Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Mini-Project Report on

"Tower of Hanoi: A Complete Recursive Visualization"

Computer Graphics

COMP 342

Submitted by

Suyog Dhungana (16)

Puspa Hari Ghimire (19)

Submitted to

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submission Date

2nd February, 2025

Acknowledgement

We express our sincere gratitude to Mr. Dhiraj Shrestha, our course instructor, whose teachings on Computer Graphics provided the foundational knowledge necessary for the development of this mini project.

Special thanks go to our peers and friends who provided us with constructive feedback and suggestions during the development phases of this project. Their inputs were invaluable in refining the features and functionality of the platform.

Abstract

This project implements a graphical simulation of the Tower of Hanoi puzzle using OpenGL. The program visually demonstrates the recursive algorithm that solves the puzzle, with users able to customize the number of disks. Interactive features include step-by-step movement of disks and an automatic solving option, allowing users to explore the solution process. The simulation enhances understanding of recursion and provides an engaging educational experience. Future improvements could involve better user interface design, smoother animations, and additional control features for a more interactive experience.

Keywords: *Tower of hanoi, Recursion, Simulation*

Table of Contents

Introduction

This project implements a graphical simulation of the Tower of Hanoi problem. The program utilizes OpenGL (via GLUT) to visually represent the movement of disks between three pegs, following the recursive algorithm to solve the puzzle.

Background

The Tower of Hanoi is a classic mathematical puzzle involving three pegs and a set of disks of different sizes. The objective is to move all the disks from the source peg to the destination peg, following these rules:

- Only one disk can be moved at a time.
- A disk can only be placed on an empty peg or on a larger disk.
- The source peg must be emptied by transferring all disks to the destination peg.

This project uses computer graphics to visualize the recursive solution in an interactive way.

Objectives

- Provide a clear visualization of the step-by-step solution for the Tower of Hanoi puzzle.
- Enhance understanding of computer graphics concepts such as transformations, rendering, and animation.

System Design

The system consists of the following key components:

- **Algorithm Implementation:**

  The algorithm is the heart of the Tower of Hanoi system. It will determine the steps to solve the puzzle based on the number of disks and the current state of the rods.
- **Graphics Design:**

  The graphical component displays simple, color-coded rods and disks with smooth animations, making the puzzle easy to understand and visually appealing.
- **Animation:**

  This component is responsible for visually showing the real-time movement of disks during the solving process.
- **User Interaction:**

  This component allows users to interact with the system by selecting a number of disks, starting the simulation and resetting the simulation.

Programming Language and Libraries Used

For the implementation of the Tower of Hanoi simulation, the project will rely on the following libraries and programming language:

Programming Language Used: C++

Libraries Used:

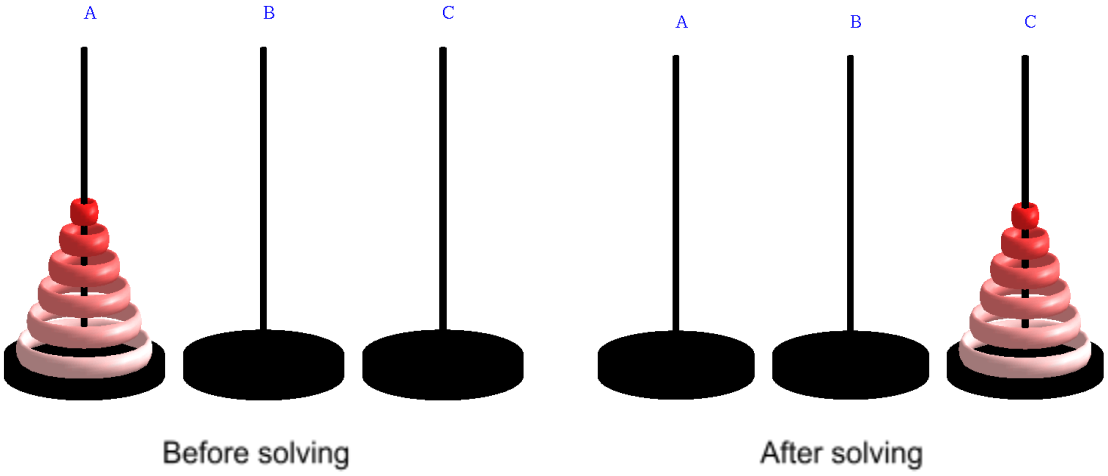
- **windows.h:** Provides Windows API functions.
- **gl/glut.h:** Used for OpenGL rendering and window management.
- **math.h:** Provides mathematical functions for calculations.
- **sstream:** Used for text stream manipulations.

