PicturePuzzle – Documentation

Hi There, First of all, Thank you for purchasing this asset and we'll make sure you like this asset, We assure you that you'll get best possible support and assistance related to this asset 24X7 from us.

About this Game Template :

> This game template is a puzzle based word recognization game with fun puzzle 240 pre-included level, Objective is to find a perfect word that is hinted from the 4 different pictures shown to you.

Getting Started :

1. After downloading 4picure1word game template unity package, please create a new unity project and import package, you'll see a welcome screen with few instructions. Please complete those instructions.

2. IMPORTANT: After importing, you need to complete 2 essential steps.

   ◆ **Setup Unity IAP :**

   ✔ Please enable unity IAP from window ~> service ~> in-app purchasing ~> on. After enabling it, please import unity in-app purchasing asset package.
   For more information, please follow instructions given here.

   ✔ After Importing Unity InApp-Purchasing SDK, please import our in-app support script from HyperByte ~> Support Modules ~> In-App Purchasing ~> Setup. You're almost done with In-App Setup. Please note that we have all code related to in-app is ready and included, so you do not need to write any code unless you want specific customization.

   ✔ Separate documentation is given for more detailed instructions.

   ◆ **Setup Unity Monetization SDK 3.0 or Google AdMob.**

   ✔ We've provided support for Unity Monetization SDK 3.0 and Google Admob Ads SDK. You can use any of it for monetizing your game.

   ✔ Please download and import Ad SDK (Unity Monetization SDK 3.0 or Google Admob) and import package within the project.

   ✔ After Importing Ads SDK, please import our Ad Integration support script from HyperByte ~> Support Modules ~> Ad Network. Then import respective Ad Network Support script.

   ✔ Please, set up ad keys from the inspector in BaseAdManager script component, its located in Hierarchy ~> Controllers ~> AdController of Main.unityscene.

   ✔ Separate documentation is given for more detailed instructions.

3. Adding More Levels.

- Adding more levels is as easy as never before, please follow below instructions to add more levels or modify levels of your choice.

  ✔ Go to ~> PicturePuzzle ~> All-Levels folder within the unity's project window.

  ✔ Create a folder and name it "EP-<EPISODE NUMBER>". The <EPISODE NUMBER> is the next number than total existing episodes.

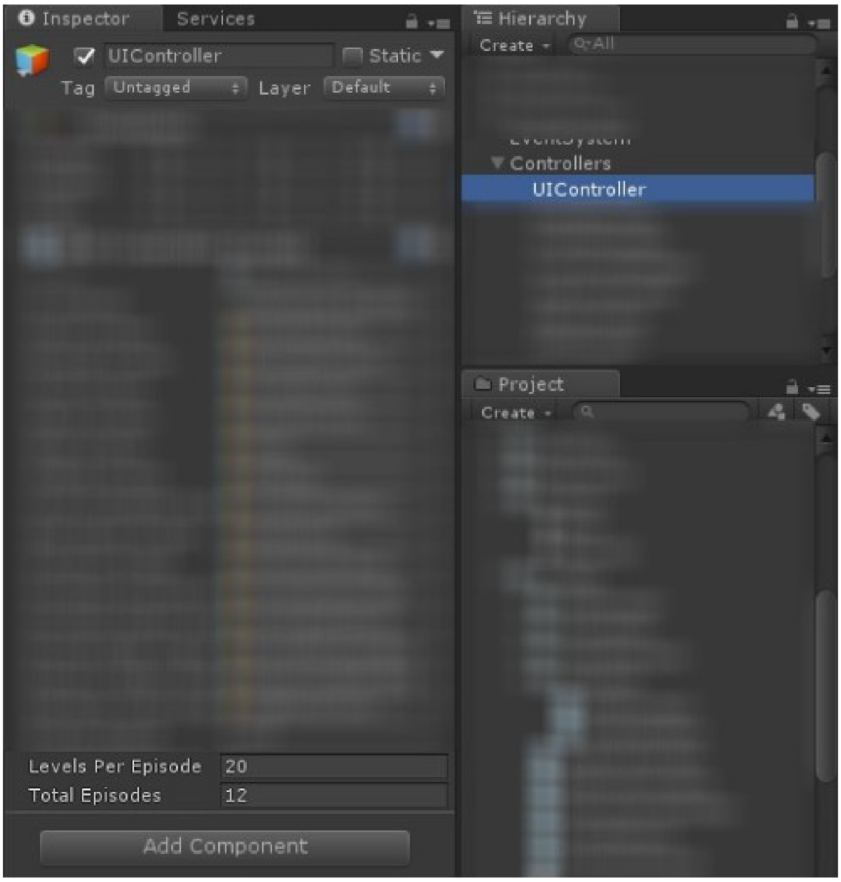
  ✔ Inside episode folder create all folders with <LEVEL NUMBER>.<LEVEL ANSWER> format. Please follow the existing episodes for more better understanding.

  ✔ Add level images in all respective folders of a level, now you need to follow a small step to get ready with working levels.

  ✔ **Localization Of Levels :**
  To make localization process simple, we are using CSV format. You need to create CSV file for each language containing all episodes. Please check the existing CSV files within the project to get better idea. After creating CSV files, please put it into PicturePuzzle ~> All-Levels ~> CSV.

  To make CSV making process simple, we used to create a excel sheet first and then export it as CSV, you can follow this method or your own way to make CSV.

