University of St Andrews

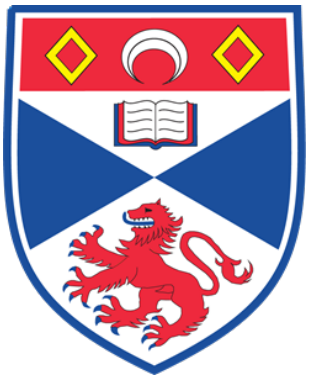
Computer Graphics

CS4102

Ring-Based Distributed System

Author:
150008022

May 14, 2020



Goal

The aim of this practical is to understand the key principles behind various techniques frequently used for the rendering of 3D objects, and to get hands-on experience with their implementation and manipulation.

1 Running the Program

The project uses `maven` for lifecycle and dependency management. `LWJGL` was used for this project, which provides a way to use `OpenGL` from Java. OpenGL *may* be a cause of compatibility issues when running, but an early and forward compatible profile was used to minimise the risk of this.

Usage
`mvn clean install` will compile the project, run all tests, and produce an exectuable jar.
`cd target` to enter the directory where the jar has been compiled.
`java -jar FaceModelling-jar-with-dependencies.jar` to run the program. Pass the `-h` flag to see options.

2 Implementation

2.1 Loading Mesh Data

2.2 Depth Testing

OpenGL does not natively implement painters algorithm for depth testing, and does not expose an API for carrying it out . Instead, it uses Z-buffers, Z-Buffers involve recording the z-index last drawn at each pixel of the window view. A pixel with a greater z-index than the current value will be drawn and the z-buffer updated to reflect the new value. Otherwise the depth test fails and the object will not be drawn at that coordinate. This functionality is enabled using the `glEnable(GL11.GL_DEPTH_TEST)` call. An optimisation called early z-testing is often used which is applied earlier on in the rendering pipeline in order to reduce the number of fragments that need to be processed in total.

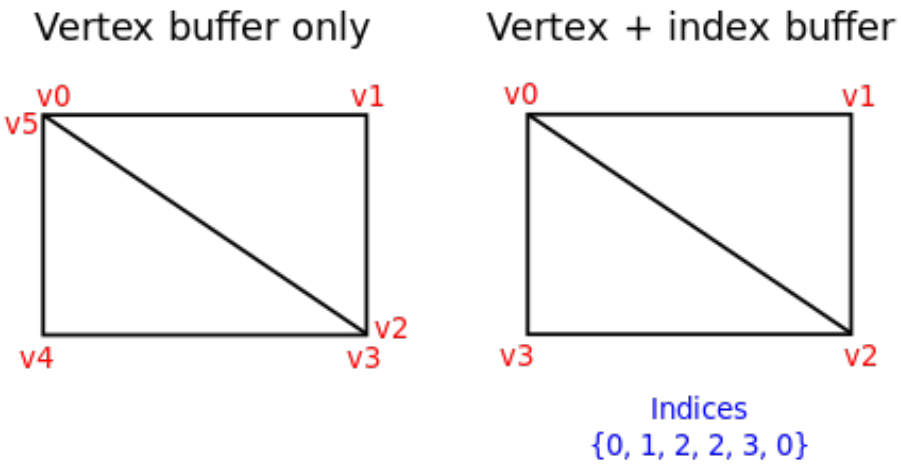


Figure 1: How indices are used when describing meshes.

Painters algorithm and Z-buffers are both viable solutions to the same problem; the former is more computationally expensive while the latter requires more memory. Z-Buffers are preferred in most cases for modern hardware since memory has become cheaper. Painters algorithm also requires additional computation in the case where objects intersect. Methods to solve this involve splitting the intersecting shapes and rendering these instead.

A separate renderer was implemented that used painters algorithm. This was achieved by