Jay Ashish Kadam

Toon Shading Using OpenGL

Project Goals

The project aims to implement a toon shading (cel shading) technique with 4-channel stylized highlighting and edge detection, aiming to create visually appealing and illustrative rendering effects for 3D objects such as spheres, torus, and stylized pineapple which are rotating along their axis.

Project Description

1. Brief Overview of Tasks:

- Developed a 3D rendering pipeline utilizing OpenGL and GLSL shaders.
- Implemented toon shading with quantized lighting intensities and stylized highlights.
- Added edge detection by rendering wireframe outlines on objects.
- Created geometries for a sphere, a torus, and a stylized pineapple with cones for leaves.
- Enabled dynamic rotation and interactive camera movement with keyboard controls.
- Incorporated Fullscreen toggle and maintained aspect ratio on window resizing.

2. Tools/API/IDE:

- **Programming Language:** C++
- **Graphics Library:** OpenGL with GLEW for extensions and GLUT for windowing and events.
- **Math Library:** GLM for matrix and vector operations.
- **Development Environment:** Visual Studio 2022

3. Workflow:

- **Initialization:** Set up shaders, compiled vertex and fragment shaders for cel shading and outline rendering.
- **Geometry Creation:** Used mathematical algorithms to generate sphere, torus, and pineapple geometries.

- **Rendering Pipeline:** Applied transformations, configured camera settings, and implemented custom lighting calculations in shaders.
- **Interaction:** Added keyboard-based camera controls and window management features.

4. Technical/Algorithmic Challenges Faced:

- **Quantized Lighting:** Ensuring smooth yet discrete transitions in light intensity for the toon shader.
- **Complex Geometry Generation:** Efficiently constructing and rendering objects like the torus and pineapple with consistent normal and indices.
- **Interaction & Usability:** Implementing real-time interaction for camera and rendering adjustments without performance degradation.

5. Strategies Used to Meet the Challenges:

- Used GLSL if conditions to implement light quantization in the fragment shader.
- Scaled objects slightly for outline rendering to prevent depth conflicts.
- Precomputed vertex positions, normal, and indices for complex geometries.
- Leveraged timer functions for smooth rotation and interaction responsiveness.

