**Display Video in Unreal**

For each video you need the media file and the audio file. import both into the engine and open the media file and check the Precache file option. Create a Media playlist and name it cutscene library. Open it and click the + sign and add both the videos. The ordering is important as that’s how we are going to refer to them later while playing the video. Make a media player and name it cutscene player and a dialog box asking whether to include video / music appears. Selecting both seems to cause a problem to the instructor so he chooses to show the video and play the audio later. So, check only the video but not the audio.

Open it and you’ll see the media player has been loaded automatically. Check the loop if you want. Make a widget called CutsceneWidget and open it. Add in an image and adjust size and then in the image input create a binding. Name this image Video. Add another image and set its color to black and mask it over the previous image and set the image opacity to 0 (for a fade in animation I want to add). Set the Z order of Video image to higher order as we want the video to always be above the black map.

Add a text on the screen and set the text to “Press Enter to skip”. Open event graph and create Custom Event called Play Cutscene with an input called CutScene index of type integer. Create a variable called MediaPlayer of type MediaPlayer – reference and set the default value to the Media player we created earlier. And Sound of type sound wave – reference. In the variable properties just next to the type of the variable is an icon and for this if you click it there is an option to convert it to array, or set or single variable. Choose array so we can add more than 1 sound. Get the Media Player, extend it and add an open playlist node and set the playlist to the one we created earlier. Hook the index to the CutScene index pin.

As we had set the media to looping but want to play it from beginning, we need to rewind it. Extend the media player variable and add a rewind node and hook the output exec pin to the input exec pin of the rewind node. Extend the media player variable again and add a play node and hook its input exec pin with the rewind pin’s output exec pin.

In the sound’s default value section click the + sign to enter as many sounds as you want. And set each value to the respective sound. Drag in a reference of the Sound variable, extend it and add a get node. Hook the index pin to the CutScene index pin. Extend the output pin and add a create sound 2D node and hook its input exec pin to the play node’s output exec pin. Extend the return value and add the play node.

Go to the binding created earlier for the Video image and split the return node and you’ll see a return value image pin. Drag in a reference to the media player, extend it and add a video texture node, extend it and hook it to the return value image pin.

If the game is a 3D game, we don’t want any actions to take place while playing the video. So extend the play node and add a game paused node and set it to true. Extend the media player variable and add a bind event to onEndReached node.

Drag the EndReached event node so we get some space to make it look clean. Beneath it add a new custom event called End. This is for stopping the player when the player presses a key to stop it. Extend the media player variable and add a pause node (To increase efficiency). Hook its input exec pin to both the output exec pin of the EndReached event node as well as the output exec pin of the End event node. Extend the return pin of the create sound 2D node and add a stop node and hook its input exec pin with the output exec pin of the stop. Extend the output exec pin of the stop node and add a set game paused node and uncheck it extend it and add a remove from parent node to remove the video, audio etc.

Go back to the play cutscene event node, extend it and add a delay node and set the delay to 1s. Add in a reference of the black image, extend it and add a set opacity node and set it to 1. Hook this between the newly inserted delay node and the open playlist index node. This gives a fade in transition effect.

But directly setting opacity to 1 doesn’t give the transition what we’ve done is to make sure that after the delay the opacity is sure to become 1. What we want to do now is to add in the transition and also display the text only 4s after the video has started playing and also check if the player has pressed enter.

Add an event tick node. Add in a reference to the black image, extend it and add a get color and opacity node, split the struct pin and extend the opacity pin and add a float+float node and set the other pin value to say 0.1. extend the image reference and add a set opacity node and hook the output of the float + float node to its value pin and hook its input exec pin to the event tick node. Make a variable called time and add in a reference to the node extend it and add a float+float node and to the other pin set the value to 0.1. Extend the return value and add a set time node and hook its input exec pin to the output exec pin of the set opacity node. Add a get player controller node, extend it and add an is input key down node and set the key to enter and extend the return value and add an AND node. Extend the output of the set time node and add a float>float node and set the other pin to 4s. hook the output pin in the other pin of the AND node. Extend the return value and add a branch node. Hook its input exec pin with the set time node’s output exec pin. Extend the true pin and add an end event node.