

**Senior Capstone
Visualization Project Report
CPSC 4910
Spring 2013**

Aju Skarla
Chris Lanham
Cody Sapp
Hallow Kareem
Liem Le
Vance Fontenot

I. PROJECT DESCRIPTION

The Visualization Group was charged with creating an interactive animation that educates users about DNS and DNS cache poisoning, and then host the animation on a website coded in HTML5. The primarily target audience is 2nd to 4th year college students.

II. TEAM MEMBERS AND THEIR ROLE

Aju Skaria is the Planning Manager. Chris Lanham is the Support Manager. Cody Sapp is the Team Leader. Hallow Karem is the Development Manager. Liem Le is the Quality and Process Manager. Vance Fontenot is the Audio and Poster Manager

III. REQUIREMENTS

Functional requirements:

1. The web site shall provide interactivity with users
2. The web site shall provide audio feeds to users
3. The web site shall provide quizzes to users
3. The web site animation scenes must be short
4. The web site scene selection must be available at all time

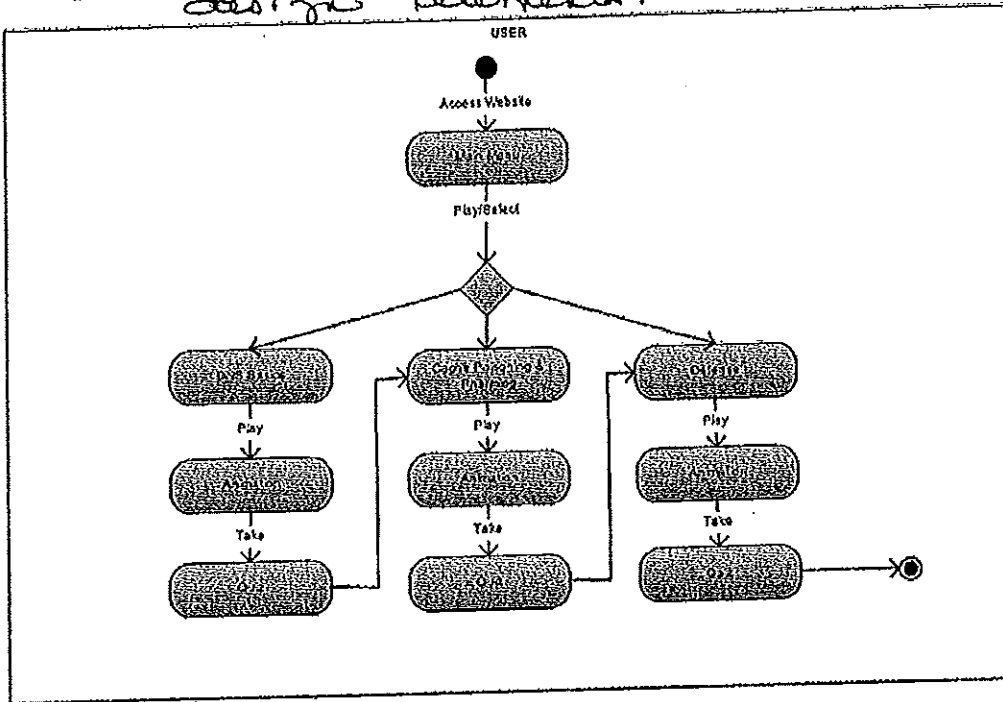
Nonfunctional requirements:

1. The web site browser must support HTML5
2. The web site browser must support JavaScript
3. The interactive animation must contain dropdown menus
4. The interactive animation must be easy to use
5. The interactive animation must be easy to understand by all users and the administrator
6. The web site start-up time should be fast
7. The web site response time should be fast

