

UTRC VR Test

Adeola Fabola

20/09/2017

Task 1: OBJ file manipulation

Tom wears a VR helmet experienced in an immersive environment, he stands in front of a sphere model and move his hand to change the sphere model to a square model. Design and develop a program to move the vertexes of sphere to generate a square model. The total length of the movement of vertexes should be as short as possible.

Tips:

1. Upload the obj file to the sketchfab, to check and analyze the sphere model. When program automatically generates new model, you can also upload to the sketchfab to check the model.
2. Modify the original obj file, develop a program to generate a new obj file.
3. The programming language is unlimited (prefer Java).

Solution

- Java is used to implement the solution. The code file is called UTRCTest.java
- An object-oriented approach is adopted. The java file contains a Geometry class, a Vertex class and a Face class (in addition to the public UTRCTest class).
- Algorithm is as follows:
 - Read and parse .obj file on disk (place this file "Test.obj" in the same folder as the java program)
 - if line starts with v, followed by three, white-space-separated values, read values as point coordinates to denote vertices
 - if line starts with f, followed by three, white-space-separated values, read values as vertex indices to denote faces
 - compute the maximum and minimum bounds of the sphere on the x, y and z axes by computing the largest and smallest values of the vertices
 - iterate through vertices; for each sphere vertex
 - compute the cube face it belongs to by calculating the absolute largest of the x, y, z values

- move the largest axis value to the closer of the maximum or minimum bound on the respective axis
 - compute the other two vertex coordinates using the formula for mapping a sphere to a cube. This was obtained from [here](#)
 - move the original sphere vertex to the newly-computed cube vertex
- create new file on disk with .obj extension
- add default smoothing group information for untextured polygonal mesh (as in the original input model)
- write out new vertices on each line, starting with v followed by three numbers separated by a space
- write out the faces on each line, starting with f followed by three numbers separated by a space
- The output model “output.obj” is generated in the same folder as the java program. The model can be uploaded so sketchfab, or imported into a 3D renderer such as meshlab, blender, etc to verify the cube shape.
- Java project folder is called “UTRC Test”
- Netbeans version used for development: Netbeans IDE 8.1

Task 2: Unity3D Programing

1. Use several cameras to render a same scene.
2. The scene should include 3 different models, square, sphere and cylinder
3. The 3 models move on a certain trajectory which should emerge mutual occlusion and the square should always display in front of others.

Solution

- A Unity3D scene called “mutual_occlusion” has been developed, which contains a cube (red), a sphere (blue), a cylinder (yellow) and three cameras.
- The cube is added to a layer called “cube_priority” and the sphere is added to a layer called “sphere_priority”. All other objects are rendered in the “default” layer.
- The three models have a rigid body attached and gravity enabled, and the script “MoveAndTrack” is used to randomly translate and rotate the three models in the scene.

- Camera 1: Renders the cube_priority layer with highest depth of 1 and culling mask set to cube_priority layer.
- Camera 2: Renders the sphere_priority layer with a depth of 0 and culling mask set to shpere_priority layer.
- Camera 3: Renders everything else (including the cylinder) with a depth of -1 and culling mask set to everything.
- This gives the cube highest priority in rendering (because its camera has the highest depth), with the sphere having the next highest priority in the scene.
- Thus, the cube always appears in front of the sphere and cylinder, and the sphere always appears in front of the cylinder but behind the cube. You can play/execute the project in the editor several times to verify this, because the movements are randomized and the cube is always given priority, followed by the sphere and then the cylinder when objects collide or overlap.
- Unity project is called "UTRC Test"
- Unity version used for development: Unity 5 2017 1.1f1 Personal