**Higher Diploma in Science in Web Technologies**

**Semester 2**

Rich Internet Applications

Cat Application

Keith Gregg

x12114090

Sandra McCormack

x12110639

David Winters

x12110396

**Introduction**

**Project Scope**

This project is an application all about cats and their unique personalities. Users can explore the site and learn all about the featured cats and test knowledge gained from the app via our quiz.

**Target Market**

The product is aimed at Cat lovers, as well as being and eduCATional tool it has an entertainment factor (Edutainment), so it can be aimed at young learners as well as those with a sense of humour.

**Benefits**

The application is a new way for users to interact with cats. It delivers information, in an interactive, visual and informative way; thus by researching the cats on the application, the user can test their knowledge by means of an interactive quiz that assesses recall and memory, by having questions with only one right answer.

**Objectives**

The main objectives of the application are:

* To test knowledge and recall based on trivia pages via the quiz
* To deliver information to users on the cats though visual interfaces and information sources
* To allow the user to have fun while learning something new.
* Guide the user through the application in a linear way.

**Goals**

The overall aim of the project is to teach users about our featured cats in a playful and enjoyable way, while assessing recall and knowledge through a memory based quiz.

**Product Features**

* Parallax scrolling homepage
  + Includes Fruit Machine Game
  + CSS3 and jQuery animated picture gallery
  + Customised alertify modal boxes
  + Scrolling Parallax interface which is visually appealing to a user
* Cat Info Page
  + Includes CSS3 tabbed box
  + HTML5 Audio Content
  + Transitions between tabs
* Cat Gallery (using CSS3 and Lightbox)
  + jQuery Lightbox Gallery
  + Information popup boxes with a modal feel using transitions
* Cat Quiz
  + Uses jQuery as well as JavaScript with JSON objects
  + Surprise when you win
* Cat Dress up game
  + Uses jQuery UI [.draggable();] and CSS3

**Technical approach:**

We are relying on a strong conceptual theme in this application; rich visuals and interactivity are essential for our target audience, cat lovers and learners.

We researched a variety of RIA/Multimedia approaches to determine what was necessary to incorporate into the application, such as Flash ActionScript, HTML5, CSS3 animation, jQuery, AJAX and JSON and vanilla JavaScript.

Using Rapid Application Development and testing, we quickly developed this prototype application.

By testing our JavaScript code, we know our system is always working, though if JavaScript is disabled, only static content will show, and it will lose its rich functionality. We have pop up alert notifications that inform the user about what page they are on using alertify.js library. We have used simple, comprehensible language, which the user can relate to, by doing so we believe that the user is always in control in the app. Each area of the application is clearly marked in the header which related to typical standards (header, navigation, footer). When an error (if one occurs) you know, such as a wrong answer in the quiz, or if you time out.

AJAX allows our site to be small and compact and more responsive, pulling in a simple and aesthetic design: we keep our information short and to the point.

**Dependencies**

This application aims to be cross-browser compatible (we have done this by including “normalize.css” - a CSS file which normalizes functions as part of our CSS bundle, as well as the modernizr.js HTML5 shim, which converts HTML5 code to more legacy friendly mark-up).

In order to have Ajax functionality, the application must be run on a server, so we decided to use a server side framework for the project. Initially we had used Ruby on Rails, but due to several constraints, we opted to move to ASP.NET.

**Languages and libraries**

* HTML5, CSS3 and JavaScript
* Ajax
* JSON
* jQuery
* jQuery UI
* A variety of open source JavaScript libraries which we have customised

The Application utilises some JavaScript libraries, including;

* Moderinzr
* jQuery
* jQuery UI
* Lettering.js
* Alertify.js

The advantages of using jQuery over vanilla JavaScript in our application is that coding time is faster though performance may suffer as vanilla JavaScript is always faster. However for this application, the constraints of vanilla JavaScript versus jQuery are negligible. jQuery also has many supporting libraries available, as it is one of the most popular Open Source JavaScript frameworks available.

