Galaxy Battle - Manual

Thank you for purchasing the “**Galaxy Battle**” game template.

Content

Version History

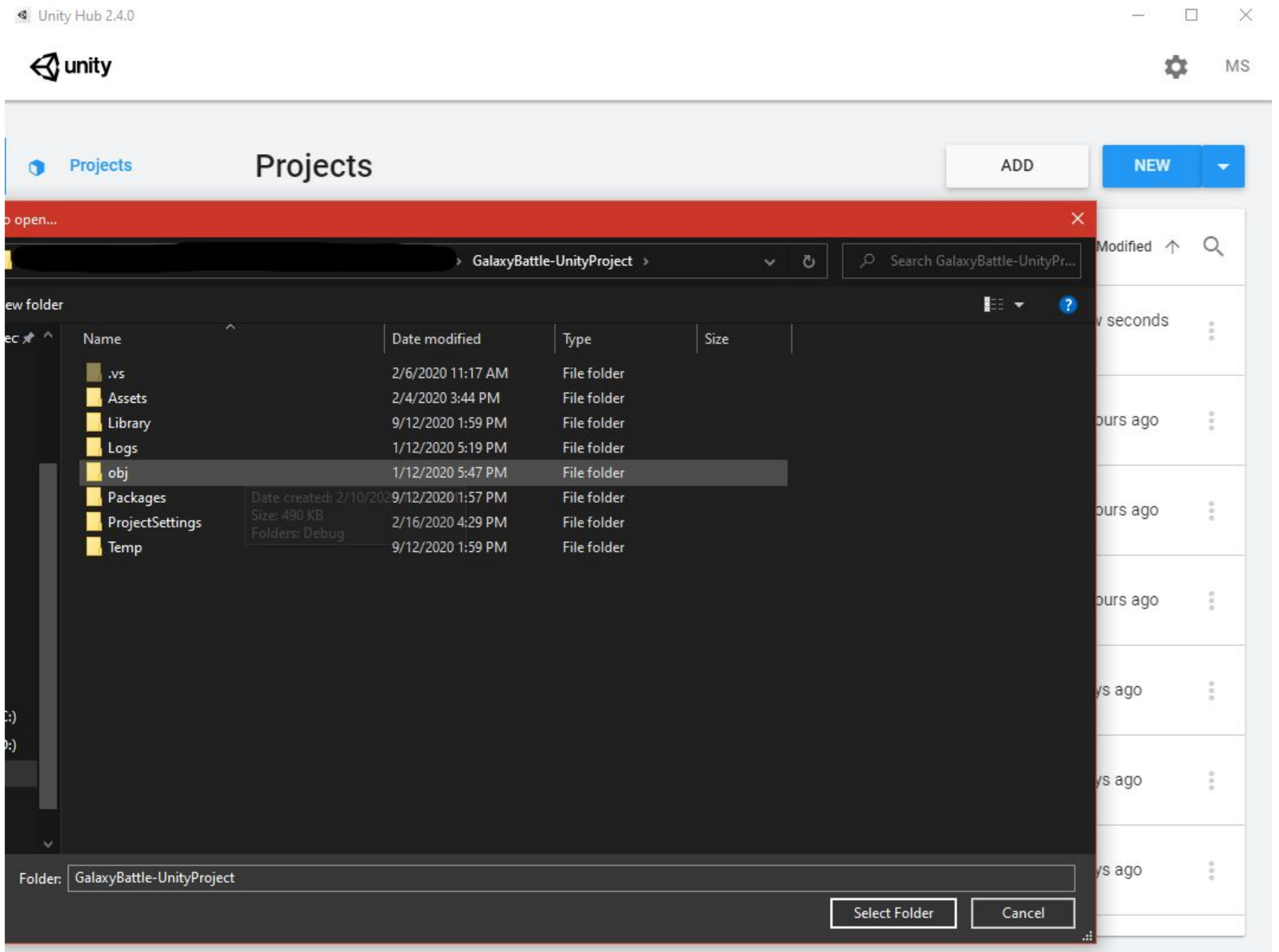
- Version 1.0
    - First release

Requirements

- Unity 3D version 2018.4.11 or higher

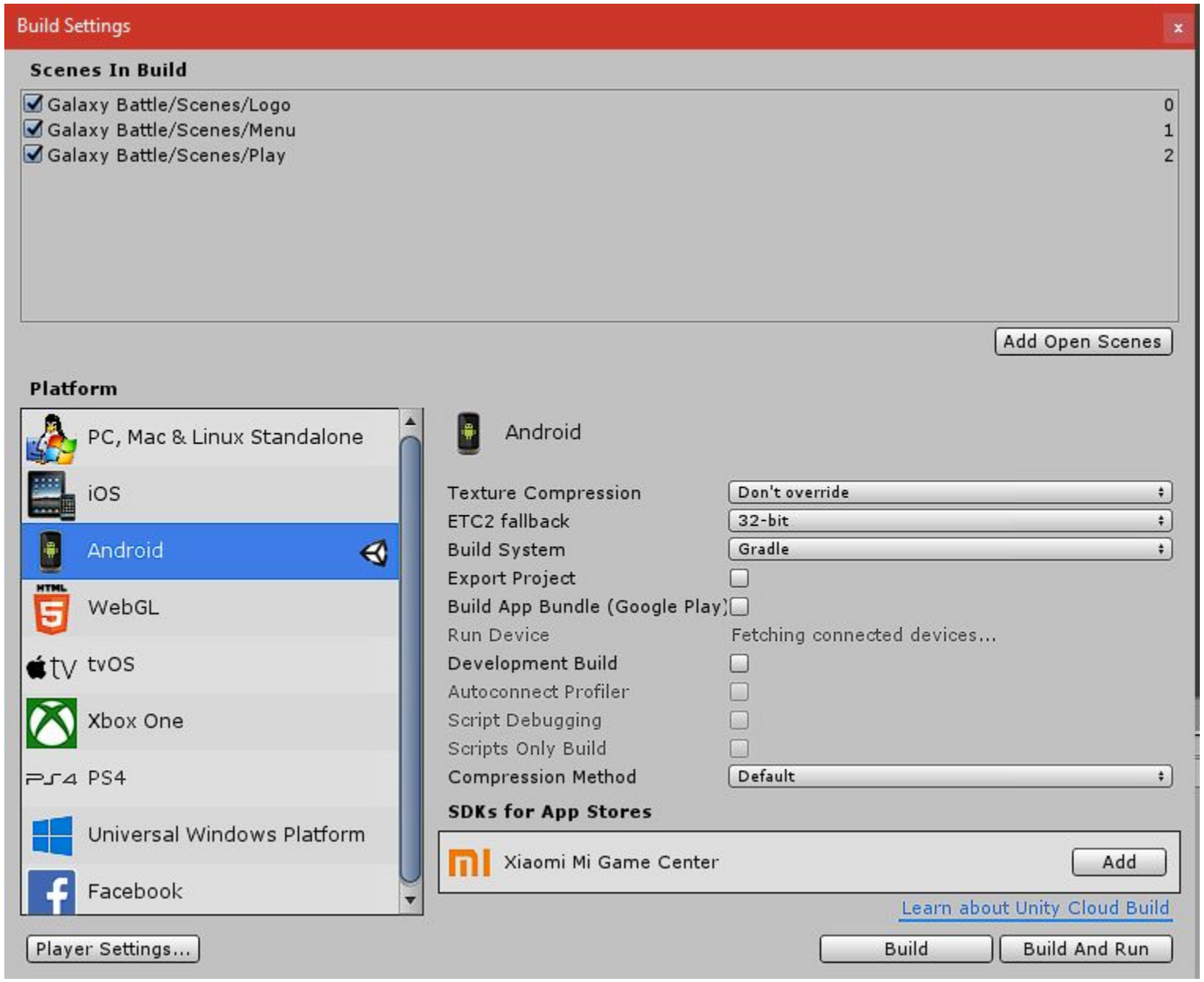
How to Open the Project

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



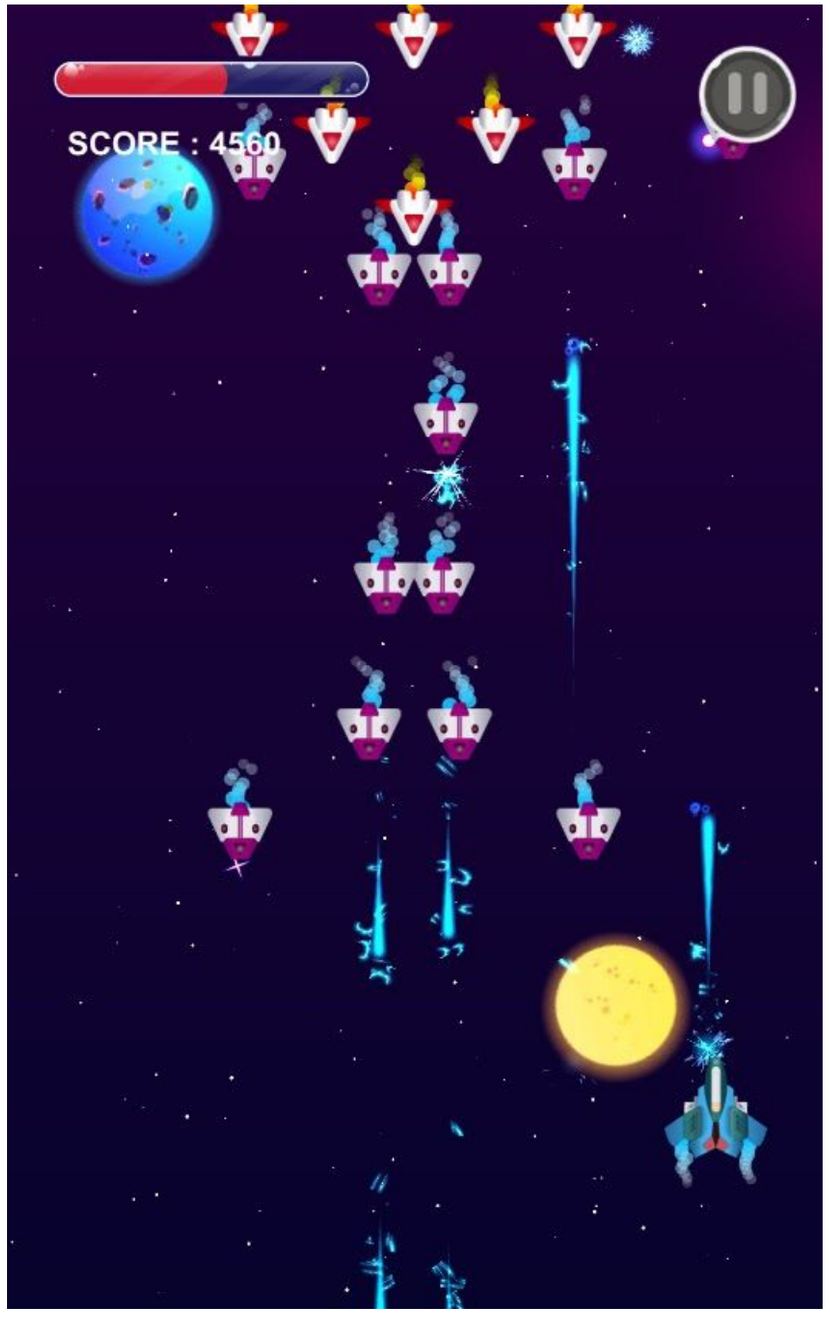
