Vr Games Dev

C.O.R.E.
Casual Oriented Rapid Engine

DECEMBER 2020

TABLE OF CONTENTS

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
December 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://www.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/Documentation/API.zip

2. OVERVIEW

C.O.R.E is a framework to develop games having all the ***"not a game"*** components, that are needed in every modern application. These components are already pre-configured and ready to use out of the box.

We use C.O.R.E as a foundation to our games, and we wanted to share it with you, maybe it could help you to speed up your journey to create amazing games.

We try our best to have everything documented, code organized, well commented with first class standards, and we will continue to develop and update this package when we develop a new module or functionality, since we use it ourselves, we are confident we can use your feedback to grow this package.

2.1 TL : DR

1. Download and install Remote config package
2. Open the scene: Assets/_VRGamesDev/5 Seconds/Scenes/ Home
3. Play ***"5 second"*** mini game and the missions.
4. Go to the menu Tools/VrGamesDev/Create new game
5. Create your own new game
6. Enjoy
7. Check the 10 sample scenes

2.2 C.O.R.E Components

C.O.R.E has some basic modules needed for every game, it is a compilation of funcionality like a swiss knife.

Systems integration

- Remote: For easy remote configuration of your application, enabling you to update without generating a new build

Toolbox modules:

- ***Editor:*** Tools menu for easier usage
- ***Skins:*** You can add and modify youe whole settings with a few configurations
- ***VRG_Remote:*** The control and integration of the Unity Remote Config
- ***VRG_GraphicalNumer*** A nice graphical number for scores and fancy numbers
- ***VRG_Fader:*** Basic Fader, a nice full screen fading UI screen
- ***VRG_FaderScene:*** Basic Fader, it offers you a smooth way to transition between scenes
- ***Scene Management:*** Go to scene, Fader, and scene reading from build settings.
- ***VRG_Game and Missions:*** Game control and its missions.
- ***VRG_Session*** All the management and the control of the player preferences
- ***VRG_Audio*** Audio controls and their channels.
- ***Utils:*** Many utility scripts to do plenty of useful funcionalities.

3. REQUIREMENTS TO USE C.O.R.E

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

3.1 Unity Remote Config

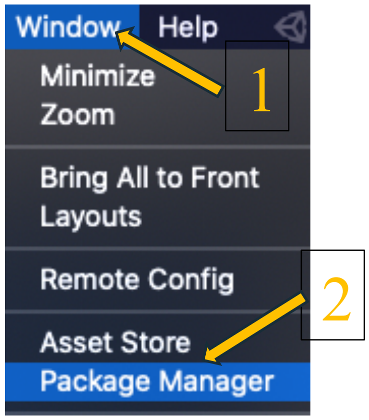
Unity Remote Config is a cloud service that allows you to tune your game design without deploying new versions of your application.

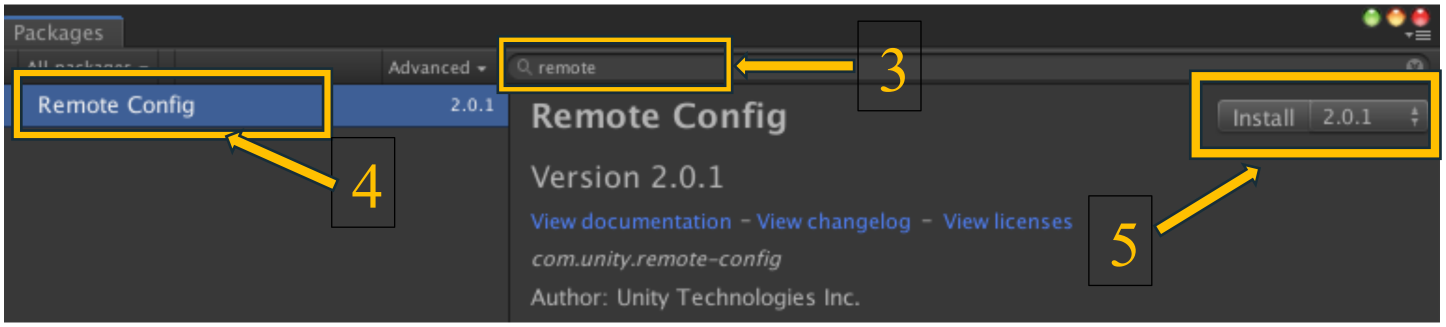
Documentation: You can read more about this module here
https://docs.unity3d.com/Packages/com.unity.remote-config@2.0/manual/index.html

Minimum version: 2.0.1

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 "***Remote***"
4) Select the package "***Remote Config***"
5) Click **Install**, the version that was tested with this package was the most recent ***(2.0.1***) at the creation of this documentation.





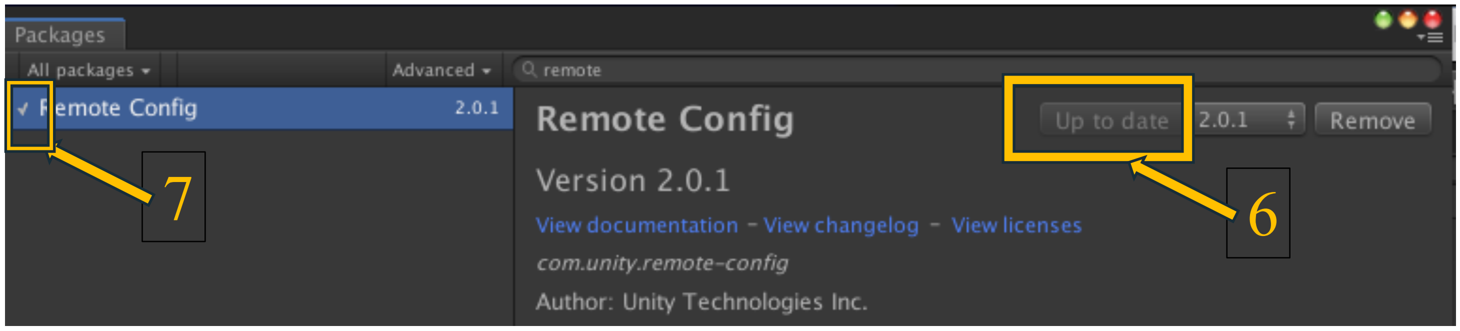
Remote Config

Version 2.0.1

com.unity.remote-config

Author: Unity Technologies Inc.

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 was successfully installed.



Remote Config

Version 2.0.1

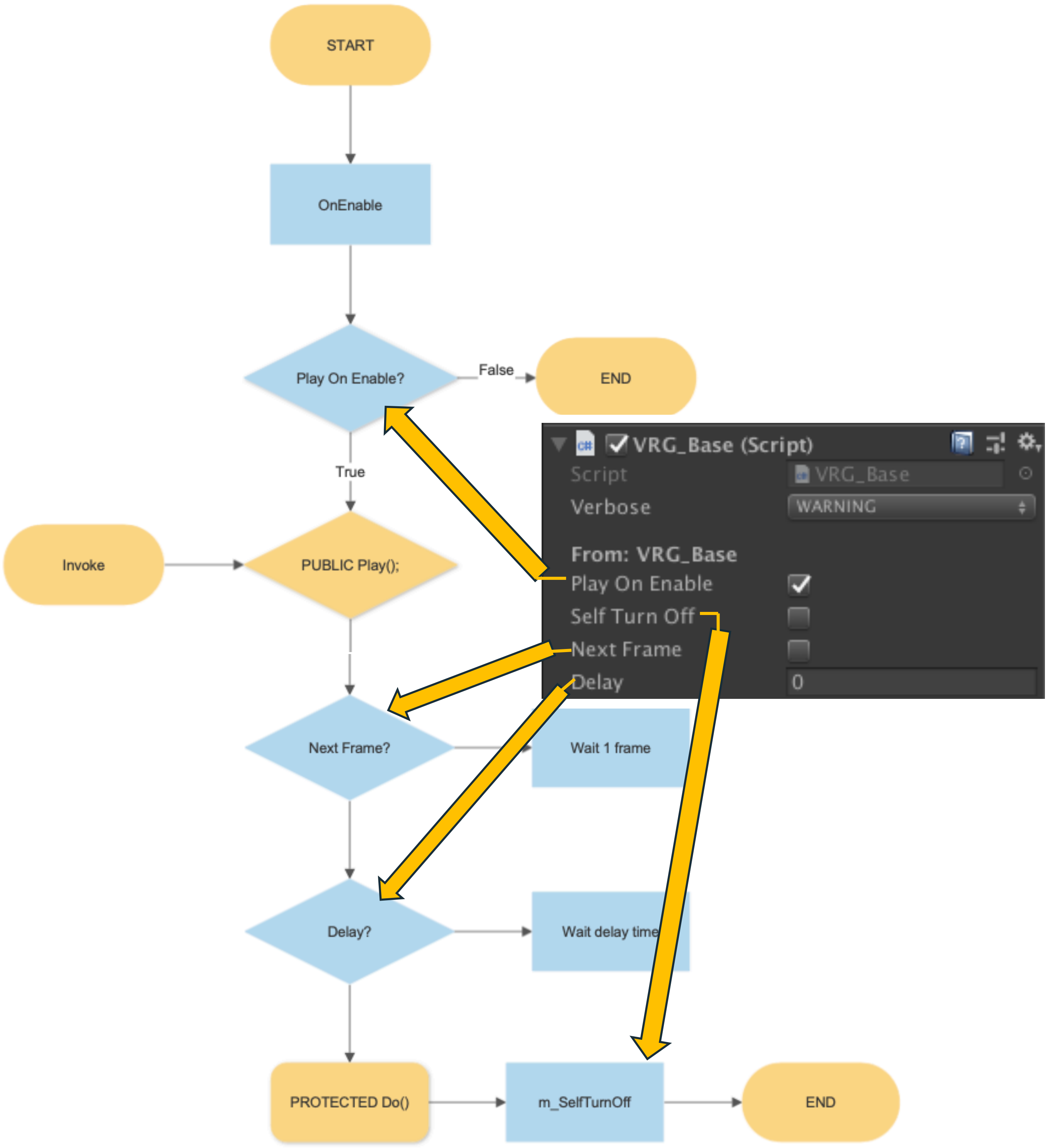
com.unity.remote-config

Author: Unity Technologies Inc.

8) Congratulations, we recommend you read the official documentation to understand the power of this module

4. VRG_BASE

This is the base class for every class in the VR Games Dev packages, and it has a common funtionality. You can configure 4 data, and the process will trigger Play, then it will wait for the delay duration and then do its thing.

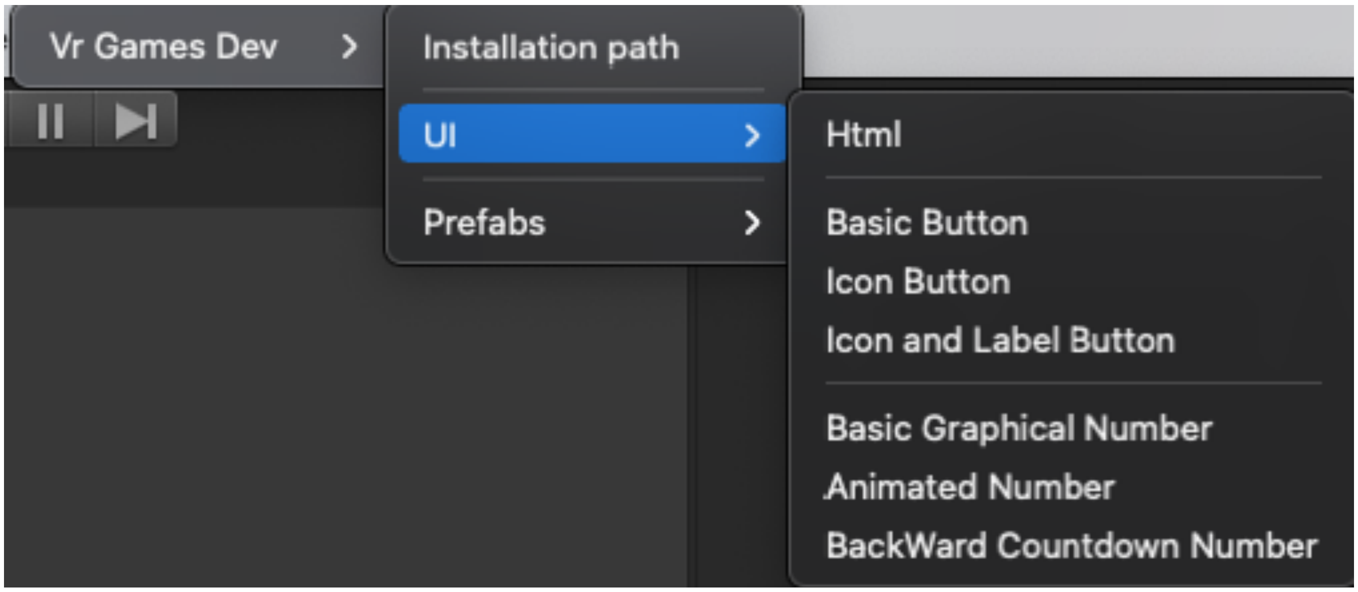


Script VRG_Base
Verbose WARNING

From: VRG_Base
Play On Enable ✓
Self Turn Off ☐
Next Frame ☐
Delay 0

5. MENU TOOLS->VR_GAMES_DEV

For your easy usage, we have added nice menus under the Tools menu option. You can use most of our swiss toolbar from there.



