

Vr Games Dev

Dynamic Downloader using Addressables

August 2020



TABLE OF CONTENTS

1.	Introduction.....	4
1.1	Technical Support.....	4
1.2	Online documentation.....	4
1.3	Offline documentation.....	4
1.4	Demo.....	4
2.	Overview	5
2.1	TL:DR	5
2.2	DDuA Components.....	5
2.3	Process Diagram.....	6
3.	Requirements to use DDuA.....	7
3.1	Unity Addressable Asset system	7
3.2	Unity Remote Config.....	8
3.3	Standard Assets.....	9
3.3.1	Errors in Unity 2019+ with Standard Assets.....	11
4.	Configuring Addressables	12
4.1	Addressable window	12
5.	Creating your Main scene (VRG_Init)	14
5.1	VRG_Core.....	15
6.	Carousels (VRG_Carousel).....	17
6.1	Add your first Carousel.....	17
6.2	Create your VRG_Carousel	18
6.3	Here comes a new challenger	21
7.	Scenes as an Addressable (VRG_Scene)	22
7.1	Add your first scene.....	22
7.2	Add your VRG_Init scene to the build list.....	23
7.3	Configure your VRG_Scene.....	24
7.4	Configuring the PreLoads.....	24
7.5	Here comes a new challenger	25
8.	Change scenes (VRG_LoadScene)	27
8.1	Here comes a new challenger (Boss)	32
8.2	Boss – Cheat code	32
9.	Singletons and downloads.....	33
9.1	Singletons in VRG_Remote.....	33
9.2	Downloads in VRG_Remote	35



TABLE OF CONTENTS

10.	Using Unity Remote Config Instead of Local VRG_Remote	36
10.1	Review the log generated with the local configuration.....	36
10.2	Configure the Remote Config.....	36
10.3	Here comes a new challenger.....	39
11.	VRG_Announcements as a GIFT.....	40
11.1	Configure your VRG_Announcements	40
11.2	Add to your home scene	40
11.3	Here Comes a new challenger	41
12.	Examples.....	42
12.1	01 Tutorial	42
12.2	02 Carousels	44
12.3	03 Spawner	45
12.4	04 MoveAndUse	46
12.5	05 UnityCar	48
12.6	06 EVERYONEEEEEeeee	51



1. INTRODUCTION

We are very glad you decided to give us the opportunity to help you build a great game. We try to make the best technology, easy, useful and well documented. Nevertheless, we are aware everything can be improved and enhanced, so we are more than happy to receive a comment from you, so drop us a line at least to say Hi.

Best regards, GG, GL HF

Hakari

August 2020

1.1 Technical Support

You can get technical support at the following email:

unity.support@vrgamesdev.com

1.2 Online documentation

We want to keep this documentation up to date and the most detailed possible, since we cannot edit and improve a document already published, we provide the latest documentation online at the following URL:

<https://assetstore.vrgamesdev.com>

1.3 Offline documentation

You need to unzip the API.zip, there is a copy of the website for your personal use, just click the “**index.html**” file and it will run in a regular browser:

_VrGamesDev/DDuA/Documentation/API.zip

1.4 Demo

You can download the game from this tutorial from this URL:

<https://vrgamesdev.itch.io/ddua>



2. OVERVIEW

When you create a light app, you also increase the user retention and the initial download size. Generating better effective cost per click, cost per Mile and your cost per acquisition

Download the content as your application needs it.

You only need to upload the changes of the assets you modify, EASIER and FASTER dev cycle iterations.

It includes the following functionalities:

- Detect and verify Network connection to internet, you can tell the player to connect to WIFI if cellular connection detected.
- Smooth transitions between scenes
- Loading screens with progress bar
- A Carousel waiting screens for promotional, hints, seasonal and mini games
- Announcement systems to inform the players of the changes and updates
- Detailed beautiful logs in html of the asynchronous processes
- Local configuration or remote for easier updates

Systems integration

- Remote: For easy remote configuration of the assets and scenes to download
- Addressables: Fully integration of the Addressable system to handle the content

Everything documented, code organized, well commented with first class standards. Tested in IOs, Android, Mac and Windows.

2.1 TL:DR

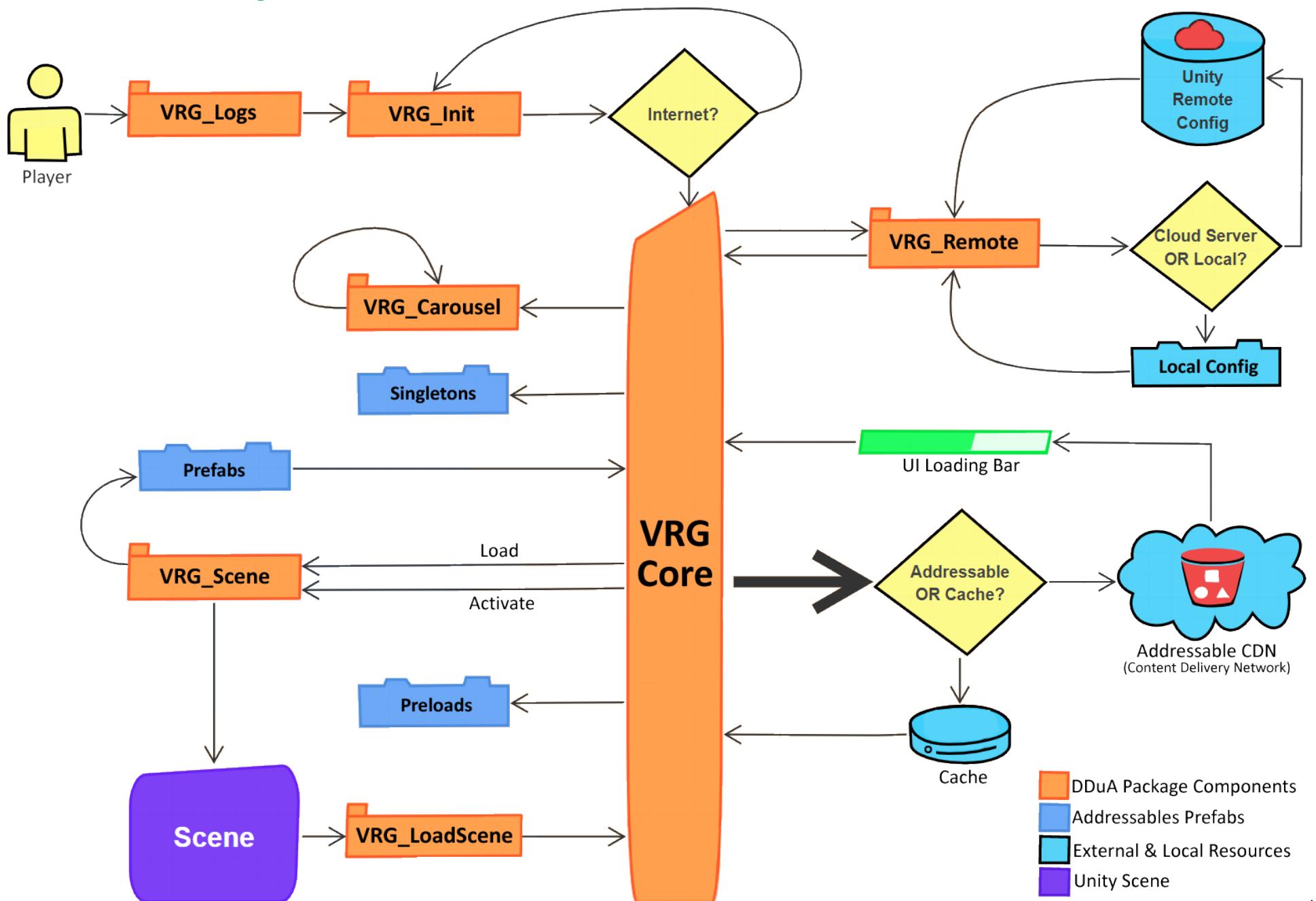
1. Download and install, Addressables, Remote config and standard assets
2. Create a Main Scene, and add it to the Build settings
3. Add the prefabs **EventSystem**, **VRG_Init**, **VRG_Remote** and **VRG_Logs**
4. Make a name simplified **VRG_Core** Addressable
5. Create a **Home** Scene, add it to the build settings and make it a name simplified addressable.
6. Configure the VRG_Remote to load the "**Home**" scene
7. Add Carousels to taste
8. Play the game and enjoy your new asynchronous package.

2.2 DDuA Components

- **VRG_Logs:** It allows to save events to understand and debug the asynchronous activity
- **VRG_Init:** This class checks for internet, and load the VRG_Core
- **VRG_Core:** The main class, it handles the communication and the interaction between all the VRG modules and the Remote servers, and the addressable systems.
- **VRG_Remote:** The main setting configuration it allows to load from local settings or cloud remote config
- **VRG_Carousel:** Elements that shows iterative to wait for download times
- **VRG_Scene:** The module that loads the Addressables scenes, It loads asynchronous its elements
- **VRG_LoadScene:** It loads the next scene and deletes the current scene



2.3 Process Diagram





3. REQUIREMENTS TO USE DDUA

This package uses and needs you to implement the following unity technologies:

3.1 Unity Addressable Asset system

The Addressable Asset system provides an easy way to load assets by “address”. It handles asset management overhead by simplifying content pack creation and deployment.

The Addressable Asset system uses asynchronous loading to support loading from any location with any collection of dependencies. Whether you use direct references, traditional asset bundles, or Resource folders for asset management, Addressable Assets provide a simpler way to make your game more dynamic.

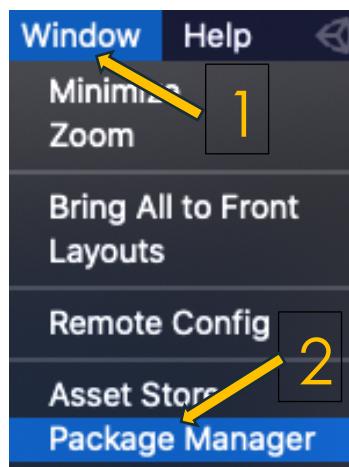
Documentation: You can read more about this module here

(<https://docs.unity3d.com/Packages/com.unity.addressables@1.11/manual/index.html>)

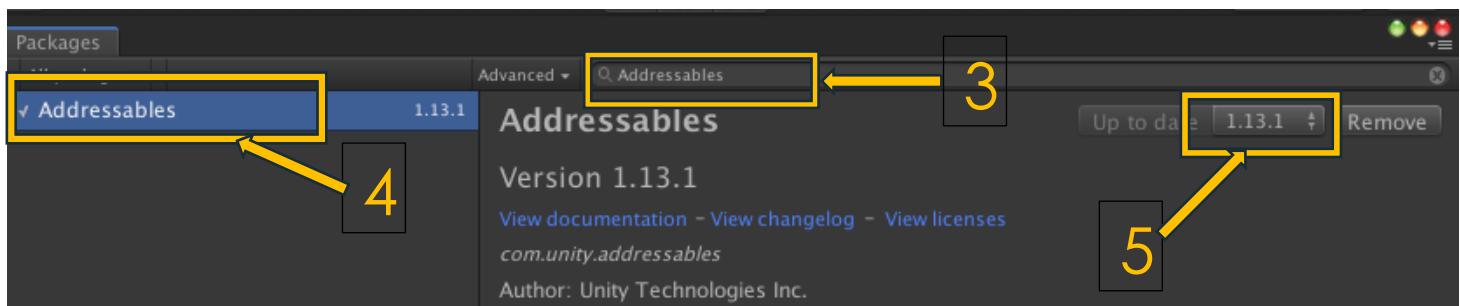
Minimum Version: 1.13.1

How to install it:

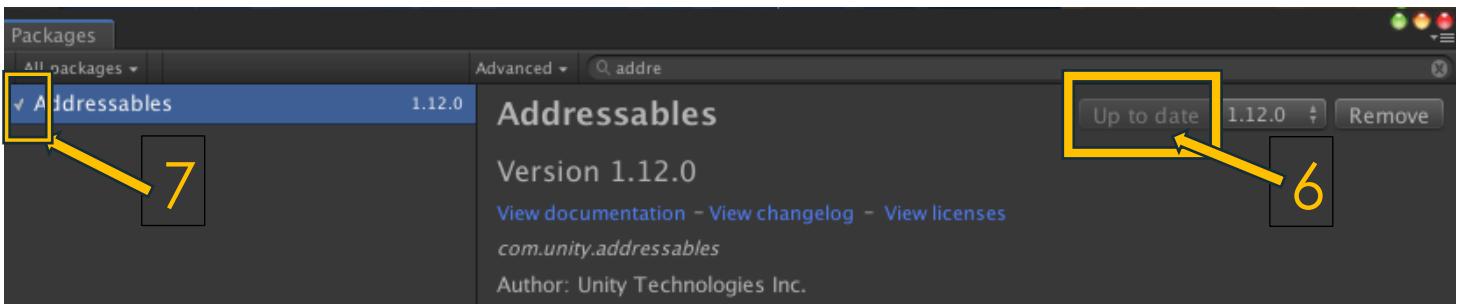
- 1) Open the **Windows** menu option
- 2) Select the option **Package Manager**



- 3) A new window will open, in the search box, Search for **“Addressables”**
- 4) Select the package **“Addressables”**
- 5) Click **Install**, the version that was tested with this package was the most recent (**1.13.1**) at the creation of this documentation.



- 6) When you finish the installation, you will have the module installed and up to date, if it is not up to date, click the update button
- 7) The name will have a tiny mark that indicates it is successfully installed.



3.2 Unity Remote Config

Unity Remote Config is a cloud service that allows you to tune your game design without deploying new versions of your application.

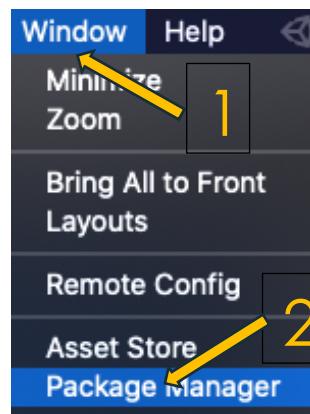
Documentation: You can read more about this module here

(<https://docs.unity3d.com/Packages/com.unity.remote-config@1.2/manual/index.html>)

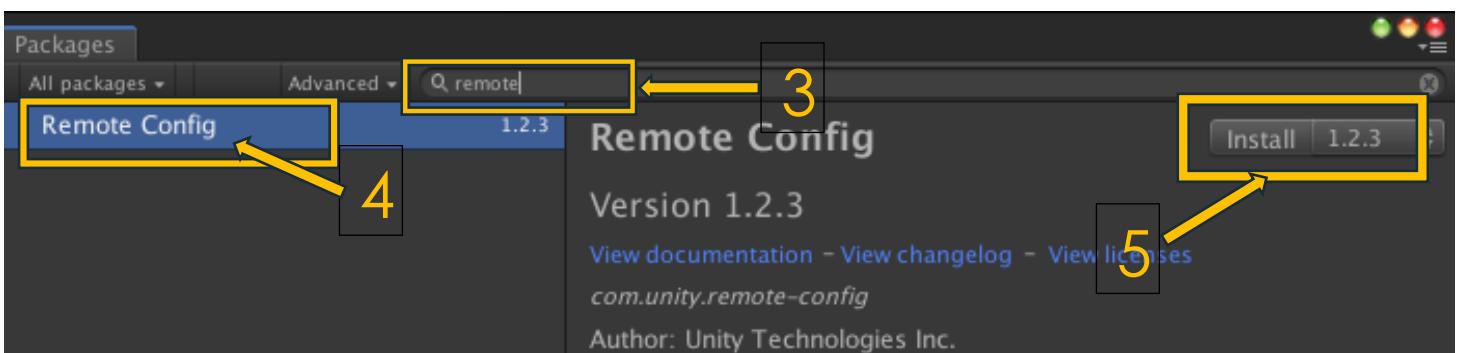
Minimum version: 1.2.3

How to install it:

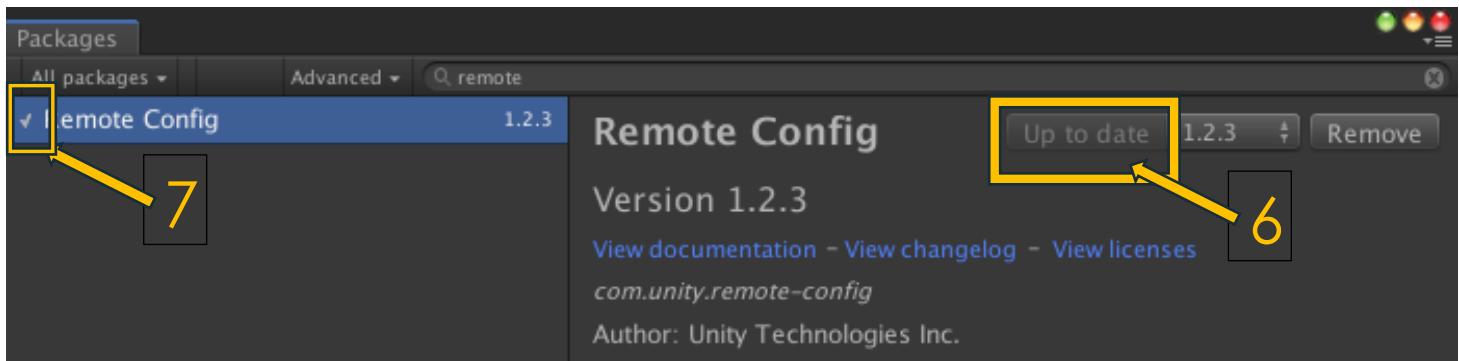
- 1) Open the **Windows** menu option
- 2) Select the option **Package Manager**



- 3) A new window will open, in the search box, Search for “**Remote**”
- 4) Select the package “**Remote Config**”
- 5) Click **Install**, the version that was tested with this package was the most recent (**1.2.3**) at the creation of this documentation.



- 6) When you finish the installation, you will have the module installed and up to date.
- 7) The name will have a tiny mark that indicates it is successfully installed.



3.3 Standard Assets

This collection of assets, scripts, and example scenes can be used to kickstart your Unity learning or be used as the basis for your own projects.

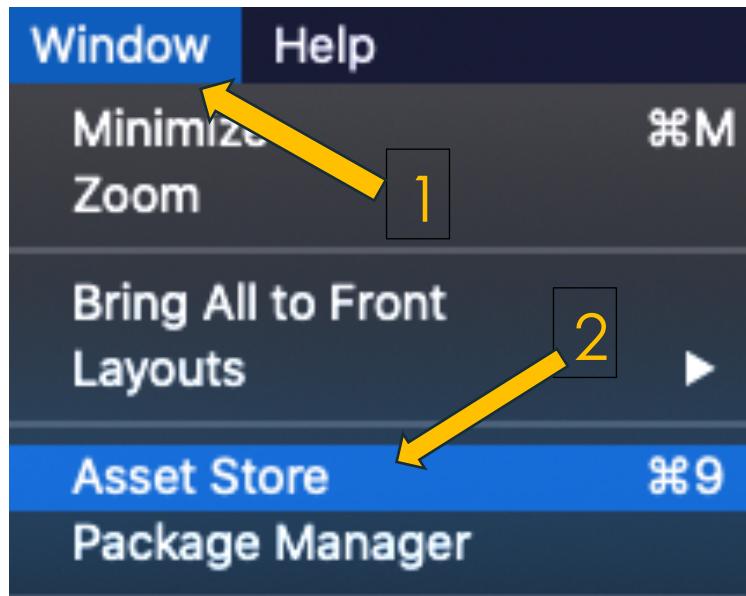
If you want to run the Demo Scenes, you need to download the standard assets.

Documentation: You can read more about this module here

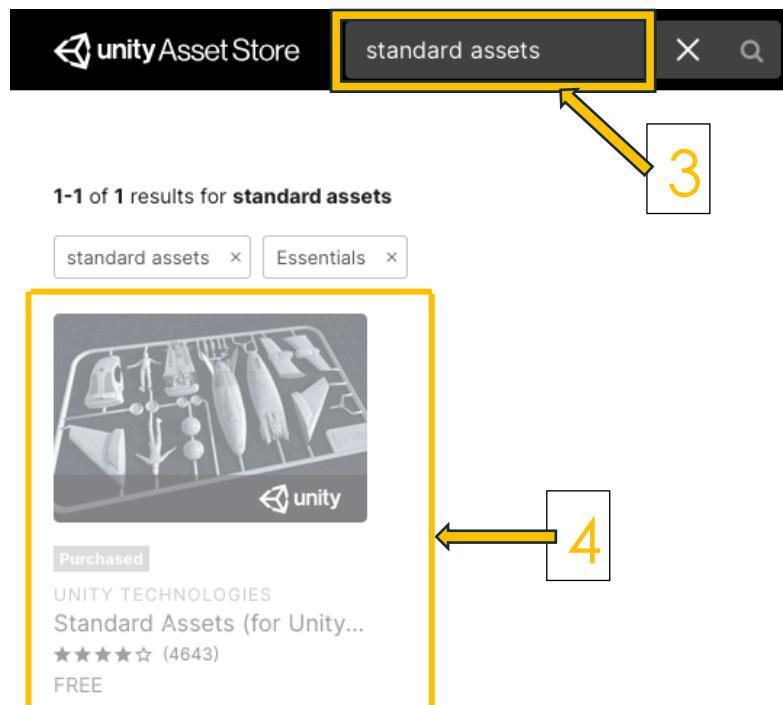
<https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2017-3-32351>.

How to install it:

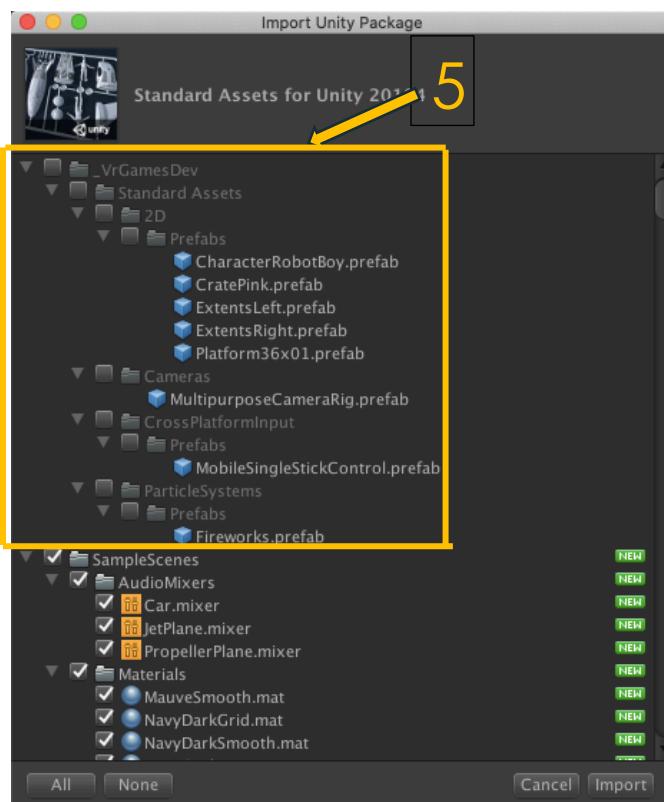
- 1) Open the **Windows** menu option
- 2) Select the option **Asset Store**



- 3) The asset store window will open, in the search box, Search for "**Standard Assets**"
- 4) Select the package "**Standard Assets**" and import like you imported this package



- 5) When you finish the download the import window will pop up, if you notice there are some prefabs that are already imported into the folder “_VrGamesDev”.



The prefabs preloaded are:

1. Standard Assets/2D/Prefabs/CharacterRobotBoy.prefab
2. Standard Assets/2D/Prefabs/CratePink.prefab
3. Standard Assets/2D/Prefabs/ExtentsLeft.prefab
4. Standard Assets/2D/Prefabs/ExtentsRight.prefab
5. Standard Assets/2D/Prefabs/Platform36x01.prefab
6. Standard Assets/Cameras/MultipurposeCameraRig.prefab
7. Standard Assets/ CrossPlatformInput/Prefs/MobileSingleStickControl.prefab
8. Standard Assets/ParticleSystems/Prefabs/Fireworks.prefab



3.3.1 Errors in Unity 2019+ with Standard Assets

Unity 2019 changed the way the **GUI_Text** component works and it is now deprecated (<https://docs.unity3d.com/2018.4/Documentation/ScriptReference/GUIText.html>), and they haven't updated their standard assets package so there are 2 errors that didn't allow to project to compile (<https://forum.unity.com/threads/standard-assets-2018-let-us-know-what-you-think.516595/page-4#post-5458527>)

```

Project Console □ 0 □ 0 □ 2
Clear Collapse Error Pause Editor ▾
[09:54:41] Assets/Standard Assets/Utility/ForcedReset.cs(6,27): error CS0619: 'GUITexture' is obsolete: 'GUITexture has been removed. Use UI.Image instead.'
[09:54:41] Assets/Standard Assets/Utility/SimpleActivatorMenu.cs(10,16): error CS0619: 'GUIText' is obsolete: 'GUIText has been removed. Use UI.Text instead.'

