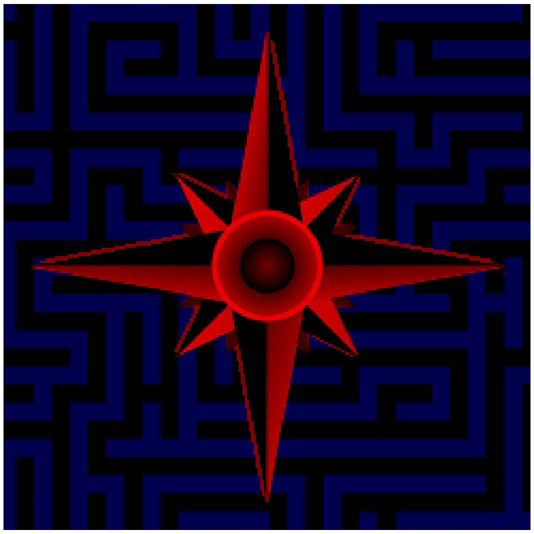
Carnation



Final Report

Index

1. *Introduction*
2. *Chapter 1: Project Summary*
3. *Chapter 2: Technologies and Libraries*
4. *Chapter 3: Lessons Learned*
5. *Chapter 4: Challenges*
6. *Chapter 5: Concepts Used*
7. *Chapter 6: Project Break-Down*

Introduction

This document contains the final report for this project. The purpose is to summarize my project over the course of the semester. In essence, it is a meta-report, similar to the smaller, weekly reports we submitted over the semester.

Chapter 1: Project Summary

Those who have visited the University of Massachusetts Lowell North campus may be impressed by the various styles and architectures which are encompassed by the many buildings, spanning from the 1930s to modern times. However, one might also rapidly become lost while trying to navigate the labyrinth of halls, classrooms, and tunnels.

It is this problem which I propose to solve. This project will consist of a web app (post likely a HTML web page) in which a person can enter their, relative location on campus, e.g. "Dandeneau Hall, room 409", and a desired destination in a similar fashion. The application will then use the universities floorplans to determine the most efficient way to traverse from the current location to to the goal location, and provide instructions.

Chapter 2: Technologies and Libraries

This project utilized a series of different technologies and libraries. In addition, I experimented with several which were eventually discarded in the current version.

Currently Used

- ***NodeJS***
  After I learned that an HTML/JavaScript web project cannot modify remote files, I realized that I needed a server to access my database (of user accounts, etc.). My server is written using NodeJS. I use XmoHttpRequests to request and process data from my “NoSQL” JSON-based database.

- ***Bootstrap***
  In my original implementation, I wrote a set of custom CSS files. However, we then had the class in which Dr. Haim discussed the importance of creating web pages which can be resized to fit all sorts of screen sizes. So I archived the my original CSS files, and re-styled my project using Bootstrap.

- ***THREE.js***
  After one of classes in which we discussed in graphics, I discovered the THREE.js library, an API for 3D WebGL applications. I was immediately interested, and added several 3D parts to my project as a result. One of these is a rotating cube with my project logo in the HTML header, and a 3D rotating scene in the “results” page.

- ***JQuery***
  I am using the JavaScript JQuery library to make DOM traversal more easy.

- ***Jquery-validation***
  This package facilitates dynamic user input validation. I am not using it for all the input fields, but have employed it in most of them.

- ***Pngjs***
  The pngjs library is a JavaScript API for image processing. I use this for floorplan manipulation.

- ***Apache CouchDB***
  Testing and analysis. At the moment, I am primarily using a simple NodeJS XmlHttpRequest server to handle my JSON database.

Chapter 3: Lessons Learned

Besides some rudimentary HTML and JavaScript programs, almost all of the technologies used in this project were for the first time.

Chapter 4: Challenges

Beyond the usual issues with using new tech, my greatest challenge was dealing with asynchronous functions. For example, when I started working on the image processing, I didn't realize that the read function was asyn. Thus, the next line in code (which starts trying to access the pixels) kept failing, because the image didn't yet exist.

