Rob Just  
CS 241  
8/4/10

Final Project Updated Design Documentation

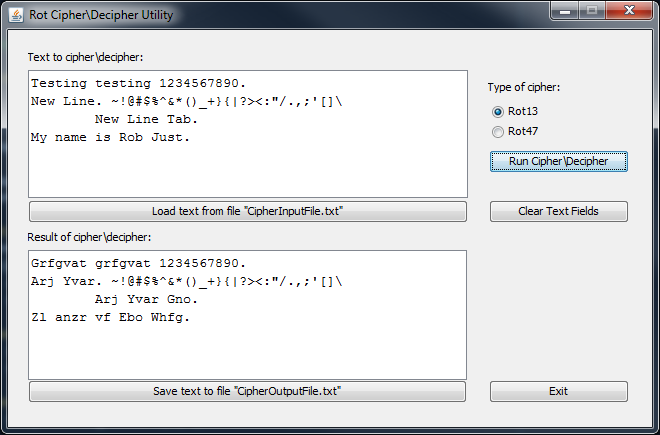
**Description\Purpose:**

Rot13\Rot47 Cipher Utility  
  
Many times I’ve wanted to send someone an email or post a message on a message board that I didn’t want someone to be able to read at a first glance. A common reason for this is to prevent spoiling TV shows or movies for people who haven’t seen them. A simple and fun way to do this is to use a rotation cipher to shift the letters of the text by a number of places to hide what the text really says. A popular cipher on internet message boards among other places is Rot13. Rot13 is a Caesar cipher that shifts each character in a string of text by 13 places in the alphabet. Another Caesar cipher is Rot47 which includes non-alphabet characters in the ASCII table and shifts the characters by 47 places from ASCII values 33 to 126. These ciphers have no security use as they can be reversed simply by running the text through the cipher again. They are simply used for entertainment and non-secure purposes.

This program will allow a user to type in, or load from a file, text to be ciphered\deciphered using either the Rot13 or Rot47 cipher. The results of the cipher\decipher can then be saved out to a text file.

1. The user will enter the text they wish to cipher\decipher into the “Text to cipher\decipher” text area box. The user can also click the “Load from text file…” button to load text already saved in the file “CipherTextInput.txt” located in the program directory. The program will display an error if the file does not exist.
2. Then they will select the cipher they wish to use by clicking the “Rot13” or “Rot47” radio buttons.
3. The user will then click the “Run Cipher\Decipher” button which will start the cipher\decipher process and output the results to the “Result” text area box. The program will display an error if no text was entered into the “Test to cipher\decipher” text area.
4. The results text area is not editable by the user, it is read-only. The user may save the results to the file “CipherTextOutput.txt” located in the program directory or copy the text out using their systems keyboard shortcut for copy. The program will display an error if the file is in use or write protected.

**Form Design:**



The above screenshot is an example of what the form looks like.

**Input Example:**

This is test input. 1234. My name is Rob.  
 **Output Example:**Rot13 = Guvf vf grfg vachg. 1234. Zl anzr vf Ebo.  
Rot47 = %9:D :D E6DE :?AFE] `abc] |J ?2>6 :D #@3]

**Pseudocode:  
  
 start** Initialize window and controls  
 Frame mainFrame  
 TextArea cipherTextArea  
 TextArea cipherResultTextArea  
 RadioButton rot13Radio  
 RadioButton rot47Radio  
 Button loadFileButton  
 Button saveFileButton  
 Button cipherDecipherButton  
 Label inputLabel  
 Label resultLabel  
 Wait for user input\interaction or user to exit program **Stop**

**loadTextFromFile() *\*Called by loadFileButton Click Event Handler*** declare variables  
 StringBuilder cipherText  
 open file inFile “CipherInputFile.txt”  
 while not end of file  
 read inFile into StringBuilder cipherText  
 end while  
 Set text of cipherTextArea to cipherText  
 close inFile  
 Display message box “File loaded successfully” **return**

**saveTextToFile() *\*Called by saveFileButton Click Event Handler***

declare variables  
 String cipherText  
 open file outFile “CipherOutputFile.txt”  
Set cipherText to the text of cipherResultTextArea  
while not end of file  
 write cipherText to outFile  
end while  
close outFile  
Display message box “File saved successfully”

**return**

**performCipher() *\*Called by cipherDecipherButton Click Event Handler*** declare variables  
 String cipherInputText  
 String cipherResultText  
 Set cipherInputText to text of cipherTextArea

If cipherInputText is empty  
 Display “Please enter some text to cipher”  
 Else  
 If rot13Radio radio button is checked  
 Set cipherResultText to return value of RotCipher.getRot13String(cipherInputText)  
 Else if rot47Radio radio button is checked  
 Set cipherResultText to return value of RotCipher.getRot47String(cipherInputText)End If  
End IfSet text of cipherResultTextArea to cipherResultText **return**

**RotCipher.getRot13String(String inputString)** declare variables  
 char currentChar  
 int counter  
 StringBuilder outputString  
 Set this.inputString to inputString  
 Set counter equal to 0  
 Loop until counter equals the length of this.inputString – 1, increment counter by 1 each iteration  
 Set currentChar to the char at index of counter in the this.inputString  
 If currentChar is greater than or equal to “A” and less than or equal to “M”  
 Insert currentChar + 13 into outputString at index of counter  
If currentChar is greater than or equal to “N” and less than or equal to “Z”  
 Insert currentChar - 13 into outputString at index of counter  
 If currentChar is greater than or equal to “a” and less than or equal to “m”  
 Insert currentChar + 13 into outputString at index of counter  
 If currentChar is greater than or equal to “n” and less than or equal to ”z”  
 Insert currentChar - 13 into outputString at index of counter  
 Else   
 Insert currentChar into outputString at index of counterConvert StringBuilder outputString to a string and set this.outputString to its value

**return** this.outputString

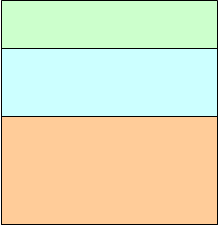
**RotCipher.getRot47String(String inputString)** declare variables  
 char currentChar  
 int counter  
 StringBuilder outputString

Set this.inputString to inputString  
 Set counter equal to 0  
 Loop until counter equals the length of this.inputString – 1, increment counter by 1 each iteration  
 Set currentChar to the char at index of counter in the this.inputString  
 If currentChar is greater than or equal to ASCII value 33 and less than or equal to value 79  
 Insert currentChar + 47 into outputString at index of counter  
 If currentChar is greater than or equal to ASCII value 80 and less than or equal to value 126  
 Insert currentChar - 47 into outputString at index of counter  
 Else   
 Insert currentChar into outputString at index of counter  
 Convert StringBuilder outputString to a string and set this.outputString to its value

**return** this.outputString

**RotCipher.getOutputString()  
return** outputString **RotCipher.getInputString()  
return** inputString

**UML Diagram:**

****

+RotCipher()  
+RotCipher(string)  
+getRot13String(string): string  
+getRot47String(string): string  
+getOutputString(): string  
+getInputString(): string

-outputString: string  
-inputString: string

RotCipher