5.1 Menu

It has many options to use the package:

1) ***Instalation path:*** Where is the _VR Games Dev packages installed
2) ***UI:*** All the User Interphase components
3) ***Prefabs:*** General purpose prefabs
4) ***Remote:*** The remote configuration pre-configureds
5) ***Missions:*** All the data needed to configure the missions and campaign
6) ***Utils:***: Plenty of useful scripts.

6. MENU TOOLS->VR_GAMES_DEV-> UI

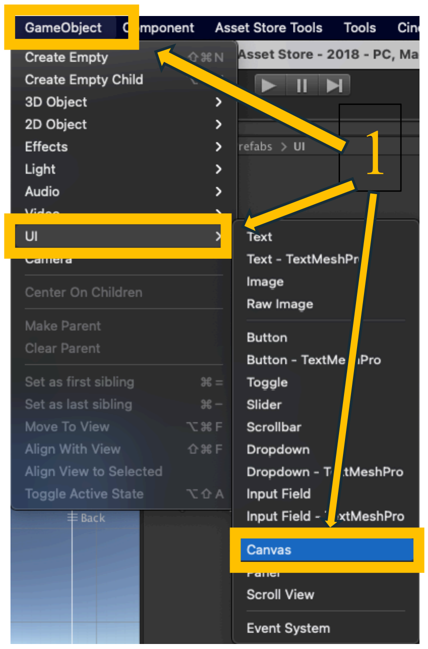
The UI Section of the menu provides some fancy elements and configured UI unity elements to provide some funcionality and to allow an easy re-skin of the UI with a few configurations. All these prefabs need a Canvas component to be created.

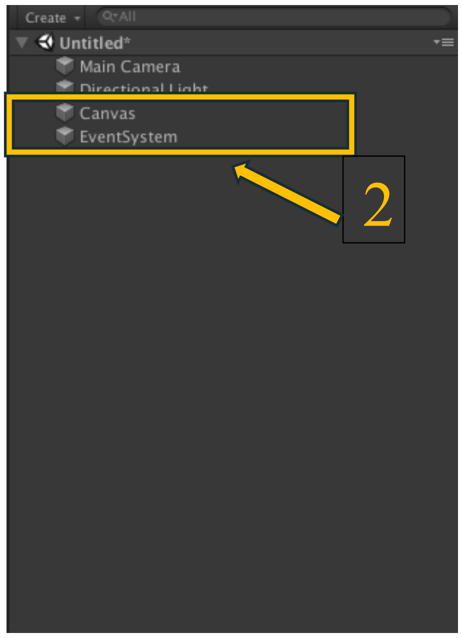
6.1 Html component

This is just a simple configuration to have a basic layout with header, body and footer, hence the name.
Scene sample***:*** *Assets/_VrGamesDev/CORE/Scenes/01 UI*

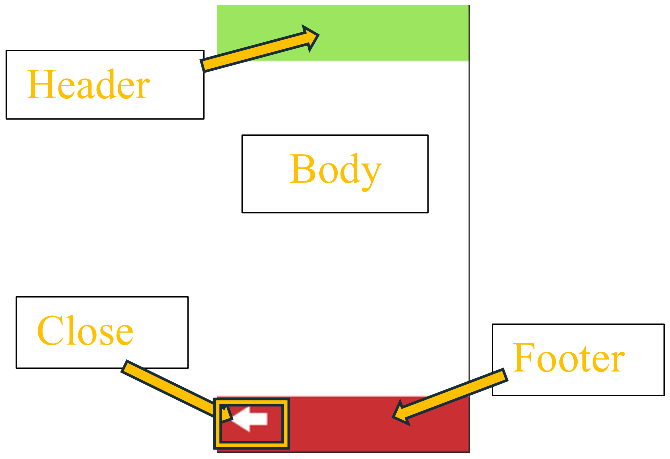
Documentation: You can read more about this prefab here
UI Canvas: https://docs.unity3d.com/2020.2/Documentation/Manual/UICanvas.html
Event system: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/EventSystem.html
Auto Layout: https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/comp-UIAutoLayout.html

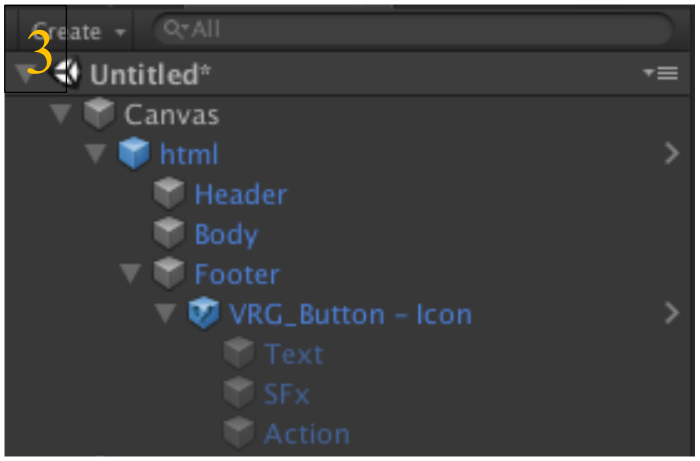
1) Create a Canvas component, from the menu select, **GameObject -> Ui -> Canvas**
2) You now have a ***Canvas*** and ***EventSystem*** components in the hiearchy.





3) Select the canvas and add a ***HTML*** component from the menu **Tools -> Vr Games Dev -> UI -> Html**. The basic funcionality Is it has a button to close itself to return to the previous menu, and some preconfigured auto layout for easier and faster prototyping, remember to configure and custom to your own needs.



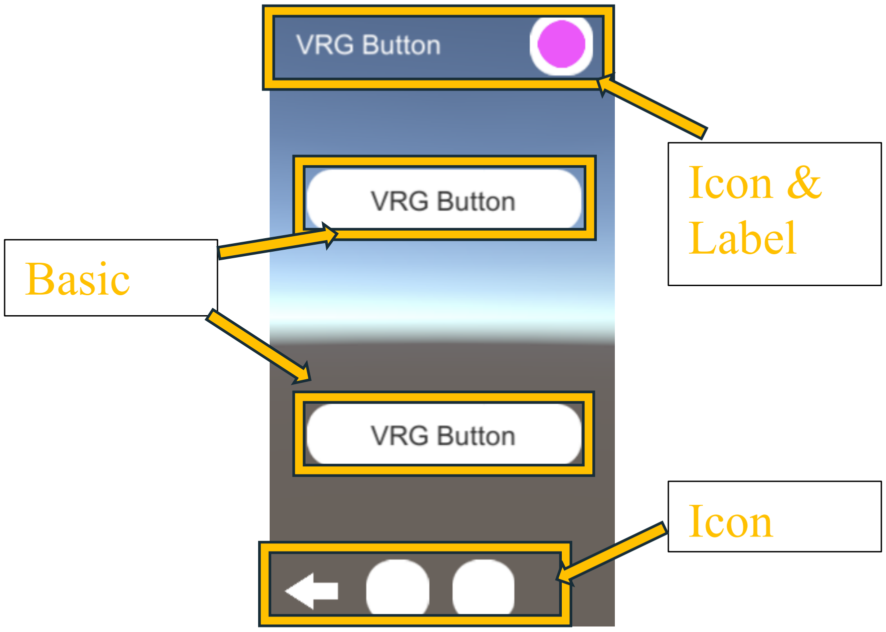


6.2 Buttons

It adds a click sound effect, and uses a "generic click" gameobject to trigger the button effect for easier interaction with other parts of the game. All these items can be configured from the VRG_Game object settings. The child object ***"Action"*** is triggered ***"OnClick"*** event.
Basic button: It'is just the most basic button, every other prefab inhertiance from this one.
Icon and Label: It has an Icon that can switch when you click it to show diferent status
Icon: A tiny icon, with a background

Documentation: You can read more about this prefab here
UI Canvas: https://docs.unity3d.com/2020.2/Documentation/ScriptReference/UIElements.Button.html

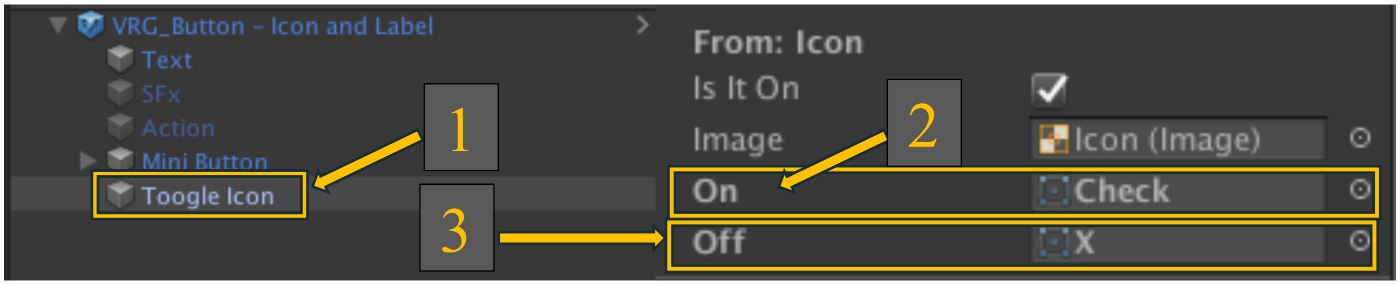
1) Select your ***canvas-html-Header*** gameobject
2) Add a **Basic button** component, from the menu ***Tools -> Vr Games Dev -> UI -> Icon and Label Button***.
3) Select your ***canvas-html-Body*** gameobject
4) Add two **Basic button** components, from the menu ***Tools -> Vr Games Dev -> UI -> Basic Button***.
5) Select your ***canvas-html-Footer*** gameobject
6) Add two **Basic button** components, from the menu ***Tools -> Vr Games Dev -> UI -> Icon Button***.
7) Play the scene and have fun



Change Icon & label images

When you click the icon and label, the icon button image toogle between the two images.

1) Select the ***VRG_Button - Icon and Label*** game object and select its child named ***Toogle Icon***
2) Into the inspector select the ***"On"*** image and browse for the ***"Check"*** image
3) Repeat the process for the "***Off***" imagen and select the ***"X"*** image
4) Play the Scene and click the Icon and Label button



6.3 Graphical Numbers

Sometimes you have the need to have beautiful fancy colored numbers for scores, so you made some images for that, Configurate the numbers name and enjoy nice score and levels.

Scene sample: *Assets/_VrGamesDev/CORE/Scenes/02 Graphical Numbers*

Documentation: You can read more about this prefab here

VRG_GraphicalNumber: https://www.vrgamesdev.com/documentation/#VrGamesDev.VRG_Base

UI Image: https://docs.unity3d.com/2020.2/Documentation/ScriptReference/UIElements.Image.html
UI Text: https://docs.unity3d.com/2020.2/Documentation/ScriptReference/UIElements.TextField.html

1) Create ***canvas*** UI and a ***Html*** component
2) Add an Image UI (any component would work, even an empty one, but the image provides a white background)
3) Set the width and height to 400,120

