15-112 Fundamentals of Programming
F11 Term Project Design Document

Name: Mirai Akagawa
AndrewID: makagawa
Section: D

NoteIt – a Digital Note Taking Experience

Overview:

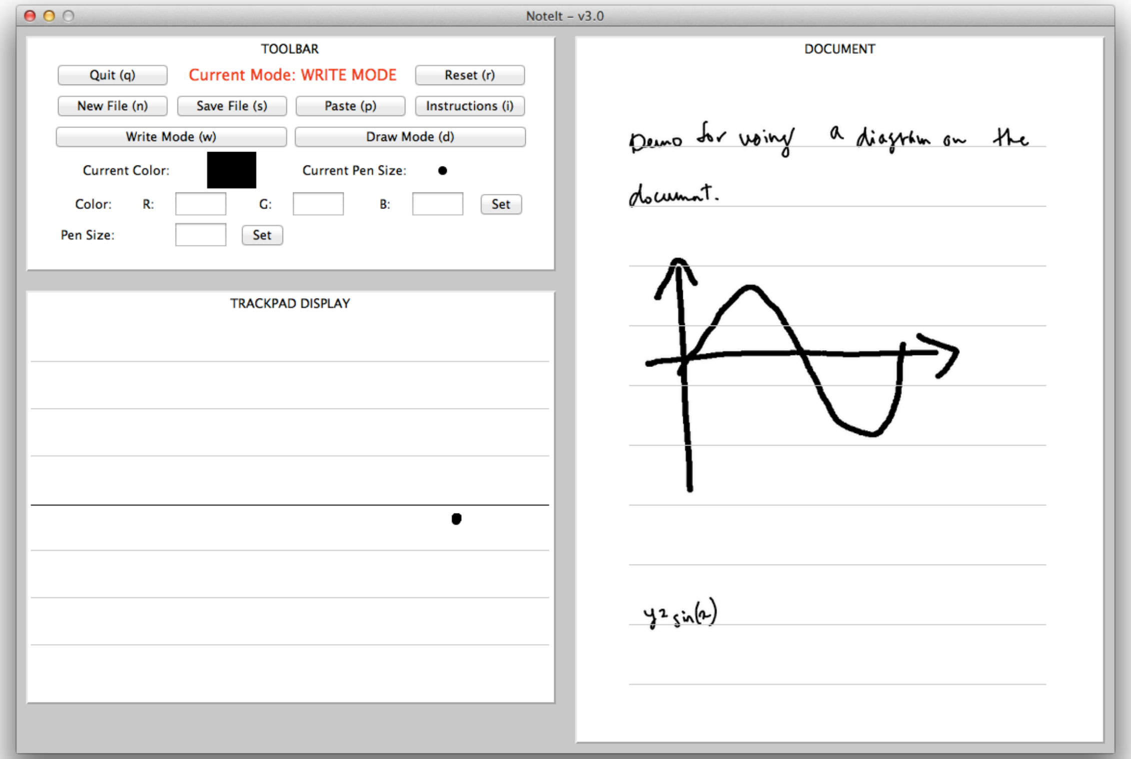
Picking what to do for the term project was no easy task – in fact, the broadness of the assignment made it difficult to decide where to explore. One thing I kept in mind however was not to be cliché. I wanted to do something that no one else will do, and this is one of the reasons why I settled into this project. I also wanted to make a utility tool that is user friendly, and enhances someone's life in someway. Although this program itself is not nearly as well as a commercial product could be, it is enough to get the general idea across of how such an program can evolve into something useful.

In this rapid growing era of technology, I do believe that for some users it is not so easy to communicate with the computer: typing, is a foreign language. Users such as the handicapped and the elderly can benefit from a program like this that connects the traditional methods of noting down information and this emerging era of computers. What's more, it is completely eco-friendly, as it does not consume any paper.

Idea:

Firstly, there are two modes in this program: Draw and Write. Let me explain Draw first as it is naturally the simpler one. This is basically using the trackpad like paper, and displaying what is being drawn on the trackpad on the computer screen. Usually, computer cursors use relative positioning to change positions. So to draw a picture, for example, in a paint program requires you to move the cursor to the desired location, then dragging it to your destination. This program doesn't require that, as it just uses the absolute position of the trackpad. If you touch the top left part of the trackpad, a dot will appear on the top left part of your canvas.

