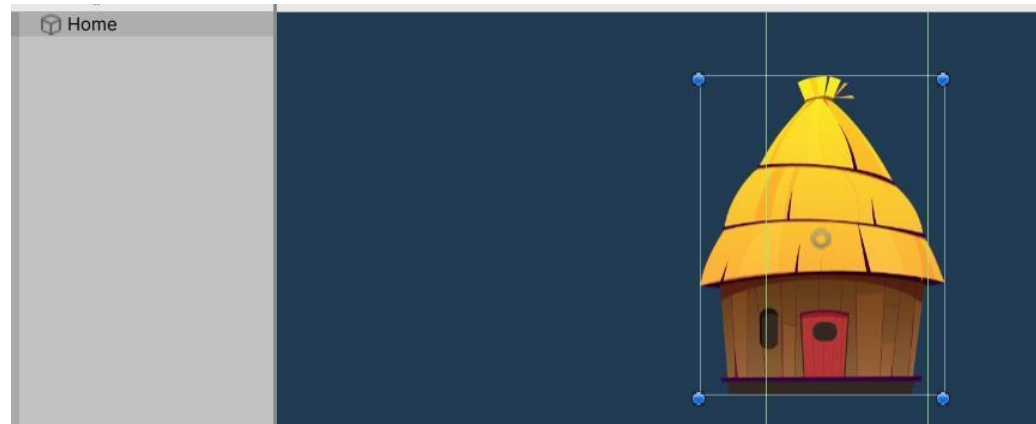


Ground is for the player to return to idle mode by hitting and end the jump.  
Stop is for the player not to move the Backgrounds by hitting it.



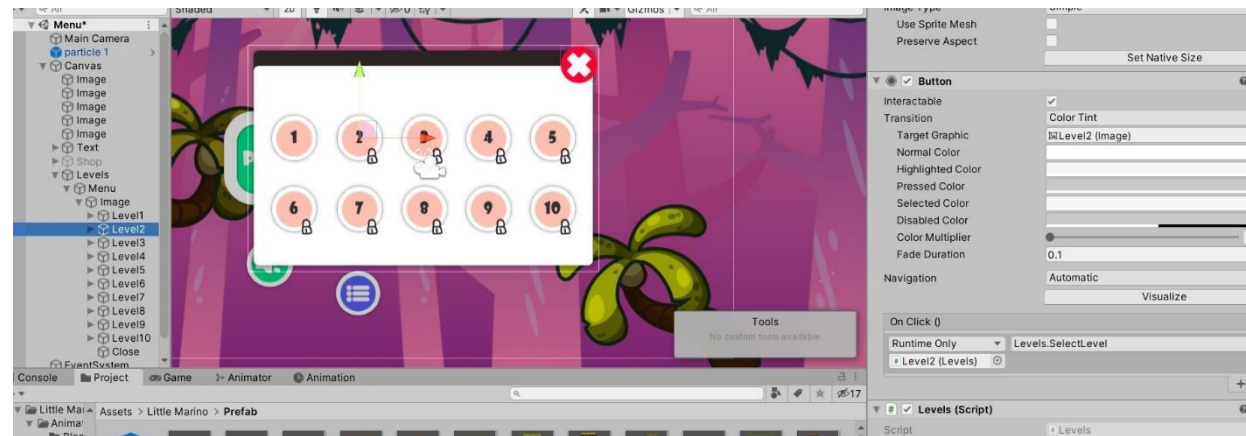
There are parts in the game where the player falls and is destroyed and this is done by herself called "LOSE".



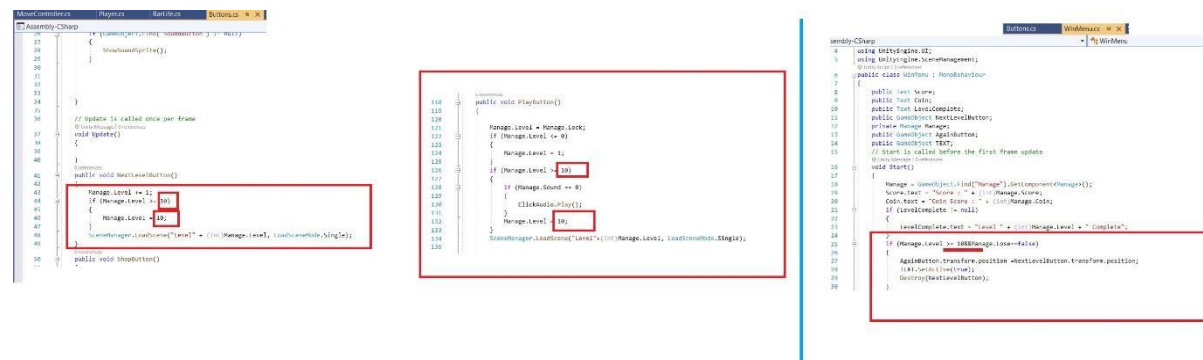


To end the game, the player must hit the name Home.

The Level of the game is controlled by "Manage.cs" (int Level).



To add steps, you need to change the step menu in the scene menu and put more buttons.



In the (Buttons.cs and WinMenu.cs) code, the number 10 indicates the last stage of the game and if you add game stages, this number must be changed.

# Scripts

All scripts are under the Scripts folder.

WinMenu.cs

This code is responsible for controlling the last menu of the game, which includes displaying points.

Buttons.cs

This code is responsible for controlling the game buttons.

In the Update method, the "B" button is programmed to shoot the apple

Shake.cs

This code is responsible for controlling the movement of the camera.

CameraFit.cs

This code is responsible for controlling the size of the camera.

DestroyThis.cs

This code is responsible for destroying the object.

Manage.cs

This code is responsible for managing the game, which includes points, saves and the end of the game.

MoveBackground.cs

This code is responsible for controlling the movement of objects. Background movement is done by "Manage.cs" management.

Player.cs

This code is responsible for controlling the player, which includes moving and losing.

Startgame.cs

This code is responsible for zeroing points and starting a new game and providing the desired items to Manage.cs.

BonusBox.cs

This code is responsible for controlling the score box.

BrownStone.cs

This code is responsible for controlling the brown stone.

Enemy.cs

This code is responsible for controlling enemies.

Help.cs

This code is responsible for controlling the help menu in the first step.

Levels.cs

This code is responsible for controlling the step menu.

MoveController.cs

This code is responsible for controlling the player's movement.

In the Update method, the movement code for the RightArrow, LeftArrow and Space buttons is written.

PlayerBullet.cs

This code is responsible for controlling the player's bullet.

## Music and SFX

To change the game's sounds and music, you have to replace the sounds in the Audio folder with what you want. But note that you must keep the files' names exactly as before.



# Reskin

All of the images that are used in the game are in the Sprite folder. You can replace the default images with the new ones that you want.

**Important Note:** For changing the game's default images, new images should have the same size and name as default ones.