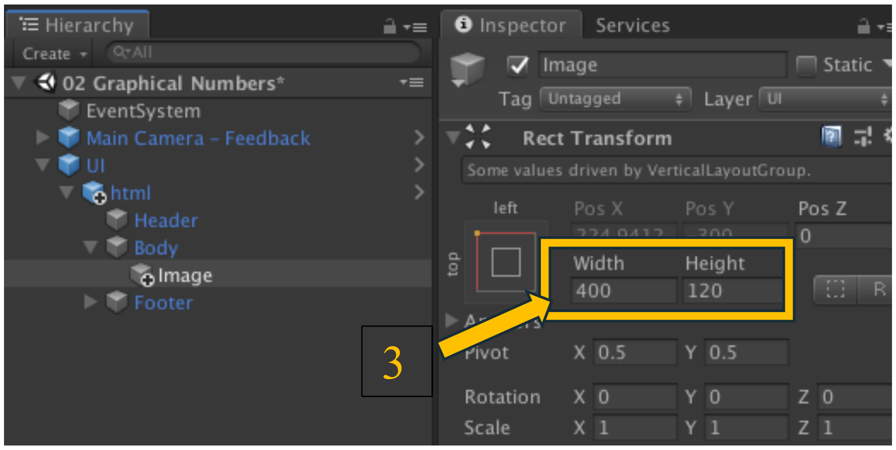
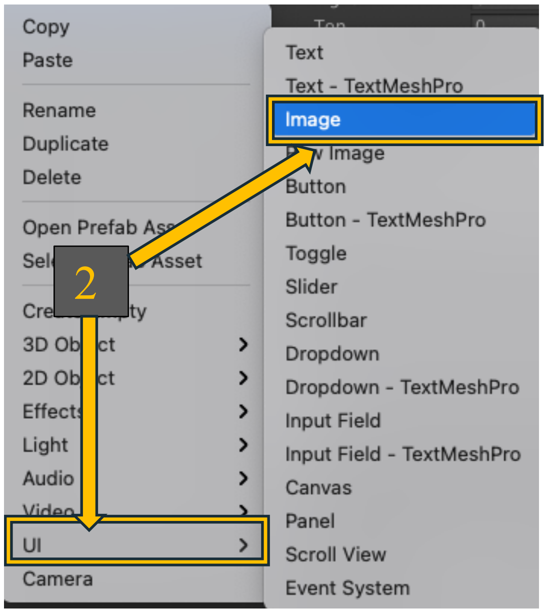
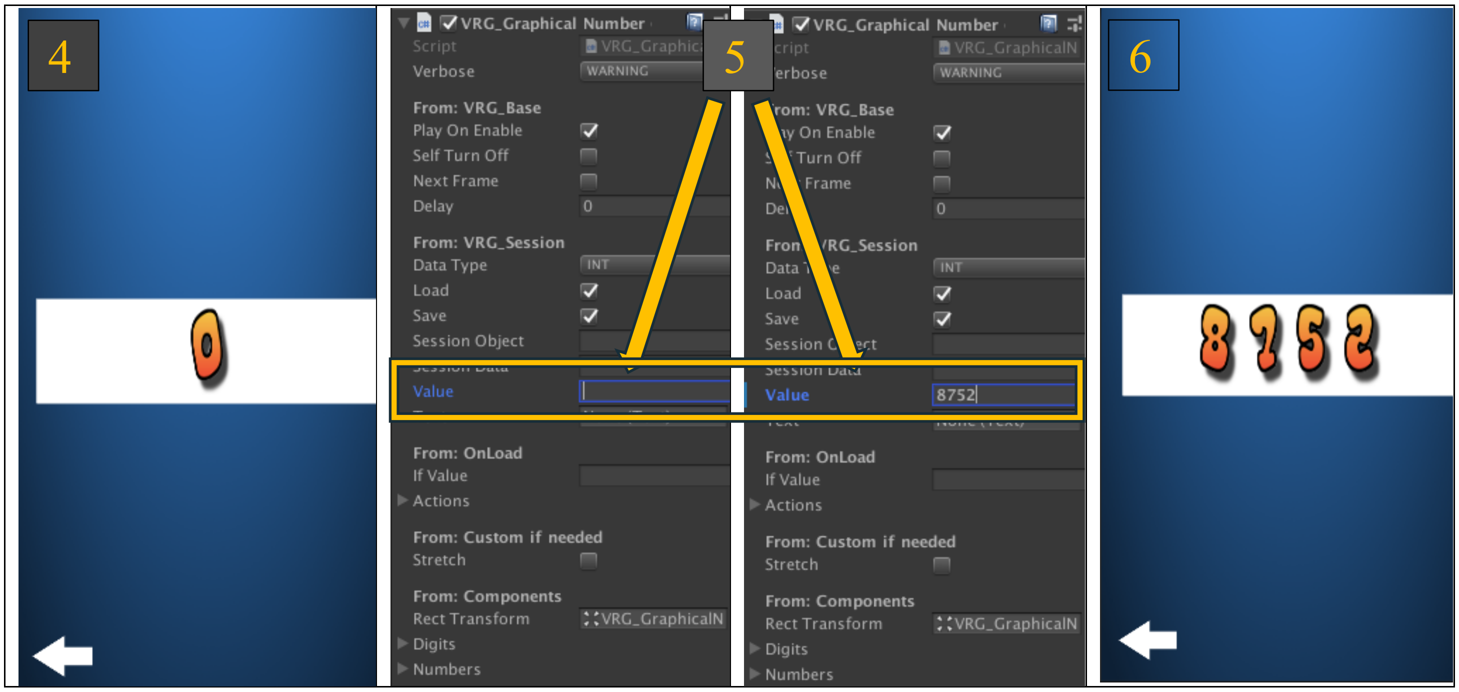


Image
Static
Tag Untagged
Layer UI
Some values driven by VerticalLayoutGroup.
Pos X 224.9412 Pos Y -300 Pos Z 0
Width 400 Height 120
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

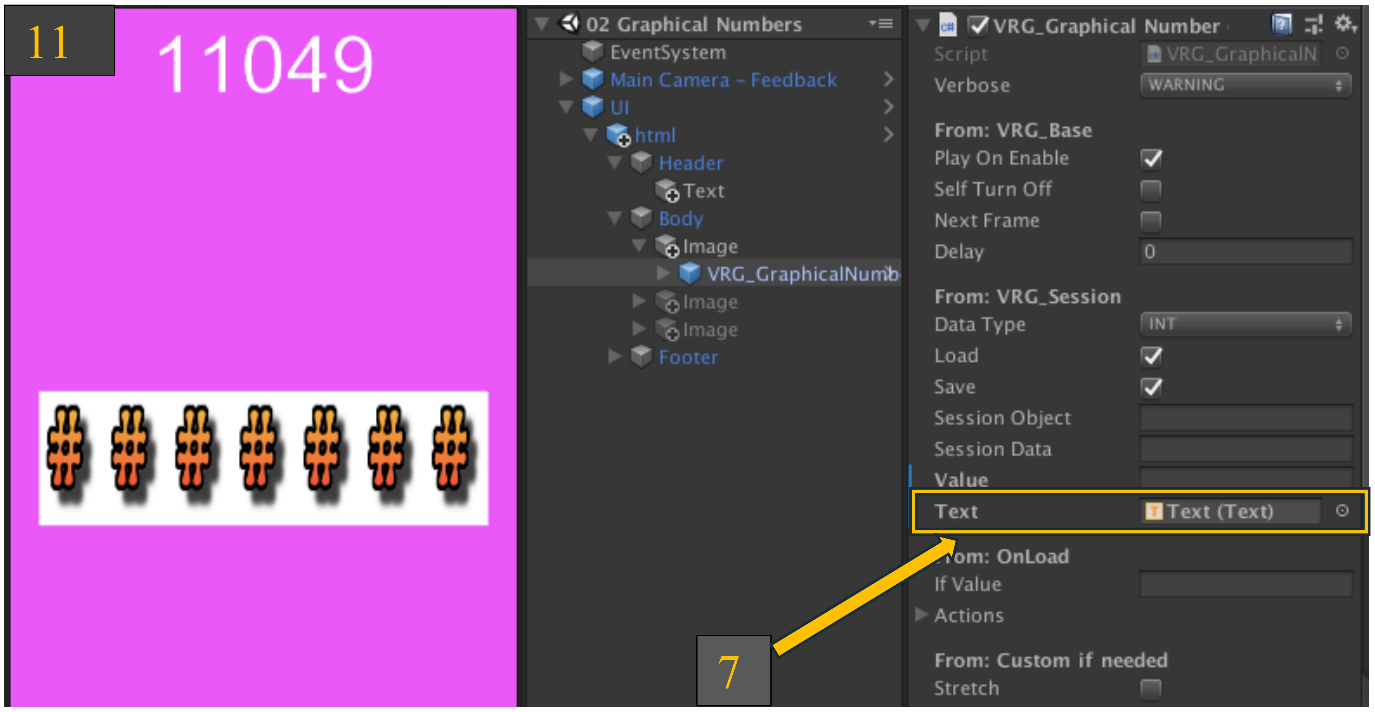


4) Play and you have a nice 0 number displayed in your image
5) Now, select the VRG_GraphicalNumber component inside of your image and add a diferent value in the Inspector, for example 8752
6) Play Again…



VRG_Graphical Number
Script VRG_GraphicalN
Verbose WARNING
From: VRG_Base
Play On Enable ✓
Self Turn Off
Next Frame
Delay 0
From: VRG_Session
Data Type INT
Load ✓
Save ✓
Session Object
Session Data
Value 8752
From: OnLoad
If Value
Actions
From: Custom if needed
Stretch
From: Components
Rect Transform VRG_GraphicalN
Digits
Numbers

7) You can also get the data from a Textfield UI object, add a Textfield component to the canvas,
8) Add a value, to the text component for example 11,049
9) Link It to the VRG_GraphicalNumber Save Text into the inspector
10) Reset the “value” from the value field, this data takes priority over the Text field.
11) Play Again…



02 Graphical Numbers
- EventSystem
- Main Camera - Feedback
- UI
  - html
    - Header
      - Text
    - Body
      - Image
        - VRG_GraphicalNumb
      - Image
      - Image
    - Footer

VRG_Graphical Number
Script: VRG_GraphicalN
Verbose: WARNING

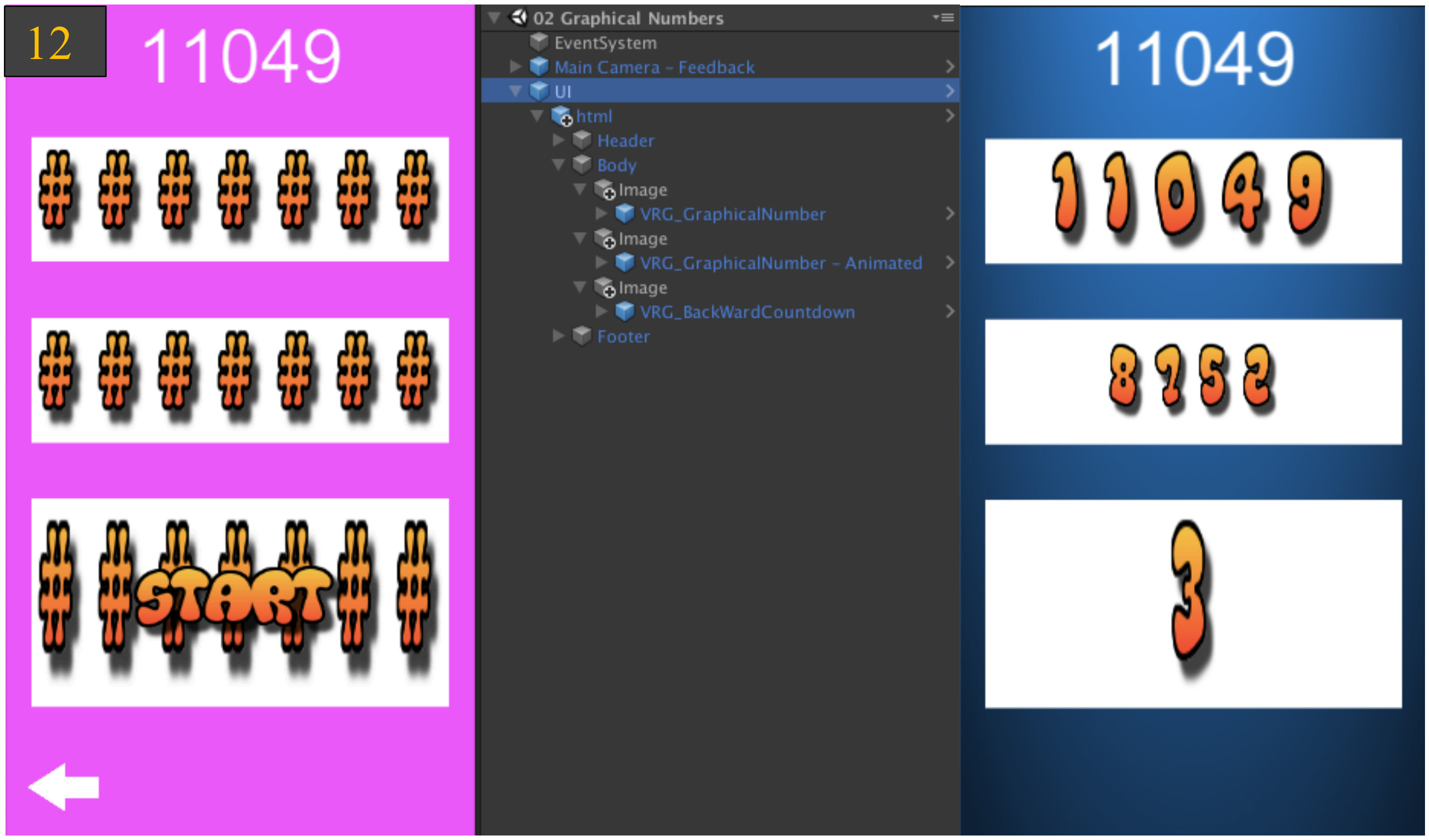
From: VRG_Base
Play On Enable: ✓
Self Turn Off
Next Frame
Delay: 0

From: VRG_Session
Data Type: INT
Load: ✓
Save: ✓
Session Object
Session Data
Value
Text: Text (Text)

From: OnLoad
If Value
Actions

From: Custom if needed
Stretch

12) Repeat from 2-11 but this time add a VRG_GraphicalNumber – Animated and a VRG_BackWardCountdown



02 Graphical Numbers
- EventSystem
- Main Camera - Feedback
- UI
  - html
    - Header
    - Body
      - Image
        - VRG_GraphicalNumber
      - Image
        - VRG_GraphicalNumber - Animated
      - Image
        - VRG_BackWardCountdown
    - Footer

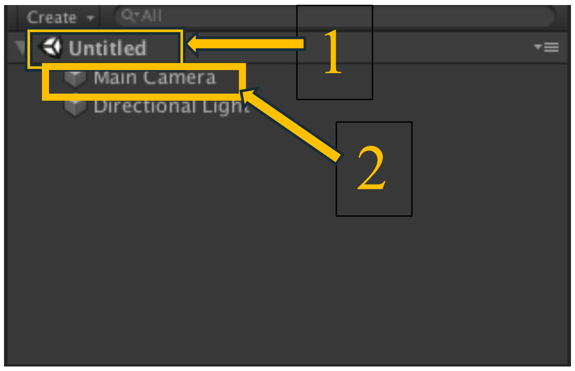
7. MENU TOOLS->VR_GAMES_DEV-> PREFABS

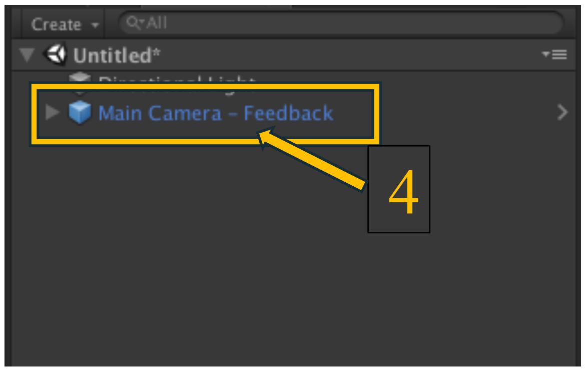
You can use most of our swiss toolbar in the following pages let's explain each of them, the prefabs classification are complex and full prefabs that include childs, and many scripts to achieve some funcionality.
Scene sample**:** *Assets/_VrGamesDev/CORE/Scenes/03 Camera And Fader*