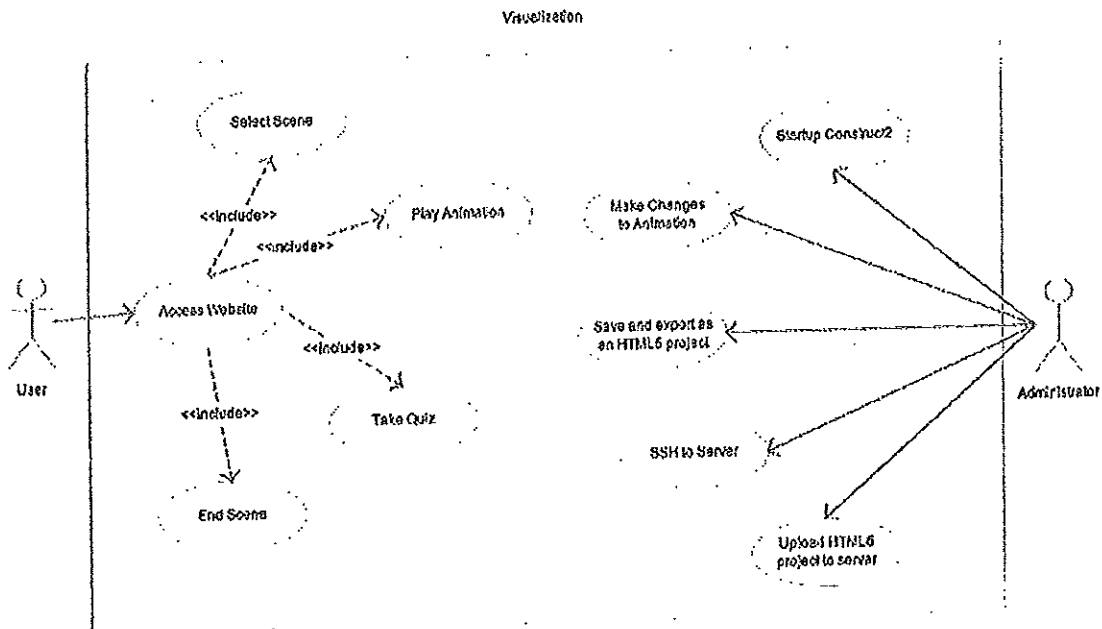
IV. DESIGN

State Diagram:

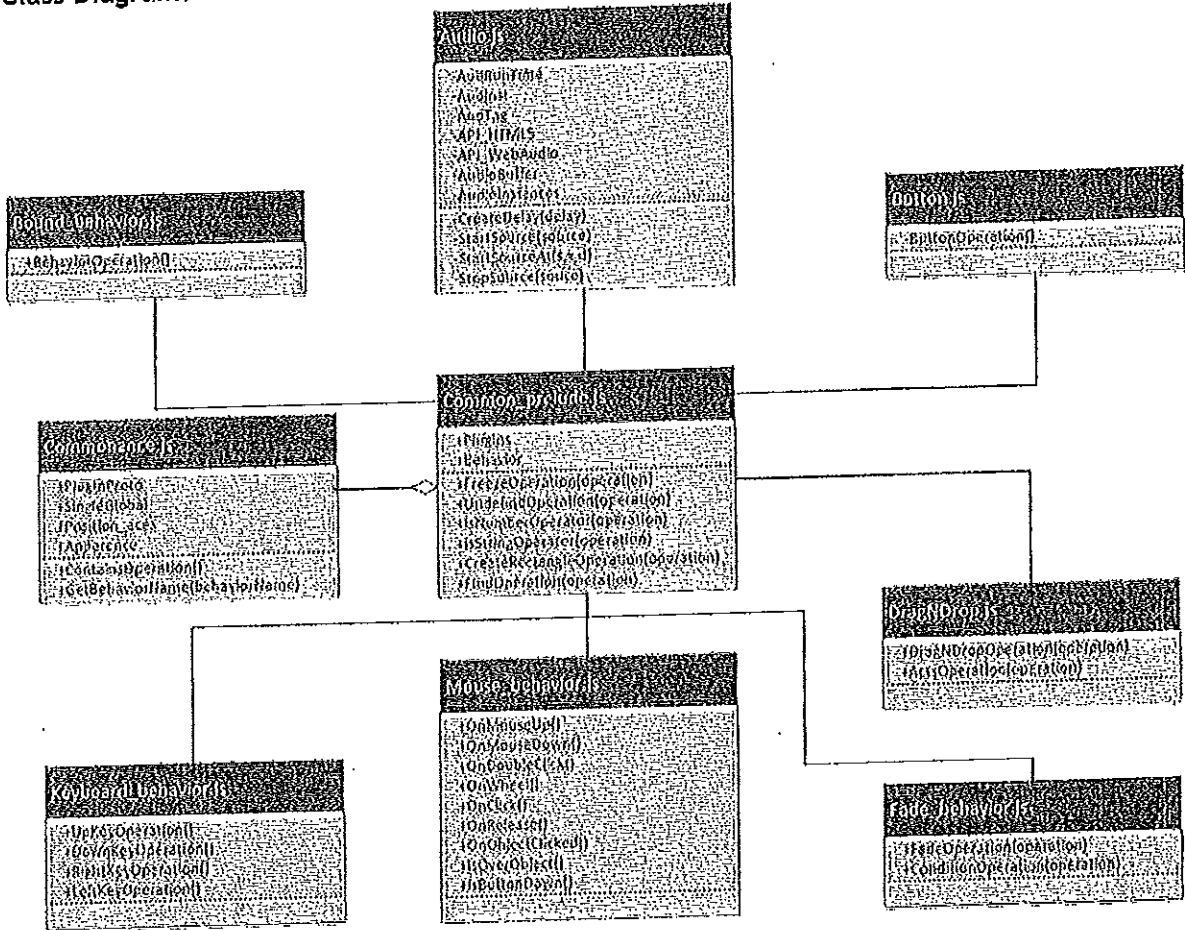
I used some words surrounding how you did this. What are the design elements.