```

Here is how to fix it:

- 1) Double click the first error
- 2) Your code editor will open with the file (Asset/Standard Assets/Utility/ForcedReset.cs)
- 3) Comment the line 6 **//[RequireComponent(typeof (GUITexture))]**
- 4) Save the file and the first error is gone.

```

ForcedReset.cs
No selection
1 using System;
2 using UnityEngine;
3 using UnityEngine.SceneManagement;
4 using UnityStandardAssets.CrossPlatformInput;
5
6 // [RequireComponent(typeof (GUITexture))]
7 public class ForcedReset : MonoBehaviour
8 {

```

- 5) Double click the error remaining and your code editor will open the file (Assets/Standard Assets/Utility/SimpleActivatorMenu.cs)
- 6) In the header section add the code: **"using UnityEngine.UI;"**
- 7) Change the code **"public GUIText camSwitchButton;"** for **"public Text camSwitchButton;"**
- 8) Save the file and the second error is gone

```

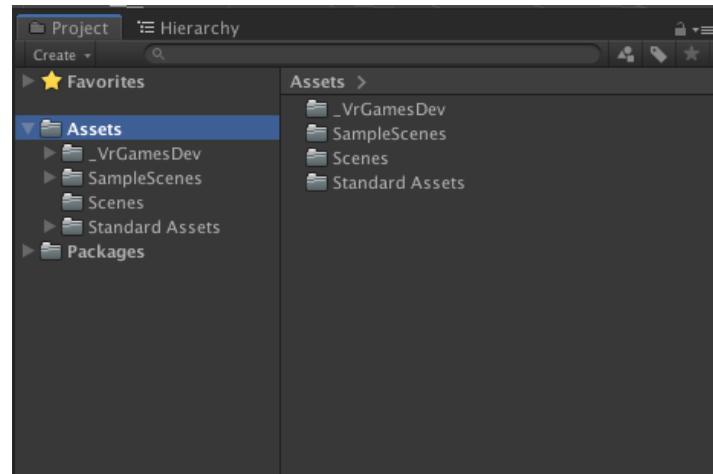
SimpleActivatorMenu.cs
SimpleActivatorMenu > camSwitchButton
1 using System;
2 using UnityEngine;
3 using UnityEngine.UI;
4
5 namespace UnityStandardAssets.Utility
6 {
7     public class SimpleActivatorMenu : MonoBehaviour
8     {
9         // An incredibly simple menu which, when given references
10        // to gameobjects in the scene
11        public Text camSwitchButton;
12        public GameObject[] objects;
13    }

```



4. CONFIGURING ADDRESSABLES

We have no compile errors, and our project looks like the image below, with 4 folders (_VrGamesDev, SampleScenes, Scenes, Standard Assets)



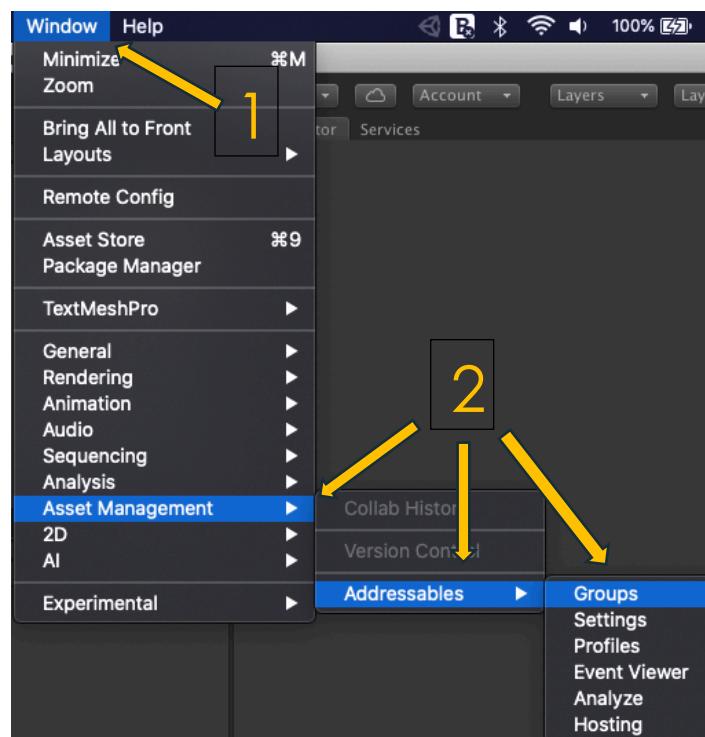
Now let's create our Addressables Groups

4.1 Addressable window

An asset is content that you use to create your game or app. Common examples of assets include Prefabs, textures, materials, audio clips, and animations. Making an asset "Addressable" allows you to use that asset's unique address to call it from anywhere. Whether that asset resides in the local application or on a content delivery network, the Addressable Asset System locates and returns it.

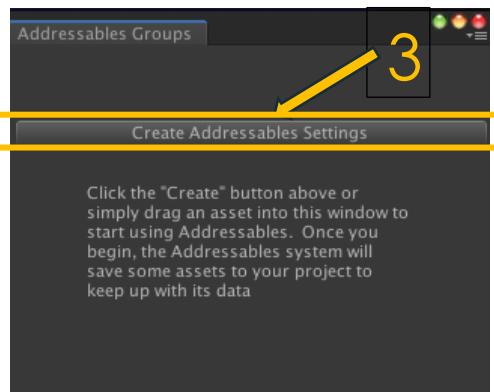
You need to configure the assets to become addressables in the group window.

- 1) Open the **Windows** menu option
- 9) Select the option **Asset Management->Addressables->Groups**

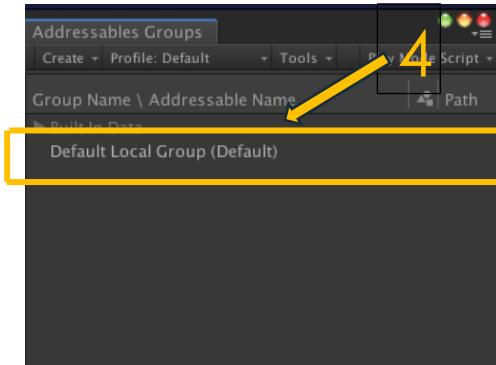




- 10) A pop-up window named “Addressables Groups” will appear, click the “Create Addressables Settings” button.

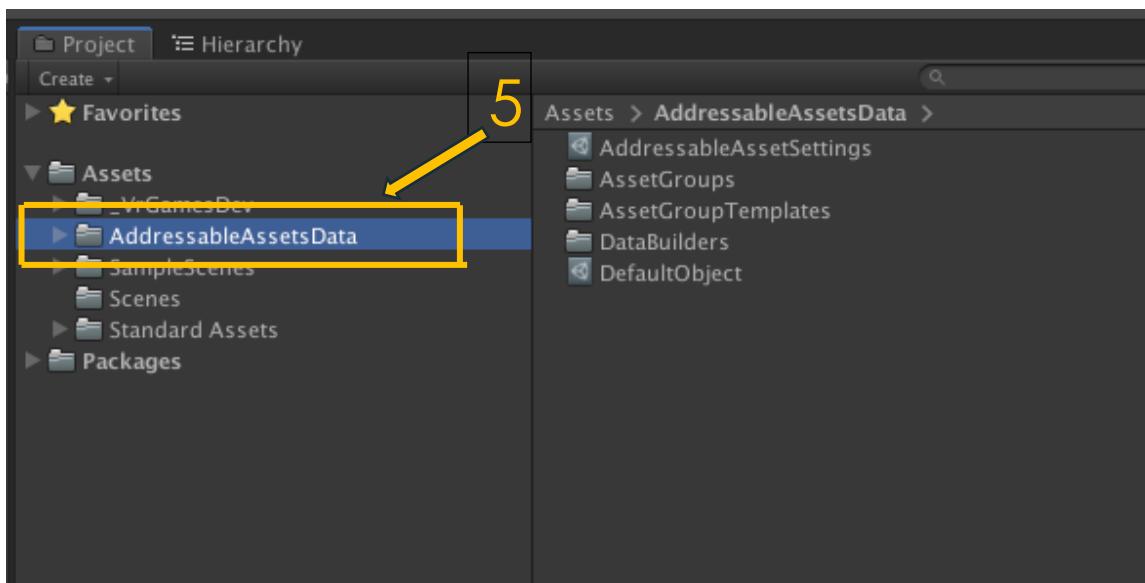


- 11) After it finished installing you will have an addressable usable group named “**Default Local Group (Default)**”



- 12) And a new folder named “**AddressableAssetsData**” was created by the package Addressables, you can read more about this functionality here

<https://docs.unity3d.com/Packages/com.unity.addressables@1.11/manual/AddressableAssetsOverview.html>.

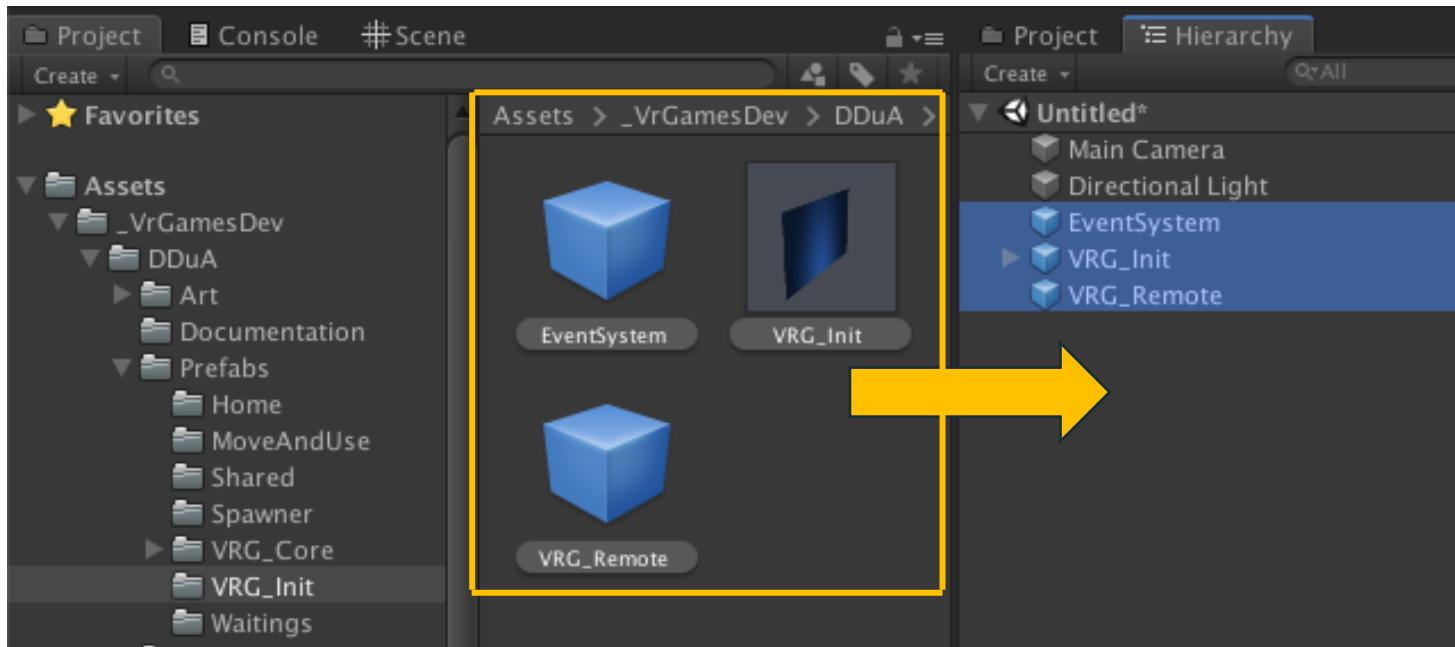




5. CREATING YOUR MAIN SCENE (VRG_INIT)

We can now create our first scene using addressables. Your first scene is responsible of loading all the assets, and it will delete itself once the game is loaded. The object needed for this job is the “**VGR_Init**”

- 1) Create a new scene and name it “**Main**” and drag the prefabs “**EventSystem**”, “**VRG_Init**” and “**VRG_Remote**” located in the folder (`_VrGamesDev/DDuA/Prefabs/VRG_Init`)



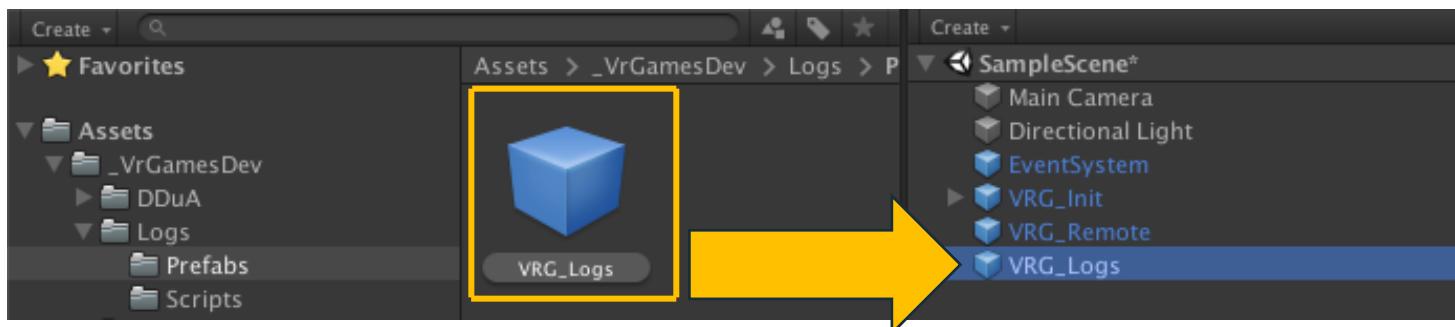
EventSystem: This allows to have a single event system manager, it is the Unity’s default Eventsystem, but it is a non-destroy on load object.

VRG_Remote: This prefab, provides the local configuration from the remote server, for easier developing and debug.

VRG_Init: This prefab, check for internet connection, and waits until there is a network ready, then proceed to load the **VRG_Core**, to load the game content asynchronously.

- 2) Drag the prefab “**VRG_Logs**”, located in the folder (`_VrGamesDev/Logs/Prefabs`)

VRG_Logs: Log module, It is useful to debug, since the technology is asynchronous, and it is hard to track the flow of the program.

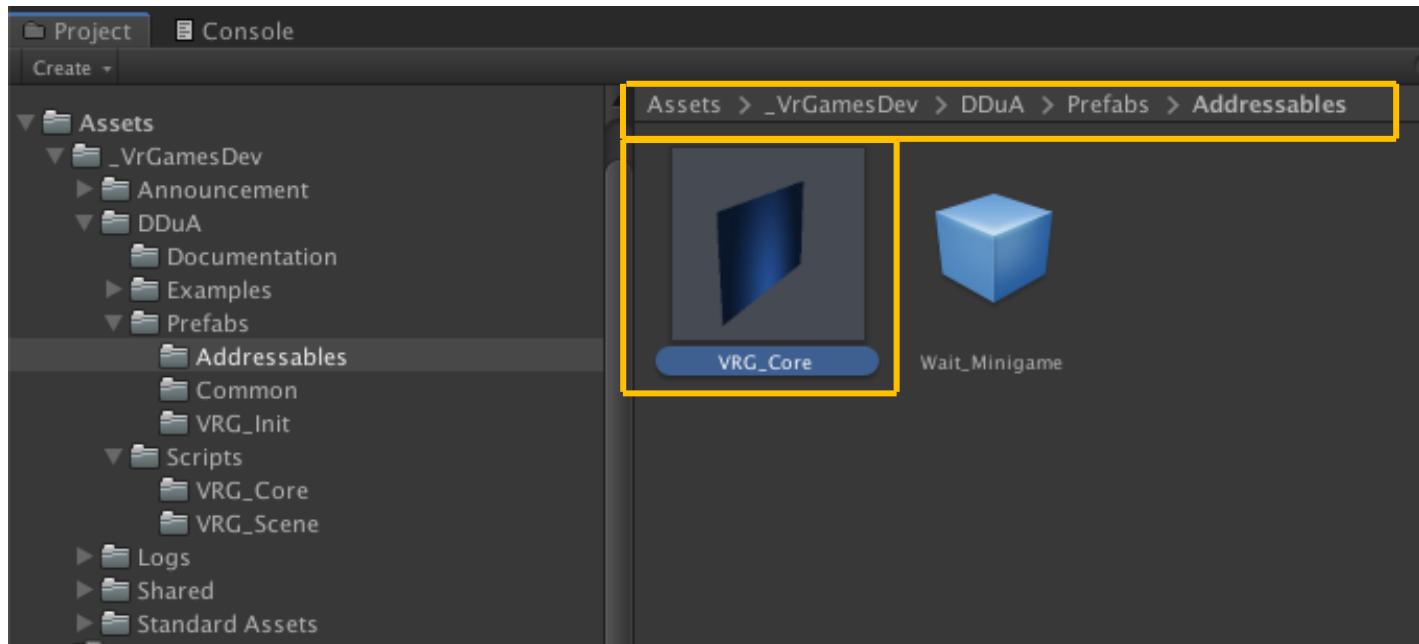




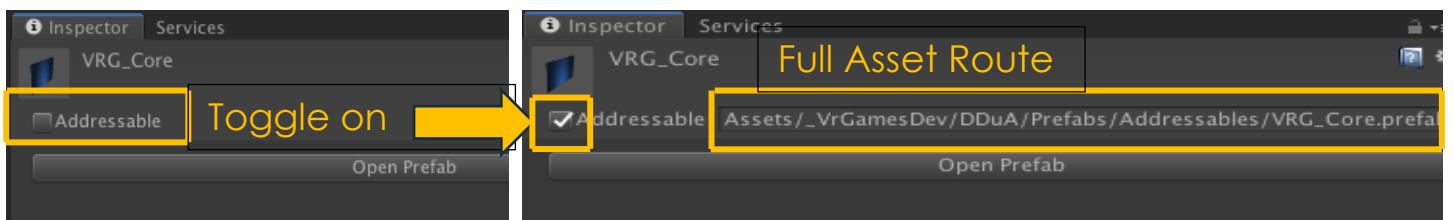
5.1 VRG_Core

VRG_Core: This prefab is the core class of the package, it handles the Addressable technology, including, scenes, prefabs, **Carousels** and control the flow of the download, informing the player with a loading slider bar.

- 1) Browse to the folder (_VrGamesDev/DDuA/Prefabs/Addressables/) and select the Prefab “**VRG_Core**”

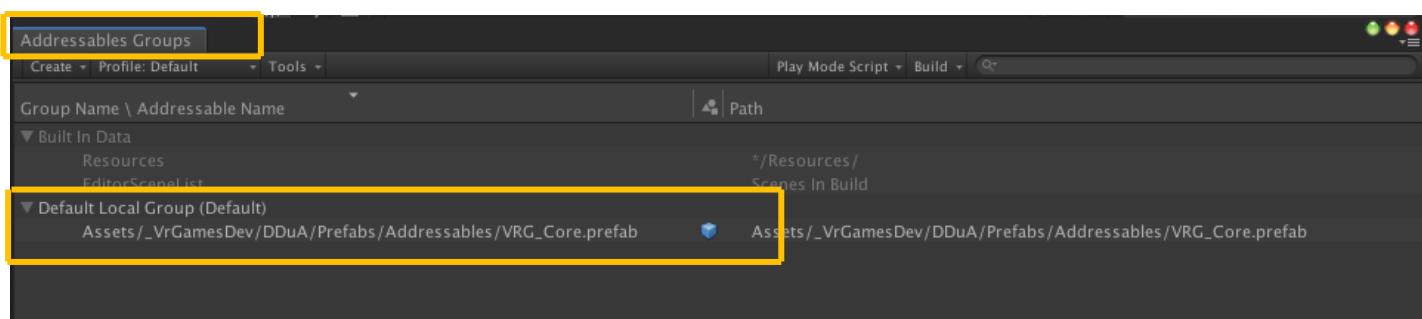


- 2) Open the “**Inspector**”, and click the **VRG_Core** prefab, toggling the “Addressable” option on, the route of the asset will be filled after you toggle the option.



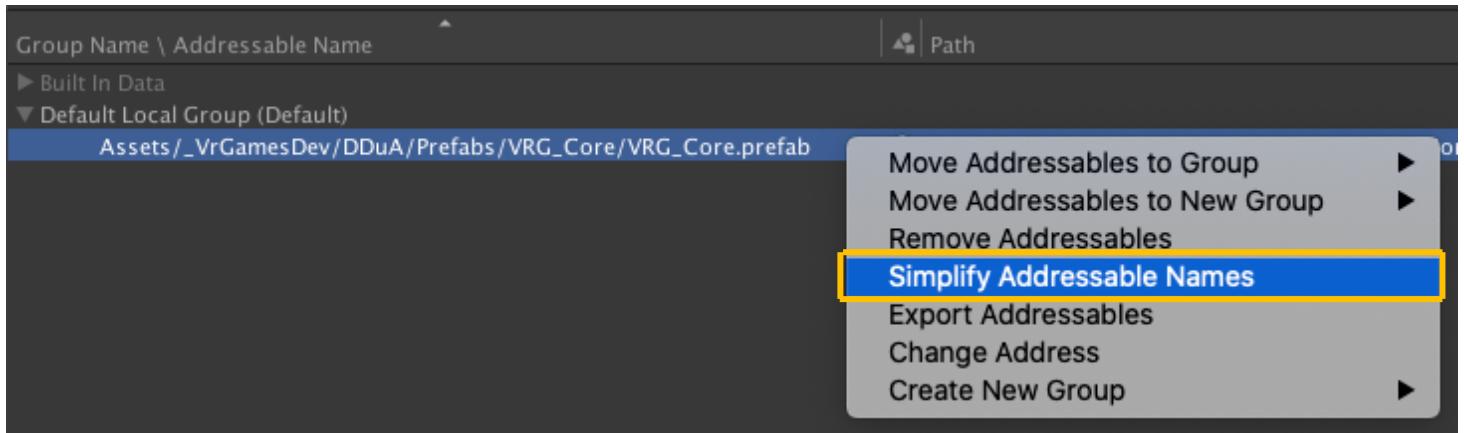
With this easy action, you declare any asset as “Addressable”, for more information here
<https://docs.unity3d.com/Packages/com.unity.addressables@1.11/manual/AddressableAssetsGettingStarted.html#marking-assets-as-addressable>.

- 3) Select the “**Addressables Groups**” window/tab, the asset “**VRG_Core**” is now under the “**Default Local Group**”, with its full path name (_VrGamesDev/DDuA/Prefabs/Addressables /VRG_Core.prefab).

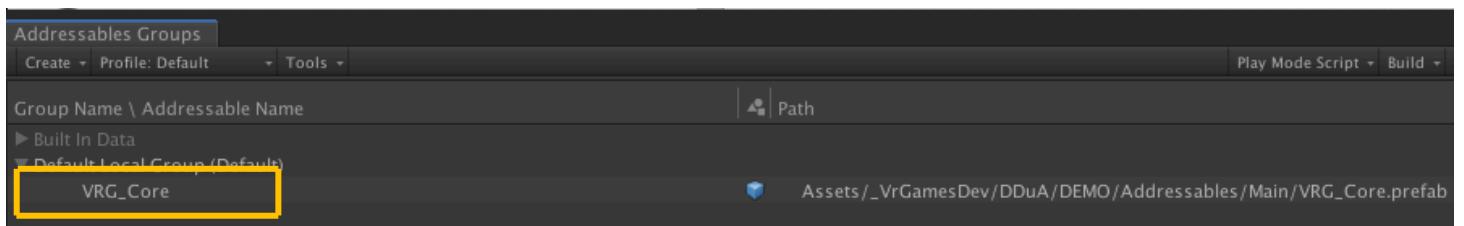




- 4) Select it, and right click to simplify the name to “**VGR_Core**”, click on the “**Simplify Addressable Names**”

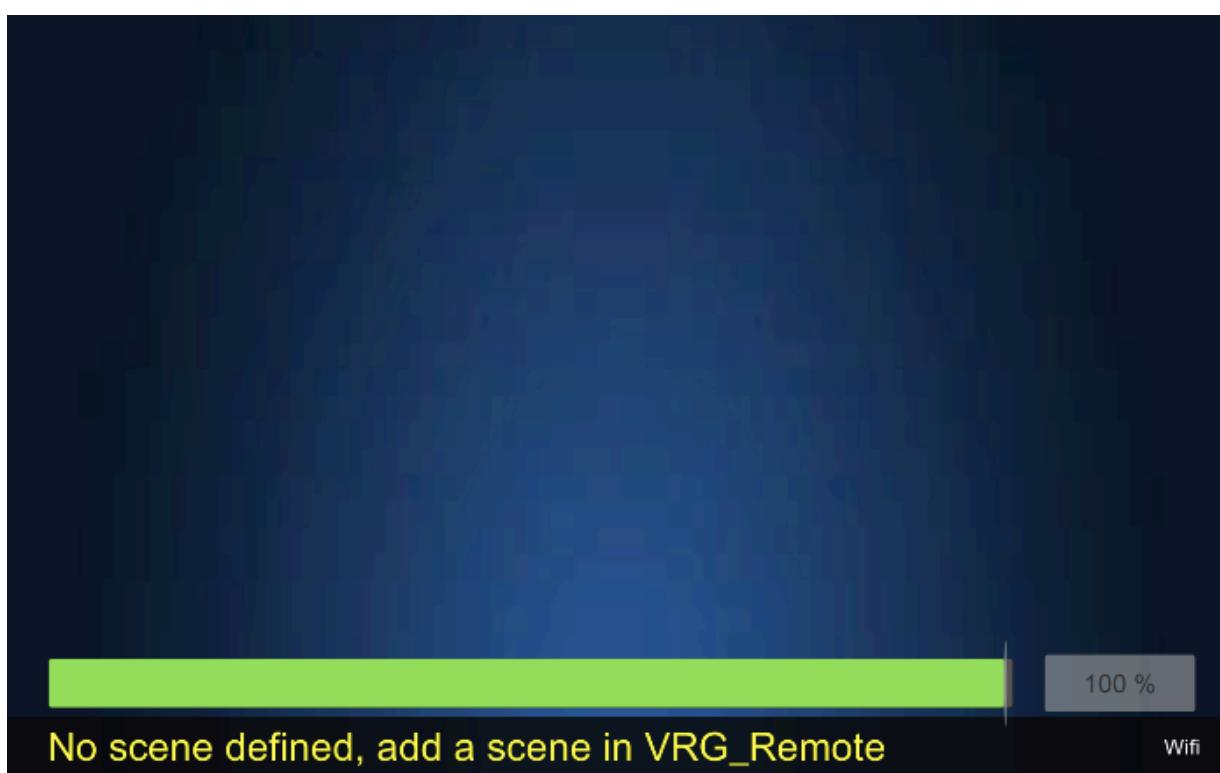


- 5) The name is changed to “**VGR_Core**”, this is the “address” of the asset, you will need it, to call it anywhere, you can read more about it here. (<https://docs.unity3d.com/Packages/com.unity.addressables@1.11/manual/AddressableAssetsGettingStarted.html#loading-or-instantiating-by-address>).



- 6) Now play the scene, you will load your first addressable the “**VRG_Core**”. You run the scene and got a warning message:

No scene defined, add a scene in VRG_Remote





6. CAROUSELS (VRG_CAROUSEL)

Carousels are a special kind of prefab, you will probably download assets that takes some seconds/minutes.

