Dhruv Patel

Francisco González

Nabeelah Khan

**FINAL REPORT**

**Design:** The program was divided into three different sections:

1. The histogram.c and histogram.h files which took the analyzed data and generated an animated pseudo .gif file. In addition to the functions that related to the histogram display itself, the histogram.c file included functions which calculated certain statistical information which was put onto the .gif display.(Nabeelah Khan)
2. The list.c and list.h files which dynamically allocated memory to store the data and generate counts. (Dhruv Patel)
3. The hw4.c file which contained the code to prompt the user for the data repository and presentation choice. This program contained functions to open the data repository and read it, then called functions in list.c to store and analyze the data. Once the data was analyzed, histogram.c functions were called to generate the animated pseudo .gif file. (Francisco González)

**Detailed Description**:

The project explores the use of the libplot library to create a histogram to represent the number of “DIVVY BIKE” rides. The use of the libplot library to created animated pseudo gif files interested us and we decided on a “data visualization” project. “DIVVY BIKE” is Chicago’s bike share program that we thought would be of interest to locals. The data was obtained from the company’s website, and we created a program which allows the user to analyze the amount of rides. Because it is more difficult to analyze large amounts of data numerically, visual representations such as histograms and pie-charts are preferred for data analysis. The graphical data representation that the program will output depends upon the category by which the user chooses to analyze the information.

The three categories are:

1. Rides per hour in a given day
2. Rides per day in a given month
3. Rides per month in a given year

The modular decomposition of this project idea was as follows:

1. The User Program to prompt the user for information
2. Abstract Data Type (ADT) to store data
3. The Display Program to generate the data visualization

After prompting the user for the data repository and choices, the program parses through the data file and stores each entry in an ADT made to record Month, Day, Year, and Time of each ride.

The underlying data implementation for this ADT was a linked list.

For option 1, the user program will create a list with all entries matching the day given by the user.

For option 2, the user program will create a list with all entries matching the month given by the user.

For option 3, the user program will create a list with all entries matching the year given by the user.

**Files Used**: list.h, list.c, histogram.h, histogram.c, hw4.c, Makefile

After our initial proposal, we all met to work on a simple scenario which related to our project idea. We were able to generate a simple pseudo .gif file, which depicted a histogram of data entered through an array, from our self-contained test program. From this point, we began to work on our individual responsibilities. We first came together to ensure that the user-driven program and the list program could communicate to read and store the data. Upon accomplishing our individual pieces of work, we reconvened to incorporate the histogram files with the list and hw4 files.

**Changes since Proposal:**

Since the submission of our proposal, no major changes were made to our plan other than deciding whether to have different files for the ADT.

**Testing:**

We started using a small set of data with which to test our program. We then expanded to files that contained data for the entire 2014 year and for the part of 2013 that DIVVY was in use. The files were stripped down to only the data that we would analyze. The files downloaded from DIVVY are organized with following attributes: “month/day/year hours:minutes.”

**What We Would Change:**

* A well thought out list of test cases with which to test our code would have made the process more efficient
* Define a means of communication amongst team members
* Better use of Git as version control to help manage changes
* Using reply-all so the entire group is on the same page

**USER GUIDE**:

**Description**: This program takes in a data file with data regarding rides of DIVVY bikes, Chicago’s bike share program.

First: The program begins by prompting the user for the data file to be analyzed.

Second: The program prompts the user for how they want the data analyzed and gives the user three choices.

1. day: which further prompts the user for the month, day, and year they want analyzed
2. month: which further prompts the user for the month and year they want analyzed
3. year: which further prompts the user for the year they want analyzed

Third: The program generates an animated pseudo-gif file representing the data based on the category of analysis the user chose for the histogram to display.

1. For day: The histogram represents the rides per hour in a given day.
2. For month: The histogram represents the rides per day in a given month.
3. For year: The histogram represents the rides per month in a given year.