MULTIMEDIA WEB PAGE DEVELOPMENT

184514

INTRODUCTION TO MULTIMEDIA - BSC COMPUTER SCIENCE 14/11/2018

TABLE OF CONTENTS

1 INTRODUCTION

This report outlines the work carried out as part of the Multimedia web page and vector development coursework for the module, Introduction to multimedia. This report will include the web development techniques used to develop and structure the website, along with the vector functions used to produce the final animation. This report will demonstrate the vector functions used in the animation to meet the specified requirements and how the functions where implemented, alongside an overview of the creation of the website and the extra functionality it contains. When testing the website, the recommendation Firefox, as the animations all run smoothly, and the video and audio elements have built in volume control. Some of the sites features have not been tested on IE or Safari, so avoid using these browsers.

2 DESIGN AND DEVELOPMENT

When developing the website and the vector animation, Notepad++ was the program used to code the majority of the site. Another program used, but not as often was Dreamweaver, and was only used to ensure that the CSS remained consistent and throughout all the pages. As Dreamweaver was a new experience, an Adobe tutorial [1] was used to help understand and become familiar with the features. Furthermore for the vector animation, Raphael Live [2] was used as a prototyping area for vector functions.

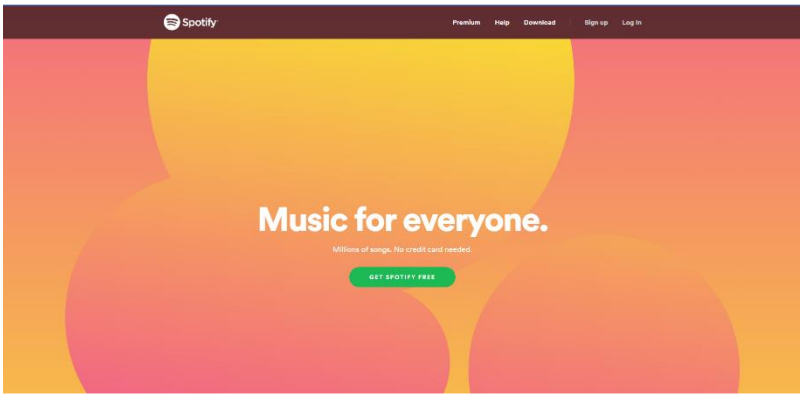
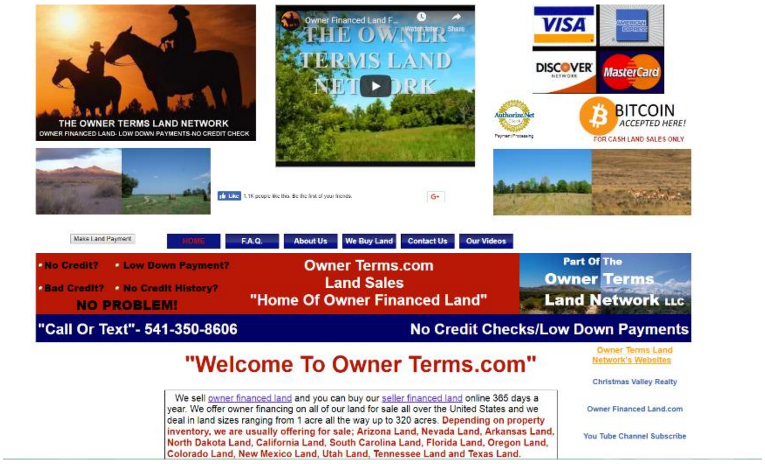


Figure I – Spotify's website with a clear and easy to use site.

2.1 WEBSITE DEVELOPMENT

When designing the site, minimalist pages such as Spotify where used to base the designs off of. The reason a minimalist style was eventually developed was due to the fact that when a website is cluttered with information, links and interactive features, the overall usability of the website decreases along with the user experience and ease of navigation. A good example of this is Spotify. This is explained well in a Site Point article [3]. In addition, the choice of a simplistic site design is useful as it is not difficult to optimize for mobiles and tablets.



We sell owner financed land and you can buy our seller financed land online 365 days a year. We offer owner financing on all of our land for sale all over the United States and we deal in land sizes ranging from 1 acre all the way up to 320 acres. Depending on property inventory, we are usually offering for sale; Arizona Land, Nevada Land, Arkansas Land, North Dakota Land, California Land, South Carolina Land, Florida Land, Oregon Land, Colorado Land, New Mexico Land, Utah Land, Tennessee Land and Texas Land.

Figure II – OwnerTerms.com, a bad example for how to structure and present a website.

The colour scheme of white and light blue used for website was mainly chosen due to the good compatibility between colours, as well as the relevancy to the images of boats used on the home page to symbolise the ocean. Moreover, on all pages, there is an option within the footer to change the style of the site which can be used by users with dyslexia to enable a more user-friendly experience. By doing some research, it was conveyed that user's with dyslexia tend to have trouble interacting with sites with black text, on a white background**.** In order to combat this, the site has a feature that enables the user to switch to a dyslexia friendly layout, black background and yellow text, which was recognised to be the friendliest colour scheme [4].

