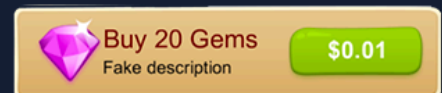
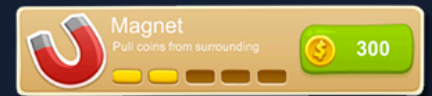
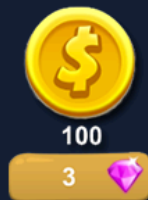


iap Manager Pro

In-app Purchase and Game Data Manager

In-App Purchase | Virtual Currency
Inventory | Game Level | Power-up
Flexible UI Template System



IAP Manager Pro

In-app Purchase and Game Data Manager

Document version 1.0.0

Support email: support@digicrafts.com.hk

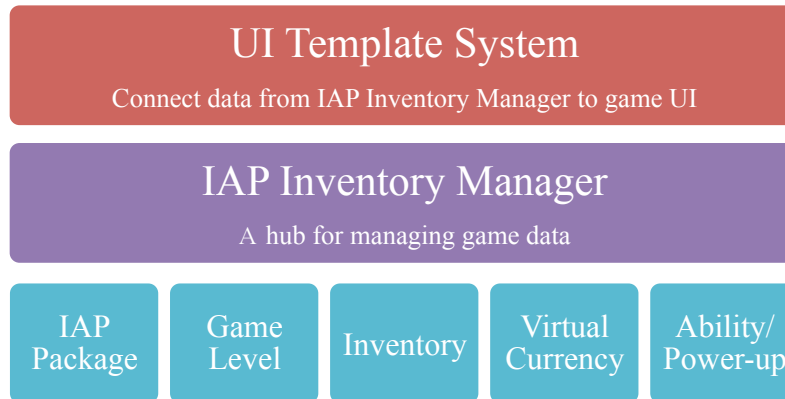
Contents

I. INTRODUCTION	4
II. BEFORE START	6
III. QUICK START: SETUP VIRTUAL CURRENCY	7
IV. QUICK START: SETUP IN-APP PURCHASE	10
Define in-app purchase packages	10
Create a list of diamond purchase button	13
V. QUICK START: SETUP ABILITY/POWER-UP	16
Define a power-up with 5 upgradable levels	16
Create the UI	19
VI. QUICK START: SETUP INVENTORY	22
Define an inventory	22
Create an inventory purchase button	24
Create a button for consume the inventory	26
VII. QUICK START: SETUP GAME LEVEL	28
Define an Game Level	28
Create the game level item template	30
Create the game level list view	32
VIII. IAP MANAGER PRO INSPECTOR	34
IX. DATA STORE TYPE	38
X. TEMPLATES	39
UI Text	39

UI Consume Button	40
UI Template	41
UI List View	42
Dialog Template	43
Template naming convention	44
XI. EVENTS	45
XII. API	46
IAPInventoryManager Class	46
IAPCurrency, IAPInventory Class	46
XIII. OTHER SETTINGS	47
Enabled Test Mode (Amazon Only)	47
Prepare for Amazon Store Submit	47

I. Introduction

IAP Manager Pro

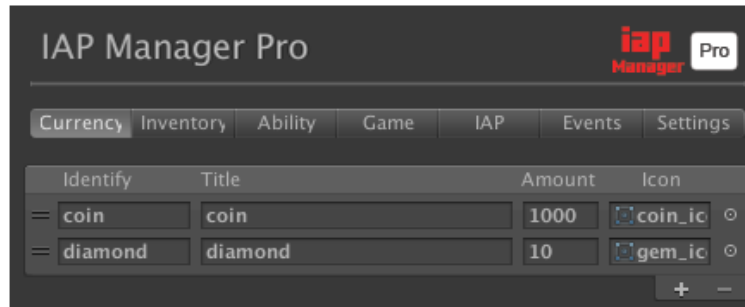


IAP Manager Pro is a total solution for in-app purchase and game data management. It provides an easy way to manage the in-app purchase, virtual currencies, inventories, game level data and player properties in the game. With a powerful UI template system for displaying managed object data.

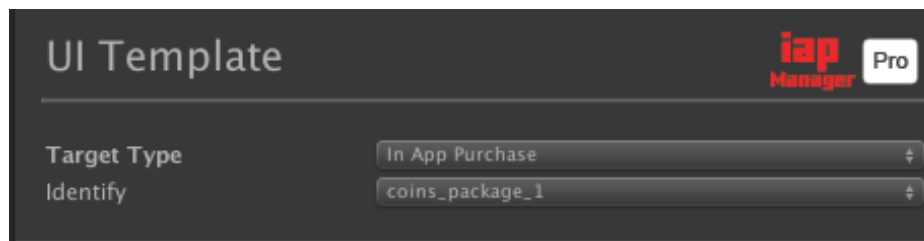
Simple and Easy Workflow



IAP Manager Pro enables an easy workflow for setup an In-App Purchase system and essential game components. To ensure an clean and easy system for developer using in their game.



IAP Manager Pro comes with an easy to use editor for managing game data. You can use it to manage the entire game data and IAP product. Minimal coding is needed for setup the system. Beside the inspector, **IAP Manager Pro** also comes with a static class provide full set of API.



IAP Manager Pro comes with a powerful template system use with Unity UI (extend to other UI such as NGUI in future version). The template system includes several scripts for different purpose such as **UI Text**, **UI Consume Button**, **UI List View** and **UI Template**. The template system provided an easy but flexible workflow to extend you creativity on game UI design.



II. Before Start

1. Add products for in-app-purchase in apps store.

In order to use the **IAP Manager Pro** with real money purchase. You need to define products in Apps Store. Follows the steps below for setup procedures.

Follows the tutorial for each store in below links.

iTunes Connect (iOS):

<https://www.raywenderlich.com/105365/in-app-purchases-tutorial-getting-started>

Play Store (Android):

<http://www.theappguruz.com/blog/implement-in-app-purchase-version-3> (See Part 4)

Amazon (Kindle):

<https://developer.amazon.com/public/apis/earn/in-app-purchasing/docs-v2/submitting-copy>

2. Setup Unity Purchasing Services.

You can follow the tutorial below to setup Unity Purchasing Services (The scripting part is not requested).

<http://docs.unity3d.com/Manual/UnityIAPSettingUp.html>

3. Download and import Digicrafts IAP Manager Pro assets package from asset store.

4. Try examples. You can found the examples in “Digicrafts/IapManagerPro/Examples/” directory.

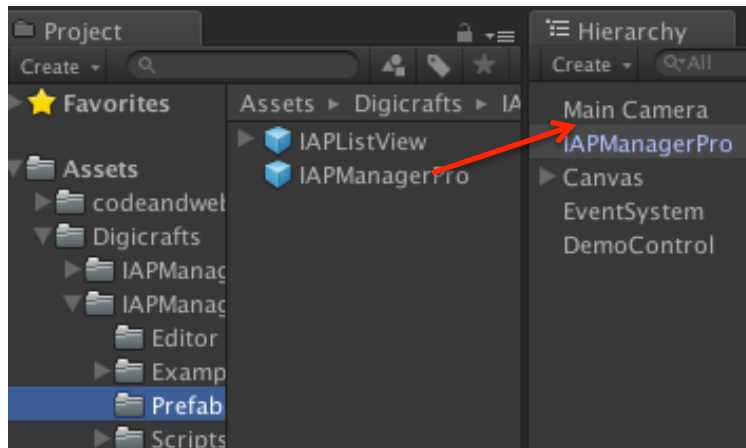
- **Demo_Game/Demo_Game_Start** – full example with in-app-purchase and game level selection.
- **Demo_UI/UI_Demo** – Show different template examples.

III. Quick Start: Setup Virtual Currency

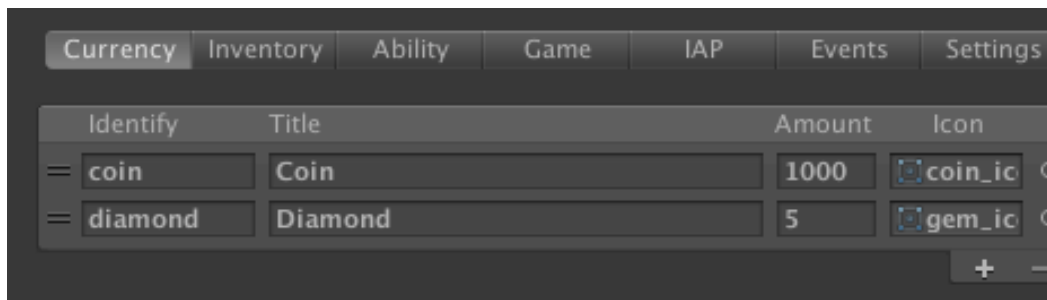
Virtual currency is the core of **IAP Manager Pro**. You need to setup as least one virtual currency before adding other object in the system. You can have one or multiple currencies. Currency can be use to purchase other inventories, upgrade ability or purchase other currencies.

In this section, you will learn the basic step for setup **IAP Manager Pro**.

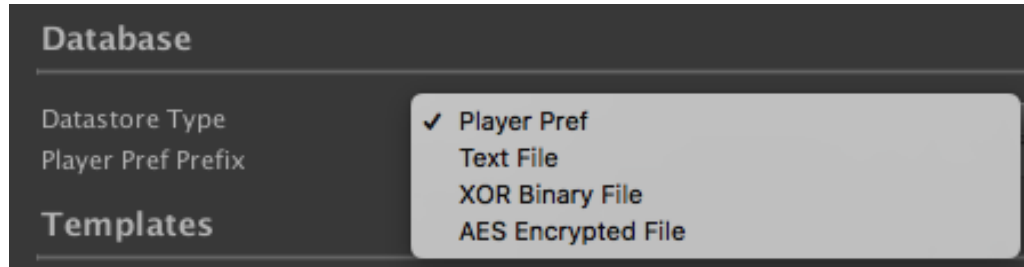
1. Drag and Drop the **IAP Manager Pro** prefab into the scene. You can find the prefab from “Digicrafts/IAPManagerPro/Prefabs/IAPManagerPro.prefab”. You only need one **IAP Manager Pro** instance in your game. Once it added, it will persistent across all the scenes.



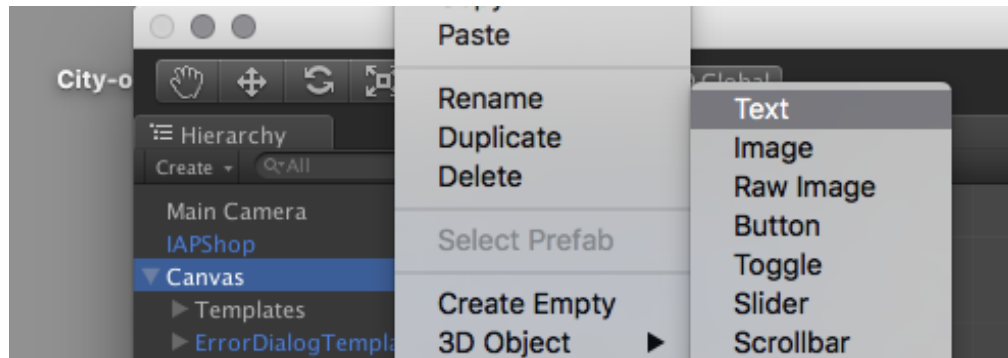
2. Select the **IAP Manager Pro** object and open inspector window. Under the **Currency** section, add a new currency by pressing the “+” button. Fill in the unique identify, title and default amount. You may also specify the icon, which use in template system.



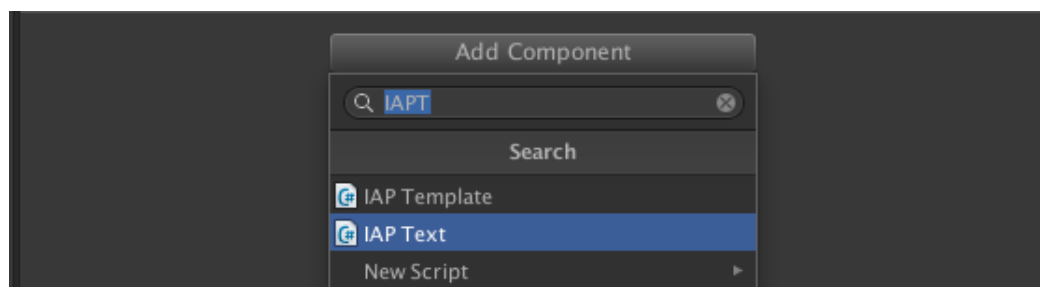
3. In the **Settings** section, select a data store type. **IAP Manager Pro** provides several options to store data including *Player Pref*, *JSON Text File*, *XOR Binary File* and *AES Encrypted File*. You can refer to Datastore section of this document and have more information about data store types.