Build Settings

Make sure you have added these scenes.



How to Play

In this action game, you control your space airplane and have to destroy enemies through various levels.

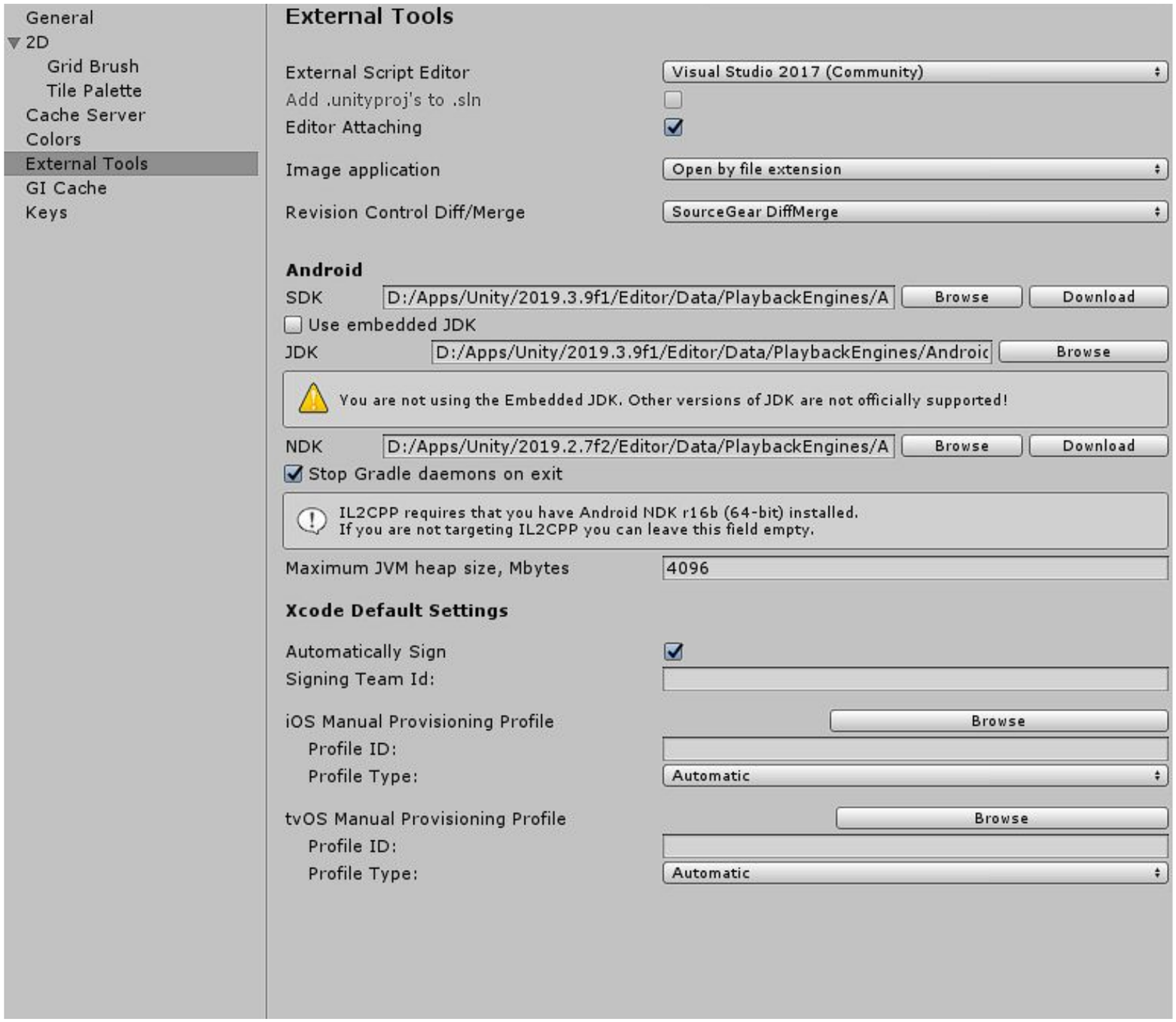


How to Export

Prerequisites

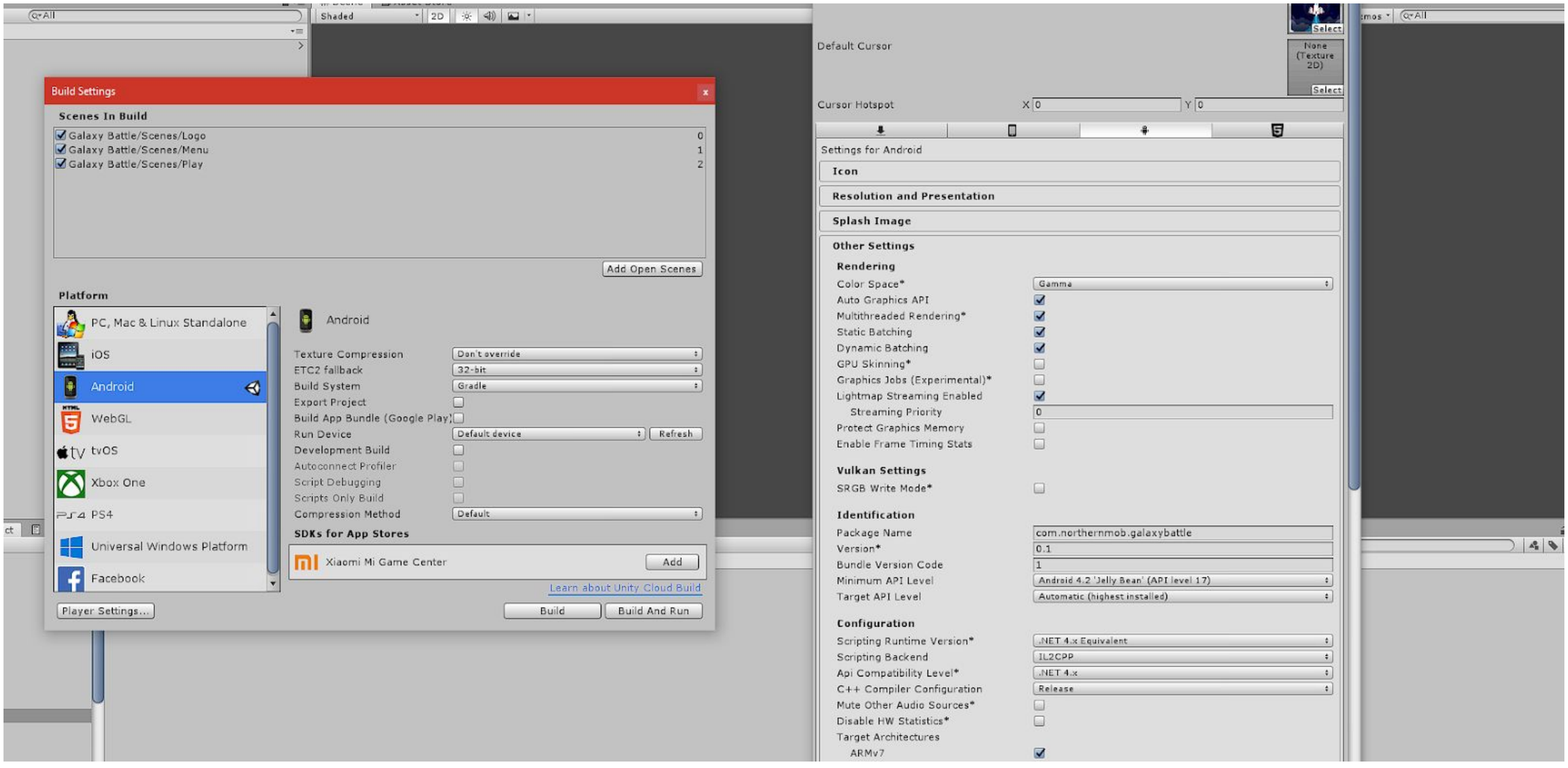
Android and Java SDK

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).



