Project Proposal

Project Description:

This is a text editor with encryption and file management system. Text editor and file management system supports using mouse and keyboard as graphical user interface. Encryption will be done as AES 128.

Competitive Analysis:

The website aesencryption.net has a very similar online application that has a text editor implemented which encrypts the text. The text editor allows dragging to temporarily highlight multiple characters, copy and paste, automated line change when text is outside the range of textbox, scrolling with a scroll bar, and movement of cursor via mouse and arrow keys. However, there aren't functionalities such as different alignment methods, does not support tab keys as input, permanent color text highlight, and save of file. Also, there is a problem that when there is a file the user wants to encrypt, the file should be copied and pasted by the user. Also, there is not support for different fonts.

TP2: ~~For my app, in order to differentiate with what is already available on the web, I will not only include the functionalities that aesencryption.net already has, but also include color highlighting, color text highlighting, multiple alignment methods and support for encryption on an uploaded file. Also, my app's text editor will allow automated text alignment even when changing the dimensions of the text editor.~~

In order to differentiate my app from the online application mentioned above, I will include features to import and export files which the application mentioned above does not support.

Structure Plan:

Encryption code would be created on a separate file and imported onto the UI file. Text editor and file management system will be written in the same file in different functions so that they are called in redrawAll().

Algorithmic Plan:

The most complex algorithm used in this app would be the AES encryption. The AES encryption algorithm would be written in top-down design and use abstraction in each layers so as to simplify the algorithm. The highest level will be running encryption by first running the function that does the "key expansion" and then run a series of loops of functions that perform "byte substitution", 'shift rows', "mix columns", and 'key add' according to the AES algorithm definition. These functions will be implemented by using predefined lookup table and addition and multiplication. The lookup tables are given in the AES definition, hence will be implemented as dictionaries. Arithmetic operations for

AES encryption is not defined on the real numbers, but are defined on a finite field of GF(2^8). Since algebra is defined on a field, addition, multiplication, and both additive and multiplicative inverses are defined. However, the operation for addition and subtraction (i.e. additive inverse) are same (bitwise exor) and the need for multiplicative inverse is eliminated by using lookup tables since calculating the inverse is computationally heavy. In defining multiplication, two strings length 8 representing bits are interpreted as coefficients of polynomials with the maximum order of . The first step is the to perform a polynomial multiplications. This will be performed by shifting the first polynomial's coefficients (bits) to the left for each coefficients of the second and performing the predefined addition to the result variable if the second polynomial's coefficients being dealt with the multiplication is 1. This will also require addition not on GF2^8 but on GF2^16. After that, the second part of the multiplication is doing polynomial modulus operation on the resulted polynomial multiplication string. The modulo will be done with a predefined irreducible polynomial x^8 + x^4 + x^3 + x + 1. This will be implemented by long division, which is implemented by performing subtraction (= addition in GF2^8). To summarize, the multiplication in GF(2^8) of A(x) and B(x) would be C(x) = A(x) · B(x) mod P(x) where P(x) = x^8 + x^4 + x^3 + x + 1.

Timeline Plan:

Friday night : finish implementation of temporary text highlighting

Saturday midday : finish implementation of automatic text alignment according to changing text editor size

Saturday midnight : finish implementation of editing text, file read and create for encryption

Sunday morning : implement drag copy and past in the middle of text

Sunday afternoon : implement permanent color text editing, backend for file management, file read and write for text editor

Monday midnight : implementation of graphics for text editing

Version Control Plan:

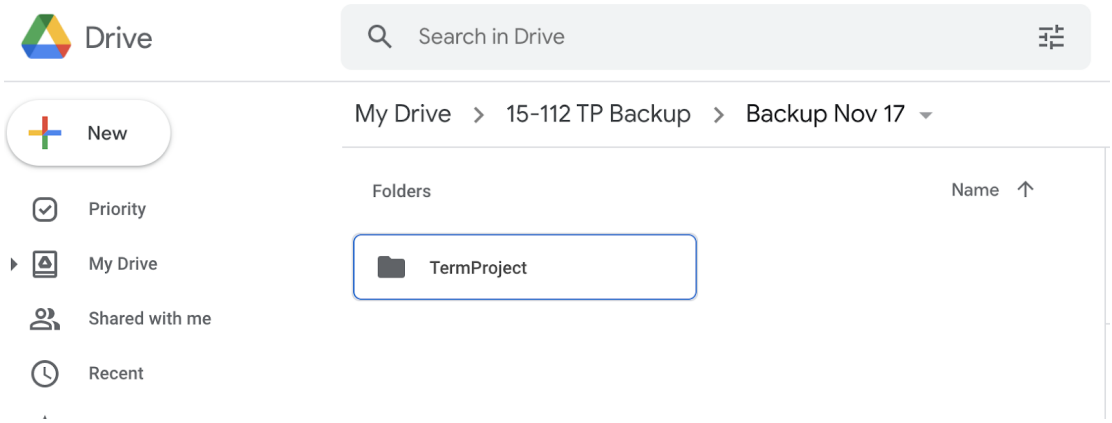
Upload multiple times on autolab:

Ver	File	Submission Date	Late Days Used	Total Score
2	junheeo@andrew.cmu.edu_2_handin.zip	2021-11-18 02:40:41 -0500	Submitted 0 days late	0.0
1	junheeo@andrew.cmu.edu_1_handin.zip	2021-11-18 02:26:47 -0500	Submitted 0 days late	0.0

Page loaded in 0.027605268 seconds

Autolab Project · Contact · GitHub · Facebook · Logout v2.7.0

Backup on Google drive whenever new function created:



Module List:

~~copy~~
~~cmu_112_graphics~~
~~string~~

~~If use amazon glacier :~~
~~(import argparse~~
~~import logging~~
~~import os~~
~~import boto3~~
~~from botocore.exceptions import ClientError~~
~~)~~
~~Tech demo finished, not included in MVP~~

TP3
No modules need to be downloaded.

copy
cmu_112_graphics
string
pickle

TP3 Upgrade:

- AES Decryption
- RSA Encryption, Decryption
    - Extended Euclidean Algorithm

        - Algorithm to approximation of huge primes (probabilistically check primes greater than 2^(256) using Miller-Rabin Test)
- Share Public Key and Max word size with another app over Socket
    - Server
    - Client