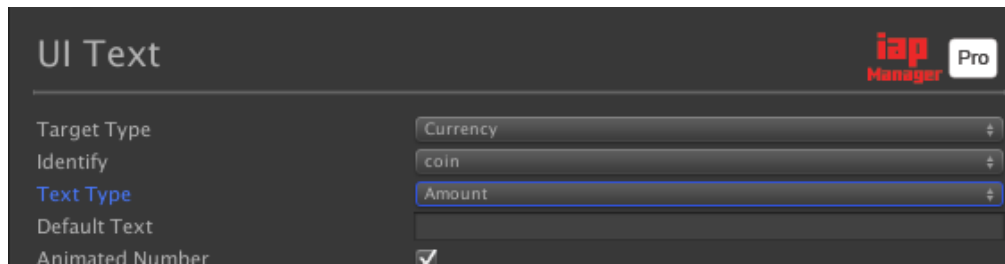
4. Now we add a text for displaying the amount of currency. Add a **UI Text** in the scene by right click on the hierarchy and select **UI>Text**.



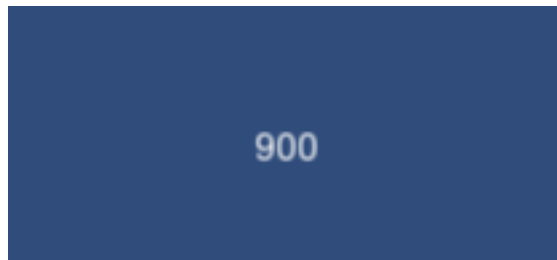
5. Select the **Text** in the hierarchy window. From the inspector, click on “**Add Component**” button and type “**IAP Text**” to add the script.



6. From the inspector window, select the “**currency**” as target type and “**coin**” in identify field. We want to display the amount of currency. So, select “**amount**” as text type”.

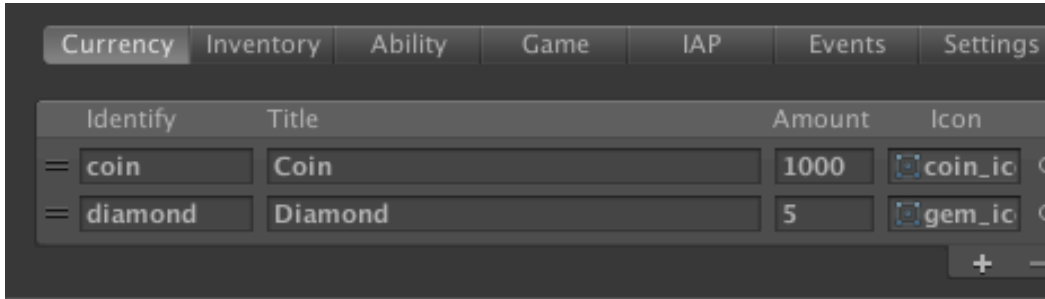


7. Basic setup is done. You can run and test. The amount of “coin” will show.



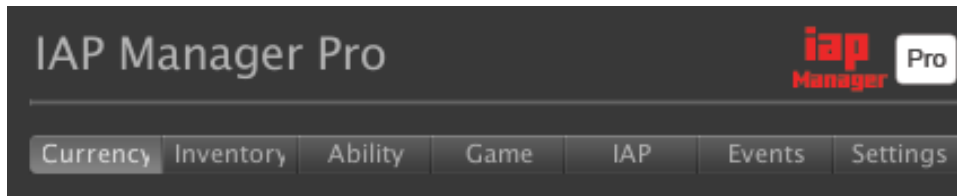
IV. Quick Start: Setup In-App Purchase

IAP Manager Pro provides an easy workflow for intergrading virtual currency and real money in-app-purchase. Assume we have 2 virtual currencies, coin and diamond. We will setup 3 packages for buying diamonds with real money and 3 packages for buying coins with diamonds.

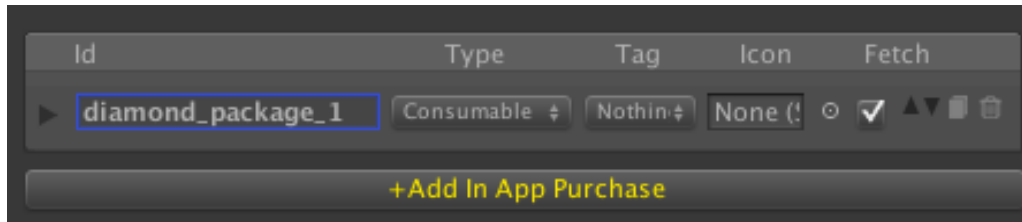


Define in-app purchase packages

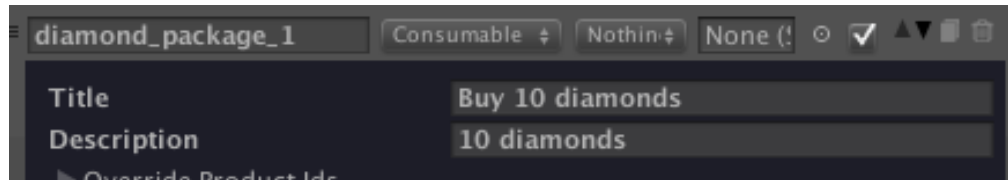
1. Follow “**Before Start**” section to setup 3 in-app purchase products in app store and named “*diamond_package_1*”, “*diamond_package_2*” and “*diamond_package_3*”.
2. Select **IAPShop** in the scene and open inspector window. Navigate to **IAP** section.



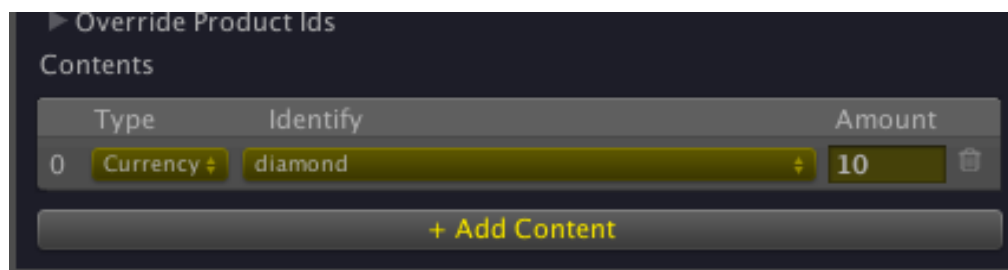
3. First, we setup a package for buying 10 diamonds. Add a new package with the “**Add In App Purchase**” button. Name it “*diamond_package_1*” (same name in app store). Select “*consumable*” and enable “*fetch*” to fetch data from app store (fetch should be enable to all real money IAP).



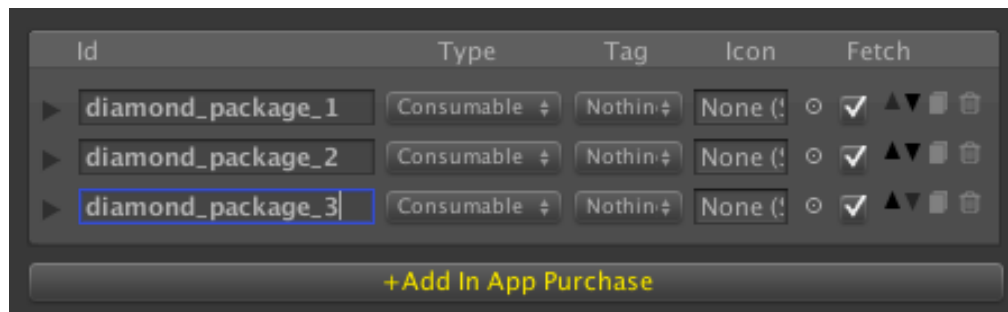
- Assign an icon for using in template. Fills in title and description if you want to override app store data. **Leave it blank if you want to fill data from app store.**



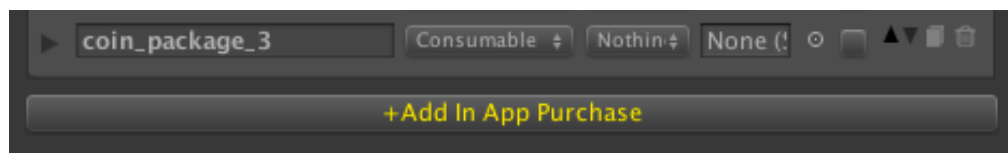
- Click “**Add content**” to add a new content in this package. We want to have 10 diamonds. So, select type “*currency*” and identify “*diamond*” and set the amount to 10.



- Repeat for “*diamond_package_2*” and “*diamond_package_3*”.



- Now, we create the packages for buying coin with diamonds. Add a new package by “**Add In-App Purchase**” button. Name it “*coin_package_1*”. Disable the “Fetch” option to indicate this is a local product.



8. Fill in the title and description. Set the price to 5 diamonds.

The screenshot shows the configuration for 'coin_package_1'. The 'Title' field is set to 'Buy 100 coins', the 'Description' field is set to '100 coins', and the 'Price' field is set to 'diamond' with a value of '5'.

9. Click “Add content” to add a new content in this package. Select “*currency*” and “*coin*” and set the amount to 100.

The screenshot shows the 'Contents' section with a table of items. The first row has 'Type' as 'Currency', 'Identify' as 'none', and 'Amount' as '0'. Below the table is a '+ Add Content' button.

Type	Identify	Amount
Currency	none	0

10. Repeat for “*coin_package_2*” and “*coin_package_3*”.

The screenshot shows a list of three coin packages: 'coin_package_1', 'coin_package_2', and 'coin_package_3'. Each package has a 'Consumable' type, 'Nothin' tag, and 'None (!)' icon. A '+ Add In App Purchase' button is at the bottom.

11. The data part is ready. Follow next section to add the UI.

Id	Type	Tag	Icon	Fetch
diamond_package_1	Consumable	Nothin	None (!)	<input checked="" type="checkbox"/>
diamond_package_2	Consumable	Nothin	None (!)	<input checked="" type="checkbox"/>
diamond_package_3	Consumable	Nothin	None (!)	<input checked="" type="checkbox"/>
coin_package_1	Consumable	Nothin	None (!)	<input type="checkbox"/>
coin_package_2	Consumable	Nothin	None (!)	<input type="checkbox"/>
coin_package_3	Consumable	Nothin	None (!)	<input type="checkbox"/>

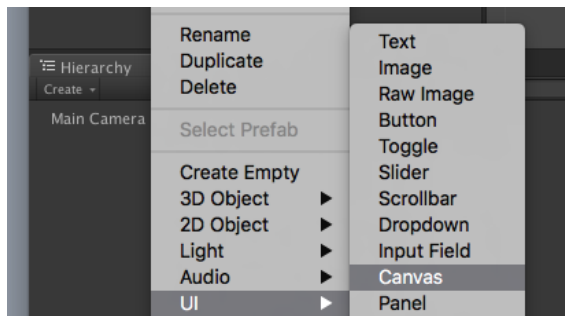
Create a list of diamond purchase button

IAP Manager Pro provides an easy way to add UI to your game. The UI system is a powerful template system base work with Unity UI. In this section, you learn how to setup UI for IAP with included template.

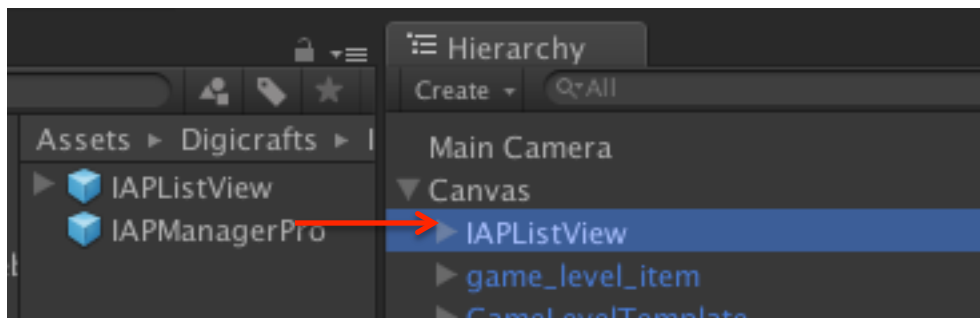
1. We are going to display a list of diamond packages, which we created in last section.



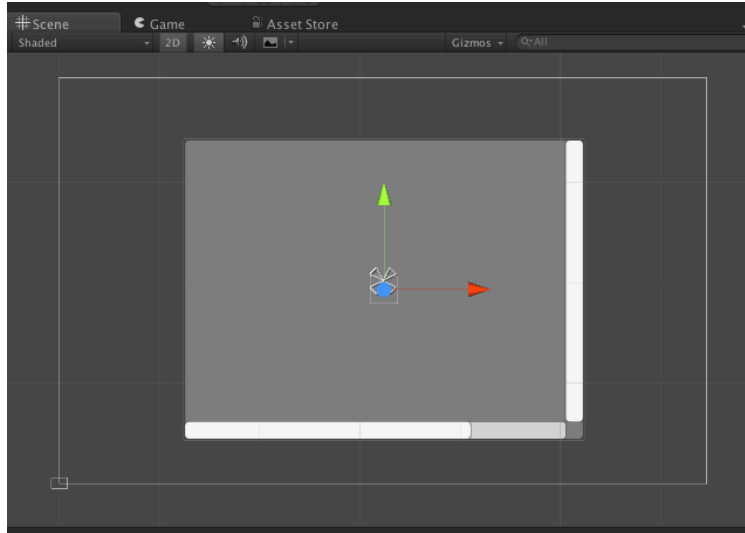
2. In the hierarchy window, create a canvas if you don't have any Unity UI created in your scene.



