INDEX

1. Introduction

1.1 Purpose

This document provides a detailed description of our software viz. ImaGe. It enumerates and explains the different requirements of the software system.
It also briefs the technical aspects of the software such as the purpose and features of the system, the interfaces of the system , what the system will do, etc.

1.2 System Overview

This program runs on both Windows and LINUX interface. The system must have a compiler preferably CODE::BLOCKS with simplecpp package installed.

1.3 References

- http://www.google.co.in/
- http://en.wikipedia.org
- http://cse.iitb.ac.in/~cs101/
- http://stackoverflow.com/
- Introduction to Problem Solving and Programming through C++ by Abhiram Ranade

2. Overall Description

2.1 Product Perspective

This software is independent and totally self-contained.

2.2 Product Functions

2.2.1 The main aim of **ImaGe** is to generate the image of an object (point or extended) in various optical instruments prompted by the user, in this case a mirror and a lens(convex or concave) .

2.2.2 The user also has to provide the required inputs like radii of curvature and the refractive indices of the different mediums

2.2.3 It will help in generating the image of the object by constructing the required ray diagrams. The user can input the object by clicking at a point in the window (in case of a point object), or by freely drawing it (in case of an extended object).

3. Details

3.1 Functionality

3.1.1 This software is constructed in the open source IDE platform Code::Blocks.

3.1.2 The first step requires the user to input the type of optical instrument to be chosen.

3.1.3 After this, the user has to input the radii of curvature and the refractive indices of the medium surrounding the instrument.

3.1.4 Upon entering the values the user has to choose the characteristics of the object,i.e,whether the object is real or virtual;and whether it is a point object or an extended one.

3.1.5 Upon entering the values, a diagram showing the principal axis, the optical instrument chosen and the corresponding foci is shown.

3.1.6 When the user clicks on the canvas the characteristics of the image and the ray diagrams are shown.

3.2 Supportability

This software is basically supported on any interface whether it is MS Windows or LINUX Ubuntu. The software should be run on minimum of 500 MHz, 256 MB RAM machine. For speed processing the preferable configuration is 1.2 GHz, 2 GB RAM machine.

3.3 Design Constraints

3.3.1 When the user is asked to input an extended object, he/she has to merely click on a number of points on the canvas. The extended object thus formed will be the collection of lines joining each consecutive point. This restricts the user's freedom to draw freely.

3.3.2 Since the software will use the clicking at a point in the workspace window by the user for the input of the object, it will not be able to take specific co-ordinates of the object w.r.t. some origin, generally the optical centre in the optics physics.

3.3.3 It is very much possible that the image generated cannot be displayed in the Canvas window. In such cases, the corresponding image distance and the image characteristics are displayed.

3.4 On-line User Documentation and Help System Requirements

A Help template, or User Manual regarding the issue on how to use the software, the basic software requirements, and the constraints and assumptions made during the development of the project will be displayed to the user on request.

3.5 Interfaces

3.5.1 User Interface

The software opens a new window designed under Microsoft Visual Basic 6.0 which serves as a mode of interaction between the user and the system.

3.5.2 Hardware Interface

The software requires only the basic hardware requirements which are monitor, mouse, keyboard and CPU as a processing unit. It do not require any other additional hardware for the processing of the software.

3.5.3 Software Interface

3.5.3.1 The written C++ code shall have a common connecting door with the MS Visual Basic 6.0 so that the code can be processed for the display of the various functions and also take the inputs which are then operated to finally showcase the image and the specifications of the image.

3.5.4 Communication Interface

3.5.4.1 The communication between the MS Visual Basic and the compiler is done using the .dll file.

3.6 Basic Assumptions

3.6.1 The user should be aware on how to deal with a computer with mere working knowledge.

3.6.2 The user should understand English.

3.6.3 The machine on which the work is being performed should meet the specified hardware and software requirements to run the system software.

3.7 Future Scope

3.7.1 The software can be enhanced by including the feature of having the inputs of the object using the co-ordinates of the object.

3.7.2 The software can be further modified by allowing the user to freely draw an extended object.

3.7.3 The software can be made more user-friendly by granting the user access to his previous projects, so as to modify them or to keep track of his/her progress.

4. Quality Control

4.1 Test Data

There are two types of input test data. First, when the object is a point object and second, when the object is an extended object. The project is expected to run smoothly under both the cases provided that the user does not exceed some input standards.

5. Risk Management

5.1 Working with MS Visual Basic:

Our project focuses on using MS Visual Basic 6.0 acting as an input interface but the hard part is none of our team member knows how to implement a code written in C++ in MS Visual Basic.

Mitigation Plan: The software will not further use the MS Visual Basic as an input interface. We may give a hand on Graphical User Interface (GUI) implemented using gtkmm which is an interface of GTK+ in C++.