Vr Games Dev

B.H.E.L.
Beautiful Html Enhanced Logs

March 2021

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
March 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

B.H.E.L is a plugin that allows you to configure your application to have an external log to debug your game, you can also see the logs into the Screen as a floating UI, or in CSV format for later analysis. These components are already pre-configured and ready to use them out of the box.

We use BHEL as 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 the remote config package
2) Add a Bhel object from ***Tools -> Vr Games Dev -> Bhel Menu***
3) Add a *VRG_Bhel_Log* Script to any object
4) Run the game
5) Check your new project folder *BHEL/<yourProjectName>.html* and csv
6) Check the sample scene 05

2.2 BHEL Components

BHEL has the following components a remote object, the VRG_Bhel main class and a log script.

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

- VRG_Bhel: The class that save the logs, it is the core of this package

- VRG_Bhel_Log: Script that allows you to send text to the logs on the OnEnable event.

3. REQUIREMENTS TO USE BHEL

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

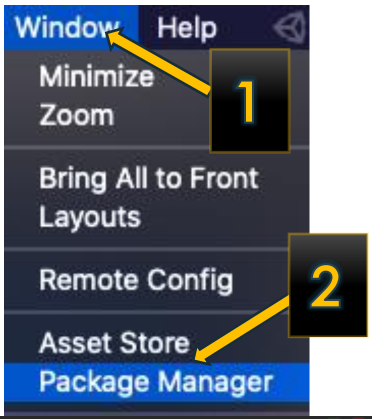
3.1 Unity Remote Config

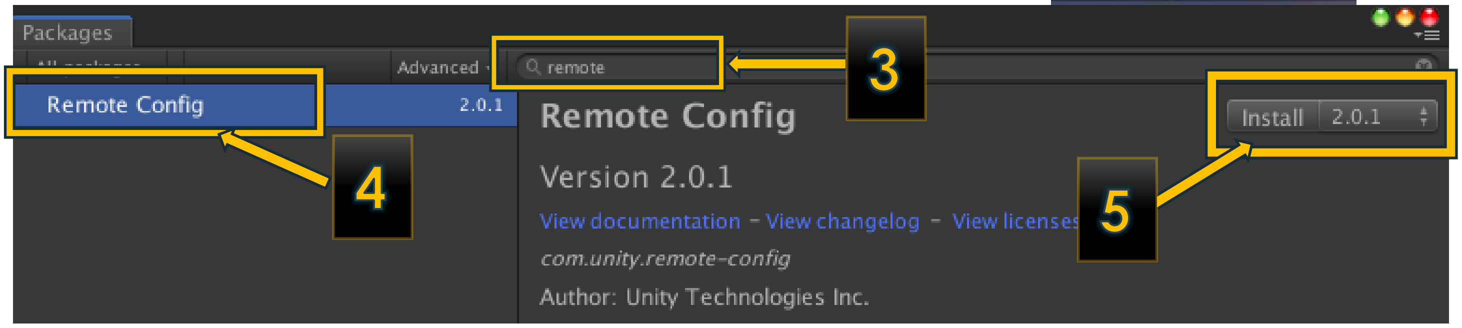
Unity Remote Config is a cloud service that allows you to tune your game design without deploying new versions of your application.

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:

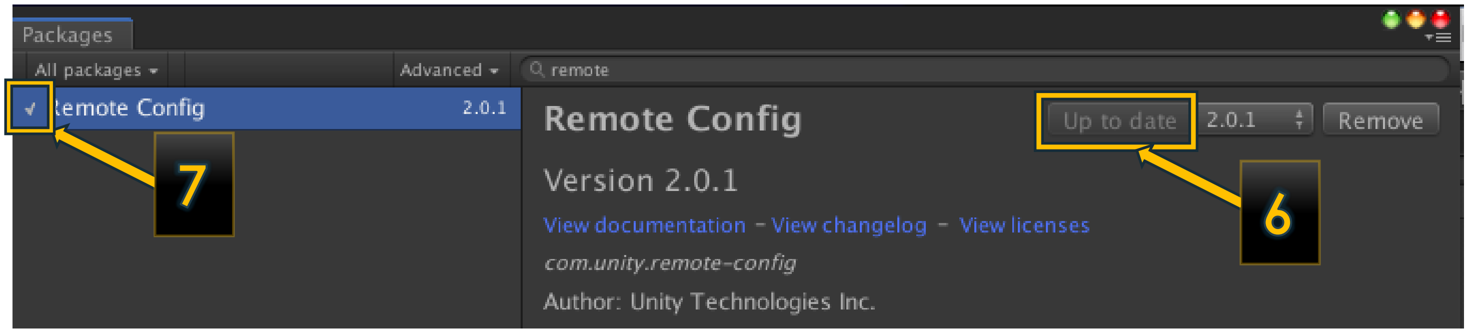
1) Open the Windows menu option
2) Select the option Package Manager
3) A new window will open, in the search box, Search for "Remote"
4) Select the package "Remote Config"
5) Click Install, the version that was tested with this package was the most recent (2.0.1) at the creation of this documentation.





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 check mark that indicates it was successfully installed.



Remote Config
Version 2.0.1
com.unity.remote-config
Author: Unity Technologies Inc.

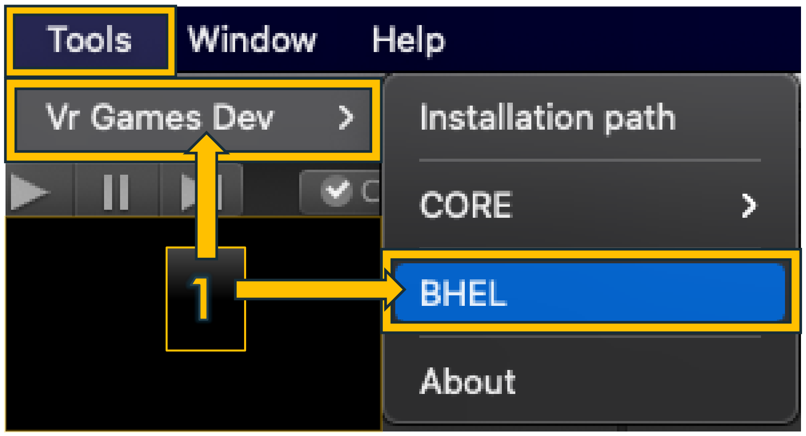
8) Congratulations, it's fully installed we recommend you read the official documentation to understand the power of this module.

4. MENU: TOOLS -> VR_GAMES_DEV-> BHEL

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

Documentation: You can read more about this module here:
https://www.vrgamesdev.com/api/#VrGamesDev.ENUM_Verbose

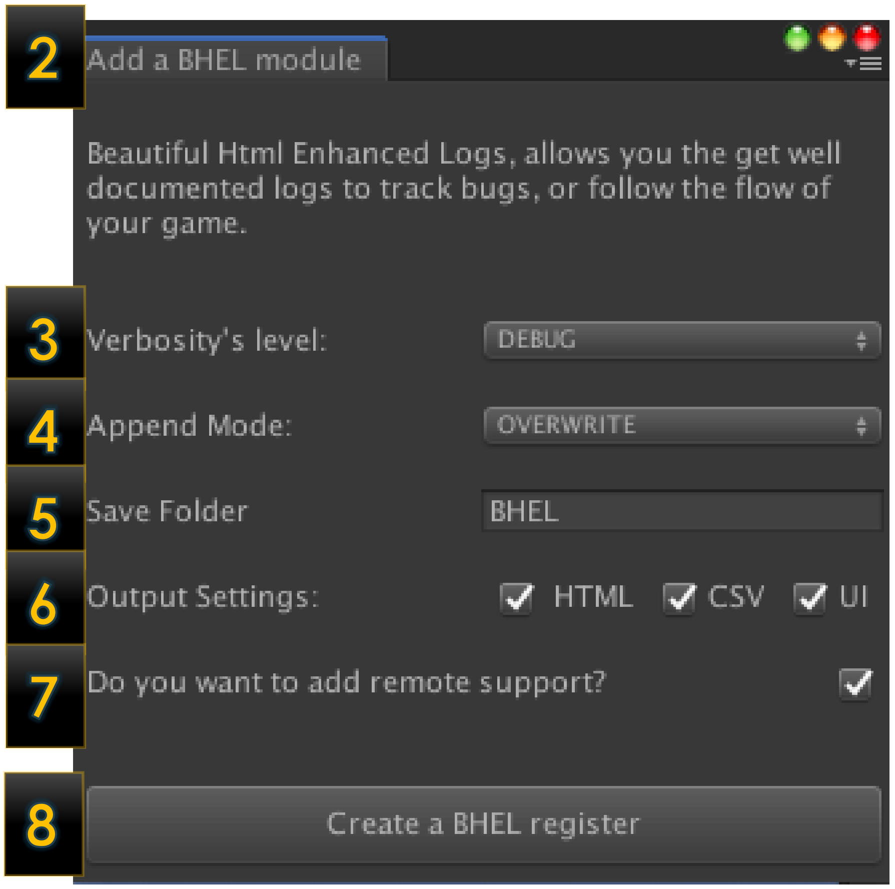
1) Select the menu option Tools -> VR Games Dev -> BHEL
2) You will get a pop-up window to configure your BHEL installation. After you select the options, the prefabs needed will be added to the scene.



3) Verbosity Level: You can select from 6 different levels of verbosity. Select the menu option Tools -> VR Games Dev -> BHEL

Value	Enum	Description
0	NONE	ENUM_Verbose.NONE = Silence, NONE is sent to the editor log window
1	ERROR	ENUM_Verbose.ERROR = When you need to show an error to the user and record a posible failure
2	WARNING	ENUM_Verbose.WARNING = Something was cheesy, if something is wacky, strange, or unexpected
3	LOGS	ENUM_Verbose.LOGS = To track some basic information, or to send logs to BHEL
4	DEBUG	ENUM_Verbose.INFO = More information, ins and outs, flow of the game, useful for debug using asynchronous activities
5	STATUS	ENUM_Verbose.STATUS = The most verbose, it basically shows EVERYTHING!!!

4) Append mode: You have three append modes:
   - **Overwrite**: It save the logs in the same file and you will lose previous run
   - **Append**: It adds the new logs at the end of the file
   - **Diferent Files**: It will save the logs in a file with the timestamp in the title
5) Save Folder: Where the logs will be saved
6) Output Settings: The format to save the logs, html, CSV or directly as floating UI
7) You can modify this prefab from the remote settings server, add support to this functionality
8) Click to create the BHEL module with the settings configured.



Beautiful Html Enhanced Logs, allows you the get well documented logs to track bugs, or follow the flow of your game.

