Steven Myers ([stelmyer@indiana.edu](mailto:stelmyer@indiana.edu))

A290 Final Project Phase 2 – Storyboard and Logical Design

I have decided to keep my original pitch for an idea from Phase 1. The description I gave for my idea in Phase 1 was the following:

**“Create an application that scrapes web pages for links and builds a visual graph of interconnected websites. Add functionality that would allow the user to specify recursion and visitation depth, so that a target website’s links get visited and added to the graph of connected websites. Perhaps add functionality for page rank displays and such as well.”**

As a recap, my audience for this application would be web designers and developers looking to improve the interconnectivity of their website, either with just their local pages or all links to any pages (not just the same website). It would provide a visual and statistical overview of how easy it is to navigate a given website. The application has the dual use of determining a website’s local connectivity and a website’s external connectivity.

I intend to do what I originally proposed, and attempt to add one more major functionality to the application. The other functionality I would like to include is the ability to open up one of the webpages we’ve queried in our graph in a C# web viewer, and highlight all of the links on that page so that the user can visibly see where the links (edges in the graph view) are located on the webpage. This extra functionality allows us to bridge the gap between user experience and user design by empowering the web designer with the ability to statistically see how many links to sites exist, and where they are visually located on the page.

I have included two pages of sketches of how I think this application should look visually. I think there will be a maximum of four different forms/windows that are going to be used. I will describe each briefly below:

* **Main Window:**

\*Liable to change – may become separate windows or different GUI objects.

* + This is the window opened up when the application first runs.
  + It has a menu bar with standard options and access to a few other windows via commands.
  + It has a text input box to enter in the URL of the target website you want to inspect and generate a graph for.
  + It has a dropdown combo box for the recursion depth/# of neighbors visited. If it is a value of zero, the search will only return all of the links (href attributes of anchor tags) in the html page chosen by the user in the target textbook. If it is a value of one, it will get all of the links of the given html target AND it will get all of the links of the target’s links. So, it will recur N number of times on each link, where N is the number in the combo box.
  + It has a search button to retrieve all of the links from the target html page, and the links from its children (and so on) depending on the chosen recursion depth.
  + \*It has a textbox to display information about the currently chosen node in the graph view.
  + \*It has a picture box to store the graph.
* **Settings Window:**
  + This is used to change some settings the application uses.
  + It has the ability to change the look of the Graph by changing the foreground/background, font size and type, and zoom. It can also further customize the color of the original target node in the graph.
  + Has some other checkbox settings. Right now, I am thinking of just two:
    - Show the text between anchors as edges in the graph
    - Show only local website pages (nothing external, e.g different dns/ip address).
  + Buttons cancel, save & exit, and apply.
* **Web Viewer Window:**
  + This window works a lot like a traditional web browser. It shows the URL and the web page rendered in the window.
  + It highlights very clearly all of the <a> tags in the document, so it is visually clear where all of the links are in the HTML page.
* **Reports/Statistics Window:**
  + This window shows some numerial data about the targeted HTML page and its links. I have not drafted this window in my sketches because I may just make it part of the main windows “logs” text input box.

Some of these windows may be merged in the main window and you can switch between them by clicking on “tabs” in the main window. I think this might be more intuitive than having a bunch of different child windows open.

As for the actual logic and programming behind the application, I will be using at least two different C# libraries. I am currently using HTMLAgilityPack to parse the HTML for its anchor tags and href attributes. It is located here:(<https://htmlagilitypack.codeplex.com/)>. I am also looking at using this C# library to draw the graph (<https://github.com/Microsoft/automatic-graph-layout)>. It is called Microsoft Automatic Graph Layout (MSAGL) and comes with a nice viewer form that allows a user to zoom in/out and explore a directed graph.

As a proof of concept, I am able to already grab all of links on a webpage. Here are some example outputs