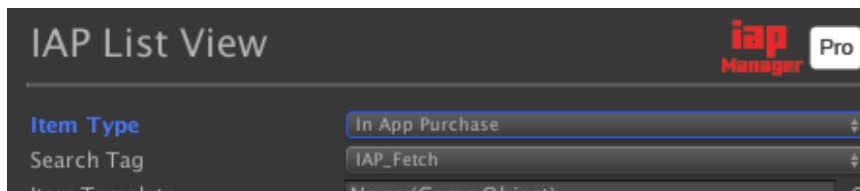
3. Drag the **IAP List View** prefab from prefabs folder to the hierarchy window under canvas.



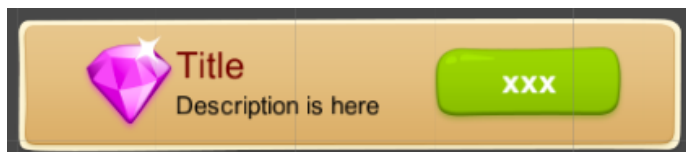
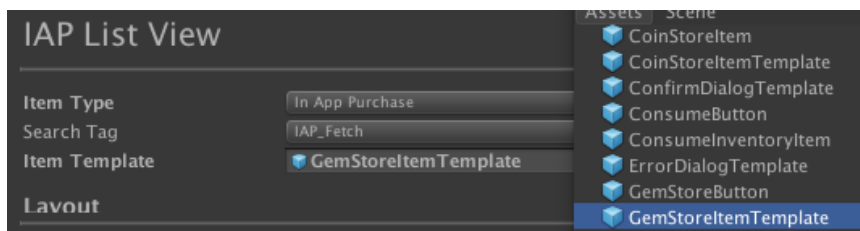
4. Resize to fit your scene.



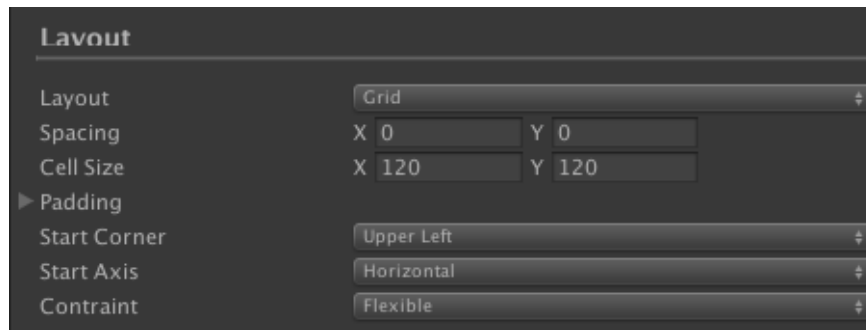
5. Open the inspector window of the **IAP List View**. Select “*In-App Purchase*” in item type. Select “*IAP_fetch*” from the search tag. (*IAP_fetch* is build-in tag for all item with fetch from app store)



6. In the template field, we select “*GemStoreItemTemplate.prefab*” from project. This is an example template, you will learn how to create template in next section.



7. (Optional) You can tweak the layout of list view from the inspector. For detail of the settings, you can refer to the Template section.

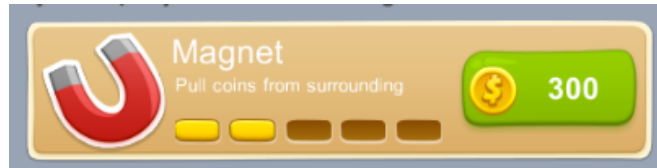


8. Run the project. You will see a list of button for buying diamonds.



V. Quick Start: Setup Ability/Power-up

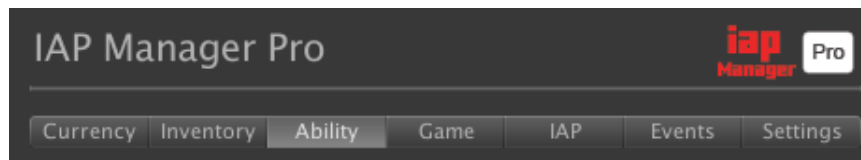
Setup Ability/Power-up System



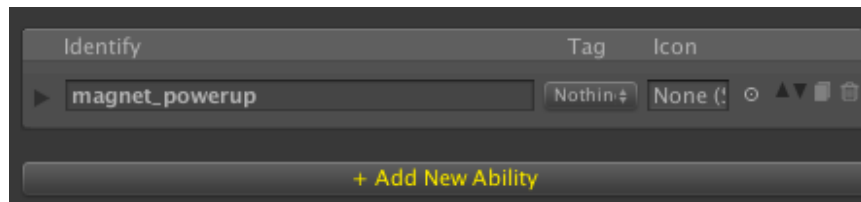
Your game may have an ability/power-up system, which allow player to upgrade their character ability. **IAP Manager Pro** provides a simple solution for this. In this section, you will learn how to add an ability system.

Define a power-up with 5 upgradable levels

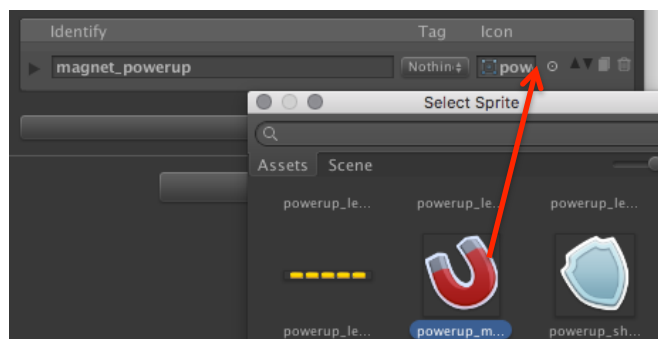
1. Select **IAP Manager Pro** and open inspector window. Navigate to “**Ability**” section.



2. Add a new ability by “**Add New Ability**” button. And set an identify name, we put “*magnet_powerup*” here.



3. Select an icon. The icon will use in the template system.



4. Fill in the title and description. Optional, you can set the text when locked and upgraded to max level.

Title	<input type="text" value="Title"/>
Description	<input type="text" value="Description"/>
Locked String	<input type="text" value="Locked"/>
Max String	<input type="text" value="Max"/>
Currency	<input type="text" value="coin"/>

5. Select the currency used for upgrading this ability.

Max String	<input type="text" value="Max"/>
Currency	<input type="text" value="coin"/>

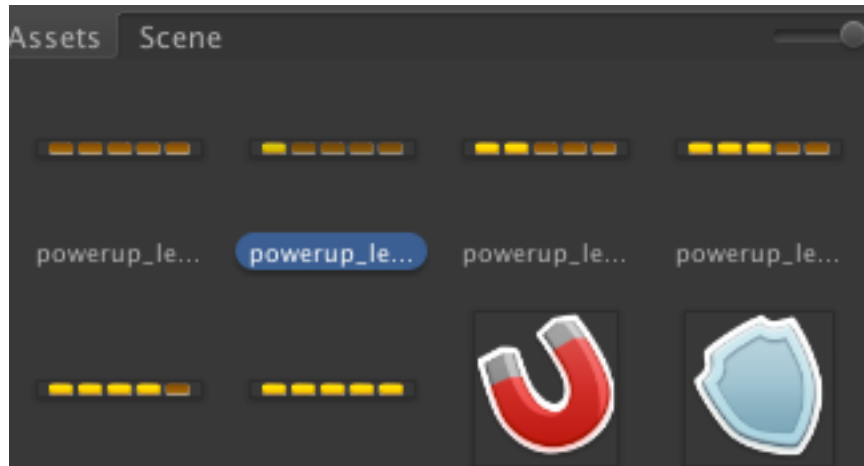
6. Next, you will add the levels for the ability. We can set the price for upgrade, icon and title for each level. Add a new level by “**Add Level**” button.

Title	Price	Icon
List is Empty.		
+ Add Level		

7. We added 6 levels. Set the price for upgrading to next level.

	Title	Price	Icon	
0	Attrach coins around	100	po	
1	Level 1 - 5 seconds	200	po	
2	Level 2 - 7 seconds	300	po	
3	Level 3 - 9 seconds	400	po	
4	Level 4 - 11 seconds	500	po	
5	Level 5 - 13 seconds	0	po	
+ Add Level				

8. The icon can be used in the template system to indicating which level reached. Here we use a series of images that light up differently when reach each level.

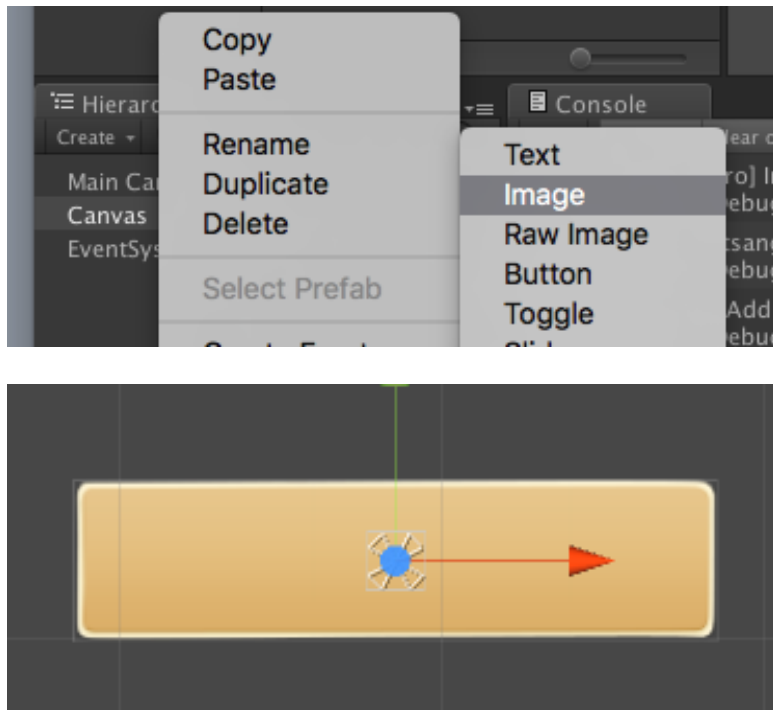


9. The data part is finished. In next section, we will add the UI.

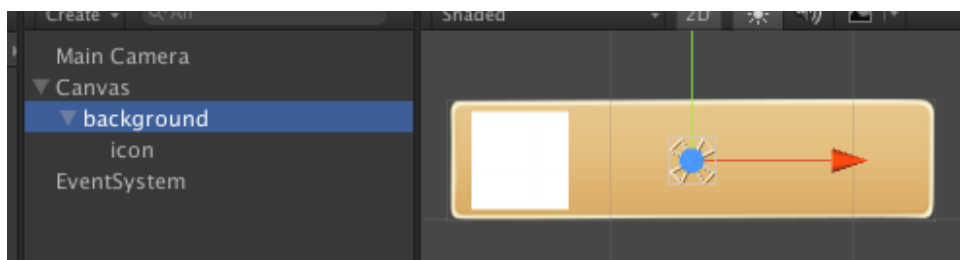
Create the UI

In this section, you will learn how to create the UI for displaying the ability level with an upgrade button.

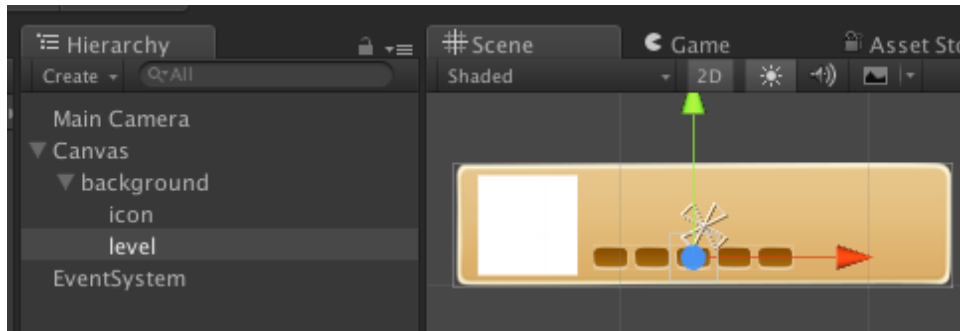
1. Right click on the canvas and create an Image for background.