7.1 VRG_Camera

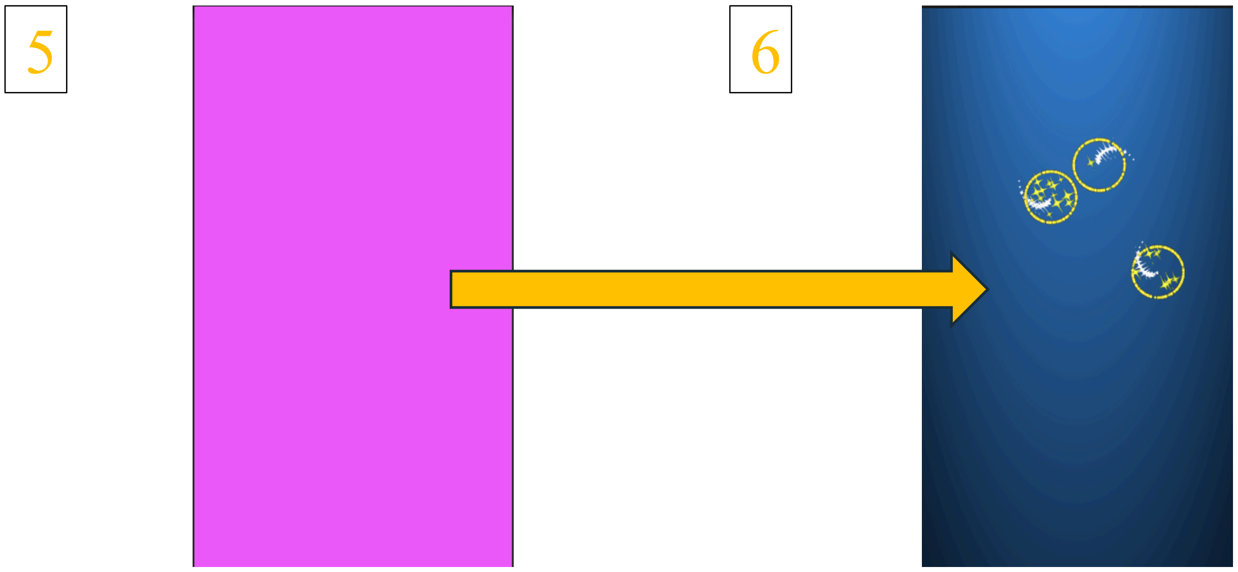
A Unity Camera with some scripts attached, it has a "onClick" feedback particles with a Sound effect and a full curtain that is displayed at the very end of the camera far clip plane. You can custom your own particle effect, but it has one to use it out of the box. To use it and undersatnd it, do the following
Documentation: You can read more about this prefab here
https://www.vrgamesdev.com/documentation/#VrGamesDev.VRG_Base

1) Create a new Scene
2) Delete de default ***"Main Camera"***
3) Select the Camera object into the menu *"Tools -> Vr Games Dev -> Prefabs -> Main Camera - Feedback*
4) You now have a "Main Camera – Feedback"





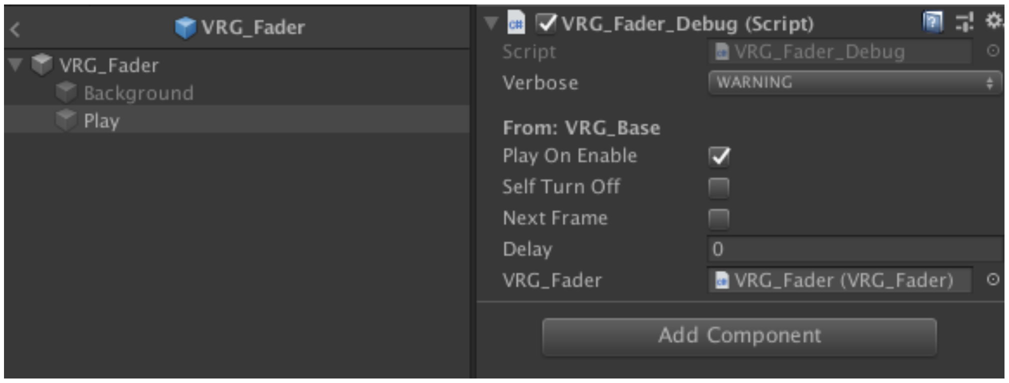
5) Now you have a pink background
6) Click play, you should see a blue gradiant background and when you click anywhere you get a particle effect as feedback and a click sound.



7.2 VRG_Fader

A full screen fader, you can suscribe to the "OnFadeIn" event and to the "OnFadeOut" events to get notified when it Is over, you can trigger on the "Play" child in the prefab to see it working
Documentation: You can read more about this prefab here
https://www.vrgamesdev.com/documentation/#VrGamesDev.VRG_Base

1) Add the Fader object from the menu *"Tools -> Vr Games Dev -> Prefabs -> VRG_Fader*
2) You now have a "***VRG_Fader"***
3) Select the "Play" child inside the ***"VRG_Fader"*** and activate it
4) It fades in and out, simple and effective



Script: VRG_Fader_Debug
Verbose: WARNING
From: VRG_Base
Play On Enable: ✓
Self Turn Off: ☐
Next Frame: ☐
Delay: 0
VRG_Fader: VRG_Fader (VRG_Fader)

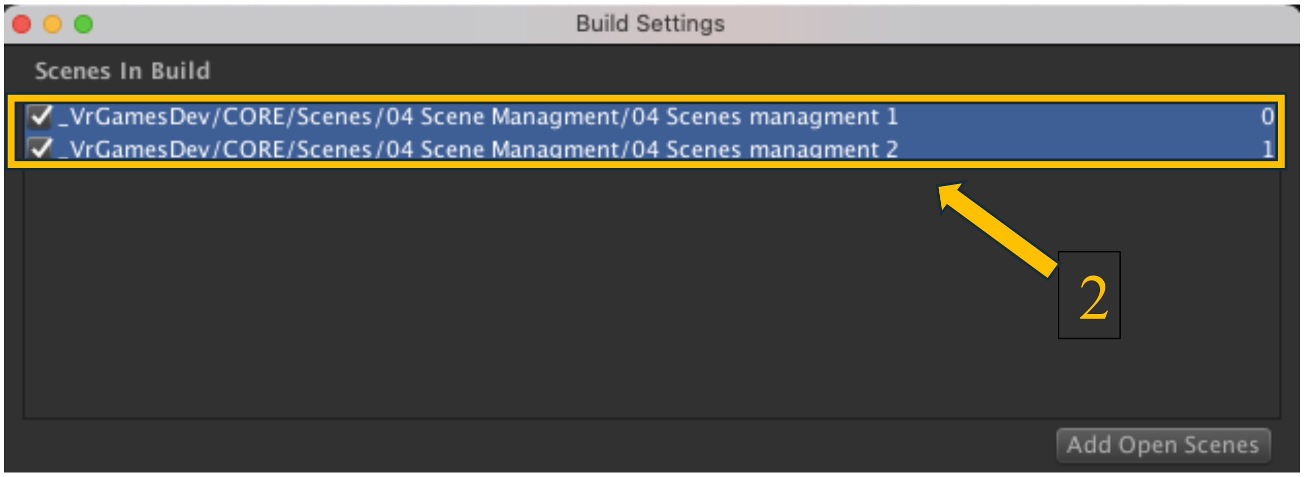
You can check the VRG_Fader_Debug.cs script, used in the "Play" chield, it plays the fade and suscribe to the event OnFadeOut to deactivate itself.

7.3 VRG_FaderScene

A full screen fader, that is activated when a new scene is loaded, it stays faded until the scene is fully loaded.
Scene sample*: Assets/_VrGamesDev/CORE/Scenes/04 Scene Management*

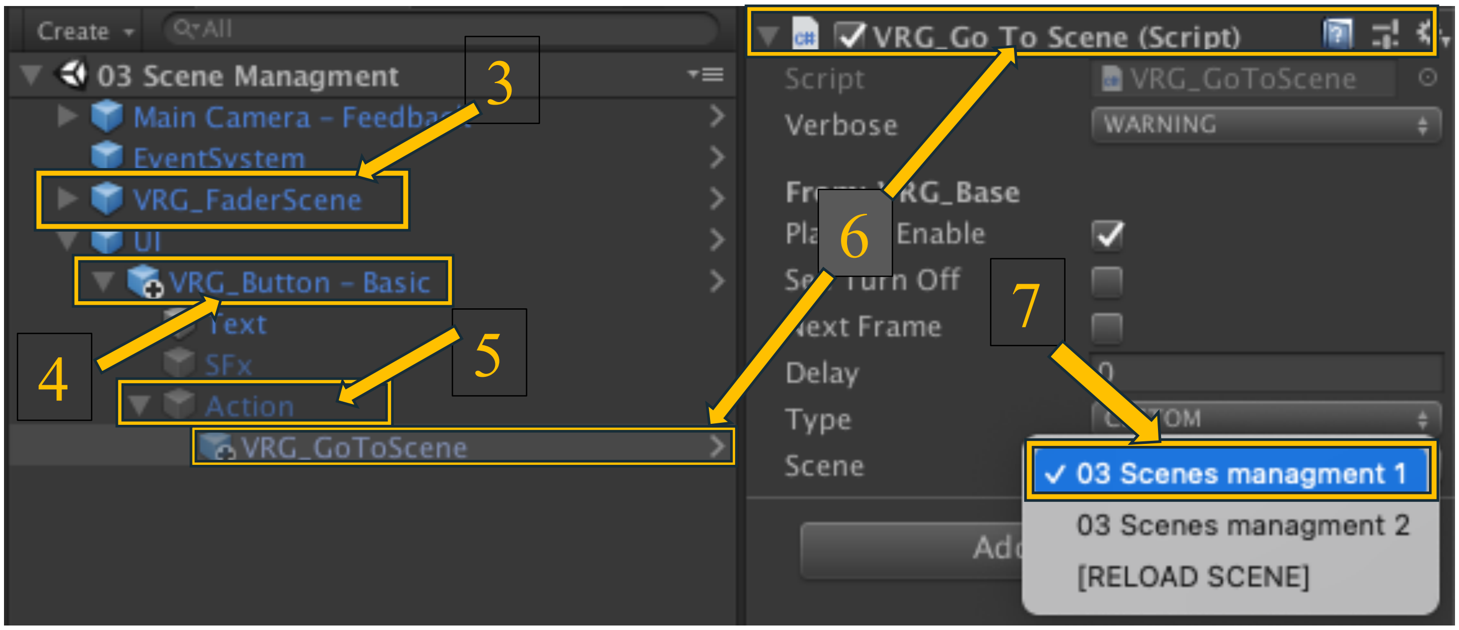
Documentation: You can read more about this prefab here
https://www.vrgamesdev.com/documentation/#VrGamesDev.VRG_Base

1) Open the File -> build settings from the menu
2) Drag to add the two scenes
   a) Assets/_VrGamesDev/CORE/Scenes/04 Scenes managment/04 Scenes managment 1
   b) Assets/_VrGamesDev/CORE/Scenes/04 Scenes managment/04 Scenes managment 2

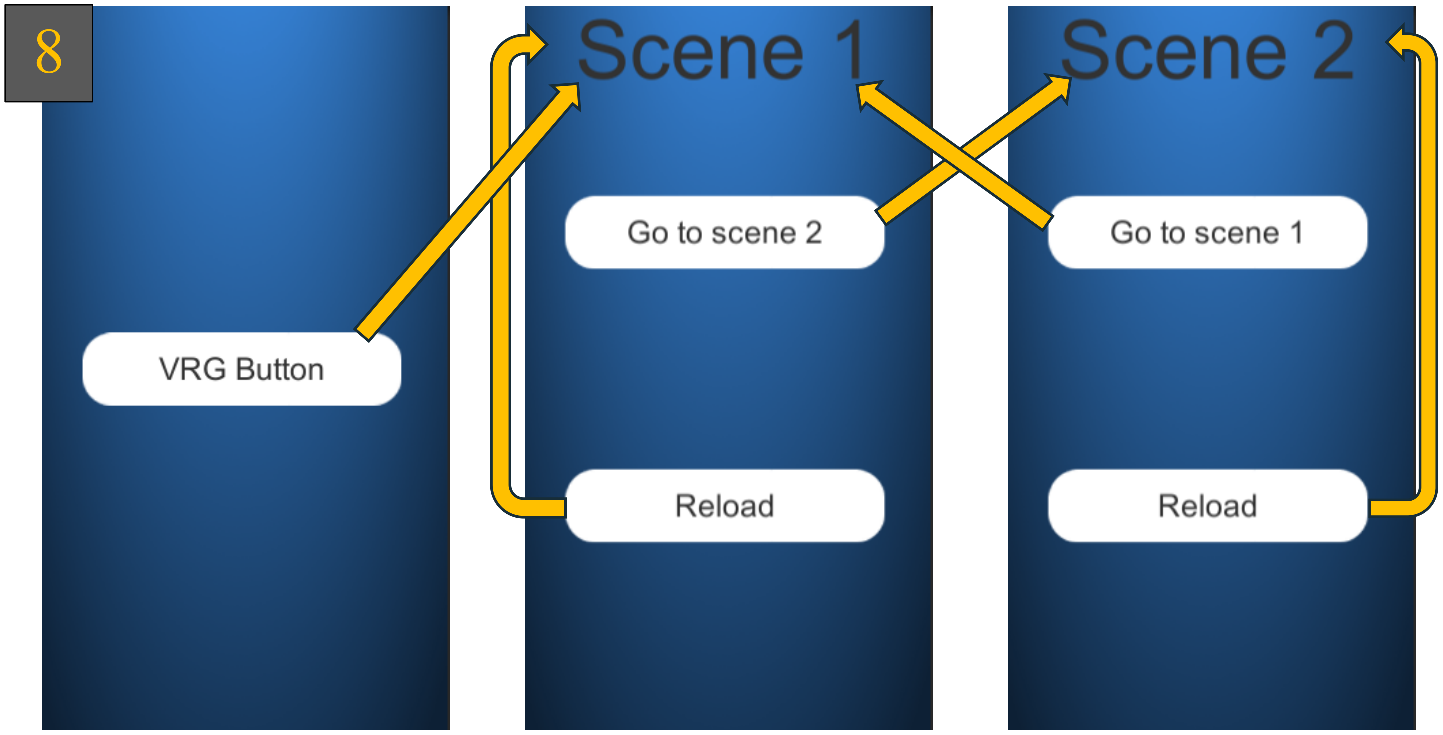


✓ _VrGamesDev/CORE/Scenes/04 Scene Managment/04 Scenes managment 1 0
✓ _VrGamesDev/CORE/Scenes/04 Scene Managment/04 Scenes managment 2 1