Verbosity's level: DEBUG

Append Mode: OVERWRITE

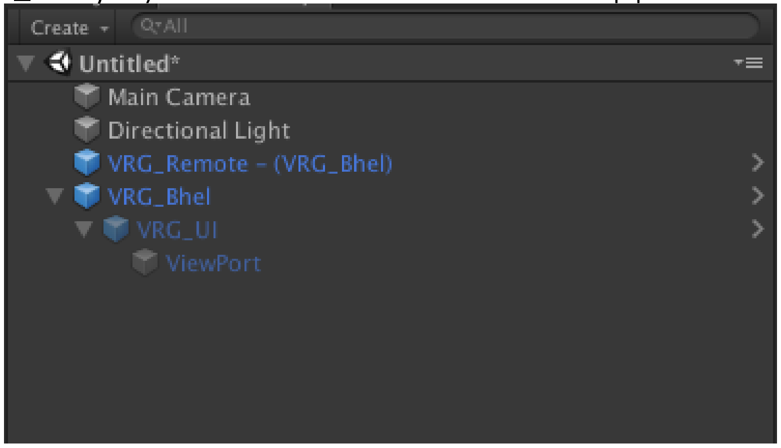
Save Folder BHEL

Output Settings: ✓ HTML ✓ CSV ✓ UI

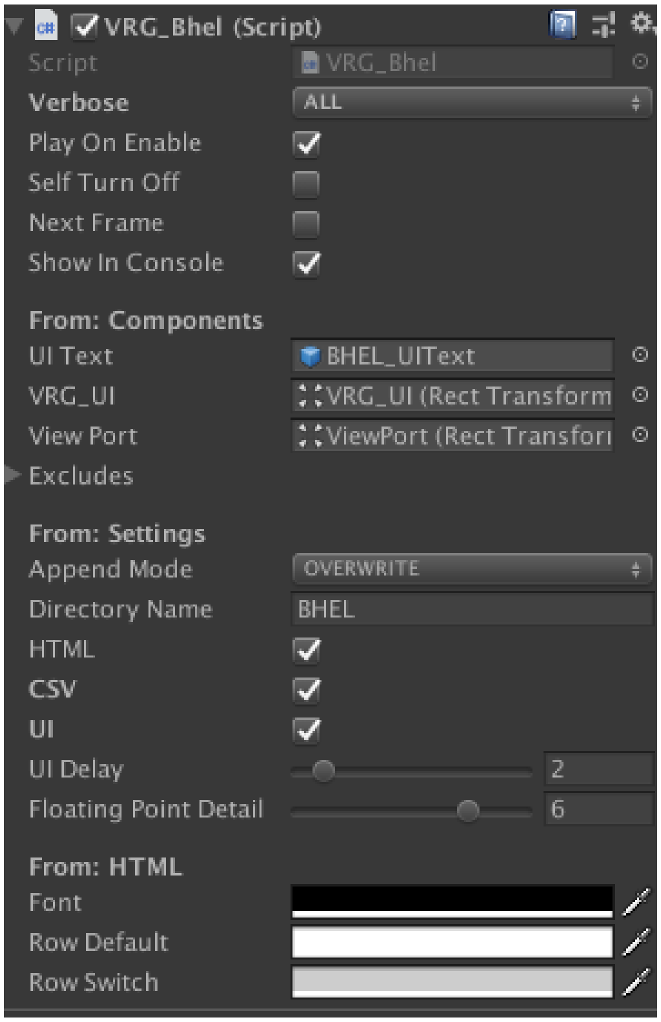
Do you want to add remote support? ✓

5. VRG_BHEL IN THE SCENE

Once you configured your BHEL you will get in the scene one or two prefabs, a VRG_Bhel and a VRG_Remote - (VRG_Bhel) if you decide to have remote support.



```
Create ▾   Q▾All
▼ Untitled*
    Main Camera
    Directional Light
    VRG_Remote - (VRG_Bhel)
  ▼ VRG_Bhel
    ▼ VRG_UI
        ViewPort
```



```
VRG_Bhel (Script)
Script                  VRG_Bhel
Verbose                 ALL
Play On Enable          ✓
Self Turn Off           ☐
Next Frame              ☐
Show In Console         ✓

From: Components
UI Text                 BHEL_UIText
VRG_UI                  VRG_UI (Rect Transform
View Port               ViewPort (Rect Transfor
Excludes

From: Settings
Append Mode             OVERWRITE
Directory Name          BHEL
HTML                    ✓
CSV                     ✓
UI                      ✓
UI Delay                2
Floating Point Detail   6

From: HTML
Font
Row Default
Row Switch
```

From: Components
The gameObjects needed to send the information to the UI and what functions to exclude from the logs.

From: Settings
These settings are the one you configured in the previous popup window. You can also set the time the UI messages stay in the screen, with the variable "UI Delay", by default is two seconds long, and the floating-point detail, by default is 6

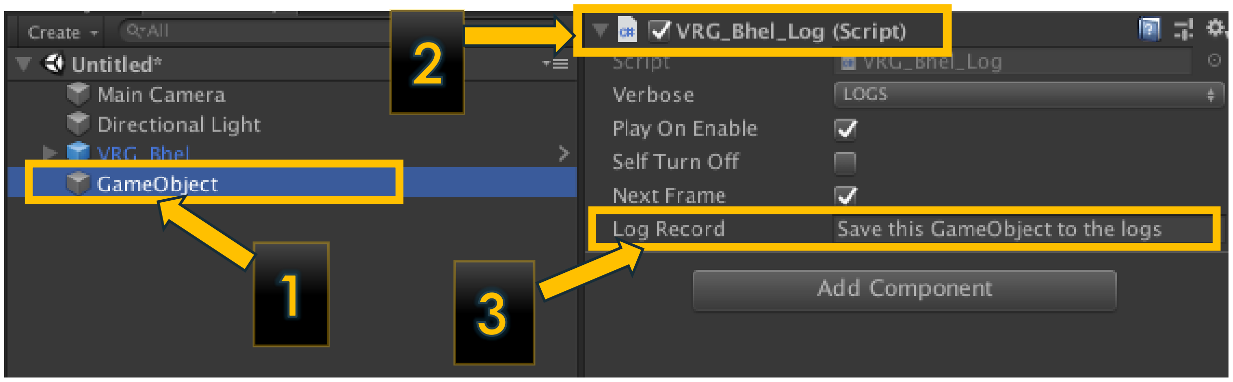
From: HTML
You can modify the color of the HTML report., the font, the default and the switch rows.

6. HOW TO ADD RECORDS TO THE LOGS

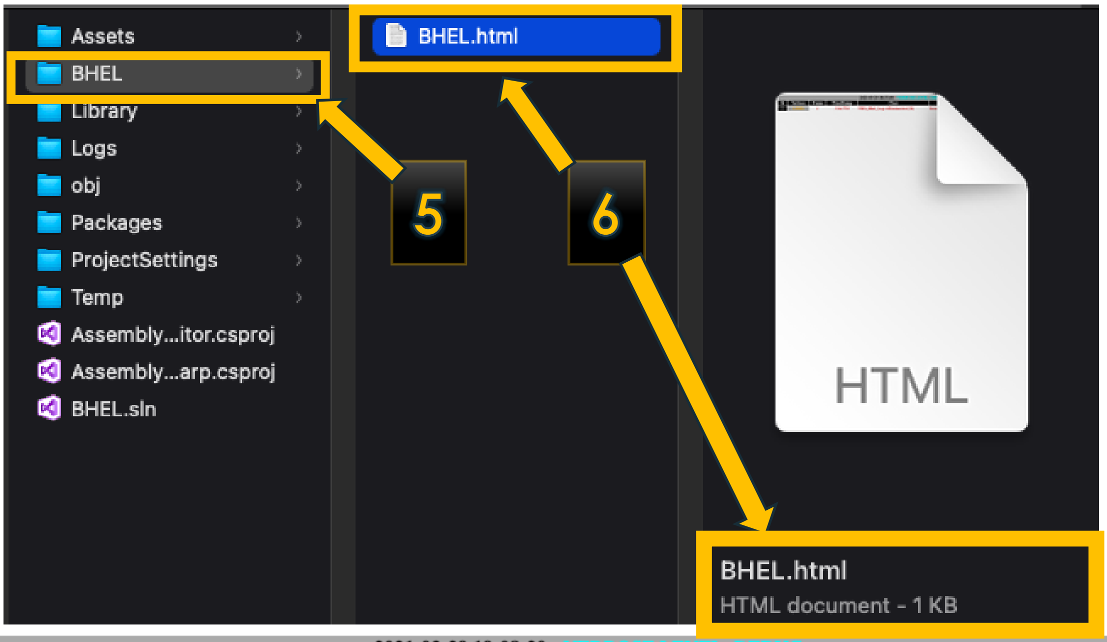
When you add the VRG_Bhel singleton object, you can add records to the logs in 3 diferent ways, you can add the script class VRG_Bhel_Log, inhertance from the VRG_Base or call the Singleton directly.

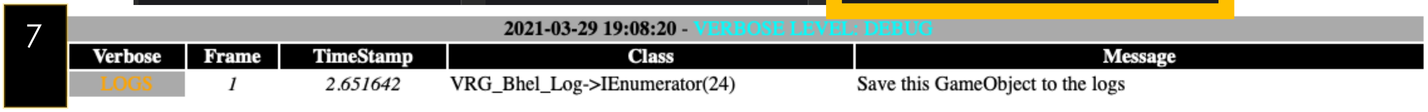
6.1 Add a script to a gameObject

1) Add a gameObject to your scene
2) Select it and add a script named "*VRG_Bhel_Log*"
3) Type the text you want to save in the "*Log Record*" field
4) Run the game
5) Check the folder BHEL
6) Double click the file *<projectName>.html*
7) You have an enhanced HTML Log



Verbose: LOGS
Play On Enable: ✓
Self Turn Off:
Next Frame: ✓
Log Record: Save this GameObject to the logs





2021-03-29 19:08:20 - VERBOSE LEVEL: DEBUG

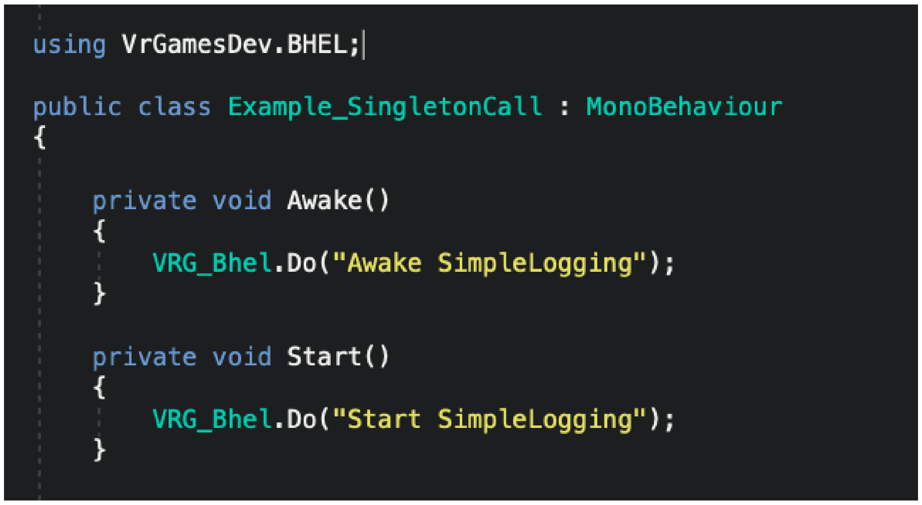
Verbose	Frame	TimeStamp	Class	Message
LOGS	1	2.651642	VRG_Bhel_Log->IEnumerator(24)	Save this GameObject to the logs

