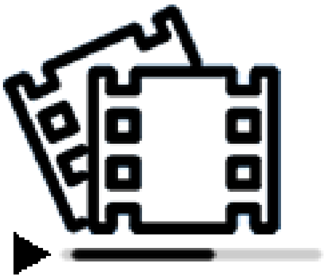
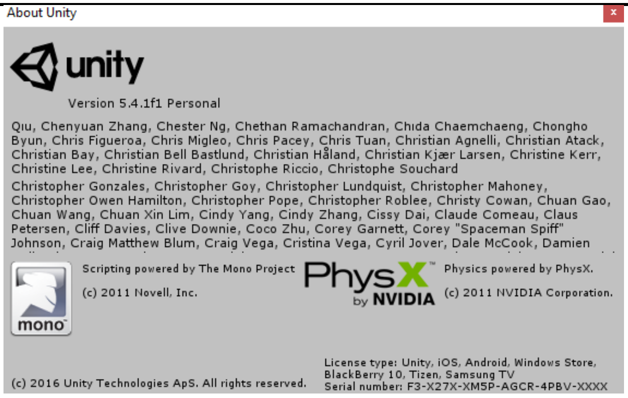
Video gallery and video player for easy movie texture



1. GENERAL INFORMATION

DATE OF DOCUMENT	09/12/2016
NAME OF THE PROJECT	Video gallery and video player for easy movie texture
AUTHOR	Michael Soler
UNITY VERSION	**5.4.1.F1 PERSONAL**
CONTACT	michael.soler.beatty@gmail.com



About Unity

Version 5.4.1f1 Personal

Qıu, Chenyuan Zhang, Chester Ng, Chethan Ramachandran, Chıda Chaemchaeng, Chongho Byun, Chris Figueroa, Chris Migleo, Chris Pacey, Chris Tuan, Christian Agnelli, Christian Atack, Christian Bay, Christian Bell Bastlund, Christian Håland, Christian Kjær Larsen, Christine Kerr, Christine Lee, Christine Rivard, Christophe Riccio, Christophe Souchard

Christopher Gonzales, Christopher Goy, Christopher Lundquist, Christopher Mahoney, Christopher Owen Hamilton, Christopher Pope, Christopher Roblee, Christy Cowan, Chuan Gao, Chuan Wang, Chuan Xin Lim, Cindy Yang, Cindy Zhang, Cissy Dai, Claude Comeau, Claus Petersen, Cliff Davies, Clive Downie, Coco Zhu, Corey Garnett, Corey "Spaceman Spiff" Johnson, Craig Matthew Blum, Craig Vega, Cristina Vega, Cyril Jover, Dale McCook, Damien

Scripting powered by The Mono Project

(c) 2011 Novell, Inc.

Physics powered by PhysX.

(c) 2011 NVIDIA Corporation.

(c) 2016 Unity Technologies ApS. All rights reserved.

License type: Unity, iOS, Android, Windows Store, BlackBerry 10, Tizen, Samsung TV
Serial number: F3-X27X-XM5P-AGCR-4PBV-XXXX

Includes video tutorial for script info!

Index

2. IMPORTING INFORMATION

This package works with the easy movie texture, it needs the following package. You need to import it BEFORE this package.

https://www.assetstore.unity3d.com/en/#!/content/10032

3. PROJECT DESCRIPTION

This is a package that woks with the "easy movie texture" and generates a video gallery, in which some covers are displayed. Each of the covers take to a different video, inside an interface like the youtube video player.

Features

Select a cover to play the video associated to it.

Loading bar to show the video progress.

Click on the loading bar to seek video position.

Auto-hiding feature during reproduction (after x-seconds).

Play and pause button in the interface.

Volume configuration.

Return to the main gallery by pressing a button.

This package contains the following:

-The necessary FBX, textures, models, scripts shown in the video.

-Scripts manage the interaction with video player.

-One complete scene that works as in the link.

-Complete documentation/video tutorial to understand the principles of each package and full email support at: michael.soler.beatty@gmail.com.