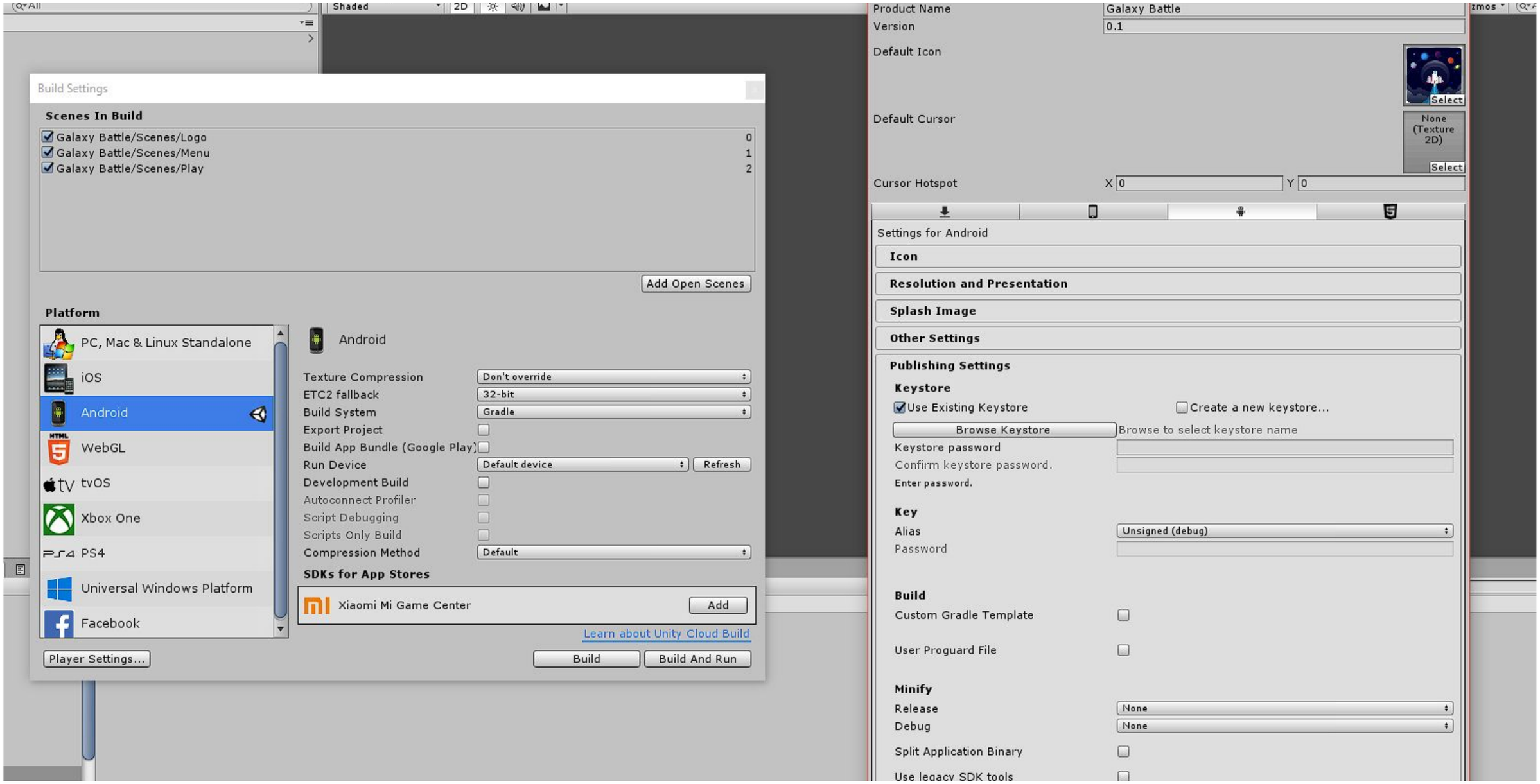
Export to Android

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

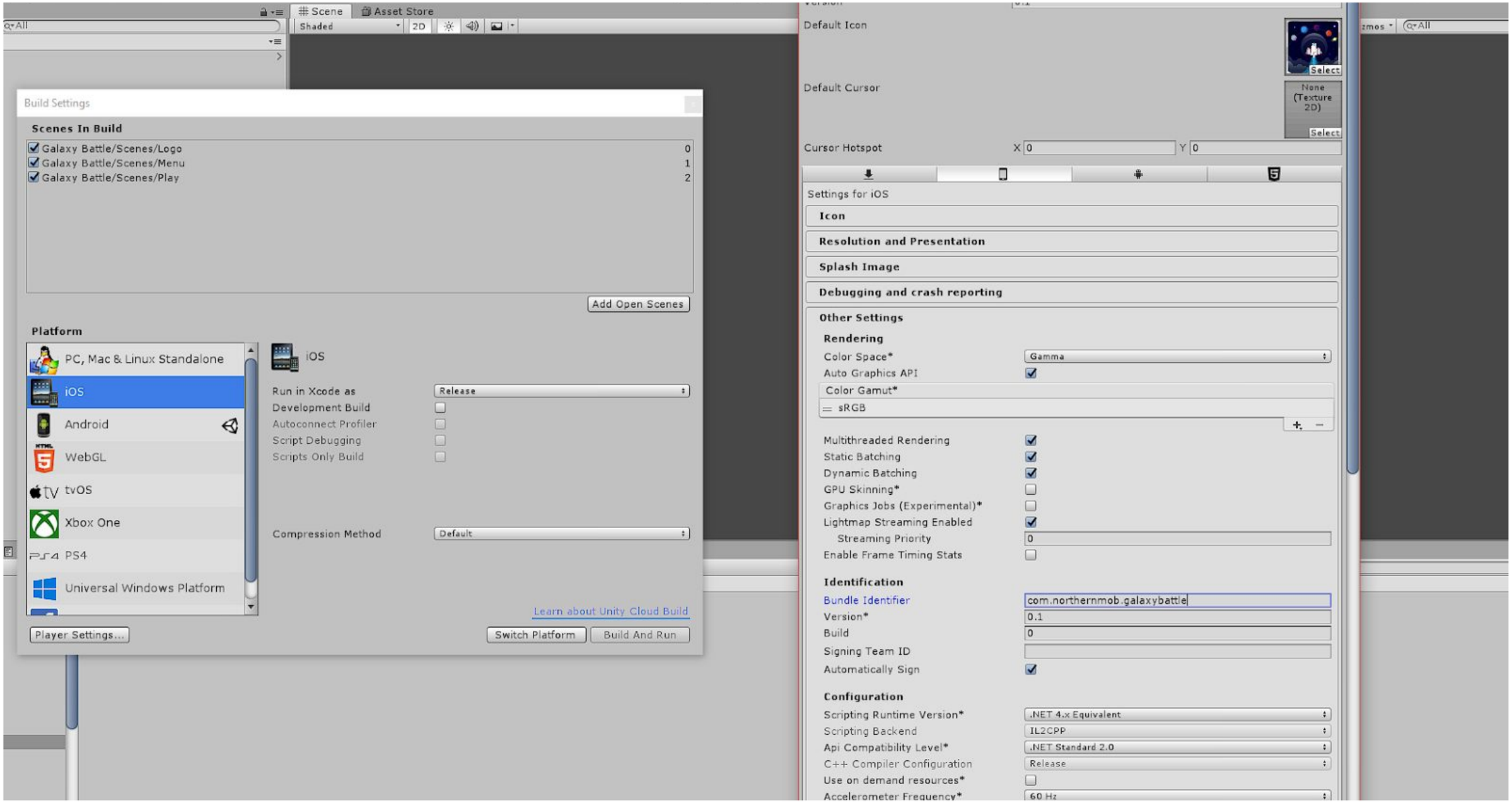
And finally, you have to sign the exported version with your keystore like the image below and then click on the Build button so that Unity starts making the exported version for Android.



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

Export for Xcode (iOS)

Click on Build Settings from the File menu in Unity.



