Documentation for Project :FLARToolkit with Video Markers and Buffering message.

Introduction

This project tries to use Squidder's MultiFLARMarker source to use different markers as Looped Video planes and also add a buffering pre-loader to each before they are loaded.

Algorithm

The Program first initializes n netstreams,video materials for each of the videos to be buffered.The play time for each is set to 0.

When a particular marker is shown,then the video is paused and a pre-buffered message is shown till it is pre-loaded upto a defined level. Once the level is reached ,the video is shown on the marker .

When the marker is removed from the screen.the video is paused .It will resume when it will be shown again to the other marker.

The videos are looped so that Once a video finishes It is played from the beginning .

Explanation of functions

public function MultiFLARExample()

The constructor MultiFlarExample initializes the array of markers and a Video_Status array(which stores whether the video is playing) and also initializes the pattern array.

override protected function _init(event : Event) : void {

This function is used to initialize the net streams and video materials for use with the other markers

override protected function _detectMarkers() : and _handleMarkerAdded & private function _addCube(id:int , index:int) : void {

Function addcube sees which marker is currently visible,then decides if the video for that marker is already playing or not .If the marker has not been shown at all,a new instance of the videomaterial is created and it is played.

If the marker was shown before then it resumes the video from the previous position.

override protected function _handleMarkerRemove(event : FLARDetectorEvent) : & _removeCube(event.codeId , event.codeIndex);

These functions handle the removal of the marker and basically toggle the video of the marker which is going to be removed.

private function onStatusEvent(stat:Object):void

This function is important because it is used to detect the status of the video being streamed.When a video is streamed in flash, the events occurring when the stream is played are stored as certain messages.Based on these messages we can decide when to stop buffering/loop the video.

How to add new markers,video and stream materials

The default number of videos and markers that are present in the source code are 3.If you want to add more videos ,please refer to the following instructions:

1) Add the videos and the marker patterns to the deploy folder.
2) In the variable list, Declare the appropriate number of variables

Eg: For adding a 4th video,

```
  private var video4:Video;
  private var netStream4:NetStream;
private var netConnection4:NetConnection;
  private var vm4:VideoStreamMaterial;
```

Note that these must be added before the Constructor declaration (public function MultiFlarExample () {

3) Now add the new pattern to the pattern list (line number 90)
   And add the line :
   _markers.push (new FLARMarkerObj("<name of your pattern>.pat ",16,50,80));

4) Now its time to declare the initializations,Go to line number 145 and paste the code similar to the below lines:

```
netConnection4= new NetConnection();

 netConnection4.connect(null);

 netStream4= new NetStream(netConnection4);
                    netStream4.bufferTime=2;
                    netStream4.client = new Object();
 video4 = new Video(320,240);
                    video4.smoothing = true;
                    video4.attachNetStream(netStream4);
```

5) Now we have to add the codes for handling the events when the marker will be displayed.
   Go to Line number 233 and add the following code.

```
if(id==3)
          {

          netStream4.play("<name of videofile>.flv");
          netStream4.addEventListener(NetStatusEvent.NET_STATUS,
onStatusEvent);
          vm4= new VideoStreamMaterial(video4, netStream4,true,true);
          plane.material = new WireframeMaterial();
           plane.material=vm4;

          }
```

6) Now,go to line number 274 and add the following code. This is because we have to pause the code whenever the marker is removed from the scene.

```
if(id==3)
{
           netStream4.togglePause();
           trace("click");
          plane.material=vm4;

}
```

8)line number 338

```
if(id==3)
                              {
                                        netStream4.togglePause();
                                        trace("paused");

                              }
```

9)Now,the final code snippet is to add the buffering and end of video event handling statements

For this go to Line number 468

```
          if(current_marker==3)
                 {
        trace((netStream4.bytesLoaded/netStream4.bytesTotal)*100);
        if((netStream4.bytesLoaded/netStream4.bytesTotal) <=0.04)
        {
trace((netStream4.bytesLoaded/netStream4.bytesTotal)*100);
        }
                        else
                        {

                         bufferbit=1;

                         trace("buffering over");

                        }

          if(stat.info.code == "NetStream.Play.Stop" &&bufferbit==1)
                        {
                               netStream4.seek(0);
                               netStream4.play("<name of video>.flv");

                        }
           }
```

Conclusion

The Netstream object has some problems and there are a lot of issues with videos. However, this implementation takes care of those issues by using Netstream.togglepause() function rather than the play and pause functionality.

Support and contact

If you have any difficulty in implementing the code. Please contact me at rahul.budhiraja.dark@gmail.com. I shall be extremely glad to help you.

Cheers!

Rahul