
Deskorder: A Recorder for What Happens at your Desk

Andrew “Jamoozy” Correa
Ali “Alawi” Mohammad

JAMOOZY@CSAIL.MIT.EDU
ALAWI@CSAIL.MIT.EDU

MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge MA, 02139 USA

1. Introduction

In this paper, we present the beginnings of a toolkit (consisting of a set of binaries and file formats) dedicated to recording, storing, and playing back time-synchronized pen data and audio. We call our program “Deskorder”.

As of this publication, you can check out and contribute to our code from our SVN repository, at:

<http://deskorder.googlecode.com/svn/trunk>

For information on how to get and run a copy for yourself, see the `README.txt` file under the link above.

Deskorder is under the GPL v.3 (1); please contribute!

1.1 Goal

Deskorder was conceived by Ali Mohammad to enable him to record recitations and compile quick explanations, thereby allowing students to focus on class instead of having to take time copying down the board. Andrew Correa started contributing to Deskorder when he realized that it could be used as a tool to aide his sketch recognition research.

1.2 Ontology

A run of Deskorder is organized into what we call a lecture. Originally, one lecture corresponded to an actual lecture—the period of time between when a professor or TA starts class and when he or she ends class—however more generally a lecture can be thought of as a “session”, not necessarily taking place in a classroom.

Each lecture is a collection of at least 0 audio snippets, at least 1 mouse move, and at least 1 slide (fig. 1). Each slide can be thought of as a separate sheet of paper. New slides are meant to be used when transitioning to a new idea, or when there is not enough room on the current slide to continue drawing. A slide is simply a collection of pen strokes—the period between mouse-down and mouse-up events—and mouse

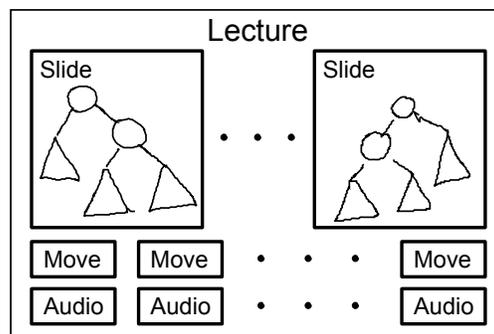


Figure 1. Structure of a lecture.

moves—the period of time between a mouse-up and a mouse-down event¹.

1.3 Purpose of this Document

This document describes the binaries and file formats we have created and plan to create to record, store, and play lectures. Its purpose is to get you, the reader, excited about Deskorder. We would like you to use this program and contribute by giving comments and suggestions, and maybe even contributing some code! We sincerely hope that Deskorder can be useful to you.

2. Related Work

There are several free software applications that enable you to draw and record freehand sketches and store them as images (2; 3) or vector graphics (4), however we are aware of no programs—free or otherwise—that also record fine-grained time stamp data to allow real-time playback.

There are proprietary systems (5) that record notes and audio, but these systems rely on special hardware or are prohibitively expensive for a private user².

¹Moves are included to allow the lecturer to point something out without having to draw a stroke

²Prices range from \$129.95 for the smallest-offered PulseTM Smartpen (11) with 2GB of storage, to \$1,595.95

In contrast to these proprietary systems, our system runs on standard computers—requiring at least a mouse, and at most a standard drawing surface and microphone—and is free software, so accessible to all.

We are aware of several desktop screen capture programs, that record everything happening on your desktop. One such program, Camstudio (6), comes with its own video codec that stores audio losslessly. It is possible to use Camstudio in combination with e.g., Xournal to perform approximately the same task as Deskcorder. However a) videos produced this way have a fixed resolution; b) this requires running multiple programs; and c) much information is often recorded that is useless for the purposes of simple demonstrations³. Programs like Camstudio were made to be able to share the answer to the question, “How do I configure X to do Y ?”, instead of demonstrating how, for example, binary tree rotations work to a novice programmer.

We believe that a huge advantage Deskcorder has over other methods of recording strokes, is that Deskcorder combines the accessibility of free software with the power to produce an output that is as viewable as a video file, while maintaining scalability and time synchronization.

3. Programs

In order for these tools to be widely adopted, one desktop program and one web-based program are included. The desktop program serves as a convenient way to record lectures given in front of a class or lectures made at a desk in private, while the web program offers easy-access playback, allowing anyone wishing to view a lecture to view it, without having to install another program.

3.1 Desktop Program

Our desktop program is written in Python (7) and GTK+ (8). This program can record and playback Deskcorder lectures as well as convert a lecture to a PDF file or PNG files⁴. Further, an “Export to Flash...” feature is being made that will produce a single, stand-alone SWF file meant to be posted online and viewed in a browser. Fig. 2 shows a screenshot of our program playing back a DCB file of a lecture describing how to perform a linked list insertion.

for NoteAmation!TM set of pen, pad, and paper.

³For example, the background and the OS application decoration.

⁴Full integration of these features is not complete.

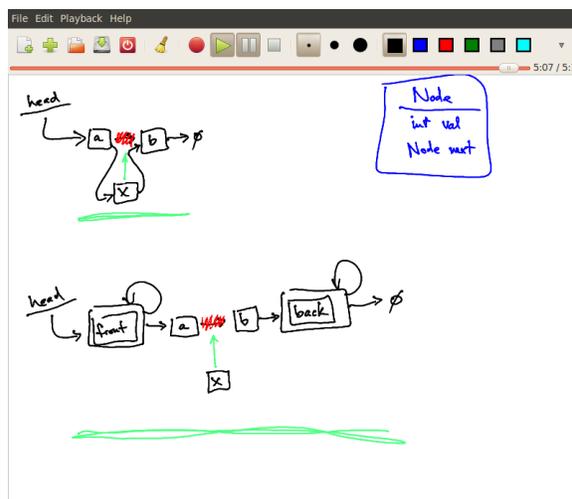


Figure 2. A screenshot of the Python/GTK+ version of Deskcorder in use while explaining linked-list insertions.

