



Department of Computer Science & Engineering

PBL Synopsis of COMPUTER GRAPHICS Mini Project - 2017-2018 (6th Semester)

Batch No: CSE-A1	Guide Name: Mr. Shankar R	Submission Date: 18 th May 2018
Tower of Hanoi Simulation		
Sl No	USN	Name
1	1BY15CS002	Aayush Nandkeolyar
2	1BY15CS010	Ankit Agrawal

Signature of Guide

Abstract

The Tower of Hanoi simulation is an interactive way for the user to understand how the tower of Hanoi algorithm works. It consists of three rods and three disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest on the top thus making a conical shape. The objective of the simulation is to simulate the movement of the entire stack from one rod to another obeying all the rules.

Introduction

The Tower of Hanoi (also called the Tower of Brahma or Lucas' Tower and sometimes pluralized) is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod.
3. No disk may be placed on top of a smaller disk.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks.

A simple solution for the toy puzzle is to alternate moves between the smallest piece and a non-smallest piece. When moving the smallest piece, always move it to the next position in the same direction (to the right if the starting number of pieces is even, to the left if the starting number of pieces is odd). If there is no tower position in the chosen direction, move the piece to the opposite end, but then continue to move in the correct direction.

For example, if you started with three pieces, you would move the smallest piece to the opposite end, then continue in the left direction after that. When the turn is to move the non-smallest piece, there is only one legal move. Doing this will complete the puzzle in the fewest moves.

For an even number of disks:

- make the legal move between pegs A and B (in either direction),
- make the legal move between pegs A and C (in either direction),
- make the legal move between pegs B and C (in either direction),
- repeat until complete.

For an odd number of disks:

- make the legal move between pegs A and C (in either direction),
- make the legal move between pegs A and B (in either direction),
- make the legal move between pegs B and C (in either direction),
- repeat until complete.

In each case, a total of $2^n - 1$ moves are made.

The above algorithm is simulated using OpenGL where we generate 3 different coloured disks. In this project we show how the disks move from the first tower to the 3rd tower without placing the larger disks over the smaller disks.

System Requirement Specifications (Functional & Non-Functional)

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system. The requirements are specified as below:

Functional requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The applications of computer graphics in some of the major areas are as follows:

- Display of Information
- Design
- Simulation and Animation
- User Interface

Non-functional requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-Functional requirements often apply to the system as a whole.

Non-Functional requirements are as follows:

- Dependability:

The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not fail in normal use

- **Availability:**

The ability of the system to deliver services when requested. There is no error in the program while executing.

- **Reliability:**

The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

- **Safety:**

The ability of the system to operate without catastrophic failure. This program is user friendly and it will never effects on the system.

- **Security:**

The ability of the system to protect itself against accidental or deliberate intrusion

Proposed Methodology

The desktop based application can run on the Windows operating system

- The simulation is completely based on the OpenGL platform.
- First, we make a Tower of Hanoi design using the OpenGL commands.
- The simulation starts on its own simulating the Tower of Hanoi algorithm.
- Pressing the r key restarts the simulation.
- The arrow keys can be used to perform rotation.

References

Reference Books:

- Edward Angle, “Computer Graphics and Visualization Using OpenGL”, 2006.
- Neider, Jackie, Davis, Tom “OpenGL Programming Guide”, 1996.
- Yeshwant Kanetkar, ”Computer Graphics under C”, 2004
- OpenGL SuperBible 7th Edition
- OpenGL ES 3.0 Programming Guide (2nd Edition)

Online Resources:

- <http://www.opengl.org>
- <http://learnopengl.com>
- <http://www.nehe.gamedev.net>
- <http://www.opengl-tutorial.org/>
- <http://www.glprogramming.com>