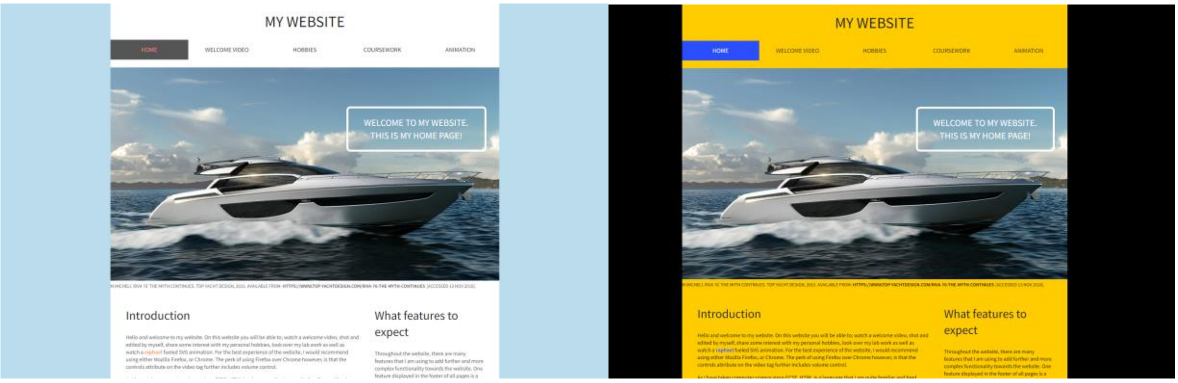


Figure C– My Website's white and blue colour scheme compared to the dyslexic colour scheme. Home page includes title, introduction text and an image properly attributed.

The navigation bar used for the site was the simplistic yet most effective navigation bar used for the site that was tested. This may be because it is consistent between all pages, large and uses features like rollover (hover) with active tab, where the page currently on is highlighted in the navigation bar, to ensure good overall usability. These features are helpful as with rollover buttons, the user is able to see which tab they are currently hovering over, which may help users that are visually impaired, and active tab highlighting to ensure that the user is aware of what page they are currently on. Moreover, when the page is on a smaller screen, for example a tablet, the navigation bar collapses for usability, as on some devices the navigation tabs may be too small to click on. This feature was developed with the aid of the Adobe tutorial [1].

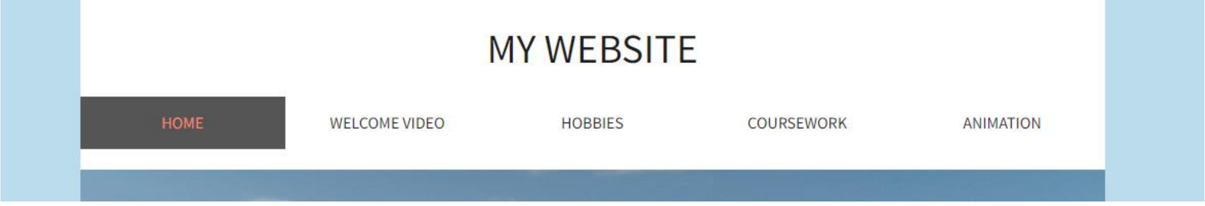


Figure D – Navigation bar, with the current page highlighted, and if hovered over, rollover buttons.

The content structured on the website is all centred and when compressed for a smaller screen, remains centred, with the same margin and padding around the images and text. This layout was used to keep the site clear and concise, with content in clear separate divisions, rather than all the content thrown into a main section of the website. Centred headings are used to divide the divisions and give the content in the site context along with structure. For the video on the welcome page, the video is the same width as the overall wrapper, due to the fact that when a user wants to watch a video, they would prefer the largest resolution for the best quality. There is a full screen feature along with volume control when interacted with on Firefox.

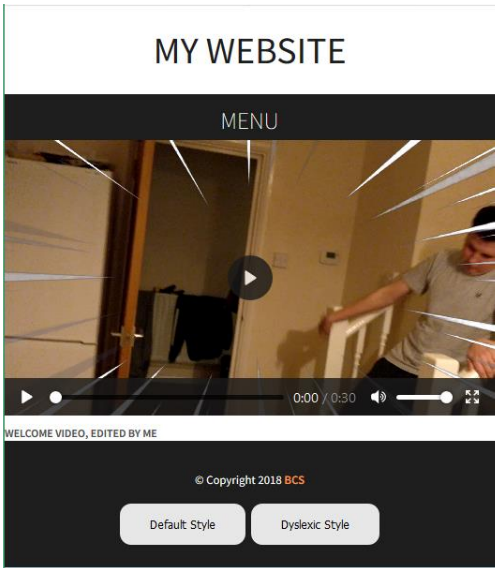


Figure E – Responsive Welcome video page.

