Report for FileDiff Project

S E 416

Nicole Bruck

Cameron Legleiter

https://github.com/camlegleiter/filediff

Table of Contents

1 Introduction

1.1 Purpose

With many tools on the market to detect plagarism in code or changes in separate versions, there are very few that provide a very simple visual representation of the changes. Our web application shows the code side-by-side and colors the changes for the User to easily see the differences and similarities.

It does so by comparing each line of the code, ignoring whitespace and “typing” them. This allows us to group the changes into categories : Moved, Identical, Modified, Inserted and Deleted. This provides a very simple grouping for the User to quickly and easily analyze the code and make their assumptions.

Our product also provides “Statistics” of the overall File in comparison to the other. This gives a quantitative way of looking at the files and gives a further overall look without the user needing to scroll through to count the number of lines of one category in relation to the file.

1.2 Design Goals

- Reliability:
    - The product will work even when at least 1 user uses the website.
- Maintainability:
    - Code will be commented and clearly written.
- Modules will be appropriately sized and easy to understand purpose.
- Extensibility:
    - The modularization will allow for simply plug-n-play.
- Response Time:
    - The response time for user-interface will be less than 3 seconds so the User can quickly and easily create a layout with necessary furniture.

2 User Interface Description

2.1 Use Cases

- A Product Manager wants to see the changes that a Developer made to the code and doesn't have a very good understanding of the language
    - Allows the User to glimpse similarities without looking at syntax/style
- A Teaching Assistant wants to compare two student's code to see if they copied/shared code
    - The User can easily see the groups of copied code, even if variable names or formatting/order has changed

2.2 Design Issues

- String comparison is limited and doesn't provide a very good analysis for very complex code clones
    - Researched using a compiled code comparison program to enhance our accuracy
        - Required a backend server which would have complicated and expanded our program from the lightweight structure we used

3 Language/Framework/Tools Used

3.1 jQuery / jQuery-FileUploader

- A JavaScript library that provides easier access to DOM elements in HTML. Also used by Backbone and Marionette for finding individual UI components within a template.
- The jQuery-FileUploader plugin was written as a way clean up the "file" input type in HTML and provide a much cleaner look than the web browser's default implementation.

3.2 Underscore

- A utility library in JavaScript. Used as the basic framework for Backbone, primarily the template functionality.

3.3 Bootstrap Web framework

- A CSS framework for providing consistent layout and formatting between browsers.

3.4 HTML5 / CSS3 / JS

- The latest supported web technologies for browsers.

3.5 Backbone.js / Marionette.js

- An MVC JavaScript framework for modularizing different parts of the work. Each displayed file is reserved to an HTML-templated view (to reduce the amount of HTML written) and is controlled using Marionette.

4 Individual Responsibilities

4.1 Nicole Bruck

- Logic for Line Detection Types (Same, Modified, Inserted, Deleted, Moved)
    - Used JavaScript to compare line-by-line the files
    - Compared File1 -> File2 and then inverse (File2->1)
        - This allowed to view the first file as the “Original” and then consider the second as the “Original”, which gave a more accurate comparison
- Statistics of File Comparison (Similar, Deleted/Inserted, Modified)
    - Sat beside the line detection typing
    - Allowed for a more overall glimpse of the code, rather than manually scrolling through the files
        - Also made it easier to see which file was the original by analyzing

4.2 Cameron Legleiter

- Responsible for creating the user interface using the above described libraries.
    - Main layout was written using HTML and CSS
    - The actual work of storing the file and rendering the file contents was done by JavaScript, utilizing Backbone, Marionette, and the jQuery-FileUploader plugin.
- Wrote basic utilities for color coding certain characters/lines.