3.2 Web Programs

We have plans for two web programs: a Flash (9) version that is currently under construction, and an HTML 5 (10) version that will be implemented when HTML 5 becomes more widespread. Work on the Flash version has already begun, in the form of a SWF-generating library written in Python. We decided to make a Flash version, because Flash is widely used (97% of users according to slashdot.org), but that it is undesirable since it is a proprietary (though open) standard, and we intend to deprecate it as soon as suitable free standards are adopted by the public.

A more practical difficulty we are experiencing with Flash, is its inability to open arbitrary files. This makes development of a stand-alone, web-accessible Flash application cumbersome. Ideally we would like an application that “just reads in” our DCB file format and plays it as the desktop version would. Instead, the desktop version exports a stand-alone Flash file that stores and plays back a lecture. We expect the HTML 5 version of the web application will not require the same special export feature to be maintained.

3.3 Deskcorder Binary File Format

The Deskcorder binary (DCB) file format is what Deskcorder uses to save and load lectures. Fig. 3 shows a graphical representation of the file format.

As discussed in § 2, the strength of this file over other recording formats, is that it uses lossless vector graphics and audio. The downside is that this is a specific format that requires a special reader.

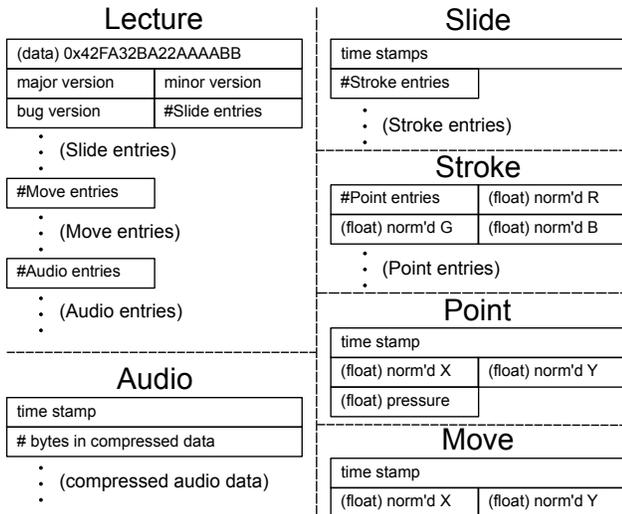


Figure 3. An outline of the most-recent Deskcorder binary file format (v.0.1.1). Long boxes are 8 bytes, short boxes are 4. All boxes without types are integers. Position, color, and pressure data are all in the range $[0, 1]$ to simplify code wishing to display a DCB lecture.

4. Future Work

Since Deskcorder is still in its infancy, there is much left to do. Firstly, we plan to finish the Flash application and the Deskcorder Flash-export feature. In the longer-term (starting in the neighborhood of half a year from the time of this writing) we plan to begin development on an HTML 5 version that we think will not suffer from the drawbacks of Flash (mentioned in § 3.2).

We plan to improve the PDF- and PNG-export process by a) integrating the PDF and PNG export features into the GTK+ interface; b) lifting the “one-slide/one-page” rule and allowing export of arbitrary points in time in the lecture; and c) allowing the addition of text descriptions for each page. These changes will make DCB files more accessible, as you will be able to export the lecture into a series of hand-selected, labeled snapshots in time, with descriptions explaining what is happening in the page.

In an effort to increase Deskcorder’s use, we plan to create versions that run on all three major operating systems: Windows, Linux, and Mac OS. We are still investigating whether to make separate GUIs for each OS or choose one GUI toolkit that works on all three OSs (e.g., Qt or Python’s TkInter).

In the very long term, we acknowledge the possibility that it may be worthwhile to create an optimized version written in a lower-level language (e.g., C). However, this decision is still far in the future.

5. Conclusion

We have presented the beginnings of a set of programs and file formats that we hope will be useful to the broader population in giving lectures, working with pen-based systems, or just explaining something to friends or colleagues. We encourage suggestions via email and ask that you become a member of our community! We hope you will enjoy using Deskcorder as much as we do.

Acknowledgments

Ali Mohammad started creating this program in the Spring of 2009 to help him teach his classes. Andrew Correa picked up the project and ran with it in the Summer 2010 to use as a tool in his research in multi-modal understanding.

Ali Mohammad is credited with envisioning the project, starting its development process, and all work on the Flash application. Andrew Correa is credited with developing the file formats and expanding the technology to be useful on the Desktop (the Python/GTK+ version).

All development was done on Ubuntu GNU/Linux systems.

References

- (1) The Free Software Foundation <http://fsf.org>
- (2) Xournal <http://xournal.sourceforge.net/>
- (3) GIMP: The GIMP Image Manipulation Program <http://www.gimp.org/>
- (4) Inkscape <http://inkscape.org>
- (5) NoteAmation!™ by Rover INK™ <http://www.roverusa.com/roverink/noteamation.asp>
- (6) Camstudio <http://camstudio.org>
- (7) Python <http://python.org>
- (8) GTK+ The GIMP Toolkit <http://gtk.org>
- (9) Adobe Flash <http://www.adobe.com/products/flashplayer/>
- (10) HTML 5 <http://dev.w3.org/html5/spec/>
- (11) Livescribe Smartpen <http://www.livescribe.com/en-us/>