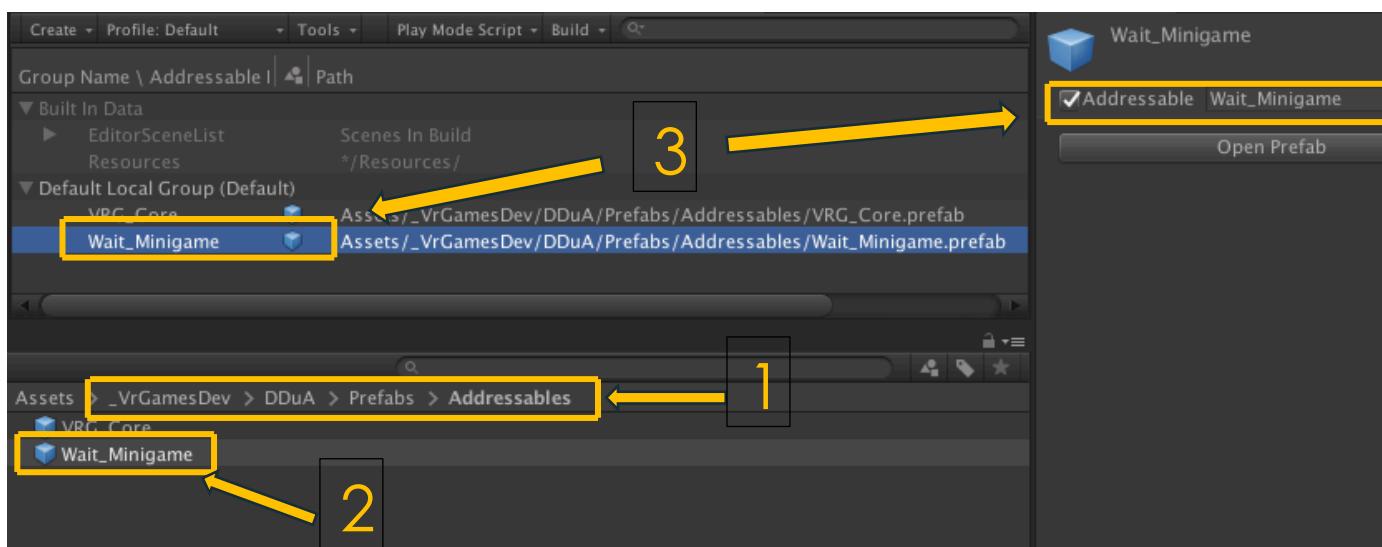
Depending of the quality and speed of a player's connection this could take many minutes to finish, meanwhile you can interact and communicate with your player base using "**Carousels**".

You can use them to:

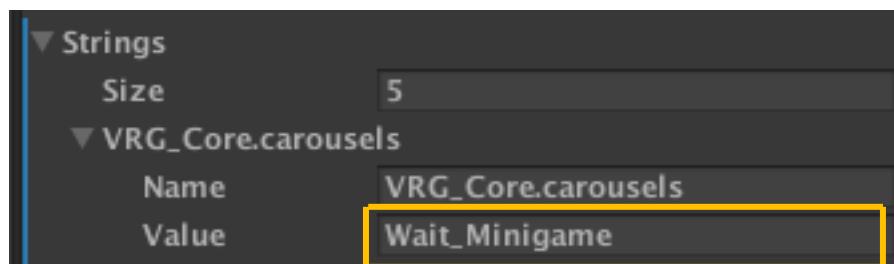
- Promote your company brand name
- Your game title, news
- A splash art
- Promotions
- Seasonal updates
- Tips for better performance
- Scene details
- MINIGAMES!!! (since the Carousels are prefabs you can create games to interact with your players)

6.1 Add your first Carousel

- 1) Browse to the folder (_VrGamesDev/DDuA/Prefabs/Addressables/)
- 2) Select the Prefab "**Wait_Minigame**"
- 3) Make it a name simplified addressable (Repeat the same process from **5.4** up to **5.7**)

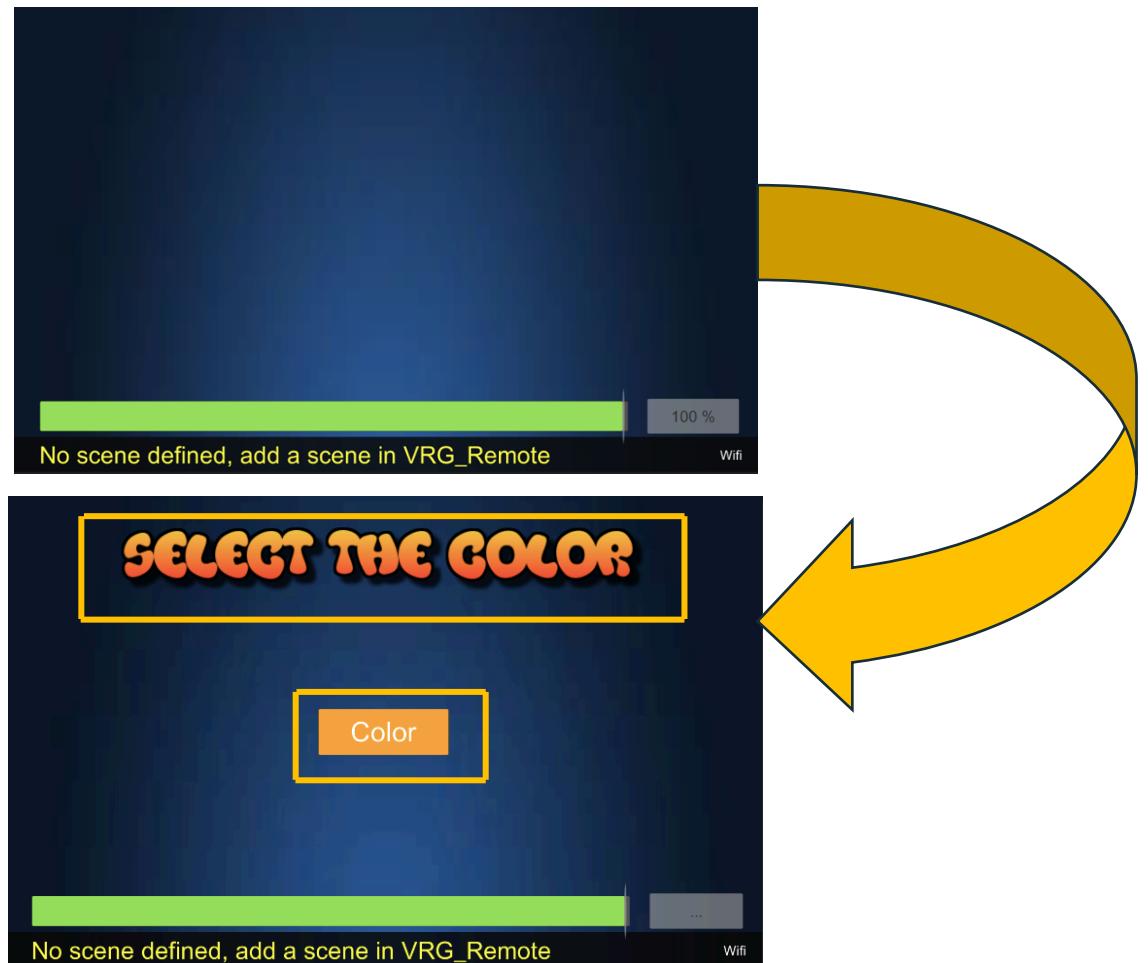


- 4) Configure the "**VRG_Remote**": we will use the "**Wait_Minigame**" as a "**Carousels**" object. Select in the hierarchy the object **VRG_Remote**, and in the inspector fill the "value" data from the "**Strings -> VRG_Core.Carousels**" struct, with the simplified address of your Carousel.





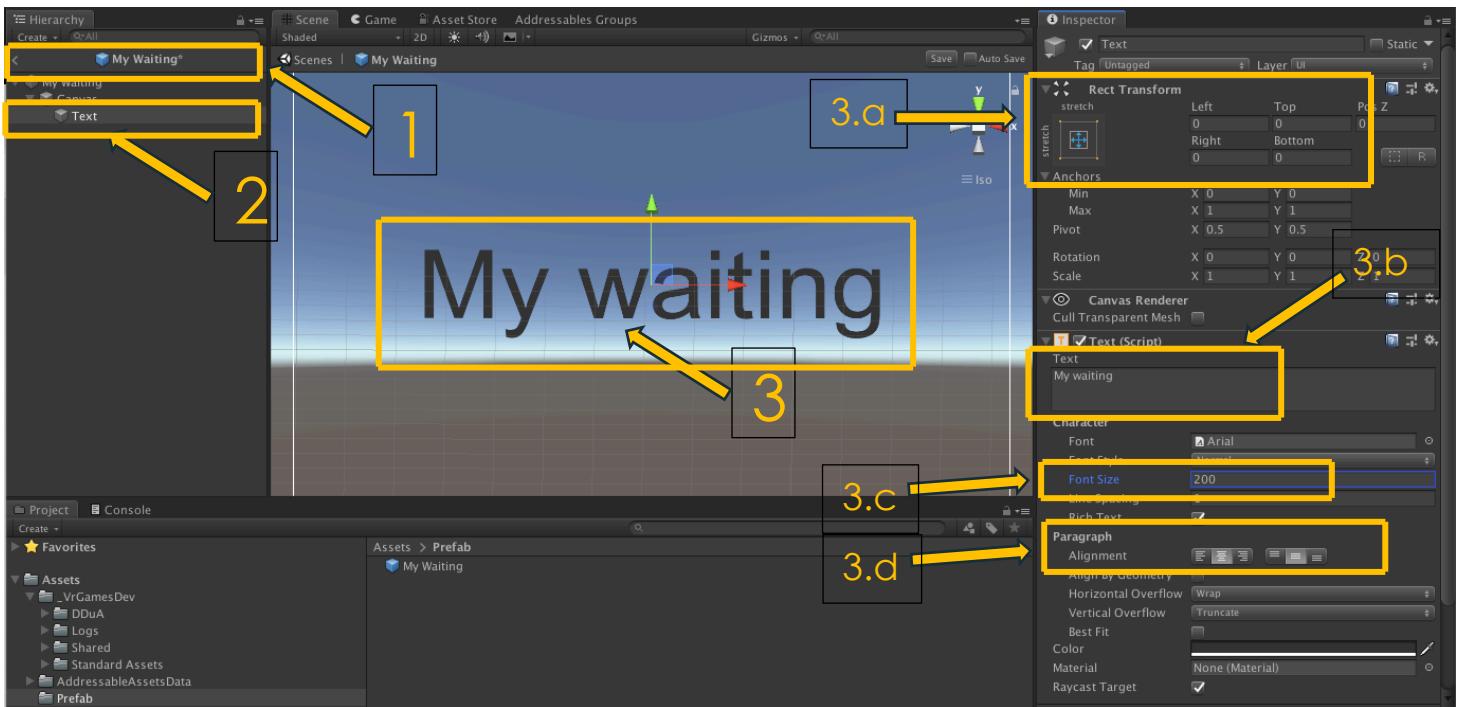
- 5) Play the game, you will see a very nice and simple Carousels prefab, to entertain your players while they wait for the download to finish



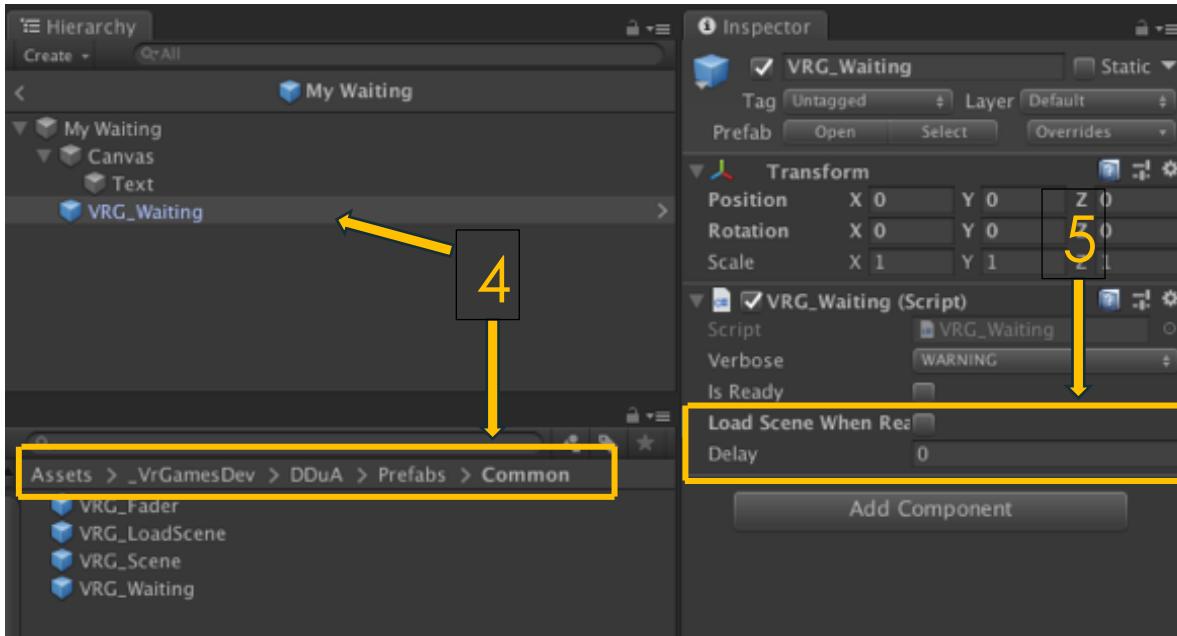
6.2 Create your VRG_Carousel

You can create your own **VRG_Carousel** objects, lets create a simple text Carousel.

- 1) Create an empty object, name it "My Carousel", and make it a prefab.
- 2) Add a UIText object
- 3) Custom the text to display "My Carousel" in the whole screen, centered and white colored
 - a) Modify the RectTransform to stretch, left, top, right and bottom to 0
 - b) In text box type "My Carousel"
 - c) In the Character->Font size change it to 200
 - d) In the Paragraph->Alignment change it to mid, center



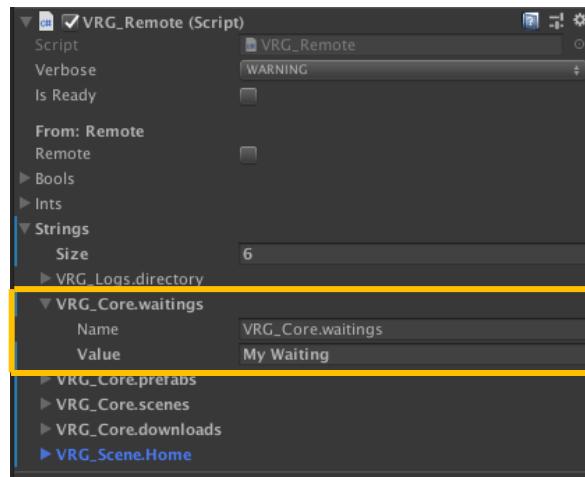
- 4) Add a “VRG_Carousel” Prefab, this prefab is in the folder (Assets/_VrGamesDev/DDuA/Prefabs/Common)



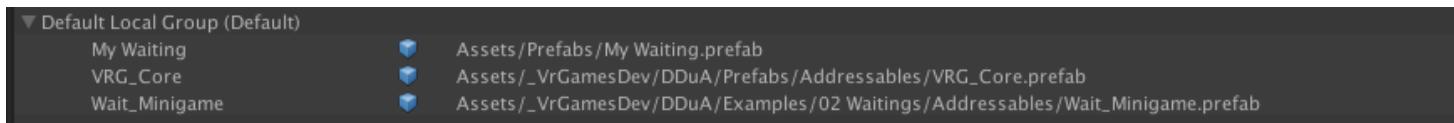
- 5) The VRG_Carousel has 2 parameters to custom

- Load Scene When Ready:** This is a toggle box, it informs the VRG_Core to load the scene when it is ready or to wait for an action from the Carousel.
- Delay:** The time in seconds it takes to change for another Carousel or to load the scene, if 0, the delay is infinite.

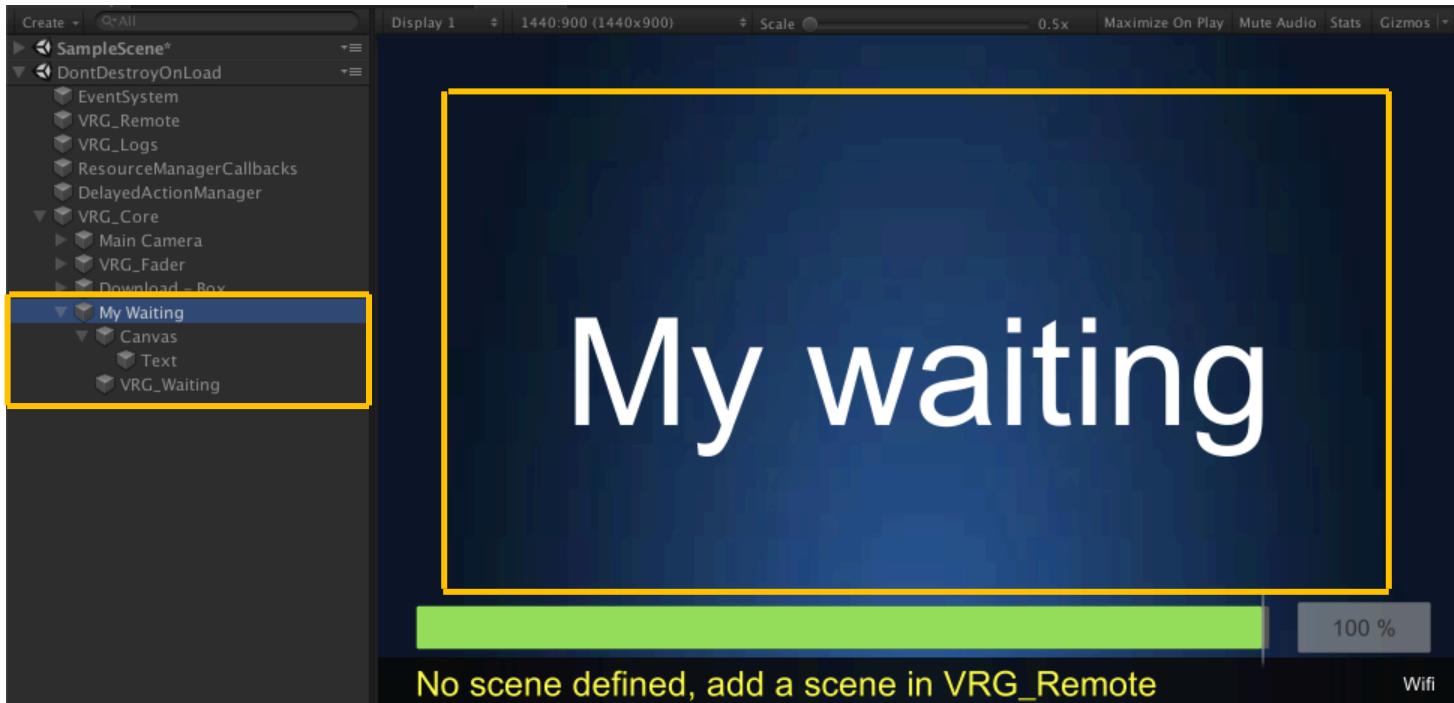
- 6) Add the name “My Carousel” to the VRG_Remote->VRG_Core.Carousels



7) Make your prefab into a name simplified addressable



8) Play the Game, and watch your newly created Carousel under the **VRG_Core** Object





6.3 Here comes a new challenger

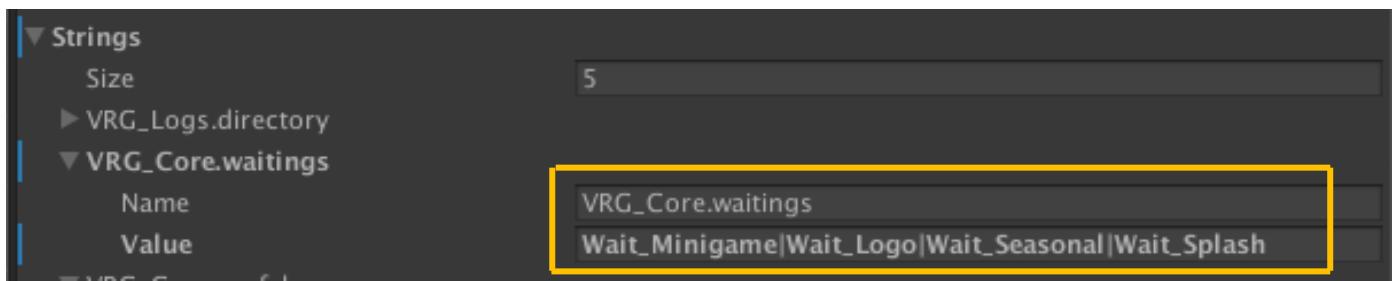
Challenger: Add more “**Carousels**” to the settings list *

You can custom the time to wait until you load a new one, or wait until the player finish the mini game.
You can find the preconfigured “**Carousels**” in the following folder (_VrGamesDev/DDuA/Examples/02 Carousels/Addressables)

We added 6 preconfigured Carousels as example (check Example 02):

- Wait_Logo
- Wait_Minigame
- Wait_MoveAndUse
- Wait_Seasonal
- Wait_Spawner
- Wait_Splash

- 1) Go to the Carousels preconfigured directory (_VrGamesDev/DDuA/Examples/02 Carousels/Addressables)
- 2) Make name simplified addressables all the following Carousels.
 - a) **Wait_Minigame**
 - b) **Wait_Logo**
 - c) **Wait_MoveAndUse**
 - d) **Wait_Seasonal**
 - e) **Wait_Spawner**
 - f) **Wait_Splash**
- 6) Assign all of them to the **VRG_Core.Carousels** in the "**VRG_Remote**" object from the hierarchy using the “|” pipe character to separate the different data
 - a) The final text will look something like this: “**Wait_Logo | Wait_Spawner | Wait_Seasonal | Wait_Splash | Wait_MoveAndUse | Wait_Minigame**”



***Pro tip:** Remember to make them Addressables

- 7) Save the scene and Play the game and enjoy your Carousel.

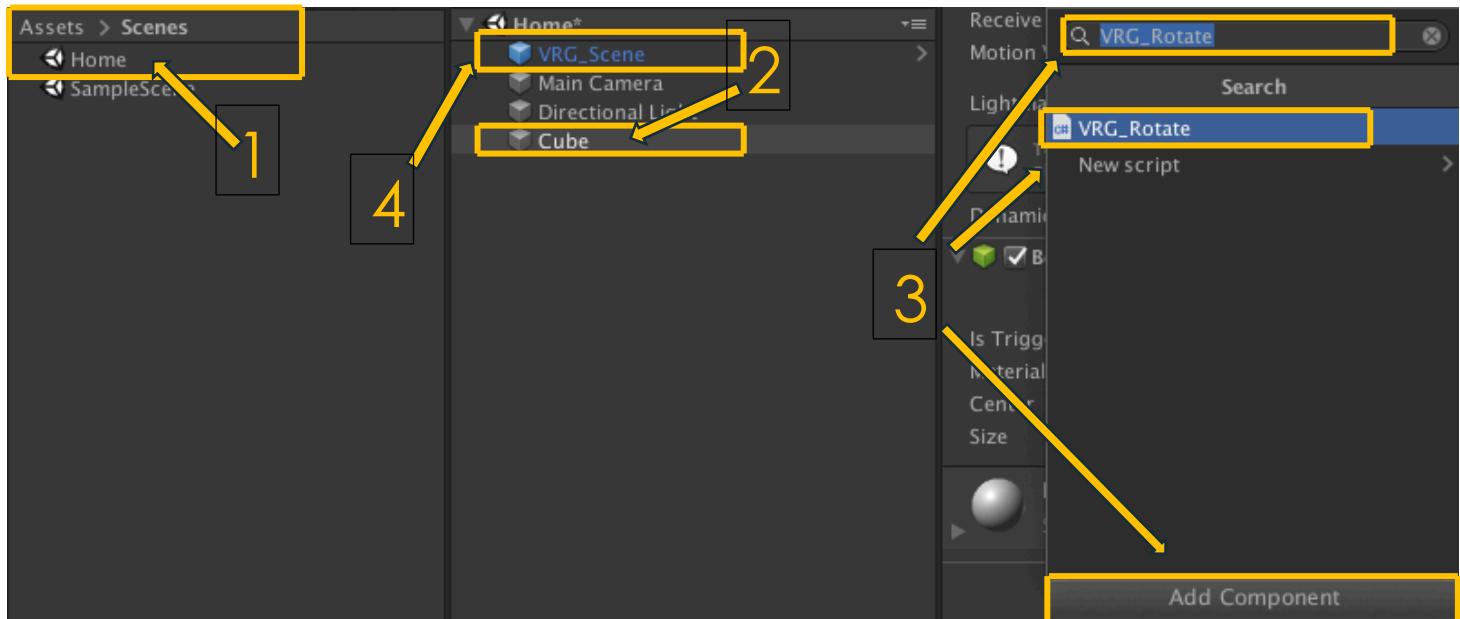


7. SCENES AS AN ADDRESSABLE (VRG_SCENE)

When the **VRG_Core** (as explained in the [5.1 VRG_Core](#) section) load is over and Carousels are running, you need to transition to your first scene from the Scene list **VRG_Core.scenes** in the “**VRG_Remote**” prefab.

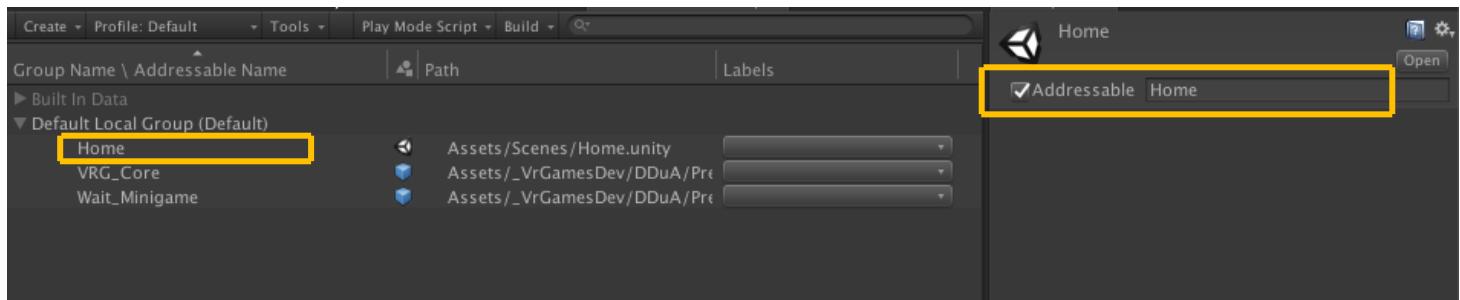
7.1 Add your first scene

- 1) Create a new scene, name it “**Home**”, you can save it in the “**Assets/Scenes**” folder.
- 2) Create a 3d Object -> Cube
- 3) Add a component to the cube, search for “**VRG_Rotate**”, and add it.
- 4) Drag and drop a prefab named “**VRG_Scene**” It is located into (`_VrGamesDev/DDuA/Prefabs/Common`).