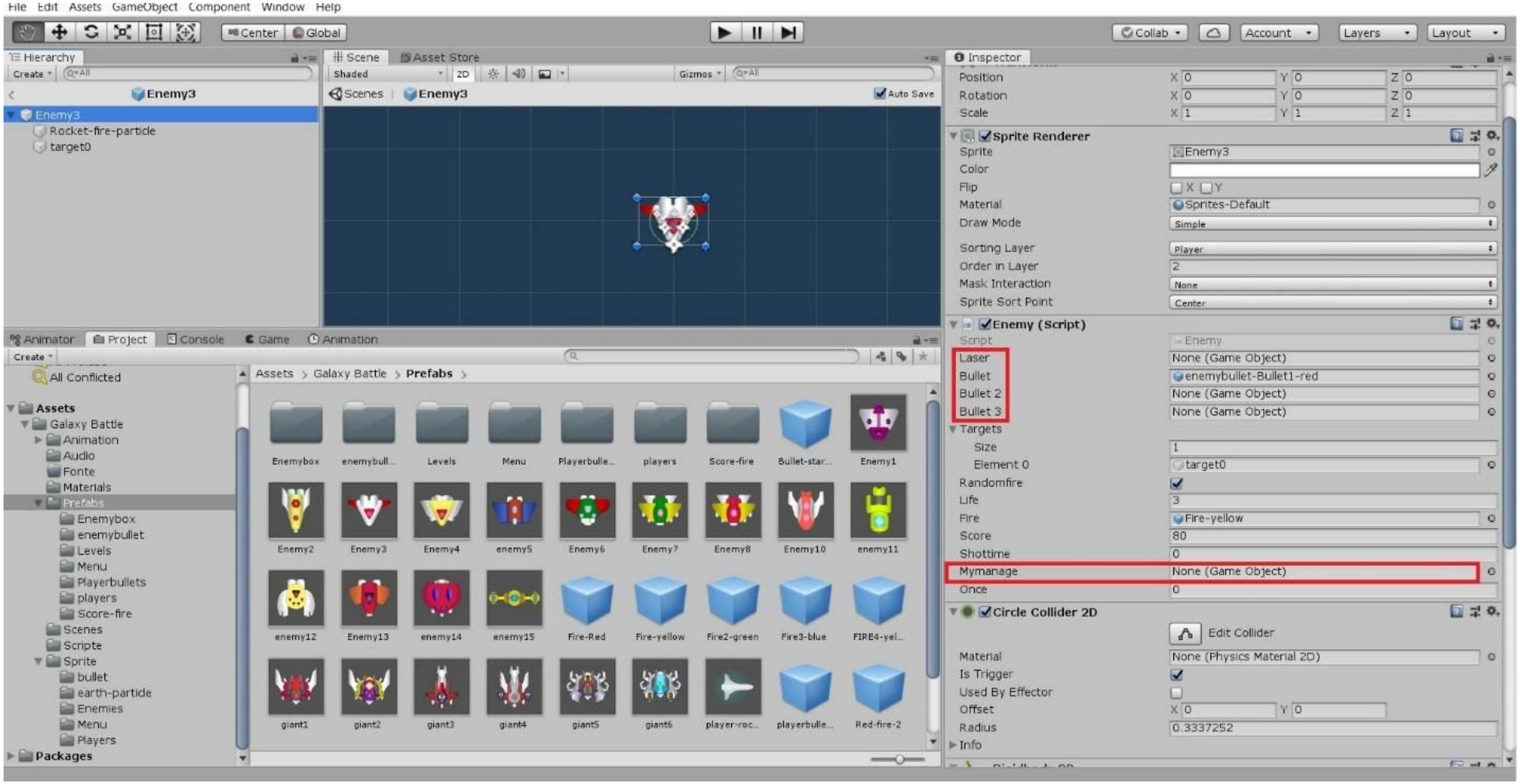
Finally to make the exported version for Xcode, click on the Build button in the Build Setting window.
And here is a guide for making Android and iOS version from the Unity website that you can read to get some extra details and information:
https://learn.unity.com/tutorial/building-for-mobile

How to Add NewLevel

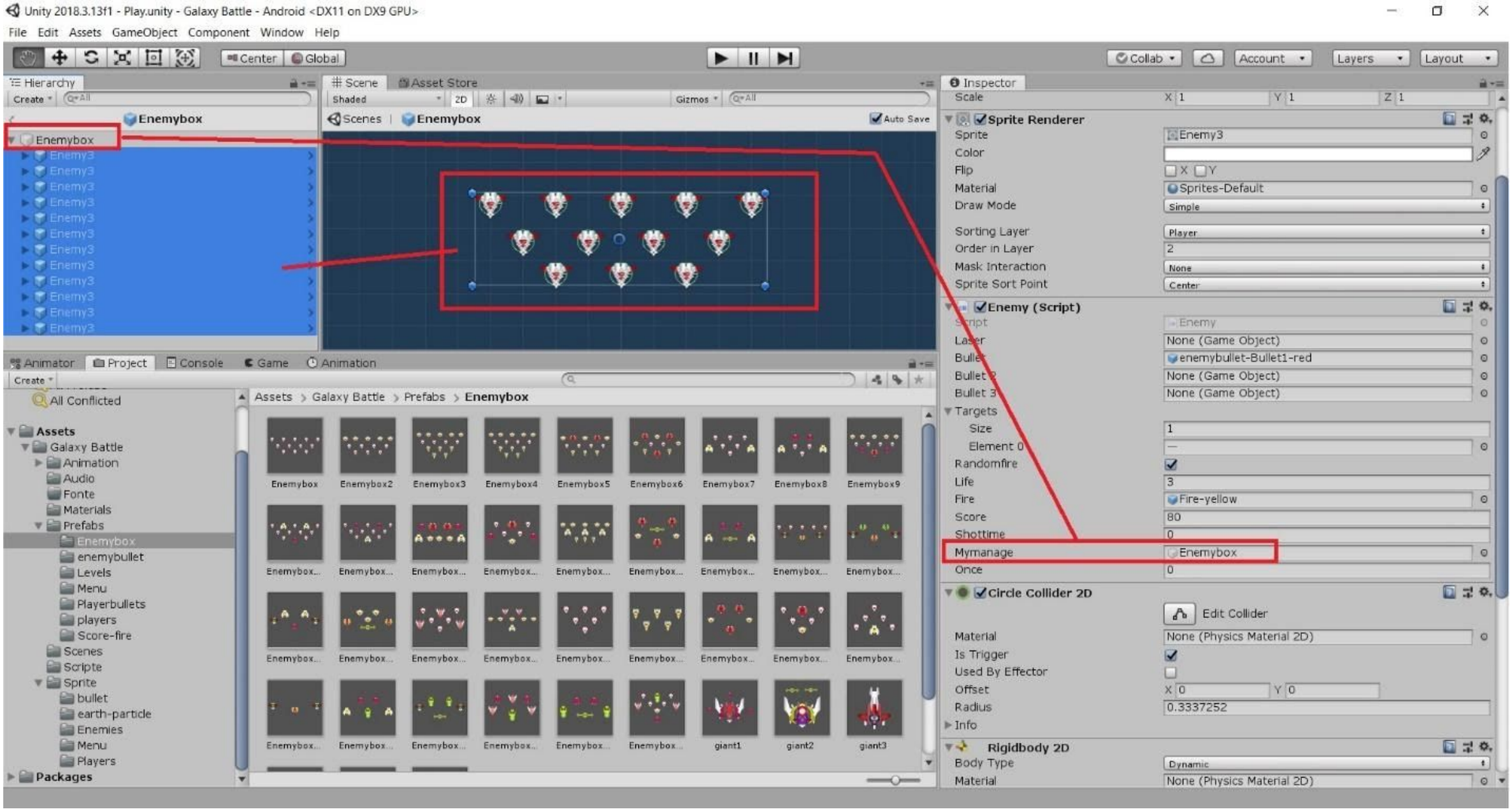
You need to pay attention to 5 codes (Startgame.cs Manage.cs Managelevel.cs Enemybox.cs Enemy.cs) to design new steps.
All of the level design prefabs are in the following two folders.

