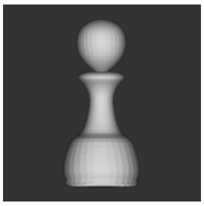
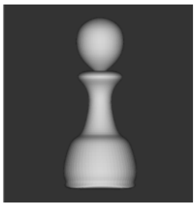
Assignment 1: Loop Subdivision and Ray tracing

Abhishek Aggarwal, 2010CS50272

Loop Subdivision:

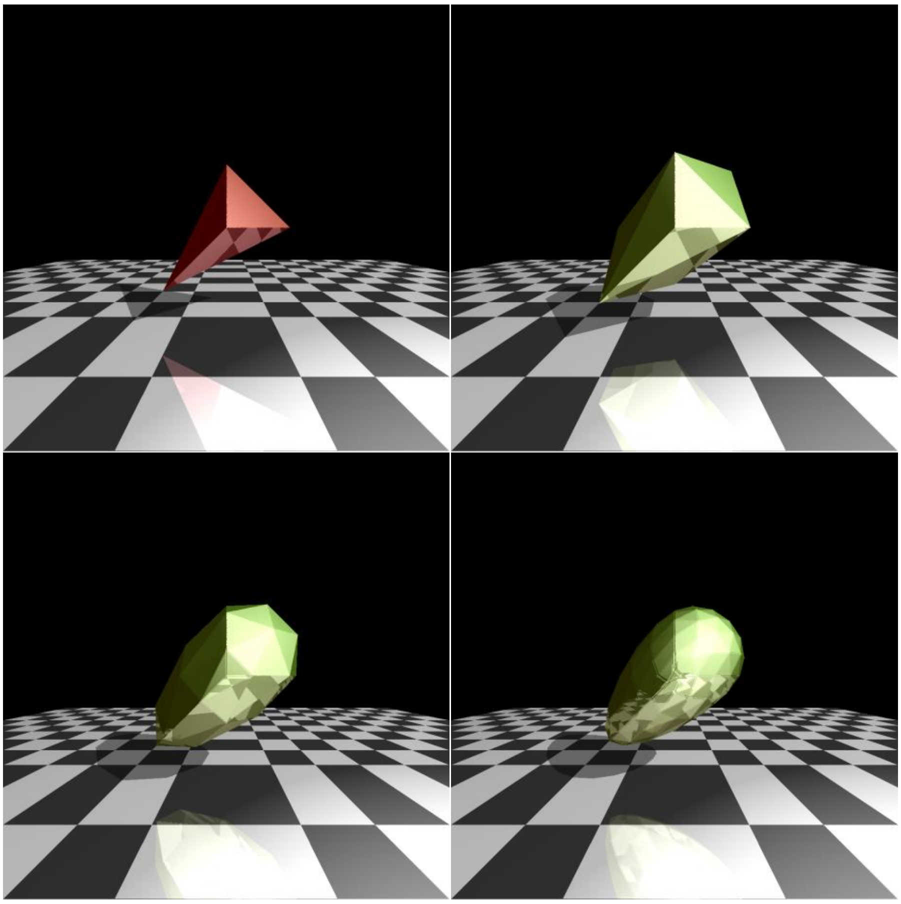
It is a method to subdivide surface and make it smooth. It also considers creases to preserve few edges or points while smoothing the surface.

In my implementation I used winged Edge data structure to store model. In this data structure, model is stored as a set of edges, vertices and faces, where each edge has information about its two end points, left and right faces, edges connecting.

I used this data structure to make efficient queries about neighbouring vertices or edges. In order to implement loop subdivision I run the algorithm in 2 pass. In first pass 'loopSubdivision::subdivide' function is called in which initially vertices are created on each edge and stored in an array, then all faces are creates with these new and old vertices and then all old vertices are updated. In this pass creases are taken care of and one can input information about creases in .obj file with an identifier 'c' followed by the vertex numbers of the end point of crease.

In second pass this array of vertices and faces is again converted to winged edge data structure and rendered on the screen. In above set of pictures one can see the effect of loop subdivision on initial coarse surface.

Raytracing



In the above pictures one can see the ray traced image of a tetrahedron with increasing number of loop subdivision depth along with reflection and shadows till 4 recursion depth.

Approach:

Whole screen is divided into pixels and rays are shot towards pixel. A shadow ray and a reflected ray were sent from the point of intersection of each ray hit at that point. Then the colour is accumulated along that ray by adding the colours obtained from reflected and shadow ray and also colour at that point using phong shading model.

Floating point errors are clearly visible by the distortion in last image of pawn and checkerboard. To take care of such errors, I incorporated a little amount of error while differencing to find the intersection of ray with the triangle but still shadows and reflection is too sharp which can be fixed by distributed ray tracing.

Below we can see the images of pawn and checkboard.