3) Open a new scene, and add a ***VRG_FaderScene*** prefab from the menu *"Tools -> Vr Games Dev -> Prefabs -> VRG_FaderScene"*
4) Add a ***Basic button*** prefab, from the menu ***Tools -> Vr Games Dev -> UI -> Basic Button***.
5) Open the GameObject from the Prefab ***UI -> VRG_Button – Basic -> Action***
6) Add a ***VRG_GoToScene*** prefab, into the "Action" child



7) You will get listed all the scenes added to the build settings. Custom the scene to load, and select the first one ***"03 Scenes managment 1"***
8) Play and click the button, you will load the scenes managments



8. VRG_SESSION

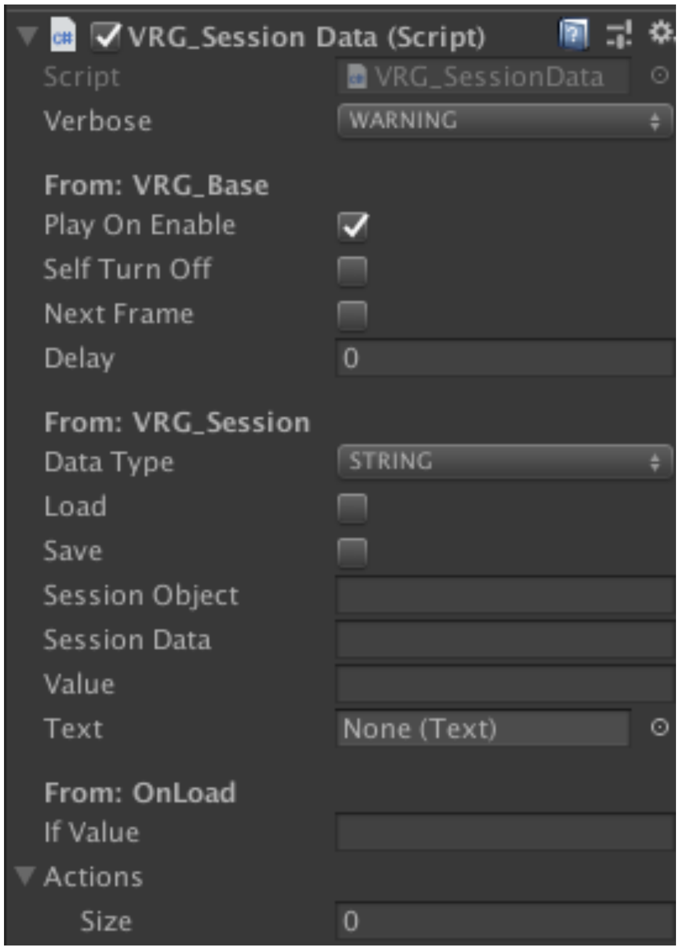
Many of our components can save its data to the player prefs, they are able to save their ownt data, let's check some of the components.

Documentation: You can read more about this prefab here
https://docs.unity3d.com/ScriptReference/PlayerPrefs.html
https://docs.unity3d.com/2020.2/Documentation/Manual/script-InputField.html
https://www.vrgamesdev.com/documentation/#VrGamesDev.VRG_Base

8.1 VRG_SessionData

This class save the data from a UI text component or a predefined value, many components inheritance from this class, these are the data needed to configure it:

Scene sample: *Assets/_VrGamesDev/CORE/Scenes/05 VRG_SessionData*



VRG_Session Data (Script)
Script VRG_SessionData
Verbose WARNING

From: VRG_Base
Play On Enable ✓
Self Turn Off
Next Frame
Delay 0

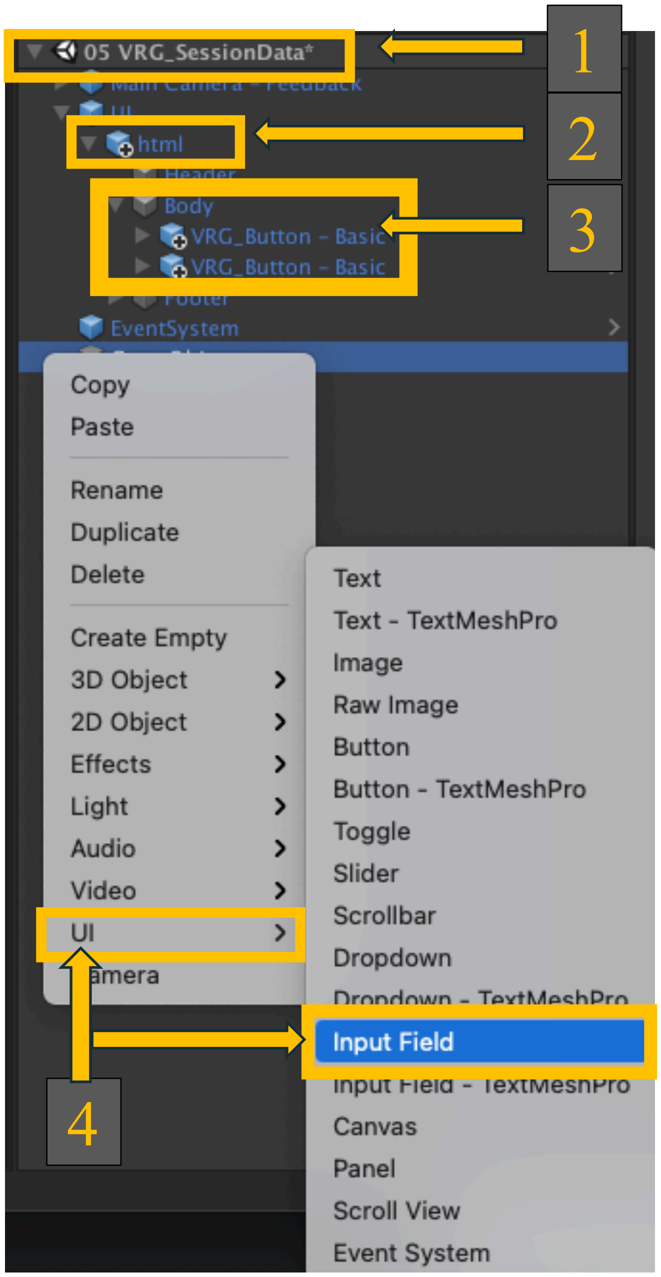
From: VRG_Session
Data Type STRING
Load
Save
Session Object
Session Data
Value
Text None (Text)

From: OnLoad
If Value
Actions
Size 0

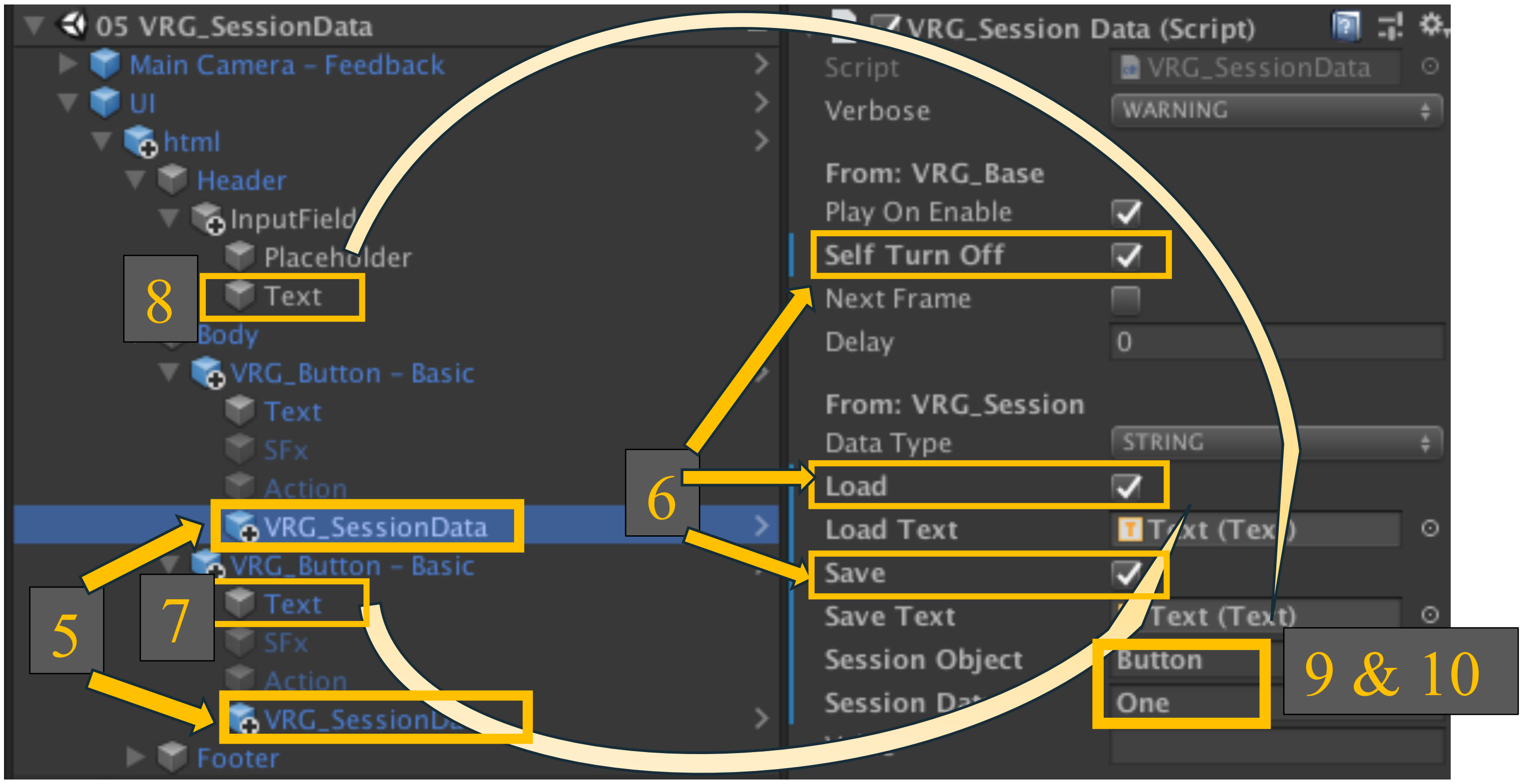
a) ***Data Type*** Select the data type to save or load, from String, int, float and bool
b) ***Load:*** Check it true, to load data from the session
c) ***Save:*** Check it true to save the value data to the session
d) ***Session Object:*** The name of the oject to save, It is useful to save class->property
e) ***Session Data:*** The data name to save it, it pair its name with the Session object name to create a unique value
f) ***Value:*** The value to save, make sure you match the data type defined above.
g) ***Text:*** If set the value data from the session will save and load to this UI text field

From: OnLoad:
a) ***If Value:*** Compare this value to the value from the session, if true, the actions objects will be activated
b) ***Actions:*** Array of game objects, this objects will activate if the value match from session

1) Create a new scene,
2) Add one Html component
3) Add two basic buttons to the body
4) Add an InputField to the header and custom it to your taste.