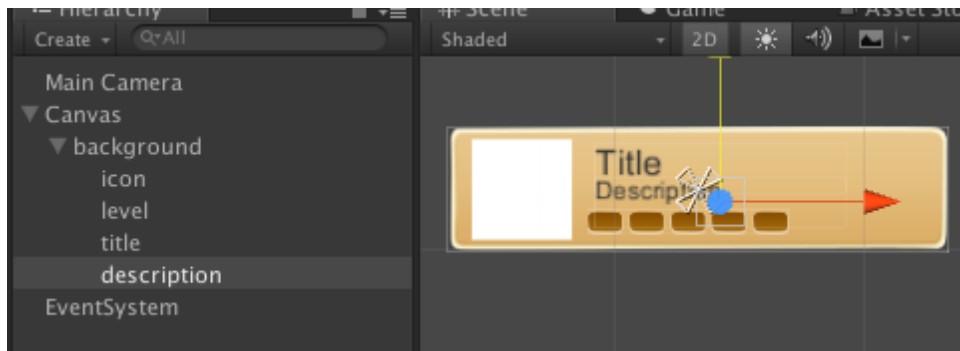
2. Create an **Image** as a child of the background image and name it “*icon*”. This will display the icon of the ability object.



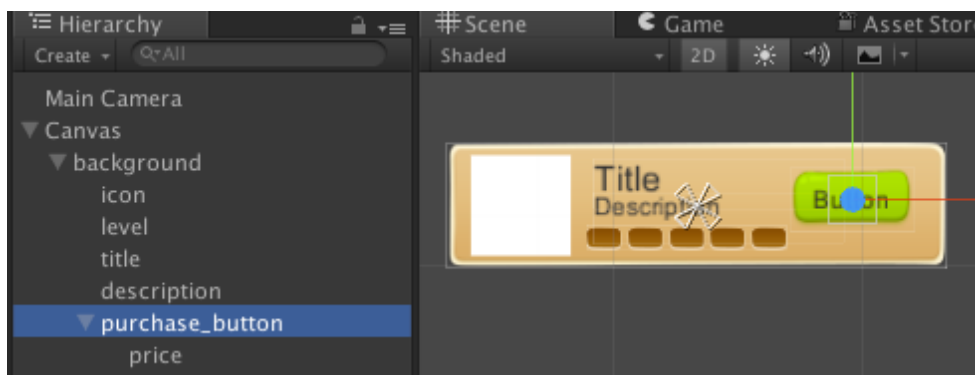
3. Create another **Image** for display levels graphics. Name it “*level*”. Resize to fit your design.



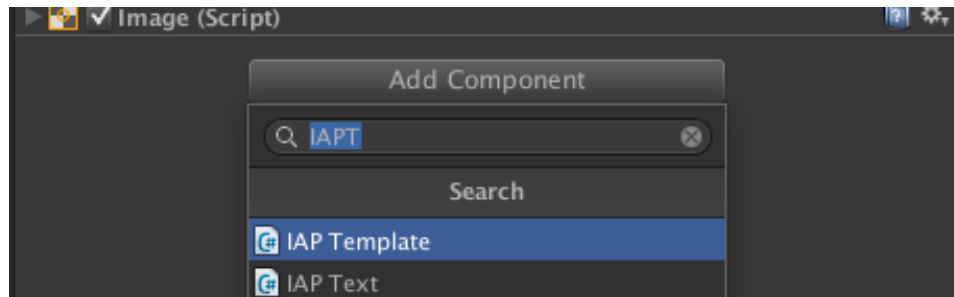
4. Create **Texts** for title and description. Named it with “*title*” and “*description*”.



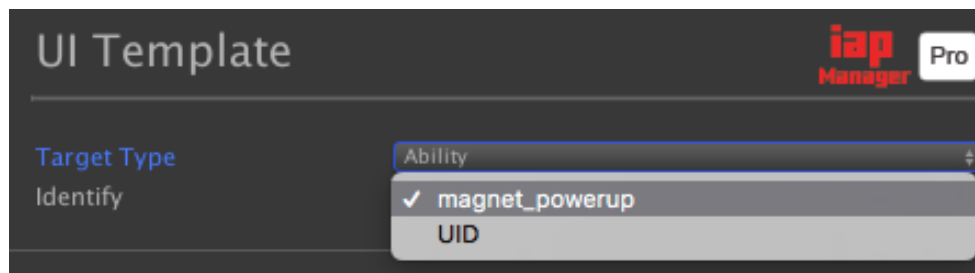
5. Add a Button. Change the name to “*purchase_button*”. Change the name of the Text within the button to “**price**”. This is used to display the price for upgrade.



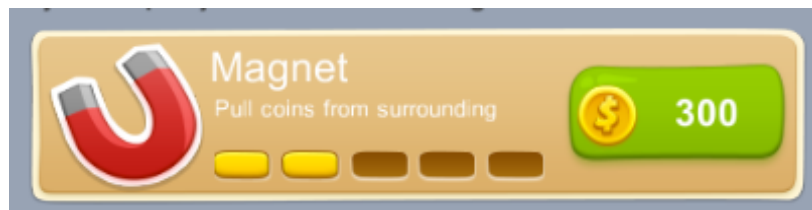
- The template is done. Select the root background Image and add a the component script “IAP Template”.



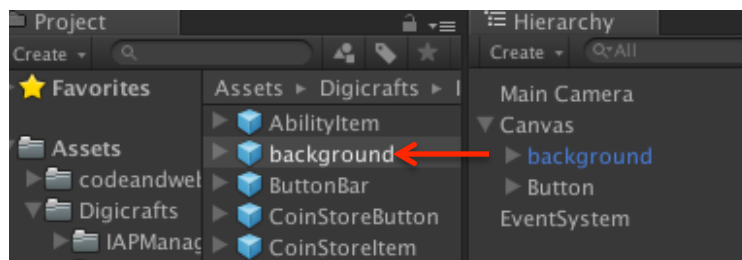
- From the inspector, select target type to “Ability” and identify to “magnet_powerup”.



- Run and test. You should see ability level and you can click on the purchase button to upgrade the ability level.



- (Optional) To use in the list view. You can create the prefab by drag the object from hierarchy window to project windows.

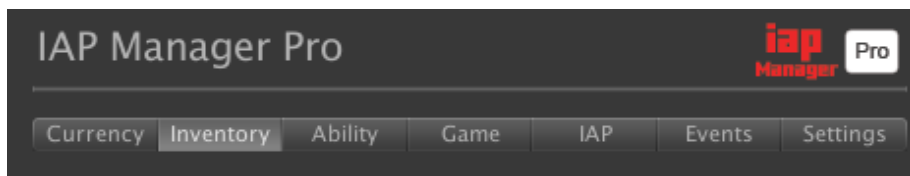


VI. Quick Start: Setup Inventory

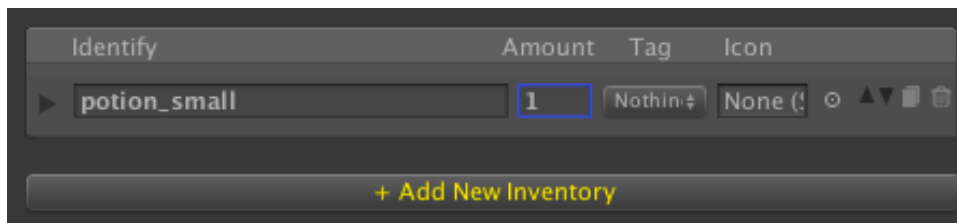
Game always contains inventory items, such as potion for restore health. They will allow player to consume it and enhancing game play. **IAP Manager Pro** provides an easy way to manage inventories in your game. In this section, you will learn how to setup the inventory data and the UIs to buy and consume the inventory.

Define an inventory

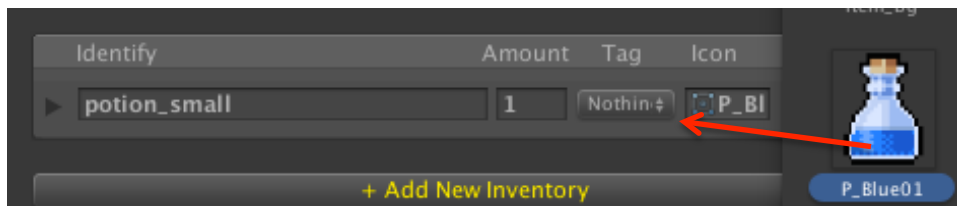
1. Select **IAP Manager Pro** and open inspector window. Navigate to “**Inventory**” section.



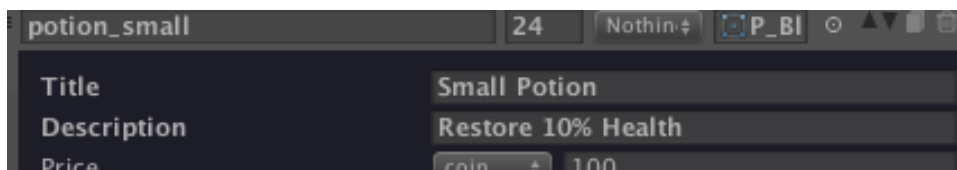
2. Add a new inventory by “**Add New Inventory**” button and specify an identify name. We put “*potion_small*” here.



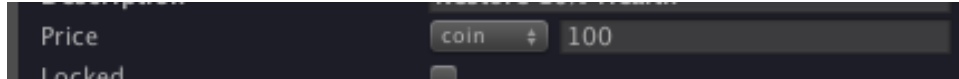
3. Select an icon for this inventory. The icon will be used in template system.



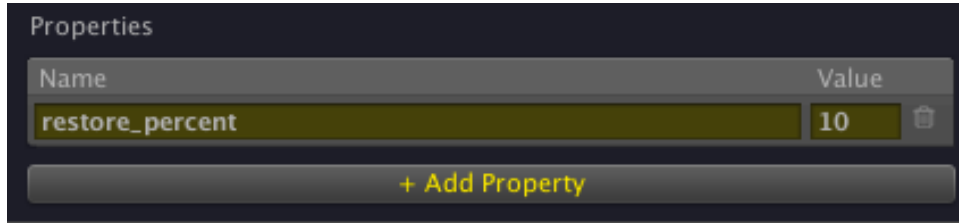
4. Fill in the title and description.



5. Select the currency and set the price for purchase this inventory.



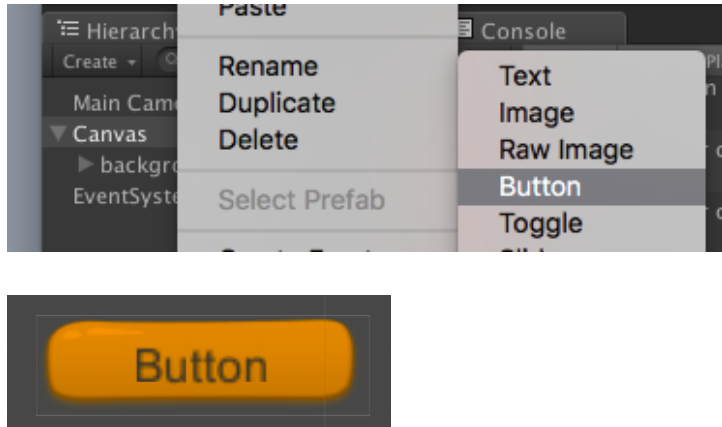
6. (Optional) In the properties table, you can define custom properties for each inventory. E.g. you can define the health-restoring percentage for the potion. You can get the property value using api. (IAPInventory.GetPropertyValue(name))



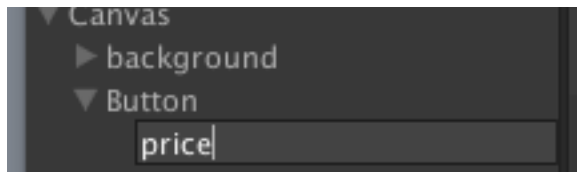
7. The inventory data are now ready to use.

Create an inventory purchase button

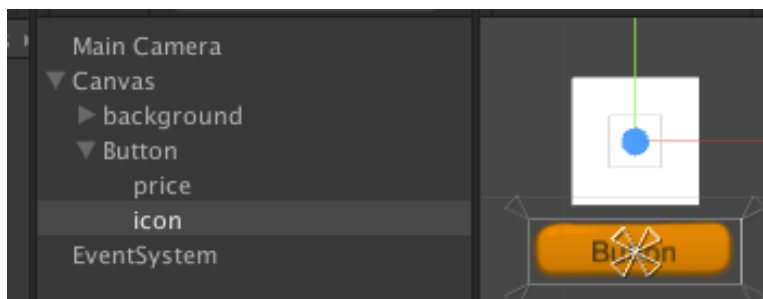
1. Right click on the canvas and create a Button. Assign an image and resize to fit your design.