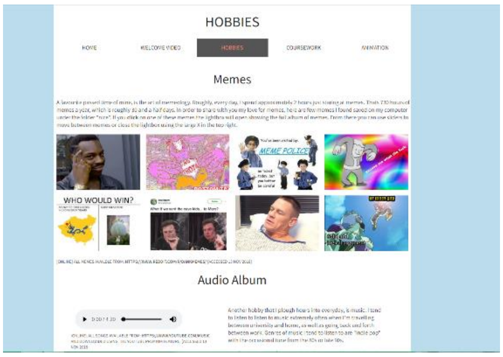


Figure F – Hobbies page with images and text.

The work carried out by Dreamweaver, was mainly styling the site to enable that it was responsive. This was mainly done by using the "max-width" attribute for images in the CSS so when the page resizes, the content resizes to ensure it fits on the screen. By using the scalable units, "%" and "em" the text along with any other content would decrease / increase in size, depending on the resizing of the screen. A web design article [5] was used for research purposes on the advantages of these units along with examples, which helped with implementation. HTML5 was primarily used to structure the content on the web pages, along with the additional features such as video, audio and PDF viewing. For using the video and audio element, W3Schools was used to give an understanding of the element syntax along with example code [6, 7]. For the object tag, a tutorial on YouTube was used to understand how the element works, and how to implement a pdf into a website using it. [8] Finally CSS was used to style the content of the website, alongside making the site responsive. To aid with understanding this, both the Adobe tutorial [1] and a YouTube tutorial [9] were used to give a greater understanding on this.

Additional elements included in the website which would come under Added Functionality which are included in my site includes: rollover buttons, JavaScript, additional multimedia components tied with a navigable photo (Lightbox) and audio album, and basic switchable CSS styles. Rollover buttons was the only additional feature that did not require further research into, as the hover attribute was a feature I was familiar with. The w3schools tutorial however was used to just browse at the values that can be set with this attribute. [10]

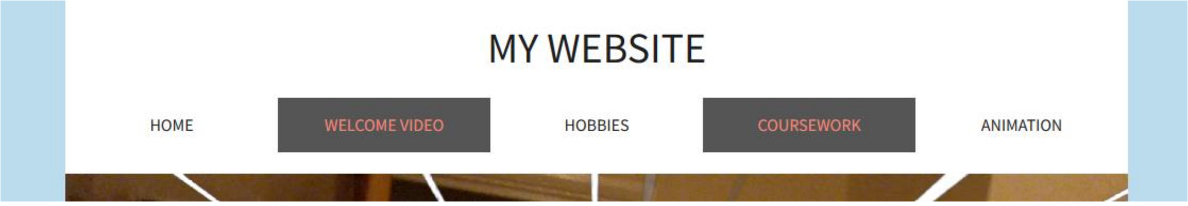
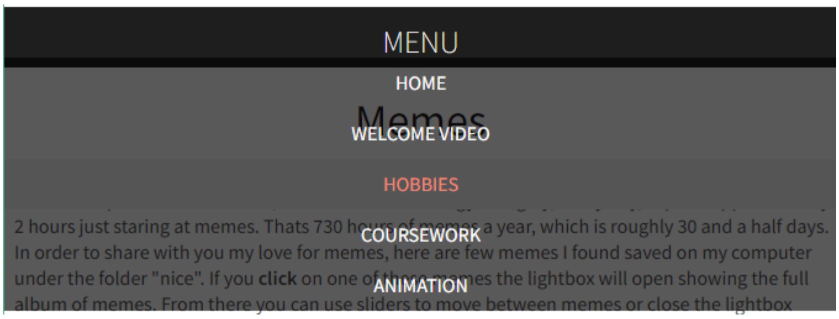


Figure G – Rollover Buttons (Cursor is hidden from print screen, but would be hovering over coursework)

JavaScript was a helpful feature that allowed for the smooth transition in the navigation bar when collapsed, navigable photo and audio albums, and the switching CSS feature. Before using a tutorials, independent research on JavaScript was done to gain a better understanding of the syntax of the language along with basic examples. The tutorial used was the YouTube video, JavaScript Basics Tutorial [11].