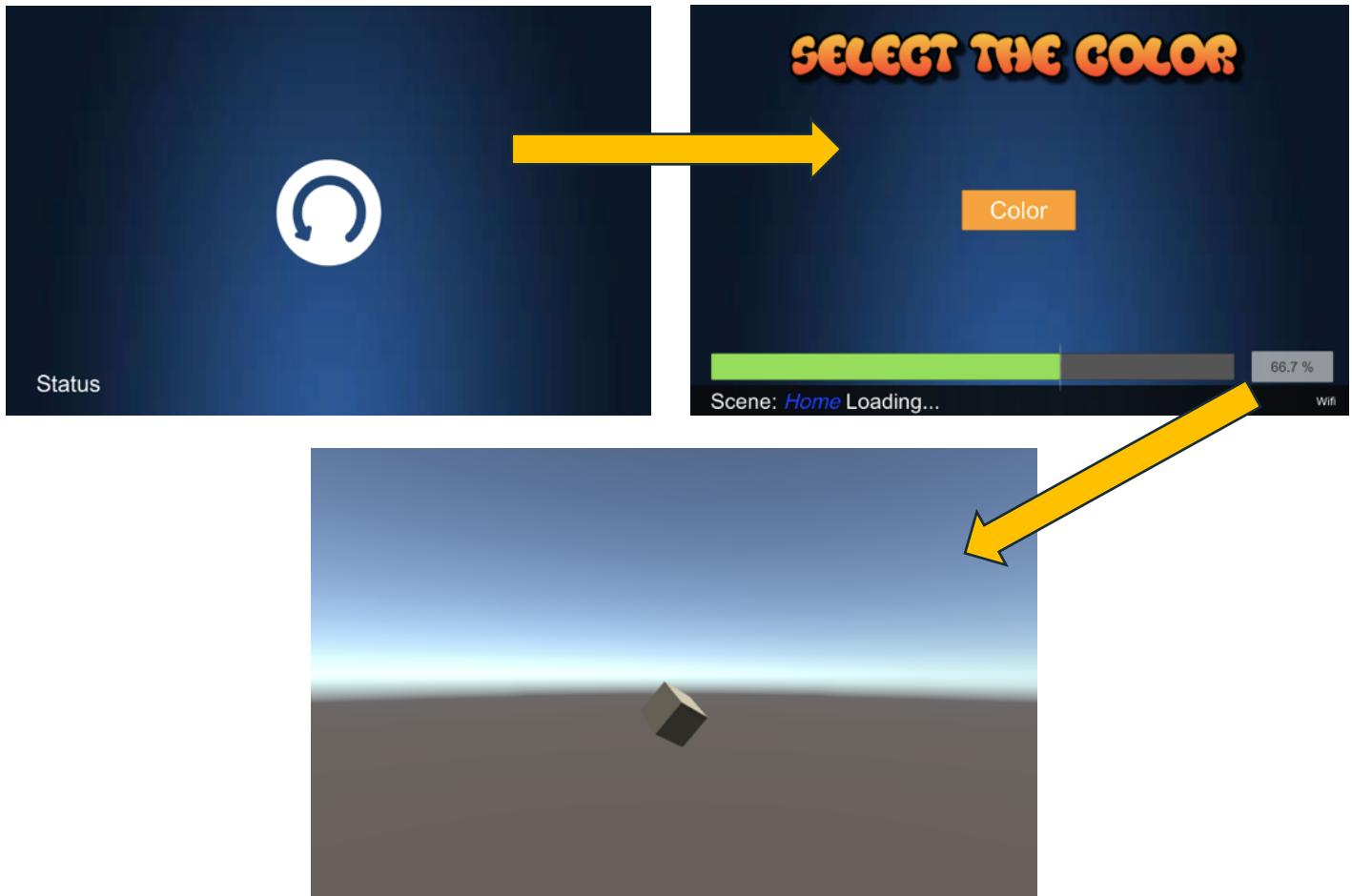
VRG_Scene: This prefab is needed to any scene that is Addressable, it makes sure everything is instantiated, and handle the scene to preload the assets needed, either from Cache or Downloading it.

- 5) Make your newly created “Home” Scene a name simplified addressable, this is how your Default group looks like.



- 6) Go to your “Main” Scene and select the VRG_Remote object, in the **Strings->VRG_Core.scenes**, setting, set the value to “Home”, the address of your “Home” scene.
- 7) Play your Main Scene!

Congratulations!

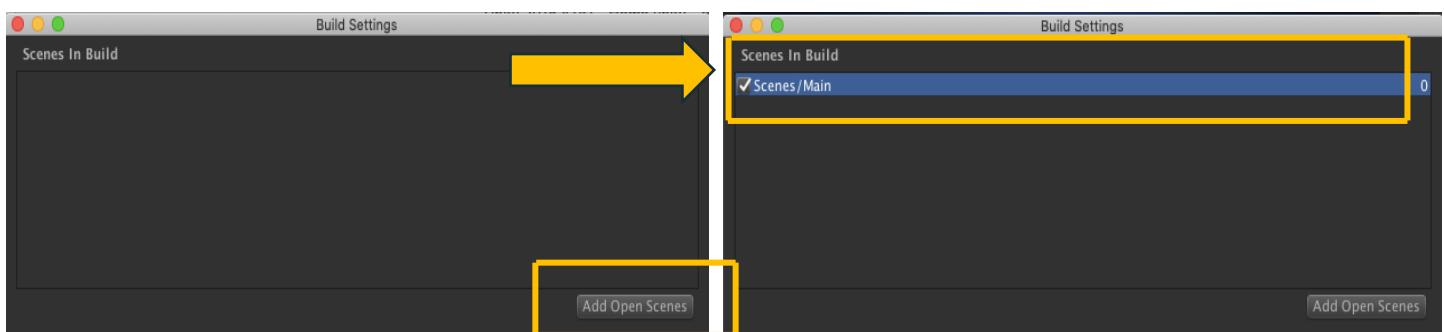


You have a full project loaded from Addressables, and you are watching your spinning cube.

7.2 Add your VRG_Init scene to the build list

Now that you created your first scene, let's understand how to customize it to suit your needs.

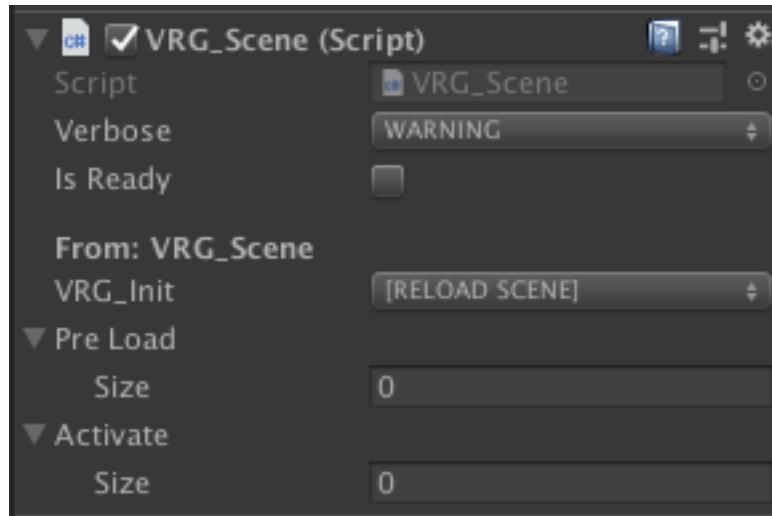
First of all, check your scenes in build list (<https://docs.unity3d.com/Manual/BuildSettings.html>) you will notice there are no scenes loaded, when you work with Addressables, you can load them dynamically and asynchronous, open the Scene "Main" that have the "VRG_Init" Object and add it to the list, this is needed since you will need to build the game with this scene included.





7.3 Configure your VRG_Scene

You can custom what objects to preload in your new scene.



- **VRG_Init:** The name of the scene that holds the **VRG_Init** object prefab, select the “**Main**” option.
- **Pre Load:** The addressables that needs to be downloaded before the scene is activated.
- **Activate:** When the scene is ready, the objects in this list will be activated “**GameObject.SetActive(true);**” These elements are gameObjects already in the scene **IT IS HIGHLY RECOMMENDED TO START THEM AS INACTIVE**, that way the awake method will not run until the scene is fully loaded.

7.4 Configuring the PreLoads

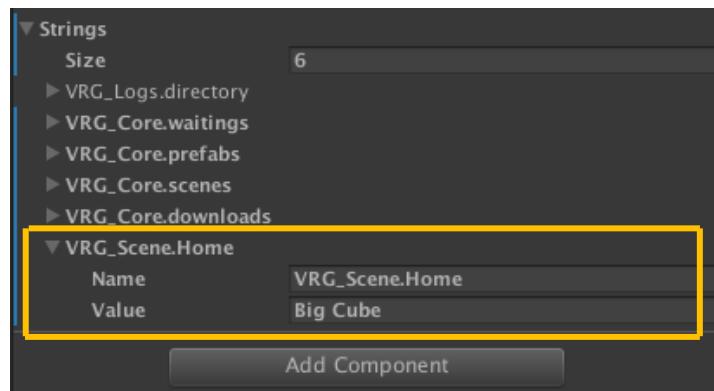
You configure the addressable prefabs needed when you are preparing the scene. These are all the addressables objects from the scene. Configure the setting in the “**VRG_Remote**” prefab in the Main Scene.

Create a setting for your scene, use the “**VRG_Scene**” prefix followed by the scene name, in this case it will be “**VRG_scene.Home**”.

Format: {String}

Name: VRG_Scene.{scene_name} from addressable group}

Value: {The adressables you want to instantiate, separated by the “ | ” pipe character}

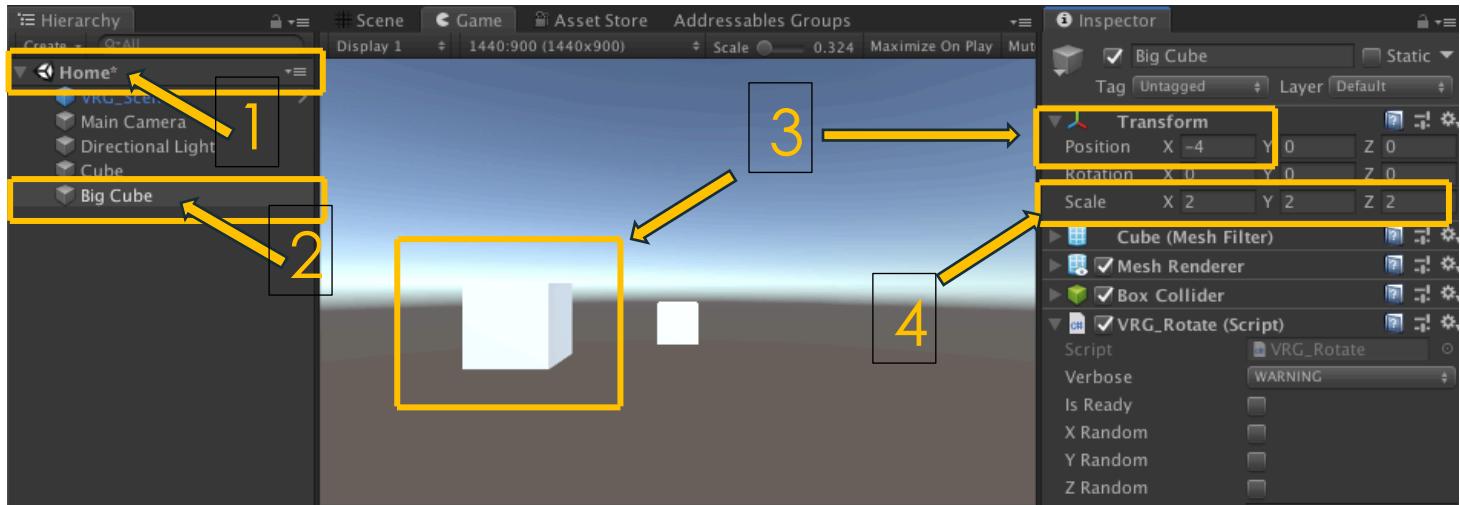




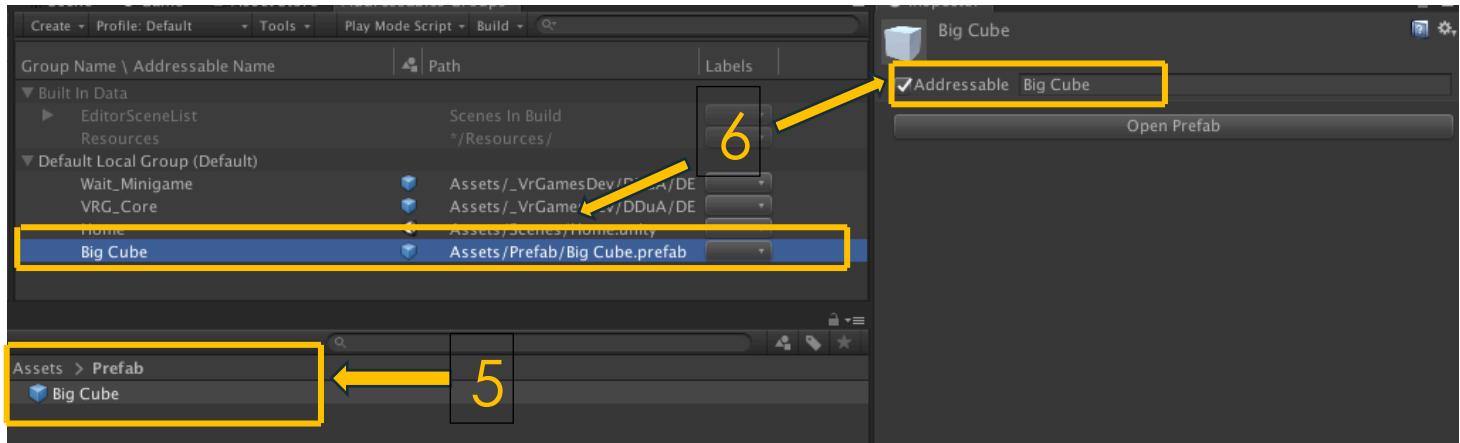
7.5 Here comes a new challenger

Challenge: Populate your “**Home**” Scene with a new addressable and load it asynchronous while creating the scene, and keeping the scene camera and directional lights deactivated until the scene is fully loaded.

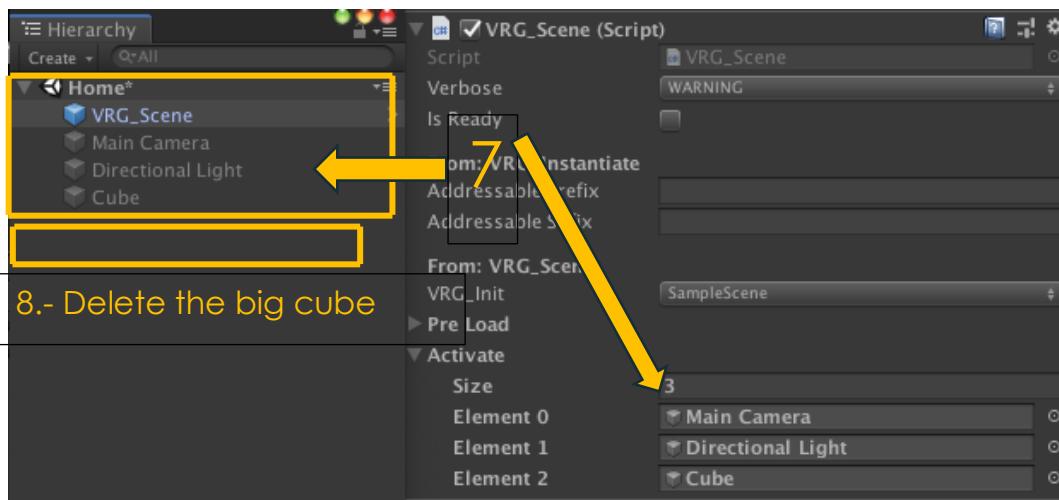
- 1) Open your scene “Home”
- 2) Duplicate your rotating cube “**Cube**” (ctrl+D or cmd + D), name it “**Big Cube**”
- 3) Move it -4 in X
- 4) Duplicate its size



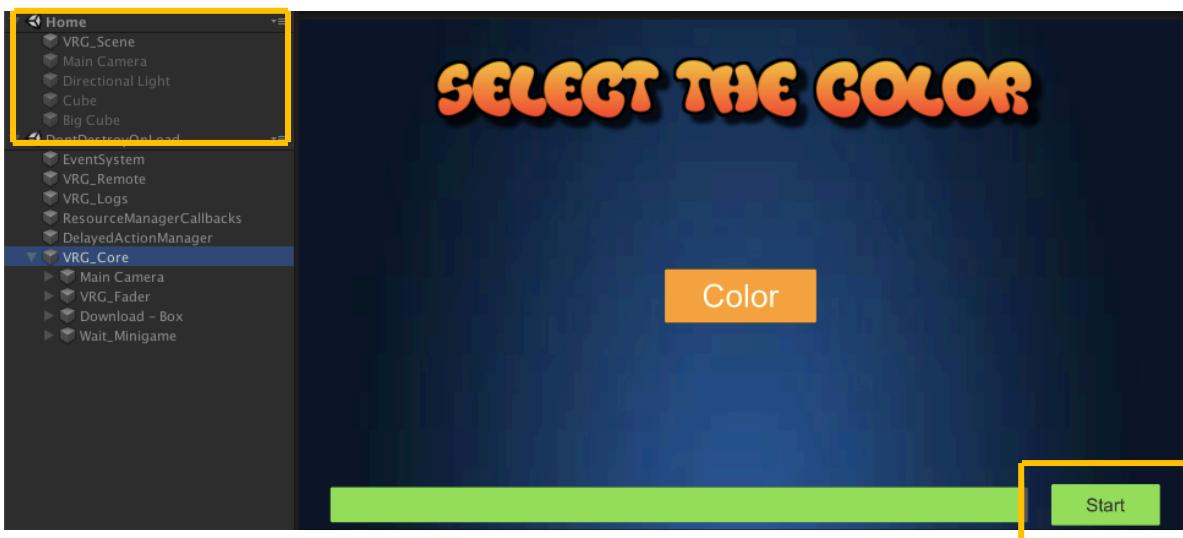
- 5) Create a folder named “Prefabs” in your root Assets folder and create a prefab of your new Big Cube,
- 6) Make it addressable, and simplify it



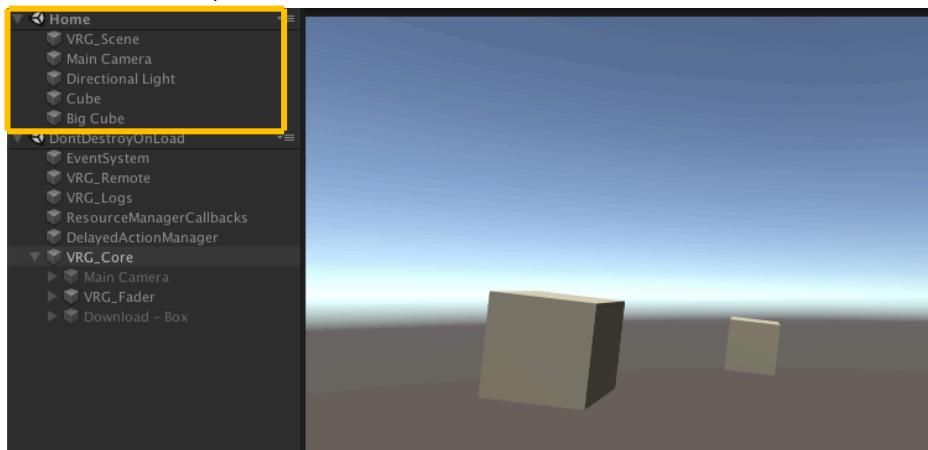
- 7) Deactivate your camera, and your directional light, and drag and drop it, in your **Activate** array
- 8) Delete your big cube from the scene hierarchy.



- 9) Inform the **VRG_Remote** in the “**Main**” scene your **String->VRG_Scene.Home** will use the new **Big Cube** as explained in [7.4 Configuring the Preloads](#) section
- 10) Play your sample Scene.
- 11) Before you click the “Start” button, notice how the scene is loaded with the new “**Big Cube**” and every object is deactivated Carousel for the start command.



- 12) **Congratulations!** You have a super new addressable scene with remote defined addressables.



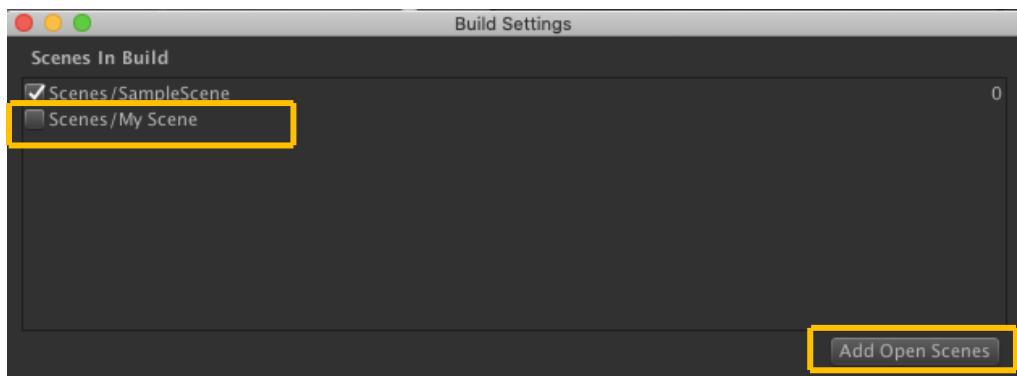


8. CHANGE SCENES (VRG_LOADSCENE)

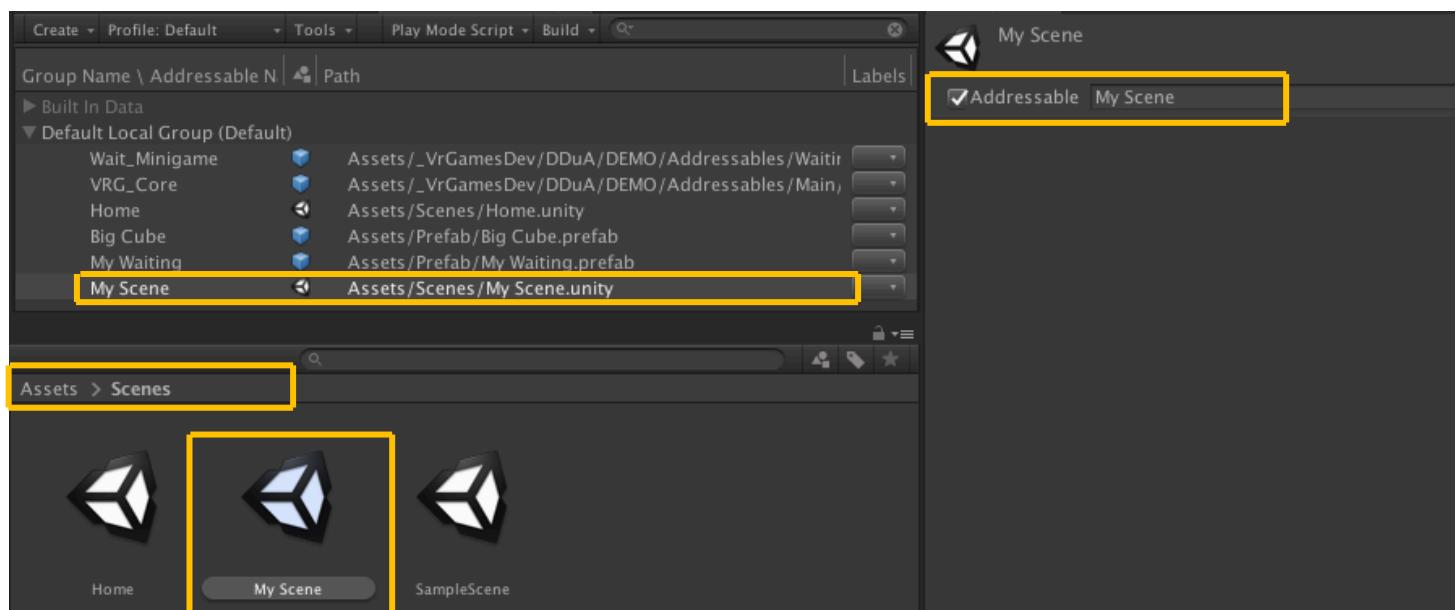
Ok, we have a full system using addressables and dynamic loading of the assets of the game. Now let's move between different scenes. You can read more about it [here](#)

(https://docs.unity3d.com/Packages/com.unity.addressables@1.11/api/UnityEngine.AddressableAssets.Addressables.html#UnityEngine_AddressableAssets_Addressables_LoadScene_System_Object_LoadSceneMode_System_Boolean_System_Int32_).

