

HyperByte Studios – Unity IAP Setup

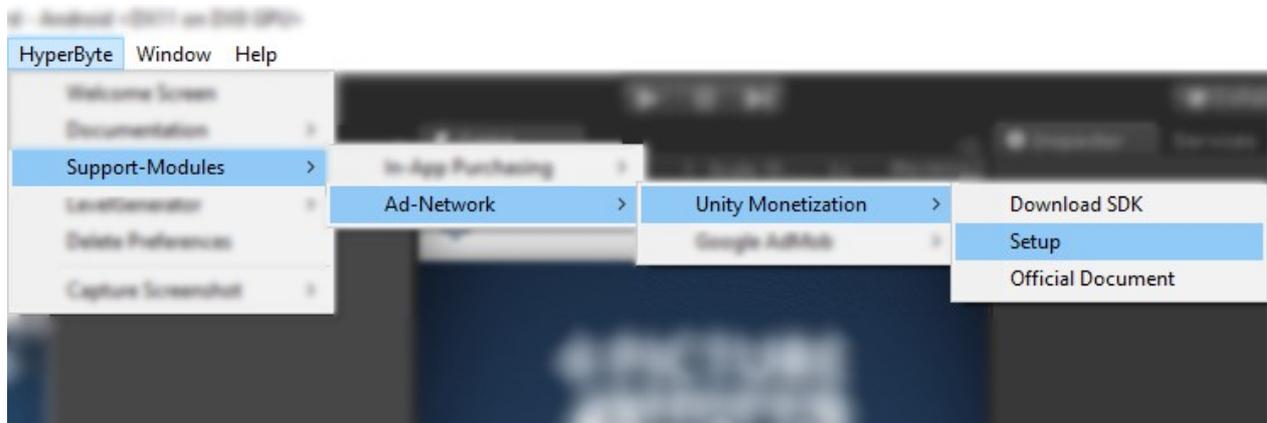
Hi There,

This package comes with pre-installed setup support for Unity IAP. Please follow below instructions for setting up ad network and make it work for your game.

Before getting Started, Please checkout IAPManager.cs script component, IAPManager.cs is a placeholder script and it will be replaced by another script which has actual Unity IAP Support SDK included.

◆ Setting Unity In-App Purchasing.

- ✓ Please, go-to Window ~> Services ~> Tab from unity'menu. Inside Services, You will see In-App Purchasing. Please enable it and complete setup.
- ✓ You'll be prompted to import Unity IAP SDK, please import it within your project.
- ✓ After SDK import completion, Please go-to HyperByte ~> Support-Modules ~> In-App Purchasing ~> Setup. It will import a unitypackage which will replace the IAPManager Script. You are almost done now.
- ✓ Import SDK after getting downloaded.
- ✓ After it gets imported, Please Go to HyperByte Menu Item ~> Support-Modules ~> Ad-Network ~> Unity Monetization ~> Setup. [See Ref. Image below]



◆ Customization and Setting In-App SKUs.

- ✓ You have full control over customization of SKUs, you can add, modify, remove, add any SKUs. Please follow below steps.

Setting Up SKUs [In-App Products]:

- Finalize you app specific InAppSKUs.
- Open IAPManager.cs and Look at InitIAP() method. You'll notice Few In-app-SKUs are added in for the initialization with PayoutDefination. Please replace those SKUs and rewards with your choice of SKUs and rewards. [See below image]

```

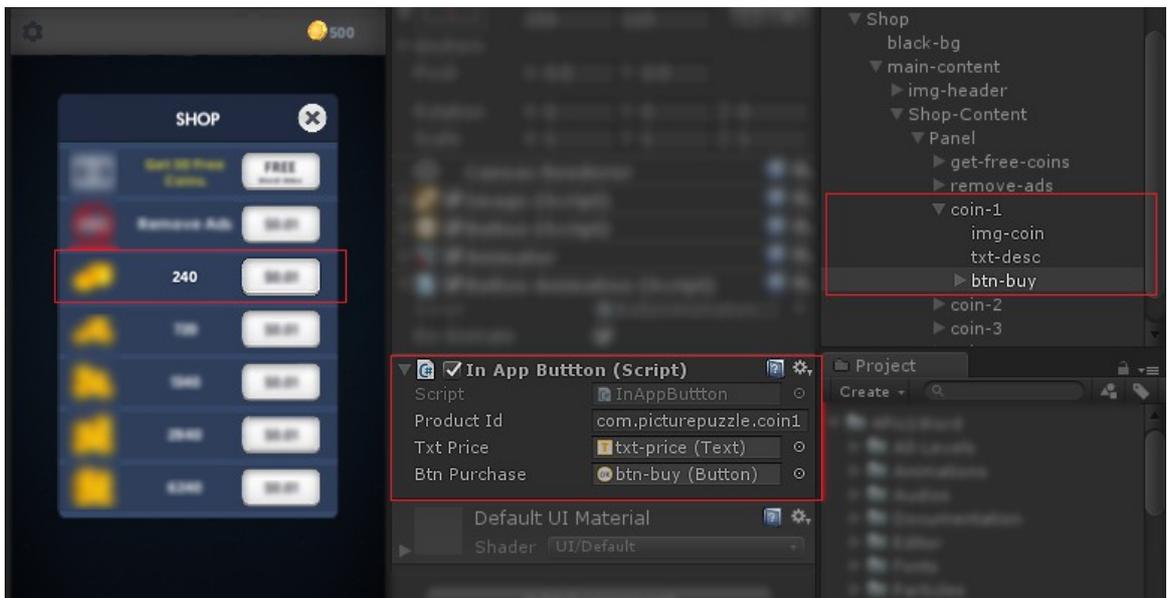
IAPManager.cs x
21  /// Start is called on the frame when a script is enabled just before any of the Update methods is called the first time.
22  void Start() {
23      InitIAP();
24  }
25
26  /// Setup youy SKUs and rewards here. Payout Defination is the amout of rewards user will get on successfully completion of purchase.
27  public void InitIAP() {
28      var builder = ConfigurationBuilder.Instance(StandardPurchasingModule.Instance());
29
30      builder.AddProduct("com.picturepuzzle.removeads", ProductType.Consumable, new IDs {
31          {"com.picturepuzzle.removeads", GooglePlay.Name}, {"com.picturepuzzle.removeads", AppleAppStore.Name}
32      }, new PayoutDefinition("Other", "adfree", 1));
33
34      builder.AddProduct("com.picturepuzzle.coin1", ProductType.Consumable, new IDs {
35          {"com.picturepuzzle.coin1", GooglePlay.Name}, {"com.picturepuzzle.coin1", AppleAppStore.Name}
36      }, new PayoutDefinition("Currency", "coin", 240));
37
38      builder.AddProduct("com.picturepuzzle.coin2", ProductType.Consumable, new IDs {
39          {"com.picturepuzzle.coin2", GooglePlay.Name}, {"com.picturepuzzle.coin2", AppleAppStore.Name}
40      }, new PayoutDefinition("Currency", "coin", 720));
41
42      builder.AddProduct("com.picturepuzzle.coin3", ProductType.Consumable, new IDs {
43          {"com.picturepuzzle.coin3", GooglePlay.Name}, {"com.picturepuzzle.coin3", AppleAppStore.Name}
44      }, new PayoutDefinition("Currency", "coin", 1344));
45
46      builder.AddProduct("com.picturepuzzle.coin4", ProductType.Consumable, new IDs {
47          {"com.picturepuzzle.coin4", GooglePlay.Name}, {"com.picturepuzzle.coin4", AppleAppStore.Name}
48      }, new PayoutDefinition("Currency", "coin", 2940));
49
50      builder.AddProduct("com.picturepuzzle.coin5", ProductType.Consumable, new IDs {
51          {"com.picturepuzzle.coin5", GooglePlay.Name}, {"com.picturepuzzle.coin5", AppleAppStore.Name}
52      }, new PayoutDefinition("Currency", "coin", 6240));
53
54      UnityPurchasing.Initialize (this, builder);
55  }

```

- As you complete it, You need to follow a simple step make your own SKU working.

Setting Up InAppButton :

- InAppButton.cs is responsible for handling purchase request when user presses the Buy button for your SKU. Please see below screenshot which will make it more clear.



- As shown in the screenshot, please set your Sku against all the InApp Button in the shop screen as well as places where you want to set your IAPs.
- Over completion of these steps, your in-app should be working in your game.

Restore IAP :

- Please call `IAPManager.Instance.RestoreAllManagedProducts()` Method for restoring all managed/non-consumable in-apps. For this game template, There is a restore button in the Setting screen provided which will restore transactions.

◆ **NOTES :**

- ✓ You need to set up in-app SKUs in respective store listing(iOS, Google Play or any other). After that only in-apps will work within your app.
- ✓ Please follow [this link](#) for an in-depth understanding of Unity IAP SDK.
- ✓ We are not using Codeless-IAP with this game template, you can use it in case you like and you've knowledge about implementing it.

~~~~~

### **Few Additional Note:**

---

- ◆ We did a lot of hard work to make this asset, so your review and feedback is most important to us. If you like asset please rate it on the asset store, we really appreciate.
- ◆ If you notice any bug or issue, please let us know via our support email address, we'll try to fix it ASAP.
- ◆ We are working on more beautiful assets already, so please be sure to check out our other game templates.
- ◆ We also take freelance project for unity, we have a talented and passionate team of people who will help you make your dream game, be sure to get in touch with us.

➔ **Support Email** : [support@hyperbytestudios.com](mailto:support@hyperbytestudios.com)

➔ **Bussiness Email** : [contact@hyperbytestudios.com](mailto:contact@hyperbytestudios.com)

➔ **Website** : <http://www.hyperbytestudios.com/index.html>