6.2 Singleton VRG_Bhel.Do()

You can call the singleton pattern class directly, with the function Do ()

Documentation: You can read more about this here:
Assets/_VrGamesDev/BHEL/Scripts/Examples/Example_SingletonCall.cs

1) Add the Using VrGamesDev.BHEL header
2) Call the Do function with your text to log = VRG_Bhel.Do("Start SimpleLogging");



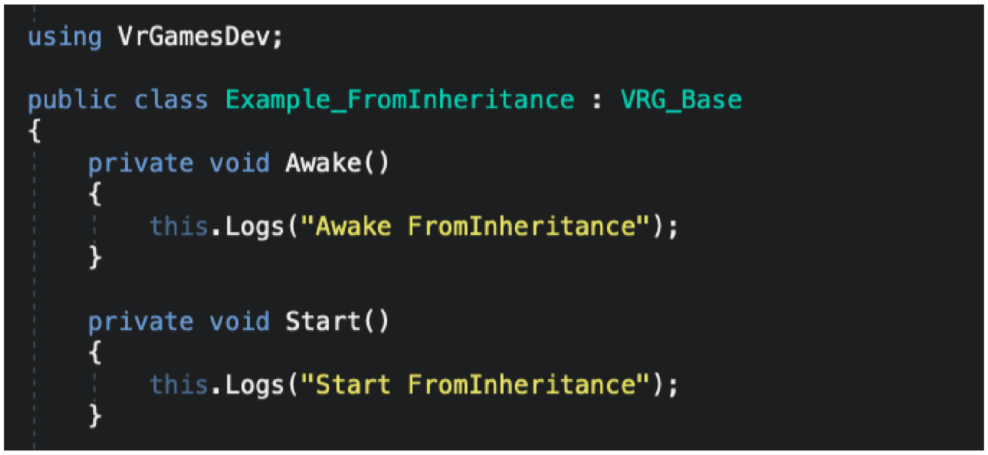
```
using VrGamesDev.BHEL;

public class Example_SingletonCall : MonoBehaviour
{

    private void Awake()
    {
        VRG_Bhel.Do("Awake SimpleLogging");
    }

    private void Start()
    {
        VRG_Bhel.Do("Start SimpleLogging");
    }
```

6.3 Inheritance from VRG_Base.Logs()

You can use inheritance from the main class of the VRGamesDev framework and use the Logs() function.

Documentation: You can read more about this here:
Assets/_VrGamesDev/BHEL/Scripts/Examples/Example_FromInheritance.cs

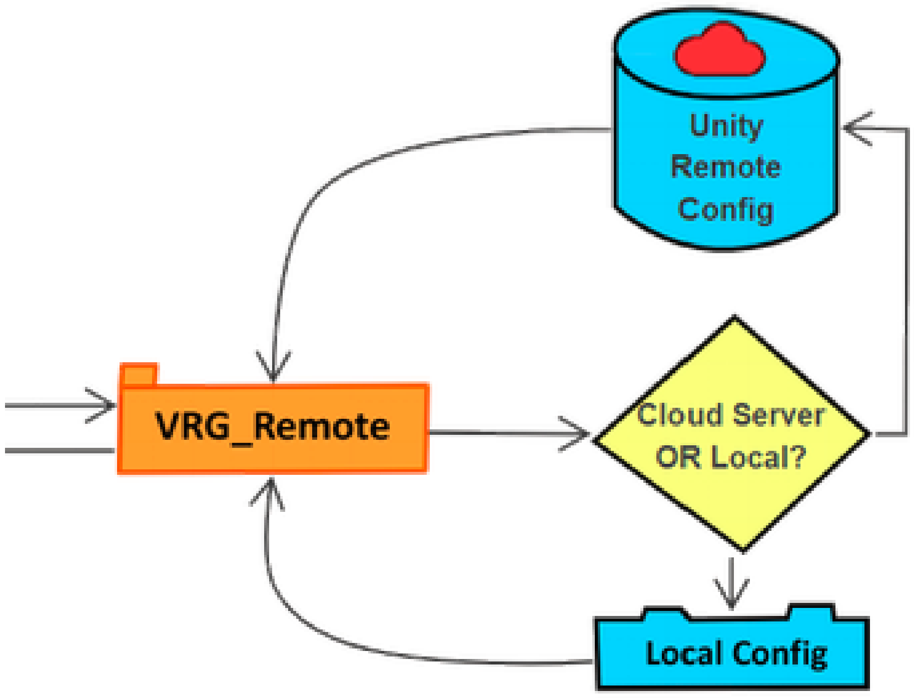
1) Add the Using VrGamesDev header
2) Inherit from the VRG_Base, this class inherit from MonoBehavior
3) Call anywhere you want the function Logs("message to log");



```
using VrGamesDev;

public class Example_FromInheritance : VRG_Base
{
    private void Awake()
    {
        this.Logs("Awake FromInheritance");
    }

    private void Start()
    {
        this.Logs("Start FromInheritance");
    }
```

7. VRG_REMOTE

The VRG_Remote class and prefab, allows you to load settings for your gameobjects, it loads them from local settings or the cloud remote config server. Really useful to test and develop different configurations without altering the remote server, when you ask for a value, you can get the server data or the local data by this class.



7.1 Description of the class

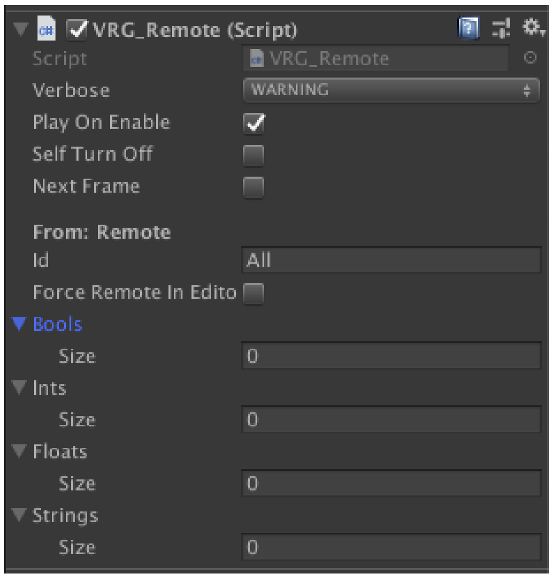
You can add the script "VRG_Remote" to any gameObject and it will connect to the remote server and have all the data available.

From: VRG_Base
Verbose, play on enable, self turn off and next frame are data inherited from the VRG_Base class, these parameters are not used in this class

From: Remote
Id: The name of the module that will affect its parameters, to this module is: ***VRG_Bhel***

Force remote in editor: Use the server data in the editor, this overwrite the local data.

Bools, Ints, Floats, strings: Arrays that configure the data used in the prefabs, classes and GameObjects

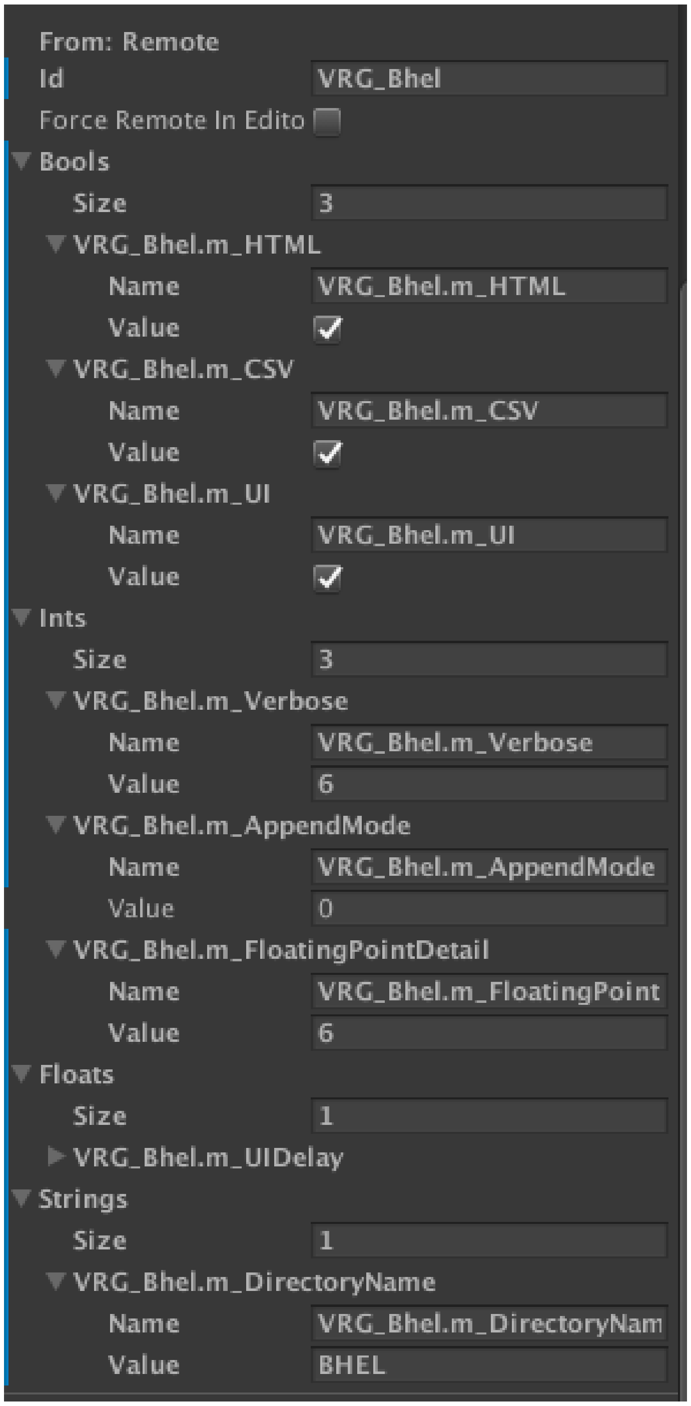


```
VRG_Remote (Script)
Script                     VRG_Remote
Verbose                    WARNING
Play On Enable             ✓
Self Turn Off              ☐
Next Frame                 ☐

From: Remote
Id                         All
Force Remote In Edito      ☐
▼ Bools
    Size                   0
▼ Ints
    Size                   0
▼ Floats
    Size                   0
▼ Strings
    Size                   0
```

7.2 Remote data for BHEL

When you add the remote component from the tools, you will get 8 data configurable in the VRG_Bhel object, every data will modify the proper setting into the VRG_Bhel class. **THESE SETTINGS overwrite the default settings in the class**