**System Features**

**Parallax Scrolling**

Description

Parallax scrolling homepage/splash page gives a taster of the application. It incorporates a variety of JavaScript code as well as using some CSS3 transitions. The idea behind parallax is that is creates a visual stimulus to engage the user in a dynamic and engaging way. By adding to the basic parallax code, incorporating mini-games at certain points, as well as a mini gallery and hot points, we have attempted to grab the users’ attention from the get-go.

Stimulus/Response Sequences

* Parallax scrolling/clicking on the Doctor Octocat
  + Each main div slide transitions from section to section giving a dynamic feel, as the background moves at a different speed to the scroll speed.
* Motion Gif animation, that moves as you scroll, mixed with a moving parallax background
  + gives as sense of user control. each frame moves as the background moves.
* CSS3 and jQuery Rotating Gallery
  + Allows the user to rotate through the images of the featured cats which there is information on, the stars of the show
* Fruit Machine Game with Customised Modal Box pop up
  + A game of chance. Match three cats and you are lucky! The system will prompt you if you are lucky, or if you are not lucky this time. The code was based on a classic “one armed bandit” machine, and edited to our applications specifications.

**AJAX Requests**

Description

Creates a “one page application” feel by pulling in content dynamically through AJAX requests.

Stimulus/Response Sequences

Links load in the same page without needing to refresh. AJAX makes the viewing of information faster, acting as an intermediary between the front end and server side code. We also have an alert transition between pages when you switch between each part of the application.

Each page is cleared when you move from “page” to “page”.

**CSS3 Tabbed Information**

Description

Trivia page featuring information on each of the featured cats in the application, using css3 tabbed boxes, incorporating HTML5 audio.

Stimulus/Response Sequences

* By clicking the tabs, users can learn all about the different cats
  + Each tab has an image
  + Facts
  + Description
  + Audio sound

**Gallery using Lightbox**

Description

A CSS based lightbox gallery (as opposed to a more traditional jQuery/JavaScript gallery) which demonstrates images and information about each of the featured cats.

Stimulus/Response Sequences

* Modal Style pop up
* Information box
* Names featured on :hover

**Basic JavaScript Quiz**

Description

JavaScript based quiz which tests what the user has learned from viewing the application.

Stimulus/Response Sequences

* List of Questions generated from JSON objects
  + Clickable, and verifiable using jQuery validation
  + If wrong, goes red and reduces in size using jQuery UI and CSS
* Countdown Timer
  + Time reduces from 15-0
  + Quiz times out if the counter hits zero
* Score Counter
  + Score goes up with each correct answer
* Progress Bar
  + Moves up with each correct answer

**Dress up Game utilising jQuery UI**

Description

The dress up game was developed based on a simple site we saw in pure JavaScript, which we wanted to replicate using simple jQuery. Using jQuery UI’s .draggable(); we were able to target images on the page, and drag them over a cat image that would be “Dressed up” in a variety of outfits and garments and accessories. The methodology behind this was pure humour, as well as a way of exploring some of the features of jQuery UI.

Stimulus/Response Sequences

Users are stimulated by the variety of dress up outfits available to dress the character up in. Each item is draggable and you can dress up the character in a variety of outfits. Additional functionality may include saving an image of your dress up character, but we did not get that far in development.

**Constraints**

Deciding on a Server Side Framework

* Time spent Rails V PHP V ASP.Net V No Server Side Site

The main issue we initially had was trying to decide what structure our application would be built with: Pure front end HTML 7 JavaScript, Ruby on Rails, or ASP.NET. Initially we looked at using rails, however Sandra and David both felt far more comfortable with ASP.NET, whereas Keith was more comfortable with Front End Development as opposed to Server Side coding, so as a group we felt we would best work as a team using ASP.NET. We decided against using a pure front end site as the AJAX requests would not work without server interaction, and we wanted to explore AJAX as a group.