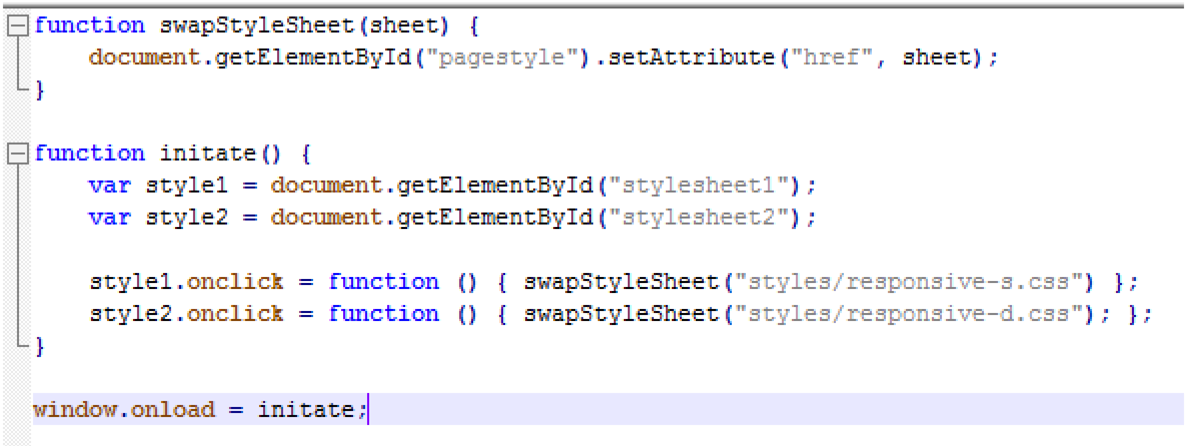
The navigation transition JavaScript was taken from the Adobe tutorial [1] with minor tweaking to suit my websites preferences.



2 hours just staring at memes. Thats 730 hours of memes a year, which is roughly 30 and a half days. In order to share with you my love for memes, here are few memes I found saved on my computer under the folder "nice". If you click on one of the memes the lightbox will open showing the full album of memes. From there you can use sliders to move between memes or close the lightbox

Figure H – Navigation menu collapses properly when screen size is reduced to below 700px wide.

This feature is overall just a nice element to give the website a more professional element to it. The switchable CSS element was developed with the help of another YouTube tutorial - Change Style Sheet Using JavaScript Tutorial CSS Swap Stylesheet – [12].



```
function swapStyleSheet(sheet) {
    document.getElementById("pagestyle").setAttribute("href", sheet);
}

function initate() {
    var style1 = document.getElementById("stylesheet1");
    var style2 = document.getElementById("stylesheet2");

    style1.onclick = function () { swapStyleSheet("styles/responsive-s.css") };
    style2.onclick = function () { swapStyleSheet("styles/responsive-d.css"); };
}

window.onload = initate;
```

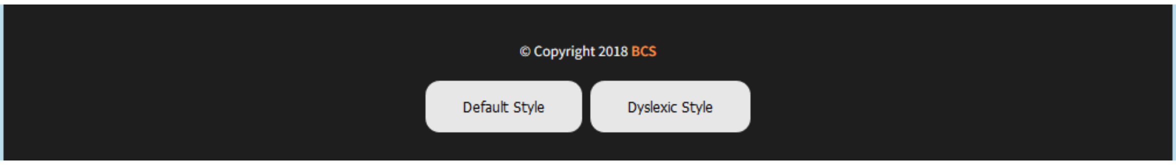
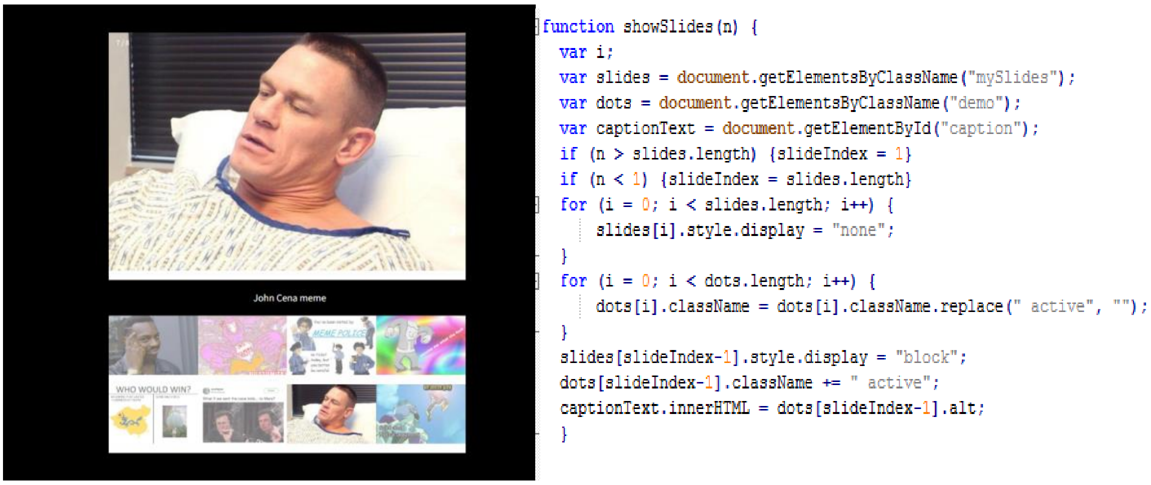