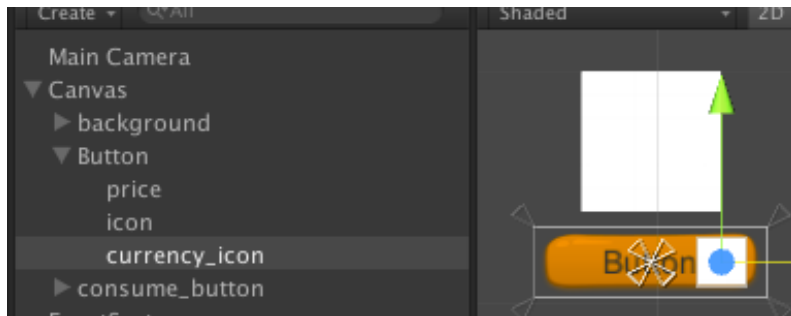
2. Change the name of the **Text** link to the button to “**price**”. This will display the cost for the inventory.



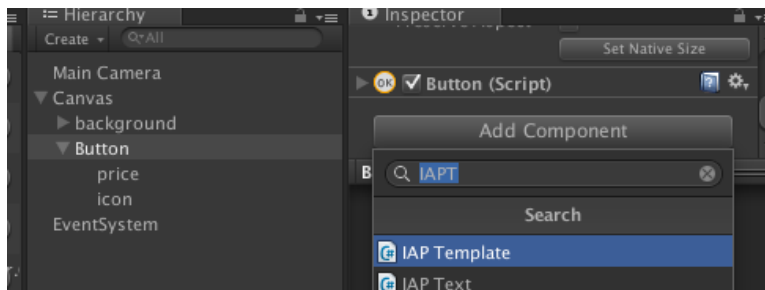
3. Create an **Image** as a child of the root button and name it “**icon**” and arrange the image to fit your design. This will display the “**icon**” of the inventory.



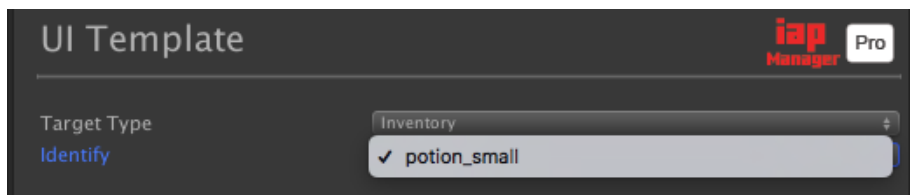
4. Create another **Image** as a child of the root button and name it “currency_Icon” and arrange the image to fit your design. This will display the currency’s “icon”.



5. Select the root button object and add a the component script “IAP Template”.



6. From the inspector, select target type to “*Inventory*” and identify to “*potion_small*”.

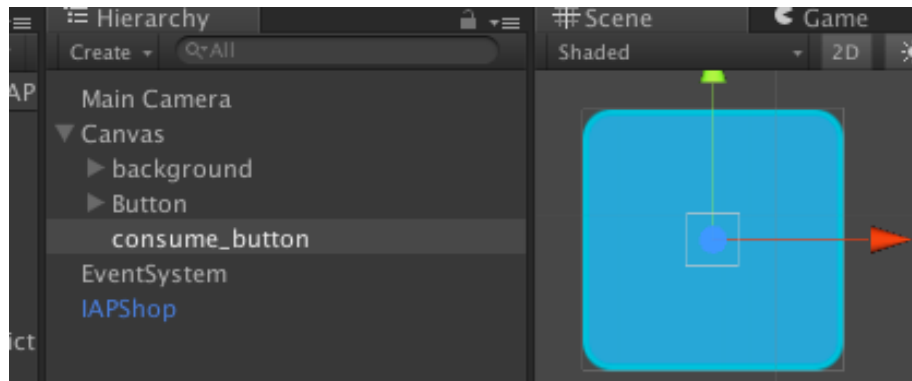


7. Run and test. Now, you can use the button to purchase the potion.

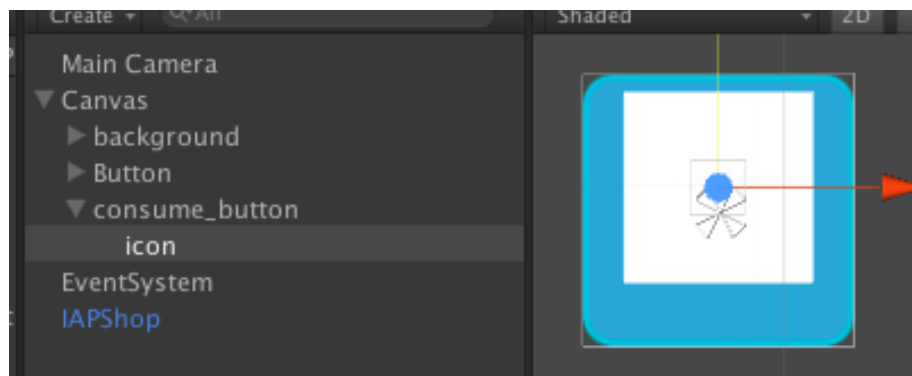


Create a button for consume the inventory

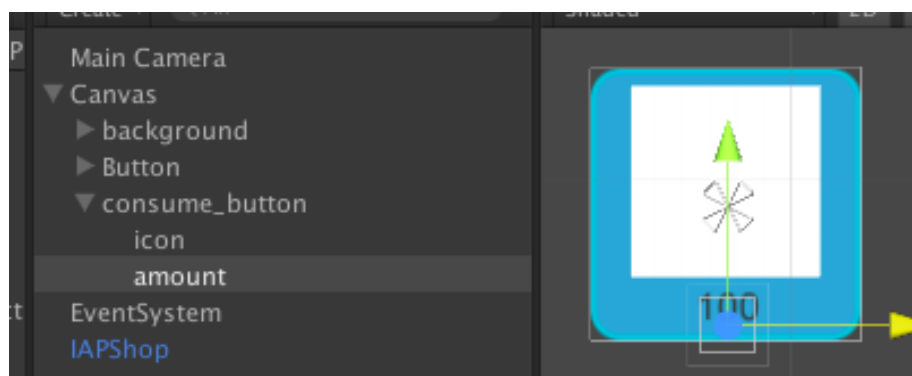
1. Right click on the canvas and create a **Button**. Assign an image and resize to fit your design.



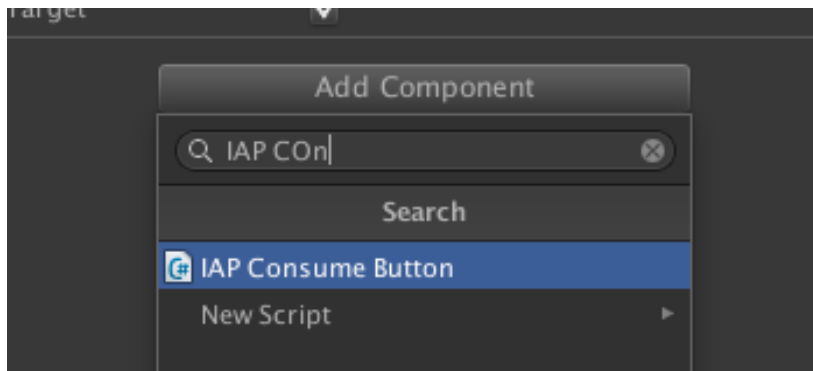
2. Create an **Image** as a child of the root button and name it "*icon*". This will display the "*icon*" of the inventory.



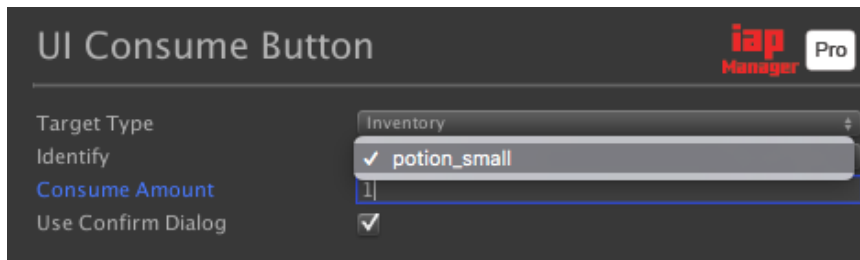
3. Create an **Text** as a child of the root button and name it "*amount*". This will display the "*amount*" of the inventory remains.



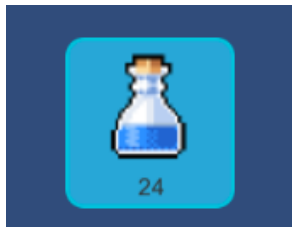
4. Select the root button and add a the component script “**IAP Consume Button**”.



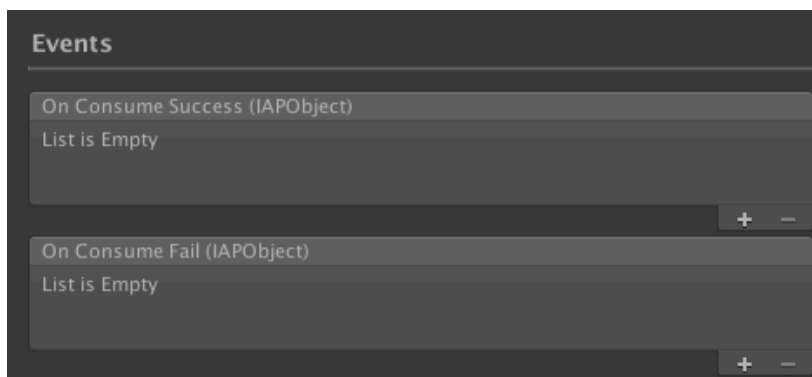
5. From the inspector, select target type to “*Inventory*” and identify to “*potion_small*”. Specify the amount will consumed.



6. Run and test. You can use the button to consume the potion.



7. (Optional) **UI Consume Button** provides consume success and fail event. You can connect to you script for action after consume.

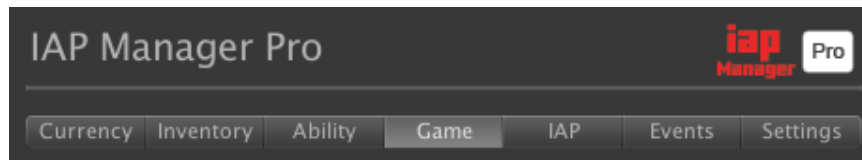


VII. Quick Start: Setup Game Level

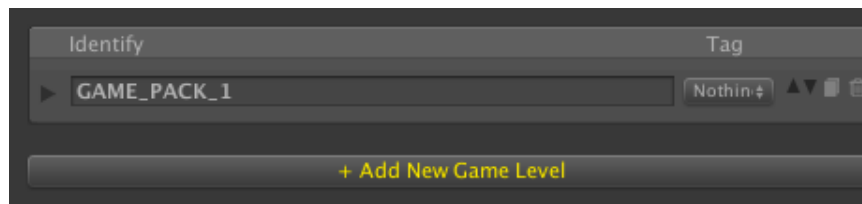
IAP Manager Pro provides an easy way to manage game levels in your game. In this section, you will learn how to setup the game level data and create a game levels selection scene.

Define an Game Level

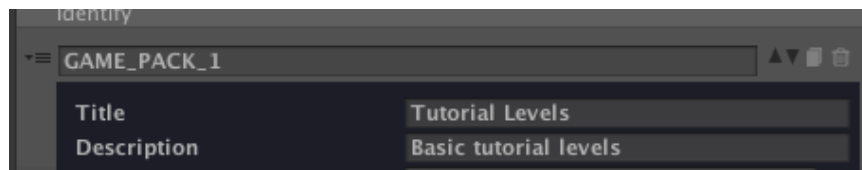
1. Select **IAP Manager Pro** and open inspector window. Navigate to “**Game**” section.



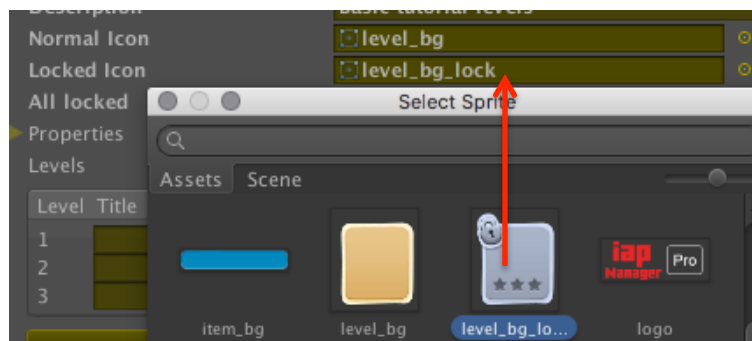
2. Add a new game level by “Add New Game Level” button and specify an identify name. We put “**GAME_PACK_1**” here.



3. Fill in the title and description.



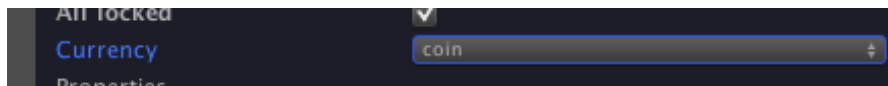
4. You can select the image for normal and locked state in normal/locked icon field.