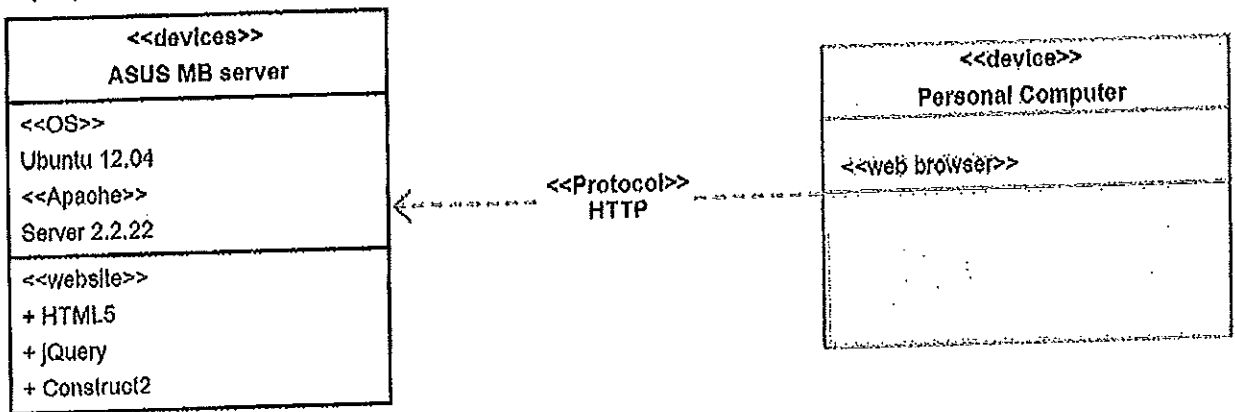
Use Case Diagram:



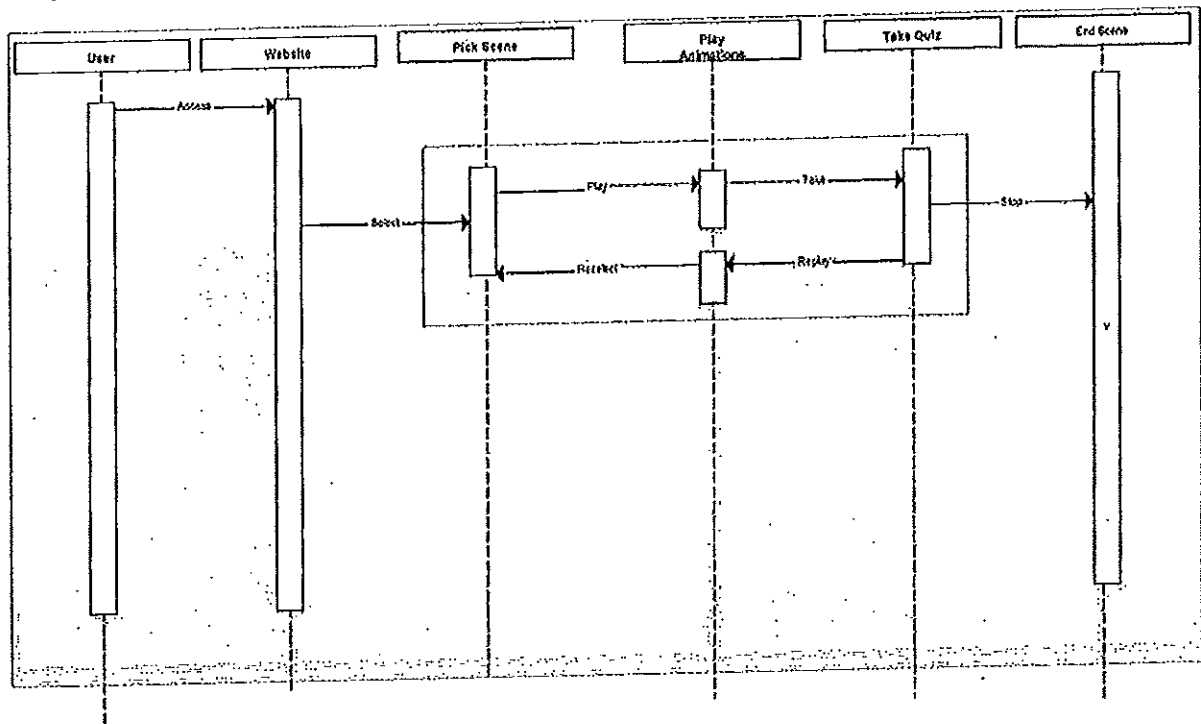
Class Diagram:



Deployment Diagram:



Sequence Diagram:



V. METHODOLOGY

The audio files were created using Audacity, and the images were created using Microsoft Paint. The interactive animation itself was created using Construct 2. The interactive animation is coded using JavaScript that Construct 2 creates when the project is exported. The animation is hosted on a server and run through a web browser. It works on every major web browser. The website that the animation is on is coded in HTML5.

The interactive animation requires the user to perform some kind of action in certain scenes. Some examples include moving an object to a certain point, answering a question, or choosing between two different websites to use for a demonstration. There are some scenes however that do not require any interaction with the user and will play completely out on their own.

This is the next methodology involved in the project - this is the methodology of the project itself



VI. IMPLEMENTATION

Setup a webserver using the desired operating system of choice (Linux with Apache is a widely used and easy to implement option). Apache 2 is the webserver that this animation has been tested on and proven to work best with so it is recommended to implement the web server using Apache 2. Although IIS, or NGINX should be able to host the website just fine (all documentation has been written with the specific implementation examples in Apache 2). For more information on how the Linux / Apache should be implemented please see the administrator manual.

VII. DEPLOYMENT

Not how to.

Take the zip file containing the HTML and JavaScript files and extract them to the web directory on the web server. For example, in apache extract these files in /var/www/<sitename> (see administration manual for more details on how this directory structure works).

If supporting browsers that are older legacy browsers be sure to see the administrator manual on how to configure MIME type overrides. The HTML5 project directory will also need to contain a .htaccess file (it can be named anything but it must be a .htaccess file type). The contents of the file needs to be the following:

```
AddType audio/ogg oga ogg
AddType audio/mp4 m4a
```