1) ***Id;*** VRG_Bhel, the name of the class that will use this data
2) ***Bools***; VRG_Bhel.m_HTML
3) ***Bools***; VRG_Bhel.m_CSV
4) ***Bools***; VRG_Bhel.m_UI
5) ***Ints;*** VRG_Bhel.m_Verbose
6) ***Ints;*** VRG_Bhel.m_AppendMode
7) ***Ints;*** VRG_Bhel.m_FloatingPointDetail
8) ***Floats***; VRG_Bhel.m_UIDelay
9) ***Strings***; VRG_Bhel.m_DirectoryName



```
From: Remote
Id                          VRG_Bhel
Force Remote In Edito       ☐
▼ Bools
   Size                     3
  ▼ VRG_Bhel.m_HTML
     Name                   VRG_Bhel.m_HTML
     Value                  ✓
  ▼ VRG_Bhel.m_CSV
     Name                   VRG_Bhel.m_CSV
     Value                  ✓
  ▼ VRG_Bhel.m_UI
     Name                   VRG_Bhel.m_UI
     Value                  ✓
▼ Ints
   Size                     3
  ▼ VRG_Bhel.m_Verbose
     Name                   VRG_Bhel.m_Verbose
     Value                  6
  ▼ VRG_Bhel.m_AppendMode
     Name                   VRG_Bhel.m_AppendMode
     Value                  0
  ▼ VRG_Bhel.m_FloatingPointDetail
     Name                   VRG_Bhel.m_FloatingPoint
     Value                  6
▼ Floats
   Size                     1
  ▶ VRG_Bhel.m_UIDelay
▼ Strings
   Size                     1
  ▼ VRG_Bhel.m_DirectoryName
     Name                   VRG_Bhel.m_DirectoryNam
     Value                  BHEL
```

8. USING UNITY REMOTE CONFIG INSTEAD OF LOCAL VRG_REMOTE

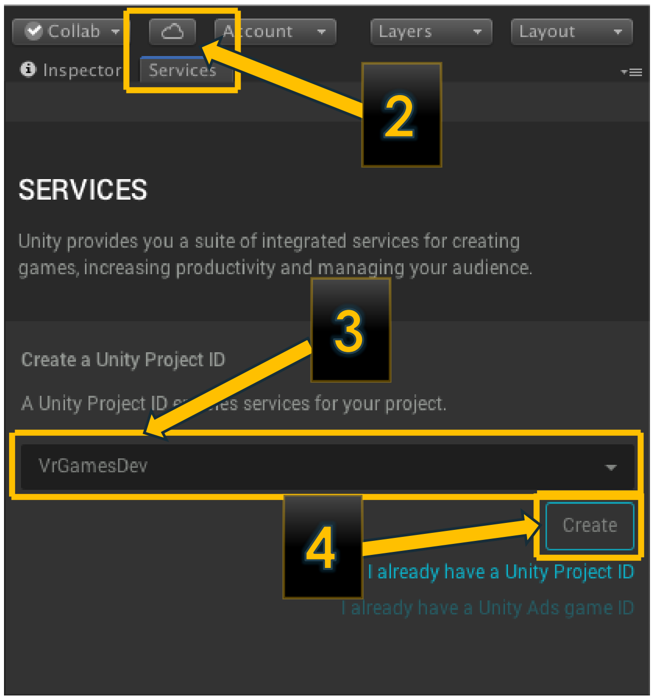
Remote config is a cloud service that lets you tune and customize your app over the air without requiring an update to your game. You can use rules to enable or disable features, change the difficulty of your game or run special events to target specific audiences. Unity manages delivery of your game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

Documentation: You can read more about this module here:
https://docs.unity3d.com/Packages/com.unity.remote-config@1.2/manual/ConfiguringYourProject.html
https://docs.unity3d.com/Manual/SettingUpProjectServices.html

8.1 Configure the Remote Config

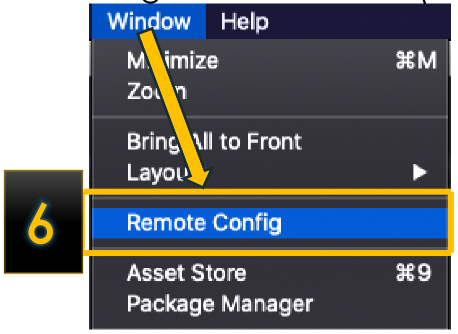
You need a profile in the unity services and a remote config server. To get started with Unity remote Config, you must first link your project to a Unity cloud services. The simplest way is as follows.

1) Create a new project
2) To open the Unity Services, click the tiny cloud icon or
3) Select your company profile
4) Click the "Create" button, and wait some seconds.

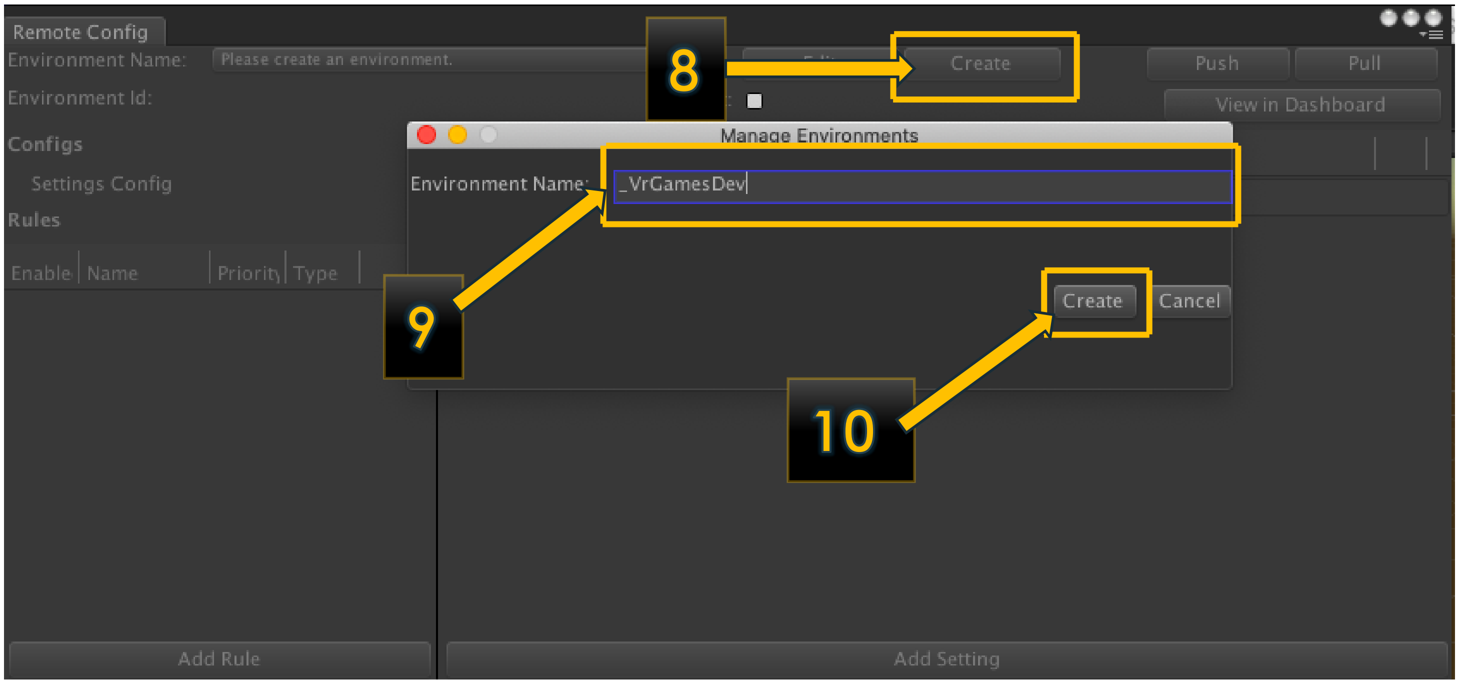

Collab
Account
Layers
Layout
Inspector
Services
2
SERVICES
Unity provides you a suite of integrated services for creating games, increasing productivity and managing your audience.
3
Create a Unity Project ID
A Unity Project ID enables services for your project.
VrGamesDev
4
Create
I already have a Unity Project ID
I already have a Unity Ads game ID

5) Be sure you Installed the Remote Config package as explained in 3.1 Unity Remote Config section

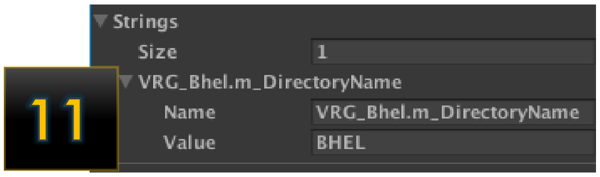
6) Go to the Window -> Remote Config, a new window (Remote Config) will popup



7) 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,
8) Click the "Create" button, and a new window will open *"Manage Environments"*
9) In the text field … name your environment, something cute, nice, creative, useful, handsome and perfect, like "_VrGamesDev"
10) Click the second "Create" button

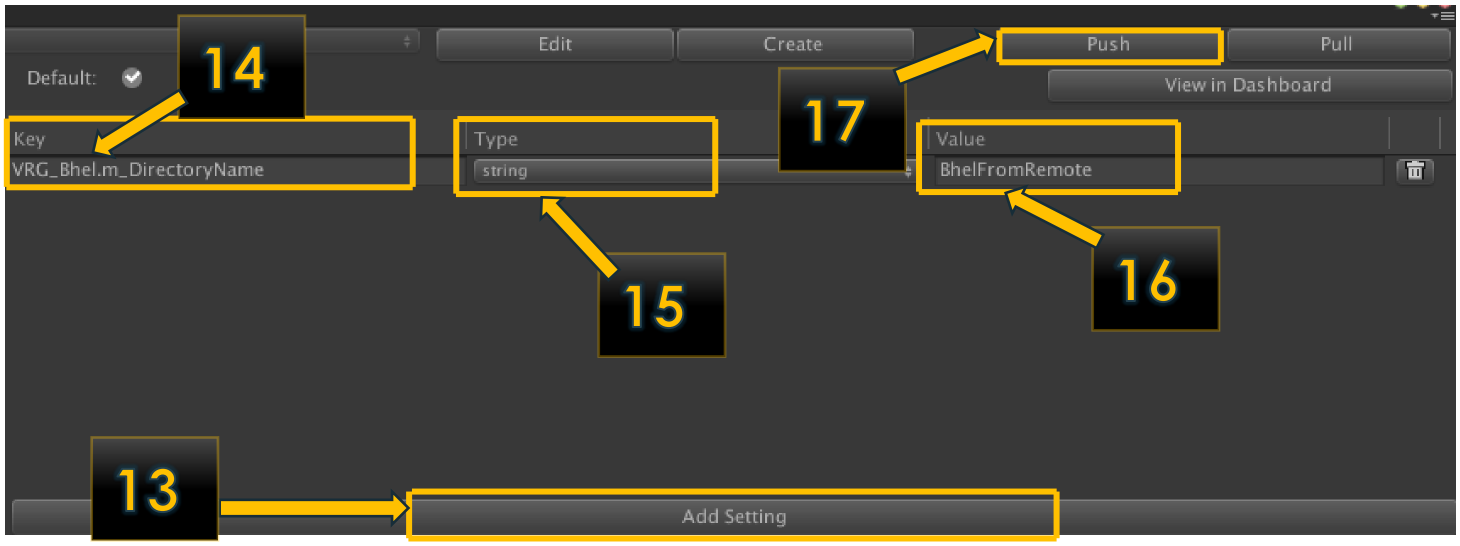


11) Create a VRG_Bhel object with support for remote config, explained in 4. MENU: TOOLS -> VR_GAMES_DEV-> BHEL, by default the folder to save is "BHEL", in the Strings section of the VRG_Remote added



