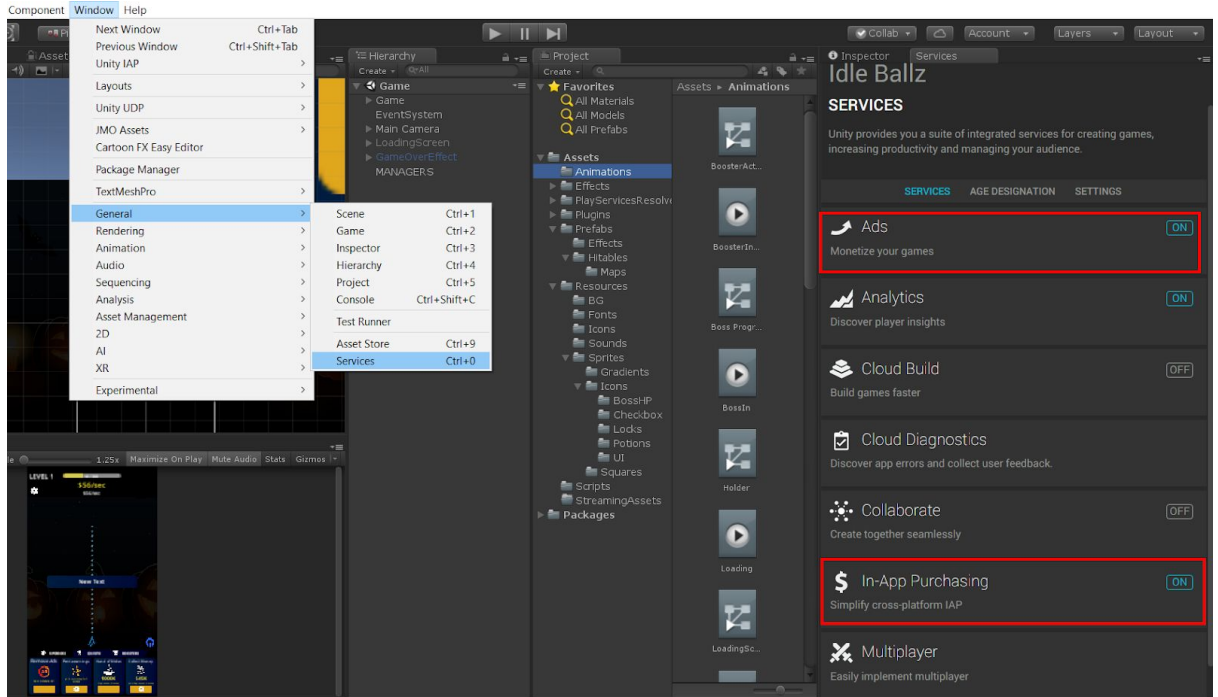


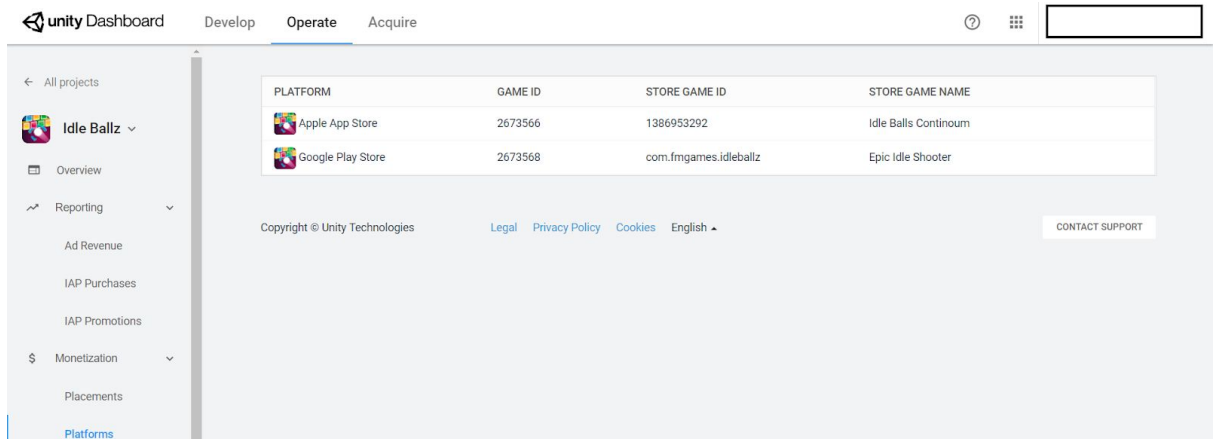
# Monetization guide

## ADs:

1. Make sure to enable Ads and IAP in the services menu



2. Copy the ids from Unity Dashboard



```

10
11 public Text adMoneyText; //For the Watch ad to get money Booster
12
13 public static Ads _instance; //instance for easier usage
14 public decimal moneyToAdd; //For double money (in upgrade / offline reward)
15
16 public GameObject glowExplosion;
17
18 public float timeToShowInterstitial = 300; //How often shows up a full-screen ads (default is 300 seconds)
