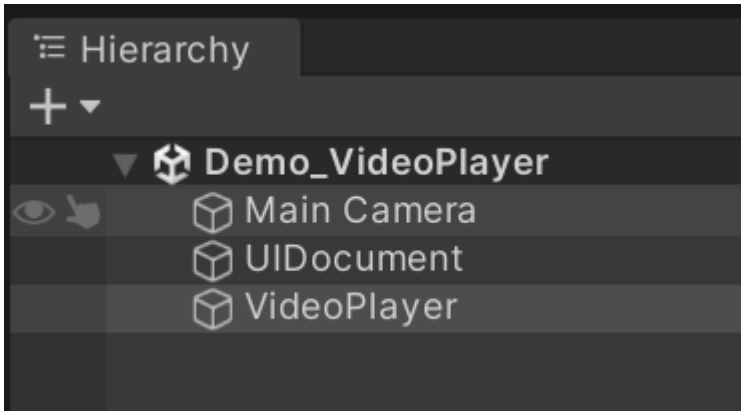
Documentation

Thanks for purchasing the video player for UI ToolKit asset. The asset comes with code and an example project. Let's get started!

Contents

1. Demo Scene

Navigate to *Assets* > *Plugins* > *VideoPlayerUIToolKit* > *Scenes* > *Demo_VideoPlayer*, then open it.



The `UIDocument` gameobject contains the following components: `UIDocument`, `VideoControlsUI`, `VideoRenderTextureUI`.

The `VideoPlayer` gameobject contains the following components: `VideoPlayer`, `VideoController`, `VideoGallery`.

The camera is standard without any changes.

2. Sample Video

A sample video can be found here: `Assets/StreamingAssets/SamepleVideo.mp4`. The app automatically loads it at the start, please see the Script Reference section on how to modify scripts.

3. Script References

VideoControlsUI

Description:
`VideoControlsUI` interfaces between the UI buttons and the backend scripts. It registers callbacks in the *OnEnable()* method.

Example Usage:
When the play button is clicked the *ButtonPlay()* method is called which changes settings in the UI and tells the backend to play the video.

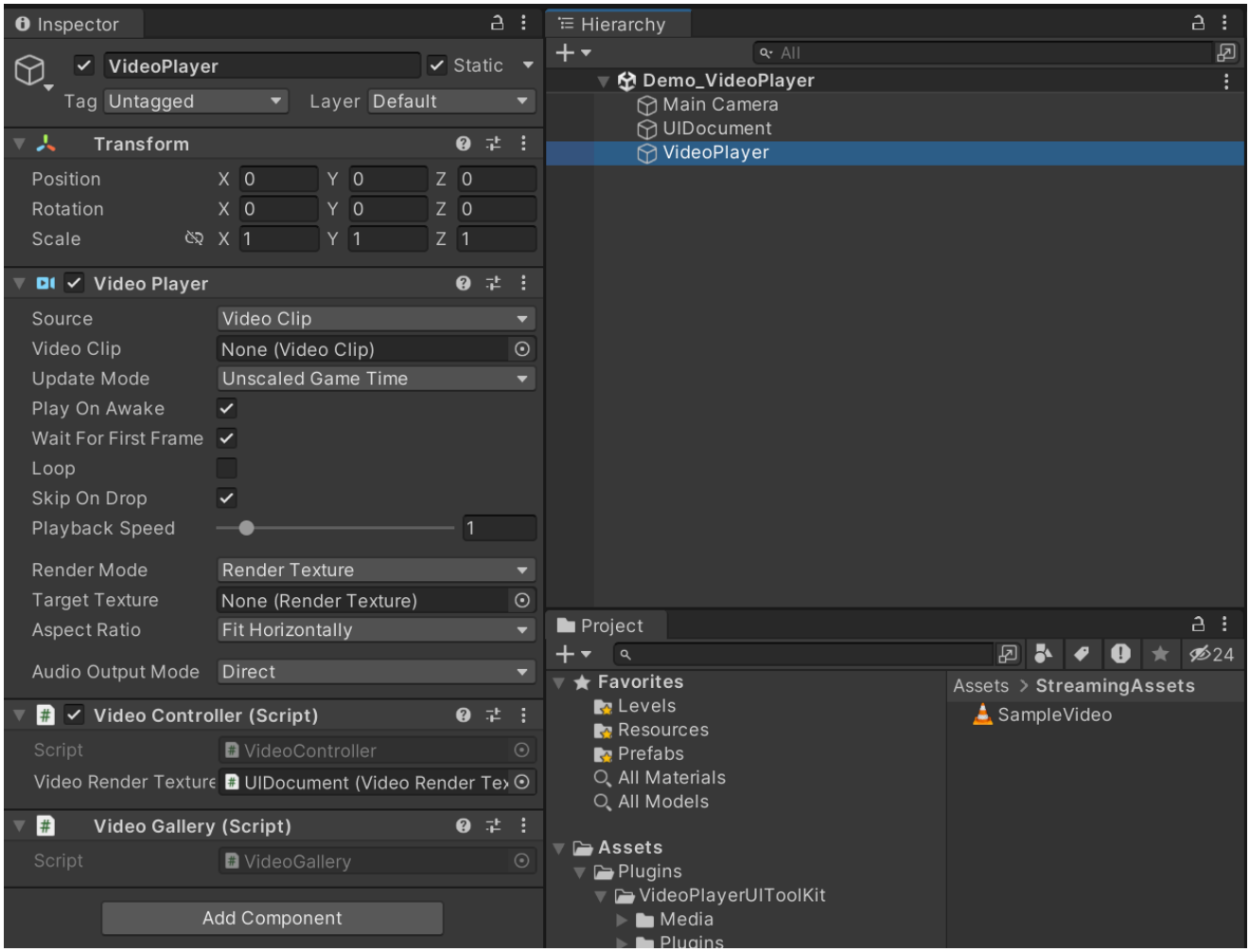
```
void RegisterCallbacks()
{
    root.Q<Button>("Button_Play").clicked += ButtonPlay;
}

private void ButtonPlay()
{
        root.Q<Button>("Button_Play").style.display = DisplayStyle.None;
        root.Q<Button>("Button_Pause").style.display = DisplayStyle.Flex;

        videoController.PlayVideo();
}
```

