

Code Workstation

Team Profile

- Lukas Rickard
 - Interface design
 - Command line experience
 - Python & C++
 - Management & Leadership
- Jake MacMillan
 - Multi-language Integrations
 - PHP
 - HTML + Javascript
 - Server & Async API design
- Team leader: Lukas Rickard
- This team is well suited for this project because:
 - Both members have experience with tool creation and usability
 - Both members are interested in having a more customizable scripting environment
 - After great thought we decide that this was the right project to work on. We decided it was going to be easier to divide work and plan before doing extensive additional research. The domain is large, but can be easily constrained.

Project Description

Purpose:

The purpose of Code Workstation is to provide programmers with an intuitive and modular code writing environment which aids in debugging and quick prototyping.

Functionalities:

- Embedded Text editor
- Customizable buttons which can:
 - Run console commands
 - Save current text
 - Output to a file
- Text view boxes which can:
 - Display console output from button commands
 - Display output from an automatic script
 - Display output from a set of console commands
 - Regex the output of a command and display it
 - Constantly run a command and update
- Resizable buttons, boxes and windows which should
 - Dock in expected places
 - Link to each other
 - Arrange inner elements into a grid
- Saveable workstation configuration

Our Customer

- Programmers and scripters will likely enjoy the features this text editor workstation provides because it will allow them to quickly set up a debugging environment using scripts and command line features they already use
- Super Geeky hobbyists will enjoy our software because it will allow them to have a modular coding environment
- Testers and QA experts will likely want to use our software because of the powerful yet simple script automation that it will facilitate for testing

Environment

We plan to develop for Linux personal computers first but use frameworks that will easily allow up to move to Windows or MacOS X if we have the time

Development Concerns

Necessary tools

- We will be using an open source text editing API called Scintilla as the basis for our text editor
- Our first choice for windowing and user interface is Qt since it runs well on most operating systems and we have only read good things about it.
- We'll likely use Vim, G++, and Python to create our tool. Eventually we hope to use our software to help in the development of Code Workstation.

Challenges

- One challenge in particular that our project is going to face is its scope. We want to add a lot of features to our text editor, and lots of features take lots of time. We have already narrowed our scope somewhat but we will likely have to decide at some point which features will make it in the final cut.
- Because of the interconnected nature of this project, it might be difficult to create working prototypes before the minimum features are created.
- If we ever do decide to move to our program to windows it maybe difficult to debug the windows command line quirks.
- There will probably be numerous challenges which we did not list here, but we are going to have to discover these in the planning process.