I have managed to resolve most of my asynchronous problems, though my "results.html" page still has an issue where it will freeze until the map and path have been computed.

Chapter 5: Concepts Used

Chapter 6: Project Break-Down

This is **not** meant to be an exhaustive list/description of all the files in this project – merely a few words on the most important and/or interesting.

- ***Carnation***
  - ***server.js***

    This NodeJS server manages account information. For example, when a new account is created, the client form sends a XmlHttpRequest to this server, and the data is added to a "NoSQL" JSON database.
- ***Carnation/public***
  - ***index.html***

    This is the sign-in page. It requests the user's username/email and password, and then does a XmlHttpRequest to the database to see if that user's credentials exist. If they do, the user will be redirected to the "main" page (html/main.html).
- ***Carnation/public/html***
  - ***about.html***

    This page contains developer information about the project. Basically, I converted my Markdown README file to HTML.
  - ***gallery.html***

    An image gallery. A thumbnail of each image on North campus is displayed, along with some text. If one is clicked, it will open full-sized in a separate tab.
  - ***loading.html***

    This page displays while the program is looking for resources. It displays a rotating loading animation. Once the resource search is complete, the animation will disappear (via JavaScript), and a "display results" button will appear.
  - ***main.html***

    This is the main HTML page, containing the dashboard, and user selection information.
  - ***privacy.html***

    This page came about when I was learning how to use the "<model>" element to have a pop-up open over the page. Also, I meant it to be a humorous reference to the class in which we discussed the lack of privacy in the Internet.
  - ***result.html***

    This page displays the floorplan(s) by copying the image data onto a canvas, and then drawing a path on it. Also this page has a 3D rotating background I write in CoffeeScript using THREE.js.
  - ***sign_up.html***

    Allows the user to create a user account. This data is sent to the Node.JS server via an XmlHttpRequest, which saves it to the "NoSQL" JSON database.
- ***Carnation/public/src***
  - ***client.js***

    Contains client-side JavaScript for parsing user input.
  - ***cube_animation.js***

    This file contains the code for my spinning cube.
  - ***flip_image.js***

    This file contains code to dynamically swap out the images of the buildings, depending on which building has been selected in "main.html".
  - ***index_validation.js***

    This files uses the jquery-validation library to do some basic validation of user data.
  - ***loading.js***

    This displays a rotating loading animation. Once the resource search is complete, the animation will disappear (via JavaScript), and a "display results" button will appear.
  - ***sign_in.js***

    Checks to see if the user exists.
  - ***sign_up.js***

    Sends user data the Node.JS server via an XmlHttpRequest, which saves it to the "NoSQL" JSON database.
- ***Carnation/public/src/geometry_msgs***

  This directory contains ROS-like (http://www.ros.org/) data structures I developed in CoffeeScript for my path-planning code.
- ***Carnation/public/src/search***

  This directory contains my CoffeeScript implmentation of the A* search algorithm.
- ***Carnation/public/src/sensor_msgs***

  This directly contains some of my image processing code. Most notably, a wrapper for pngjs which allow the user to query pixel data similar to how one would in OpenCV.
- ***Carnation/public/test***

  This directory contains meriad tests, mainly for image processing, and the search.
- ***Carnation/public/css***

  This directory contains my CSS files.
- ***Carnation/public/css_legacy***

  This directory contains my old CSS files – the ones I created before switching to Bootstrap.
- ***Carnation/public/html_legacy***

  When we first started the project, I wrote a number of generic HTML pages, just to practice, and get familiar with the technology. That is what this directory contains.
- ***Carnation/public/res***

  This directory contains various resources, including images and floorplans.
- ***node_modules***

  This directory contains my JavaScript libraries – both the ones I am currently using, their dependencies, and a few which I was experimenting with, but ended up cutting out (some of these are referenced in my test scripts).