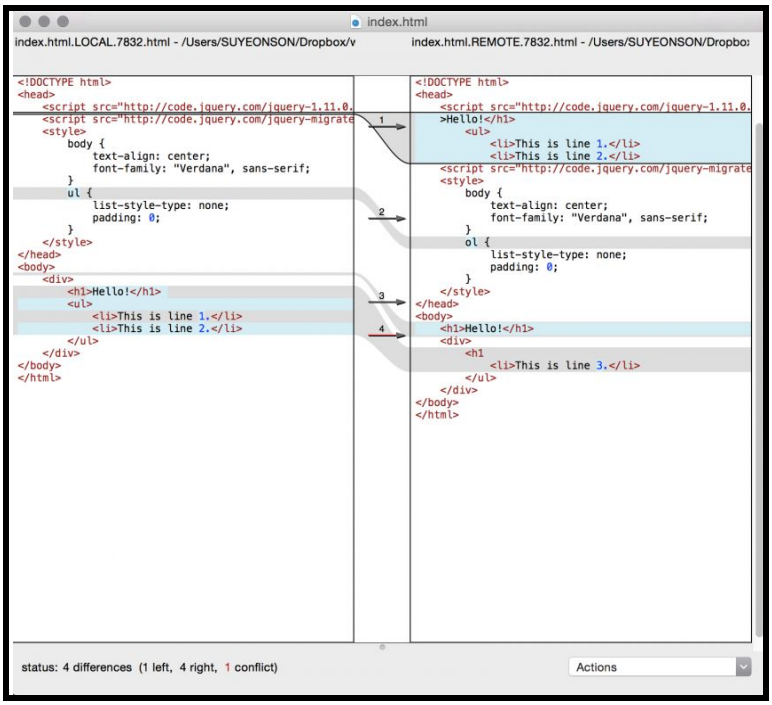
Haley Ruth - haleymr
Arjun Lalwani - arjun11

Merge Conflict Tool Pitch : Magic Merge

Motivation

One of the biggest problem that software developers face with version control is merge conflicts. The current way of handling merge conflicts by version control systems is line based. This means that when two developers change the same line of code in the program, the version control system gives an error and reports a merge conflict to the developer who attempts to pull in the new changes from the remote branch. The developer then has to solve this conflict by accepting the new changes, removing the new changes, or editing them in a different way. Although, the current version control system reports the error as a merge conflict, there are times when it's not really a merge conflict and the merge can be resolved. For instance, if one developer added a line break and the second developer changed the name of the variable on the same line, then the version control system would detect this as a merge conflict. However, this conflict is not a real one as different parts of the code on the same line were altered. Another instance is when one developer surrounds a code by a block of an if statement, then the entire block of code gets intended more, and this change conflicts with any change within the block of code. The current way to handle this problem is by using a git mergetool that launches an interactive GUI for developers to quickly accept or reject incoming changes. However, this does not mean it actually solves the problem of unnecessary merge conflicts. Our proposed tool is to create a tool that will automatically resolve such merge conflicts by analysing the incoming changes and make appropriate changes if possible to do so. The following figure details an example of a merge conflict resolution that our tool would perform.



Example of a Merge Conflict

Haley Ruth - haleymr
Arjun Lalwani - arjun11

Approach

Our high level approach to solve this problem is to find a way to parse the two programs and determine where the conflict occurs. Based on this information, we can then decide which changes should be accepted. One implementation to parse the two programs could be by using a AST (Abstract Syntax Tree) and noting which edits are in conflict. We can apply the changes to the AST and the AST then can be converted back to code. A potential limitation or drawback of our approach could be inferring context of the program before resolving any merge conflicts. For instance, accepting a change in the variable name and not treating that as a merge conflict could potentially make the problem crash as there might be other places in the program where that particular variable could be used. Catching these edge cases might be a harder technical challenge than expected.

Challenges & Risks

One of the challenges with our project proposal is the ability to finish the entirety of the project on time. Because our tool can possible have a wide variability in what kinds of merge conflicts in can handle, it will be a challenge deciding how many of these types of merge conflicts that our tool should be able to handle given how long we have to complete our project. For example, we could use our tool to only resolve merge conflicts where the portions of changed code do not overlap when compared to the user's last pulled file. We could further this tool then to also handle cases where two "innocent" changes were made on the same line like two different changes in whitespace before a command. We feel that we should determine the extent to which our merge tool should be able to resolve merges before we begin our process, but we are worried that we may give ourselves too much or too little work because we are inexperienced in building this kind of a tool and therefore would not be able to accurately estimate the amount of time it would take to complete each extension of the tool. Our overall goal is to be completely finished at the end of the project with a tool that we are proud of, so we believe to best combat this problem that we should spend more time in our planning stage. This will give us more time to research our project so that we can come up with a specification that we know we can complete in the time given.