19
20 private void Awake() {
21     _instance = this;
22 }
23
24 // Use this for initialization
25 void Start() {
26     #if UNITY_IOS
27         Advertisement.Initialize("2673566");
28     #elif UNITY_ANDROID
29         Advertisement.Initialize("2673568");
30     #endif
31
32     StartCoroutine(UpdateTexts());
33 }
34
35 // Update is called once per frame
36 void Update() {
37     adTimer += Time.deltaTime;

```

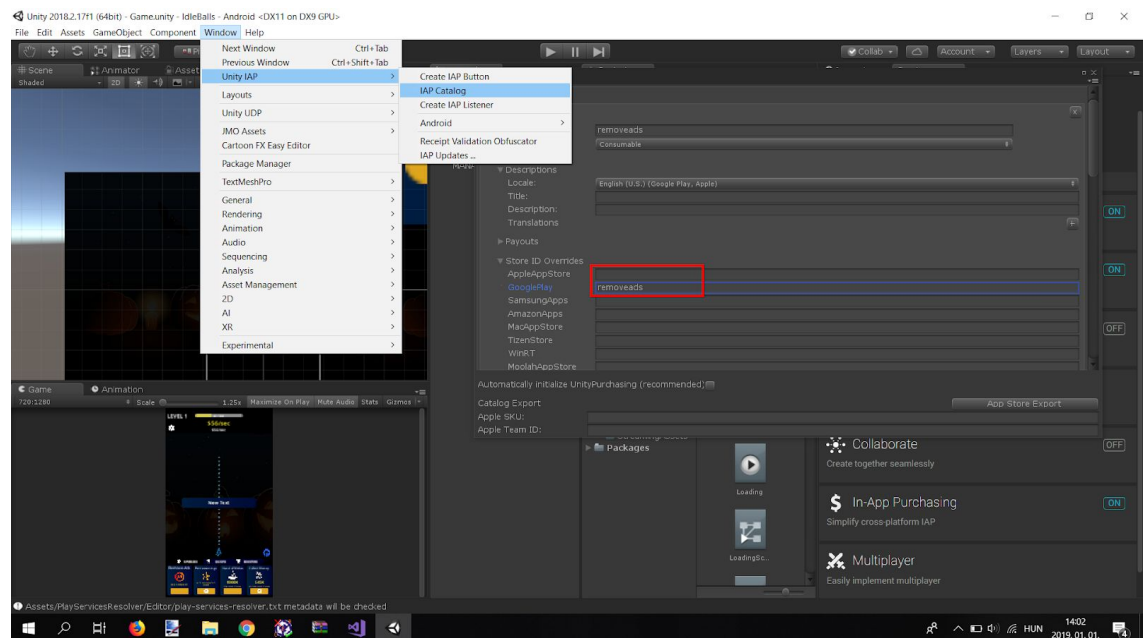
And paste the values for Android and IOS. Now the ADs shown in the game will be connected to your account.

