Table of Contents

Acknowledgement

We would like to express our special thanks and gratitude to our teachers Mr. Daya Sagar Baral and Ms. Shanti Tamang for providing us the chance of a lifetime opportunity to collaborate with our fellow classmates and work on the project. The project would only be a daydream if it weren’t for their valuable guidance and support.

Secondly, we would also like to express our special appreciation to our respected seniors who have always been tremendously supportive and helpful to us by providing necessary resources, assistance and sharing knowledge whenever required.

Introduction

"HTMLER" is a group project proposed by the undergraduate students of Pulchowk Campus. The project promises to create a basic prototype of HyperText Markup Language (HTML) parser, basically used to render HTML file with Cascading Style Sheets (CSS) in the web browser. HTML is simply a standard markup language for displaying our desired contents in an organized form in a web browser and parsing, in general, means breaking down something into its constituent parts and components particularly for the study and analysis of those individual parts. An example of parse is to break down a sentence to explain each element to someone. So, HTML parser extracts each and every element/tag of the entire HTML document and renders and displays it exactly as the webpage, as the browser does.

The main objective of our project is not to optimize the processing of our HTML file since all the browsers do it perfectly. Rather, it is just our curiosity to learn and implement how browsers do all this functionality to process the HTML tags and attributes to provide the user data and information flawlessly. And finally, we aspire to implement our proposition into action using object oriented programming to make the best use of inheritance, encapsulation, polymorphism concepts.

Objectives

General Objectives

- To get acquainted with the fundamentals of object oriented programming paradigm.
- To understand and implement object oriented concepts like polymorphism, inheritance, encapsulation and abstraction at its core.
- To gain experience with coding in C++ to build applications as per necessity.
- To develop problem solving skills for real world problems with the help of code.

Specific Objectives

- To get acquainted with the concept of parsing, tokenization and syntax tree construction.
- To create the Document Object Model (DOM) and render the layout accordingly.
- To implement basic styling features of CSS using CSS Object Model (CSSOM).
- To get to know the basics of a GUI application like painting, rendering and updating the window.

Proposed System

Description

The proposed application promises to parse the HTML file to create a Document Object Model and paint the window accordingly implementing the basic styling properties with the help of external libraries. The overall system can be broken down into following categories:

1. Construction of Document Object Model

The construction of DOM is not as straightforward as it sounds. It includes multiple steps, these steps being converting the HTML bytes into tokens, creating nodes with tokens and associating them finally to create DOM tree. The proposed system supports all but basic html tags like html, body, title, ul, ol, li, div, h1-h6, p etc.

2. Creation of CSS Object Model

The CSSOM formation process is similar to the formation of DOM. Color, font size, background color, border, margin, padding, width, height are the basic CSS properties supported by the proposed application.

3. Formation of Render Tree

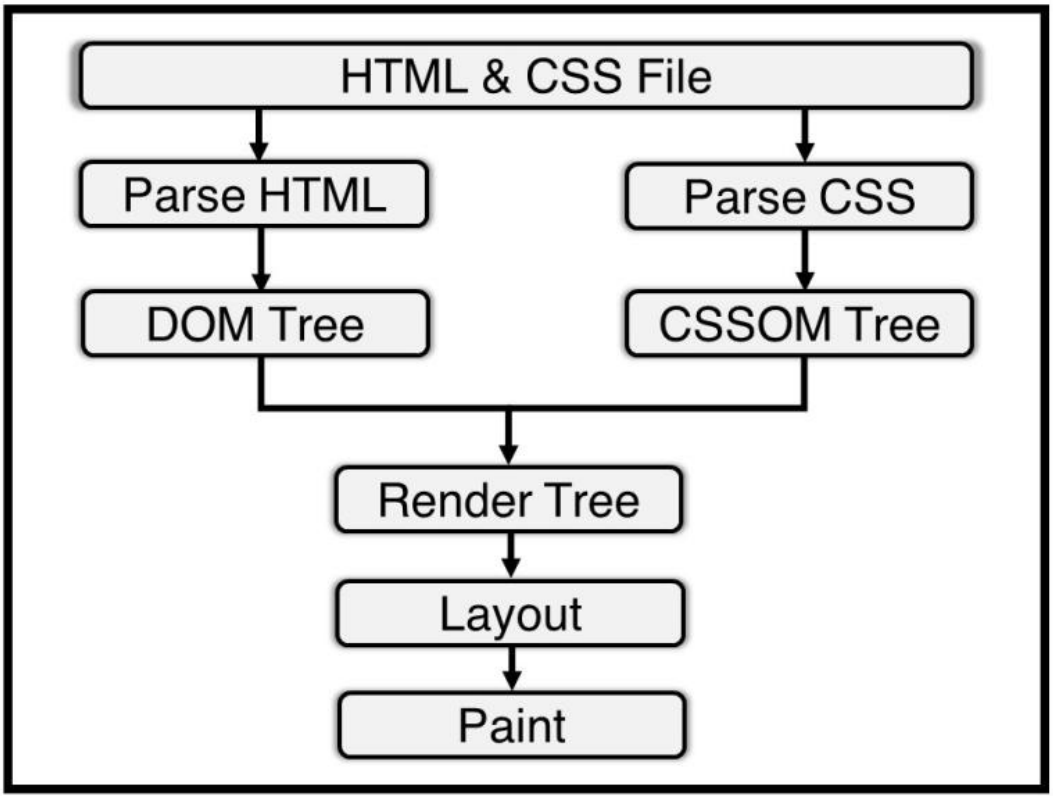
In this process, the DOM and the CSSOM are linked together to create the render tree which is the foundation for the rendering and painting of HTML element. In render tree, only the elements which are to be displayed are included.

