coffee_shop: Investigations into modern word processors.

Nathaniel Welch, Dr. Clark Turner

California State Polytechnic University

May 22, 2011

Abstract

My senior project was spent building a desktop application similar to WriteRoom and OmmWriter. These two applications are both word processors for the Macintosh OS X operating system. Both of these applications are designed to be a reset on word processing software (cite?), bringing their interfaces back to the days of Microsoft Word 3.0 and Word Perfect. They do this by spending more time on focusing on the design of the software interface, and focusing on keeping it minimalistic, instead of filling the product with new obscure features, which is a common complaint against the current iterations of Microsoft Word (cite?).

The final application, named coffee_shop, ended up not meeting my expectations. Having spent most of my education on developing applications for the internet, instead of the desktop, I ran into pitfalls which, if this had been an internet application wouldn't have been an issue.

Contents

1 Introduction

Depending on the job, people use different tools. Some tools are incredibly specialised, such as post hole diggers and PVC pipe warmers. Others are much more generic, such as hammers and cameras. Notice though, that in the hardware world, the majority of tools serve one job. That job, such as hammering, can be applied to a wide variety of ways (hammering in a nail, breaking apart structures, putting stakes in the ground). In the software world, historically tools have been built to be much more generic. Microsoft Word for example not only let you write documents, but also create spreadsheets, resumes and many other things. While this has let Microsoft make more money by selling their software to a wider variety of needs, it has created software that is hard to use and hard to maintain. To fight this, companies and individuals are starting to create software that have less features and are easier to use.

2 Problem description

The problem coffee_shop tries to tackle is whether or not developing a desktop application focused on writing, instead of formating, is a good idea.

3 Survey of relevant work

Before starting the project, I surveyed a variety of word processor programs and interviewed individuals about their writing habits. The survey I distributed was simple and open-ended. I asked users what types of documents they wrote, what they disliked about their current word processor and what they liked and wanted in their word processor.

I discovered that there were two groups of users of word processors in my survey group. There were those who wrote in a corporate or academic environment and those that wrote for themselves. The users that wrote more for academia and corporate positions tended to want lots of formatting options. Users writing for themselves usually wanted something that hid everything and let them just write.

Every single one of my responders despised Microsoft Word's auto-correct but still wanted spell check.

I decided to name my user group Johnny, based on two different people that I interviewed and the responses from my survey. Johnny writes fiction in his free time and aspires to be a writer. He is currently employed doing other things, so he uses his word processor as he commutes and during his time off. Most of what Johnny writes tends to be one or two pages, but he has been known to turn out novels depending on his mood.

3.1 Professional Offerings

As stated earlier, there are a variety of choices in the word processor market. The ones I tried and examined were chosen both by their reputation and based on how easily I could get my hands on a copy.

3.1.1 Microsoft Word 2010

...

3.1.2 Microsoft Word 5.5

Originally released in WHEN, Microsoft Word 5.5 is an interesting beast. 5.5's initial downside is that it needs to be run in dosbox. The cool thing about dosbox though, is that it means I can run 5.5 on Linux and on just about every other machine that it supports. But 5.5 comes from before the days where everything is just one executable. The user install experience is incredibly slow and time consuming. 5.5 extracts four hundred files before setup, and then another two hundred during the install process.

(Picture here...)

Once you are inside 5.5, the experience is relatively nice. Document navigation was done entirely via the keyboard. It reminded me a lot of Pico, and was a joy to use. The user can also select text with the mouse, and we can see how the experience in 5.5 turned into that of Microsoft Word 2010.

5.5 offered no color or font customizations, although the user could go full screen, which was very nice. Page breaks were implied with an emphasized dotted line.

3.1.3 Pages

The Apple offering in the word processor space.

3.1.4 Open Office

Needs to be turned into paragraph:

- A substantial amount of features.
- Very easy to just start typing, although you feel like you need to format your writing.
- Has a full screen mode, crashes some times when coming in and out.
- export to pdf, very useful
- Spell check. Seemed much more obvious than the other apps
- Had to select all of text to change font, colors, etc.
- Supported so many fonts it blew my mind.
- very easy to zoom with slider in bottom right.

- Lots of buttons that I had no idea what they did, kind of scared me.

3.1.5 Vim

According to the Vim website, vim.org, "Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems" [1]. I love Vim, and since starting this project, I've pretty much decided that in almost every scenario, I would use Vim to write things. I currently am using Vim to write this paper, I used it to write almost all of the code for coffee_shop and I have been using it pretty exclusively for the last four years.

Vim has its issues though. It has a really difficult learning curve. In fact, without working for a company where almost every developer used it, I probably would have given up long ago. Most people give up quickly because they do not understand why Vim tries to make sure you never use the mouse. In their defense, this is fair, the user interface does not tell the user much at all, and expects the user to know all of the magic incantations to get things to happen. I have heard that emacs is the same way, but I have never used it.

3.2 Similar implementations

These are programs I am trying to emulate.

3.2.1 WriteRoom

Needs to be turned into paragraph:

- Lets you change font and colors
- obscure key combinations to do any sort of display settings

- ESC lets you turn app into normal text editor. Almost seems like a wrapper around text edit.
- Growl notifications still appear
- Wrap to page is pretty cool

3.2.2 OmmWriter

Needs to be turned into paragraph:

- Lets you change font (sans, serif, script), font-size, themes (background/color combinations), background music and typing noise
- I found typing noise very soothing, but I preferred to pick my own background music.
- very limited choice in themes. No Black/Green, low contrast option
- the sound the program produced an emotion that seemed to promote typing, although I'm not sure if I could say it was distraction free.
- UI was much more visible
- gave scroll bar and word count
- adjustable work area

4 coffee_shop

Screen shots of the program.

Brief mention of what was implemented, not implemented, etc.

5 Evaluation of coffee_shop

5.1 Development decisions

I made some decisions while developing and designing coffee_shop, which may be considered controversial or unintelligent. The two big decisions I made were to use Ruby as my programming language and Qt as my graphics framework.

5.1.1 Ruby

Ruby is an interesting beast. I selected it because I had little experience with the language and it had a very large community online.

5.1.2 Qt

Java and C++ libraries are powerful, but do not translate to Ruby well.

5.2 Features and their implementations

5.2.1 File writing

Pretty simple, although lots of discussion.

5.2.2 Pagination

The bane of my existence.

5.2.3 Printing

This was surprisingly easy thanks to Qt.

5.2.4 Customization and preference storing

Talk about how I did it (YAML) and how other programs do it.

6 Conclusion

References

[1] [Online]. Available: http://www.vim.org/about.php

[2] "Ieeeannot: A latex ieee annotated bibliography style template." [Online]. Available: http://www.barik.net/sw/ieee/

> the web page where you can download the IEEE annotate style to allow annotations in a bibliography. Put it in the same folder as the .bib and .tex files.