Freeseer Project Proposal for Fall, 2014

Authors: Benjamin Buckley

GitHub: benbuckley

Overview

I would like to make improvements in the user interface and in the documentation for the project, in order to make it easier for new users to use the software, and easier to find help in the documentation if they need it.

Problem Statement

To quote the User Guide: “Our goal is for Freeseer to become so straightforward to use that this user guide should be seldom referenced, but we’re not there yet.” This is an admirable goal. At this point, there is plenty of room for improvement in fixing bugs in the user interface, and filling gaps in the documentation. For example, at the moment, if a user is having trouble figuring out how to use the Configuration window, there is very little help for them: http://freeseer.readthedocs.org/en/latest/userguide/config/index.html. Filling in these gaps would make the software easier to use, as well as making it look more professional. In addition, fixing some of the small bugs in the user interface would make a big difference in how smooth and effortless it is for the user to perform simple tasks.

Objectives

Since “improving the user interface” and “filling holes in the documentation” are huge, vague projects, I would like to start by focussing on the following open issues. If they turn out to be quick fixes, I will move on to other open issues, and perhaps open some issues of my own:

- Issue #532: Talks should not be unselected after being modified through the table view. Fixing this would improve the user experience, and give me some hands-on experience with the code.
- Issue #442: Reorganize Getting Started Guide. Having made some mistakes in setting up Freeseer, I know the importance of having a good starting guide.
- Issue #541: Config Window Resizing. Like #532, this is a small but frustrating error, and fixing it would give the program a better sense of consistency.

Added on September 20th, 2014:
- Issue #501: Warn user about losing new talk info when an existing talk is selected.

Technical Details

I expect issue #532 to be a frustrating one. I believe that solving it will require me to spend a lot of time learning about Qt, and a lot of time sifting through the code for the Talk window to figure out what is causing this issue. At this point, I am not familiar enough with the code to know exactly where to look for the problem, so I suspect that I will have to ask questions to the mentors and possibly to the other students in the project about the code.

Issue #442 is more in my comfort zone, but I still expect it to be difficult in its own way. Dennis Ideler's post on issue #442 has a checklist of things to do, and if I get sidetracked, and in the interest of cooperation, I will focus on that checklist. However, I would also like to spend some time thinking about how else the Getting Started guide could be improved. This could mean looking at the program from the point of view of a new user and considering what questions a new user might have. It might also mean researching the existing guides on the internet to see what problems people have had when installing Freeseer, and how they were eventually resolved.

I would expect issue #541 to be easier after I have wrestled with #532 a little bit. Once I have some familiarity with the code of the project and with Qt, figuring out what's causing this issue should be a slightly less excruciating process.

Expected Problems

I have never used Git before, but I have already learned a bit about it and made my first commits and pull request (the pull request was rejected, but the point is, I learned something). I have been reading the guide to Git here: http://git-scm.com/book . I have also never used Qt or GStreamer before, and I expect that to take a lot of my time. I am currently in the process of finding useful PyQt tutorials -- this list seems promising: https://wiki.python.org/moin/PyQt/Tutorials . I will do my best to look for general principles and patterns that I can apply when using any graphical user interface library.

(EDIT, November 3rd 2014: I should probably mention that, ever since September, rather than using the above PyQt guide, I have been relying on the Qt documentation here: http://qt-project.org/doc/. It is usually fairly easy to "translate" into Python; the important thing is the concepts.)

I have never used reStructuredText before, and most of the documentation is in the form of .rst files. However, I don't anticipate this being an extremely difficult markup language to learn, and I expect to pick up most of what I need from reading the documentation files and from Googling whatever questions I have. In addition, this tutorial looks promising: http://docutils.sourceforge.net/rst.html

I have a slight predisposition toward depression and anxiety, which might lead me to procrastinate. I have found that, when I tell other people about it, it becomes less of a burden for me and I am able to function better. I have been keeping track of the hours I have spent learning and working on the project each week, aiming for 10 hours of work each week. Provided that I sit down and work on the project for 10 hours each week, even in my least productive week, I am bound to make *some* progress.

One more thing: my laptop has been acting weird recently. It has a broken hinge, and lately it's been taking a long time to start up. I think I'll be fine as long as it survives the Code Sprint.

Schedule

Here is what I have done so far:

August 31st to September 6th: Researched Freeseer's documentation and history. Set up Freeseer for Ubuntu. Skimmed the code repository and made note of things I should learn about, including Qt, GStreamer, reStructuredText, Python logging, Bash scripts, makefiles, and open source projects in general.

September 7th to September 13th: Made first pull request. Became more familiar with Git, learning the importance of branching and the difference between Rebase and Merge. Successfully recorded video and audio with Freeseer. Brainstormed possible projects to work on, finally writing my first blog post and filling out Project Proposal document.