.NET Features

* Initially converting Static HTML5 to ASP.Net
* Deciding whether to attempt C# MVC or VB WebForms
* Running and coding asp.net on a mac

We had considered trying to use ASP’s MVC framework and using Visual C#, however we thought that this would be too much of a learning curve to develop, so we stuck to VB web forms. We initially developed out front end code in pure HTML, and merged it with the .aspx pages, however this threw up problems, so development moved to pure ASP. As Keith used a mac, there was a frustration about not being able to code and test on a mac without a windows server, however Dreamweaver allows ASP coding support, which is how we overcame that problem, also by pushing code to github, it became easier to share, even though we did not push and pull.

Advanced Quiz Features

* Trying to lower score for an incorrect answer
* Progress bar (jQuery UI)
* JavaScript corruption with AJAX

We had intended to use the jQuery UI to develop a progress bar, which showed you how much of the quiz you had answered. As we will explain later, we did not have sufficient time to develop this and we focused on generating other functionality.

Problems we had with AJAX by not setting global variables when we had set them locally, corrupted the code for the quiz, which was frustrating, but by debugging, research and support we fixed this by purging the AJAX code of errors.

Github & File Sharing

* Learning protocol
* Branching and forking
* Lack of Git being usable on college computers
* Email Fail
* Recreating the wheel
* Missing bits of code

Github was an issue. It does not work on college computers, in particular the user friendly GUI does not work at all. Sandra and David felt it was a huge learning curve given the limited amount of time we had after having decided what to develop for our project. Ideally we would have pushed and shared the files via github, however code got messed up through emailing different version of the application to one another. Another problem was that we kept re-writing the same lines of code, and each of us was developing the same areas differently. We combatted this by developing a better work structure and taking responsibility for different areas of the application.

Confidence with JavaScript and AJAX

* New knowledge
* Not as confident with JavaScript as with CSS and HTML
* Attempting to incorporate innovative functionality
* Learning AJAX
* Learning from basic mistakes

We all felt jQuery, but more so JavaScript was challenging and a bit overwhelming to begin with. We felt we had good competencies in HTML and CSS, however JavaScript was not something we were particularly familiar with. Our main problem is we started off too complex, and did not attempt simple functionality to begin with.

Through developing the application, and through trial and error, our confidence grew and we began to be able to debug and test our application and make better progress.

Time constraints

* SASS versus vanilla CSS
* jQuery UI Further Features
* Advanced typography with Lettering.js
* The (human) cat centipede game
* JavaScript quiz progress bar

We had looked at generating dynamic SASS which would compile to better formed CSS, which would have global variables for the app. In the end we decided as a group, despite some members having some knowledge of the technology, it was felt for a rapid development cycle that it would be better to use vanilla CSS for this development phase.

As a group, we discovered jQuery UI very late in the day. jQuery UI adds more functionality to jQuery, allowing such functionality as .draggable(); as seen in the “Dress Up Game”. Ideally would like to have better employed this library into our application, via developing further games.

The typography is something both Keith and Sandra have wanted to develop further, given our respective backgrounds in Graphic Design. While David used his sound engineering background to research and develop HTML5 Audio/AJAX sound functionality, we discovered quite late the lettering.js extension to jQuery. Ideally, given more time, we would employ some interesting and interactive Typographic functionality and behaviours, however, be best felt time would be spent developing the dress up game instead.

The Cat Centipede was an idea for a game we were going to employ, but in the end we opted for the “Dress up” Game. The Cat Centipede would have worked in a fashion similar to snake on old Nokia phones, where the “cat centipede” would grow each time it ate a ball of wool. The centipede cat would then need to navigate around the DOM avoiding edges. We felt this was a bit too advanced for our current level and knowledge of jQuery and JavaScript, but would like to attempt this in future developments.

Images Appendix











