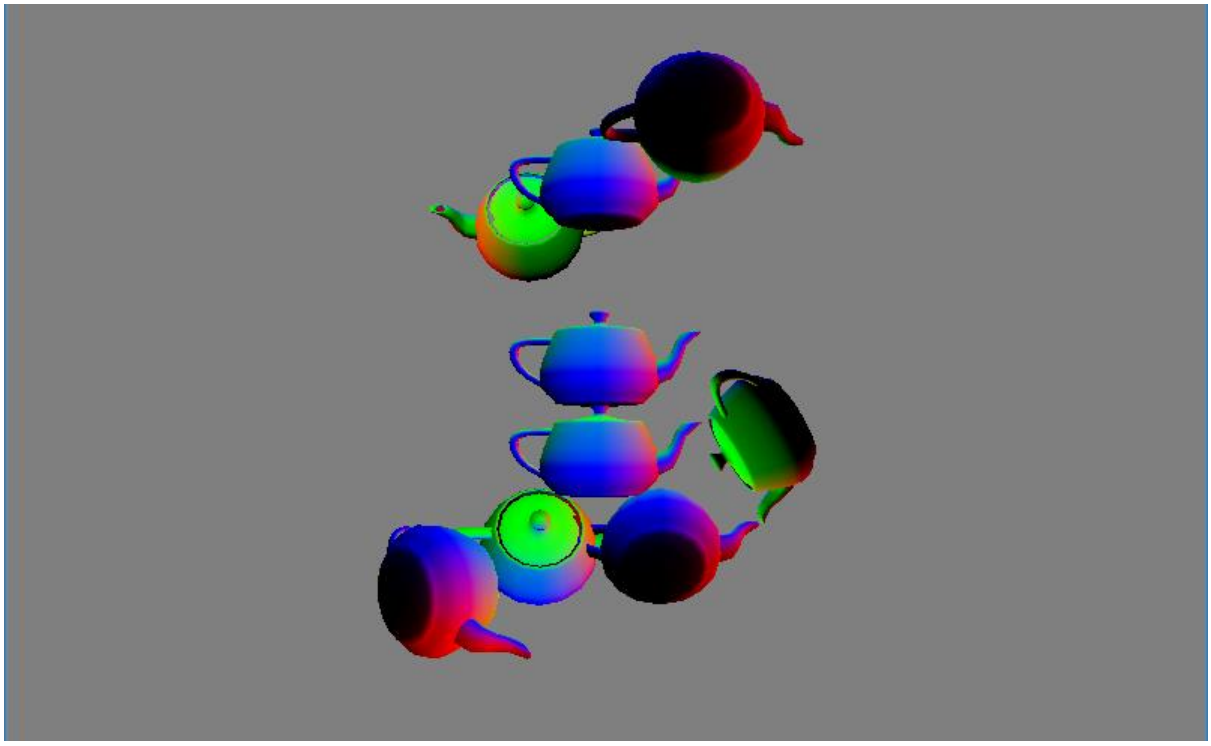
The hierarchy should be constructed from at least 5 teapots

There are 9 teapots in my hierarchy, it contains two one-to-one relationships and two one-to-many relationships.



Show a one-to-one relationship

The one-to-one relationship is accomplished by multiplying one local matrices by one global matrices and never reusing the same global matrices to multiply. The two structures here have a single teapot each rotating at an angle in the x direction.

```
mat4 local6 = identity_mat4();
local6 = rotate_y_deg(local6, -20.0f)*rotate_x_deg(local6, 15.0)*rotate_z_deg(local6, 60.0);
local6 = translate(local6, vec3(15.0, -15.0, 5.0));
local6 = rotate_x_deg(local6, rotatez);
// global of the child is got by pre-multiplying the local of the child by the global of the parent
mat4 global6 = global2*local6;
// update uniform & draw
glUniformMatrix4fv(matrix_location, 1, GL_FALSE, global6.m);
glDrawArrays(GL_TRIANGLES, 0, teapot_vertex_count);

mat4 local7 = identity_mat4();
local7 = rotate_y_deg(local7, 20.0f)*rotate_x_deg(local7, -15.0)*rotate_z_deg(local7, -60.0);
local7 = translate(local7, vec3(-15.0, -15.0, 5.0));
local7 = rotate_x_deg(local7, rotatez);
// global of the child is got by pre-multiplying the local of the child by the global of the parent
mat4 global7 = global5*local7;
// update uniform & draw
glUniformMatrix4fv(matrix_location, 1, GL_FALSE, global7.m);
glDrawArrays(GL_TRIANGLES, 0, teapot_vertex_count);
```

Show a one-to-many relationship

The one-to-many relationship is accomplished by multiplying different local matrices by the same global matrices. The structure here rotates two teapots to the left and right of another teapot in the x direction.

```
mat4 local8 = identity_mat4();
local8 = rotate_x_deg(local8, 180.0)*rotate_y_deg(local8, 180.0)* translate(local8, vec3(15.0, -15.0, 0.0));
// global of the child is got by pre-multiplying the local of the child by the global of the parent
local8 = rotate_x_deg(local8, rotatez);
mat4 global8 = global4*local8;
// update uniform & draw
glUniformMatrix4fv(matrix_location, 1, GL_FALSE, global8.m);
glDrawArrays(GL_TRIANGLES, 0, teapot_vertex_count);

mat4 local9 = identity_mat4();
local9 = rotate_x_deg(local9, 0.0)*rotate_y_deg(local9, 0.0)* translate(local9, vec3(15.0, -15.0, 0.0));
// global of the child is got by pre-multiplying the local of the child by the global of the parent
local9 = rotate_x_deg(local9, rotatez);
mat4 global9 = global4*local9;
// update uniform & draw
glUniformMatrix4fv(matrix_location, 1, GL_FALSE, global9.m);
glDrawArrays(GL_TRIANGLES, 0, teapot_vertex_count);
```

Keyboard control of the translation of the root object

Using global variables for the translation vector parameters, as the button is pressed the root object is translated. Due to the one to many structure of the root object the entire structure is moved

```
// Root of the Hierarchy
mat4 view = identity_mat4 ();
mat4 persp_proj = perspective(45.0, (float)width/(float)height, 0.1, 100.0);
mat4 local1 = identity_mat4 ();
local1 = rotate_y_deg(local1, rotatez)*scale(local1, vec3(0.5f, 0.5f, 0.5f));
local1 = translate (local1, vec3 (translatex, translatey, translatez));

void keypress(unsigned char key, int x, int y) {

    if (key == 'z') {
        translatex += 0.5;
    }
    if (key == 'x') {
        translatex -= 0.5;
    }
    if (key == 'c') {
        translatey += 0.5;
    }
    if (key == 'v') {
        translatey -= 0.5;
    }
    if (key == 'b') {
        translatez -= 0.5;
    }
    if (key == 'n') {
        translatez -= 0.5;

    }
    glutPostRedisplay;
}
```

Interesting/Inventive/Unusual Structure