Figure I –Buttons that change the CSS stylesheet

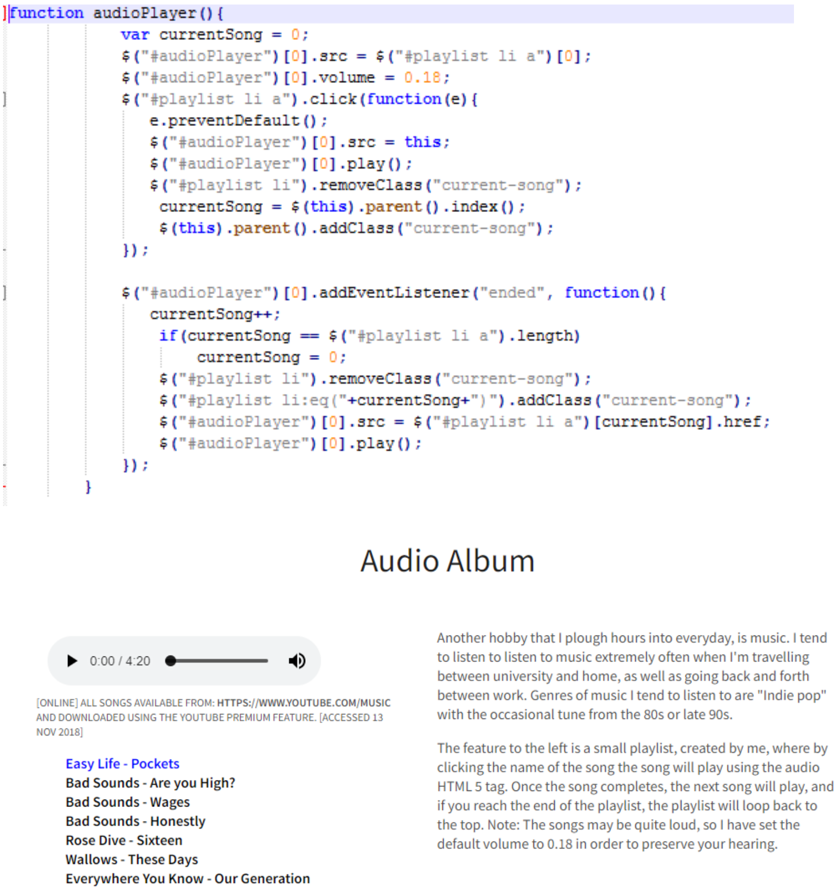
The feature was implemented for dyslexic users, who may find the colour scheme difficult to use, so a more user-friendly colour scheme can be switched to. The use of the lightbox feature was developed using HTML5 and JavaScript with some assistance of W3Schools [13] and is present on the hobbies page. This additional feature gives the site a more user interactive-aspect enabling for further functionality between site and user. The lightbox images used were all collected from Reddit [14] over the last few days, just to be used as examples for what kind of content can be displayed in a lightbox. For this feature, CSS, HTML5 and JavaScript are all used together.



```
function showSlides(n) {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("demo");
  var captionText = document.getElementById("caption");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
      slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
      dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  captionText.innerHTML = dots[slideIndex-1].alt;
  }
```

Figure J – Lightbox feature on Hobbies page.

CSS is used to style the lightbox's features such as the next and previous slider buttons, the close button as well as the surrounding box feature. HTML5 was used to implement structure to the lightbox element, along with providing the content to be displayed. JavaScript is finally used to display the lightbox feature along with moving between images in the album. This script and CSS used was based off of an example from W3Schools. The audio music playlist also on the hobbies page is the final additional element using JavaScript on the website. For this feature, all the songs where downloaded and saved from YouTube, using the YouTube premium download feature [15].



```
function audioPlayer(){
            var currentSong = 0;
            $("#audioPlayer")[0].src = $("#playlist li a")[0];
            $("#audioPlayer")[0].volume = 0.18;
            $("#playlist li a").click(function(e){
               e.preventDefault();
               $("#audioPlayer")[0].src = this;
               $("#audioPlayer")[0].play();
               $("#playlist li").removeClass("current-song");
                currentSong = $(this).parent().index();
                $(this).parent().addClass("current-song");
            });

            $("#audioPlayer")[0].addEventListener("ended", function(){
               currentSong++;
                if(currentSong == $("#playlist li a").length)
                    currentSong = 0;
                $("#playlist li").removeClass("current-song");
                $("#playlist li:eq("+currentSong+")").addClass("current-song");
                $("#audioPlayer")[0].src = $("#playlist li a")[currentSong].href;
                $("#audioPlayer")[0].play();
            });
        }
```

Audio Album

[ONLINE] ALL SONGS AVAILABLE FROM: HTTPS://WWW.YOUTUBE.COM/MUSIC AND DOWNLOADED USING THE YOUTUBE PREMIUM FEATURE. [ACCESSED 13 NOV 2018]

Easy Life - Pockets
Bad Sounds - Are you High?
Bad Sounds - Wages
Bad Sounds - Honestly
Rose Dive - Sixteen
Wallows - These Days
Everywhere You Know - Our Generation

Another hobby that I plough hours into everyday, is music. I tend to listen to listen to music extremely often when I'm travelling between university and home, as well as going back and forth between work. Genres of music I tend to listen to are "Indie pop" with the occasional tune from the 80s or late 90s.