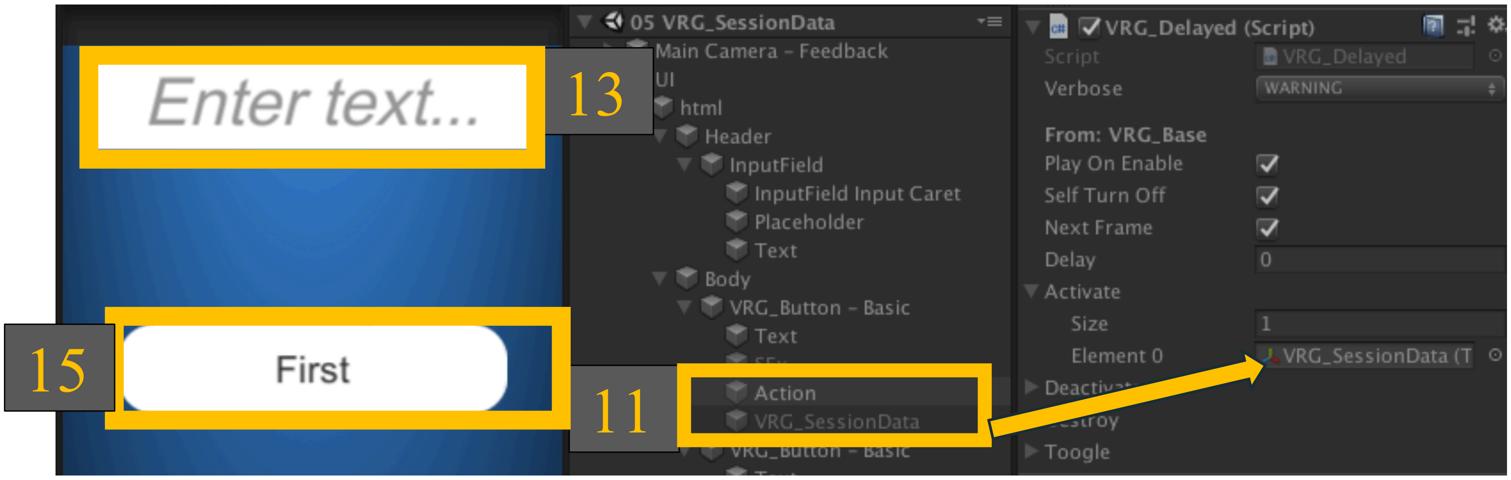
5) Add a VRG_SessionData prefab in each of the Basic Buttons. From menu menu ***Tools->VR Games Dev -> General -> SessionData.*** It saves the Inputfield string when you click it, let's configure it.
6) Select the newly added ***VRG_SessionData*** prefab, and toogle true the options ***Self turn off,*** the ***Save*** and ***Load***
7) Now let's add the text UI component where the session's data will be loaded, Drag the Text element from the ***VRG_Button – Basic*** prefab to the ***Load Text*** property.
8) Repeat the procedure from where the session data will take its data to save it for future reference, and drag the Text from the inputField from the Header into the ***Save text*** property.
9) Now let's name our Session, for the Sesison Object pick anything you want, in this example we will use "***Button***"
10) Same with the ***Session data***, we will use the number of the button "One" and "Two" respectively



VRG_Session Data (Script)

Script	VRG_SessionData
Verbose	WARNING
From: VRG_Base	
Play On Enable	✓
Self Turn Off	✓
Next Frame	
Delay	0
From: VRG_Session	
Data Type	STRING
Load	✓
Load Text	Text (Text)
Save	✓
Save Text	Text (Text)
Session Object	Button
Session Data	One

11) Finally, add this object to the action child of our ***VRG_Button – Basic***, drag the ***VRG_SessionData*** GameObjet to the "activate" property.
12) Play it
13) Type something in the input boxDrag then press the any one button
14) Stop the game
15) Replay it, see how what you typed is now loaded from the previous game session.



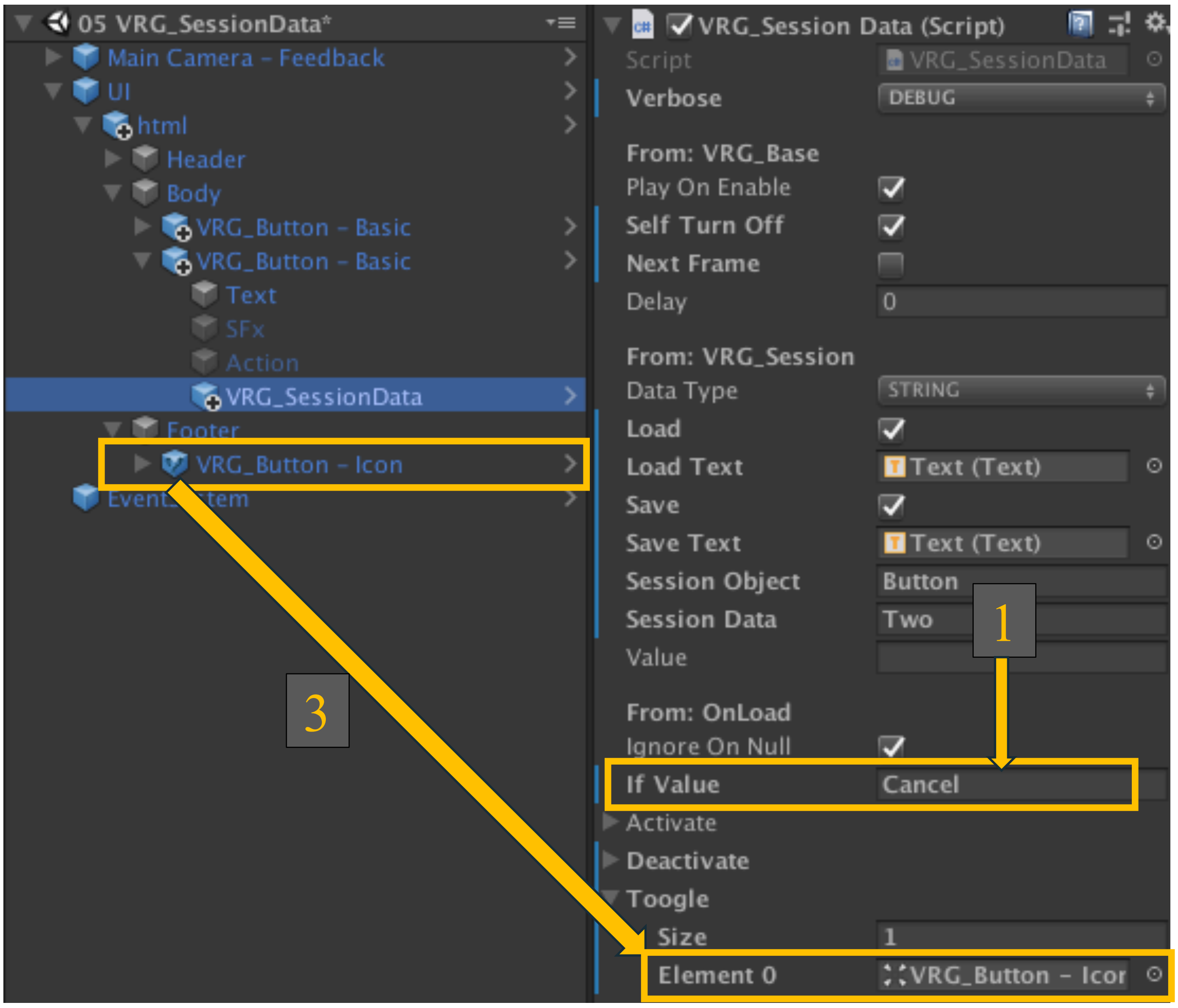
VRG_Delayed (Script)

Script	VRG_Delayed
Verbose	WARNING
From: VRG_Base	
Play On Enable	✓
Self Turn Off	✓
Next Frame	✓
Delay	0
Activate	
Size	1
Element 0	VRG_SessionData (T
Deactivate	
Destroy	
Toogle	

8.2 "From: Onload" section

Let's configure the last section, when you expect a certain value (maybe a score, a name, or a secret passcode) do something if the value of the data from session is equal to something.

1) In the Button two, configure the ***"From: Onload"*** section, in the ***"If Value"*** property type something you want, in this example we will use the "Cancel" word.
2) You have three array options to use if the data is meet:
   a) Activate: All the game objects in this array will activate
   b) Deactivate: All the game objects in this array will deactivate
   c) Toogle: All the game objects in this array will switch between activate -> deactivate
3) Add the ***VRG_Button – Icon*** from the Footer section, into the Toogle array
4) Play the game
5) Type "Cancel" in the input feld
6) Watch the little arrow dissapear
7) Click as much as you want, every click will appear-dissapear
8) Type something else into the inputfield, clicj button two.



Script	VRG_SessionData
Verbose	DEBUG
From: VRG_Base	
Play On Enable	✓
Self Turn Off	✓
Next Frame	
Delay	0
From: VRG_Session	
Data Type	STRING
Load	✓
Load Text	Text (Text)
Save	✓
Save Text	Text (Text)
Session Object	Button
Session Data	Two
Value	
From: OnLoad	
Ignore On Null	✓
If Value	Cancel
Activate	
Deactivate	
Toogle	
Size	1
Element 0	VRG_Button – Icor

8.3 Other SessionData Objects

We preconfigured some frecuently used prefabs that need to save its data to the session inheritance from this class.

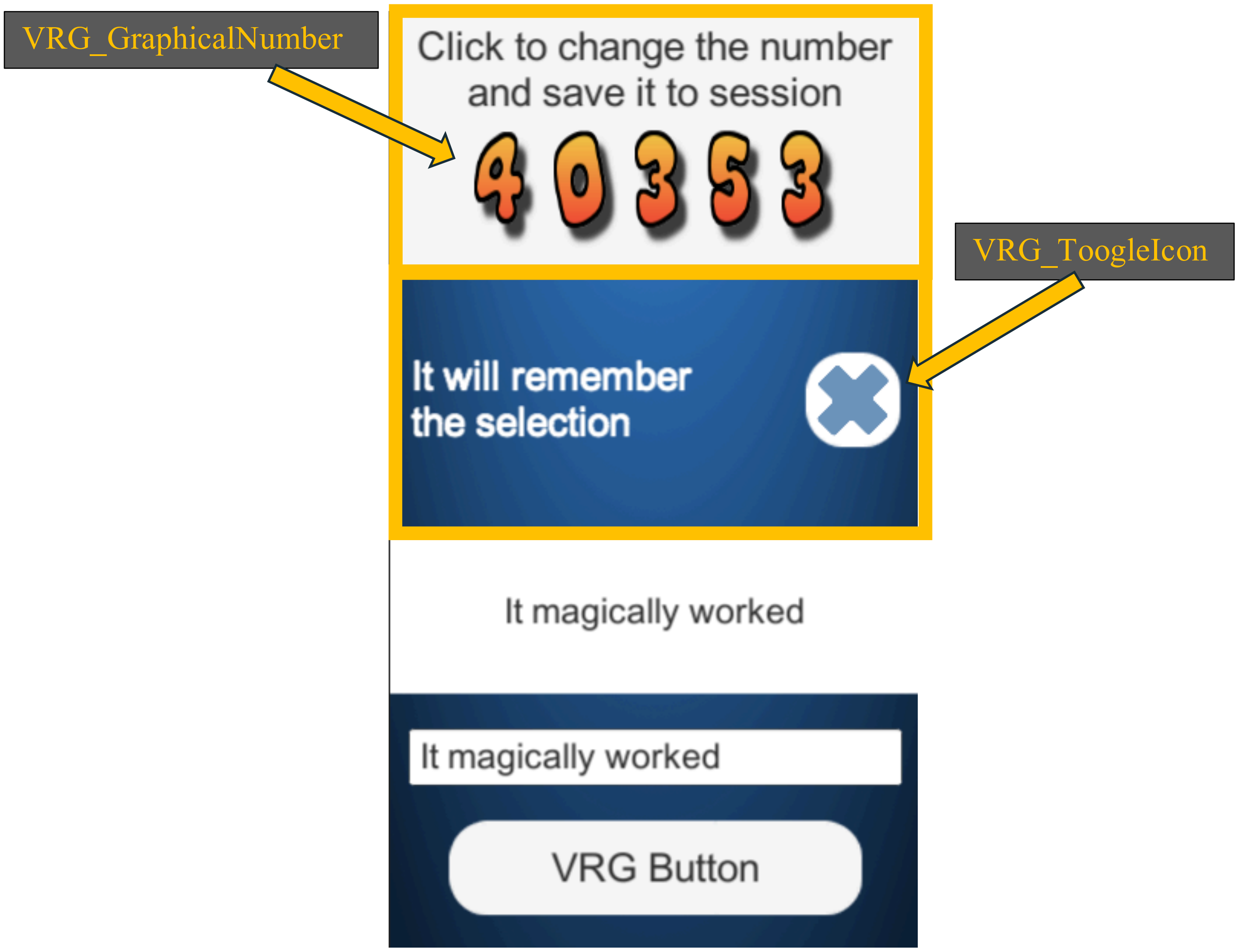
Scene sample: *Assets/_VrGamesDev/CORE/Scenes/06 VRG_SessionData UI*

VRG_GraphicalNumber: You can use this Graphical number to save records, coins, gems or whatever your player collect in your game, it doesn't require a Text UI to save or load its data. Just set the Session Object and th Session Data, and it will magically save it.

In the provided scene you can check how it is done.

VRG_ToogleIcon: You can use this element to save player preferences, or any option your player select in your game, It doesn't require a Text UI to save or load its data. Just set the Session Object and th Session Data, and it will magically save it.