6. Screen-Grabs for Explanation:

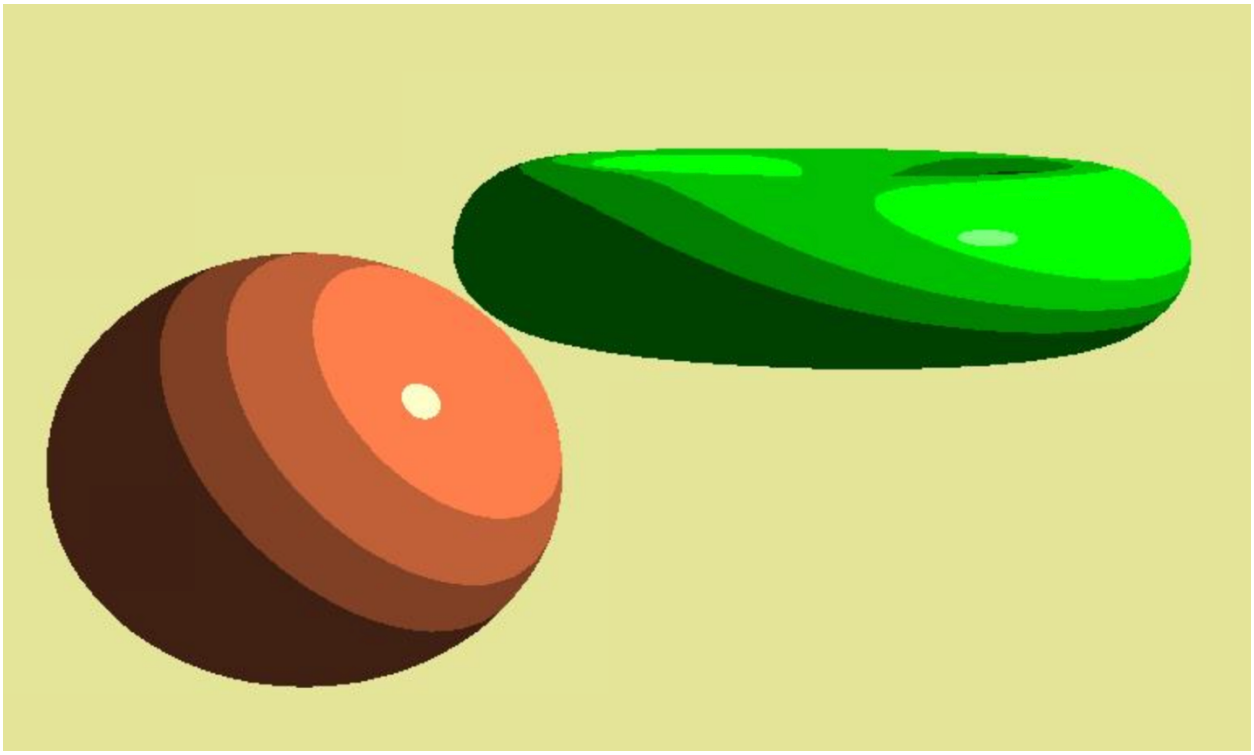

Fig.1(A)- cel-shaded sphere & Torus with discrete lighting steps and specular highlights

Fig-1(B)-A stylized pineapple with a detailed body and cone-based leaves.

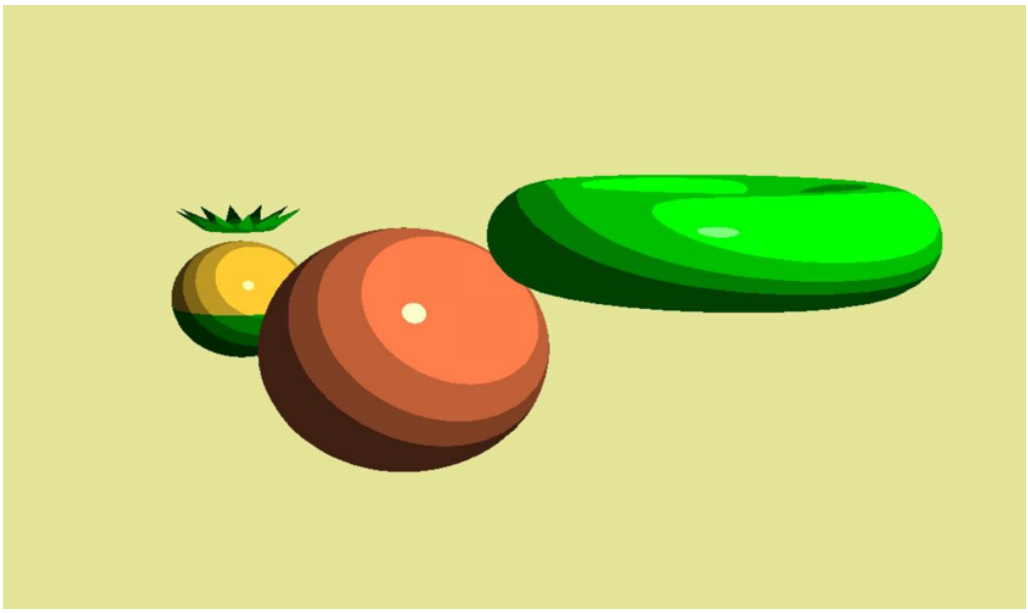

Fig-1(c)-Camera control and rotation effects captured at various angles

Experiments, Results, and (Technical) Learnings

1. Experiments:

- Experimented with different light positions and intensities to analyze their effect on cel shading.
- Tested various numbers of quantization levels for lighting to achieve optimal visual appeal.
- Compared performance for dynamically generated geometries versus static buffers.

2. Results:

- Achieved a smooth, visually distinct cel-shaded appearance with discrete lighting transitions.
- Successfully combined multiple shading techniques, including specular highlights and outline rendering.
- The framework performed well with real-time interactivity, maintaining a high frame rate of 60 FPS.

3. Learnings:

- GLSL provides powerful tools for stylized rendering but requires careful optimization for real-time applications.
- Dynamic geometry generation requires a balance between accuracy and computational efficiency.
- Edge detection and outline rendering are visually effective for enhancing object definition in stylized renders.

Conclusion

This project successfully implemented a toon shading system with stylized highlighting and edge detection using OpenGL and GLSL. The shading and geometry generation techniques enabled the creation of visually distinct and dynamic 3D objects. Key technical challenges such as lighting quantization, real-time rendering, and user interaction were effectively addressed. This project paves the way for further experimentation in non-photorealistic rendering