Letter of transmittal goes HEER

This is a title page
Included is an abstract

Contents

1 Introduction

I'd like to introduce some things.

2 Literature Review

Here we will discuss projects similar to ours, as well as technology we plan to use for our project.

- Building an In-Browser JavaScript VM and Debugger Using Generators
  http://amasad.me/2014/01/06/building-an-in-browser-javascript-vm-and-debugger-using-generators/

  In this blog post, Amjad Masad describes how he implemented debug.js, a JavaScript debugger running inside the web browser. Since we wish to implement a c0 debugger running inside the web browser, Masad's notes seem to be relevant. Specifically, this post discusses the architecture of debug.js, as well as various challenges Masad faced in developing it. Debug.js was designed in two separate parts: a virtual machine and a debugger. The virtual machine handled the task of evaluating the JavaScript program being debugged, adding support for stopping, starting, and analyzing the program. The debugger was the visual interface to the virtual machine, allowing users to control the virtual machine and see its output.

  Masad also discusses challenges he overcame while writing debug.js. These included being able to step line-by-line through a program, keeping track of a call stack, handling errors and exceptions, implementing native APIs, and dealing with events. While many of the details will be different when working with c0, we must still consider all of these challenges in developing our project.

- The Architecture of Open Source Applications (Volume 2): Processing.js
  http://www.aosabook.org/en/pjs.html

  In Chapter 17 of Mike Kamermans' book The Architecture of Open Source Applications, he discusses the design of Processing.js. Processing is a Java-based programming language designed to help teach computer programming in a visual context. Processing.js is a project designed to run Processing programs in the web browser using only JavaScript. This was done by writing a own Java-to-JavaScript compiler, and running the resulting code attached to a HTML canvas. Along the way, the developers ran into several different challenges, mostly due to differences between the Java and JavaScript languages. The largest difference between the languages was that JavaScript programs do not get their own thread; the browser freezes if a JavaScript program tries to run for too long. We must consider this issue among others for our project.

- Node.js Documentation
  http://nodejs.org/documentation/

  This is the documentation for the node.js platform. We plan to use node.js to write the server-side code for our project. We believe that node is a good fit for our project since we are writing JavaScript for the client side of our code, so this will let us work in the same language on the server and client side. Also, we can make use of the existing cc0 compiler to translate c0 source code to the bytecode our virtual machine will run. This is the same compiler used in 15-122, and integrating it with our server will make it feasible to run actual c0 source code.

3 Plan

Our goal is to build a web application that can debug c0 code. The user will type in or upload c0 source files. Once this is done, these files will be transferred to our server, where the cc0 compiler will be used

to generate bytecode corresponding to the user's source code. This bytecode will be sent back to the user's web browser, where we will be running a c0 virtual machine. The user will be able to control this virtual machine as it executes their code. This will give the user the ability to run their code line-by-line, to set breakpoints, view stack traces, and see the values of variables. By providing access to all this information, we hope to make it easier for users to write and debug c0 programs.

For version control, we will use a git repository hosted on GitHub. We will use a Gantt chart, shown later in this proposal, to stay on schedule.

4 Benefits

1. Write c0 on cement
2. Write c0.js
3. Write cQuery
4. ???????
5. Profit!

5 Approach

Use JavaScript and c0.

6 Evaluation Criteria

Can you c it working, or not?

7 Qualifications of Team Members

We are all c0 experts.

8 Sources Cited

TBD