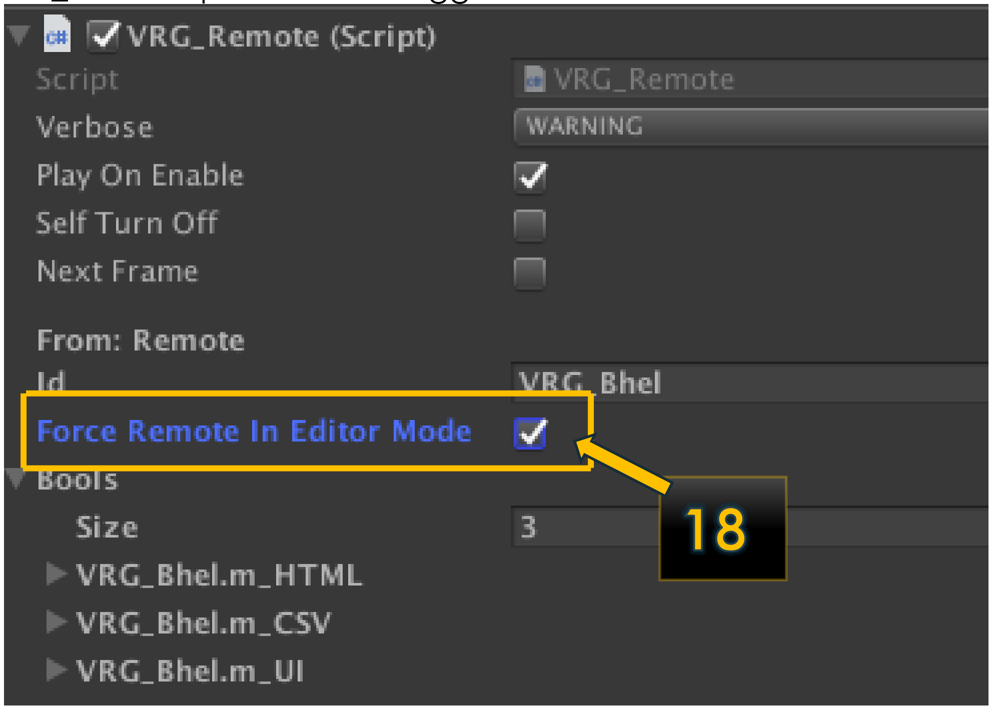
▼ Strings	
Size	1
▼ VRG_Bhel.m_DirectoryName	
Name	VRG_Bhel.m_DirectoryName
Value	BHEL

12) Play the game, and find the html and csv logs in the folder <Project>/BHEL/Bhel.html and <Project>/BHEL/Bhel.csv
13) Let's change the folder from the remote server, in the remote config, click the "Add Setting" button at the bottom of the RemoteConfig Window
14) Name the key the same with the same name as defined it in the VRG_Remote prefab, "*VRG_Bhel.m_DirectoryName*"
15) Declare the same type as defined, in this case, the type Is *string*.
16) Now set the desired value, in this case, for example "*BhelFromRemote*".
17) Finally Push it to the server



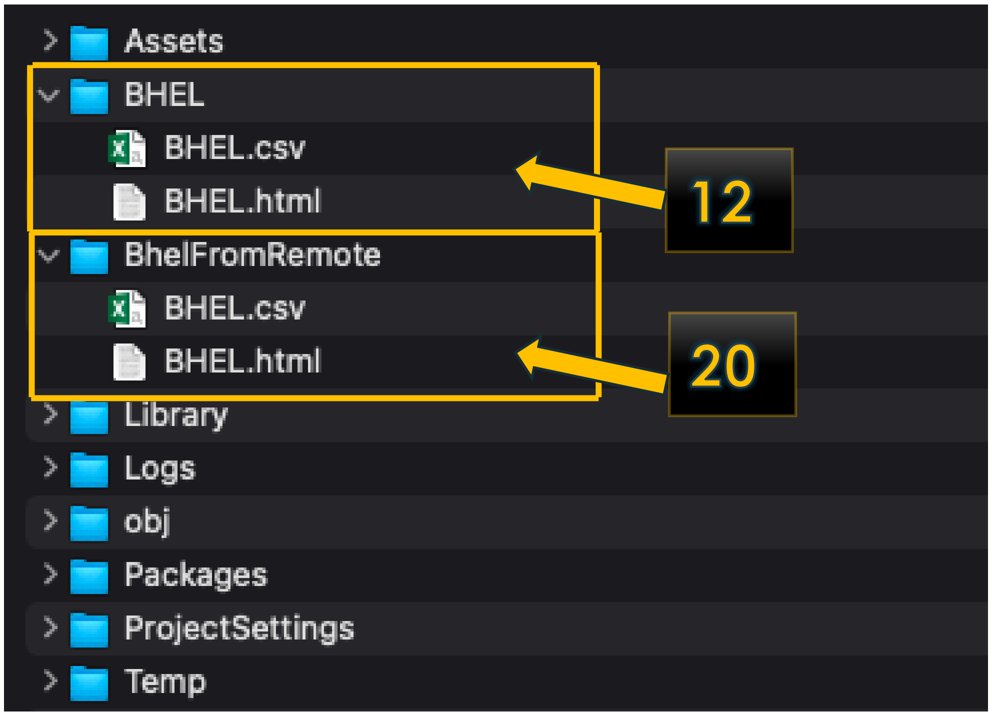
Key	Type	Value
VRG_Bhel.m_DirectoryName	string	BhelFromRemote

18) Select the VRG_Remote prefab, and toggle on the "*Force Remote In Editor Mode*" option



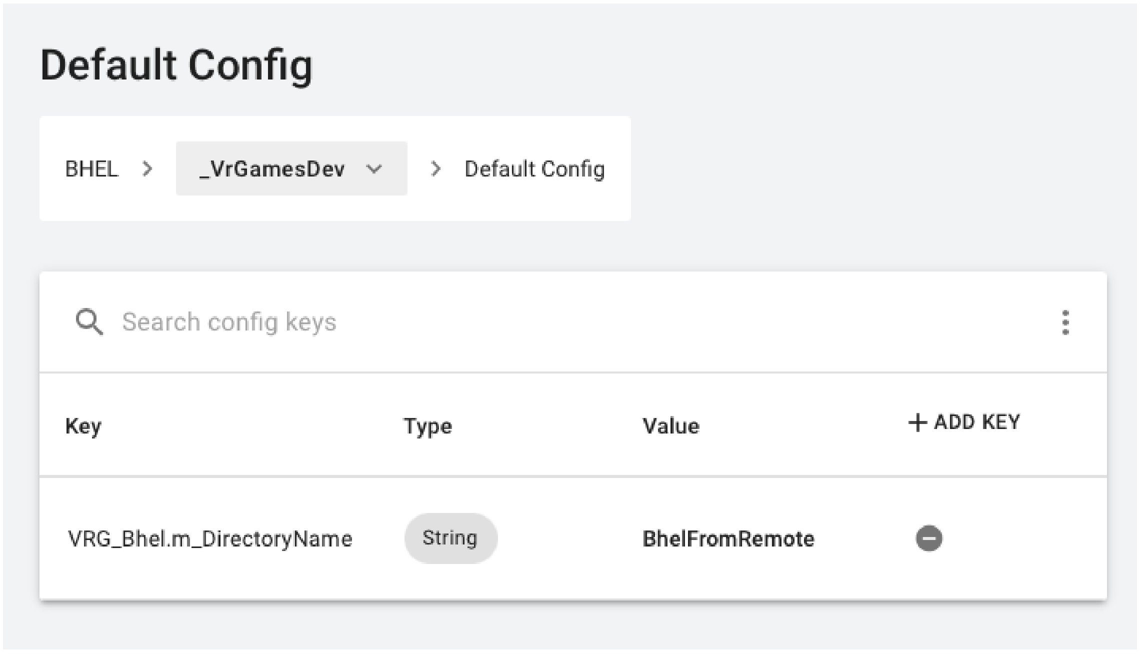
Script	VRG_Remote
Verbose	WARNING
Play On Enable	✓
Self Turn Off	
Next Frame	
From: Remote	
Id	VRG_Bhel
Force Remote In Editor Mode	✓
Bools	
Size	3
VRG_Bhel.m_HTML	
VRG_Bhel.m_CSV	
VRG_Bhel.m_UI	

19) Run the game and go to your <project> folder, you will have a new folder named "*BhelFromRemote*"
20) The new logs are in the folder, according to your remote configuration



8.2 Dashboard

You can see this same setting in the server, click the "View in Dashboard" button, in the remote config window.



Default Config

BHEL > _VrGamesDev > Default Config

Key	Type	Value
VRG_Bhel.m_DirectoryName	String	BhelFromRemote

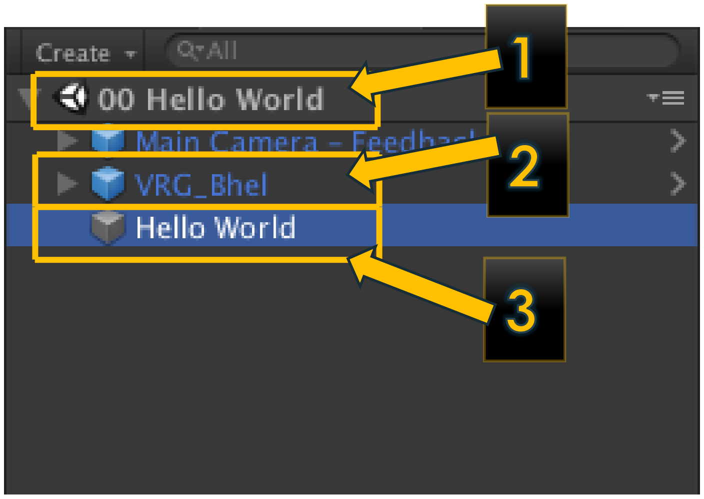
9. EXAMPLES

We created some example scenes to have a better understanding, the examples are in the _VrGamesDev/BHEL/Scenes folder.