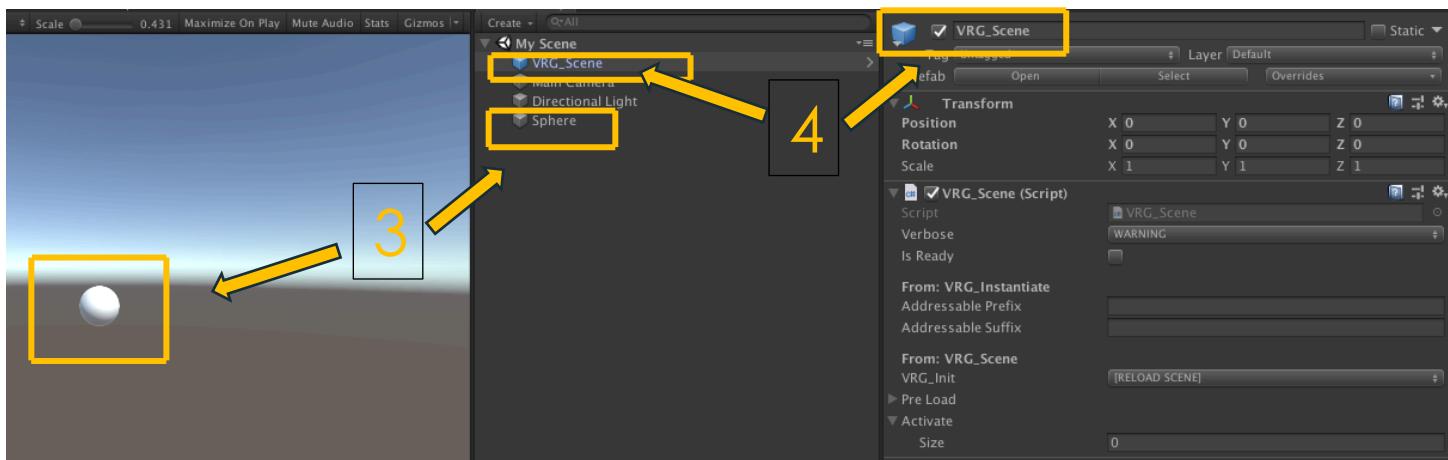
- 1) Create a second scene, name it "My Scene" and add it to your scene build settings



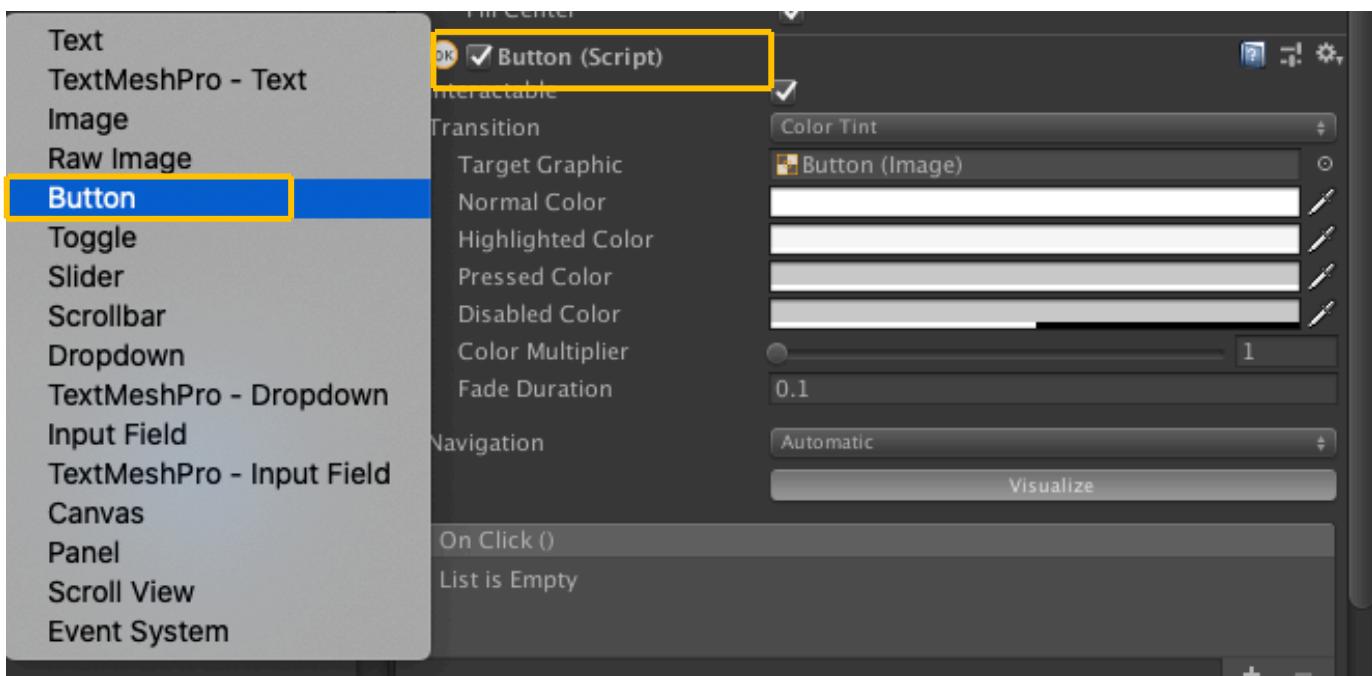
- 2) Make it a name simplified addressable.



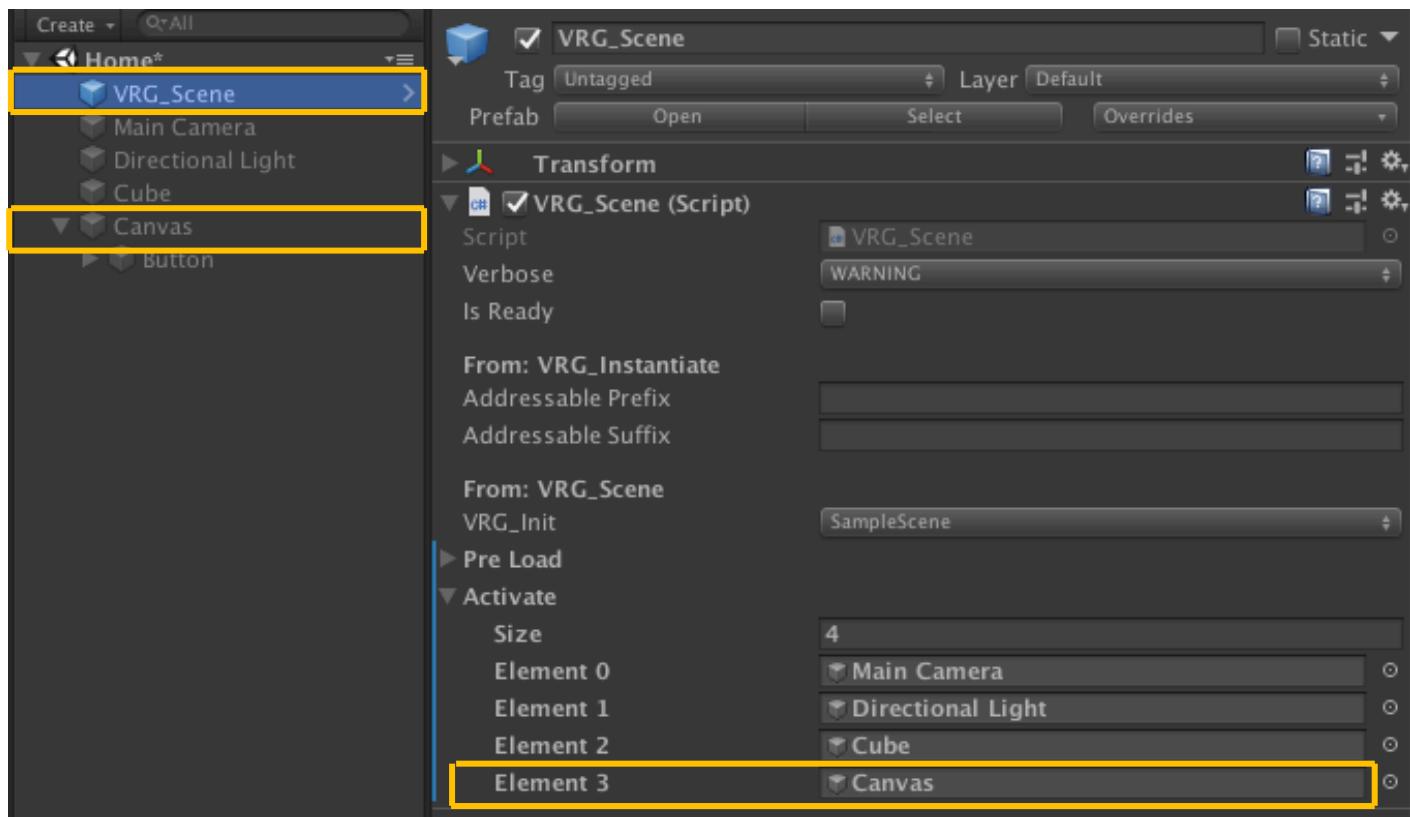
- 3) Add to the Hierarchy a 3dObject -> Sphere, so you know something changed from the previous scene.
- 4) Add the "VRG_Scene" prefab, this prefab **IS ABSOLUTELY REQUIRED** to communicate with the VRG_Core
- 5) That's is with this new scene, let's configure the home to load it with a button click.



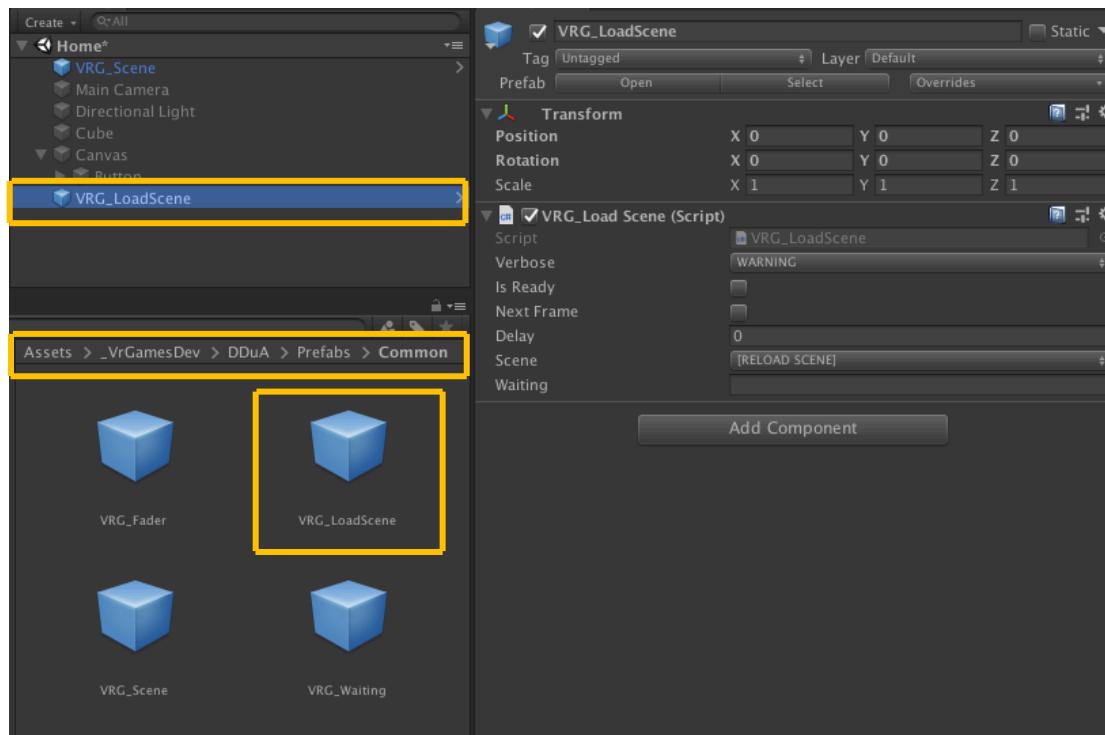
- 6) In your Home scene (created in [7.1 Load your first scene](#) section), add a [UI Button](#) and customize the size and text to your taste.



- 7) Add your new Canvas->button to the list of the VRG_Scene Activate properties, and deactivate the canvas object, it will be colored gray. (If you don't the UI elements are drawn as a camera overlay, and it will be visible while loading the scene)

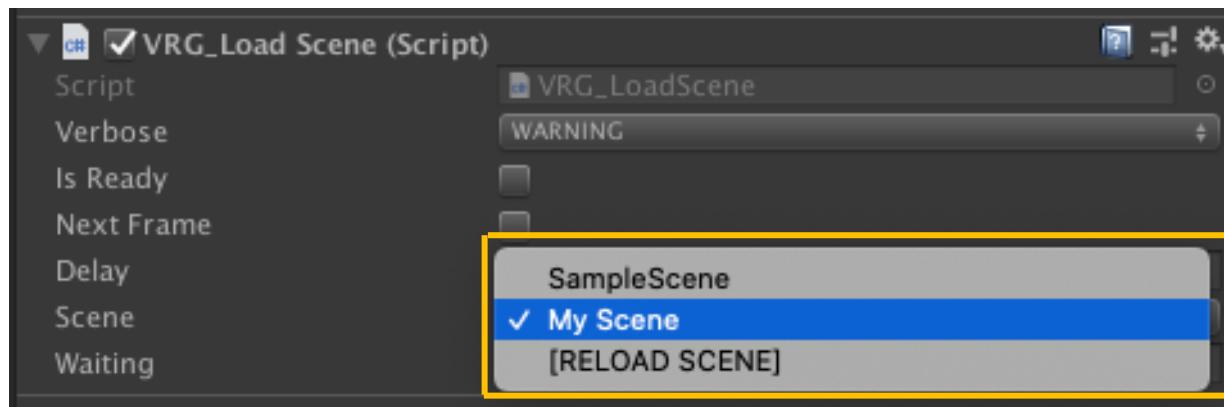


8) Add a **VRG_LoadScene** Prefab, located in (Assets/_VrGamesDev/DDuA/Prefabs/Common)

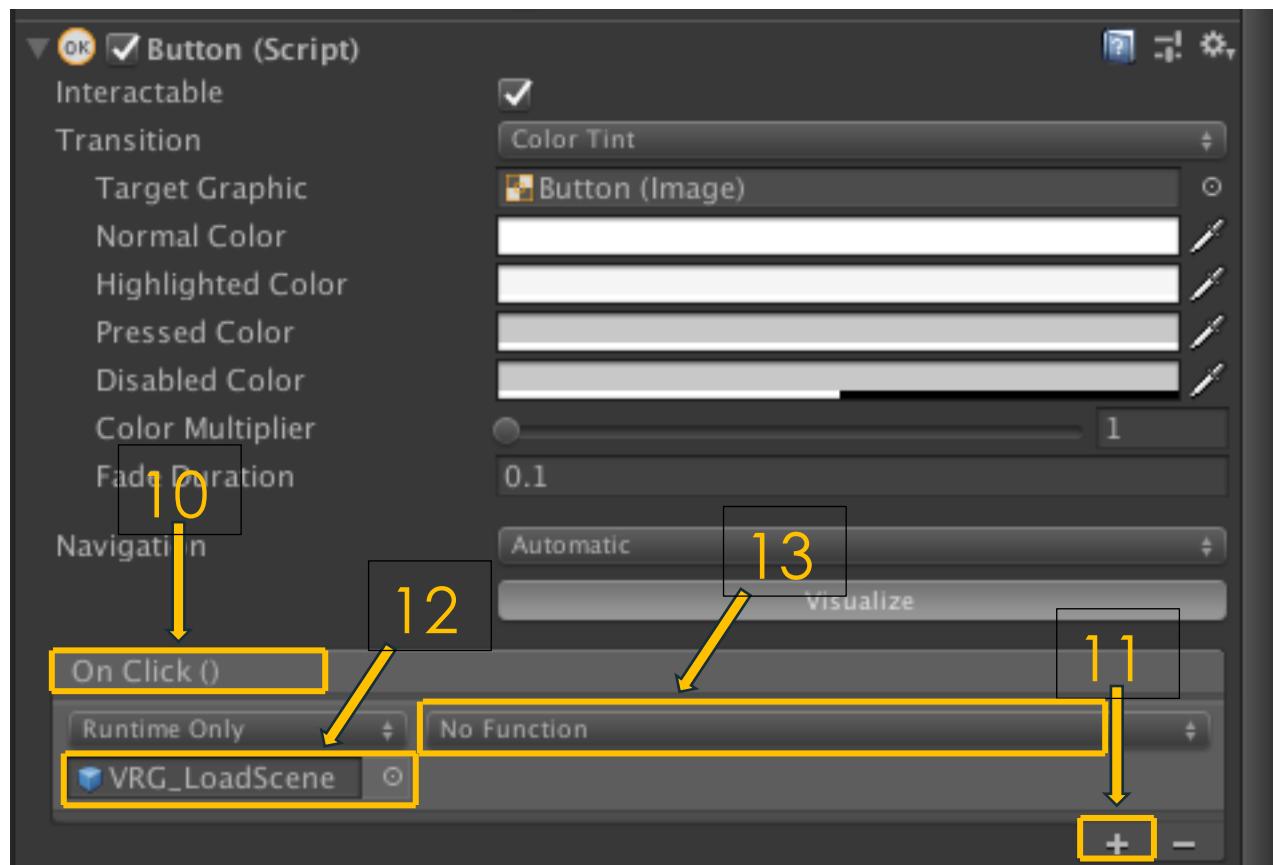




- 9) Select the **VRG_LoadScene** object and open the drop-down menu, select "My Scene", you will get the list of the scenes added to the build list (actives or not).

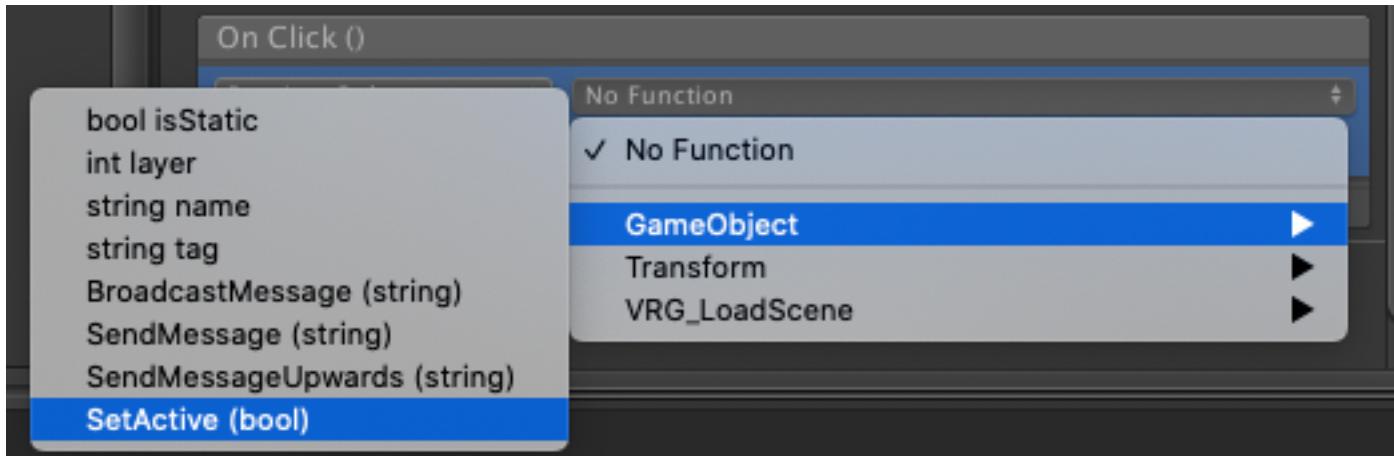


- 10) Deactivate the **VRG_LoadScene** prefab, (when it's active, it will load the selected scene), let's activate it when you click the button created.
 11) Select the UI button element, add an action in the "On Click () Section"
 12) Click the "+" button
 13) Drag and drop the **VRG_LoadScene** prefab to the box where it says "None (Object)"
 14) Click the dropdown "No function"

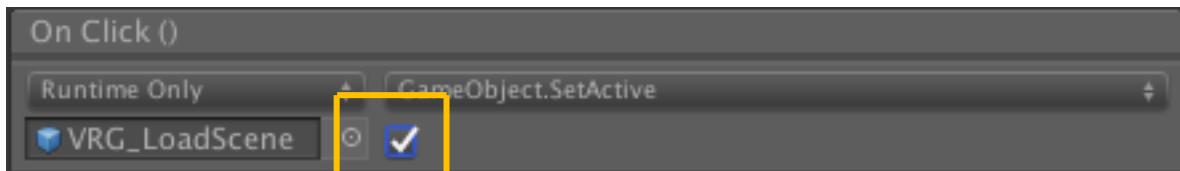




15) Select the gameObject option, and a submenu will open, select “**SetActive (bool)**” option



16) Now toggle the value true. You can learn more about UI and buttons [here](#)



17) Save the scenes and play the game.



Congratulations!



8.1 Here comes a new challenger (Boss)

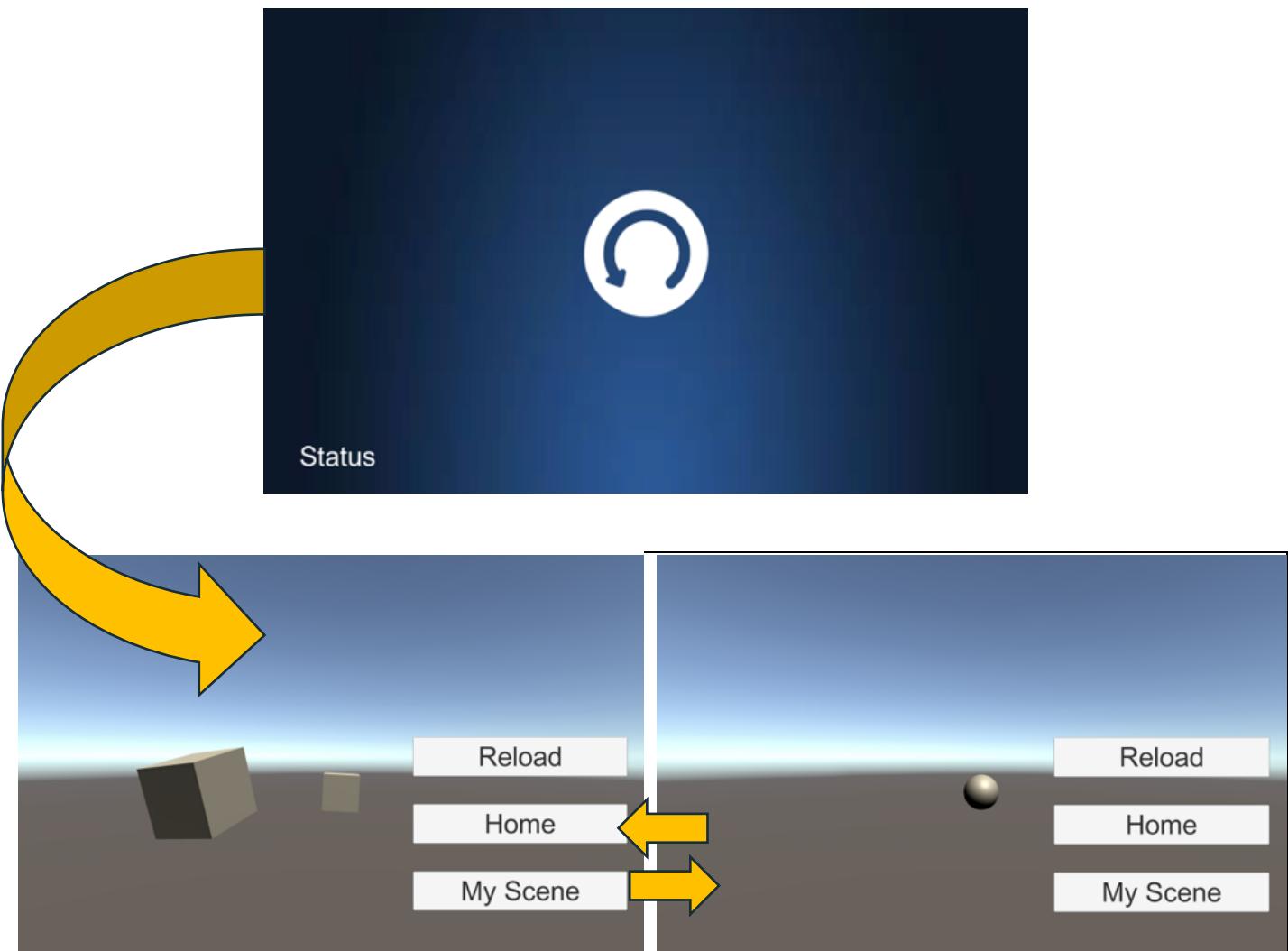
Challenger: Make the Canvas -> Button from "Home" into an addressable and load it

Challenger: Make the Sphere from "My Scene" into an addressable and load it

Challenger: Create a 3 button UI, one button to reload and the other to load the other scene, so you can navigate between them.

8.2 Boss – Cheat code

The previous challengers are solved in the Example "01 Tutorial"





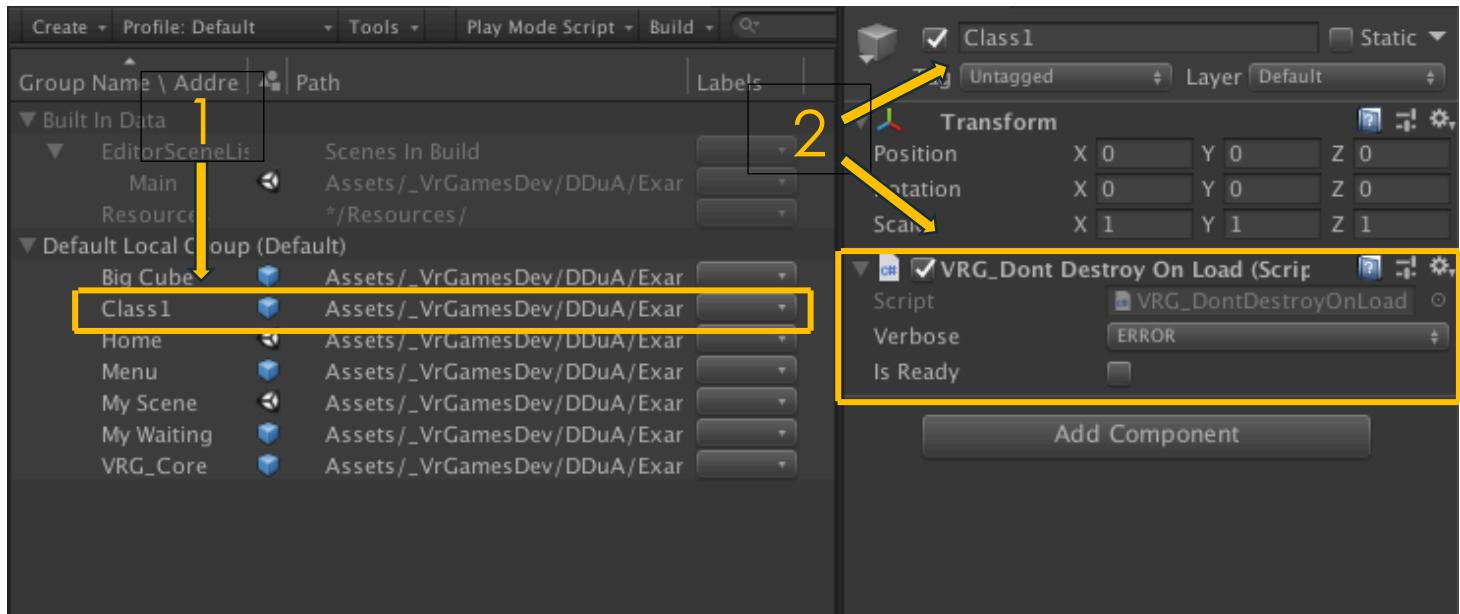
9. SINGLETONS AND DOWNLOADS

The last subject of this package is how to Instantiate and download assets that do not belong to any scene (or all of them) and you need them before any scene is loaded.