  ✔ GoTo HyperByte Menu Item ~> Select Level Generator and then Select Generate All Levels. Please let the process complete, your levels are added and ready.



Levels Per Episode 20
Total Episodes 12

- ✔ Please set the number of total episodes you want to use in the game in the Inspector of UIController script component. [Image attached below].
- ✔ The Episode screen within the gameplay will automatically recognize new episode and will show up in episode screen. You don't need to do any additional code for that also.

4. Technical Implementation and Workflow.

- ◆ We tried to make this game template in such a way where even a non-technical folk also can modify and publish this game. But for the techie who wants to make a detailed modification or wants to add more features, below is few notes that may help you understand structure and workflow of this asset.
- ◆ This game works only on a single scene which is main.unityscene.
- ◆ This game is entirely made with uGUI including gameplay. *There may be very in opinion about gameplay should be made with uGUI or not but we tried to make in the best possible way which managed quality and performance.*
- ◆ The workflow of the game works based on Singleton Structure, we've made singleton script for the majority of the modules, ***you can see those singletons in the Hierarchy ~> Controllers.***

  - ✔ **UIController :**

    This controller is responsible for almost all UI Workflow and Game Progress Handling. Any UI Screen/Popup can be activated and deactivated using this controller. i.e., you want to enable the episode screen then you need to call like

    ```
    UIController.Instance.episodeScreen.Activate();
    ```

    You can add as many popups you want here and you can also modify existing. Simply create popup ~> create a public game object inside UIController to assign that popup/gameobject. Assign it from the inspector.

  - ✔ **InputManager :**

    You might not need to touch this most of the time, this handles input, touch delay, touch lock etc. For example after pressing any button if you don't want to disable touch for a particular delay then you can do something like below.

    ```
    if(InputManager.Instance.canInput(0.5F)) {
     gameObject.Deactivate();
    }
    ```

  - ✔ **ProfileManager :**

    This controller handles the status of user setting like music, sound and notification preference. Mostly you won't need to touch this controller unless you want to add any more user choice and save it.

  - ✔ **CurrencyManager :**

    This controller handles saves, restore and keep track of currency(coins) within the game. For example, you can deduct as well as a reward to a user using this controller. See the below snipper for more understanding.

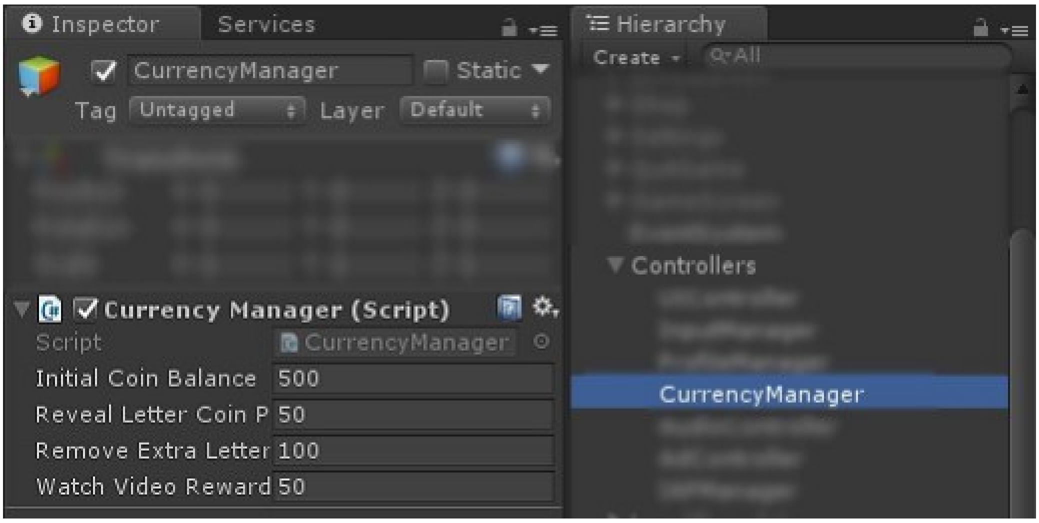
Add Coins :

```
CurrencyManager.Instance.AddCoinBalance(10);
```

Deduct Coins :

```
if(CurrencyManager.Instance.GetCoinBalance() >= 100) {
      CurrencyManager.Instance.deductBalance(100);
} else {
      UIController.Instance.getMoreCoinsPopup.Activate();
}
```

You also need to setup and configure in-game reward, default coins amout etc from here, please see below image for better understanding.



☑ CurrencyManager □ Static
Tag Untagged Layer Default
☑ Currency Manager (Script)
Script CurrencyManager
Initial Coin Balance 500
Reveal Letter Coin P 50
Remove Extra Letter 100
Watch Video Reward 50

✔ **AudioController**

This controller in-game sound and music. You can add an audio clip and play with this singleton script component.

✔ **Ad Manager :**

NOTE: The default AdManager script is just a placeholder script, please import original Ad Manager script based on respective Ad Network you choose for game monetization. More detailed instructions are given in separate documentation.

✔ **IAP Manager :**

NOTE: The default IAP Manager script is just a placeholder script, please import original IAP Manager script after enabling and importing in-app purchasing SDK. More detailed instructions are given in separate documentation.

~~~~~~
~~~~~~

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

➔ **Bussiness Email :** contact@hyperbytestudios.com

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