Verisium Drone

PinDown ML-Debug Visualisation Tool

Introduction

Verisium Drone is a visualisation tool for large amounts of data, mainly that of code diagnostics from large code repositories. Verisium Drone has been made in collaboration with Cadence and they have developed a tool called PinDown. PinDown can predict with machine learning where it is most likely bugs will appear in large repositories of code. Verisium Drone functions by reading large sets of data from PinDown in the form of a text file and then transposing that information into a treemap model.

Problems & Solutions

1. What type of data structure will be used for visualisation?
By creating multiple lo-fi prototypes of different data models it was found that a treemap model suited this interface best.

2. How do we display the debug information and/or the diagnostics?
The user will be able to choose what category of data they want to display in the interface. The data values will be represented as either color or size differences in the treemap elements.

3. What type of programming language and framework will suit us best?
Because the interface needs to be implemented as a static webapplication it was decided that Javascript and HTML worked best for the interface.

4. How do we read data from files and use it in the interface?
A Python script was written that reads the data from the input and copies that into the main Javascript file where the interface code is written. At the two first lines of the file it will create two new Javascript variables containing the data as a string.

Method

The project started by doing research on different data models and finding out what model suited this project best. Prototypes were made in an itarative and agile method and the user testing was done continuously every iteration. Every iteration a backlog would be written and updated to keep track of development.

Result

The result of this thesis is an interface for PinDown created as a static website which allows it to be easily implemented into other systems. The interface allows sorting and filtering of different categories of data and is navigated by clicking on the boxes of the treemap model. Each box is either a file or a folder and can be determined by whether or not the box name has a file extension or not. Information about the file or folder can be seen by hovering the mouse cursor over that box and more information like commit messages can be found by right clicking on the box.

Discussion

From the result that has been achieved by this thesis conclusions could be made of which data model to be used and how to visualise values as colors and sizes. But one problem remains regarding color, a lot of people today have some form of colorblindness and the current interface uses a red and green color scale to show value proportions. This can be difficult for users with colorblindness to see, therefore a suggestion for future development would be to implement a colorblindness option so even users with colorblindness can use the interface.