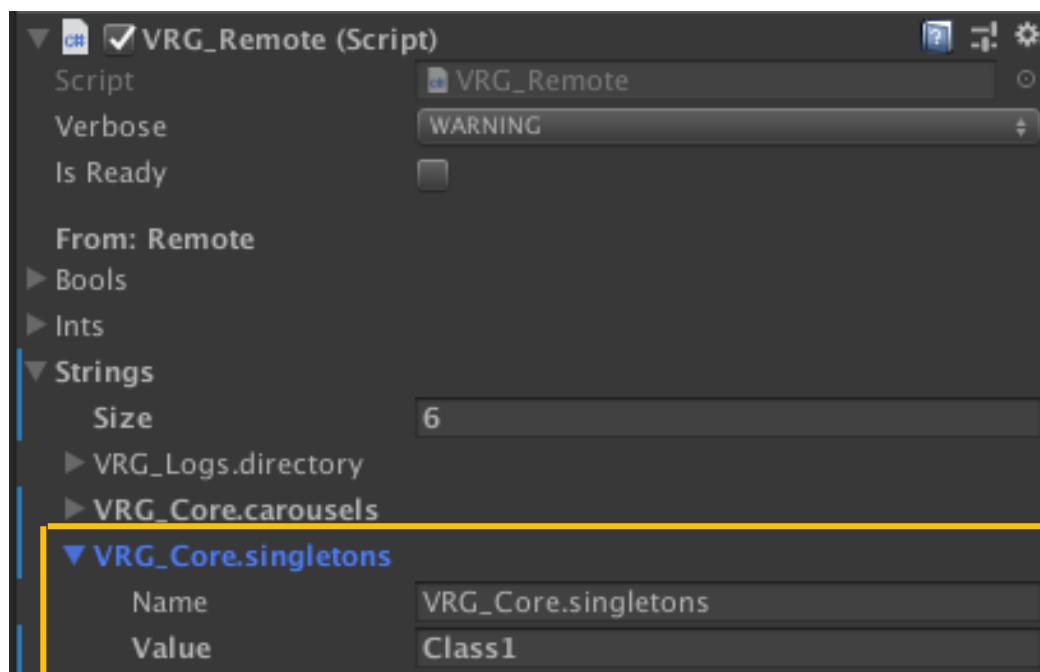
This is useful if you are using [game design patterns](#) like [singletons](#), [Pooling](#) or Game managers that you need them ready available.

9.1 Singletons in VRG_Remote

- 1) Create a prefab, name it "Class1", make it addressable and simplify its name.
- 2) Add the component "VRG_DontDestroyOnLoad" to the Class1 prefab

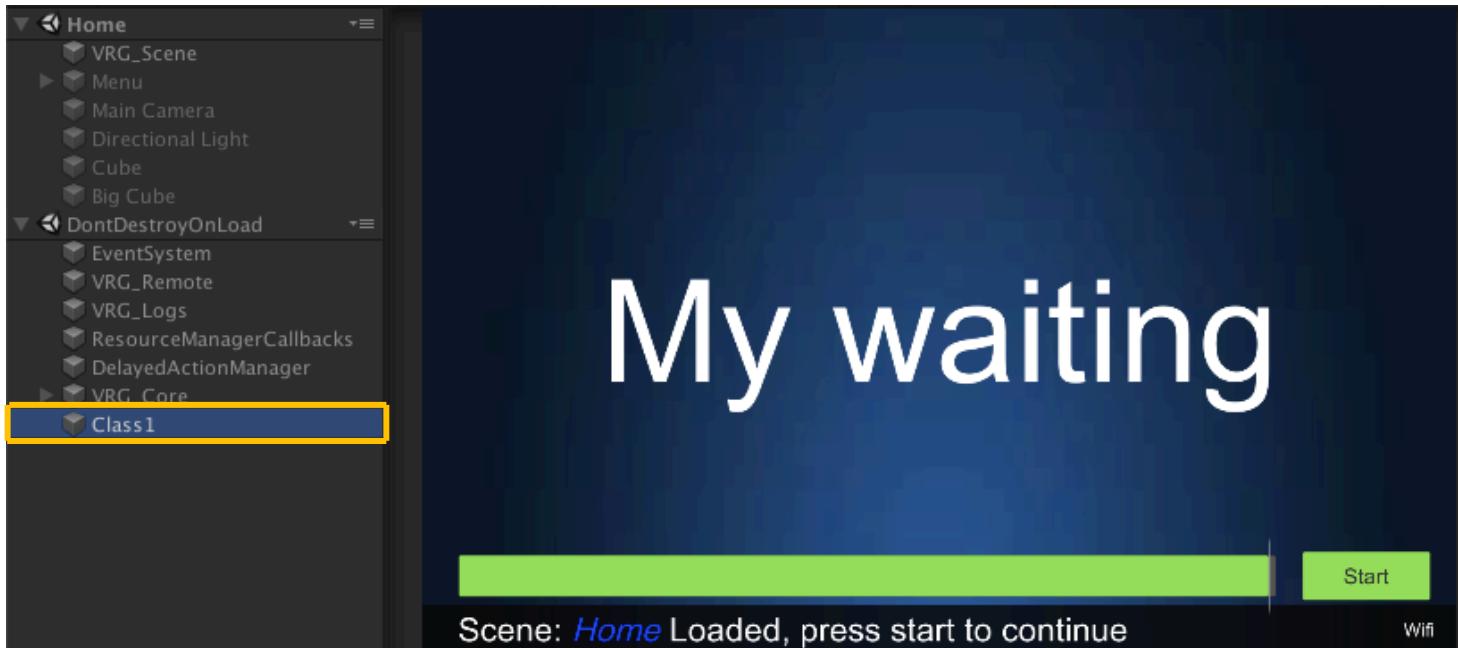


- 3) Add the address "Class1" of this new prefab to the **Strings -> VRG_Remote -> VRG_Core.singletons**, remember that you need to separate every item with the " | " pipe character.





- 4) Save the scene and play the game, notice the prefab Class1 was loaded before the scene.



- 5) Check the logs, to further understand the process of loading and instantiating of the prefabs from VRG_Core.
- It reads from VRG_Remote how many singletons will try to instantiate
 - Retrieve the size of the asset
 - Find it in cache (or download it from a server if not in cache)
 - Finished installing and trying to instantiate
 - Asset Loaded, and installed

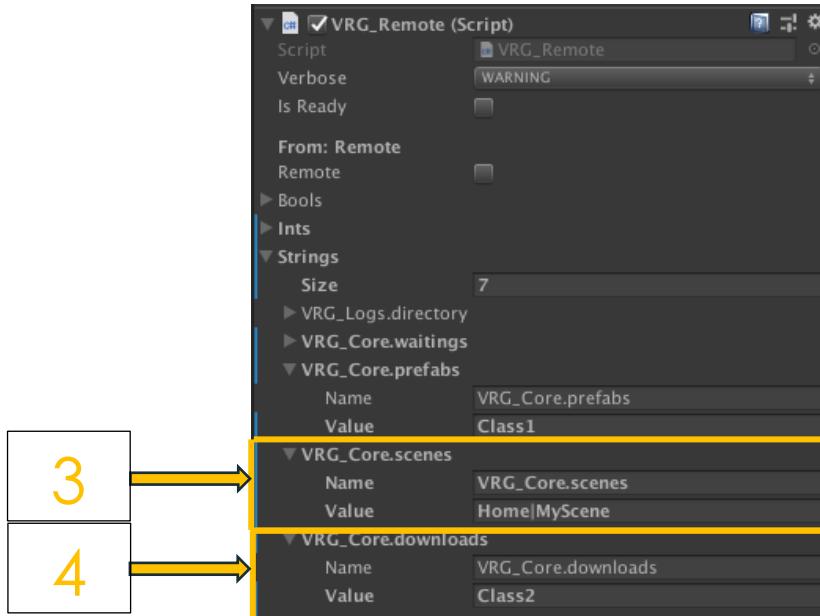
8	STATUS	4.831039	VRG_Core->RemoteSettings_Remote(289)	(1) Carousels: - Wait_Minigame
9	STATUS	4.831039	VRG_Core->RemoteSettings_Remote(339)	(1) Singletons: - Class1 a)
10	STATUS	4.831039	VRG_Core->RemoteSettings_Remote(388)	(1) Scenes: - Home
11	STATUS	4.831039	VRG_Core->RemoteSettings_Remote(459)	(2) Downloads: - Class2 - Big Cube
12	DEBUG	4.831039	VRG_Core->CreateCarousel(672)	Creating Wait_Minigame
13	DEBUG	4.831039	VRG_Core->GetSize(800)	SIZING: Class1
14	INFO	4.831039	VRG_Core->GetAddressable(859)	CACHE: Class1
15	DEBUG	4.911039	VRG_Core->SetProgress(1120)	100.0% - download complete
16	STATUS	4.911039	VRG_Instantiate->InstantiateAll(97)	1.- Class1 (0) Trying to install 1 assets
17	DEBUG	4.911039	VRG_Core->SetProgress(1120)	0.0% - 0.0%
18	STATUS	4.911039	VRG_Instantiate->InstantiateAll(133)	Installing 0 Class1 = 50.00%
19	DEBUG	4.911039	VRG_Core->SetProgress(1120)	50.0% - 50.0%
20	INFO	5.011039	VRG_Instantiate->GetInstantiate(231)	Class1 Loaded
21	DEBUG	5.011039	VRG_Core->Carousel_Completed(729)	Wait_Minigame created successfully
22	DEBUG	5.031039	VRG_Core->SetProgress(1120)	100.0% - download complete
23	INFO	5.031039	VRG_Instantiate->InstantiateAll(178)	1 assets installed e)



9.2 Downloads in VRG_Remote

This is useful to preload elements you know you will need later, these elements are downloaded in the idle time of the game, so you can download a heavy game in the background, VRG_Core, also downloads the prefabs needed in the scenes configured in **VRG_Remote->VRG_Core.Scenes**

- 1) Create a prefab, name it "Class2", make it addressable and simplify its name.
- 2) Add the component "VRG_DontDestroyOnLoad" to the "Class2" prefab
- 3) Add the scenes needed in the game, the "**Home**" Scene and the "**My Scene**" scene, remember that you need to separate every item with the "|" character
- 4) Add the address "Class2" of this new prefab to the **VRG_Remote -> VRG_Core.downloads**



- 5) Play the game. Notice that nothing visible happens, but check the logs to know your downloads are ready. There are 3 downloads ready, the one you defined "**Class2**" and the prefabs used in the scenes, "**Big cube**" and "**Menu**", the VRG_Core will try to download them silently and while the network connection is Idle.

11	STATUS	4.367114	VRG_Core->RemoteSettings_Remote(455)	(3) Downloads: - Class2 - Big Cube - Menu
45	STATUS	9.034837	VRG_Core->CheckScene(592)	Scene: Home Activating ...
46	DEBUG	10.034840	VRG_Core->DestroyWaiting(688)	Destroying My Waiting
47	DEBUG	10.034840	VRG_Core->PreloadAssets(1603)	PRELOAD: Class2 (0 / 3)
48	DEBUG	10.034840	VRG_Core->GetSize(796)	SIZING: Class2
49	LOGS	10.034840	VRG_Scene->OnSceneActivated(168)	Scene: Home activated
50	INFO	10.114840	VRG_Core->GetAddressable(855)	CACHE: Class2
51	DEBUG	11.674840	VRG_Core->PreloadAssets(1603)	PRELOAD: Big Cube (1 / 3)
52	DEBUG	11.674840	VRG_Core->GetSize(796)	SIZING: Big Cube
53	INFO	11.694840	VRG_Core->GetAddressable(855)	CACHE: Big Cube
54	DEBUG	13.254840	VRG_Core->PreloadAssets(1603)	PRELOAD: Menu (2 / 3)
55	DEBUG	13.254840	VRG_Core->GetSize(796)	SIZING: Menu
56	INFO	13.274840	VRG_Core->GetAddressable(855)	CACHE: Menu
57	LOGS	14.814840	VRG_Core->PreloadAssets(1619)	PRELOAD FINISHED

10. USING UNITY REMOTE CONFIG INSTEAD OF LOCAL VRG_REMOTE

Remote config is a cloud service that lets you tune and customize your app over the air without requiring an update to your game. You can use rules to enable or disable features, change the difficulty of your game or run special events to target specific audiences. Unity manages delivery of your game content with minimal performance impact.

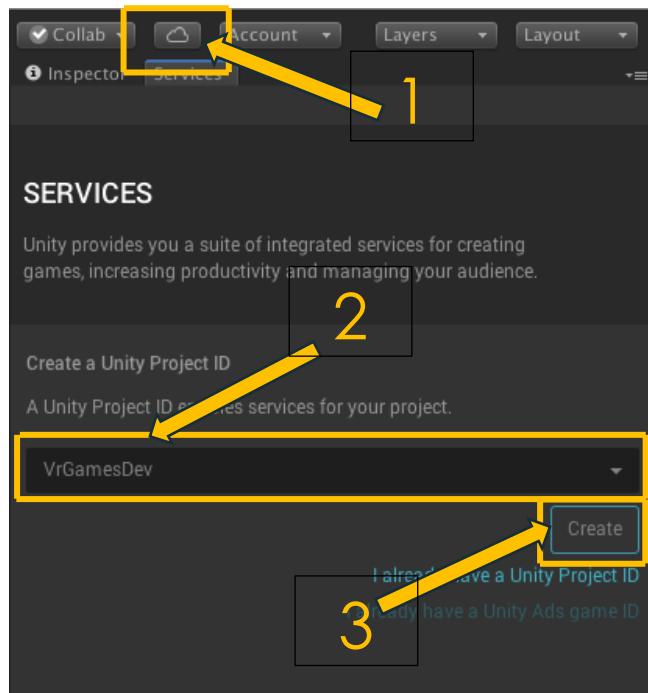
The **VRG_Remote** class variables overwrites the cloud service, it is useful to play and test locally, in this tutorial we used the local settings, I will explain how to use the remote cloud service configuration.

10.1 Review the log generated with the local configuration

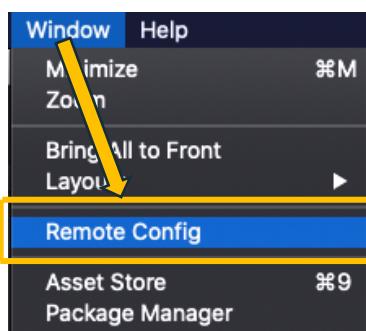
Open the log in the folder (*LogsLocal*), read it, and rename it as “Full Verbose”, we will modify it from the remote cloud server setting, let's configure it

10.2 Configure the Remote Config

- 1) Now create a profile for your project, open the Unity Services, click the tiny cloud icon,
- 2) Select your company profile
- 3) Click the “Create” button

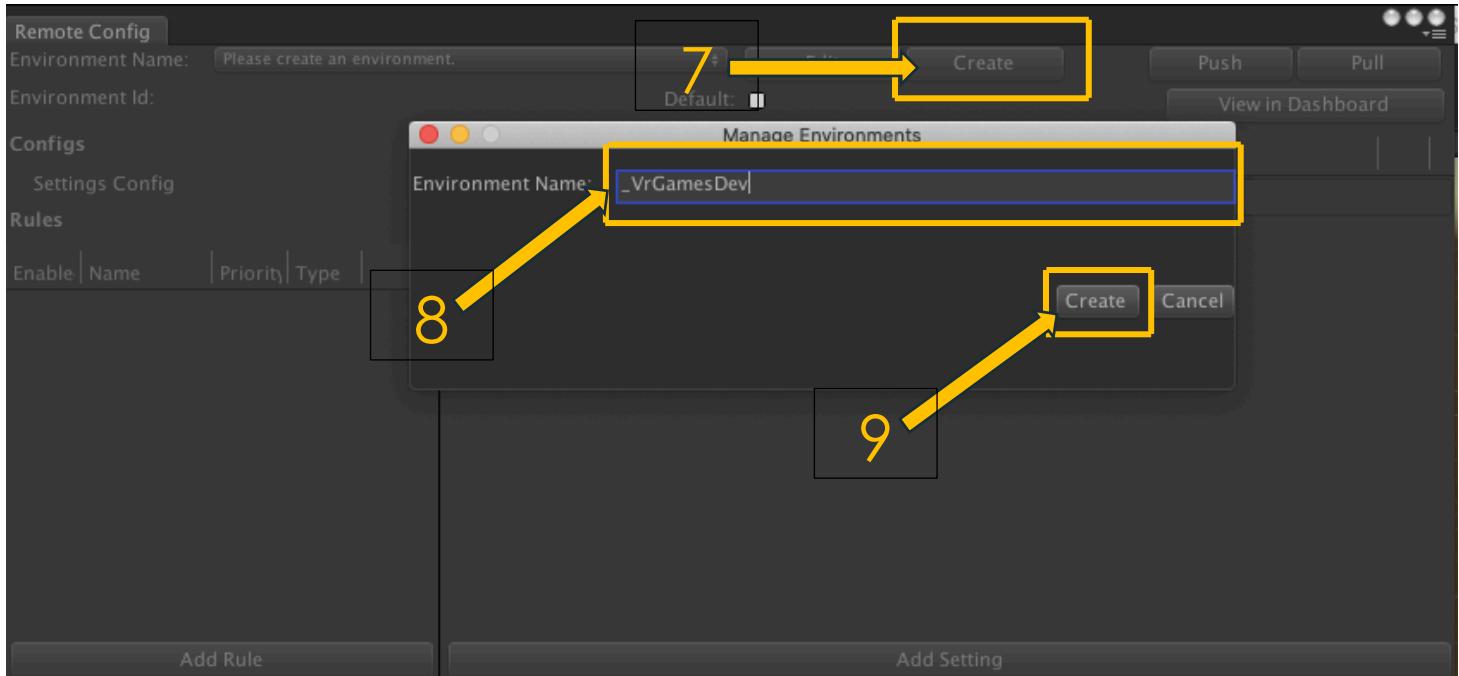


- 4) Be sure you Installed the Remote Config package as explained in [3.2 Unity Remote Config](#) section, you can read more about it [here](#).
- 5) Go to the Window -> Remote Config

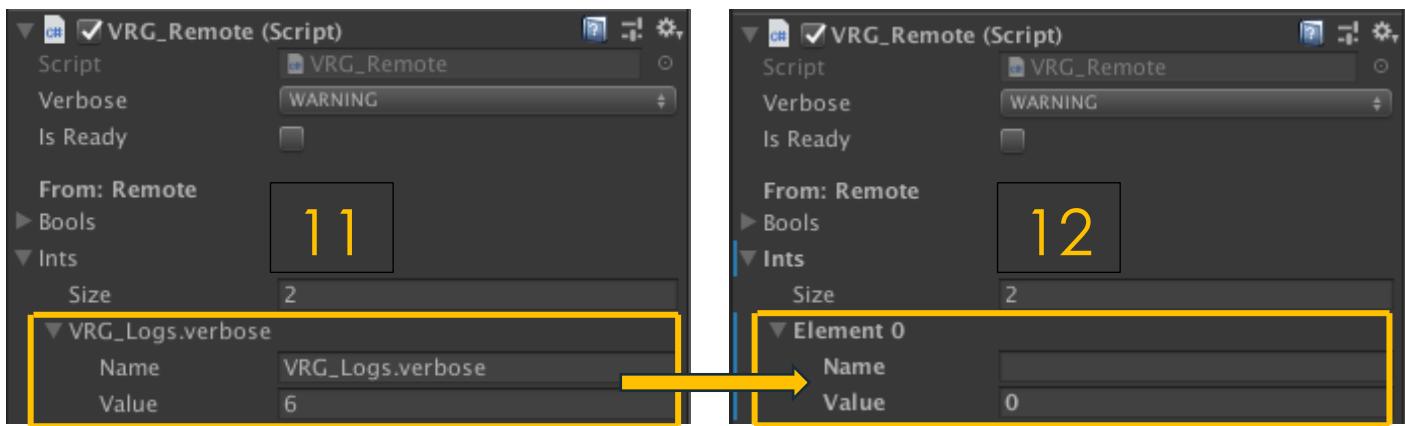




- 6) To get started, create an environment and give it a name. Note environment names are immutable. The first environment created will be set to the default environment,
- 7) Click the “Create” button
- 8) In the text field ... name your environment, something cute, nice, creative, useful, handsome and perfect, like “_VrGamesDev”
- 9) Click the second “Create” button



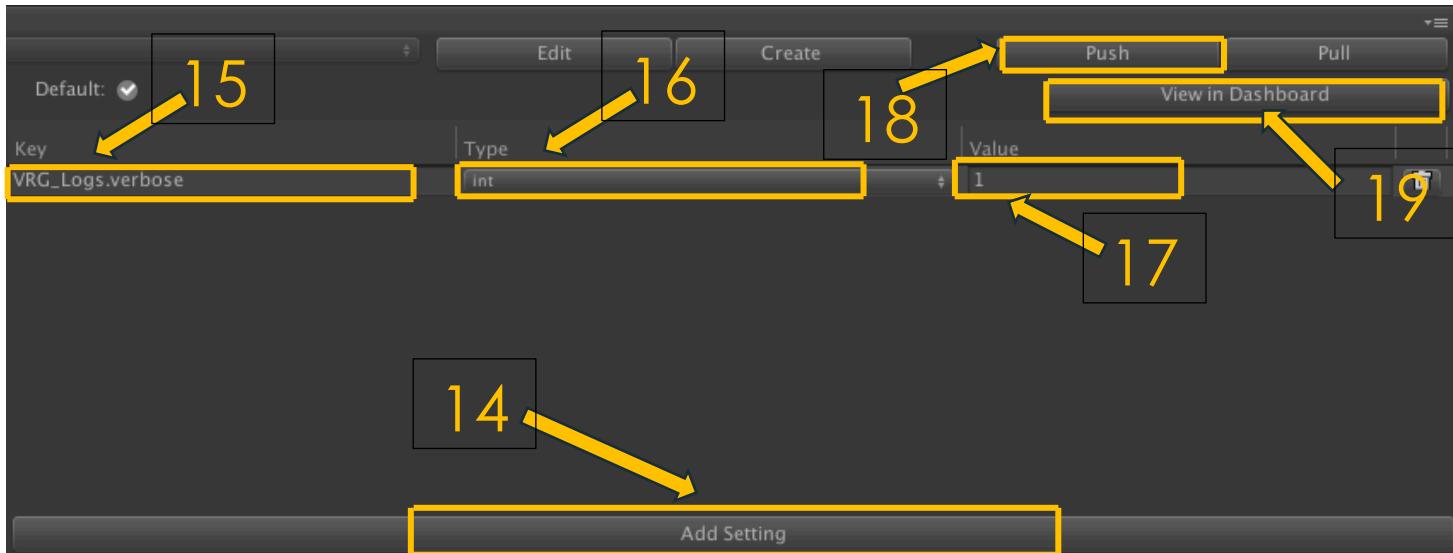
- 10) Now let's create a remote setting, take by example, the log verbose level of detail as explained in [6.1 Configure your VRG_Logs](#) section.
- 11) The setting **VRG_Logs.verbose** is set to 6 in the local VRG_Remote, which is the most detailed, let's change it to the less detailed, the level 1.
- 12) Delete it, remember that it overwrites the remote setting for easier and faster iteration cycles.



- 13) Play the game, and notice how the log wasn't created. Because the setting **VRG_Logs.verbose** was not found so the default value is 0, which means “Verbose empty”
- 14) To use the remote settings, click the “Add Setting” button
- 15) Name the key the same as you defined it in the **VRG_Remote** prefab, “**VRG_Logs.verbose**”
- 16) Declare the same type as defined, in this case, the type is **Int**.
- 17) Now set the desired value, in this case, the value is 1, the less verbose setting.



18) Finally Push it to the server



19) You can see this same setting in the server, click the “View in Dashboard” button.

Key	Type	Value	+ ADD KEY
VRG_Logs.verbose	Int	1	-

20) Play the game and Enjoy.

21) The new log file is configured according to the cloud remote server, with the less verbose configuration, it just logs errors, since we have no error right now, the log is empty.

2000-01-01 00:00:01 - VERBOSITY LEVEL: ERROR				
ID	Verbose	TimeStamp	Class	Message



10.3 Here comes a new challenger.

Challenge: Migrate all the local settings to the cloud Remote Server

Challenge: Play overwriting some of the remote settings with local data.

Remember that the cloud server will be used for your player base, that is why it is important to be able to modify the settings from the editor. When you are ready push the changes to the main server for your player base.

Key	Type	Value
VRG_Logs.verbose	Int	6
VRG_Logs.mode	Int	0
VRG_Logs.directory	String	LogsRemote
VRG_Core.carousels	String	My Waiting
VRG_Core.singletons	String	Class1
VRG_Core.scenes	String	Home My Scene
VRG_Core.downloads	String	Class2
VRG_Scene.Home	String	Big Cube Menu
VRG_Scene.My Scene	String	Menu

Key	Type	Value
VRG_Logs.verbose	int	6
VRG_Logs.mode	int	0
VRG_Logs.directory	string	LogsRemote
VRG_Core.carousels	string	My Waiting
VRG_Core.singletons	string	Class1
VRG_Core.scenes	string	Home My Scene
VRG_Core.downloads	string	Class2
VRG_Scene.Home	string	Big Cube Menu
VRG_Scene.My Scene	string	Menu



11. VRG_ANNOUNCEMENTS AS A GIFT

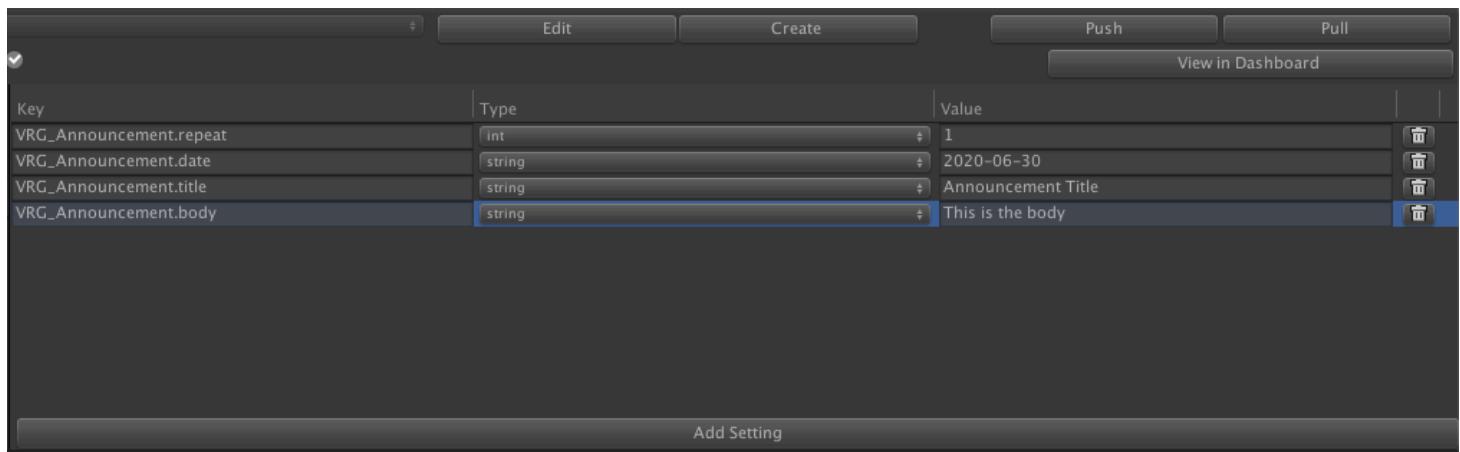
This functionality is useful to inform your players about new changes and patch notes. It will show a popup with date, title and the body of the message, it will show as many times as you want. And then wait for a new combination of date + title + body.