This package needs the following:

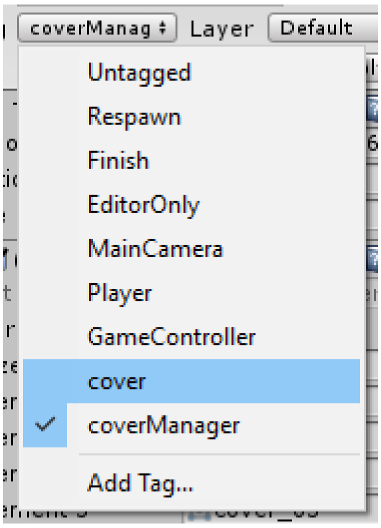
The easy movie texture package:

https://www.assetstore.unity3d.com/en/#!/content/10032

4. TAGS

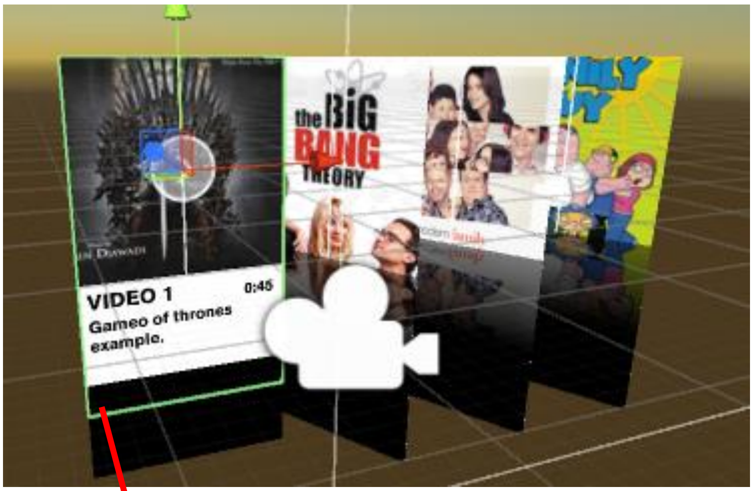
TAGS:

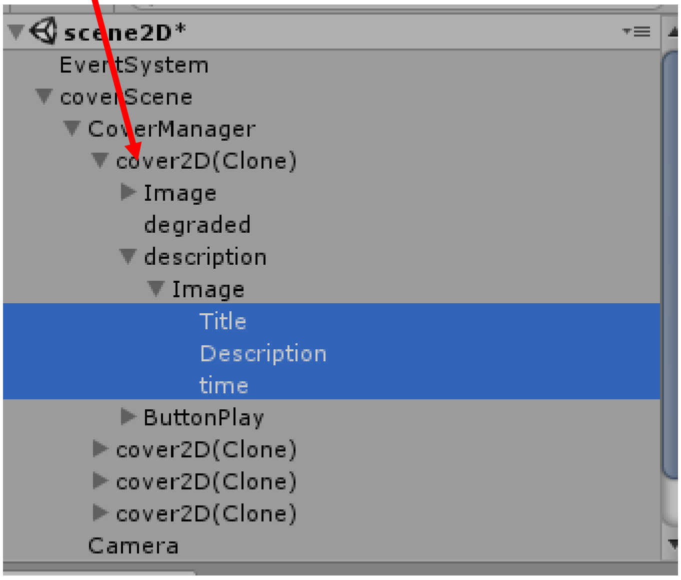
Verify that the following tags exits and are well associated to the gameobjects.