Enemy.cs code performance

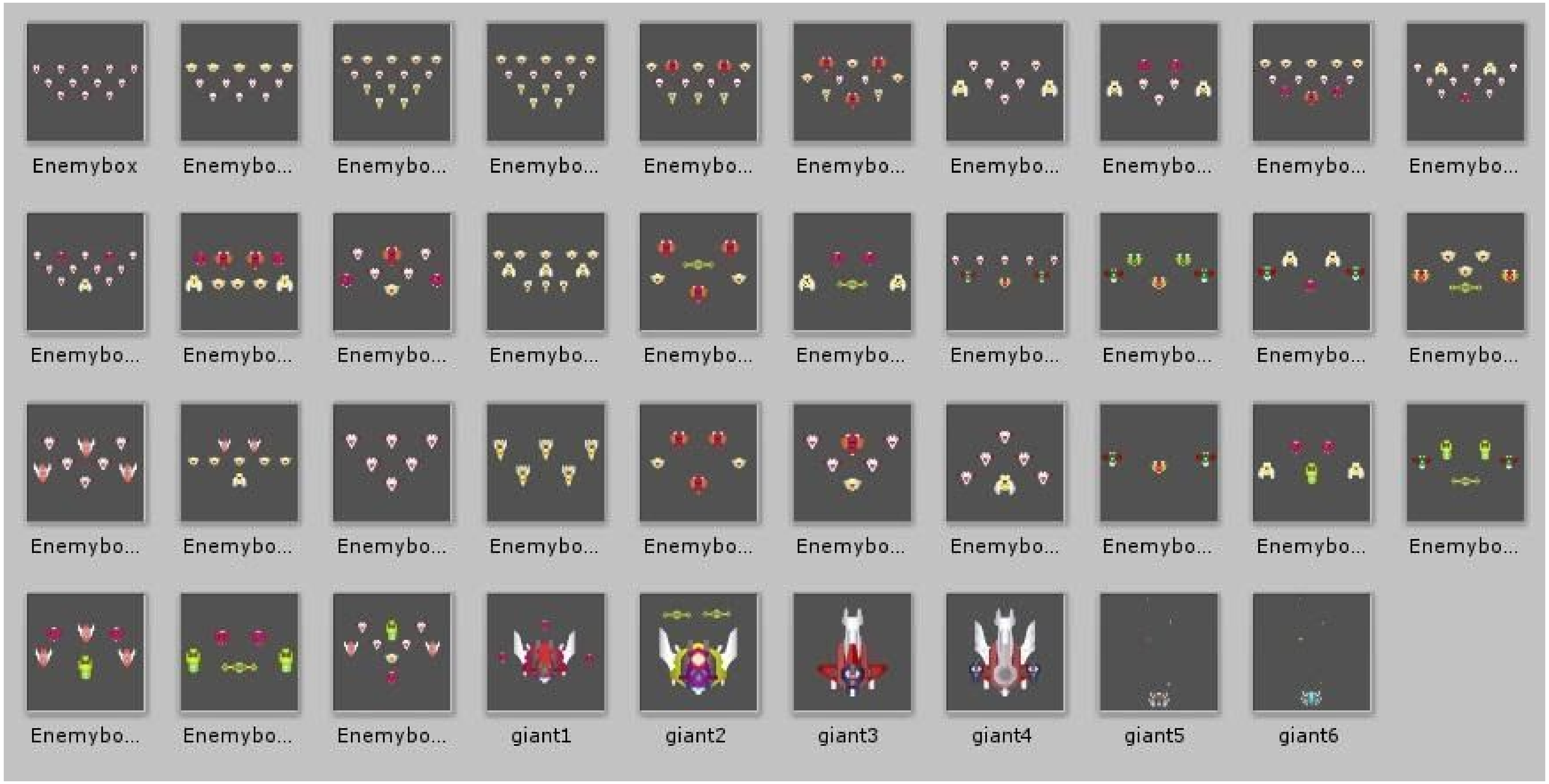
This code is applicable to most enemies and controls the number of lives shown by the Life Integrator, how to fire bullets or lasers, the time of firing, and the score of each enemy and ...



The type of shot fired is due to the four gameplay Laser Bullet Bullet2 Bullet3 objects and Targets for initializing the bullets, for example in this form the bullet starts moving at the target0 coordinates.
This code alone cannot work, but it does need to have access to Enemybox.cs code, and the gamepad that has this code should be given to Mymanage, as shown below.

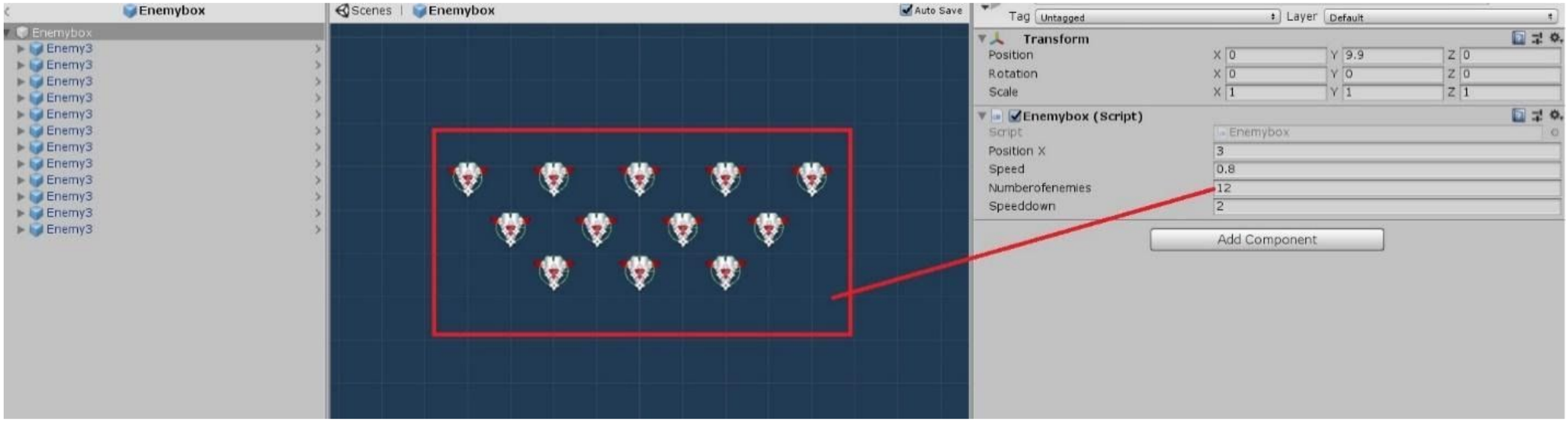


In this game the enemies are divided into separate groups as shown below, each group of enemies is controlled by Enemybox.cs code and with the destruction of all the enemies of each group is ordered to create one. Returns the new group in Managelevel.cs code.



Enemybox.cs code performance

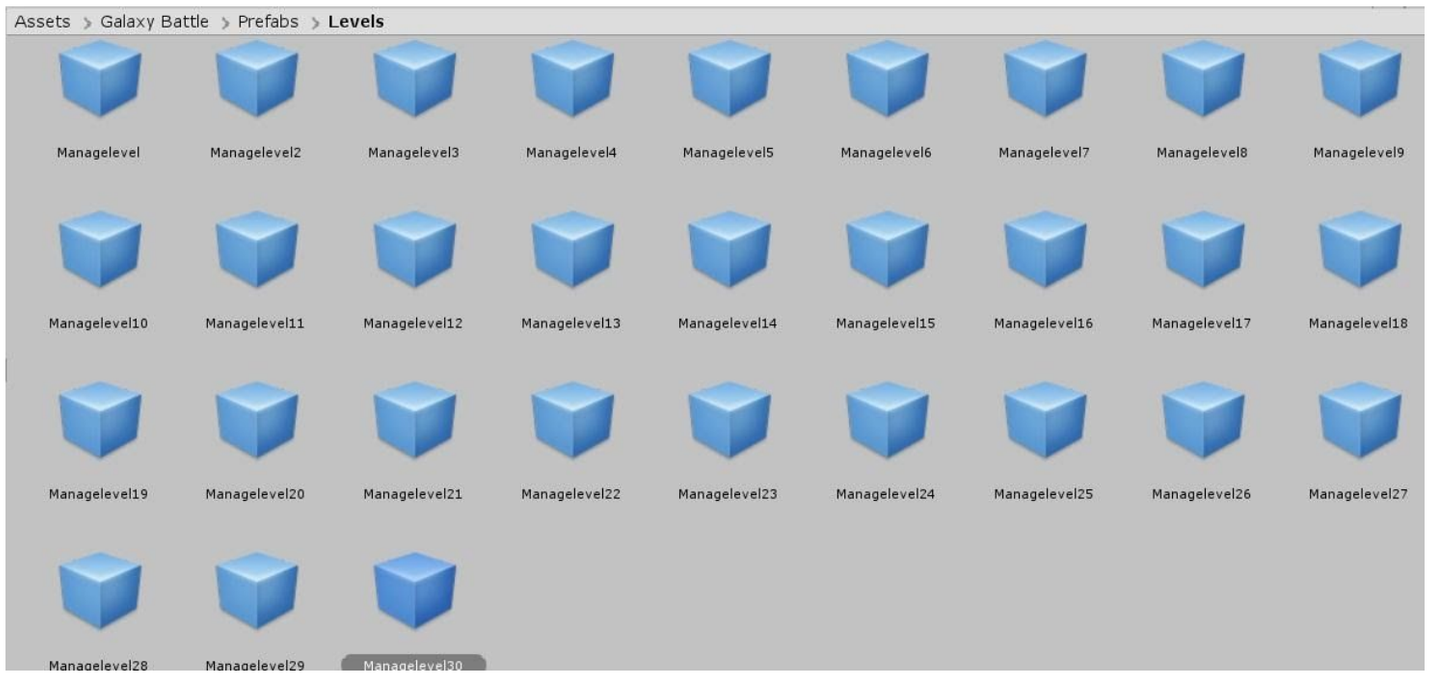
This code has a movement command and moves the enemies that have its subset with it. This moves down to Y = -3 and then moves left and right.
PositionX = Left and right moving rates
Speeddown = The amount of speed to move down