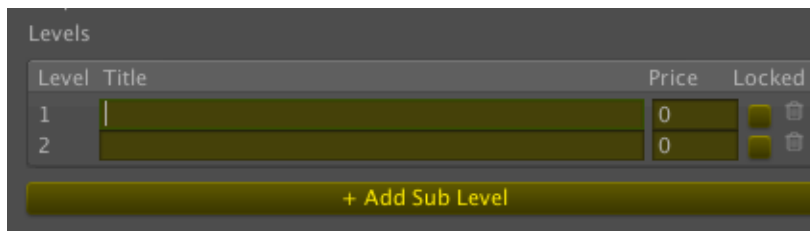
5. “**All locked**” option set all sub levels to be locked or unlocked in default. When in “**All locked**” mode, you can override by each locked option in sub level.



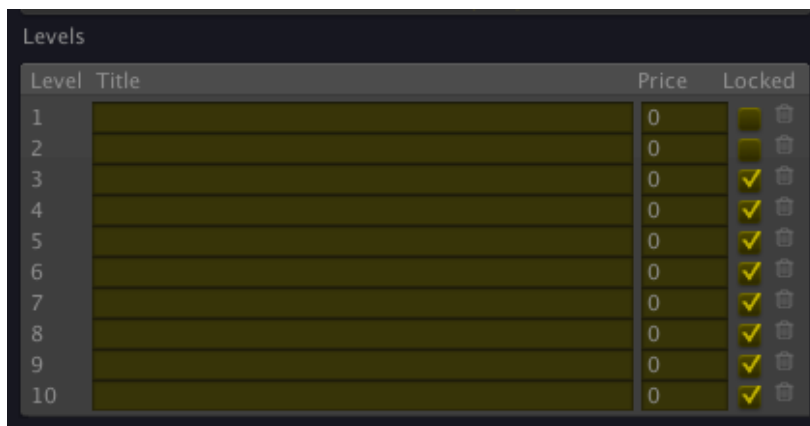
6. “**Currency**” field specify the currency used to unlock each sub level.



7. In the levels field, you can define how many sub levels contain in this game. Use “**Add Sub level**” button to add a new level. In each sub level, you can set the unlock price, title and locked option. **Keep price to 0 if you don't want to unlock with currency.**

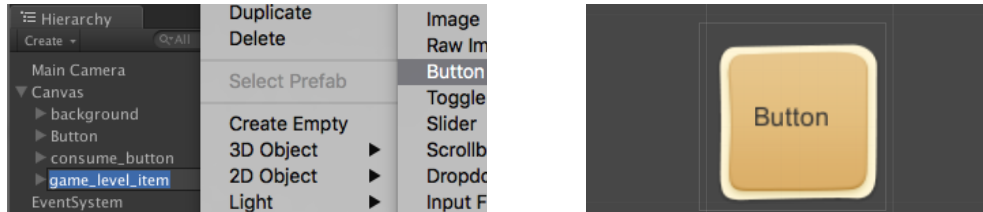


8. We add 10 levels and unlock the first and second level. The game level data is now ready to use.

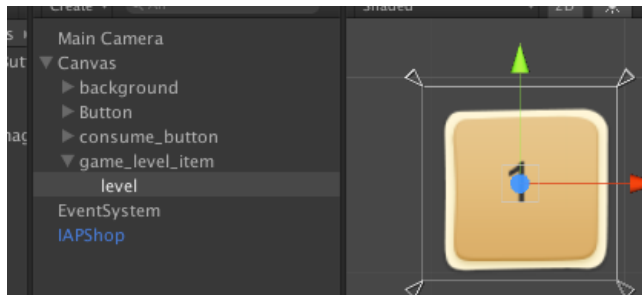


Create the game level item template

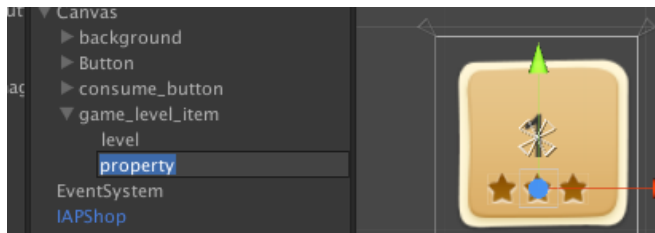
1. Right click on the canvas and create a Button. Assign an image and resize to fit your design.



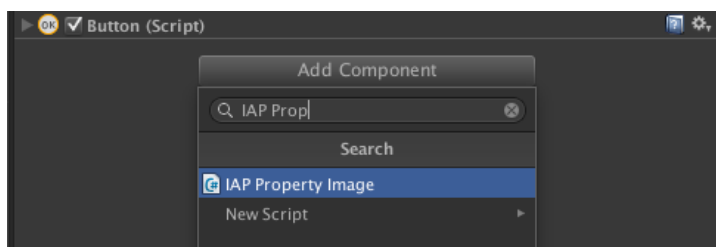
2. Rename the **Text** under the root button and name it "*level*". Resize and adjust the font size to fit the design. This will display the level number of the corresponding level.



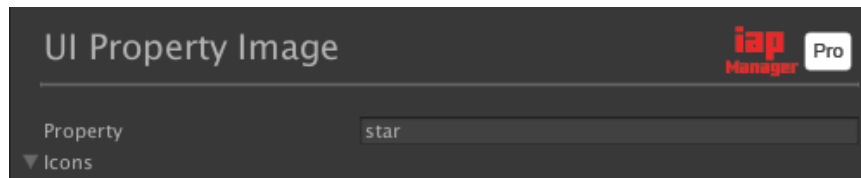
3. Create an **Image** as a child of the root button and name it "*property*". This image use to display the start got in the corresponding level.



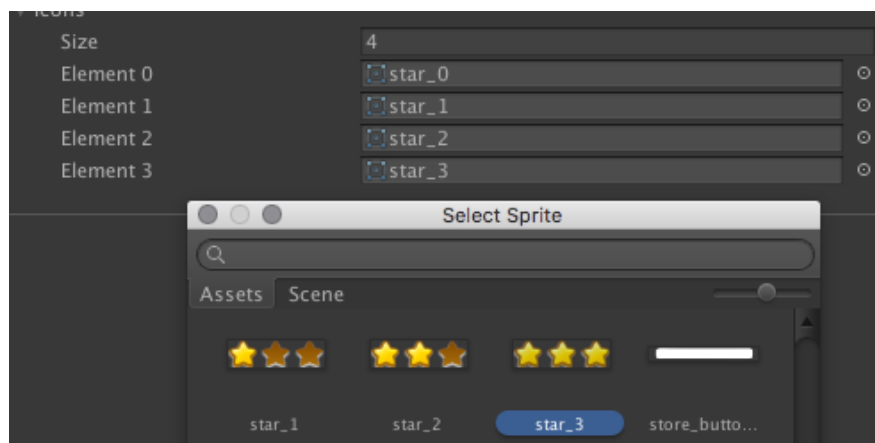
4. Add a component script "**IAP Property Image**". This script used to display image represent the custom property.



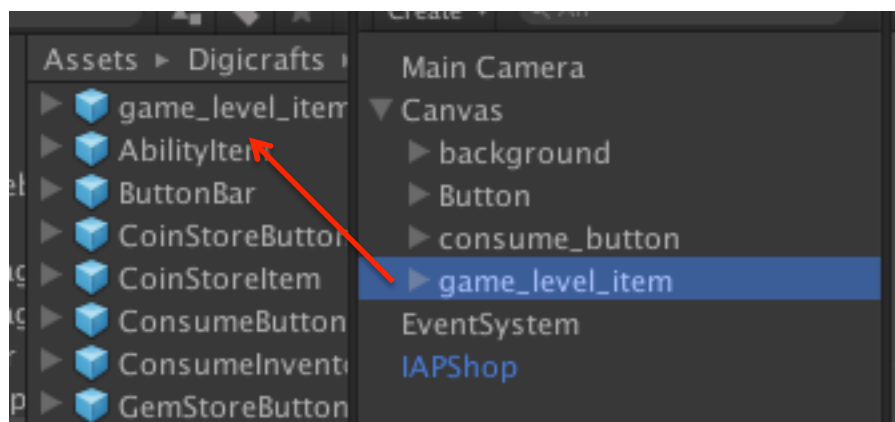
- From the inspector, type “star” in the property field.



- Specify the image for each value. The image will set to “star_0” if star=0, “star_1 if star=1, etc.

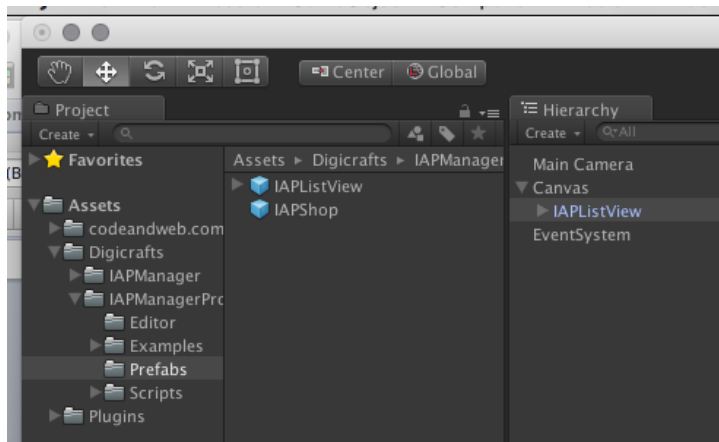


- To use in the list view. You need to create the prefab by drag the object from hierarchy window to project windows.

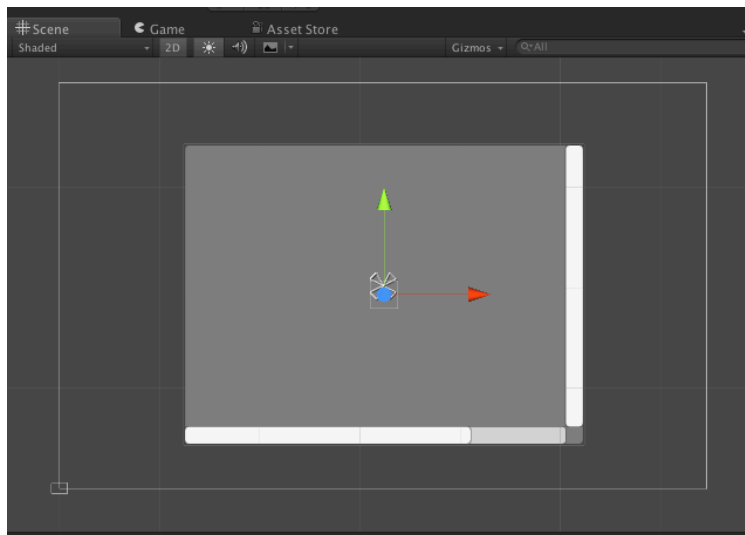


Create the game level list view

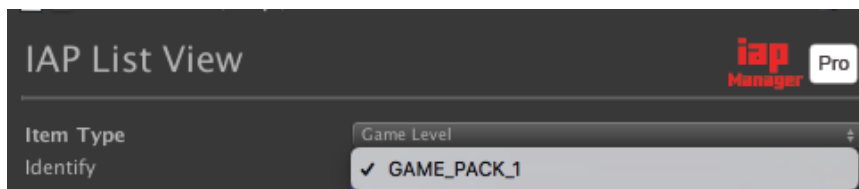
1. Drag the **IAP List View** prefab from prefabs folder to the hierarchy window under canvas.



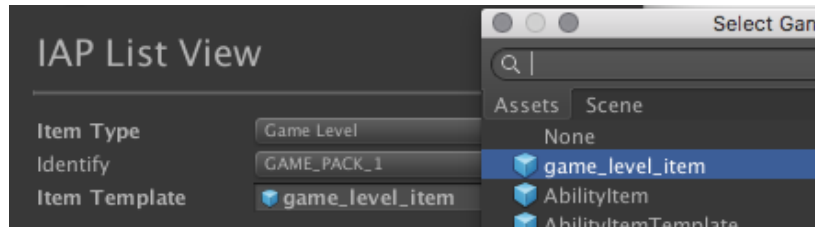
2. Resize to fit your scene.



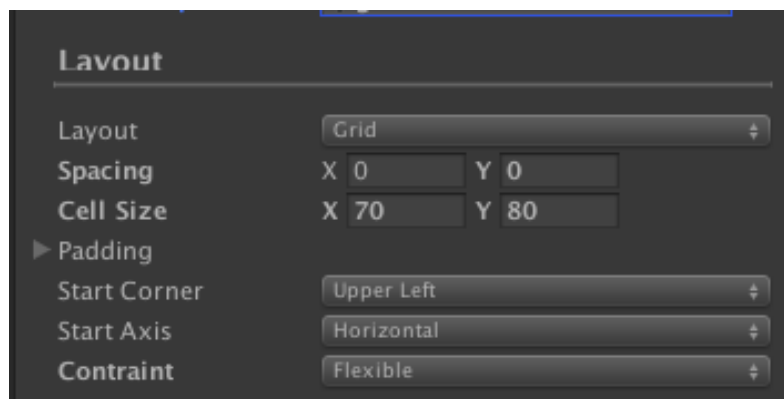
3. Open the inspector window of the **IAP List View**. Select "**Game Level**" in item type. Select "**GAME_PACK_1**" from the identify.