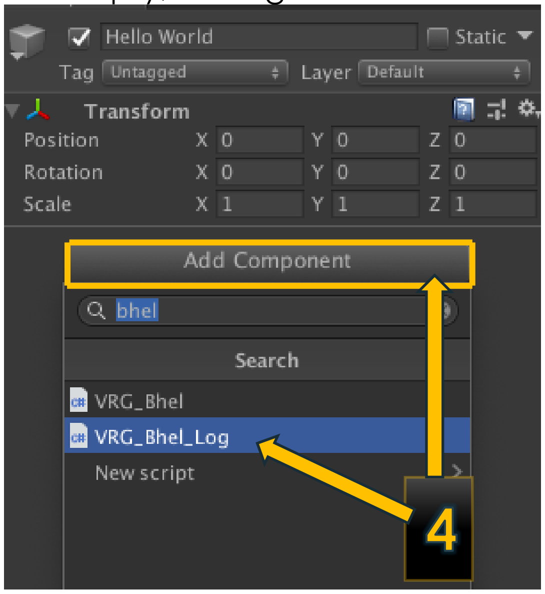
9.1 00 Hello World

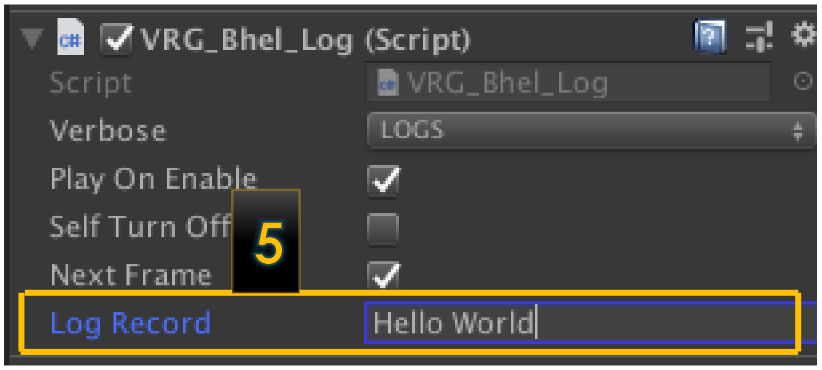
The simplest implementation, just an empty bhel with a hello world

1) Create a new scene, we call it "00 Hello World"
2) Create a BHEL object with default configuration, as explained in the section **4. Menu: Tools -> VR_Games_DEv-> BHEL**
3) Add an empty game object, we name it "Hello world"



4) Add the script component: VRG_Bhel_Log, as explained in the section **6. HOW TO ADD RECORDS TO THE LOGS**
5) In the property *log record* from the script VRG_Bhel_Log, type "Hello world", If you leave the field empty, the log will be the name of the game object.

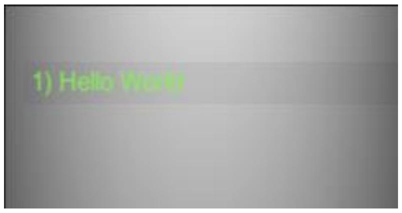




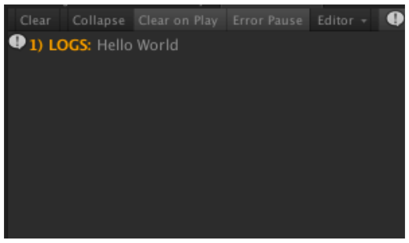
Script	VRG_Bhel_Log
Verbose	LOGS
Play On Enable	✓
Self Turn Off	
Next Frame	✓
Log Record	Hello World

6) Play the game! You will notice the following effects.

7) A "Hello World" green message in the screen, as a green floating UI

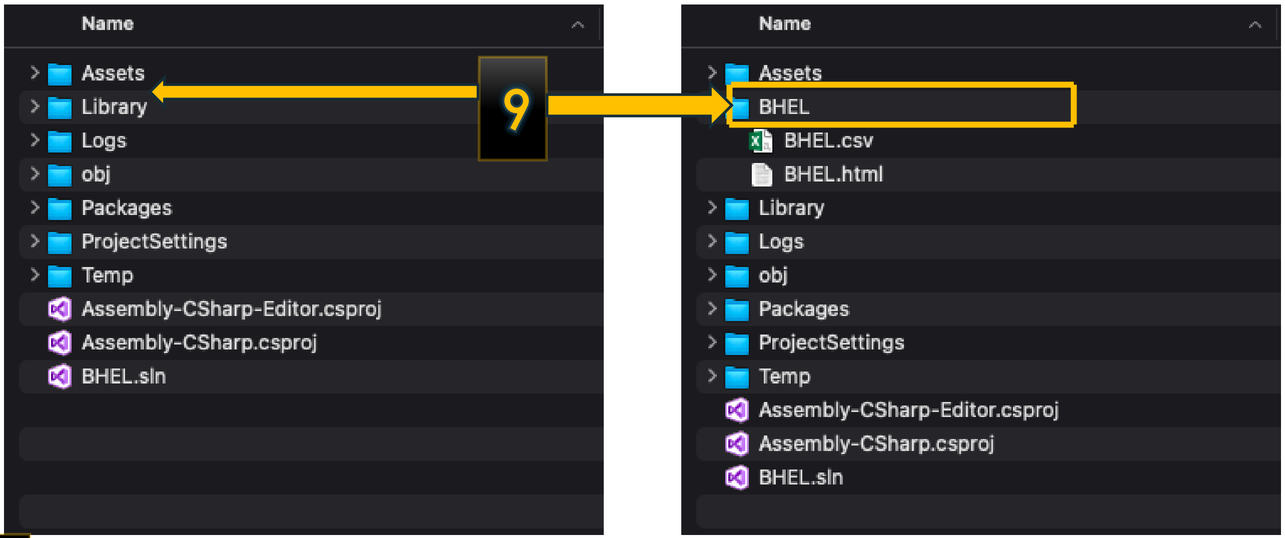


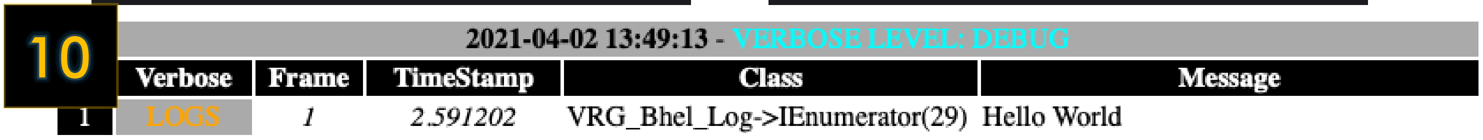
8) A console message displaying what was sent to the files



```
1) LOGS: Hello World
```

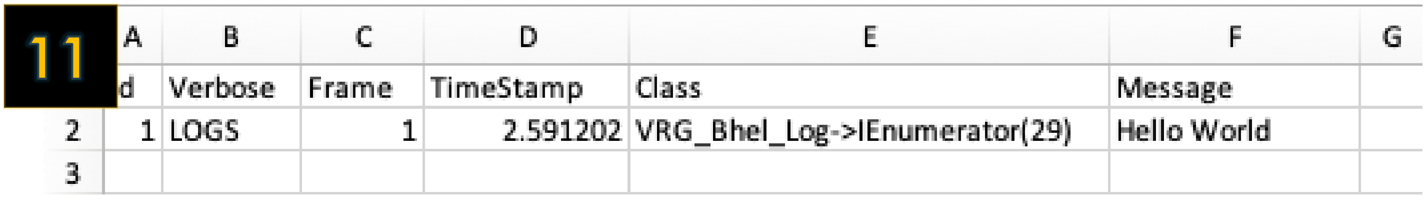
9) You will now have a new folder named "BHEL" in the main folder of your project
10) A HTML file with a "hello world" record
11) A CSV file with a "hello world" record





2021-04-02 13:49:13 - VERBOSE LEVEL: DEBUG

	Verbose	Frame	TimeStamp	Class	Message
1	LOGS	1	2.591202	VRG_Bhel_Log->IEnumerator(29)	Hello World

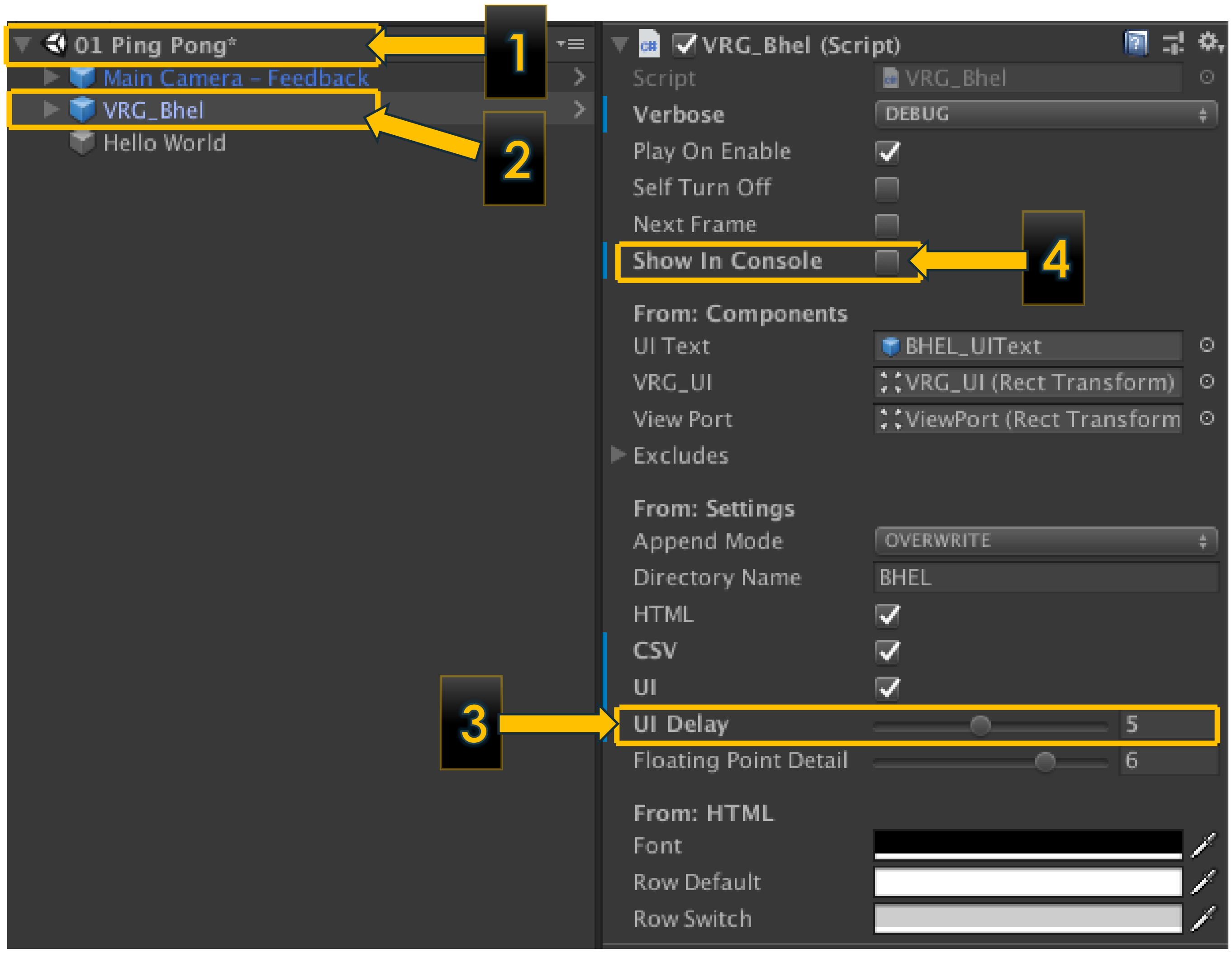


	A	B	C	D	E	F	G
	d	Verbose	Frame	TimeStamp	Class	Message	
2	1	LOGS	1	2.591202	VRG_Bhel_Log->IEnumerator(29)	Hello World	
3							