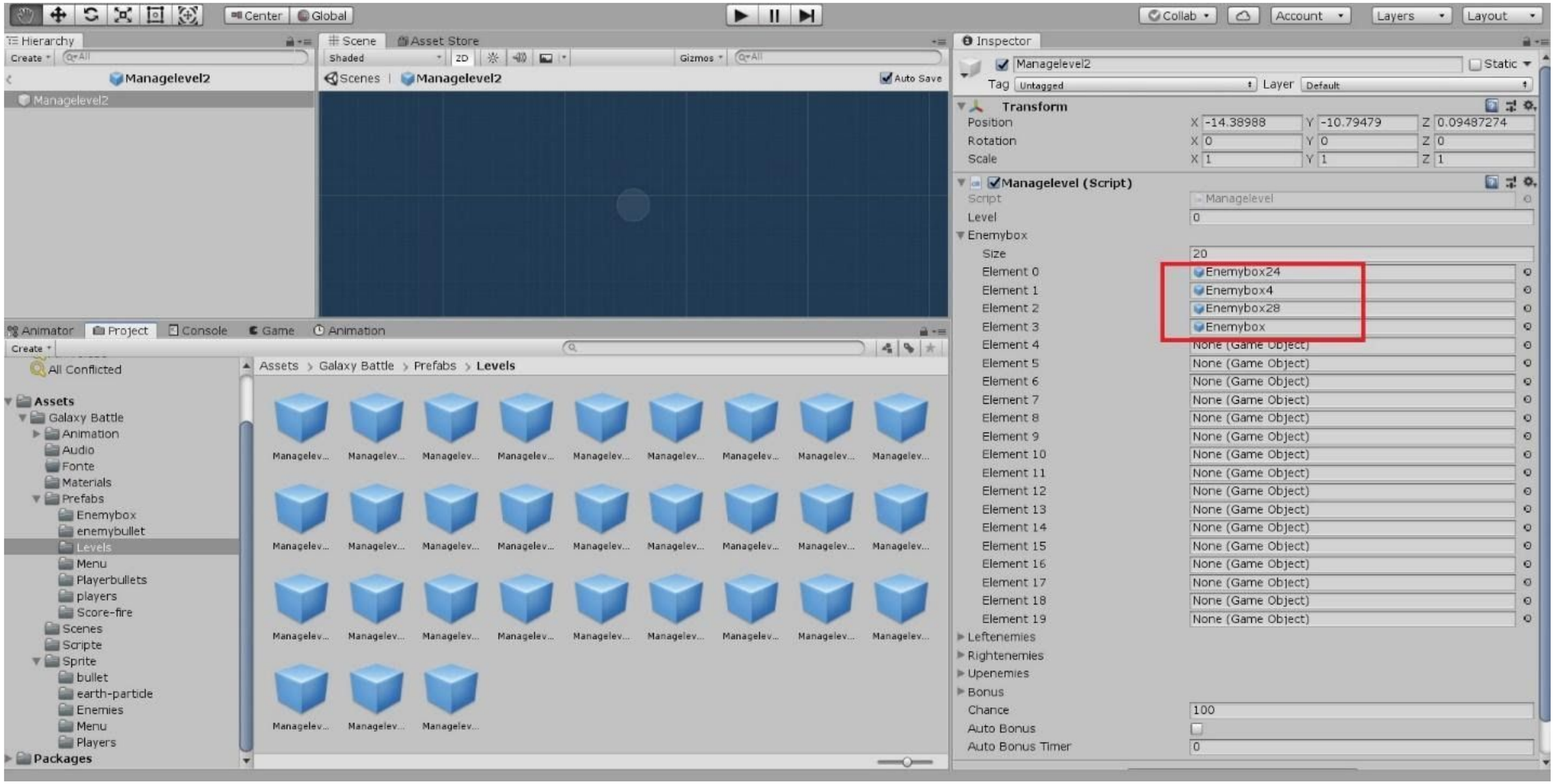


In the Numberofenemies section we need to write down the exact number of enemies we have put in Enemy.cs, which is inside the enemies after Enemybox.cs is destroyed and adds the number to the unit. For example, if these 12 enemies are destroyed in this picture, the Numberwill be equal to "Numberofenemies", 12, and will command a new group of enemies in the Managelevel.cs code.

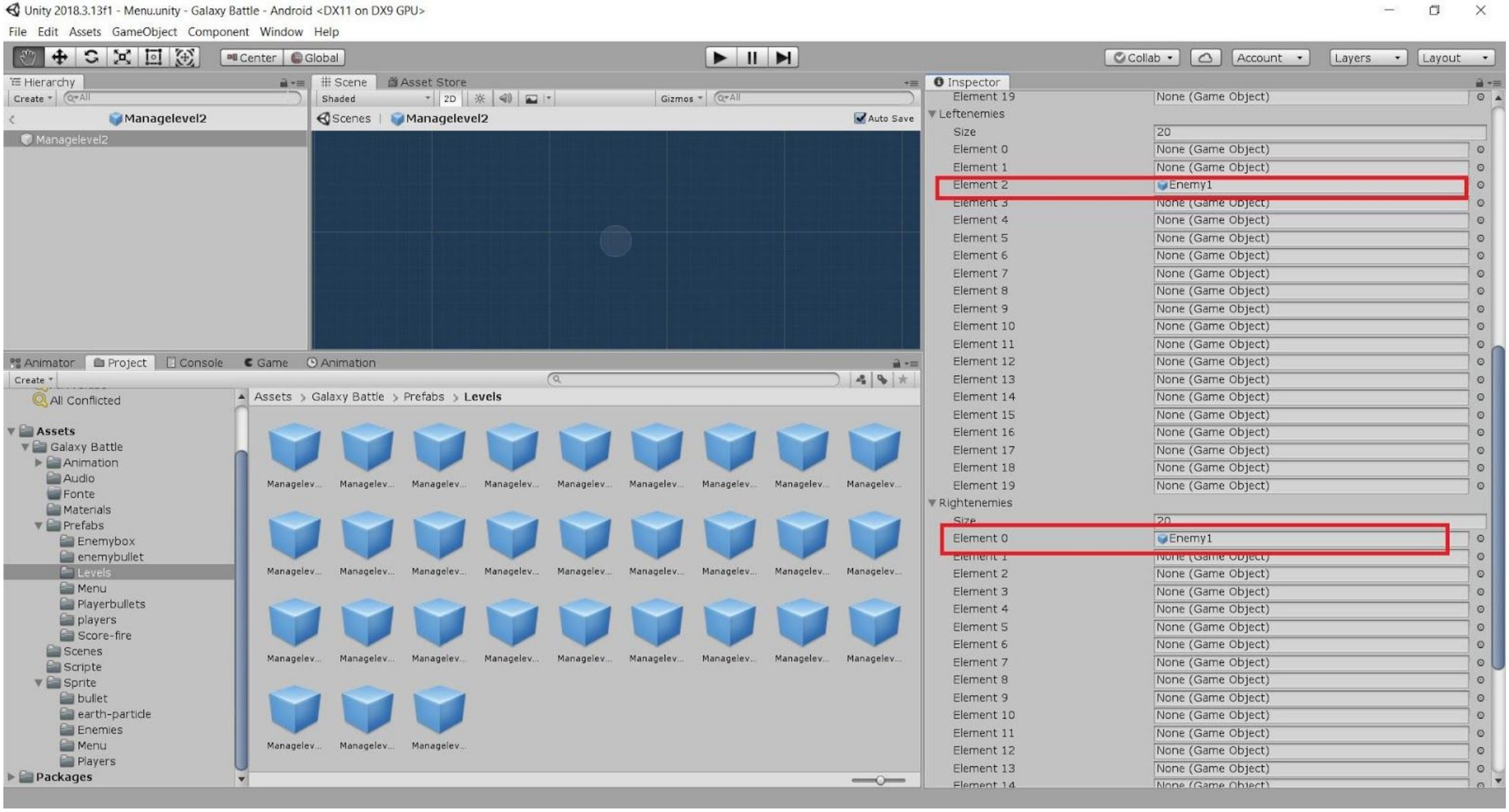
Managelevel.cs code performance

This code is the hinge we want to design, and inside it is a set of enemies with groups we've already designed.