5. PREFABS & SCRIPT INFORMATION

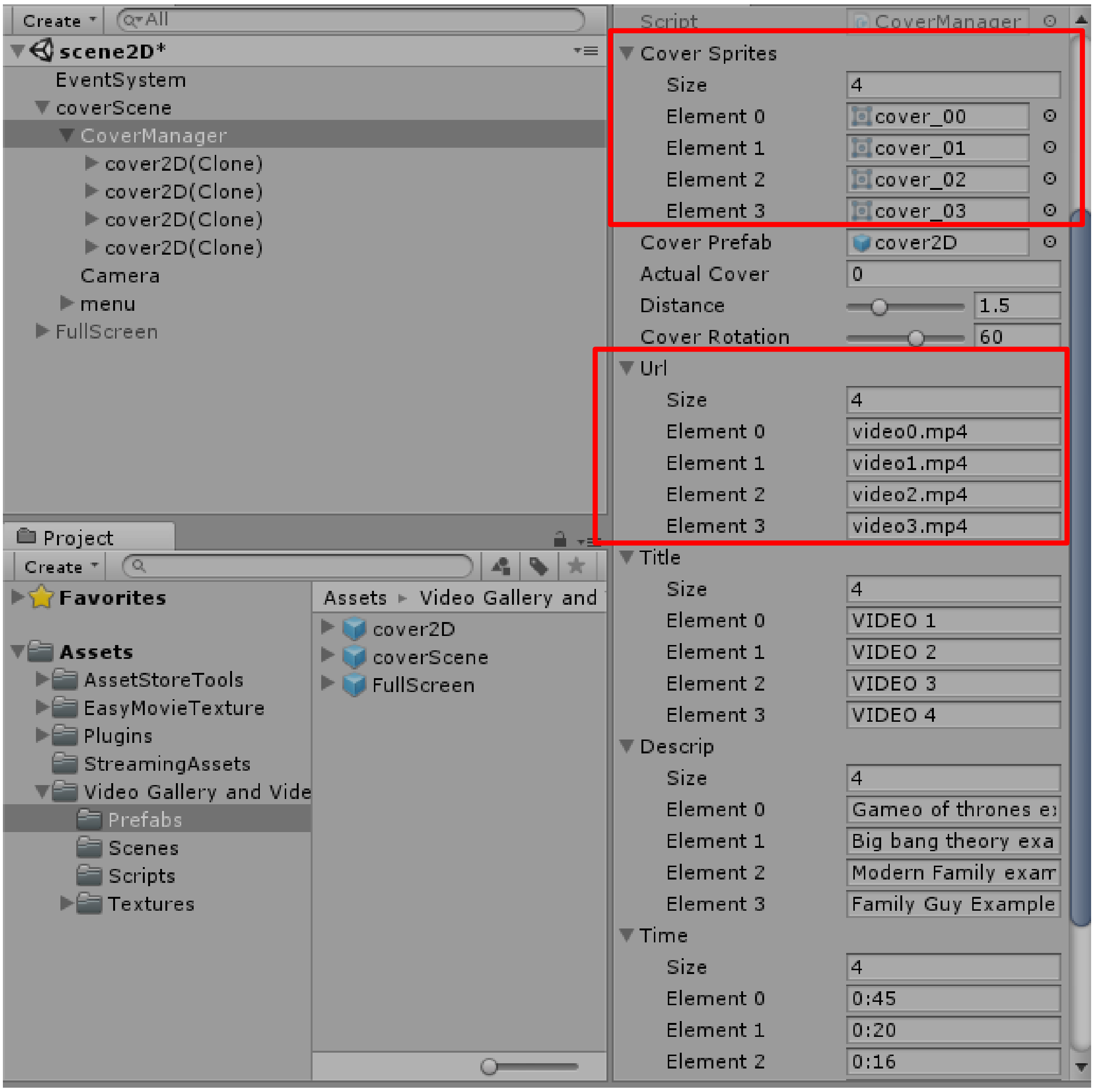
The “cover2D” is each of the gameobjects that are created when the application runs. This is done thanks to a “coverManager” script that gives the titles, descriptions, durations and video links.





```
▼ scene2D*
    EventSystem
  ▼ coverScene
    ▼ CoverManager
      ▼ cover2D(Clone)
        ▶ Image
          degraded
        ▼ description
          ▼ Image
              Title
              Description
              time
        ▶ ButtonPlay
      ▶ cover2D(Clone)
      ▶ cover2D(Clone)
      ▶ cover2D(Clone)
      Camera
```

The “covermanager” script is the most important one in the package, because it manages all the global behavior if each of the covers.



Script: CoverManager

Cover Sprites
Size: 4
Element 0: cover_00
Element 1: cover_01
Element 2: cover_02
Element 3: cover_03
Cover Prefab: cover2D
Actual Cover: 0
Distance: 1.5
Cover Rotation: 60

Url
Size: 4
Element 0: video0.mp4
Element 1: video1.mp4
Element 2: video2.mp4
Element 3: video3.mp4

Title
Size: 4
Element 0: VIDEO 1
Element 1: VIDEO 2
Element 2: VIDEO 3
Element 3: VIDEO 4