9.2 01 Ping Pong

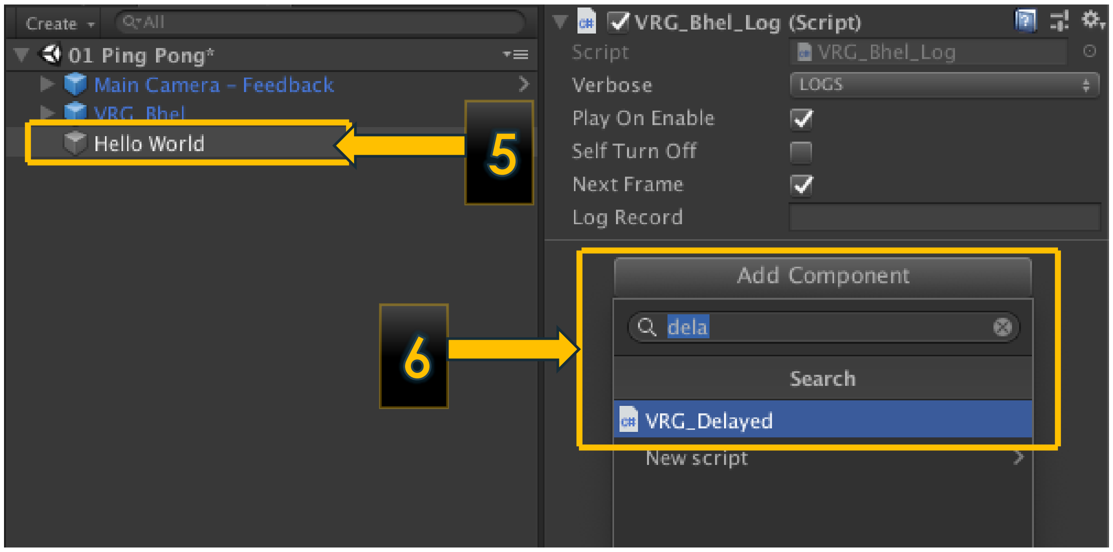
Let's create a scene that auto logs every X seconds two messages, that toggle on and off each other and wait 1 second to resend the message, for this purpose we will use the VRG_Delayed script from the CORE package.

1) Clone the Scene *00 Hello World* and rename it as *01 Ping Pong*
2) Select the gameObject "*VRG_Bhel*"
3) Modify the "*UI_Delay*" field from 2 to 5
4) If you don't want to see the logs in the console, toggle off the "*Show In Console*" field

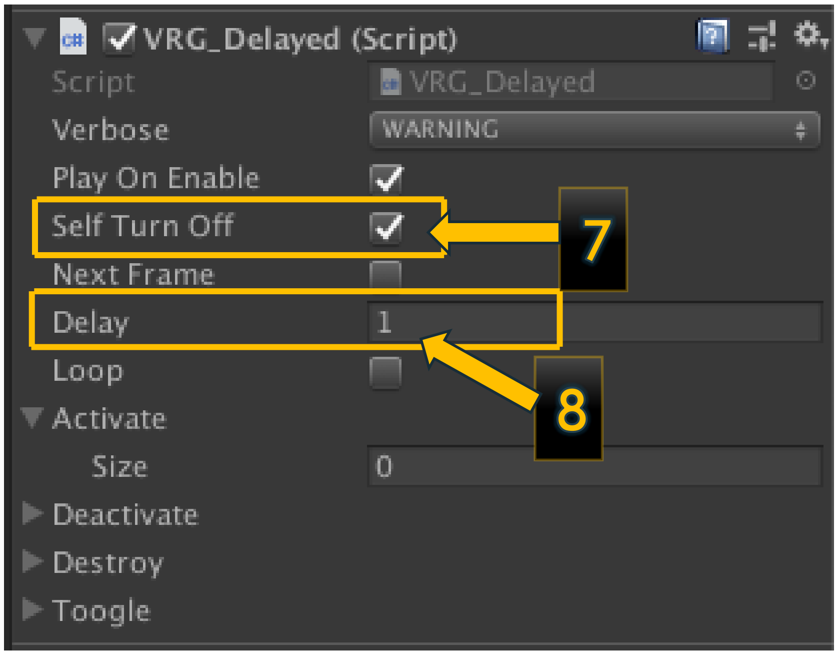


Script: VRG_Bhel
Verbose: DEBUG
Play On Enable: ✓
Self Turn Off: ☐
Next Frame: ☐
Show In Console: ☐
From: Components
UI Text: BHEL_UIText
VRG_UI: VRG_UI (Rect Transform)
View Port: ViewPort (Rect Transform
Excludes
From: Settings
Append Mode: OVERWRITE
Directory Name: BHEL
HTML: ✓
CSV: ✓
UI: ✓
UI Delay: 5
Floating Point Detail: 6
From: HTML
Font
Row Default
Row Switch

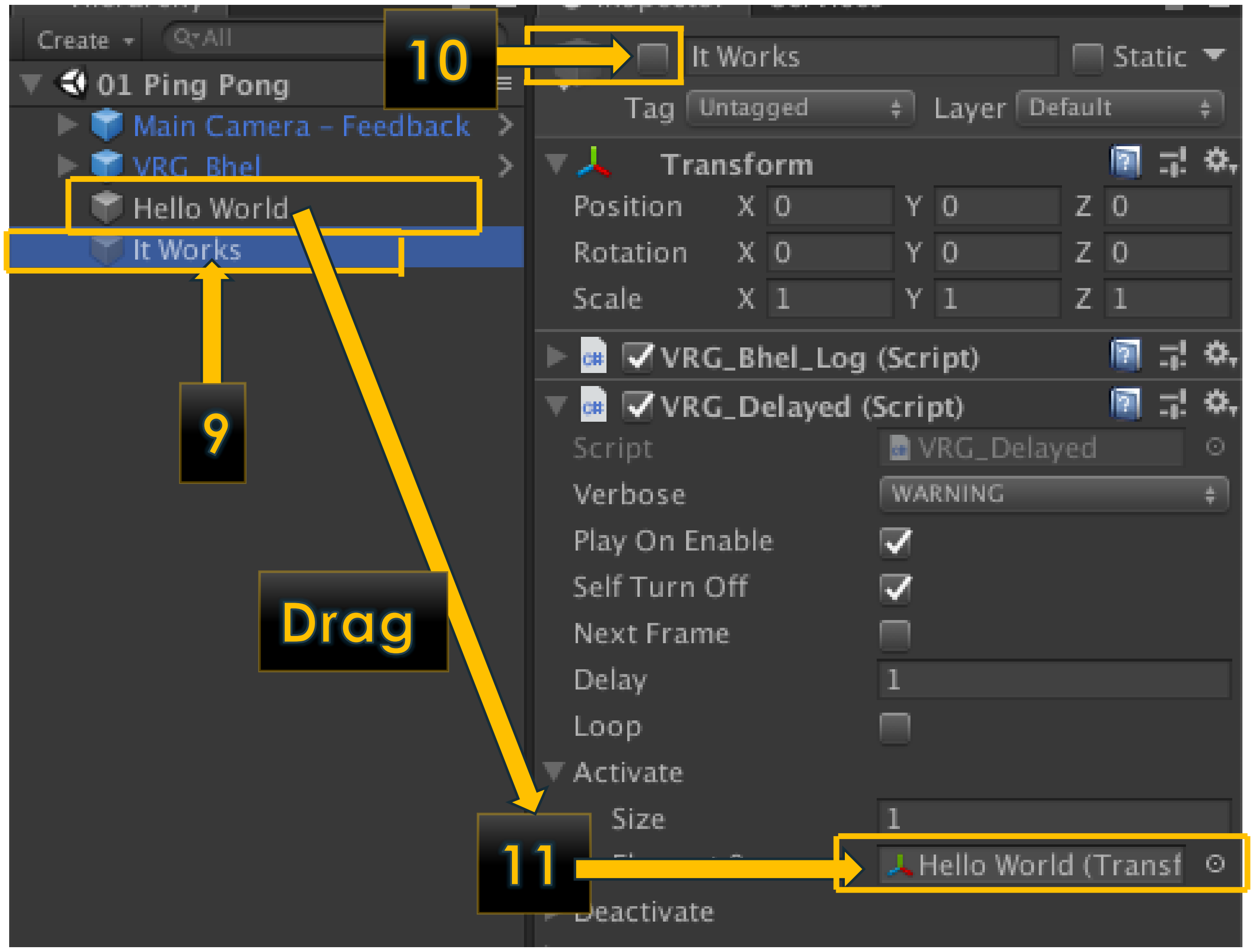
5) Select the *Hello World* game object
6) Add the script “VRG_Delayed” to the “Hello World” game object



7) In the VRG_Delayed script, Toggle **ON** the option “Self Turn Off”
8) Change the value of *“Delay”* field from 0 to 1



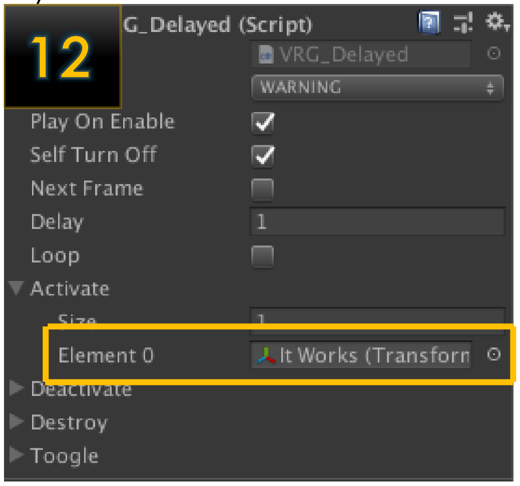
9) Clone the "*Hello world*" game object and rename the clone as "*It works*"
10)Select the "*It works*" game object and toggle it off
11)Drag from the hierarchy, the game object "Hello world" into the "Activate" array



Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

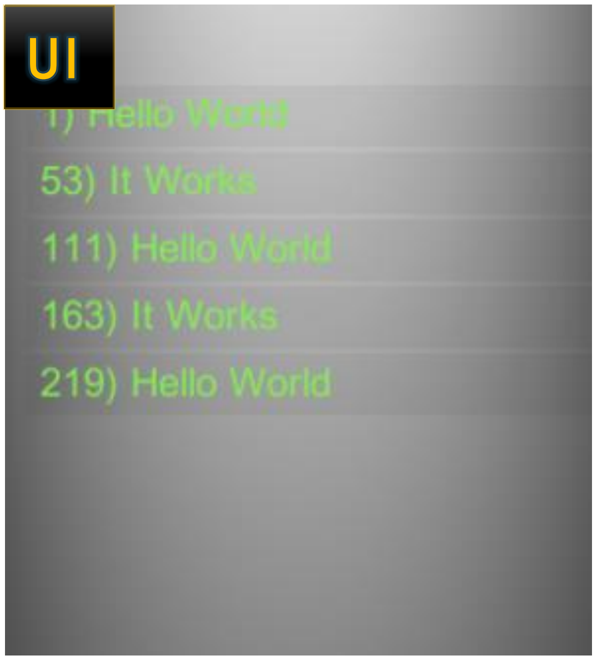
Script VRG_Delayed
Verbose WARNING
Play On Enable ✓
Self Turn Off ✓
Next Frame
Delay 1
Loop
Size 1
Hello World (Transf

12)Repeat the process with the "*Hello world*" game object and drag the the game object "*It works*" into its "Activate" array.