## Monetization:

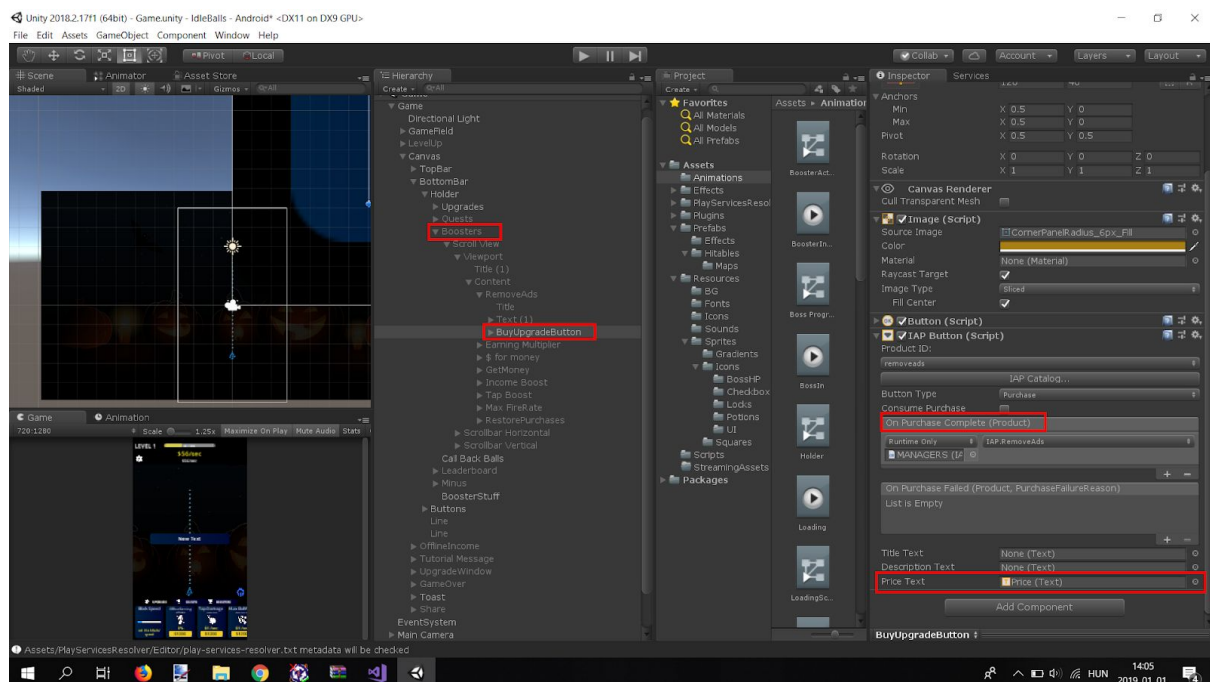
1. Create the products in your Itunes / Google play account

FireRate Booster (fire_rate)	HUF 999	2018. szept. 4.	Aktiv
FireRate Booster (firerate)	HUF 949	2018. nov. 13.	Aktiv
Hand of Midas (handofmidas)	HUF 650	2018. szept. 4.	Aktiv
Income Boost (income_boost)	HUF 650	2018. szept. 4.	Aktiv
Income Boost (incomeboost)	HUF 499	2018. nov. 13.	Aktiv
Launch Offer (launch_offer)	HUF 1 649	2018. nov. 4.	Aktiv
Remove the ads (remove_ads)	HUF 500	2018. szept. 4.	Aktiv
Remove the ads (removeads)	HUF 299	2018. nov. 13.	Aktiv

