3D Wireframe Renderer

Game Engines E2013

Jakob Melnyk, jmel@itu.dk

October 22, 2013

1 Introduction

the project

constraints

source of math /pseudo-code [1]

grooss

2 Features

What does it do (show wireframe from hardcode, camera position/lookpoint moveable)

2.1 Camera controls

Movement of the camera position and lookpoint is done in 10-unit intervals. This interval is hardcoded and should probably be a changeable variable.

The controls for the camera are:

- a/s/q decrease the value of the x/y/z position of the camera, respectively.
- d/w/e increase the value of the x/y/z position of the camera, respectively.
- j/k/u decrease the value of the x/y/z position of the camera's lookpoint, respectively.
- l/i/o increase the value of the x/y/z position of the camera's lookpoint, respectively.

3 Overview of the Program

3.1 Objects

Vector

Matrix

Vertex

Triangle

Camera

Loader

[1] https://blog.itu.dk/MGAE-E2013/files/2013/09/transforms.pdf

3.2 Renderer

how does renderer draw (when)

drawing process (calculate camera transforms, see 3.3.1) (foreach triangle : update triangle according to camera, draw triangle based on view points) (only draws lines, not points)

3.3 Math

vertices are given the camera and calculate their own screen points (instead of renderer doing so) [point out problem with calculating points that are offscreen]

no frustum culling

3.3.1 Single calculation

One calculation of camera transforms per draw

4 Issues with the renderer

There are a few issues with my wireframe renderer implementation: screenpoint calculations, lookpoint not following the camera movement and crashing at certain camera/lookpoint positions.

4.1 Screenpoint calculations

4.2 Camera/lookpoint crash

5 Conclusion

6 Future Work

6.1 Load triangles/vertices from a file

6.2 Test performance

6.3 Frustum culling

6.4 Better camera controls

6.4.1 Change camera variables

7 Appendix

7.1 Screenshots