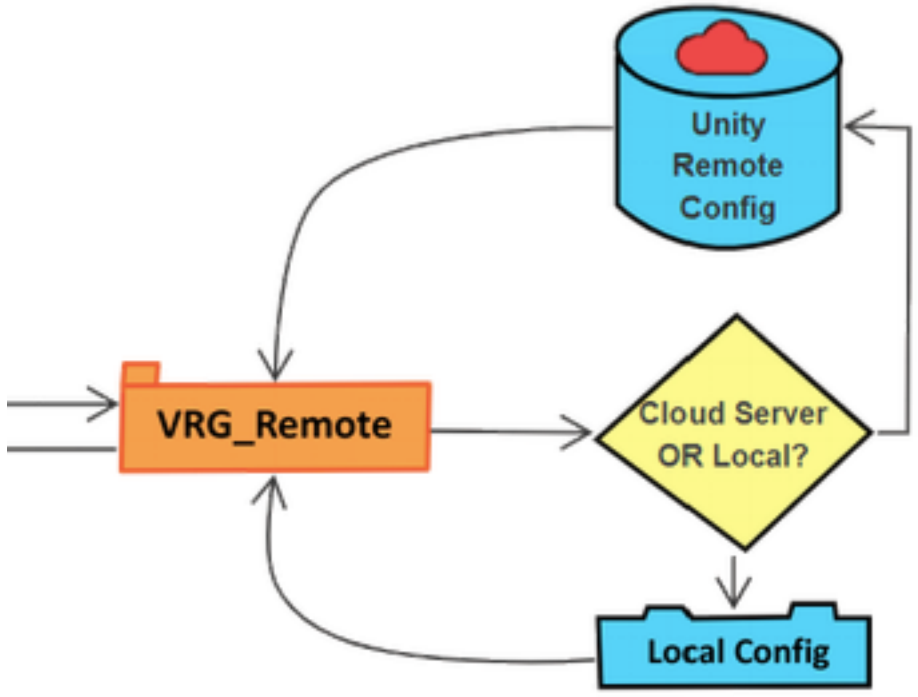
In the provided scene you can check how it is done.



9. VRG_REMOTE

The main setting configuration it allows to load from local settings instead of cloud remote config. Really useful to test and develop different configurations without altering the remote server.

Every VRG_Base class inheritance has access to the server data to custom its behavior, when you ask for a remote value, you can get the server data or the local provided to you by this class.



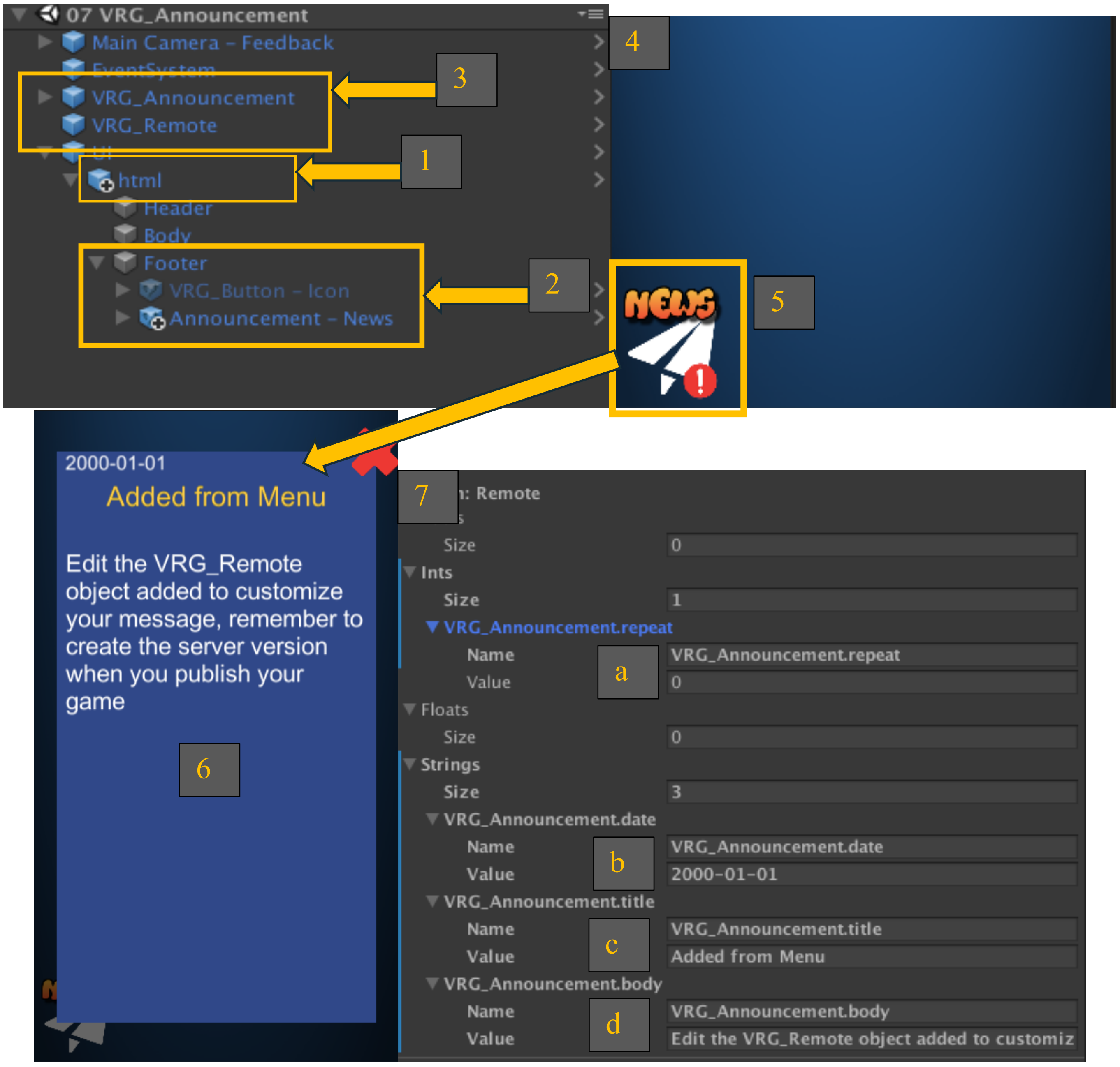
9.1 Announcement

We preconfigured a basic anncouncement, you can custom it to your game's visual style, it includes the icon to display it, if there are new notifications it will display a ! red mark, and a popo up with a customizable title, date and content, all this data is dinamically configured from the remote server, you can learn how to do it in the next chapter.

Scene sample**:** *Assets/_VrGamesDev/CORE/Scenes/07 VRG_Announcement*

Let's create an announcement popup that uses this Remote setting.

1) Add one ***HTML*** componentn
2) In the footer section of the ***html*** component, add a ***Tools->VR Games Dev -> UI -> Announcement - News*** component
3) Add a ***Tools->VR Games Dev -> UI -> Announcement - News*** component
4) Play the game
5) Click the News button



2000-01-01
Added from Menu

Edit the VRG_Remote object added to customize your message, remember to create the server version when you publish your game

```
n: Remote
  Size                          0
Ints
  Size                          1
  VRG_Announcement.repeat
    Name                        VRG_Announcement.repeat
    Value                       0
Floats
  Size                          0
Strings
  Size                          3
  VRG_Announcement.date
    Name                        VRG_Announcement.date
    Value                       2000-01-01
  VRG_Announcement.title
    Name                        VRG_Announcement.title
    Value                       Added from Menu
  VRG_Announcement.body
    Name                        VRG_Announcement.body
    Value                       Edit the VRG_Remote object added to customiz
```

6) The data you configured for this message, is displayed in a pop up, You can custom this info **IN THE LOCAL DATA** in the editor mode in the **VRG_Remote** gameObject.
7) There are 4 data you can custom:
   a) **VRG_Announcement.repeat** (int) = It determines how many times the message is considered "*new*", when the player opens the pop up, 0 for "always new"
   b) **VRG_Announcement.date (string) =** The date when this announcement was published
   c) **VRG_Announcement.title (string) =** The title of the announcement, it is labeled in yellow
   d) **VRG_Announcement.body (string) =** The body, here you paste all your news announcement.

10. USING UNITY REMOTE CONFIG INSTEAD OF LOCAL VRG_REMOTE

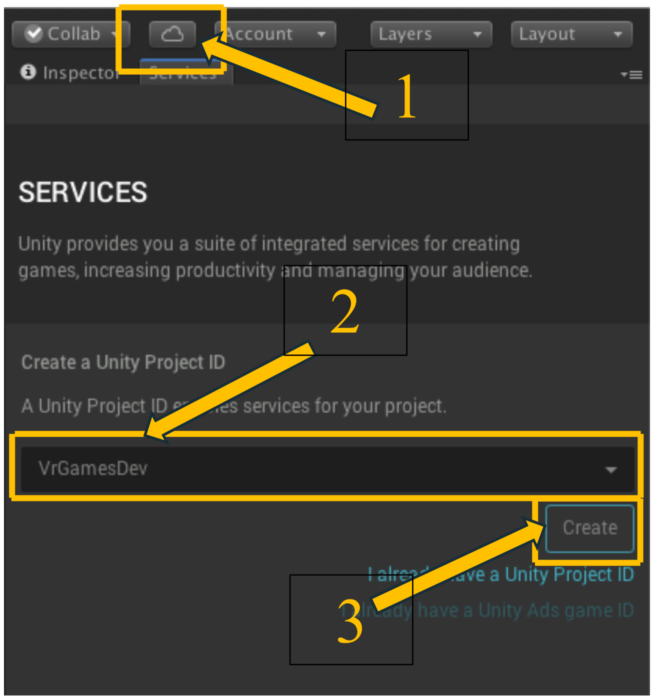
Remote config is a cloud service that lets you tune and customize your app over the air without requiring an update to your game. You can use rules to enable or disable features, change the difficulty of your game or run special events to target specific audiences. Unity manages delivery of your game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

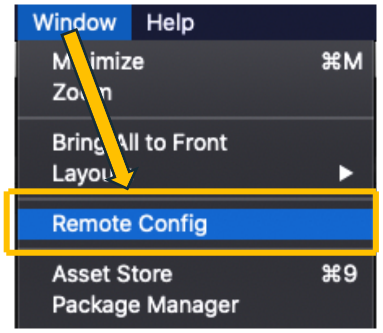
Open the log in the folder *(LogsLocal)*, read it, and rename it as "Full Verbose", we will modify it from the remote cloud server setting, let's configure it

10.2 Configure the Remote Config

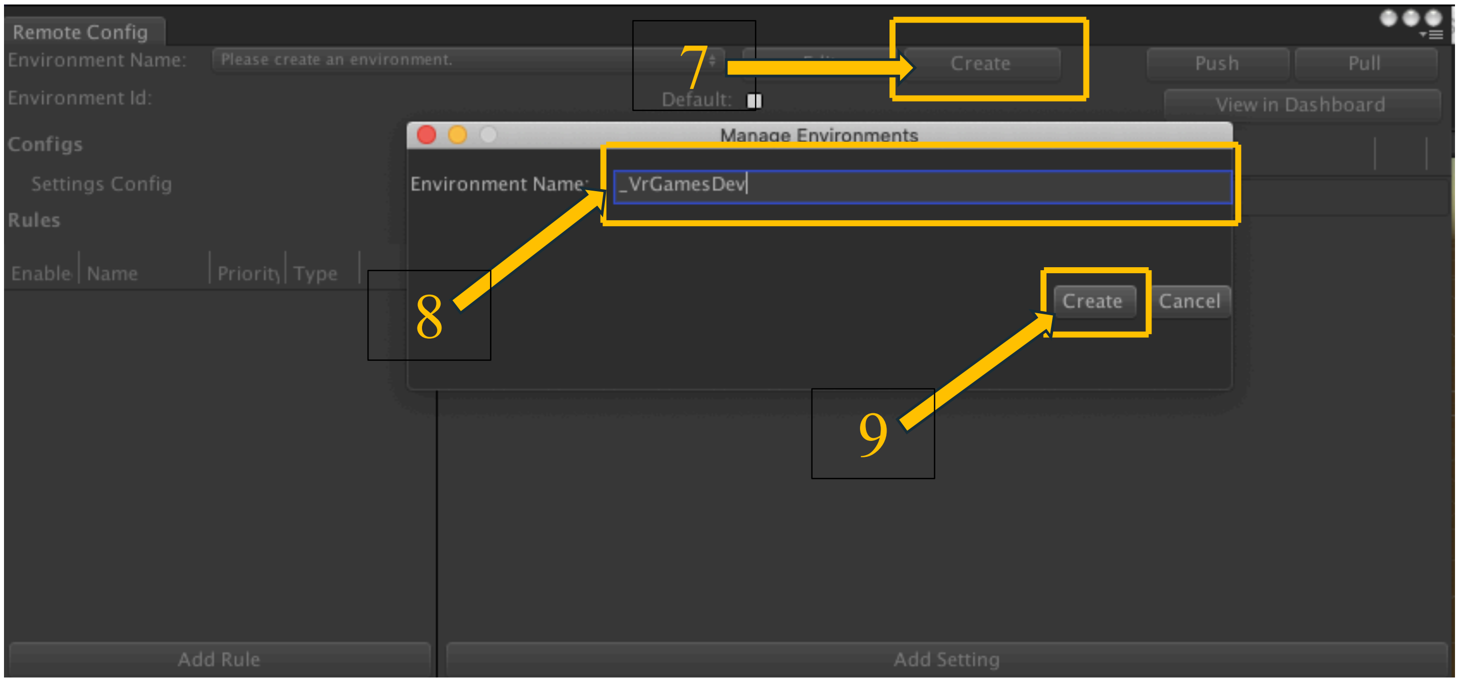
1) Now create a profile for your project, open the Unity Services, click the tiny cloud icon,
2) Select your company profile
3) Click the "Create" button


Collab
Account
Layers
Layout
Inspector
Services
1
SERVICES
Unity provides you a suite of integrated services for creating games, increasing productivity and managing your audience.
2
Create a Unity Project ID
A Unity Project ID enables services for your project.
VrGamesDev
Create
I already have a Unity Project ID
I already have a Unity Ads game ID
3

