Web Multimedia Lesson 3 Assignment – Video Thumbnails

Using the completed video player inClass project, enhance the video player app by adding the feature of video thumbnails that appear as the user moves the mouse along the progress bar (similar to the effect seen with YouTube videos).

Check out the short video in E360 that shows what the completed project should look like.

You will need to do some research for this using the provided resources. Ready? Here we go…

Step 1: Creating your thumbnail images using ffmpeg

Create separate thumbnails images for each 1 second of your video using the tool **ffmpeg**.

This will give us 92 thumbnail images for our video which should suffice for our purposes. The thumbnail images should use the naming convention of **thumb*x*.jpg** where ***x*** will be a number (1-92) representing the seconds value into the video (eg thumb1.jpg, thumb2.jpg, …, thumb92.jpg).

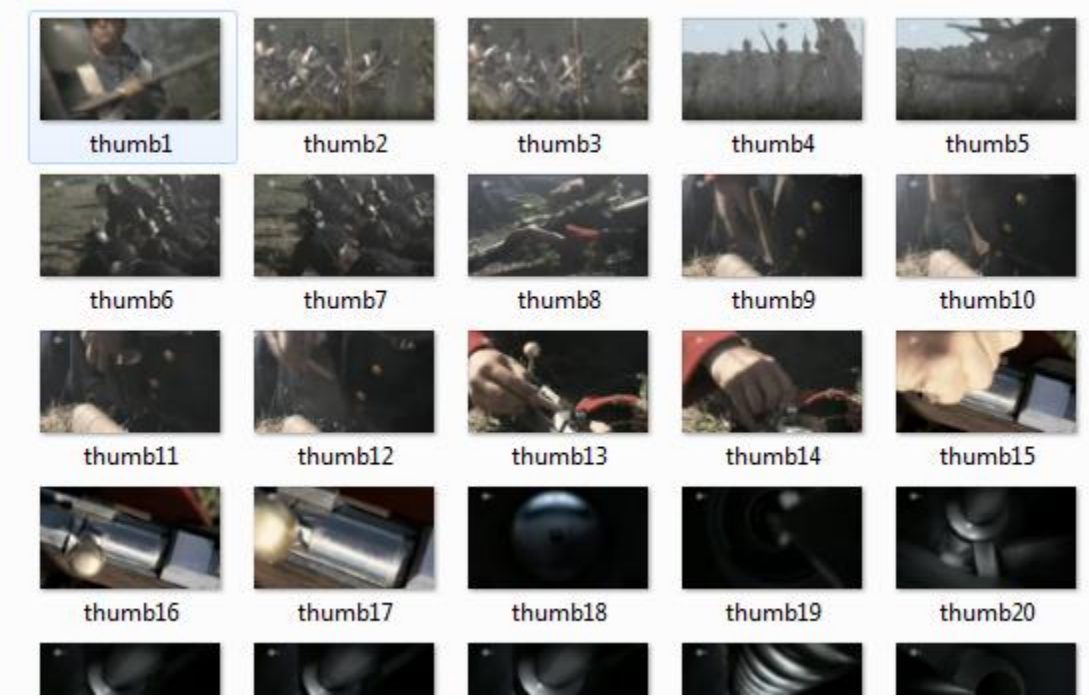
Go to https://www.ffmpeg.org/download.html to download the **ffmpeg** tool.

Check out https://trac.ffmpeg.org/wiki/Create%20a%20thumbnail%20image%20every%20X%20seconds%20of%20the%20video to see how to run the terminal command **ffmpeg** you need to produce a thumbnail image every X seconds from your video. It will be the first example under the ***fps video filter*** section. Of course, you will need to give it your video file (rather than input.flv) and change the output from out%d.png to **thumb%d.jpg**.

Note that terminal commands need to run from a DOS prompt in Windows (go to Start and type ***cmd*** to get a terminal window) or a Unix terminal prompt in MacOS using the appropriate paths for where you downloaded ffmpeg to and to where your video file is. Let me know if you need help with this.

Store your created thumbnail images in a folder in your project folder named **images/thumbs**.

