Senior Project Design Proposal

**Student Name:** Mark Neitzel

**Advisor Name:** Dr. Hayes

**Expected Date of Graduation:** December 2020

**Description of Project:** People want to interact with the content of a recorded video presentation more directly if the recording included the semantic structure of the content rather than just raster images. For example, viewers may copy and paste the text from the slides, clicking on links, giving feedback, jumping to logical segments. By creating a system to record and playback video-like content with the semantic meaning intact via HTML will meet this need. The implementation will include adding features to a recorded video presentation such as copy and paste text from the slides, clicking on links, giving feedback, and jumping to logical segments.

**Proposed Implementation Language(s):** HTML/Javascript

**Any software/equipment needed:** [GitHub](https://github.com/), [Reveal.js](file:///Users/MarkNeitzel/Documents/Charleston%20Southern%20University/Spring%202020/CSCI%20497%2003/Reveal.js), [Visual Studio Code](https://code.visualstudio.com/), and [Visual Studio Code Storyteller Plugin by markm208 on GitHub](https://github.com/markm208/storyteller)

**Motivation and Problem Statement:**

If there was a program that allows people to highlight specific text from a video slide presentation, they would be able to use that text in their own notes so that they can view them later on. People want to interact with the content of a video presentation more directly by creating a system to record and playback video-like content with the semantic meaning intact via HTML will meet this need. Ideally, users will be able to interact with the content of a “video” presentation by copying text, clicking on links, and answer prompts for information (e.g., questions check understanding). However, the semantic meaning and structure of content is lost during the usual video recording process, which only captures a sequence of images. The demand for these features in education is apparent based on decisions with fellow students and teachers. YouTube provides partial support for these features through “cards”, captions, and description links, which indicates that these features are helpful. With the use of Reveal.js, Visual Studio Code, and VS Code storyteller by markm208 on GitHub, I will be able to design a way for people to be able to review the notes as they were presented in the slides by being able to copy and paste the exact text from the slides into their own notes.

**Outline of Future Research Efforts:**

*How will the student complete his/her project – what deliverables can be expected?*

I will complete my project by doing some research on how to use Reveal.js and all of the features that it involves and how to use the Visual Studio Code Storyteller Plugin and all of the features that are involved with it.

Another feature that I need to do some research on is identifying ways to add audio to a Storyteller file in Visual Studio Code.

Some deliverables that can be expected are:

* A GitHub repository
* A “video” presentation player code created using Visual Studio Code
* Requirements Specification
* Test Plan Specification

**Schedule:**

*A rough idea of the dates of the deliverables for CSCI 498, 499*

February 24, 2020 - Start creating GitHub repository

February 26, 2020 - Start working on Test Plan and Requirements

February 28, 2020 - Start setting up “video” presentation code from Storyteller Plugin for Visual Studio Code by markm208

March 6, 2020 - Start working on implementing “video” presentation code from Storyteller Plugin

March 16, 2020 - Add an HTML feature, such as, search for a word that is used within the video at a particular time

March 23, 2020 - Add another HTML feature, such as, presenters may get feedback on understand through interaction (analytics)

March 30, 2020 - Add another HTML feature, such as, nest videos within other videos (avoiding copyright issues by using original sources)

April 6, 2020 - Add another HTML feature, such as, click directly on links to connect with sources

April 13, 2020 - Have the majority of the “video” presentation code working

April 20, 2020 - Work out the kinks and errors in the code

April 27, 2020 - Test and make sure that the code for the “video” presentation works properly

April 27, 2020 - Turn in completed Test Plan Specification and Requirements Specification documents

Finish beta version

Complete testing on beta version

December 1, 2020 - Analyze results from testing and make changes (bug fixes, design changes, etc.)

November 21, 2020 - Complete project documentation

December 1, 2020 - Defend project