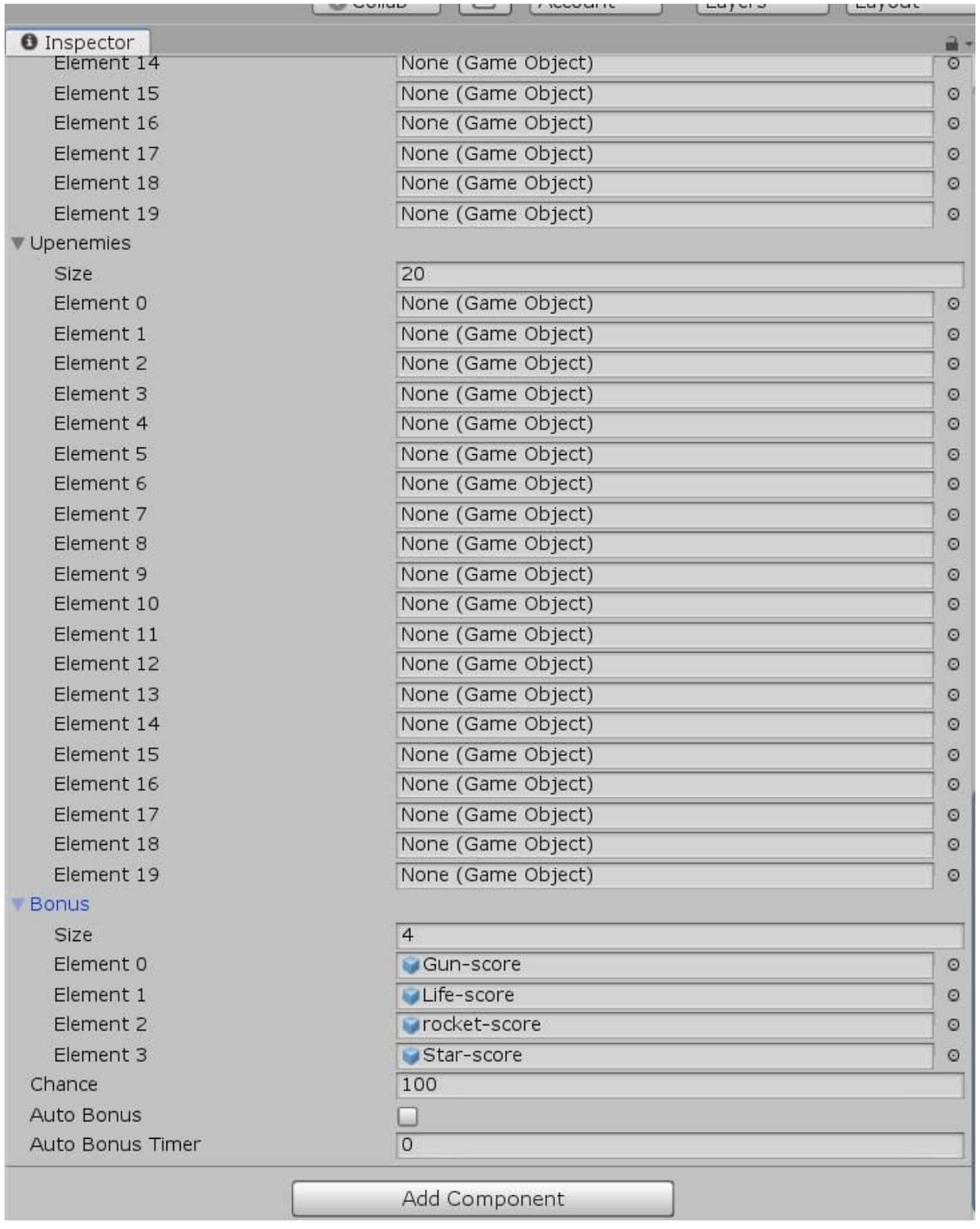


In this code, in the Enemybox section, we add as many enemy groups as we want to take turns. For example, in the above figure, in the second stage of the game you will encounter four groups of enemies and after completing the four groups will order the end of the level in Manage.cs.



In the Leftenemies and Rightenemies section you can put Enemy1, this game object can enter the screen from the left or right of the screen, and with what group of Enemyboxes it comes down to putting it in the desired render. For example, in level two, with the first group of enemies, another series of enemies appear from the right, and in the third group of enemies, a series of enemies from the left ... as in the figure below.

This is the case with the Upenemies, with the exception of Enmey5 instead of Enemy1.



In-game bonuses are controlled by Bonus, Chance, Autobonus and AutobonusTimer.
For example, in the figure above, each enemy group you come up with a prize, and this random prize is selected from the four available prizes.
If Chance is 100%, that is, each enemy group gets 100% a bonus, so Chance stands a chance of winning.

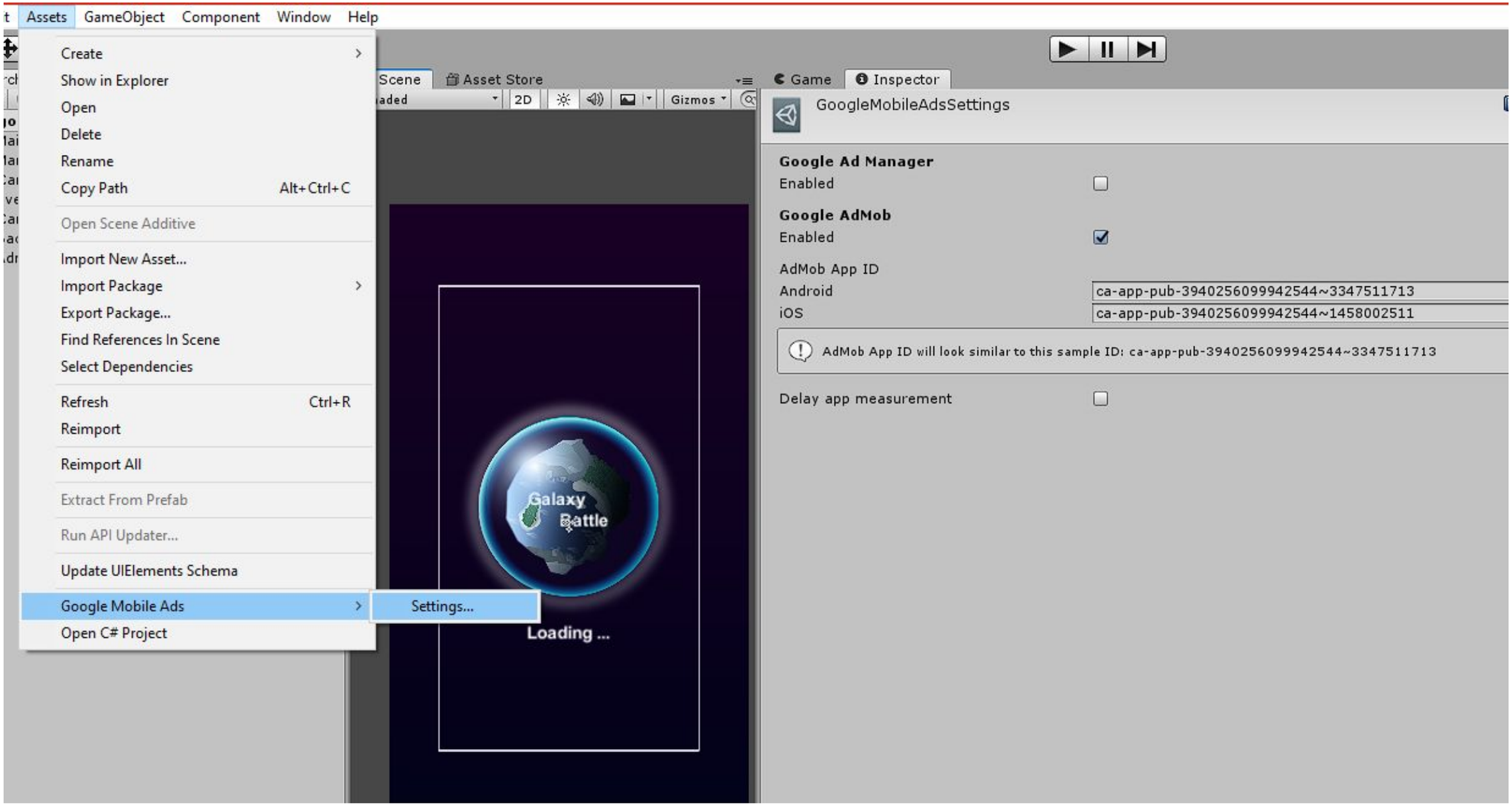
But if AutoBonus is enabled, it will give you a bonus for the Autobonustimer and a 100% chance of Chance.
Such as 5 and 10 and 15 and 20 and 25 and 30 pipe.

Finally, all enemies are destroyed by accessing Manage.cs, giving the command to end the level.

```
        Level += 1;

        if (Enemybox[Level]!=null)
        {
             Instantiate(Enemybox[Level], new Vector3(0, 11, 0), Quaternion.identity);
            instantiatebonus();
        }
        if (Rightenemies[Level] != null)
        {
            instantiatRight = true;

        }
        if (Upenemies[Level] != null)
        {
            instantiatUp = true;

        }
        if (Leftenemies[Level] != null)
        {
            instantiatLeft = true;

        }
        if (Enemybox[Level] == null)
        {
            Manage.Endgame();
            Destroy(gameObject);

        }
    }
}
```