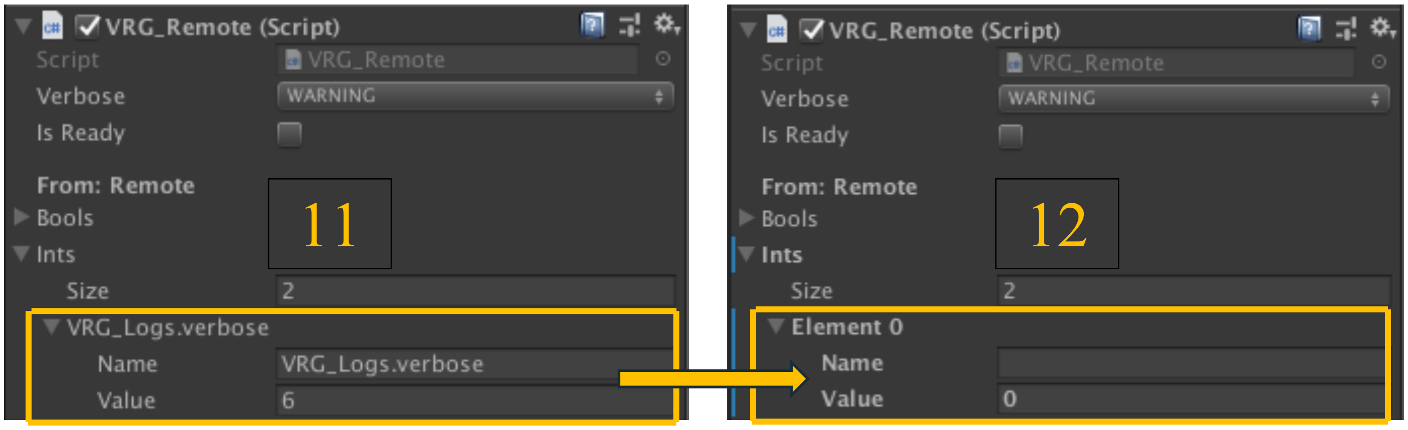
4) Be sure you Installed the Remote Config package as explained in **3.2 Unity Remote Config** section, you can read more about it here.
5) Go to the Window -> Remote Config



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

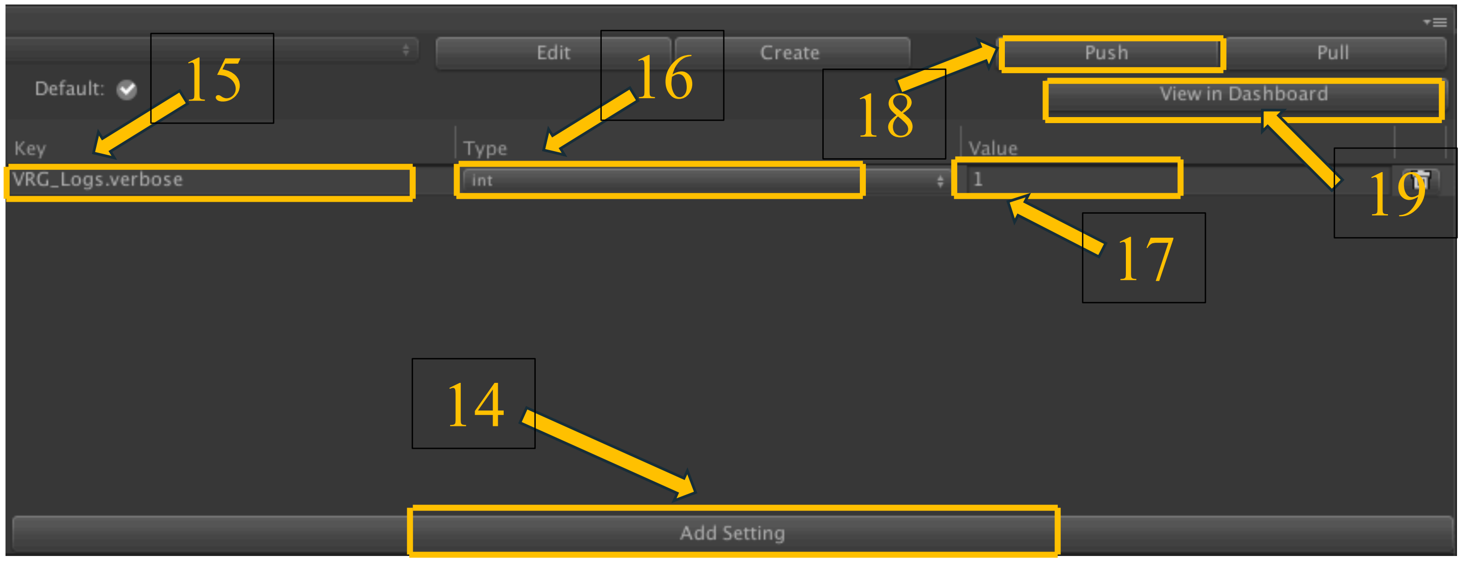


10) Now let's create a remote setting, take by example, the log verbose level of detail as explained in **6.1 Configure your VRG_Logs** section.
11) The setting ***VRG_Logs.verbose*** is set to 6 in the local VRG_Remote, which is the most detailed, let's change it to the less detailed, the level 1.
12)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



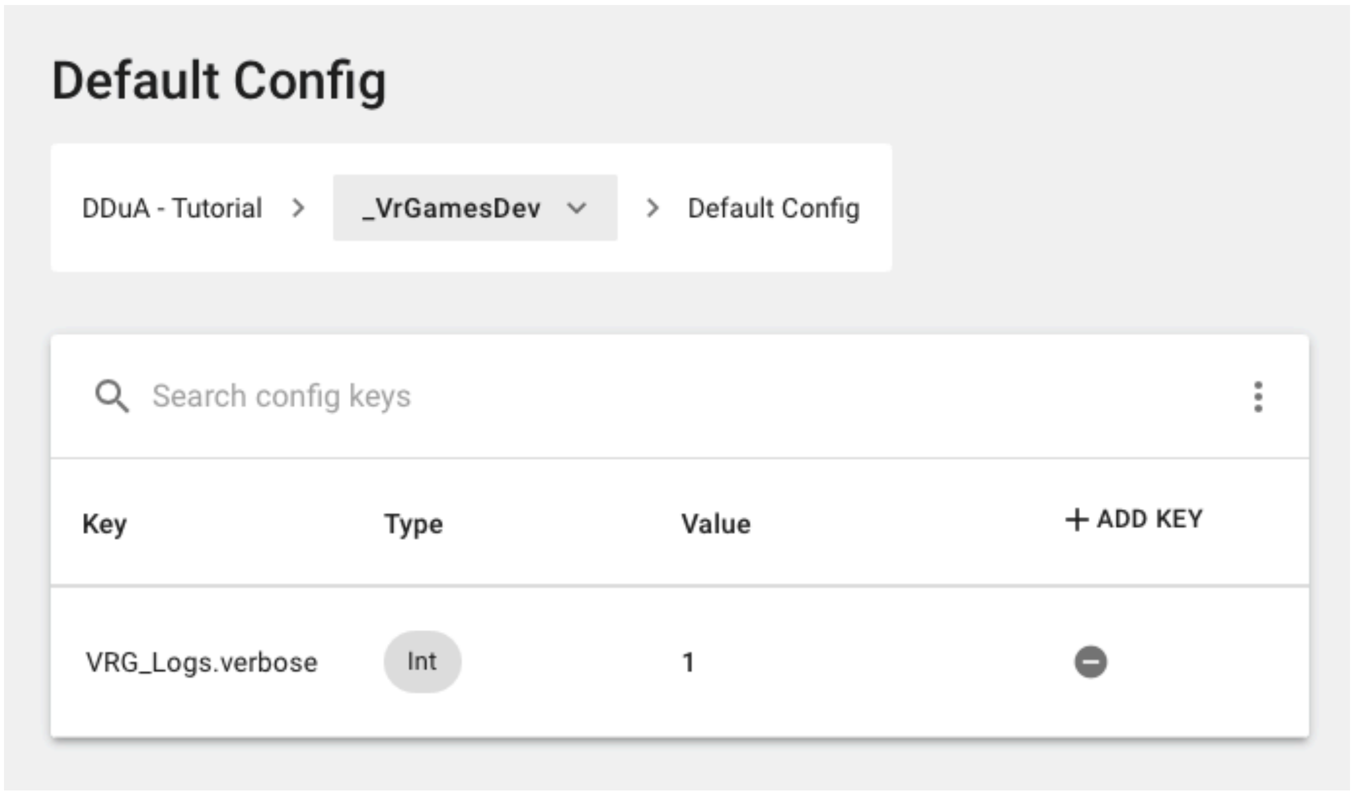
13) Play the game, and notice how the log wasn't created. Because the setting ***VRG_Logs.verbose*** was not found so the default value is 0, which means "Verbose empty"
14) To use the remote settings, click the "***Add Setting***" button
15) Name the key the same as you defined it in the ***VRG_Remote*** 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



Key	Type	Value
VRG_Logs.verbose	int	1

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



Key	Type	Value
VRG_Logs.verbose	Int	1

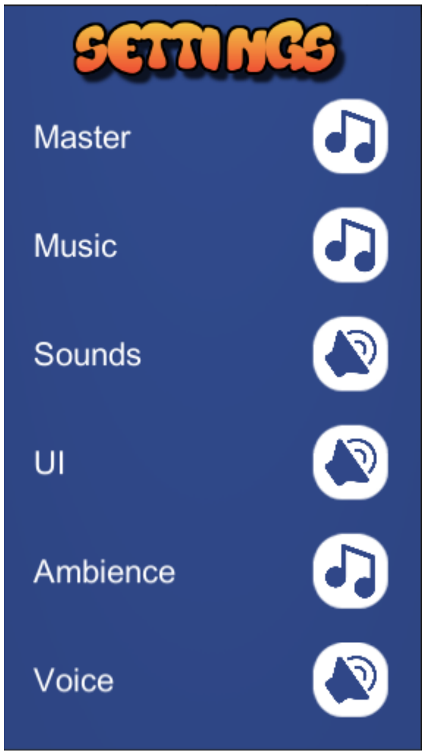
20) Play the game and Enjoy.
21) The new log file is configured according to the cloud remote server, with the less verbose configuration, it just logs errors, since we have no error right now, the log is empty.

11. SOUNDS AND MUSIC

There are modules included that handle the sound and music, it has preconfigured 5 channels

1) Music
2) Sounds (SFx)
3) UI
4) Ambience
5) Voice

You can turn them off or on there is a demo included in the following scene

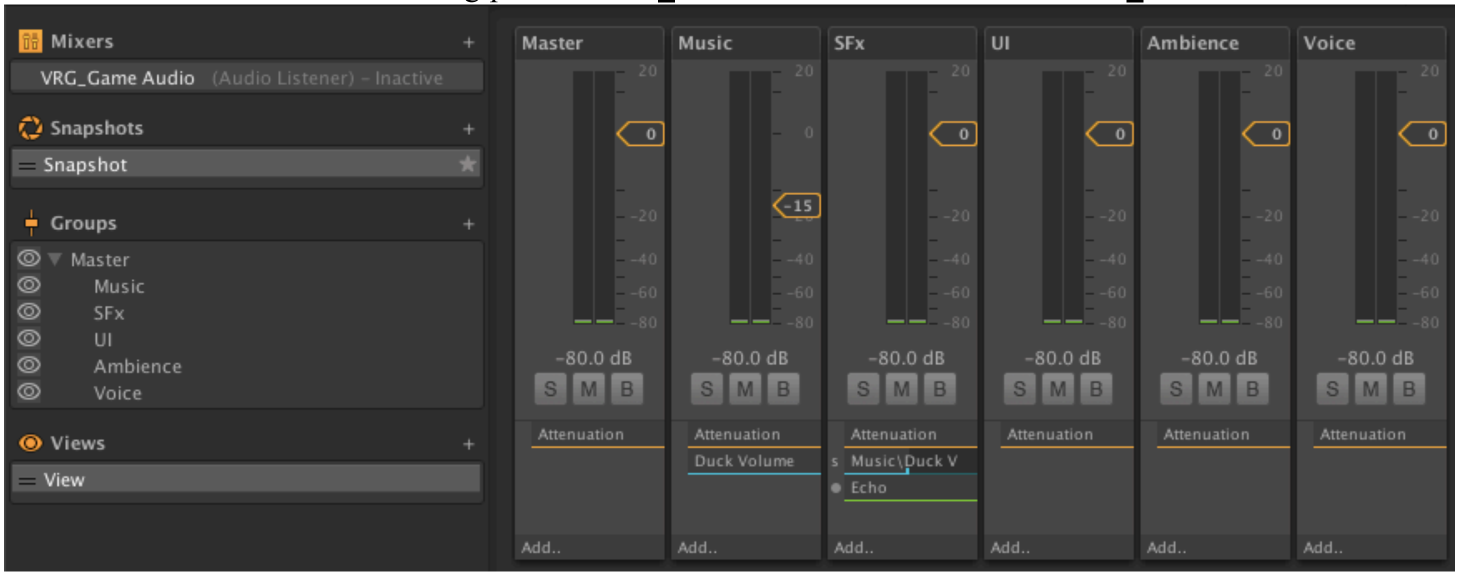


Scene sample: *Assets/_VrGamesDev/CORE/Scenes/08 Sounds and music*

Play the scene and open the audiomixer window and toogle the diferent channels to have a sense of the control you have over the game audio.

Documentation: You can read more about this prefab here https://docs.unity3d.com/2020.2/Documentation/Manual/AudioMixer.html

The audio mixer is in the following path: ***Assets/_VrGamesDev/CORE/Audio/VRG_Game Audio***



12. THANK YOU

Thanks a lot, to download my package, I hope you will enjoy it, and help you to get amazing apps and nice games.