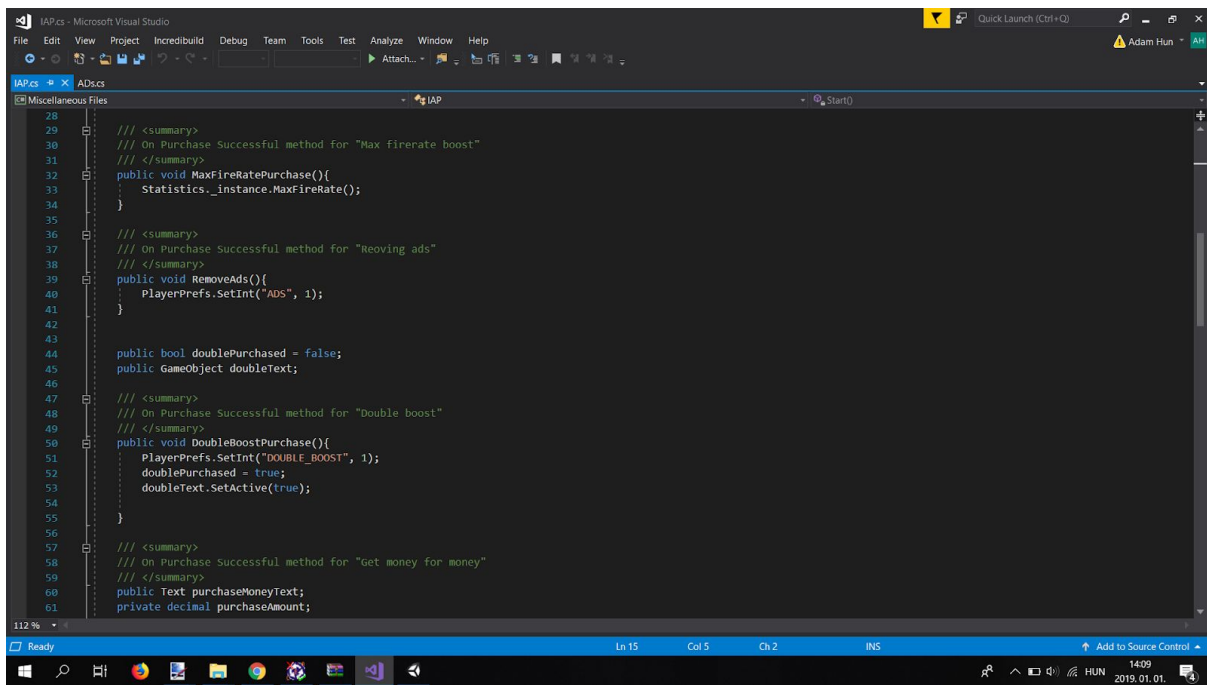
2. Paste these IDs to the Unity IAP Catalog in the editor - Be sure to copy the correct ids and add them to the Store ID Overrides section for Appstore and GooglePlay



3. You can modify the current IAPs. Every IAP is under the Booster gameobject in the inspector. On the IAP Button (script) the OnPurchaseComplete method is called when a successful purchase occurs. The Price Text is automatically overwritten by the exact price of the item converted to local currencies.



#### 4. Successful purchases are handled in IAP.cs



```
IAP.cs - Microsoft Visual Studio
File Edit View Project Incremental Build Debug Team Tools Test Analyze Window Help
IAP.cs IAP Start()
Miscellaneous Files
28
29 /// <summary>
30 /// On Purchase Successful method for "Max fireRate boost"
31 /// </summary>
32 public void MaxFireRatePurchase(){
33     Statistics.Instance.MaxFireRate();
34 }
35
36 /// <summary>
37 /// On Purchase Successful method for "Reoving ads"
38 /// </summary>
39 public void RemoveAds(){
40     PlayerPrefs.SetInt("ADS", 1);
41 }
42
43
44 public bool doublePurchased = false;
45 public GameObject doubleText;
46
47 /// <summary>
48 /// On Purchase Successful method for "Double boost"
49 /// </summary>
50 public void DoubleBoostPurchase(){
51     PlayerPrefs.SetInt("DOUBLE_BOOST", 1);
52     doublePurchased = true;
53     doubleText.SetActive(true);
54 }
55
56
57 /// <summary>
58 /// On Purchase Successful method for "get money for money"
59 /// </summary>
60 public Text purchaseMoneyText;
61 private decimal purchaseAmount;
```