11.1 Configure your VRG_Announcements

You can configure this in the local VRG_Remote, or in the server. Let's configure it in the cloud remote server as explained in [10.2 Configure the Remote Config](#) section.

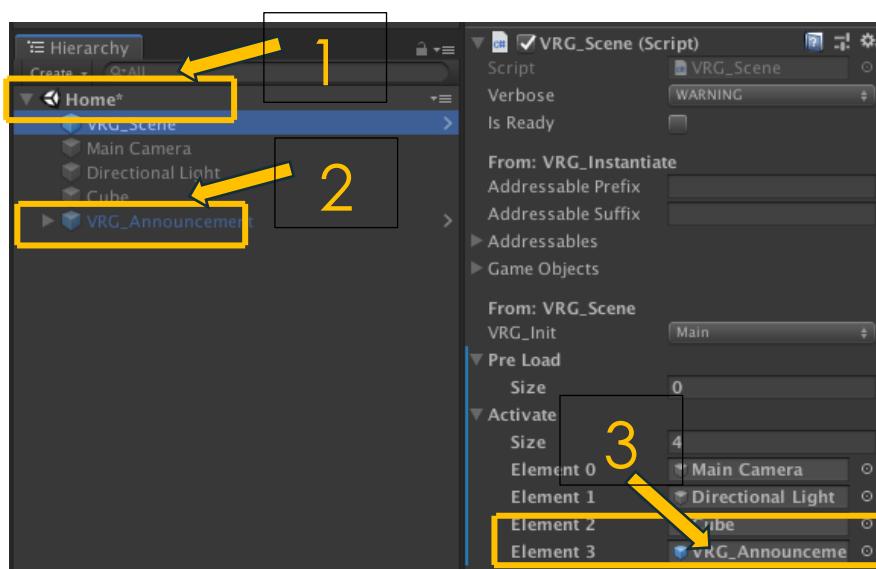
There are 4 settings you can customize **VRG_Announcement.date**, **VRG_Announcement.title**, **VRG_Announcement.body**.

The **VRG_Announcement.repeat**: Is to custom how many times will display, 0 for infinite times.



11.2 Add to your home scene

- 1) Open the Home Scene.
- 2) Add to the hierarchy the **VRG_Announcement** prefab from (*Assets/_VrGamesDev/ Announcement /Prefabs/*)
- 3) Add the **VRG_Announcement** to the VRG_Scene activate array as explained in [8 Change Scenes \(VRG_LoadScene\)](#) section.
- 4) Make sure the prefab is deactivated
- 5) Play the game and enjoy!

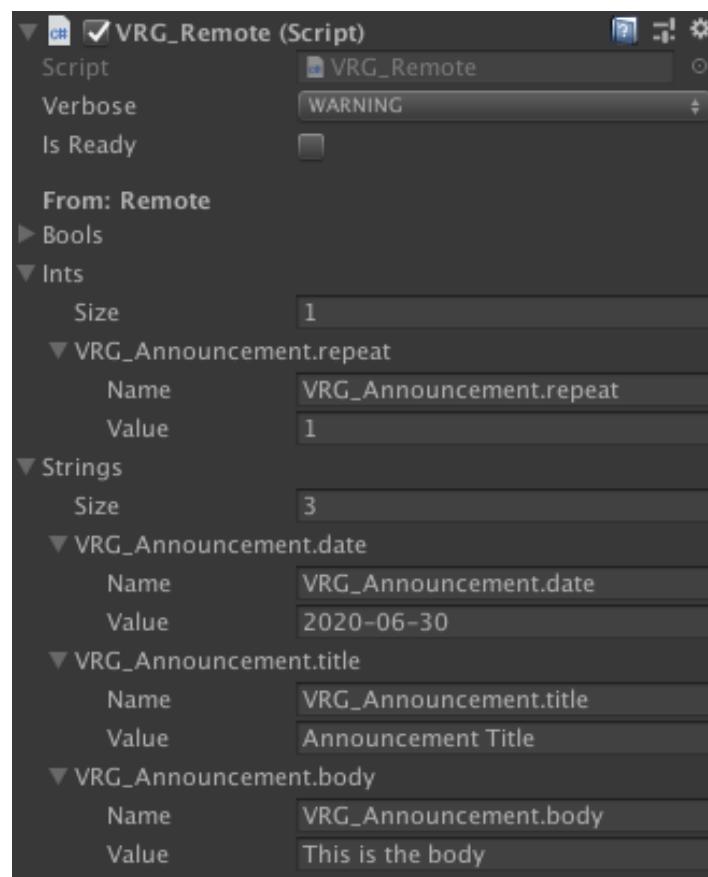
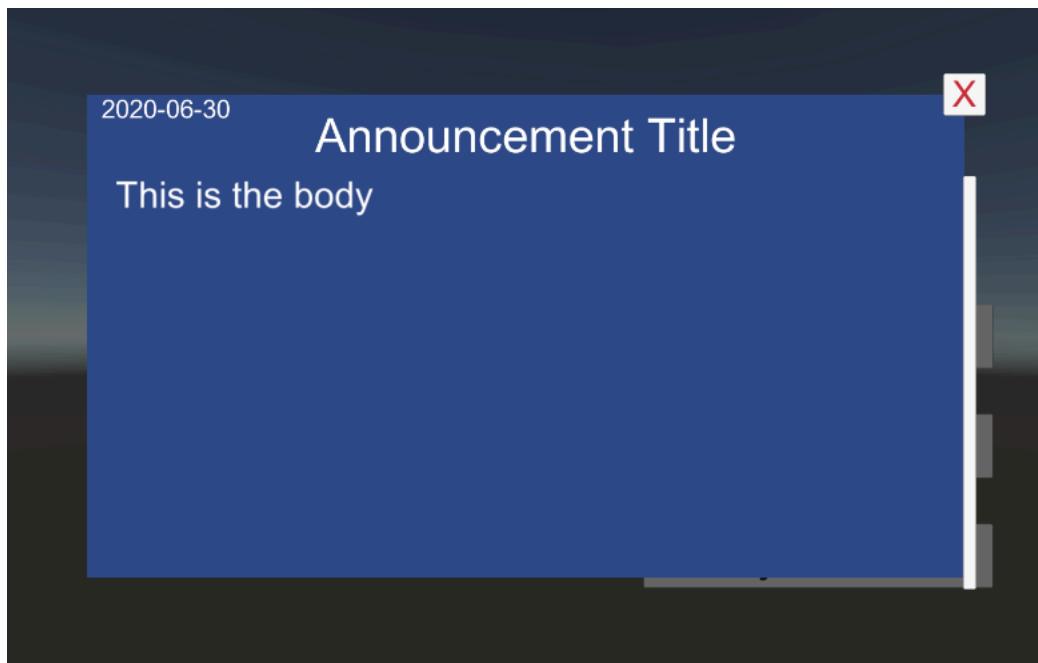




11.3 Here Comes a new challenger

Challenge: Overwrite some of the remote settings with local data.

Challenge: Make the **VRG_Announcement** an addressable and load it in the Home Scene from **VRG_Core**.





12. EXAMPLES

There are 6 examples that show you how to use DDUA as documented in this project.

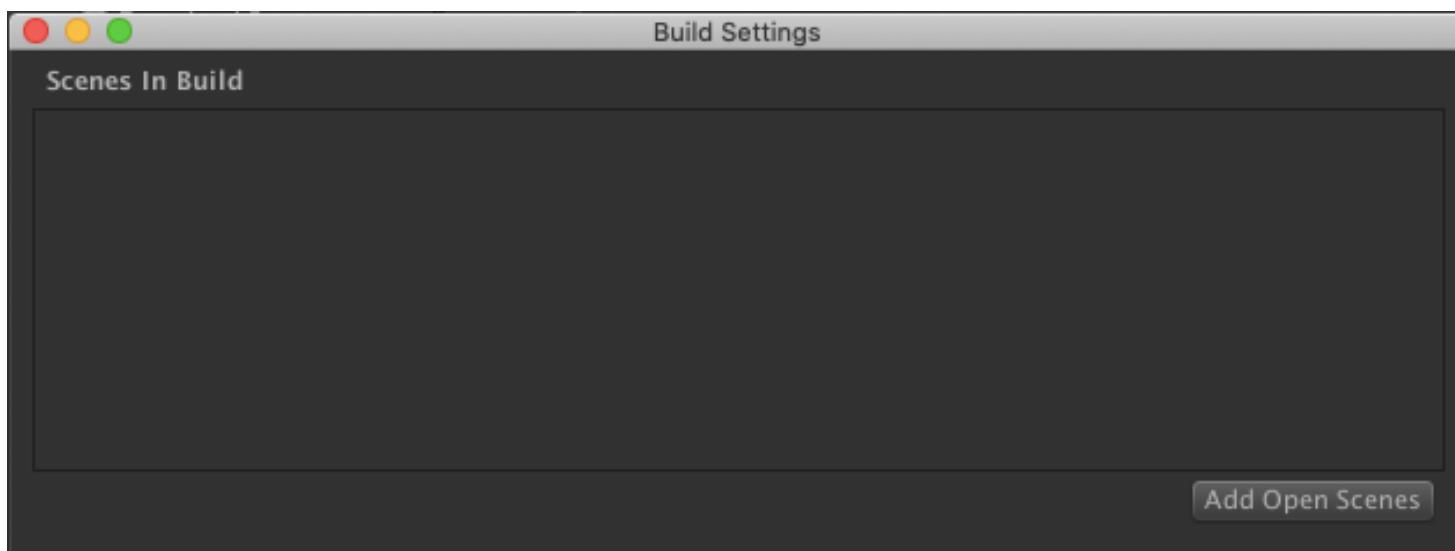
Follow the instructions to run the examples. Thank you for downloading my package.

XoXo...

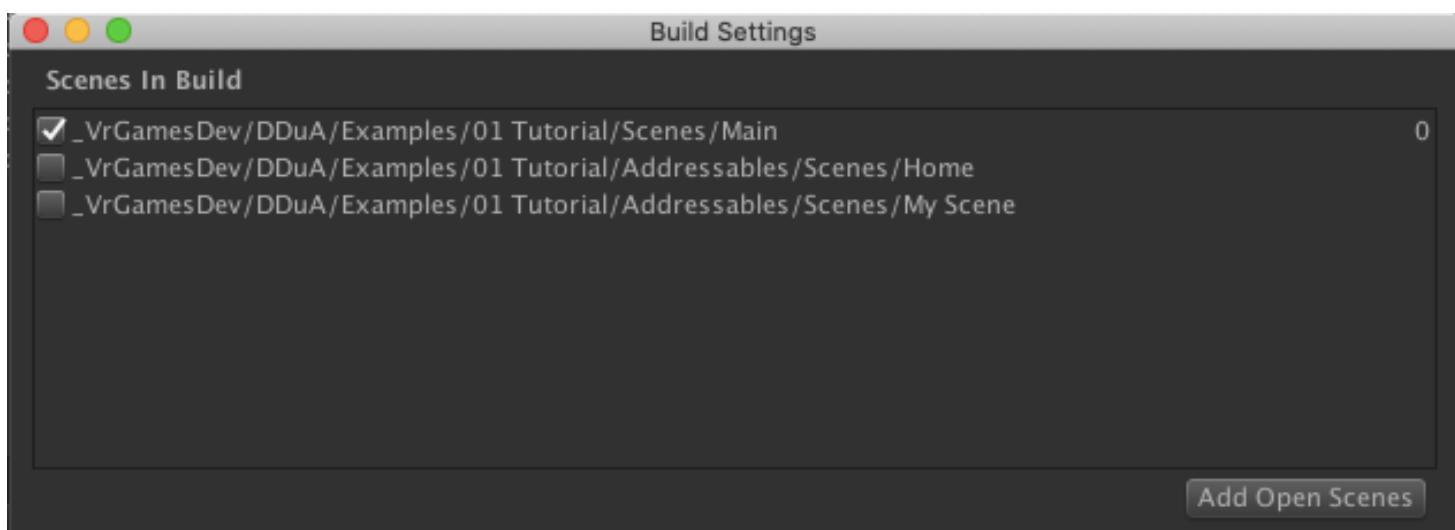
12.1 01 Tutorial

In the folder (*Assets/_VrGamesDev/DDuA/Examples/01 Tutorial*) you will find everything we did in this tutorial, up to the section 11.

- 1) Clean and delete all the previous scenes in your Build settings



- 2) Add to the build settings list the 3 scenes from the example folder, **Main** scene being the first one
 - a) Scenes / Main
 - b) Addressables / Scenes / Home
 - c) Addressables / Scenes / My Scene





- 3) Clean and delete the Addressable Group from previous entries

The screenshot shows the Unity Addressables Manager interface. The main table has three columns: 'Group Name \ Addressable Name', 'Path', and 'Labels'. Under 'Group Name \ Addressable Name', there is a section titled 'Built In Data' containing 'EditorSceneList', 'Main', and 'Resources'. Below this is a section titled 'Default Local Group (Default)' which contains several entries. Under 'Path', it shows 'Scenes In Build' and '*/Resources/'. The 'Labels' column contains two dropdown menus.

- 4) Make all the prefabs and scenes from the folder "Addressables" into name simplified addressables
- Addressables / Big Cube
 - Addressables / Class1
 - Addressables / Class2
 - Addressables / Menu
 - Addressables / My Carousel
 - Addressables / VRG_Core
 - Addressables / Scenes / Home
 - Addressables / Scenes / My Scene

The screenshot shows the Unity Addressables Manager with a context menu open over the 'Default Local Group (Default)' list. The menu options include 'Move Addressables to Group', 'Move Addressables to New Group', 'Remove Addressables', 'Simplify Addressable Names' (which is highlighted in blue), 'Export Addressables', and 'Create New Group'. The 'Simplify Addressable Names' option is selected.

Pro Tip: You can select them all to make them addressables and to simplify them.

The screenshot shows the Unity Addressables Manager after the simplification process. The 'Default Local Group (Default)' list now contains eight entries: 'Big Cube', 'Class1', 'Class2', 'Home', 'Menu', 'My Scene', 'My Waiting', and 'VRG_Core'. Each entry is preceded by a small icon representing its asset type (e.g., cube for Big Cube, script for VRG_Core).

- 5) Play the game and enjoy.

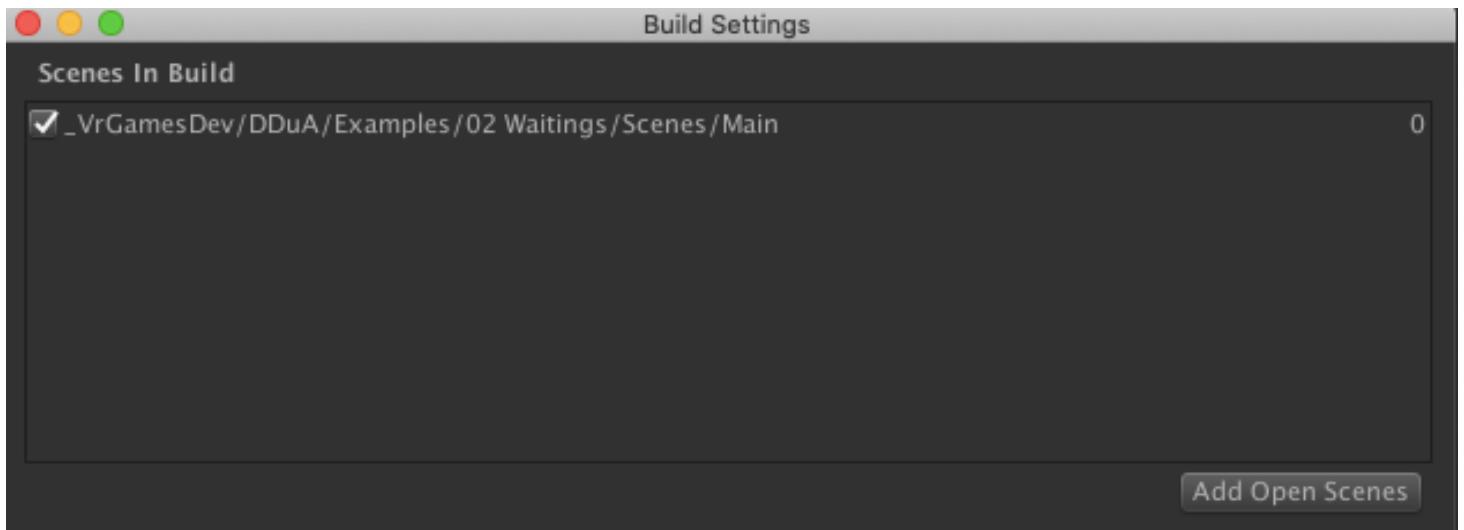


12.2 02 Carousels

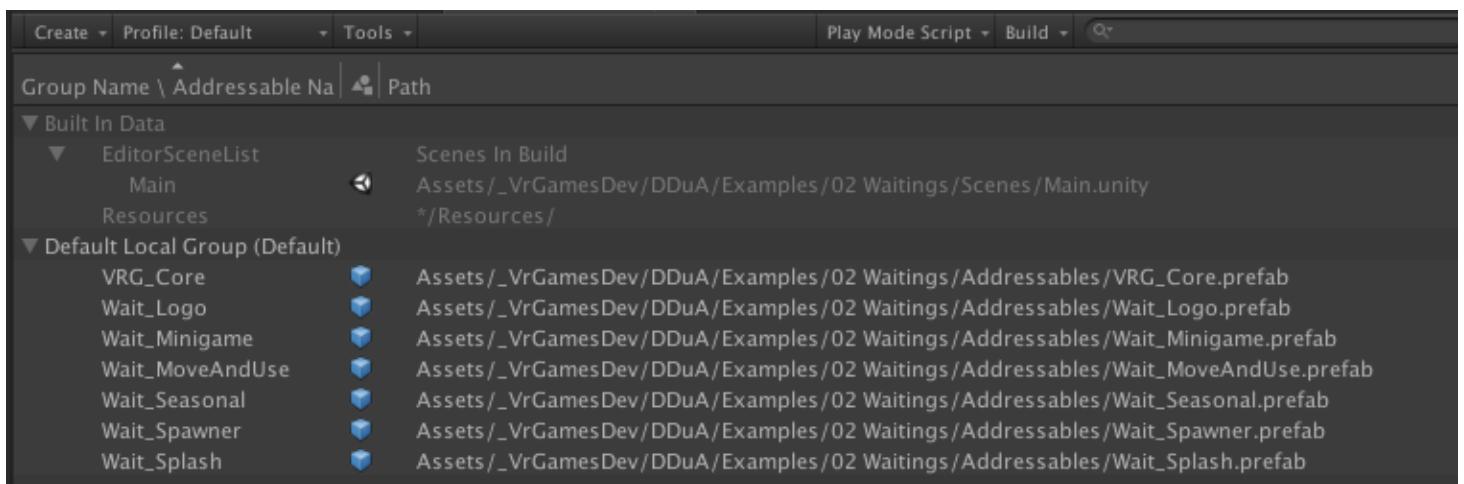
Just sit and relax, and enjoy some Carousels, every 3 seconds it switches to another one, but the mini game, you need to "play" it, just click the colored button.

In the folder (`Assets/_VrGamesDev/DDuA/Examples/02 Carousels`)

- 1) Clean and delete all the previous scenes in your Build settings
- 2) Add to the build settings list the scene from the (`Examples/02 Carousels`) folder, **Main** scene being the first one



- 3) Clean and delete the Addressable Group from previous entries
- 4) Make all the prefabs and scenes from the folder "Addressables" into name simplified addressables
 - a) Addressables / VRG_Core
 - b) Addressables / Wait_Logo
 - c) Addressables / Wait_Minigame
 - d) Addressables / Wait_MoveAndUse
 - e) Addressables / Wait_Seasonal
 - f) Addressables / Wait_Spawner
 - g) Addressables / Wait_Splash



- 5) Play the game and enjoy.

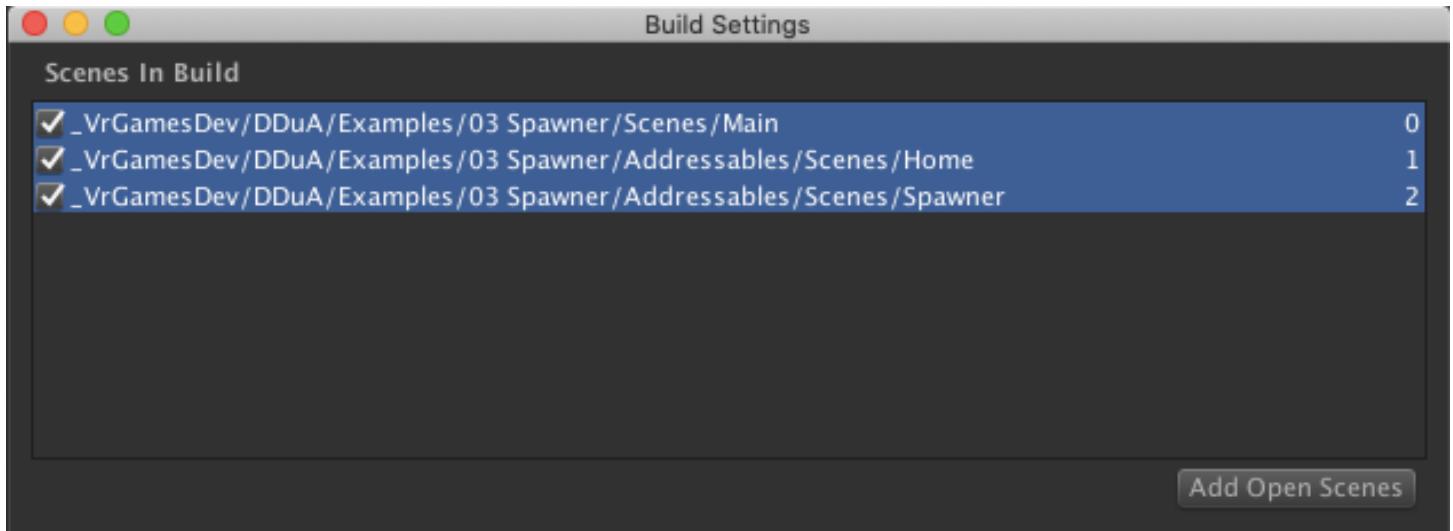


12.3 03 Spawner

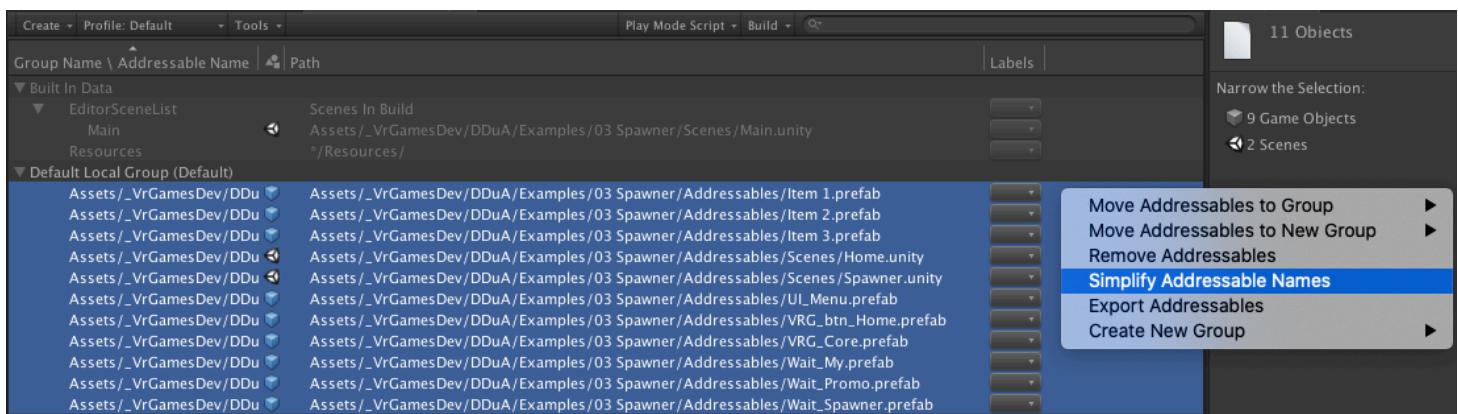
In this example you will learn how to preload some prefabs to use and release them without releasing the whole scene, and how to use a dedicated addressable when transitioning to a scene, plus giving your user an announcement with this information.

This example is in the folder (`Assets/_VrGamesDev/DDuA/Examples/03 Spawner`)

- 1) Clean your PlayerPrefs
- 2) Clean and delete all the previous scenes in your Build settings
- 3) Add to the build settings list the 3 scenes from the example folder, **Main** scene being the first one
 - a) Scenes / Main
 - b) Addressables / Scenes / Home
 - c) Addressables / Scenes / My Scene



- 4) Clean and delete the Addressable Group from previous entries
- 5) Make all the prefabs and scenes from the folder "Addressables" into name simplified addressables
 - a) There are 9 prefabs game Objects
 - b) There are 2 scenes



- 6) Play the game and enjoy.



