Graphical Enigma Simulator

System Software Requirements Specification

Glossary

Conformance Glossary

The following keywords are used to differentiate between different levels of requirements and optionality, as defined in IEEE Std 100-1992 [RD11].

Shall: indicates a mandatory requirement. To ensure interoperability with other products conforming to this standard, all mandatory requirements must be followed strictly with no deviation.
Should: indicates a recommended but not mandatory requirement. Allows flexibility of choice between several possible alternatives while indicating a strongly preferred alternative. Indicates that a certain course of action is desirable but not mandatory, or indicates that a certain course of action is deprecated but not prohibited.
May: indicates a suggested course of action without implying preference over any other possible course of action.

Introduction

A Graphical Enigma Simulator is to be developed. This document provides a specification of the requirements for this system.

Background / Vision

The Enigma machine was used by the German forces during World War II. It was used to communicate with their allies using the means of telecommunication. Using the Enigma machine they sent over ciphered text which they believed was unbreakable. Before World War II, Polish mathematicians finally cracked the Enigma machine, which help the British defeat the Germans.

The purpose of this application will be to simulate the inner workings of an Enigma machine by graphically representing the process of encrypting and decrypting the text.

The design and development of the application will be proceeding immediately.

FUNCTIONAL REQUIREMENTS

Menu/Options

R1 Main Menu
The simulator shall have a main menu.
Rationale: To allow navigation of the different options available.
Priority: High.

R2 Main Menu – options
The main menu shall contain three options: Encrypt, Decrypt and Exit.
Rationale: To allow the user to choose from the different options at the outset.
Priority: High.

R3 Encrypt Option
The simulator shall allow the encryption process to be simulated after selecting Encrypt from the Main Menu.
Rationale: To allow the user to view the process of encryption.
Priority: High.

R4 Decrypt Option
The simulator shall allow the decryption process to be simulated after selecting Decrypt from the Main Menu.
Rationale: To allow the user to view the process of decryption.
Priority: High.

R5 Exit Option
This option shall allow the simulator to close once after selecting Exit from the Main Menu.
Rationale: To allow the user to quit the simulator.
Priority: High.

Encryption

R6 Encryption
The simulator shall scramble plain text into cipher text.
Rationale: This function will encrypt the plain text provided by user input into cipher text.
Priority: High.

R7 One Rotor - Encryption
The simulator shall demonstrate the operation of encryption in one rotor.
Rationale: This will allow the user to see graphically the process of encryption in one rotor. This shall include the demonstration of current flowing in the rotor from the initially selected character from the keyboard to the cipher character on the switch board.
Priority: High.

R8 Three Rotors - Encryption
The simulator may demonstrate the operation of encryption in three rotors.
Rationale: This will allow the user to see graphically the process of encryption in three rotors. This shall include the demonstration of current flowing in the rotors from the initially selected character from the keyboard to the cipher character on the switch board.
Priority: Medium.

Decryption

R9 Decryption
The simulator shall unscramble cipher text into plain text.
Rationale: This function will decrypt the cipher into the original plain text.
Priority: High.

R10 One Rotor - Decryption
The simulator shall demonstrate the operation of decryption in one rotor.
Rationale: This will allow the user to see graphically the process of decryption in one rotor. This shall include the demonstration of current flowing in the rotor from the initially selected character from the switch board to the character on the keyboard.
Priority: High.

R11 Three Rotors - Decryption
The simulator may demonstrate the operation of decryption in three rotors.
Rationale: This will allow the user to see graphically the process of decryption in three rotors. This shall include the demonstration of current flowing in the rotors from the initially selected character from the switch board to the character on the keyboard.
Priority: Medium.

Miscellaneous

R12 Visual Representation

The simulator shall visually demonstrate the principle of poly-alphabetic substitution in operation in the scrambling unit of an enigma machine.
Rationale: This will give a clear understanding on the operation of poly-alphabetic substitution.
Priority: High.

R13 Animation

The simulation shall be demonstrated using animation.
Rationale: As well as being graphically presented the simulation will also demonstrate the various operations of encryption using animations.
Priority: High.

R14 Attack Method

The simulator may include an attack method, which could become a game, where the user would guess the encrypted plain text.
Rationale: A sort of game could almost be created using this simulator in which the user of the system could attempt to decrypt the cipher text manually.
Priority: Low.

NON-FUNCTIONAL REQUIREMENTS

R15 Graphical User Interface

The interface shall be presented in a graphical format.
Rationale: The simulator must be graphically presented to the user rather than text based as this is much more interactive and easier to use.
Priority: High.

R16 Operating System - Windows

The simulator shall be compatible on Windows Operating Systems.
Rationale: Since majority of the users would be running a Windows operating System, the system must be compatible and fully function on Windows.
Priority: High.

R17 Operating System – Mac/Linux

The simulator may be compatible on Mac/Linux Operating Systems.
Rationale: A minority of the users may use a mac or Linux operating system therefore it would be ideal to cater for their needs thus the system may be compatible with these operating systems.
Priority: Medium.

R18 Development

The simulator should be developed using C++ and the Visual Studio IDE.
Rationale: In order to promptly start development, a familiar programming language and environment should be used.
Priority: High.