VIII. MAINTENANCE

If the interactive animation needs to be updated, open up the Construct 2 project file and make the necessary changes. After the changes have been made and the animation has been exported as an HTML5 project, the index.html file will need to be changed so that it looks exactly like this:

```
<!DOCTYPE html>
```

```
<html manifest="offline.appcache">
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<!-- This ensures the canvas works on IE9+. Don't remove it! -->
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
```

```
<title>New project</title>
```

```
<!-- Note: running this exported project from disk may not work exactly like preview, since  
browsers block some features on the file:// protocol. Once you've uploaded it to a server, it should work  
OK. -->
```

```
<!-- This outlines the canvas with a black border and makes the page background black. -->
```

```
<style type="text/css">
```

```
body
```

```
{
```

```
background: #000;
```

```
min-height: 100%;
```

```
height: 100%;
```

```
-ms-touch-action: none;
```

```
}
```

```
#frameT {
```

```
width: 100%;
```

```
padding: 0;
```

```
display: table;
```

```
height: 100%;
```

```
position: absolute;
```

```
top: 0;
```

```
    left: 0;
    margin: 0;
}
```

```
#frameTC {
    padding: 0;
    vertical-align: middle;
    display: table-cell;
    margin: 0;
}
```

```
#borderwrap {
    background: #616161;
border: 1px solid #8F8F8F;
padding: 15px;
border-radius: 10px;
    margin: 0 auto;
    width: 800px;
    height: 600px;
}
```

```
#c2canvasdiv
{
    width: 800px;
    height: 600px;
}
```



```

    canvas
    {
        -ms-touch-action: none;
    }
</style>

</head>

<body>
    <div id="fb-root"></div>

    <div id="frameT">
    <div id="frameTC">
    <div id="borderwrap">
        <!-- The canvas must be inside a div called c2canvasdiv -->
        <div id="c2canvasdiv" width="800"; height="600">

            <!-- The canvas the project will render to. If you change its ID, don't forget to
change the
            ID the runtime looks for in the JQuery ready event (above). -->
            <canvas id="c2canvas" width="800" height="600">

                <!-- This text is displayed if the visitor's browser does not support
HTML5.
                You can change it, but it is a good idea to link to a description of a
browser
                and provide some links to download some popular HTML5-compatible
browsers. -->

                Your browser does not appear to support HTML5. Try upgrading your
browser to the latest version. <a href="http://www.whatbrowser.org">What is a browser?</a>

```

```
<br/><br/><a
href="http://www.microsoft.com/windows/internet-explorer/default.aspx">Microsoft Internet
Explorer</a><br/>
<a href="http://www.mozilla.com/firefox/">Mozilla Firefox</a><br/>
<a href="http://www.google.com/chrome/">Google Chrome</a><br/>
<a href="http://www.apple.com/safari/download/">Apple
Safari</a><br/>
<a href="http://www.google.com/chrome/frame">Google Chrome Frame
for Internet Explorer</a><br/>
```

```
</canvas>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Pages load faster with scripts at the bottom -->
```

```
<!-- Construct 2 exported games require JQuery. To save bandwidth, by default
this is set to grab it off the Google content delivery network (CDN). Fall back to local if not
available. -->
```

```
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
```

```
<script>window.JQuery || document.write("<script
src='jquery-1.7.1.min.js'>\x3C/script>")</script>
```

```
<!-- The runtime script. You can rename it, but don't forget to rename the reference here as well.
This file will have been minified and obfuscated if you enabled "Minify script" during export. -->
```

```
<script src="c2runtime.js"></script>
```

```

<script>
    jQuery(window).resize(function() {
        If (window.c2resizestretchmode === 1)
        {
            window.c2resizestretchmode = 2;           // put back when
breaking back out of fullscreen
            var canvas = document.getElementById("c2canvas");
            window.c2oldcanvaswidth = canvas.width;
            window.c2oldcanvasheight = canvas.height;
            window.c2eventtime = Date.now();
            var w = jQuery(window).width();
            var h = jQuery(window).height();
            cr_sizeCanvas(w, h);
        }
        else if (window.c2resizestretchmode === 2)
        {
            // Size event fires twice on FF + Chrome, ignore second trigger
            if (Date.now() > window.c2eventtime + 50)
            {
                window.c2resizestretchmode = 0;
                cr_sizeCanvas(window.c2oldcanvaswidth,
window.c2oldcanvasheight);
            }
        }
    });

    // Start the Construct 2 project running on window load.
    jQuery(document).ready(function ()

```

```

    {
        // Create new runtime using the c2canvas
        cr_createRuntime("c2canvas");
    });

    // Pause and resume on page becoming visible/invisible
    function onVisibilityChanged() {
        if (document.hidden || document.mozHidden || document.webkitHidden ||
document.msHidden)
            cr_setSuspended(true);
        else
            cr_setSuspended(false);
    };

    document.addEventListener("visibilitychange", onVisibilityChanged, false);
    document.addEventListener("mozvisibilitychange", onVisibilityChanged, false);
    document.addEventListener("webkitvisibilitychange", onVisibilityChanged, false);
    document.addEventListener("msvisibilitychange", onVisibilityChanged, false);

</script>
</body>
</html>

```

Be sure to add MIME type overrides if support for legacy browsers is something that is required (see implementation / deployment sections of report for more info).

The HTML5 project directory can then be moved to its proper location on the server.