VTK based primitive CAD modeller



Centre for Computational Technologies

Transforming human life by democratization of technology

[https://www.cctech.co.in](https://www.cctech.co.in/)

* Introduction

This document outlines the plan for the "VTK based primitive CAD modeller" detailing its purpose, scope, system overview, functional requirements, tools, milestones, timeline, and conclusion.

* Purpose

The purpose of the VTK based primitive CAD modeller is to develop a user-friendly software tool that allows users to view and model primitives (drag and drop primitives) With c#.

* Scope

A VTK-based 3D viewer in C# allows users to visualize, interact with, and manipulate 3D objects like cubes, spheres, and other primitives through drag-and-drop actions. This approach is ideal for building interactive applications for scientific visualization, educational tools, and design software, combining the flexibility of VTK with the user-friendly C# environment..

* System Overview
* The 3D Viewer and Modeling Application will feature a graphical user interface (GUI) built in C# with the Windows Presentation Foundation (WPF) framework, providing a user-friendly environment for interaction. VTK will be used to render high-quality 3D graphics, allowing users to view, drag, drop, and manipulate geometric primitives in real- time.

3 Functional Requirements

**VTK based primitive CAD modeller:**

**1) 3D Viewer and Scene Manipulation:**

The system must be able to load, render, and display 3D primitives like cubes, spheres, cylinders, etc.

Users must be able to pan, zoom, and rotate the 3D scene with mouse or keyboard interactions.

**2) Drag-and-Drop Functionality:**

Users must be able to drag and drop primitives into the 3D scene from a UI component, such as a toolbox.

The system should allow users to reposition objects within the 3D scene by dragging them.

**3)File Operations:**

The system must allow users to save the current scene to a file, preserving object positions and properties.

Users should be able to load saved scenes to continue work or to share with others.

4 Tools

* Integrated Development Environment (IDE)(Visual Studio)
* Graphics and Visualization Toolkit(VTK (The Visualization Toolkit))
* VTK Wrapper for .NET(ActiViz.NET)
* Graphical User Interface Framework(WPF (Windows Presentation Foundation))

5 Milestones and Timeline

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Milestones** | **Date and Time** |
| 1. | Project Problem Statement | 06 May 2024  05:00 PM IST |
| 2. | SRS Preparation and Initial Presentation | 07 May 2024  12:00 PM IST |
| 3. | SRS Approval | 07 Mar 2024  02:00 PM IST |
| 4. | Setup for Project | 07 May 2024  05:00 PM IST |
| 5. | GUI Implementation | 08 May 2024  07:00 PM IST |
| 6. | VTK based primitive CAD modeller Development | 09 May 2024  07:00 PM IST |
| 7. | Real-Time Rendering Integration | 10 May 2024  07:00 PM IST |
| 8. | Testing and Debugging | 11 May 2024  07:00 PM IST |
| 9. | Finalization and Presentation | 13 May 2024  01:00 PM IST |

6 Conclusion

A VTK-based 3D viewer project in C# allows users to visualize, drag, drop, and manipulate 3D primitives with a user-friendly interface. By combining the robust rendering capabilities of VTK with the flexibility of C#, the project demonstrates a practical and interactive approach to 3D graphics, ideal for a range of applications, from design to scientific visualization.

7 User Interface



**1) Main Workspace:**

This is the primary area where users draw, edit, and view shapes. It should be a large, open canvas with support for mouse interactions such as clicking and dragging.

**2) Toolbar:**

A Vertical toolbar located at the top right side of the workspace, providing quick access to shape creation tools and other functionalities.

**3) Shape Tools:**

**Circle:** Button/icon to draw a circle on the workspace.

**Line:** Button/icon to draw a straight line between two points.

**Point**: Button/icon to create a single point in the workspace.

**Ellipse**: Button/icon to draw an ellipse.

**Rectangle:** Button/icon to draw a rectangle with two opposite corners as reference points.

**4) Actions:**

**Save**: Button to save the current design to a file. Prompts the user to choose a file format and location.

**Clear:** Button to clear the workspace, removing all shapes.