A

PROJECT PROPOSAL

ON

3D Modelling and Rendering of Dharahara

Submitted To:

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Submitted By:

Sandeep Acharya (075BCT074)

Sangam Chaulagain (075BCT078)

Saujan Tiwari (075BCT083)

June 30 2021

Table of Contents

Introduction

Computer graphics is a field of computer science that studies methods for creating and manipulating visual content on a display device. Computer graphics uses different mathematical concepts and algorithms to display the images on the screen. Many hardware devices algorithms have been developed for improving the speed of picture generation with the passing of time. It includes the creation, storage of models and images of objects. These models are used in various fields like engineering, mathematics and so on.

Objectives:

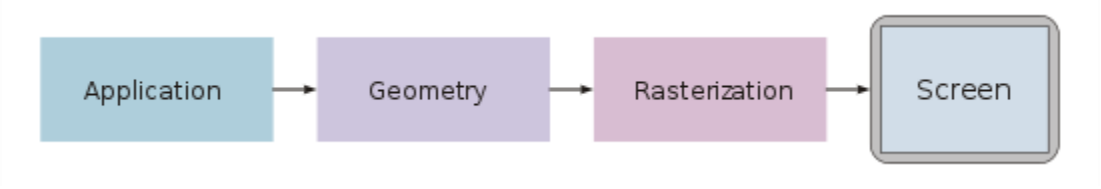
1. To learn principles of 3D transformations and rendering,
2. To understand Raster method and Graphics Pipelining ,
3. To get familiar with OpenGL API,
4. To learn Blender 3D for designing 3D Models,
5. To learn to work in a team through version control.

Theory

Graphics Pipeline:

Graphics pipeline is a conceptual model that describes the steps needed to perform to render a 3D scene to a 2D screen. Once the 3D model has been created, the graphics pipeline is the process of turning that 3D model into what the computer displays on the screen.

A graphics pipeline can be divided into three main parts: Application, Geometry and Rasterization.

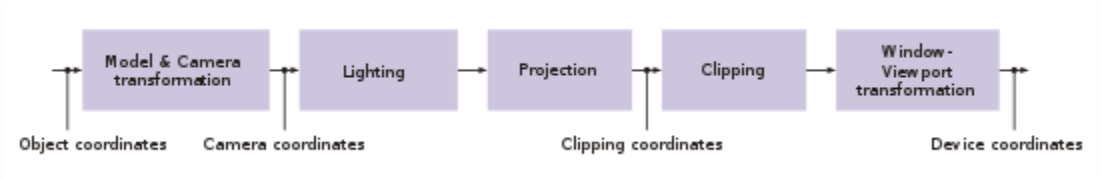


Application:

In the application step, changes are made to the scene as required, by the means of input devices or during an animation. The new scene with all its primitives, usually triangles, lines and points is then passed to the next step on the pipeline. The tasks such as collision detection, animation are done in the application step.

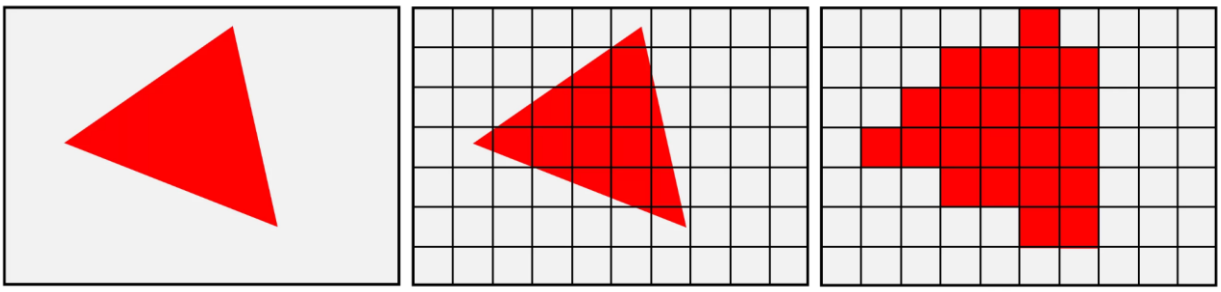
Geometry:

The geometry step (with Geometry pipeline), which is responsible for the majority of the operations with polygons and their vertices (with Vertex pipeline), can be divided into the following five tasks. It depends on the particular implementation of how these tasks are organized as actual parallel pipeline steps.



Rasterization:

Rasterization is the task of taking an image described in a vector graphics format and converting into a raster image which is a series of pixels, dots or lines, which when displayed together, create the image represented by the vector graphics format. The rasterized image then can be displayed on a computer display.



Methodology:

The programming language that we will be using for this project is C++. OpenGL, which is a cross-language, cross-platform API will be used to render our model. Blender 3D will be used to create the model. Similarly, GLFW will be used for windows and event management. Some other libraries like assimp, glm etc will be used for various operations like loading the model and performing various mathematical operations.

Operations

The overall project will involve following operations in sequence

1. **Model Creation with Blender**

   We will use blender to create a 3D model for our project. Blender allows us to resize, scale and perform various operations on the model. Then the model created in blender will be exported in obj file format so that it can be imported into OpenGL modules.

2. **Importing the Model**

   The obj file exported from blender will then be imported into OpenGL by using assimp library, which loads the model's vertices and faces into different data structures which then will be used for performing various operations.

3. **Rendering Model**

   After extracting the data of the model from assimp, different OpenGL functions will be used to render the model through different stages of graphics pipelining. Various shading and lighting algorithms will be implemented to illuminate as well as depict the image into a computer window in 3D space.

Features

The project will have the following features:

1. **3D View**

   The project will provide the 3d view of the model from different camera positions. Users will be able to rotate and move the camera based on the real time keyboard and mouse inputs.

2. **Illumination Model**

   Illumination models will be used on the project to compute the color of the object positions within the scene according to the position of the light, normal vector and camera. It will make the scene more realistic

3. **Texture Mapping**

   Real texture input will be mapped onto the model surface to make it photorealistic

Object to be Modeled:

A 3D model of Dharahara will be rendered on the window with appropriate coloring, texturing, illumination and camera view. The application will feature a real world view of the model with a camera which can rotate and move. Appropriate visible surface detection algorithms will be applied to provide realistic scene view from different viewing positions.

Fig: Initial Model of Dharahara created in Blender and a real world image of Dharahara

References:

Learn Opengl: *https://learnopengl.com/*

Computer Graphics with OpenGL, Donald Hearn & Pauline Baker

Graphics Pipeline: *https://en.wikipedia.org/wiki/Graphics_pipeline*

Rasterization: *https://www.techspot.com/article/1888-how-to-3d-rendering-rasterization-ray-tracing* ,

https://en.wikipedia.org/wiki/Rasterisation