The feature to the left is a small playlist, created by me, where by clicking the name of the song the song will play using the audio HTML 5 tag. Once the song completes, the next song will play, and if you reach the end of the playlist, the playlist will loop back to the top. Note: The songs may be quite loud, so I have set the default volume to 0.18 in order to preserve your hearing.

Figure K – Audio Album on Hobbies page.

To create this playlist of songs with additional functionality such as being able to pause, play, decrease/increase the volume as well as choose the song, with further automated features such as skipping to the next song once the previous one is complete, and restarting the playlist once the whole playlist has been played through, a YouTube tutorial was used - Insert Audio into a Website Using HTML5 [16]. I believed this would be a cool feature to add to the website as it is quite a unique way of presenting a hobby, as it allows the user to interact with something that you enjoy. This feature is also easily expandable and can use a much larger list of songs for a playlist.

2.2 Animation Development

When designing and developing the animation, most of the prototypes where developed using Raphael Live [2], since the site had helpful examples of functions. Another site that was heavily used in the development of the animation was Raphael – JavaScript library [17]. This was extremely helpful as it defines all the functions and attributes available when using the Raphael library. Functions such as hide() and show() where found using this site and enabled me to make quite a complex animation. Additionally, a two part YouTube series, Raphael Tutorial [18], helped with syntax, especially when a vector performs more than one action, i.e. transforms, then translates.

When developing the vector animation, the aim was to meet all the basic functionality in the project brief, and some of the advanced functionality stated in the brief. When the user loads the animation page, all the have to do is click the link stated "Play Button" **[I]**

>>>>>>> PLAY BUTTON <<<<<<<

Figure I – Play button

In order to start the animation. When the button is pressed, the animation begins and the audio begins to play **[II].**

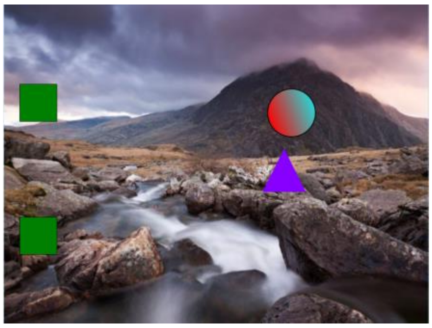


Figure II – Animation starts when button is pressed

The animation contains a bitmap image as the background [19] **[III]** and multiple vector shapes throughout the animation **[IV].**

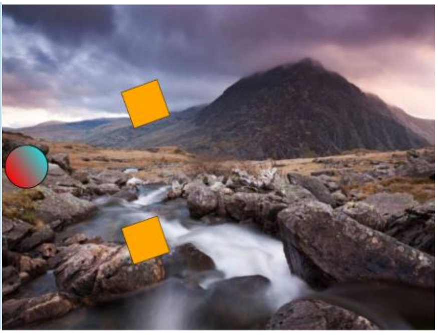


Figure III – Bitmap image used as a background   *Figure IV – Vector shapes*

This feature was stated as an additional functionality in the specification. In the animation, there is more than one instance of a size change. When going in between the square gates, the circle initially enlarges at a scale factor of 2, and then on the second gate reduces at a scale factor of 0.6 **[V].**

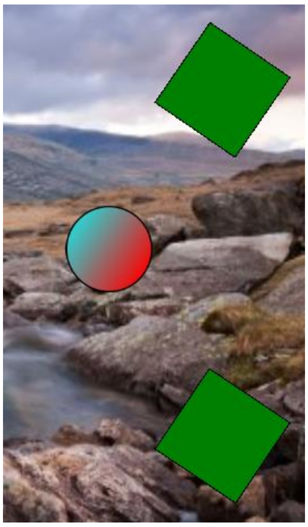


Figure IV/VI – Size Reduced (More clear if you watch the animation) while moving

The next animation function is the motion of the vector objects. The main vector shape, the circle, moves throughout the animation, as well as the square gates once the ball passes through them **[VI].** Many colour changes occur in the animation, with the first being the triangle the circle climbs over **[VII]** and also the ball in the final scene when the user hovers over it the animation lasts for around 20 seconds, and shows reasonable synchronisation with the track, as the both end at roughly the same time.

