Freeseer Project Proposal

Freeseer on Mac OSX

Project start date: September 23, 2011
Project completion date: December 8, 2011
Project implementer: Joseph Yeung
Sponsor organization: UCOSP
Last modified: October 06, 2011

Table of Contents

Freeseer on Mac OSX by Joseph Yeung

Abstract

The project to port Freeseer on Mac requires code changes to the existing code base such at Freeseer will compile and work on OSX Lion. New plug-ins will need to be created using YAPSY plug-ins platform. Documentation with details on how to compile and install Freeseer is also required. This project will benefit many Mac users who wish to use Freeseer to record screencast without having to boot into another operating systems or running a virtual machine.

https://github.com/fosslc/freeseer/issues/32

Document Version

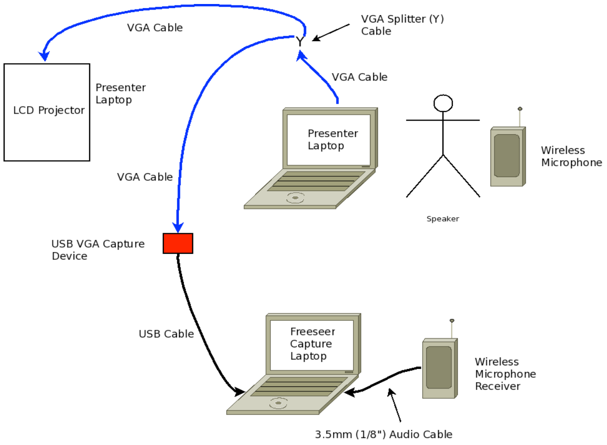
Version	Date	Changed by	Description of change
0.1	Jan. 22/2011	A. Ross	Initial version
0.2	Sept. 27/2011	J. Yeung	Initial revision
0.3	2011-10-06	J. Yeung	Added scope and future-work

Approvals

Approver	Approved Y/N	Comment
Andrew Ross		
Thanh Ha		

Background

Freeseer is a portable presentation recording station. In other words, it is an application which can record presentations, lectures, or demos for hosting in a video hosting service such as YouTube. The following diagram depicts a typical use of Freeseer.



Freeseer itself is licensed under the GPL license version 3. Freeseer itself is implemented using open source software such as Python, Qt, and Gstreamer. Since 2009, it has been used to record over 250 presentations including major conferences. This includes talks on Python, Perl, Ruby, Java, Eclipse, Fedora, Mozilla, OSGeo, and many more. Visit the videos section of http://fosslc.org to view these videos for free.

Example videos for reference:

- Beyond Nerds Bearing Gifts: The future of the Open Source Economy
- The KDE Show: An Overview of Where the Free Software Desktop Is Heading

Freeseer is optimized for recording large conferences. A powerful feature it has today is a locally cached list of talks, thus the person operating the recording station can simply pick the talk from a drop down list rather than having to type it in between talks or somehow figure out which video is which talk later. The meta data for the talk is encoded into the video itself and can be automatically retrieved later.

Requirements

1. Freeseer should maintain all or most of its functionalities - Record videos/audios and edit talks.

2. The ability to start Freeseer Record, Talk editor from command line.

3. Users should be able to run Freeseer just like other apps.

4. Freeseer should look at work similar to what it is now on Windows/ Linux

Design approach

The way the experimental branch is setup, allow plugins to be setup very quickly. To allow Freeseer to work on Mac, it will require atleast 3 YAPSY plugins; plugin for I-Sight(the built-in webcam), plugin for audio and a plugin for recording desktop.

Some coding changes such as paths and OSX specific python calls that deal with logging must be fixed.

User interface changes

None, other than fixing Freeseer such that its the same as other platform.

Program logic changes

None

Platform considerations

The code will only work on Mac OSX Lion, also maintain functionalities to Windows and Linux

Some code will be different across platform, for example the logging feature is different in OSX than in other platform, a specfic example is a call to logging.debug OSX requires “logging.DEBUG” instead of “logging.debug”

Code location

Most of the code will be in the plug-ins file

Also Readme and cmake files will also be modified.

Stream strategy

The code depends tightly with Gstreamer's support on OSX Lion

Packaging changes

Yes, Freeseer will need to be packaged properly for OSX
And Py2App will be used to package the code for Mac

Third party component dependencies

Depends on Gstreamer or other components that offer video stream for Mac OSX

Test considerations

Freeseer
- Start Freeseer
- Record Video/Audio
- Check preview
- Open Options, check whether options are being saved.

Talk Editor
- Add talks
- Remove talks
- Modify talks
- Download list of talks from RRS

Schedule

Recommended framework:

1) Project proposal (this) document agreed upon (Andrew & Thanh approve)

2) Update duties completed

- One blog to start to share the draft document (week 1) October 3
- Start on one of the plug-ins (week2) October 3 - 9
- One blog to update the document and finalize the schedule (week 3) October 10 - 16
- Start on second plug-ins (week 4) October 17– 23
- Start on installation documentation (Week 5) October 24 - 30

Freeseer on Mac OSX by Joseph Yeung

- One blog to update status and report progress (week 6) - October 31 – November 6
- Start on third plug-ins (week 7) - November 7 - 13
- Complete install documenation (week 8) November 14 – 20
- Code submitted to review (week 9) November 21 – 27
- Addressed comments in code review (week 10) November 28 – December 5
- One final blog + a short screencast video to report on work done (week 11) December 5 - 8
- One IM per week minimum to Andrew to touch base
  Participate in email mailing list. To achieve full/best feedback (i.e. A, positive progress in other projects), you must participate in emails outside your project also.
- Code committed to github regularly for backups & progress tracking
- Calls/Meetings as needed.

3) Project completed

- Code submitted for review into Reviewboard
- All issues reported as part of code review fixed or addressed via. email discussion and agreed upon by Andrew/Thanh
- Document updated and submitted into github (docs/features directory)
  - Any outstanding issues should be documented and include a bug in the issue tracker
  - Please refer to the issue in the document so one can cross reference quickly

Scope

- Port files and submission is not included in the scope of this project, but as much work will be done to prepare for port creation and submission.
- This project include the intergration of Freeseer into mac, but not include to any missing dependencies which is missing from any of the dependent packages.

Key Challenges

- As Freeseer depends on gstreamer, it might be hard to integrate mac specific stream and with little to none documentation each of the available source would need to be tested and tried.
- Py2App might be difficult to integrate.
- Testing of USB/firmware camera without the actual device.

Future-work

- If Gstreamer does not have support for OSX Lion, plugins will be setup for easy integration of videosrc as it becomes available.