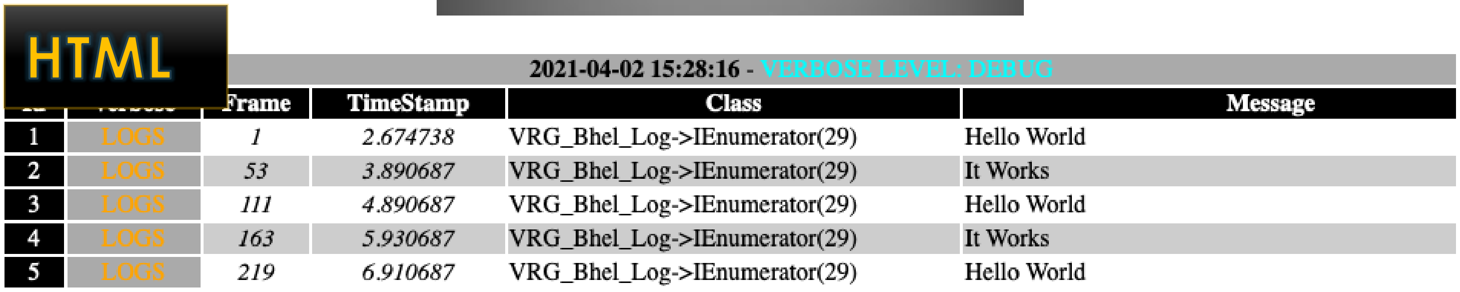


VRG_Delayed
WARNING
Play On Enable ✓
Self Turn Off ✓
Next Frame
Delay 1
Loop
Size 1
Element 0 It Works (Transform

13)Play the game
14)Review the logs UI, Html and CSV

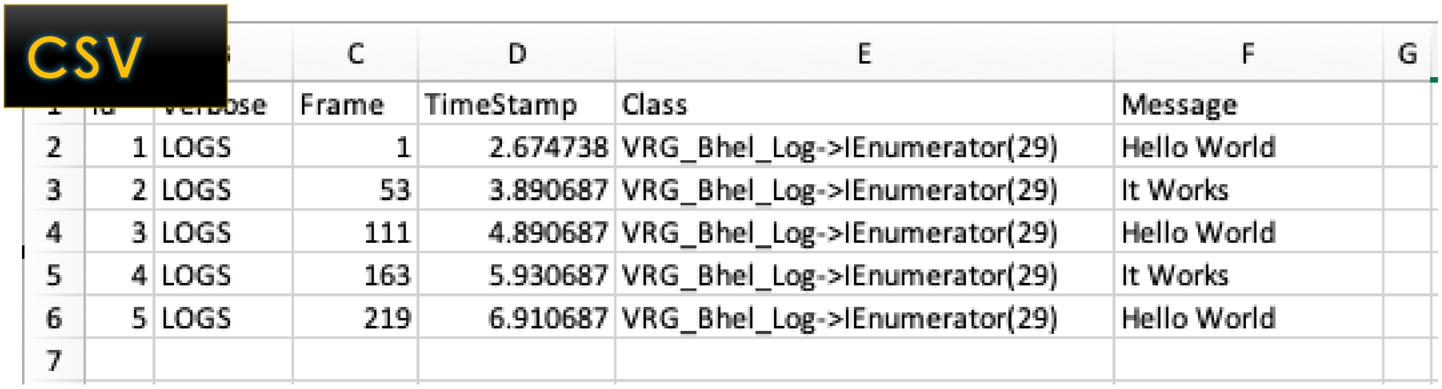


1) Hello World
53) It Works
111) Hello World
163) It Works
219) Hello World



2021-04-02 15:28:16 - VERBOSE LEVEL: DEBUG

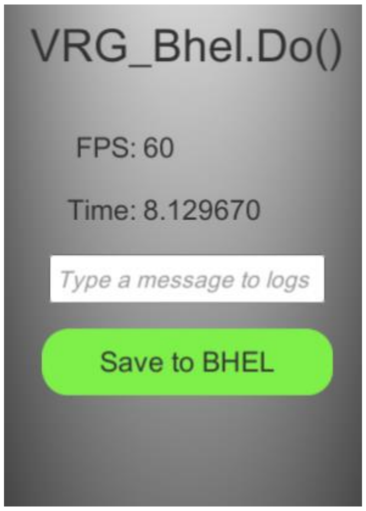
		Frame	TimeStamp	Class	Message
1	LOGS	1	2.674738	VRG_Bhel_Log->IEnumerator(29)	Hello World
2	LOGS	53	3.890687	VRG_Bhel_Log->IEnumerator(29)	It Works
3	LOGS	111	4.890687	VRG_Bhel_Log->IEnumerator(29)	Hello World
4	LOGS	163	5.930687	VRG_Bhel_Log->IEnumerator(29)	It Works
5	LOGS	219	6.910687	VRG_Bhel_Log->IEnumerator(29)	Hello World



			C	D	E	F	G
1			Frame	TimeStamp	Class	Message	
2	1	LOGS	1	2.674738	VRG_Bhel_Log->IEnumerator(29)	Hello World	
3	2	LOGS	53	3.890687	VRG_Bhel_Log->IEnumerator(29)	It Works	
4	3	LOGS	111	4.890687	VRG_Bhel_Log->IEnumerator(29)	Hello World	
5	4	LOGS	163	5.930687	VRG_Bhel_Log->IEnumerator(29)	It Works	
6	5	LOGS	219	6.910687	VRG_Bhel_Log->IEnumerator(29)	Hello World	
7							

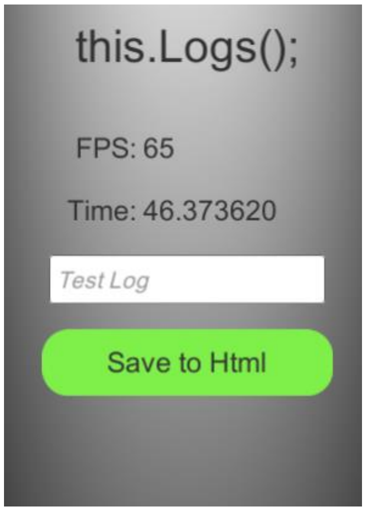
9.3 02 Singleton

This scene provides an example class that save custom Logs using the VRG_Bhel singleton as explained in the section 6.2 Singleton VRG_Bhel.Do(). You have a UI interphase and every time you type a message and click the "save to html" button, a message is saved into the log files.



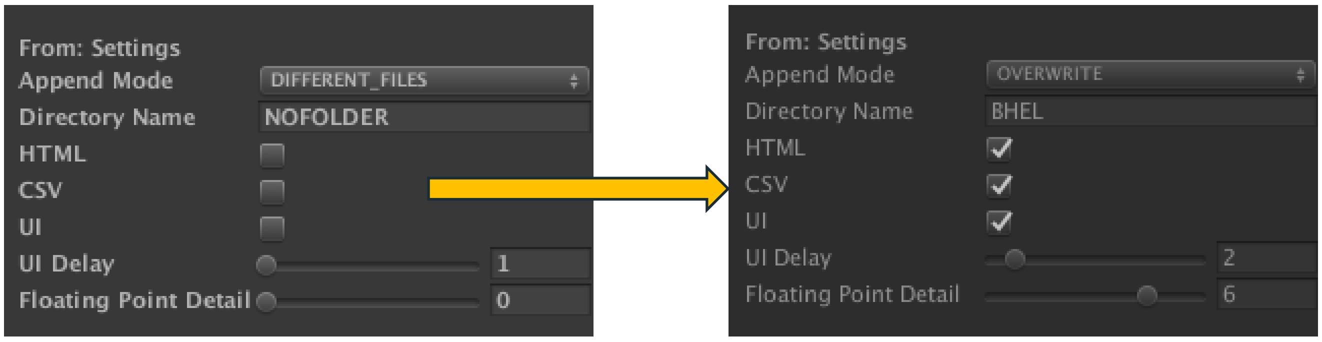
9.4 03 Inheritance

This scene provides an example class that save custom Logs using inheritance from VRG_Base as explained in the section 6.3 Inheritance from VRG_Base.Logs(). You have a UI interphase and every time you type a message and click the "save to html" button, a message is saved into the log files.



9.5 04 VRG_Remote

This scene provides an example class that uses and custom the VRG_Bhel settings using the VRG_Remote as explained in the section 7.2 Remote data for BHEL.
Remember that the settings defined in the VRG_Remote object overwrites the data defined in the VRG_Bhel object.
The settings are all false and set to minimum, and the settings in the VRG_Remote are different and plenty, play the game and inspect the VRG_Bhel.



From: Settings

Append Mode	DIFFERENT_FILES
Directory Name	NOFOLDER
HTML	☐
CSV	☐
UI	☐
UI Delay	1
Floating Point Detail	0

From: Settings

Append Mode	OVERWRITE
Directory Name	BHEL
HTML	✓
CSV	✓
UI	✓
UI Delay	2
Floating Point Detail	6

9.6 05 EVERYONEEEEEeeeee

This scene includes all the examples into a single scene, VRG_Remote support, and the script, inheritance and singleton modes to record a log.

10. … AND THEY LIVED HAPPILY EVER AFTER

I hope this package will help you to develop faster and easier, drop me a line and a 5 star rating if you enjoy it.

10.1 Here comes a new challenger

Play with the settings from the *VRG_Bhel* object to achieve different results, then play the game multiple times, and check the logs generated, some examples:

1) Change the append mode from "*Overwrite*" to "*Append*"
2) Change the append mode from "*Append*" to "*Different files*"
3) Toggle on / off the different controls:
    a) Html
    b) CSV
    c) UI
    d) Show in console
4) Change the verbose detail in the *VRG_Bhel* from none (silent, no logs generated), to Debug, the most verbose,
5) Complete the tests changing the verbose of the "Hello World" object
6) Adjust the values of the settings:
    a) UI Delay
    b) Floating Point detail
    c) Font color
    d) Row Default color
    e) Row Switch Color

10.2 Good Bye