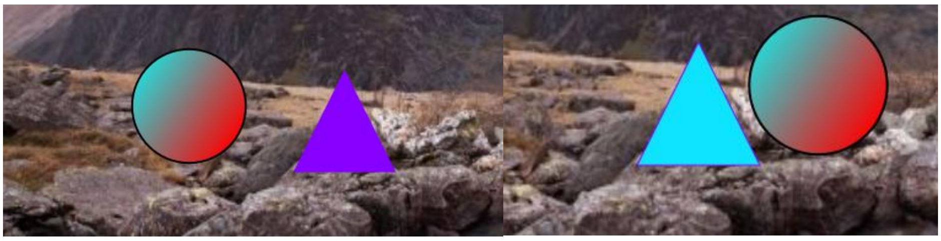


Figure VII - Triangle changes colour

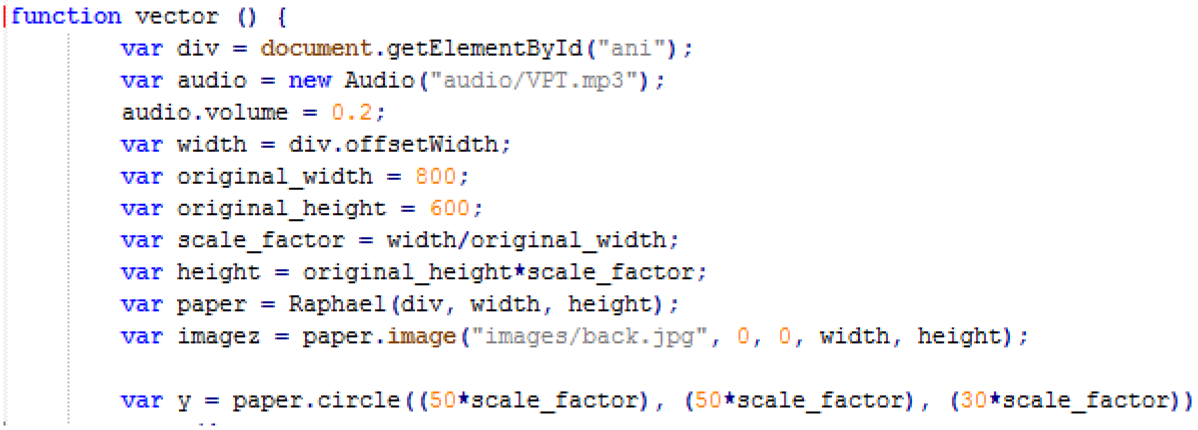
Finally, the animation shows decent technical execution, as no images contained are of incorrect resolution. All of these functions used where researched from the JavaScript library then tested with Raphael Live [2].

With additional functionality, the animation contains a “play button” that the clicks in order to start the animation and the audio simultaneously. This was done by using the onclick attribute in the <a> tag. The reason an <a> tag was used over an actual button, was that I found the text based button to look nicer in styling **[VIII].**

>>>>>>> PLAY BUTTON <<<<<<<

Figure VIII – Play button – when hovered over goes bolder and underlines.

Further, the animation has some realistic behaviour, for example after the ball climbs the triangle, it rolls back down the other side, like a ball would. In addition, the vector animation has some interactivity with the user. In the final scene, the user is prompted to attempt to hover over the ball. If the user succeeds, the ball will change colour. If the user hovers over the ball multiple times, the ball will continue to change colour. Another additional feature with the vector animation that has been implemented, was that it is scalable and responsive. For this to happen, the user must re-size the browser, then refresh the browser, so the JavaScript can get the new width and height dimensions and can scale all objects accordingly [IX].



```
function vector () {
        var div = document.getElementById("ani");
        var audio = new Audio("audio/VPT.mp3");
        audio.volume = 0.2;
        var width = div.offsetWidth;
        var original_width = 800;
        var original_height = 600;
        var scale_factor = width/original_width;
        var height = original_height*scale_factor;
        var paper = Raphael(div, width, height);
        var imagez = paper.image("images/back.jpg", 0, 0, width, height);

        var y = paper.circle((50*scale_factor), (50*scale_factor), (30*scale_factor))
```

Figure IX – A variable called scale factor is used to make the animation responsive. The only problem with this is JS requires the page to be refreshed in order o get the new width and height values.

This feature was read about and used from “Creating Responsive and Interactive Animations with Raphael.js” [20].

3 Conclusion

Overall, the project was a success as all fundamental aims were completed such as the basic functionality of the website and the basic functions within the animation, along with most of the additional functionality implemented within website such as the lightbox feature and audio playlist. With the website’s functionality, further work could be put towards the styling features to allow for more user-friendly styles, to increase the accessibility of the content, this could be a text to speech interpreter for the blind, increased font-sizing for the long-sighted or even further basic styling for conditions similar to dyslexia. For the animation, lots of additional features could be worked on and implemented, such as making the animation include a plot and story-line, using features like beat-detection or even adding video controls such as pause and rewind as well as volume controls for the music

4 References

1 - D. Powers. Project overview and site setup. Adobe, 2015. [Online] Available from: https://helpx.adobe.com/dreamweaver/how-to/make-website-pt1-site-setup.html [Accessed 13th Nov 2018]