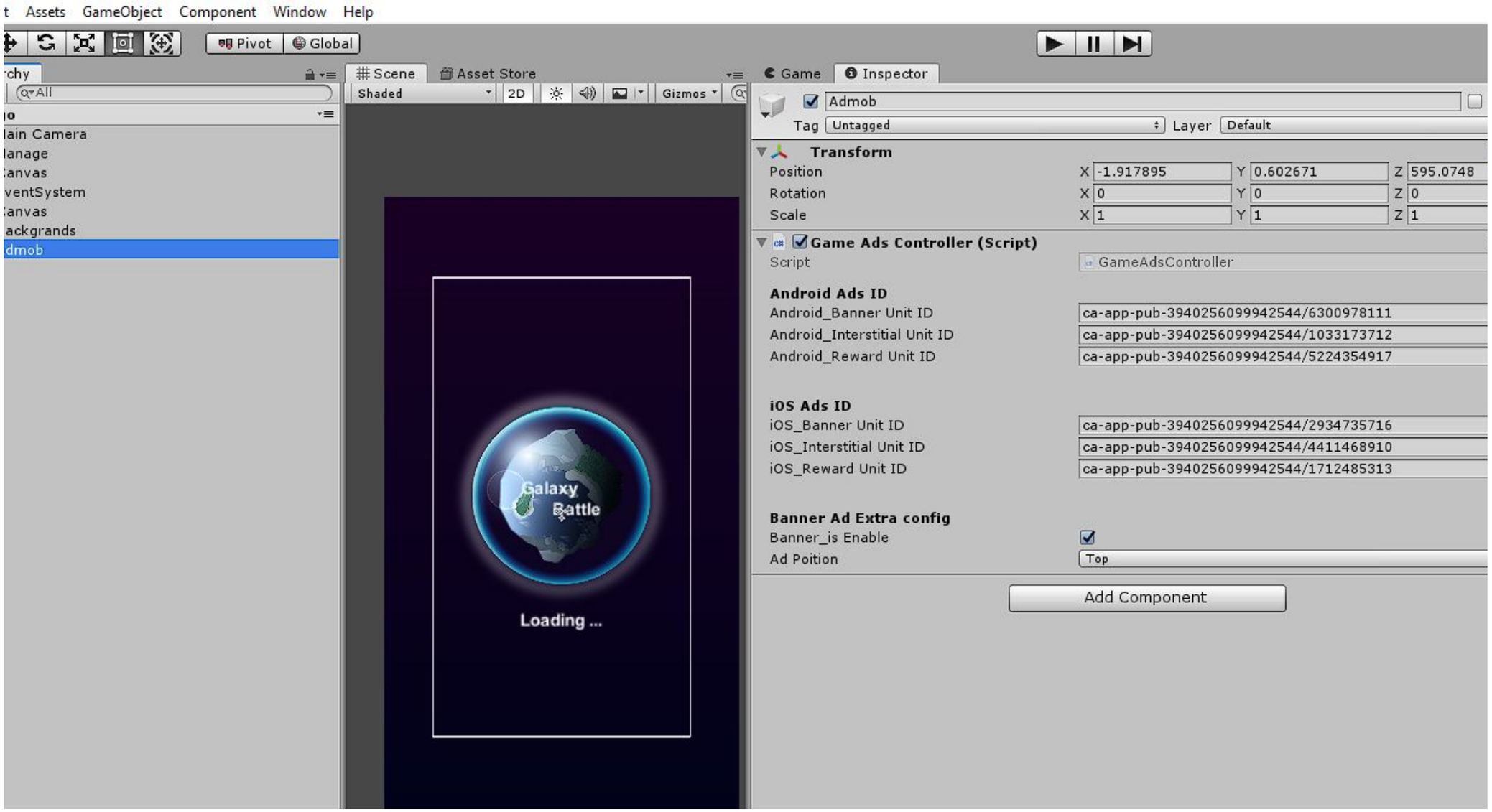
Admob

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.



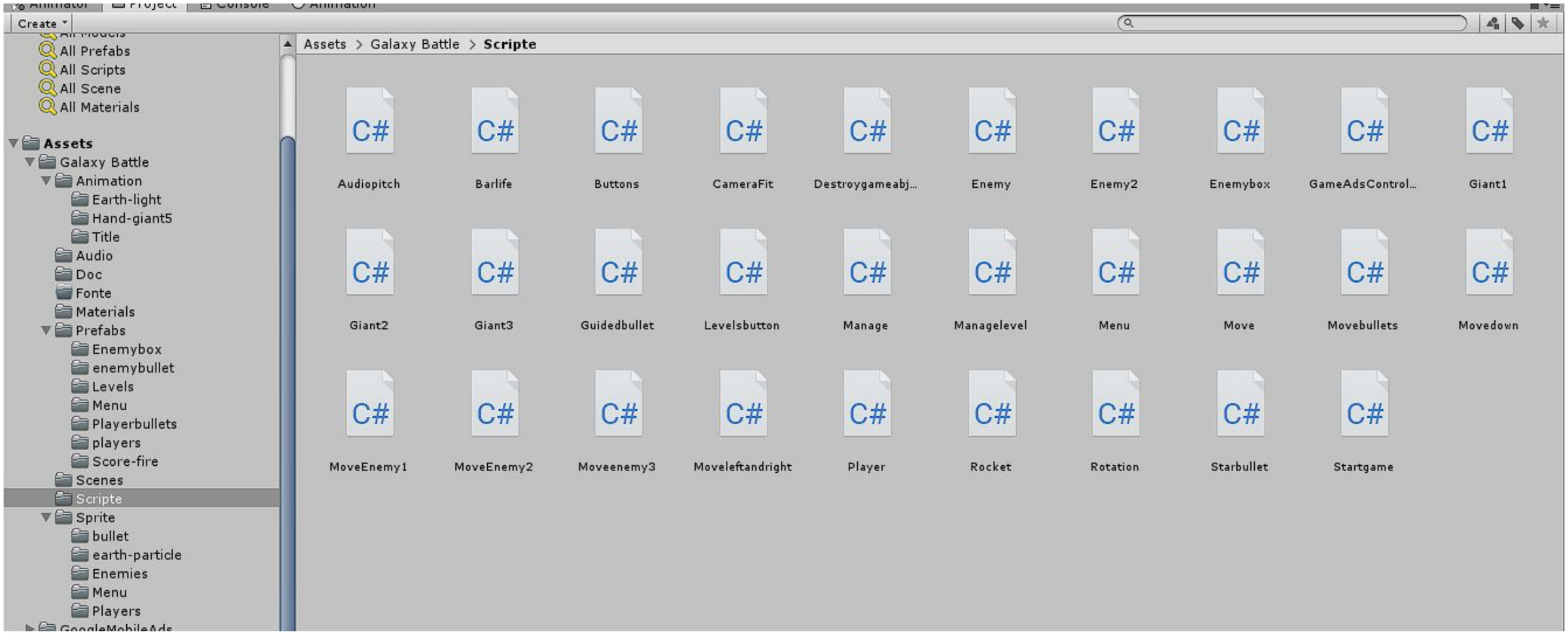
Then as shown in the below image, click on AdmobAds in the Logo scene and in the inspector, enter the Ad Unit IDs for banner and full screen ads for Android and iOS.

And if you don’t need banner advertisements, you can uncheck the Banner_is Enable option and you can also change the position of the ad from Ad Position section.



Scripts

All scripts are under the Scripts folder.



Audiopitch.cs
This code controls the laser when the game is paused.

Barlife.cs
This code controls the player's life.

Buttons.cs
This code controls the game's buttons.

CameraFit.cs
This code controls the camera's size.

Destroygameabjects.cs
This code is for destroying objects and sometimes controlling their sounds.

Enemy.cs Enemy2.cs Giant1.cs Giant2.cs Giant3.cs
These codes control the enemies which include their attacks and lives.

Enemybox.cs
This code controls a series of enemies that come in a group and when all of them are destroyed, the command for creating a new set of enemies is issued in managelevel.

Guidedbullet.cs Starbullet.cs Movebullets.cs
These codes control the bullets movements.

Levelsbutton.cs
This code is for controlling the buttons of levels' menu.

Manage.cs
This code controls general aspects of the game which includes sounds, score, life and saves.

Managelevel.cs
This code controls the set of enemies that should appear in a level.

Menu.cs
This code controls the menus of game's UI.

Move.cs
This code controls the player's movements.

Movedown.cs
This code is for moving down in objects.

MoveEnemy1.cs MoveEnemy2.cs MoveEnemy3
This code controls the type of enemies' movements.

Moveleftandright.cs
This code is for moving left and right in objects.

Player.cs
This code controls firing, collisions and sprites for player.

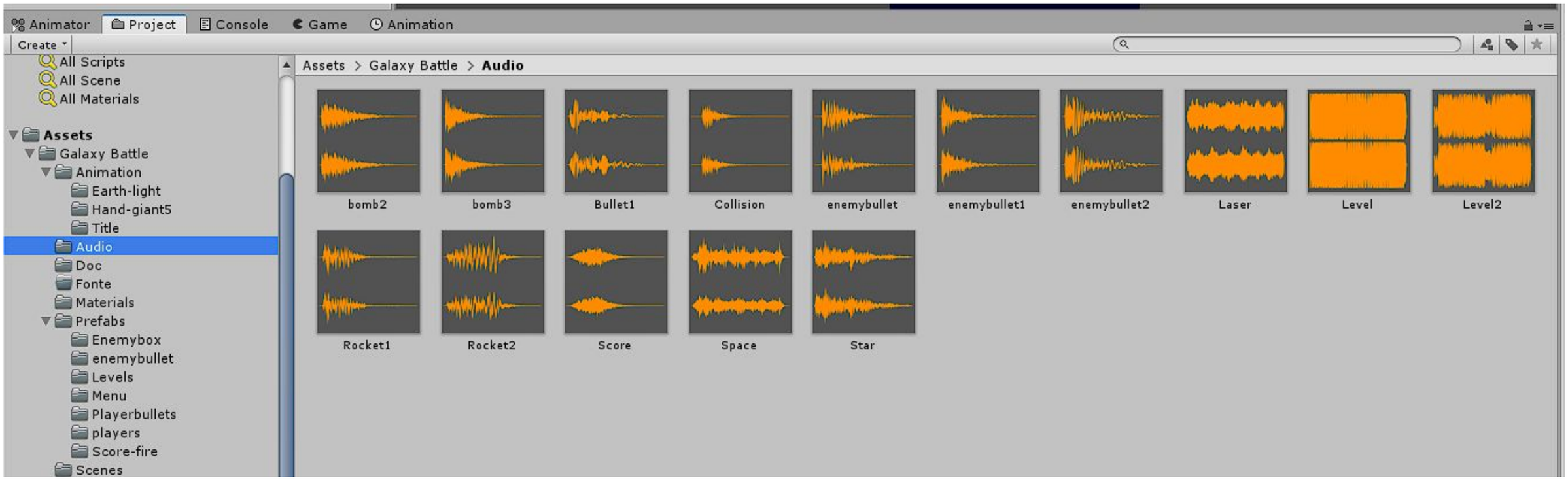
Rocket.cs
This code controls the missiles movements and collision.

Rotation.cs
This code controls rotation of objects.

Startgame.cs
This code is for starting the game.

Music and FXs

To change the game 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 default images, new images should have the same size and name as default ones.