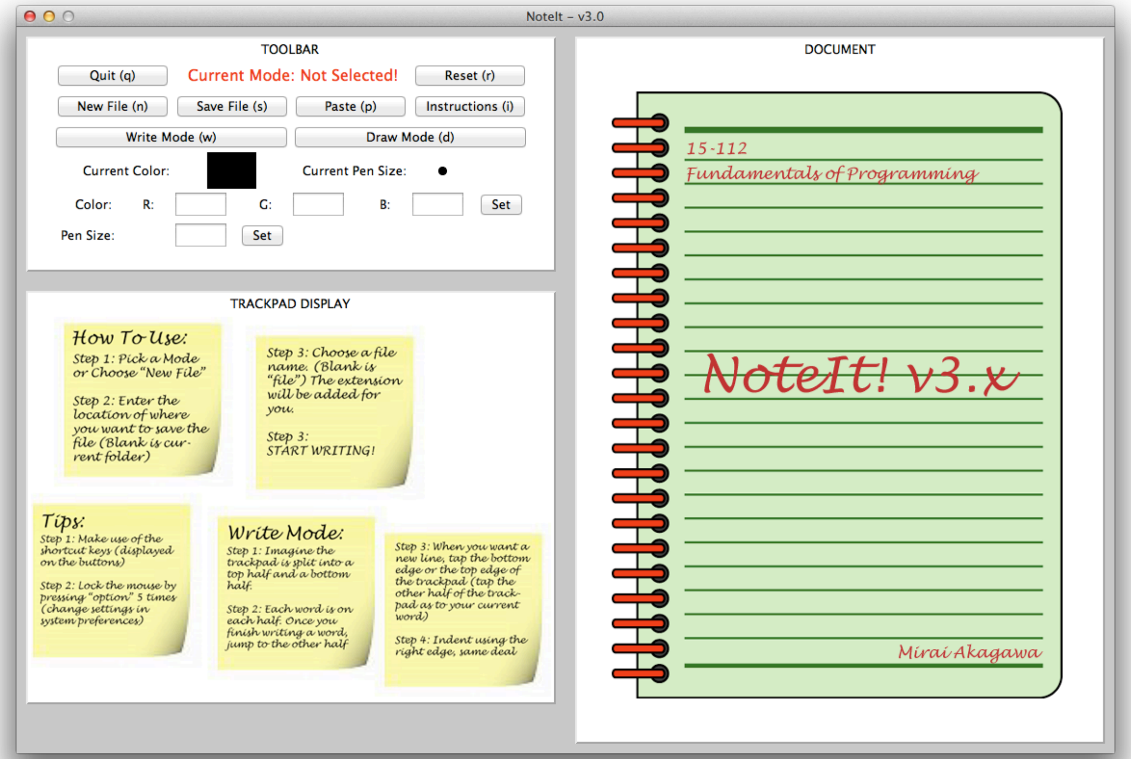
Now taking this idea further, I wrote the Write mode. The write mode basically takes multiple images of the Draw mode and compiles them into a document. A slight difference however, is that instead of using the whole trackpad as one big canvas, I split the canvas into two, horizontally. This way, the program can recognize when the user finished writing a word: simply when the pen moves to the other half of the trackpad. It is also easier for the user to keep writing when the motion goes up and down, compared to writing on the same location constantly. As soon as the user starts writing a new word, the program will crop the previous word into just the part where the letters exist and paste it into the document adding a space next each word. Repeating this process will lead to a series of words, creating a sentence.



Putting a diagram in a document

Features:

- **New File:** Creates a new file depending on which mode the user picks. The user can also enter a desired file name, or leave it blank and use a default "file.png" name.
- **Save File:** Saves the file in the current directory, whether it is a document or a drawing.
- **Paste:** When the user starts a document and wants to add an image using the draw mode, the user can do so and paste the image onto the document. This enables the use of diagrams in the document.
- **Instructions:** Whenever the user needs to access the instructions, the user can press "i" or use the button on the toolbar. A popup window appears which has the general instructions on it.



How To Use:
Step 1: Pick a Mode or Choose "New File"

Step 2: Enter the location of where you want to save the file (Blank is current folder)

Step 3: Choose a file name. (Blank is "file") The extension will be added for you.