2 - R. Jones. Raphael Live. Craic, 2016. [Online] Available from: http://www.craic.com/code_examples/raphael_live/raphael_live.html# [Accessed 13th Nov 2018]

3 – R. Jain. When less is more – Why Minimalism STILL Rules the Web. Sitepoint, 2015. [Online] Available from: https://www.sitepoint.com/less-minimalist-websites-still-rule/ [Accessed 13th Nov 2018]

4 - Dyslexia: colour and contrast. Stack Exchange, 2015. [Online] Available from: https://ux.stackexchange.com/questions/73916/what-is-the-most-dyslexia-friendly-colour-combination [Accessed 13th Nov 2018]

5 – K. Schaeffer. CSS Font-Size: em vs. px vs. pt vs. percent. Kyle Schaeffer, 2008. [Online] Available from: https://kyleschaeffer.com/development/css-font-size-em-vs-px-vs-pt-vs/ [Accessed 13th Nov 2018]

6 – HTML 5 Video. W3Schools, 2018. [Online] Available from: https://www.w3schools.com/html/html5_video.asp [Accessed 13th Nov 2018]

7- HTML 5 Audio. W3Schools, 2018. [Online] Available from: https://www.w3schools.com/html/html5_audio.asp [Accessed 13th Nov 2018]

8 - How to embed a pdf file into your HTML file and make it responsive. YouTube, 2016. [Online] Available from: https://www.youtube.com/watch?v=ZLLgzD2bae4 [Accessed 13th Nov 2018]

9 - CSS: hover Selector. W3Schools, 2018 [Online] Available from: https://www.youtube.com/watch?v=Wm6CUkswsNw [Accessed 13th Nov 2018]

10 - Build An HTML5 Website With A Responsive Layout. YouTube, 2015. [Online] Available from: https://www.w3schools.com/cssref/sel_hover.asp [Accessed 13th Nov 2018]

11 – JavaScript Basics Tutorial. YouTube, 2018. [Online] Available from: https://www.youtube.com/watch?v=0QeFAE03qJc&vl=en [Accessed 13th Nov 2018]

12 - Change Style Sheet Using JavaScript Tutorial CSS Swap Stylesheet. YouTube, 2016. [Online] Available from: https://www.youtube.com/watch?v=_XAQH41rjio [Accessed 13th Nov 2018]

13 - How TO – Lightbox. W3Schools, 2018. [Online] Available from: https://www.w3schools.com/howto/howto_js_lightbox.asp [Accessed 13th Nov 2018]

14 – r/dankmemes, 2018 [Online] Available from: https://www.reddit.com/r/dankmemes/ [Accessed 13th Nov 2018]

15 – Using YouTube Premium benefits. Google Support, 2018. [Online] Available from: https://support.google.com/youtube/answer/6308116?hl=en-GB [Accessed 13th Nov 2018]

16 -Insert Audio into a Website Using HTML5. YouTube, 2016. [Online] Available from: https://www.youtube.com/watch?v=9F49XgzlZgA&index=4&list=LLKYe9oj9Lhed-UQeGgqOE3A [Accessed 13th Nov 2018]

17 – D. Baranovskiy. Raphaël—JavaScript Library [Online] Available from: http://dmitrybaranovskiy.github.io/raphael/ [Accessed 13th Nov 2018]

18 – Z. Rosser. Raphael Tutorial Part 1. YouTube, 2012. [Online] Available from: https://www.youtube.com/watch?v=IPy77zMsd0Y&index=7&t=0s&list=LLKYe9oj9Lhed-UQeGgqOE3A [Accessed 13th Nov 2018]

19 – A. Davies. Best UK landscape photography locations for a weekend break. Amateur Photographer, 2017. [Online] Available from: https://www.amateurphotographer.co.uk/latest/photo-news/best-uk-landscape-photography-locations-weekend-break-104470 [Accessed 13th Nov 2018]

20 – C. Youderian. Creating Responsive and Interactive Animations with Raphael.js. Specky boy, 2014. [Online] Available from: https://speckyboy.com/creating-responsive-interactive-animations-raphael-js-2/ [Accessed 13th Nov 2018]

4.1 Figures

A - https://www.spotify.com/uk/ [Accessed 13th Nov 2018]

B -https://www.ownerterms.com/ [Accessed 13th Nov 2018]

C, D – home.html [Accessed 13th Nov 2018]

E – Welcomevid.html [Accessed 14th Nov 2018]

F - Hobbies.html [Accessed 14th Nov 2018]

G - Welcomevid.html [Accessed 14th Nov 2018]

H, I - hobbies.html [Accessed 14th Nov 2018]

J - Hobbies.html + autoplayer.js [Accessed 14th Nov 2018]

K - Hobbies.html + lightbox.js [Accessed 14th Nov 2018]

I – IX - Animation.html + vectoran.js [Accessed 14th Nov 2018]