They should look like this…



Step 2: Update your .html and .css files

In your *videoCustomControls.html* file add the following lines of code:

Add the following line inside your **<video>** tag after the second **<source>** tag:

<track src="thumbs.vtt" kind="metadata" default>

Note: you will need to create the **thumbs.vtt** file *manually* to hold your thumbnail images data carefully using the following format. It should be located in the same folder as your HTML file.

What is the .vtt file used for?

Web Video Text Tracks Format (**WebVTT**) is a format for displaying timed text tracks (such as subtitles or captions) using the <track> element. The primary purpose of**WebVTT files** is to add text overlays to a <video>. **WebVTT** is a text based format, which must be encoded using UTF-8. We will be setting this up in our **thumbs.vtt** file and using it to add thumbnail image overlays for our video.

WEBVTT

00:00.000 --> 00:01.100
images/thumbs/thumb1.jpg

00:01.100 --> 00:02.100
images/thumbs/thumb2.jpg

00:02.100 --> 00:03.100
images/thumbs/thumb3.jpg
.
.
.
01:30.100 --> 01:31.100
images/thumbs/thumb91.jpg

01:31.100 --> 01:32.100
images/thumbs/thumb92.jpg

Note the pattern of the time values for each thumbnail image above going up by one second for each thumbnail image…

Add the following line as the first tag inside ** **:

```
<div class="showThumbs"><span class="thumb"></span></div>
```

You will be using the .showThumbs div as a container for holding the video thumbnails and the .thumb span

In your *controls.css* file add the following CSS style rules right after the style rule for **#controls**:

```
.showThumbs {
  width: 100%;
  height: 70px;
  position: absolute;
  top: -80px;
}

.thumb {
  display: block;
  position: absolute;
  top: 0;
```

```
    left: 0;
    width: 120px;
    height: 70px;
}
```

Step 3: Update controls.js to implement video thumbnails feature

In your controls.js file's **initializePlayer** function:

1) Get a reference to *span.thumb* storing it in a global variable named **thumbnail**

2) Using plain JavaScript(no jQuery), create event listeners for the ***mouseenter*** and ***mouseleave*** events on the progress bar that set the CSS *display* attribute for *span.thumb* to values of '**block**' and '**none**' respectively.

3) The rest of the coding for this feature will be in another event listener function you create on the progress bar for the event ***mousemove***.

   Make the listener function an anonymous function which should have the code to set up and show the appropriate thumbnail image based on where the mouse is on the progress bar.

   Refer to the code example under the Preview Thumbnails section in the web page at https://hacks.mozilla.org/2014/08/building-interactive-html5-videos. Be aware that there will be a few differences between the example code shown there and what we are doing. Namely:

   a) Rather than their obscurely named **p** variable, use the variable name **mousePos** instead as their **p** is referring to the position of the mouse relative to the left edge of the progress bar when the mousemove event occurred.

   b) Rather than their hardcoded value of ***480***, use the progress bar's *offset width* property to get the correct width of the progress bar as needed for this effect.

c) Use the *offsetX* mouse event property rather than *pageX* as then you won't need to subtract off the distance from the left edge of the progress bar to the left edge of the page.

d) Rather than their obscurely named **c** variable, use the variable name **cuesList** instead as this is more accurately what that variable refers to.

   Check out the following two web pages to learn more about **TextTrack** interface objects (whose data was populated when our **thumbs.vtt** file was loaded) and their **cues** property which is a **TextTrackCueList** object which contains all of the track's cues.

   https://developer.mozilla.org/en-US/docs/Web/API/TextTrack

   https://msdn.microsoft.com/library/Hh772674

e) There is no need for use to use the split() method on our cue's text like they do since they are using a single stitched together image while you are using separate images for each "frame" of the video. They are using split() to break apart the text to position their single background image. You will simply use the text property as the path to be used as a *background image* on your span.thumb (thumbnail).

f) Once you have span.thumb's background image set it is just a matter of correctly *positioning* it to reflect where the mouse was on the progress bar when it was moved. Set span.thumb's (thumbnail) *left* property to do this based on where the mouse is horizontally on the progress bar when it was moved keeping in mind you will need to take into account the width of the thumbnail image to center it correctly on the mouse's x-position.

Be sure to name your project folder **Lesson03_videoThumbnails_*yournamehere*** and zip it up before you submit it.