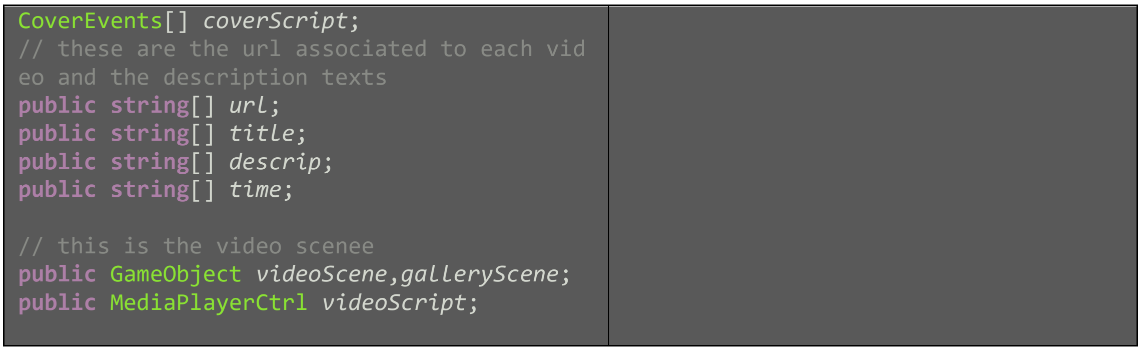
Descrip
Size: 4
Element 0: Gameo of thrones e:
Element 1: Big bang theory exa
Element 2: Modern Family exam
Element 3: Family Guy Example

Time
Size: 4
Element 0: 0:45
Element 1: 0:20
Element 2: 0:16

Is the variable "coverSprites", that allows to change the images of each cover and, the "url" represents the names of the videos (or its http address). By changing the sizes o these variables, you will be able to **add more videos and covers.**

coverManager.cs	
Important variables	**Important functions**

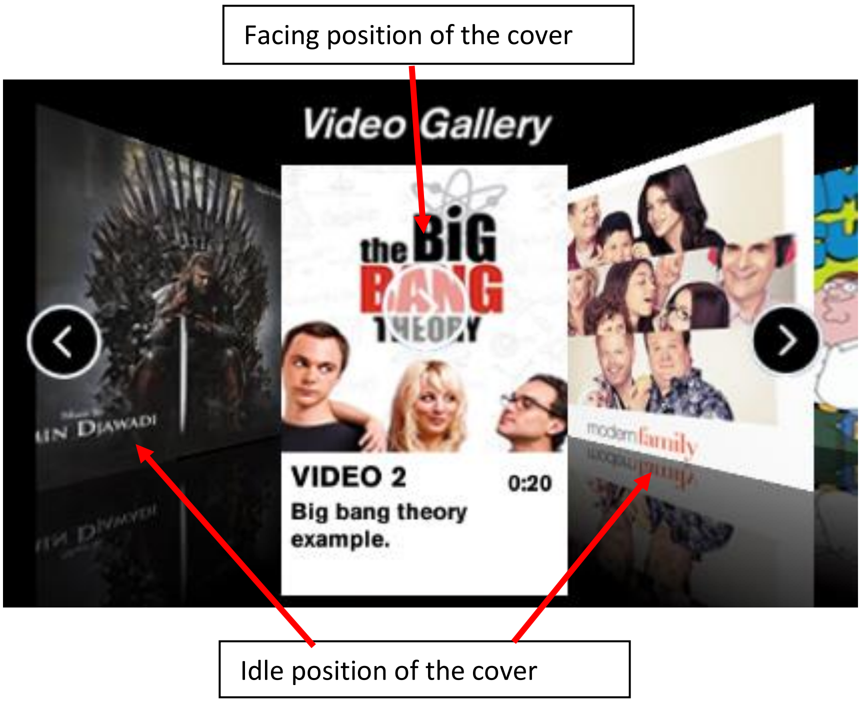
//this is the number of covers, given by the textures
public Sprite[] coverSprites;
//this is the prefab cover gameobjec
public GameObject coverPrefab;
//this is the actual cover
public int actualCover;
//this is the distance between covers
[Range(0.2f,5.5f)]
public float distance=1.5f;
//this is cover rotation when idle
[Range(45f,70f)]
public float coverRotation=60f;
//these are the covers referenced as gameobjects
GameObject[] go;
//these are the script of the covers
	//creates the covers
	public void create ()
	//changes the position of the covers to face the camera or at an angle
	public void ResetPositions ()
	// goes to the video using the reference of the cover number
	public void goToVideoScene (int a)



```
CoverEvents[] coverScript;
// these are the url associated to each vid
eo and the description texts
public string[] url;
public string[] title;
public string[] descrip;
public string[] time;

// this is the video scenee
public GameObject videoScene,galleryScene;
public MediaPlayerCtrl videoScript;
```