Implementation

The Tower of Hanoi algorithm

Tower of Hanoi is a mathematical puzzle where we have three rods (A, B, and C) and N disks. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod A. The objective of the puzzle is to move the entire stack to another rod (here considered C), 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 i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.



Recursion Algorithm:

```
Procedure Hanoi(disk, source, dest, aux)
   IF disk == 1, THEN
      move disk from source to dest
   ELSE
      Hanoi(disk - 1, source, aux, dest)
      move disk from source to dest
      Hanoi(disk - 1, aux, dest, source)
   END IF
```

OpenGL Rendering Functions

This project uses several OpenGL rendering functions to create and display the pegs, disks, and movements. The following are the key functions for rendering the visuals:

- `drawcylinder()`:

  This function is responsible for drawing the pegs in the Tower of Hanoi puzzle. It uses OpenGL's `gluCylinder` function to create a long solid cylinder which represents each peg(pole) in the visualization. The `GLUquadricObj` is used to define the cylinder's properties.

- `drawBase():`

  This function draws the base of the pegs, using two disks. The `gluCylinder` function is used to create the cylindrical base, while `gluDisk` is used to add the top and bottom faces of the base.

- `drawPegs():`

  This function draws all three pegs in the puzzle. Each peg is drawn using `drawcylinder()` to create the peg itself and drawBase() to add the base. The `glTranslatef()` function is used to position the pegs on the screen, while `glRotatef()` ensures the correct orientation for the pegs. The pegs are arranged along the X-axis at fixed positions, and their 3D transformations ensure that they are displayed correctly in the scene.

- `drawText():`

  This function handles displaying textual information on the screen, such as the current move number, disk details, and source/destination peg. It uses the `glutBitmapCharacter` function to render each character of the string at specified positions. This function allows users to follow the progress of the puzzle-solving process by showing move details like "Move: X," "Disk Y," and "from A to B."

- `drawSolved():`

  After the puzzle is solved, this function displays a "Solved!!" message, indicating the end

of the game. It uses `drawText()` to show the appropriate message on the screen and is triggered when the number of moves reaches the maximum.

- `lighting()`:
  This function sets a white light and makes objects shiny with white color and highlights in OpenGL. `GL_COLOR_MATERIAL` is used to apply the light and material settings to create realistic shading on the disks.

- `display()`:
  The display() function is the main rendering function that updates the visualization. It is responsible for clearing the screen, applying camera transformations using `gluLookAt()`, drawing the pegs, and rendering the disks. The disks are represented as toruses (donut shape), and their positions on the pegs are updated based on the current state of the puzzle. This function is called in every frame to refresh the visualization.

  These functions work together to create the visual representation of the Tower of Hanoi, allowing users to follow the progress of the puzzle and see how the disks move between the pegs.

User Interactions

The users can interact with the system for providing inputs using a right-click menu. The right-click menu provides with the options to perform the following actions:

- **Number of Disks:** Menu option to choose the number of disks for simulation. This sets the input number of disks on the pegs.
- **Solve:** Menu option to start the simulation for solving the Tower of Hanoi problem. This option is responsible for initiating the algorithm
- **Restart:** Menu option to reset the simulation. This option is used to reset the disks back to the starting peg.
- **Exit:** Menu option to end the simulation.

Discussion

This Tower of Hanoi visualization offers a user-friendly and engaging approach to understanding the recursive algorithm behind the puzzle. The graphical representation using OpenGL effectively demonstrates the disk movement, making the concept of recursion more accessible and visually clear.

Limitations

The limitations of this project are:

- Fixed maximum number of disks
- No pause/resume feature for animation
- No speed control for animation

Future Enhancements

These limitations can be addressed in the future as well as following improvements can be added:

- Optimizations for handling larger number of disks
- Improved and more interactive user interface
- Refined disk movement animations

Conclusion

The OpenGL-based Tower of Hanoi simulation is successfully implemented, providing an interactive and visually engaging experience that combines theoretical knowledge with practical computer graphics. With features like a customizable disk count, it serves as an effective educational tool, illustrating the recursive algorithm through clear visualization. Although the project does not allow manual control of the process, it provides step-by-step and automatic solutions, demonstrating the power of graphical representations.

References

GeeksforGeeks. (2024, May 9). *Program for tower of Hanoi algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/

Ahn, S. H. (n.d.). *OpenGL Projection Matrix*. https://www.songho.ca/opengl/gl_projectionmatrix.html

Tower of Hanoi using recursion. (n.d.). https://www.tutorialspoint.com/data_structures_algorithms/tower_of_hanoi.htm

Khronos OpenGL® and OpenGL® ES Reference pages - The Khronos Group Inc. (n.d.). https://registry.khronos.org/OpenGL-Refpages/

Appendix

Source Code: https://github.com/puspah-ghimire/Tower-of-Hanoi