4. Layout and Painting

The exact position and dimension of all the elements are calculated under layout. This step is carried out with the help of render tree.

Based on the layout, we finally paint the screen and the html file is ready to be displayed.

System Block Diagram



Tools

We have decided to use the bare minimum tools required for this project which are as follows:

Tools
IDE: Microsoft Visual Studio Compiler: Microsoft Visual Studio C++ Compiler Language: C++ Standard Version Control System: Git and GitHub Collaboration Utility: Microsoft Visual Studio Live Share Utility Libraries: SDL

1. Integrated Development Environment

The IDE that we have chosen for this project is Microsoft Visual Studio. It has great support for debugging C++ code that other IDEs simply fail to come even close to. The IntelliSense and code completion features of Microsoft Visual Studio are beyond the need of any developer.

2. Simple DirectMedia Layer

Simple DirectMedia Layer is a cross-platform development library designed to provide low level access to graphics and hardware via OpenGL and Direct3D. SDL officially supports Windows, Mac OS X, Linux. So, the programs run smoothly on the majority of the devices. SDL is written in C, works natively with C++.

3. Version Control System and Live Collaboration:

Version control is a software that records the changes made to a specified files and documents. We can also collaborate with a large number of people to simultaneously work on a single codebase. We can run, test and develop certain files without affecting the main or user end of the software.

We have decided to use git among the various version control systems available in the market. Git is open source software distributed under the terms of the GNU (General Public License) that has an emphasis on speed and performance.

Methodology

Among various ways of solving the problem, we choose to go along with following steps to obtain the desired result:

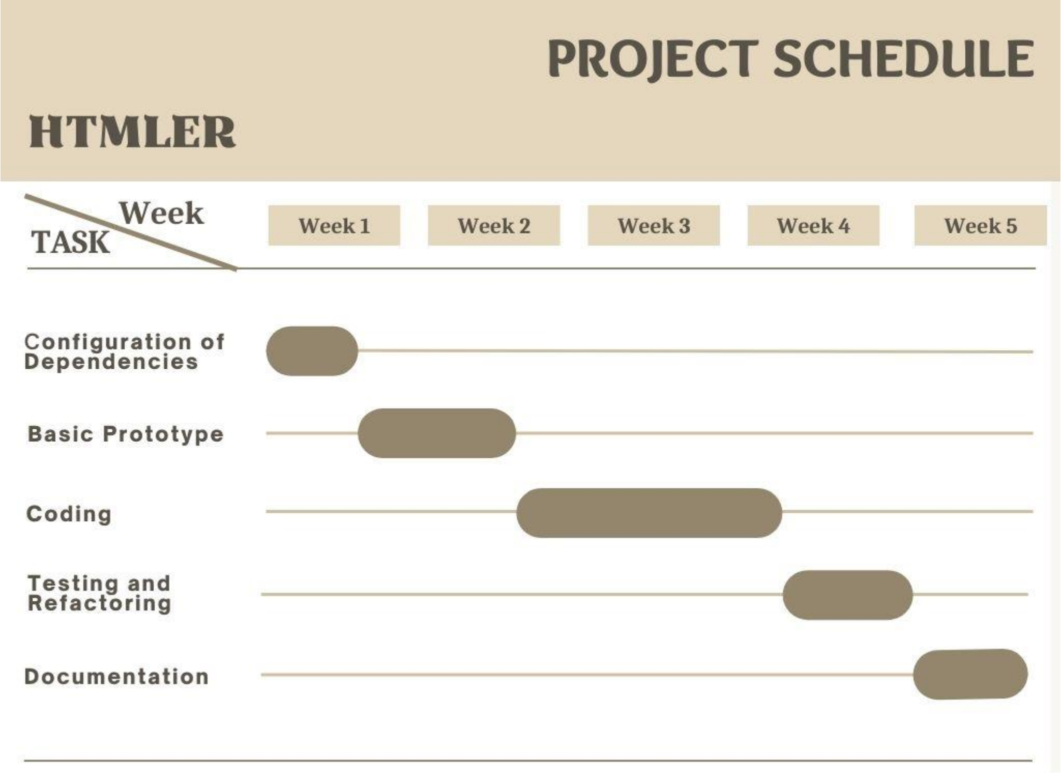
- Analyzing the concept of parsing.
- Creating an algorithm for HTML parser.
- Refining and optimizing the algorithm.
- Rendering the parsed tree into relevant form.
- Finally, adding more detail or features if necessary and starting to code.

Project Scope

The “HTMLER” project primarily focuses on parsing the HTML file contents and rendering it with the default and other additional CSS properties. The scope of this project is to create a webpage based on the HTML and CSS code similar to the browser mechanism. So, our project operationalizes some of HTML and CSS codes, what the browser performs in the backend to process, organize and deliver the web contents with required stylings.

Project schedule

From the five weeks of time we have been given to complete the project, we have decided to divide the time for various works as follows:


PROJECT SCHEDULE
HTMLER
Week
TASK
Week 1
Week 2
Week 3
Week 4
Week 5
Configuration of Dependencies
Basic Prototype
Coding
Testing and Refactoring
Documentation