Step 3:
START WRITING!

Tips:
Step 1: Make use of the shortcut keys (displayed on the buttons)

Step 2: Lock the mouse by pressing "option" 5 times (change settings in system preferences)

Write Mode:
Step 1: Imagine the trackpad is split into a top half and a bottom half.

Step 2: Each word is on each half. Once you finish writing a word, jump to the other half

Step 3: When you want a new line, tap the bottom edge or the top edge of the trackpad (tap the other half of the trackpad as to your current word)

Step 4: Indent using the right edge, same deal

<u>Start Screen</u>

- **Write Mode:** Turns the write mode on or off. If there are no existing files open, the program will prompt the user for a file name and starts a new file. If there already is a file, it will just keep adding to the existing file.
    - **Top/Bottom Edge New Line:** During write mode, instead of writing a new word, the user may choose to tap the bottom edge or the top edge of the trackpad (depending on which half the current word is on. If writing on the top half, then tap the bottom half, and vice-versa.) This means "new line". When the user starts writing the next word, it will start on a new line.
    - **Right Edge Indent:** Likewise, tapping the right edge of the trackpad will indent. This also has to be on the other half of the word currently writing on. This will add an indentation to wherever the user is currently writing.
    - **Left Edge Cross Out:** If the user makes a mistake and wants to cross out a word, the user can tap the left edge (as a new word) and the program will ignore that previous input.
    - **Pasting Diagrams and Images:** if the user wants to add an image to the document, he or she can simply turn the write mode off by pressing "w", triggering the draw mode by pressing "d", drawing the desired image, and then pressing "p" for paste. This will automatically adjust the image drawn

on the canvas and paste it onto the document. The user can keep writing below the image as they wish.
    - **Keyboard Shortcut "w":** The user can stop the write mode anytime by pressing "w". Maybe the user wishes to look at the instructions, or just take a break. Either way, once "w" is hit again, the user can continue writing from wherever they ended last time.
- **Draw Mode:** Same as the Write Mode but uses the whole canvas. Again, using the "d" key, the user can turn on and off the draw mode. Perhaps it is useful for changing colours or changing the size of the pen width.
- **Changing Colour:** The user can input an integer value between 0 and 255 to represent the RGB values of the desired colour, and clicking set. A sample colour will appear on the toolbar to let the user know if that is what was desired
- **Changing pen size:** No need to find a different pen, just change the size of the pen tip right on the screen. Integer values between 1 and 50 are accepted, representing the radius of the point drawn on the canvas.

The Code:

This program uses ctypes to extract data directly from the trackpad, a command a normal python code cannot execute. To decode this mystic module was one of the most challenging parts of this project, as a lot of it was nonsense. However, I was able to decode most of what was happening, and was able to edit the contents for use in my purposes. Mainly, it was making the module return the points of the trackpad.

An issue I faced early in the coding process was that the timerFired on Tkinter was too slow. The fastest refresh rate was 1ms, which was not fast enough for this program as it needs to constantly fetch the coordinates of the trackpad position. To combat this issue, I used sets to store values that the module was returning while Tkinter was refreshing, and iterate through the set to regain all the values that were to be lost. Then, I clear the contents of the set as adding too many items leads to the program lagging.

This program has three separate canvases packed into a single frame, because of the nature of the features needed. However, this enabled me to keep track of each canvas separately and update the information as I needed, instead of through timerFired. The toolbar makes extensive use of the grid module in Tkinter, which was not so easy to implement but makes positioning the buttons slightly easier.

PIL (Python Imaging Library) was used to add the .png format, for opening and saving files. PIL was also used to resize images, crop them and paste them into different images. At most three images were kept, for drawing (draw mode), word (write mode) and doc (also for write mode). Each of these images were manipulated to construct what the user wanted.

Adding useful features (new line, indent, cross out, paste) to the write mode was one of the most difficult parts of this project. Although the concept seems pretty simple, because of the unique nature of the program, a lot of experimentation with the trackpad's response and the program's function was required before all were working together.

Overall I implemented keyboard shortcuts to each button as at times it is difficult to reach for the button when using the trackpad at the same time. It also has graphical error messages to any mal input by the user.