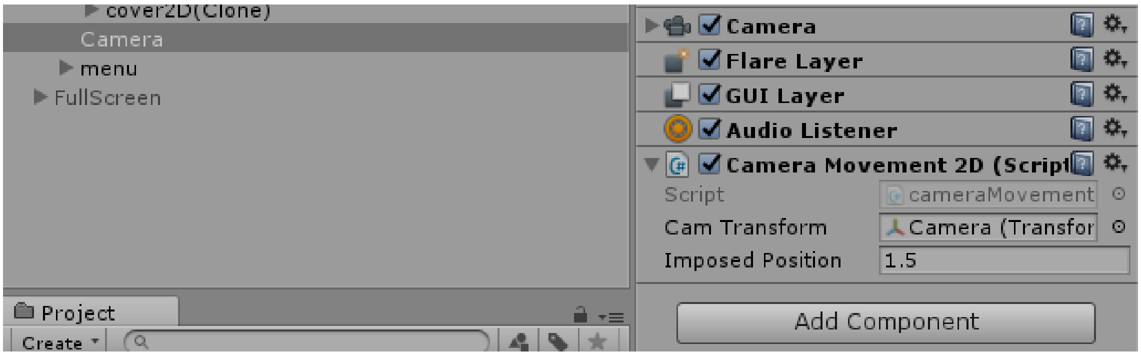
So what basically is occurring is, when the player uses the arrows to move from left to right (that is controlled with the cameraMovement2D), the covers start to face or stay at an Idle position.



Facing position of the cover

Idle position of the cover

The cameraMovement2D is a script that moves the camera to the right and left 1.5f (which corresponds to the distance between each of the center of the covers).