4. In the template field, we select “*game_level_item.prefab*” which created in last section.



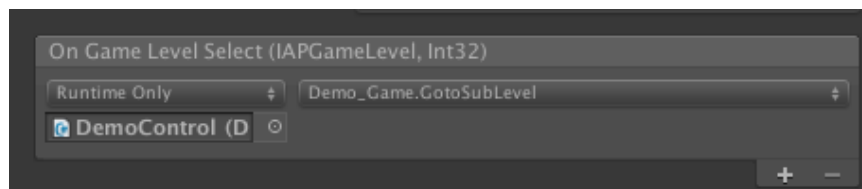
5. Set the Layout type to Grid, so the item arrange with container size. Cell size to 70 and 80.



6. Run the project. You will see the grid for selecting each level.



7. (Optional) You can connect the game level select event to your script in order to start the level.



VIII. IAP Manager Pro Inspector

IAP Manager Pro
Currency

Section Tab
Selecting different section

Add/Remove Currency
Use [+] button to add and [-] button to remove the currency

Icon
Sprite to use in template for the icon

Identify
Unique identify for the currency

Title
Title name of the currency

Amount
Default amount of currency

IAP Manager Pro
Inventory

Icon
Sprite to use in template for the icon

Arrange Buttons
Move the item up or down in the list.

Duplicate / Delete Button
Duplicate the item and delete the item.

Title / Description
Title and description of the item

Price
Set the currency and price to purchase this inventory

Icon
Sprite to use in template for the icon

Add Property Button
Add a new property

Name
Name of the property

Properties
Custom properties for this inventory.

Identify
Unique identify for the inventory

Amount
Default amount of inventory

Tag
Tag selection popup

IAP Manager Pro 1.0

IAP Manager Pro Ability

Tag
Tag selection popup

Icon
Sprite to use in template for the icon

Title / Description
Title and description of the item

Locked / Max String
The string replace the price when locked and reach max level

Price
Set the currency and price to purchase this inventory

Icon
Sprite to use in template for the icon in eac level

Price
The cost to unlock next level

Title
Text title for each level

Add level
Add a new ability level

Add New Ability
Create new ability data

Levels
The level of the ability.

Title	Price	Icon
0 Unlock Level	100	[po]
1 Level 1	200	[po]
2 Level 2	300	[po]
3 Level 3	400	[po]
4 Level 4	500	[po]
5 Max Level	0	[po]

IAP Manager Pro Game Level

Identify
Unique identify for the IAP package. Product id defined in app store

Title / Description
Title and description of the item

Normal/ Locked Icon
Sprite to use in template for the normal and locked icon

All Locked
Set all level locked. You can override this setting in levels section below.

Properties
Defines custom properties. Properties can access with api

Title
Text title for each level

Locked
Define if the level is locked. Only work with "All locked" enabled.

Sub Levels
Define how many sub level in each game level

Level	Title	Locked
1		
2		
3		
4		✓
5		✓
6		✓
7		✓
8		✓
9		✓
10		✓

IAP Manager Pro In-App Purchase

Identify

Unique identify for the IAP package. Product id defined in app store

Type

Set the product type. Can be consumable, non-consumable and subscription

IAP Manager Pro

Fetch Data On Start ☐

Id	Type	Tag	Icon	Fetch
gem_package_1	Consumable	Nothin	gem	<input checked="" type="checkbox"/>
bonus_package_1	Consumable	Nothin	coin	<input type="checkbox"/>

gem_package_1

Title: Buy 5 Gems

Description:

Override Product Ids:

Contents:

Type	Identify	Amount
0	Currency: diamond	5

+ Add Content

bonus_package_1

Title: Buy 1000 coins

Description:

Price: diamond 22

Contents:

Type	Identify	Amount
0	Currency: coin	1000
1	Currency: diamond	5

+ Add Content

Icon

Sprite to use in template for the icon

Fetch Store Data

Fetch data from app store. Enable: Real money purchase. Disable: Local product

Title / Description

Title and description of the item

Override Product Ids

You can set here to override the product id in each store.

Contents

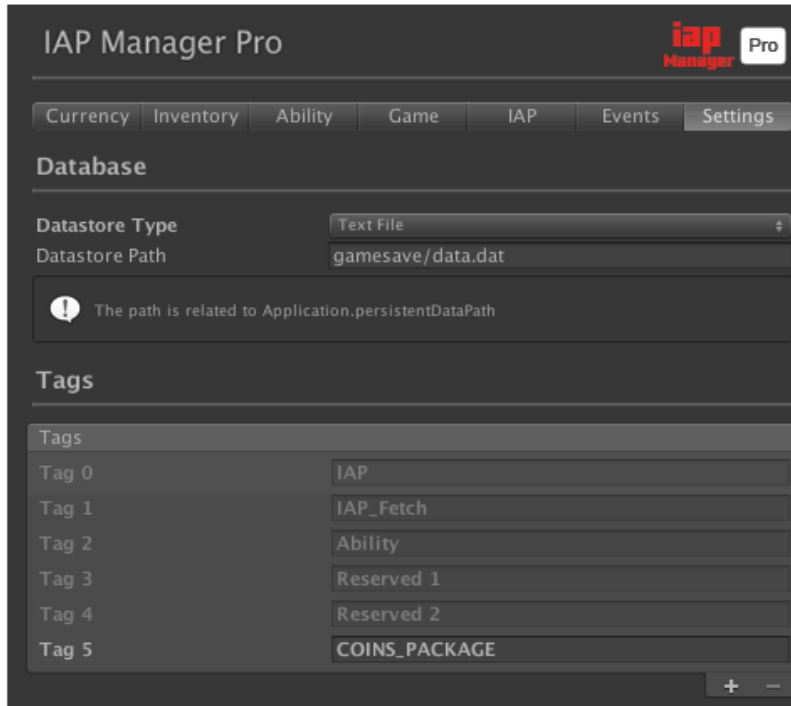
Set the contents included in this IAP package.

Price

You can set the price if it is a local product.

Add Content

Button to add a new content

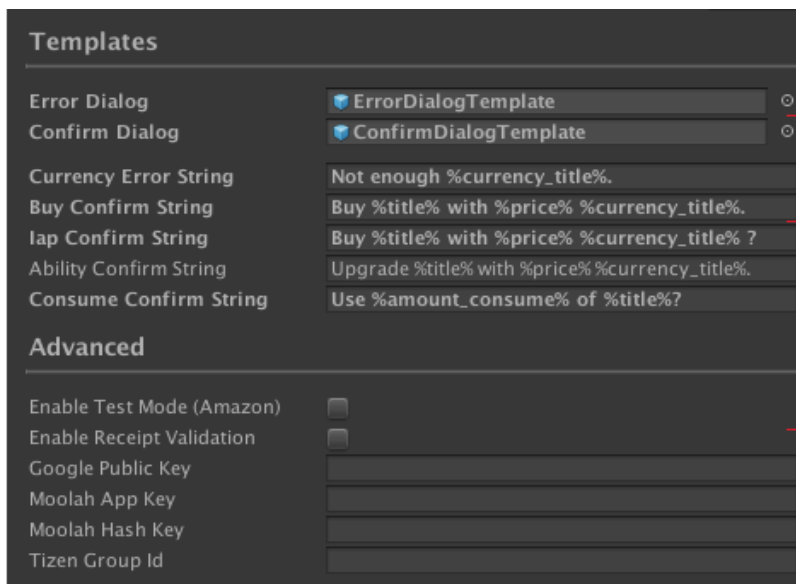


Datastore

Set how to store the data.
Support Player Pref, JSON
Text File, XOR Binary File
and AES Encrypted File.

Tags

You can manage tags here.
Tag is use for grouping items
which display in list view.



Dialog Templates

Set the prefab use for error
and confirm dialog.

Templates' String

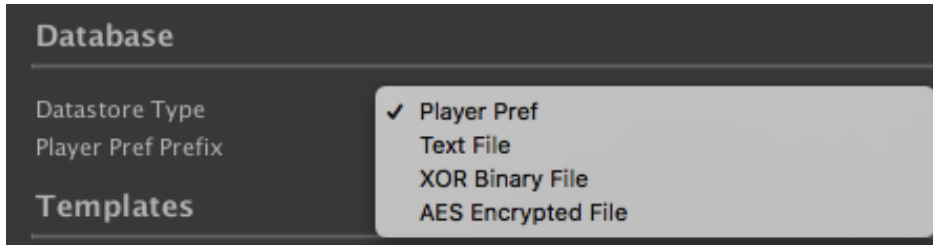
Set the template's string
value use in dialog.

Advanced Settings

Test Mode: Enable test
mode for amazon store
Receipt Validation: Enable
client side receipt validation

IX. Data Store Type

IAP Manager Pro provides several options to store data including *Player Pref*, *JSON Text File*, *XOR Binary File* and *AES Encrypted File*.



Database Type	Description
Player Pref	Data will store using Unity PlayerPrefs class. <ul style="list-style-type: none"> • Player Pref Prefix defined the prefix of the key when saving data using PlayerPrefs.
JSON Text File	Data will store in a json form text file. Use this if you want a simple format for your game data. <ul style="list-style-type: none"> • DataStore Path defines the path to save file related to Application.PresistantDataPath.
XOR Binary File	Data will store in a json format and encoded with XOR. Use this if you want simple security of your saved data. <ul style="list-style-type: none"> • DataStore Path defines the path to save file related to Application.PresistantDataPath.
AES Encrypted File	Data will store in a json format and encoded with AES and hash key. Maximum security is ensured for saved data. <ul style="list-style-type: none"> • DataStore Path defines the path to save file related to Application.PresistantDataPath. • Crypto Hash is the hash key for encoding the data. Use a different key for different game.

X. Templates

IAP Manager Pro comes with a powerful template system use along with Unity UI (Extend to other UI such as NGUI in the future). The template system includes scripts for different purpose.

UI Text

UI Text is used for displaying object properties as text, such as displaying amount of currency.

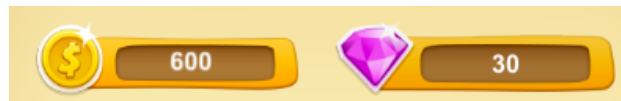


Figure 1 Examples using UI Text

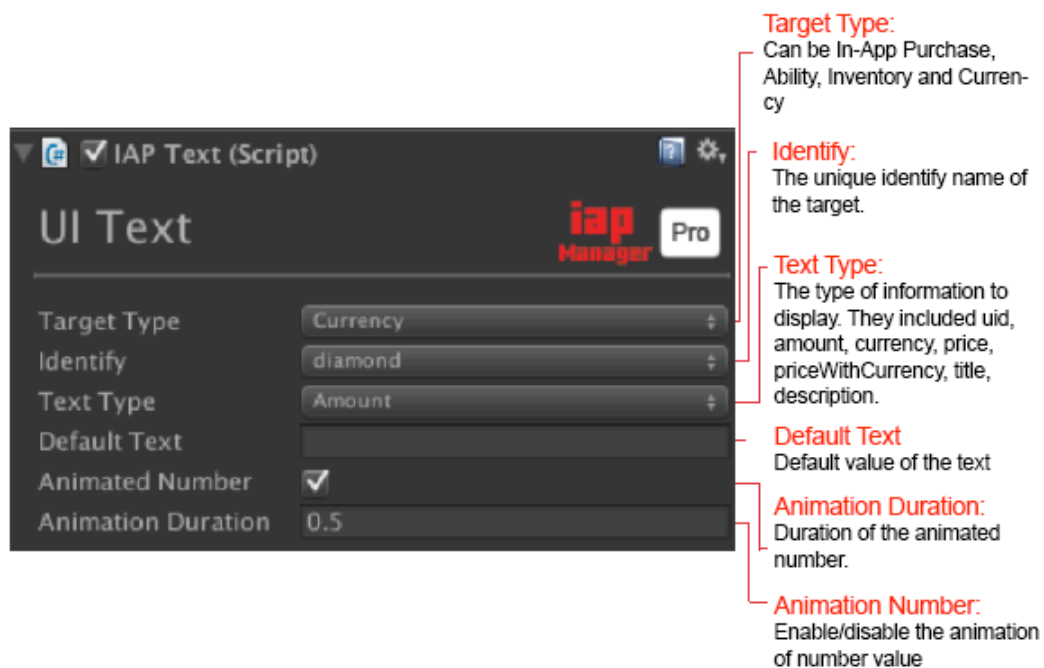


Figure 2 UI Text Inspector

UI Consume Button

UI Consume Button is used for adding a button for consumes the object (e.g. a button to consume potion) with events to connect to your own script.