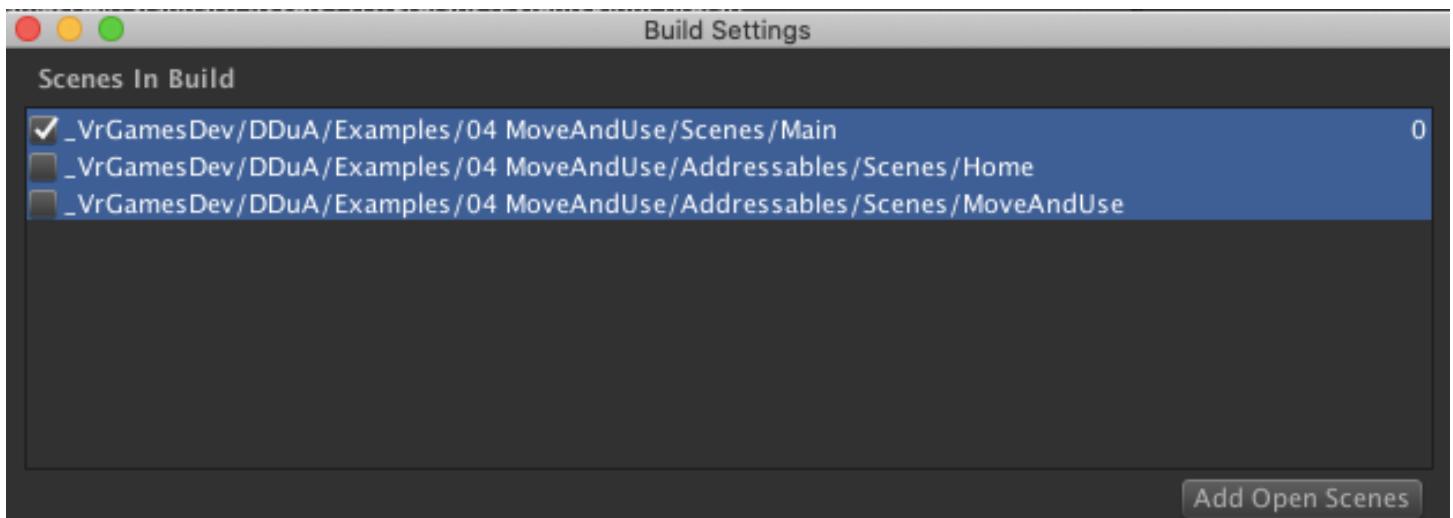
12.4 04 MoveAndUse

In this example you will learn how to load and use some existent prefabs from other packages or your own, we will use some standard assets, access, modify and enhance them to suit them to our own needs to create a playable mini game.

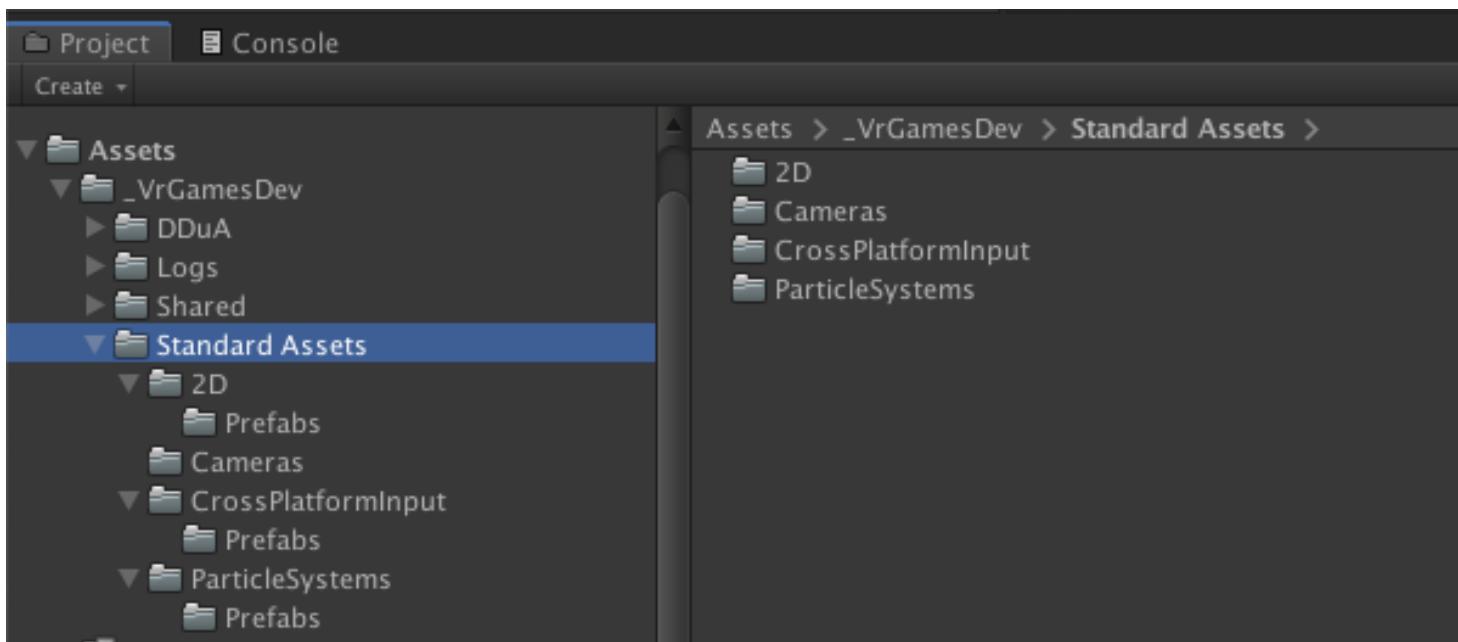
Use WASD and space bar to jump into the boxes to destroy them, get 10 and win the game.

This example is in the asset (`/Assets/_VrGamesDev/DDuA/Examples/04 MoveAndUse`). Unpackage it.

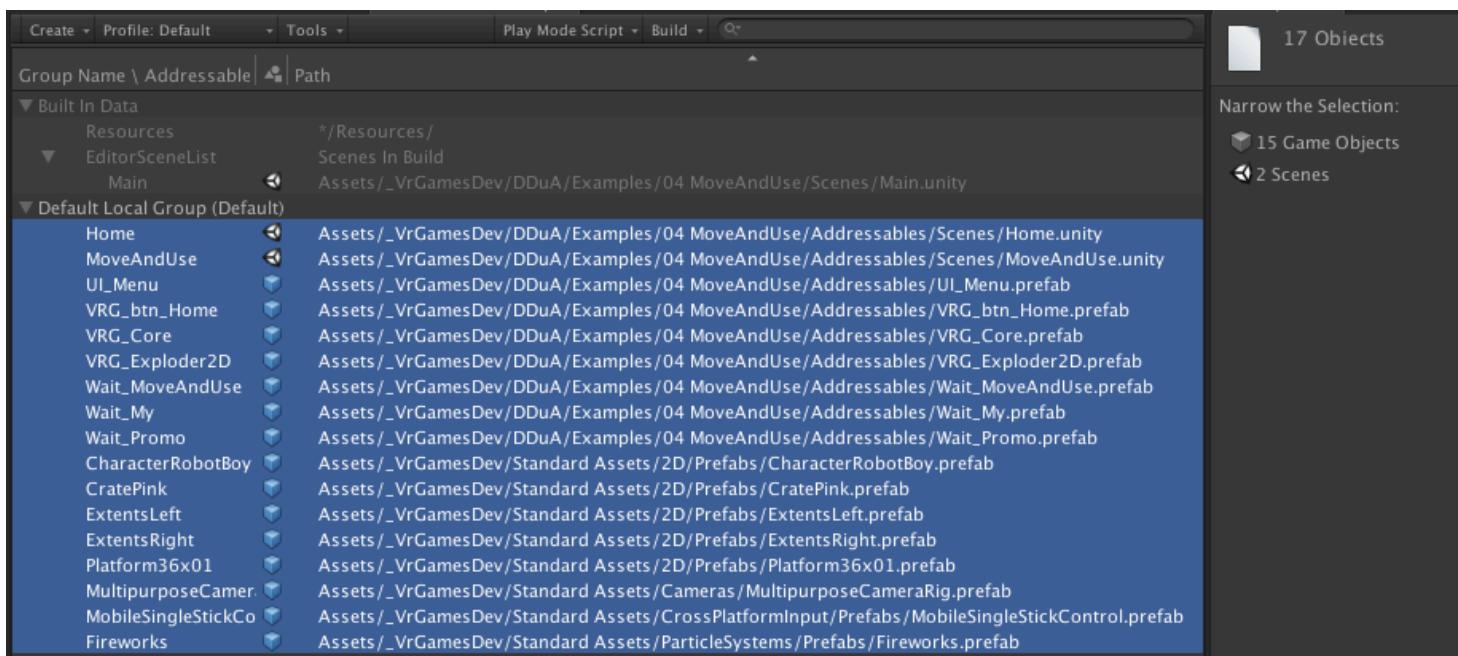
- 1) Clean your PlayerPrefs
- 2) Clean and delete all the previous scenes in your Build settings
- 3) Add to the build settings list the 3 scenes from the example folder, **Main** scene being the first one
 - a) Scenes / Main
 - b) Addressables / Scenes / Home
 - c) Addressables / Scenes / MoveAndUse



- 4) Clean and delete the Addressable Group from previous entries
- 5) Make all the prefabs and scenes from the folder “Addressables” into name simplified addressables
 - a) There are 2 scenes
 - i) 04 MoveAndUse / Addressables / Scenes / Home.unity
 - ii) 04 MoveAndUse / Addressables / Scenes / MoveAndUse.unity
 - b) There are 7 prefabs game Objects
 - i) 04 MoveAndUse / Addressables / UI_Menu
 - ii) 04 MoveAndUse / Addressables / VRG_btn_Home
 - iii) 04 MoveAndUse / Addressables / VRG_Core
 - iv) 04 MoveAndUse / Addressables / VRG_Explorer2D
 - v) 04 MoveAndUse / Addressables / Wait_MoveAndUse
 - vi) 04 MoveAndUse / Addressables / Wait_My
 - vii) 04 MoveAndUse / Addressables / Wait_Promo
- 6) Make all the prefabs from the folder (`/Assets/_VrGamesDev/Standard Assets`) into name simplified addressables
 - a) There are 4 folders that need to have their prefabs addressable
 - i) 2D
 - ii) Cameras
 - iii) CrossPlatformInput
 - iv) ParticleSystems



- 7) You need to make 8 prefabs into name simplified addressables
- Standard Assets / 2D / Prefabs / CharacterRobotBoy
 - Standard Assets / 2D / Prefabs / CratePink
 - Standard Assets / 2D / Prefabs / ExtentsLeft
 - Standard Assets / 2D / Prefabs / ExtentsRight
 - Standard Assets / 2D / Prefabs / Platform36x01
 - Standard Assets / Cameras / MultipurposeCameraRig
 - Standard Assets / CrossPlatformInput / Prefabs / MobileSingleStickControl
 - Standard Assets / ParticleSystems / Prefabs / Fireworks



- 8) Play the game and enjoy



12.5 05 UnityCar

In this example you will learn how to load any scene, we will load a Unity default scene from Standard assets, the car scene.

Use WASD to move the car. This example is in the folder (Assets/_VrGamesDev/DDuA/Examples/05 UnityCar)

- 1) Clean your PlayerPrefs
- 2) Make sure you downloaded "Standard Assets" as explained in [3.3 Standard Assets](#) section

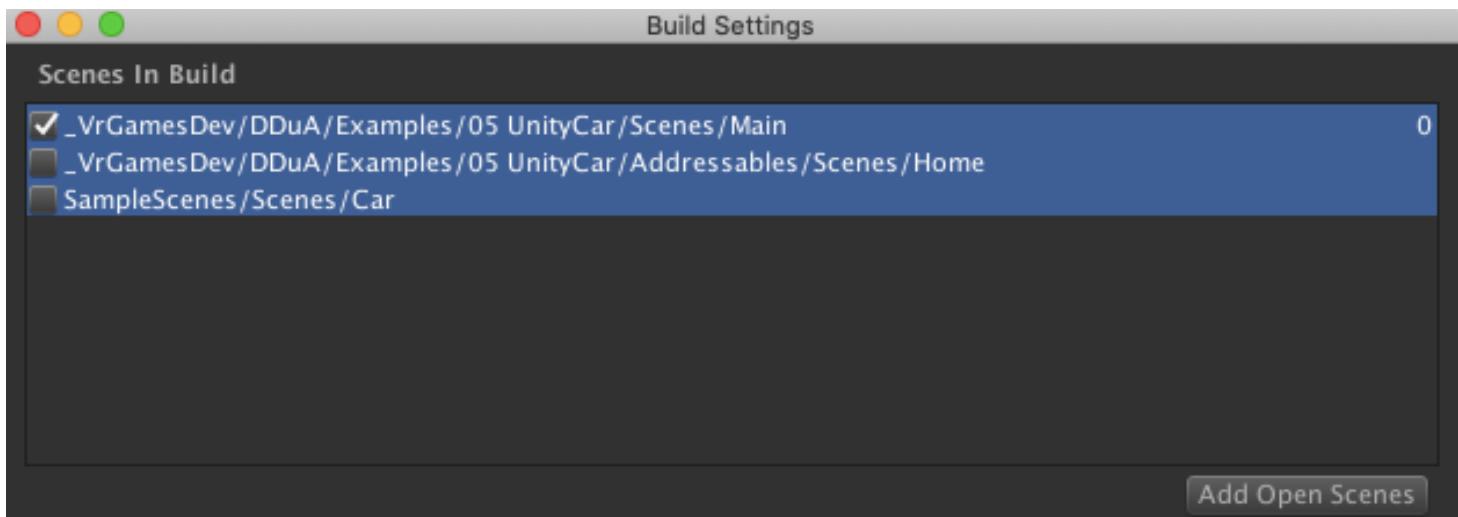
Version: 1.1.5 • Mar 5, 2018

Added warning about Environment assets not working in VR.

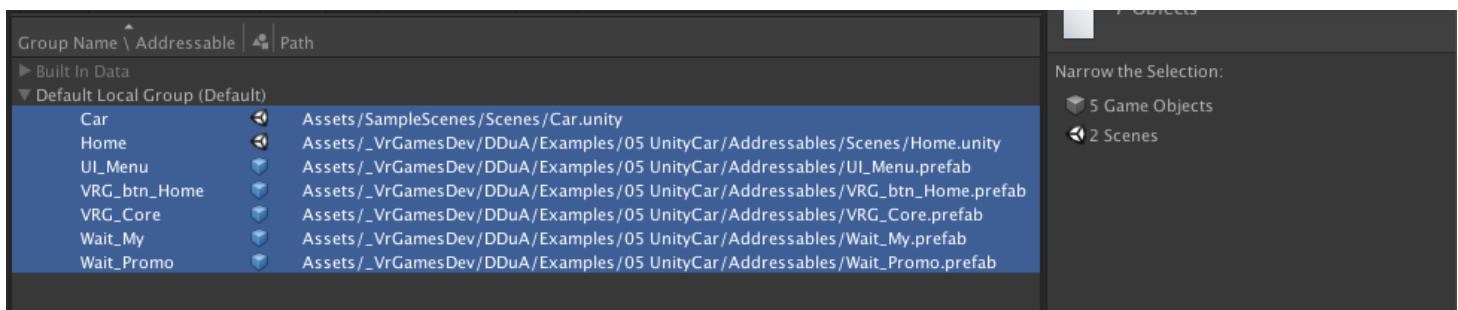
[+ Add label](#) [Hide asset](#)

- 3) Clean and delete all the previous scenes in your Build settings
- 4) Add to the build settings list the 2 scenes from the example folder, **Main** scene being the first one
 - a) Scenes / Main
 - b) Addressables / Scenes / Home
- 5) Add to the build settings list the scene from the Assets/SampleScenes/Scenes/Car.unity

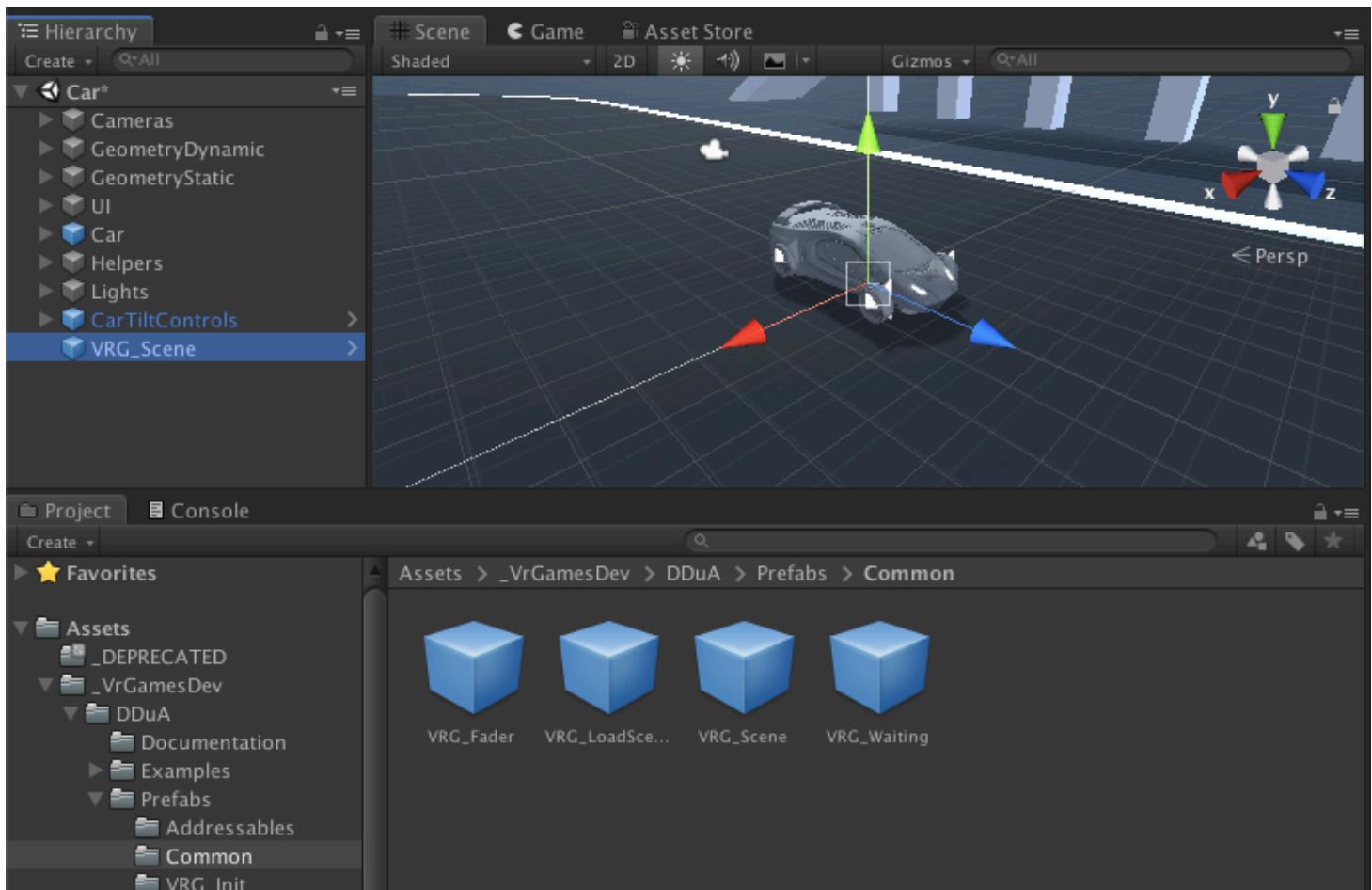
Assets > SampleScenes > Scenes >	
2dCharacter	
AircraftJet2Axis	
AircraftJetAI	
AircraftPropeller4Axis	
AircraftPropellerAI	
Car	
CarAIWaypointBased	
CharacterFirstPerson	
CharacterThirdPerson	
CharacterThirdPersonAI	
CharacterThirdPersonAI	
Particles	
RollerBall	



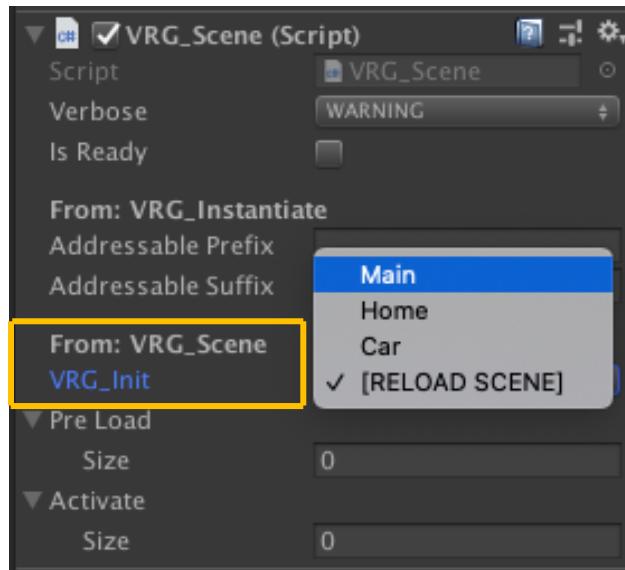
- 6) Make all the prefabs and scenes from the folder "Addressables" into name simplified addressables
 - a) There is one scene
 - i) 05 UnityCar / Addressables / Scenes / Home.unity
 - b) There are 5 prefabs game Objects
 - i) 05 UnityCar / Addressables / UI_Menu
 - ii) 05 UnityCar / Addressables / VRG_btn_Home
 - iii) 05 UnityCar / Addressables / VRG_Core
 - iv) 05 UnityCar / Addressables / Wait_My
 - v) 05 UnityCar / Addressables / Wait_Promo
- 7) Make the "car" scene from the folder (Assets/SampleScenes/Scenes) from the "Standard assets" package you downloaded into a name simplified addressable
 - i) Assets / SampleScenes / Scenes / Car.unity



- 8) Open the car Scene, and add a "VRG_Scene" prefab to the hierarchy, the prefab is in (Assets/_VrGamesDev/DDuA/Prefabs/Common/) folder.
This process is needed for every scene you want to use as an addressable



9) Configure the **VRG_Scene** prefab added, set the Scene that contains the **VRG_Init** prefab as "**Main**"



10) Play the game, Enjoy!

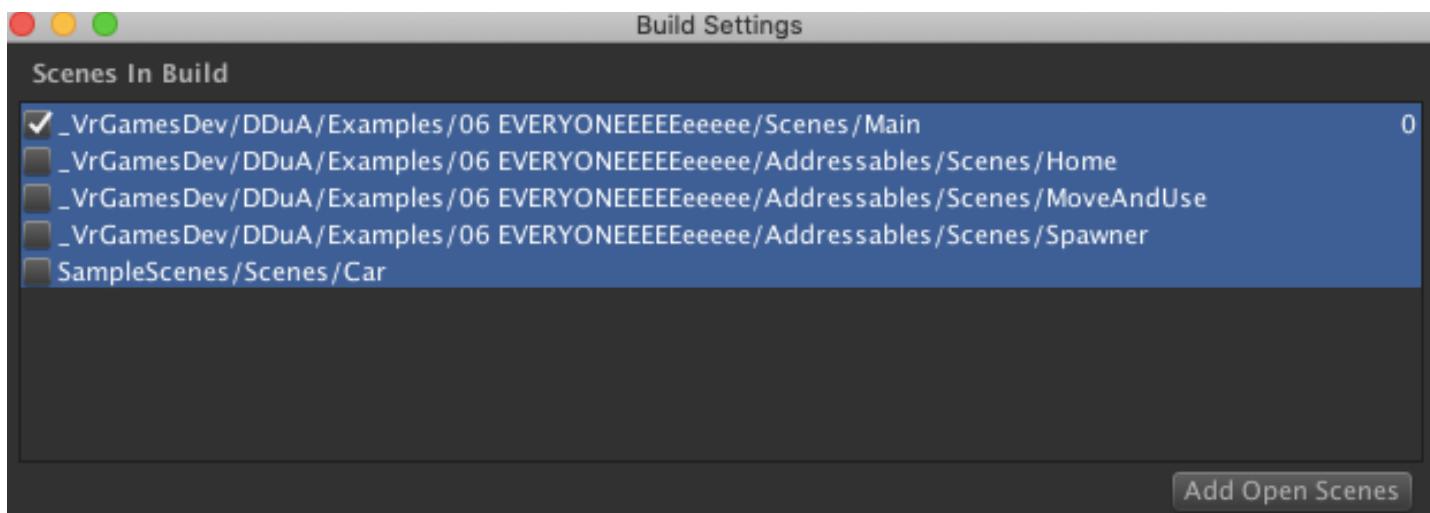


12.6 06 EVERYONEEEEeeeeee

This is the example that was compiled to showcase this package you can download it at <http://www.itchi.io/>
it includes all the examples into a single app.



- 1) Clean your PlayerPrefs
- 2) Make sure you downloaded "Standard Assets" as explained in [3.3 Standard Assets](#) section
- 3) Clean and delete all the previous scenes in your Build settings
- 4) Add to the build settings list the 4 scenes from the example folder, **Main** scene being the first one
- 5) Add to the build settings list the scene from the Assets/SampleScenes/Scenes/Car.unity



- 6) Make all the prefabs and scenes from the folder (Assets / _VrGamesDev / DDuA / 06 EVERYONEEEEeeeeee / Addressables) into name simplified addressables
 - a) You need to make 15 prefabs
 - b) You need to make 4 scenes
- 9) Make all the 8 prefabs from the folder (Assets / _VrGamesDev / Standard Assets) into name simplified addressables



Screenshot of the Unity Editor showing the Addressable Asset Manager. The left panel shows a tree view of addressable assets under 'Default Local Group (Default)'. The right panel shows a summary: 27 Objects, 23 Game Objects, and 4 Scenes. The middle section shows the details of the selected asset, 'Car', which points to 'Assets/SampleScenes/Scenes/Car.unity'.

- 10) Repeat the process as explained **14.5 05 UnityCar** section, subsection 8)
- 11) Make the “car” scene from the folder (Assets/SampleScenes/Scenes) from the “Standard assets” package you downloaded into a name simplified addressable
- 12) Open the car Scene, and add a “VRG_Scene” prefab to the hierarchy, the prefab is in (Assets/_VrGamesDev/DDuA/Prefabs/Common/) folder.
- 13) Configure the **VRG_Scene** prefab added, set the Scene that contains the **VRG_Init** prefab as “**Main**”
- 14) Play the game, Enjoy!