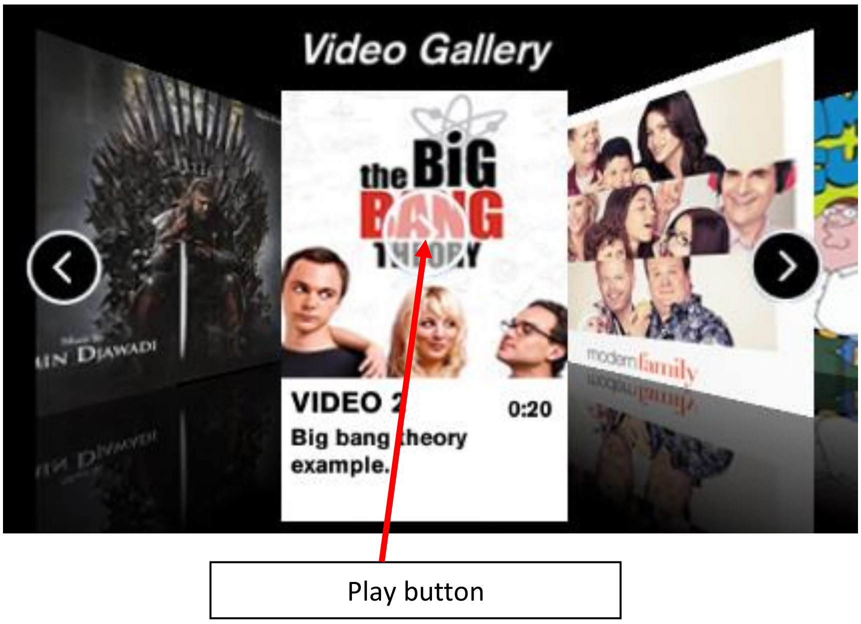
Script: cameraMovement

Cam Transform: Camera (Transfor

Imposed Position: 1.5

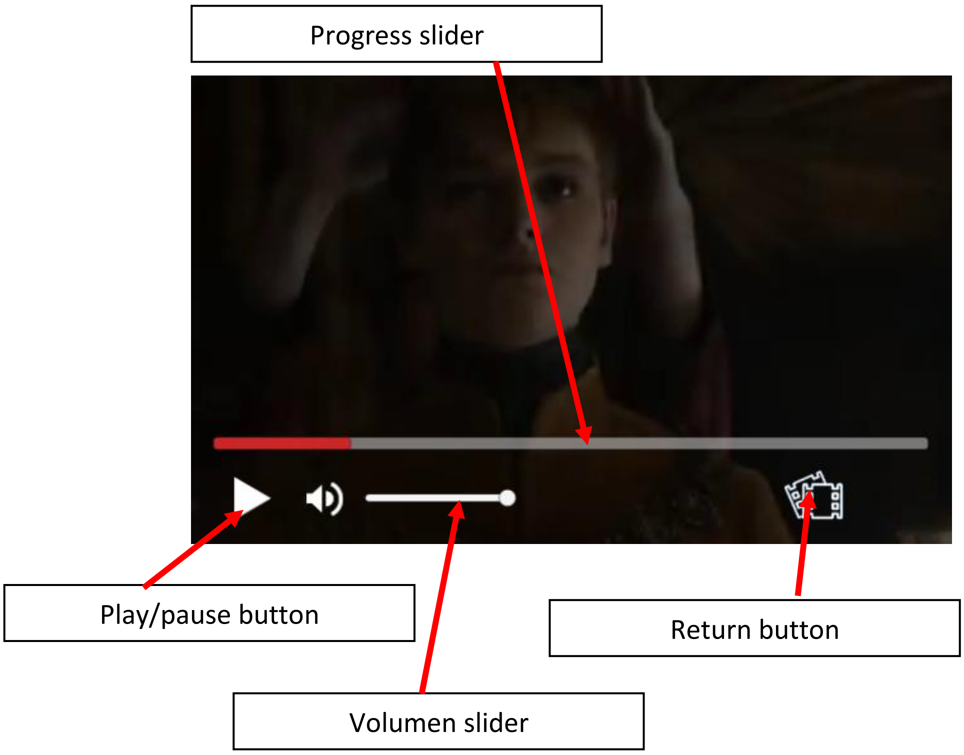
The imposed position indicates where it is placed at this current moment.

Once a video is selected by pressing the center “play” button, the script will take the user to the video scene, in which a different behavior is programmed.



Play button

In this case, a new menu similar to the youtube’s video interface appears with similar functionalities.



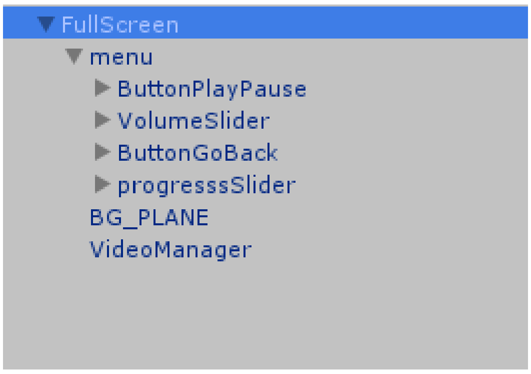
Progress slider

Play/pause button

Return button

Volumen slider

Each of the objects is controlled by and independent script:



```
▼ FullScreen
   ▼ menu
      ▶ ButtonPlayPause
      ▶ VolumeSlider
      ▶ ButtonGoBack
      ▶ progresssSlider
     BG_PLANE
     VideoManager
```

PROGRESS BAR → SliderFunctions.cs	
Important variables	**Important functions**

//Reference to the video that is playing
public MediaPlayerCtrl *videoScript*;
// reference to the slider
Slider *slider*;
// this is necesary to be able to seek the video position
bool *pointerEnter*=false;
//variables that show the video time
public float *duration*, *videoTime*;
	// this function allows to change the position of the video
	public void *changeValue* ()

PLAY-PAUSE BUTTON → playPauseManager.cs	
Important variables	**Important functions**

//these are the two sprites used (pause and play)
public Sprite *playSpr*, *pauseSpr*;
// this is the button used for the interaction
public Image *playPauseButon*;
// this is the reference to the videos script
public MediaPlayerCtrl *videoScript*;
// the video starts playing and we need to control when it is paused:
public bool *isPlaying*=true;
	// this is the part where we check if the video is playing and we choose the action to perform
	public void *playPause* ()

VOLUME SLIDER→ VolumeManager.cs	
Important variables	**Important functions**

// this is the reference to the videos script
public MediaPlayerCtrl *videoScript*;
// reference to the slider
Slider *slider*;
	// this is called in the onChangedValue in the slider
	public void *changeValue* ()

<table>
<tr><th colspan="2">RETURN BUTTON→ GoToGallery.cs</th></tr>
<tr><th>Important variables</th><th>Important functions</th></tr>
<tr><td>//this are the scenes that will deactivated and activated
public GameObject deaScene, actScene;</td><td>// it is called onclick button
public void back()</td></tr>
</table>

6. VIDEO TUTORIAL FOR SCRIPT/TECHNICAL DETAILS

We are trying to give users a more advanced technical service by creating some videos that explain the main aspects of our packages and gives a detailed explanation about how the package works. You will find this tutorial at the following link:

https://youtu.be/V38-7ln6lSY