Figure 3 Example of consume button

Target Type:
Can be In-App Purchase, Ability, Inventory and Currency

Identify:
The unique identify name of the target.

Consume Amount:
The amount to consume by this button.

Use Confirm Dialog:
User will prompt with a confirm dialog before process the consumption.

OnConsumeSuccess Event:
Event feedback when consume success.

OnConsumeFail Event:
Event feedback when consume fail.

Figure 4 Inspector of UI Consume Button

UI Template

UI Template is used for building custom UI with object data. Template can build with Unity UI with specify naming convention and you can design your own UI freely.



Figure 5 Examples of UI Temple with Ability

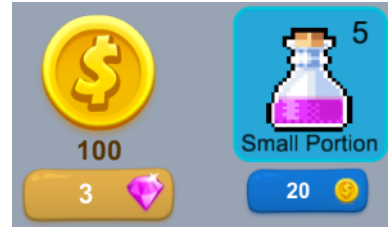


Figure 6 Examples of buy button



Figure 7 Inspector of UI Template

UI List View

UI List View is used for displaying a list of objects with grid layout, e.g. a list of coin packages in the shop.

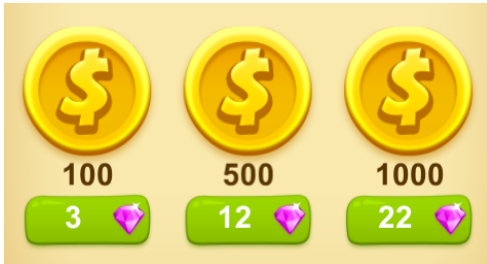


Figure 8 horizontal layouts



Figure 9 Vertical Layouts

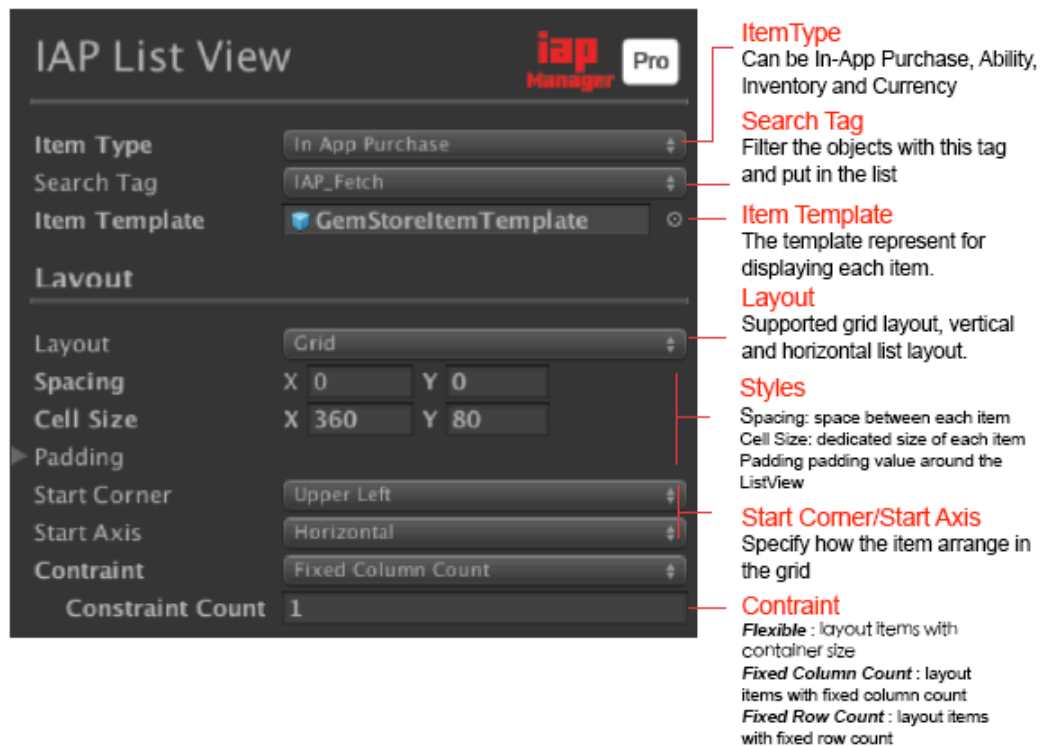
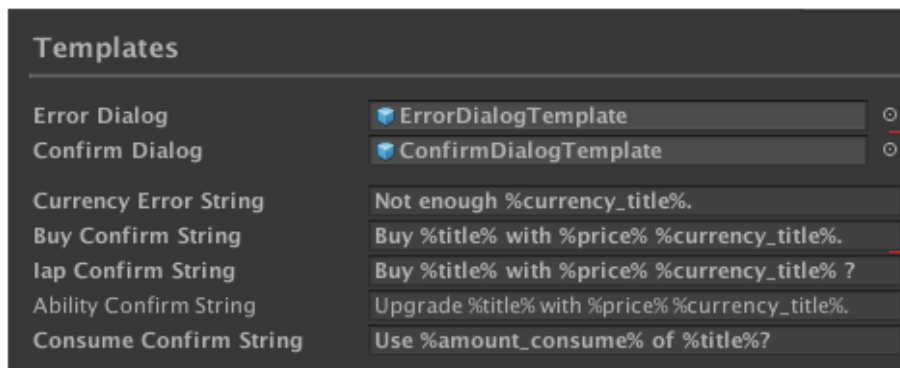


Figure 10 Inspector of IAP ListView

Dialog Template



IAP Manager Pro provides confirm dialog when using virtual currency to buy something. You can assign the dialog template in “Settings” section of IAP Manager Pro inspector. Also, you can change the template string in the inspector.



Template naming convention

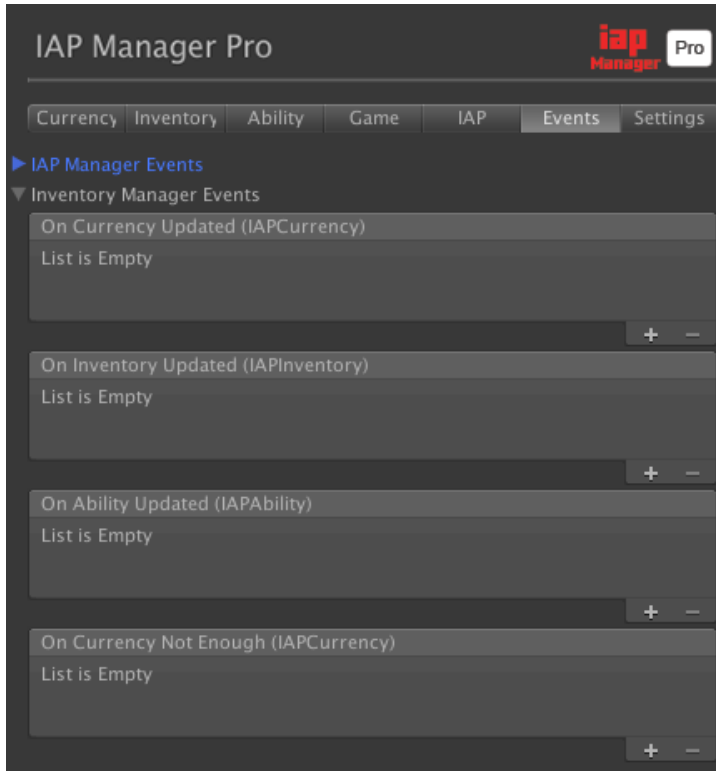


To create the template, you create a Unity UI object and follow the naming convention. **IAP Manager Pro** will search the child of your template and fill the data/content automatically. The naming convention is listed in following table.

Name	UI Type	Support Objects	Description
<i>uid</i>	Text	Inventory, Ability, Currency, IAP Package	Unique identify
<i>price</i>	Text	Inventory, Ability, Currency, IAP Package	Price for purchasing
<i>currency</i>	Text	Inventory, Ability, Currency, IAP Package	Currency for purchasing
<i>title</i>	Text	Inventory, Ability, Currency, Game Level, IAP Package	Title string
<i>description</i>	Text	Inventory, Ability, Currency, IAP Package	Description string
<i>icon</i>	Image	Inventory, Ability, Currency, Game Level , IAP Package	Icon image
<i>amount</i>	Text	Inventory, Ability, Currency	Amount remain
<i>content_xxx_amount</i>	Text	IAP Package	Amount of the content with uid xxx in the package
<i>currency_icon</i>	Image	Inventory, Ability, Currency	Icon of the currency for purchasing
<i>level</i>	Image	Ability, Game Level	Icon of the current level of ability
<i>level</i>	Text	Game Level	Current level of the game level.

XI. Events

IAP Manager Pro provides a easy to capture events. You can connect the events to your script by the inspector UI. You can find the event connectors in the “**Events**” section.



Event Name	Parameters	Description
OnCurrencyUpdated	IAPCurrency	Event when any currency updated.
OnInventoryUpdated	IAPInventory	Event when any inventory updated.
OnAbilityUpdated	IAPAbility	Event when any ability updated.
OnPackageUpdated	IAPPackage	Event when any package updated.
OnGameLevelUpdated	IAPGameLevel	Event when any game level updated.
OnCurrencyNotEnough	IAPCurrency	Event when consume something and not enough currency.
OnIAPInitialized	Dictionary<string, IAPProduct>	Event when app store IAP initialized.
OnIAPInitializeFailed	string	Event when app store IAP failed to initialize.
OnIAPPurchaseStart	IAPProduct	Event when start purchases a product.
OnIAPProcessPurchase	IAPProduct,string,string	Event when purchase success.
OnIAPPurchaseFailed	IAPProduct,string	Event when purchase failed.
OnIAPPurchaseDeferred	IAPProduct	Event when a purchase deferred.
OnIAPTransactionsRestored	bool	Event when a transaction restored.

XII. API

IAP Manager Pro provides a simple API for managing the data in IAP Manager Pro.

IAPInventoryManager Class

Static Method Name	Description
IAPCurrency GetCurrency (string uid)	Get the currency object by uid.
IAPInventory GetInventory (string uid)	Get the inventory object by uid.
IAPAbility GetAbility (string uid)	Get the ability object by uid.
IAPPackage GetPackage (string uid)	Get the package object by uid.
IAPGameLevel GetGameLevel (string uid)	Get the game level object by uid.
List<IAPCurrency> GetCurrencyList ()	Get a list of all currency objects.
List<IAPInventory> GetInventoryList ()	Get a list of all inventory objects.
List<IAPAbility> GetAbilityList ()	Get a list of all ability objects.
List<IAPPackage> GetPackageList ()	Get a list of all package objects.
List<IAPGameLevel> GetGameLevelList ()	Get a list of all game level objects.
List<IAPInventory> GetInventoryListByTag (string tag)	Get a list of all currency with tag name.
List<IAPAbility> GetAbilityListByTag (string tag)	Get a list of all ability with tag name.
List<IAPPackage> GetPackageListByTag (string tag)	Get a list of all package with tag name.
string[] GetTagsName ()	Get all tag name.

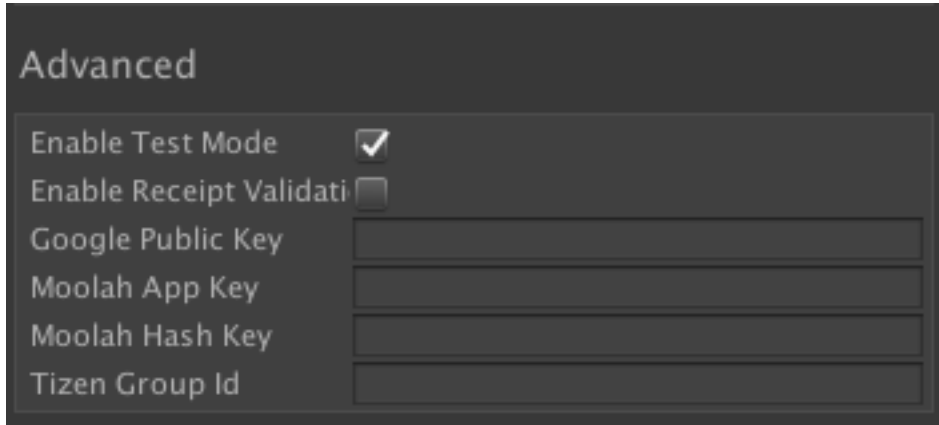
IAPCurrency, IAPInventory Class

Method Name	Description
bool Consume (int amount, bool delaySave = false)	Consume the object with amount
bool Refill (int amount, bool delaySave = false)	Refill the object with amount
Unlock (bool delaySave = false)	Unlock the object

XIII. Other Settings

Enabled Test Mode (Amazon Only)

You can enable/disable test mode from the inspector of IAP Manager. This will enable IAP Manager to use test data when testing on device. **Remember to disable it when you submit to app store.**



Prepare for Amazon Store Submit

Amazon kindle use Android as platform. You need to package the apps with Android Apps.

Before build to Amazon, please select Window>Unity IAP>Android>Target Amazon. Then, start builds your Android apk.

