Display Lists

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

void Sphere(float radius, int slices, int stacks)
{
    struct point top, bot;          // top, bottom points
    struct point *p;
    NumLngs = slices; NumLats = stacks;

    if( numLats <= 0 || numLats <= 1)
        return;

    Pts = new struct point[ NumLngs * NumLats ];
    for (int ilat = 0; ilat < NumLats; ilat++)
    {
        float lat = -M_PI/2. + M_PI * (float)ilat / (float)(NumLats-1);
        float xz = cos( lat );
        float y = sin( lat );

        for (int ilng = 0; ilng < NumLngs; ilng++)
        {
            float lng = -M_PI + 2. * M_PI * (float)ilng / (float)(NumLngs-1);

            float x =  xz * cos( lng );
            float z = -xz * sin( lng );

            p = PtsPointer( ilat, ilng );
            p->x  = radius * x;
            p->y  = radius * y;
            p->z  = radius * z;
            p->nx = x;
            p->ny = y;
            p->nz = z;
            p->s = ( lng + M_PI    ) / ( 2.*M_PI );
            p->t = ( lat + M_PI/2. ) / M_PI;
        }
    }
}

void glBegin(GL_QUADS);
{
    for (int ilat = 0; ilat < NumLats-1; ilat++)
    {
        for (int ilng = 0; ilng < NumLngs-1; ilng++)
        {
            p = PtsPointer( ilat, ilng );
            DrawPoint( p );
            p = PtsPointer( ilat, ilng+1 );
            DrawPoint( p );
            p = PtsPointer( ilat+1, ilng+1 );
            DrawPoint( p );
            p = PtsPointer( ilat+1, ilng );
            DrawPoint( p );
        }
    }
}

Drawing a Sphere – Notice a lot of time-consuming Trig Function Calls!

Even worse, the trig calls are inside single or nested for-loops!

dBegin(GL_QUADS);
{
    for (int ilat = 0; ilat < NumLats-1; ilat++)
    {
        for (int ilng = 0; ilng < NumLngs-1; ilng++)
        {
            p = PtsPointer( ilat, ilng );
            DrawPoint( p );
        }
    }
}
You don't want to execute all that code every time you want to redraw the scene, so draw it once, store the numbers in GPU memory, and call them back up later.

Without Display List:
The CPU re-computes and transmits the sphere coordinates every time they need to be drawn.

With Display List:
The CPU computes and transmits the sphere coordinates once and then they are grabbed from GPU memory every time they need to be drawn.

Creating the Display List in InitLists():

Calling up the Display List in Display():