Here is what I intend to do in the coming weeks. I have set up the tasks each week with the assumption that I will fail almost everything I try, but that I will learn something in the process. I intend to put 10 hours each week into this course, and not more than 12 (with the exception of the week of the Codesprint). This will include the time I spend communicating, blogging and learning new things. I recognize that it is important to put time into this course, but I also don't want it to consume my life to the point where I lose sleep and my other commitments suffer. I also want to stress that I consider reading, learning and communicating to be legitimate work, as I cannot be expected to contribute to the project if I don't have the necessary technical knowledge, and communicating with other people is an important part of working on a software project. With this in mind, I think 10 to 12 hours per week is fair. If any mentors take issue with this, I encourage them to take this up with me privately:

September 14th to September 20th: Read more about Git, and read up on Qt (I expect that I will have a lot of reading to do in the ensuing weeks as well, but I don't want to get too repetitive). Attend Code Sprint in Toronto, focussing on issue #532. I don't expect to make any huge functional changes to the code this week, and I expect to shed a lot of blood, sweat and tears at the Code Sprint. However, I expect to know a lot more about Qt than I did before.

EDIT: At Dennis's suggestion (see his comment at 3:13 PM on September 14th), I have asked to be assigned to Issue #501 and have spent my time at the Code Sprint learning Qt with my focus being on Issue #501. In particular, I am experimenting with talkeditor.py. I would like to focus on Issue #501, so in the projections for the coming weeks, replace "issue #532" with "issue #501".

September 21st to September 27th: Read more about Qt, this time making sure to do some hands-on experimentation, or at least some tutorials. Acquire a deep understanding of the files associated with the Talk window.

September 28th to October 4th: Hopefully have some magical epiphany that solves Issue #532 (EDIT: Probably #501). Failing that, keep experimenting with Qt and making sure I understand the code, doing my best to keep my blood pressure at a reasonable level.

October 5th to October 11th: If issue #532 (EDIT: Probably #501) is solved, great. If not, take a break and start addressing issue #442. Take a close look at the Getting Started guide, looking for any typos or obvious mistakes. Make any obvious changes to the guide that need to be changed, or anything that seems easy from Dennis Ideler's checklist.

October 12th to October 18th: Research the internet to find issues people have had installing Freeseer -- possibly do some undercover work and ask what they found confusing about it? Make notes, and compare them with the current Getting Started guide. Try to find some guides about how installing stuff works in Ubuntu in general?

(EDIT, October 15th 2014: Wow, I was really obsessed with the Getting Started guide when I wrote this. Documentation is important, but I think if I want to get outside my comfort zone, it would probably be a good idea for me to look at an aspect of the project that I'm interested in, but don't know much about yet. Unit testing, maybe?)

October 19th to October 25th: With my newfound knowledge, see what improvements I can suggest for the Getting Started guide. By this time, I expect to know the Getting Started guide so well that it's burned into my retinas and I see it whenever I close my eyes, so I should have some idea of how it can be improved.

(EDIT, October 19th 2014: More likely, I'll be spending this week wrestling with PyTest in order to write the tests for my changes to TalkEditor.py. My last code-related PR took 3 weeks from start to finish. Let's see if I can get this one down to 2.)

October 26th to November 1st: Take a break from the Getting Started guide and go back to Issue #532, if it hasn't been resolved yet. I am hoping that, having taken a break from it, I can look at the problem with fresh eyes. If #532 has been solved, then move on to Issue #541.

(EDIT: In the interest of keeping my promises, I will try to schedule some time this week to look at #532. However, I will probably focus more on creating tests TalkEditor.py)

November 2nd to November 8th: Quickly reread Getting Started guide just in case any of the changes I made turned out to be really stupid. Otherwise, keep working on #532 and #541. If, by some miracle, I have solved both, then move on to the Configuration page in the documentation and see if I can make any additions to it.

(EDIT, November 1st 2014: Where has the time gone? My actual goals have diverged from my initial plan, which I hope is not too much of a problem. It might be necessary to refactor the code in TalkEditor.py to make it easier to test.)

November 9th to November 15th: Take a close look at the list of open issues again, and see if there are any that look like easy fixes, given all the knowledge I have now, whether or not I have solved #532 or #541. Also, see if I can learn a little something about GStreamer.

(EDIT, November 9th 2014: I'm still pretty busy refactoring the Talk Editor, which I think is (a) more interesting, (b) more educational, and (c) probably more useful than looking for new issues to work on right now.)

November 16th to November 22nd: Step back and skim through the entire project's code one more time. Hopefully I can understand it a little better this time, so that I can understand what everybody else in the project is working on. If I can make any suggestions, do so -- better late than never.

(EDIT, November 23rd 2014: I guess I underestimated what I could do in a month, but I overestimated what I could do in 3 months. Most of my work ended up revolving around one issue, #501, which wasn't even part of my original project proposal. I don't think this is necessarily a bad thing. I will forge ahead with the tests for the talk editor. Realistically, this will AT LEAST take me until the end of the semester.)

November 23rd to November 29th: Look at the documentation one more time and see if there are any changes I'd really like to make before the end of the semester. Research the individual parts of the program and the code as necessary, as well as researching online to see what issues other people have had.

November 30th to December 6th: Keep working on the documentation until somebody tells me to stop.

December 7th to December 13th: This is where we get into exam time. I probably will not be working on the project as much during this week. If there is any paperwork that has to be done by the end of the semester, then ideally, I would like to do it this week or next week.