VideoController

Description:
`VideoController` controls Unity's *VideoPlayer* component found in the Editor (see screenshot below). All the video controls are found here such as Play, Stop, Mute.
A sample video is automatically loaded in the *Start()* function. The *Start()* function can safely be removed or altered.

Functions:
The most notable function is *OnVideoPrepared()* creates a new *RenderTexture* every time a video is loaded. This is so that the video's dimensions can be set (it is not recommended to set a *RenderTexture's* dimensions at runtime).
The remaining functions are used to control the *VideoPlayer* such as: *PlayVideo(), StopVideo(), LoopVideo(), Mute(), Scrub()*. Each function references the VideoPlayer and does the action (ie. *VideoPlayer.Play()*).

Location of *VideoPlayer* in the Editor:



VideoSliderManipulator

Description:

`VideoSliderManipulator` allows the user to "scrub" the video. The class base a Manipulator. The reason for creating a Manipulator is that the Slider consumes the pointer events stopping, so the Manipulator class allows more control.

Functions:

PointderDown() tells the script that the scrub bar has been clicked.
SliderValueChanged() called when the Slider value changes. In turn it sets the position of the video.
SliderProgress() updates the position of the slider as the video plays.
PointerUp() tells the script to stop scrubbing. This method contains a delay to smooth the position of the slider otherwise it can jump.

VideoRenderTextureUI

Description:

`VideoRenderTextureUI` is used by `VideoController` to set the current video's *RenderTexture* in the UIDocument.

Functions:
SetRenderTexture() sets the RenderTexture to the corresponding VisualElement in the UIDocument.

VideoGallery

Description:
`VideoGallery` interfaces with `NativeGallery` which is a gallery picker for all platforms. It is an an open source plugin hosted on GitHub.

For more information please check out the Licence section and also see the GitHub repositry:
https://github.com/yasirkula/UnityNativeGallery

Functions:
Pick() uses `NativeGallery` to copy the filename and path of a video file. The video file path is sent to *VideoController* to be prepared and loaded.

VideoPlayer_UXML & VideoPlayer_USS

The user interface (UI) has been created in UIBuilder. To edit the UI locate the files: `VideoPlayer_UXML` and `VideoPlayer_USS` located in *\Assets\Plugins\VideoPlayerUIToolKit\UI Toolkit*.
For more information about UXML and USS see the Unity documentation for UIToolKit.

4. Licenses

UIToolKit Video Player

Standard Unity EULA.
https://assetstore.unity.com/browse/eula-faq
https://unity.com/legal/as-terms

Sample Video

The video has no copyright restrictions associated with it. It is royalty free and can be used in any way.

Unity Native Gallery Licence

MIT License

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE
SOFTWARE.