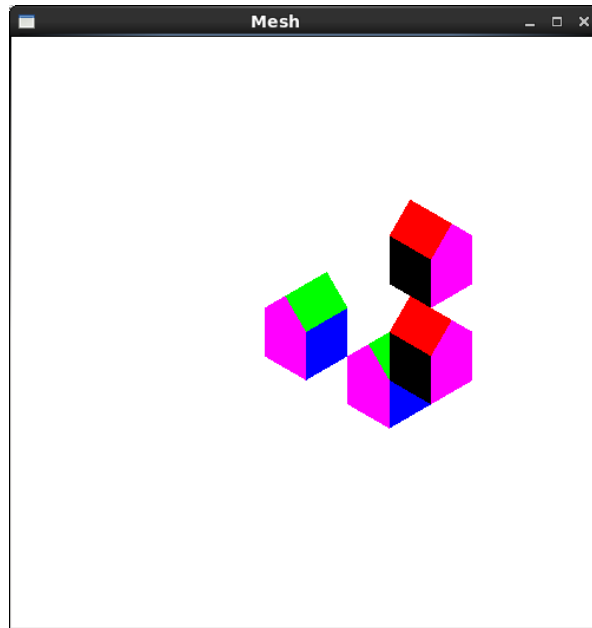


Yazhuo Liu
Lab 10

Add the statements

```
SDL_Delay ( 2000 );  
glFlush();
```

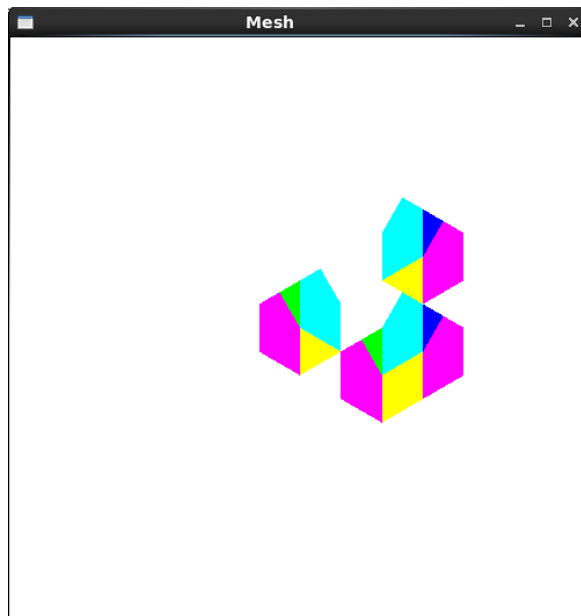


The polygons show up every two seconds.

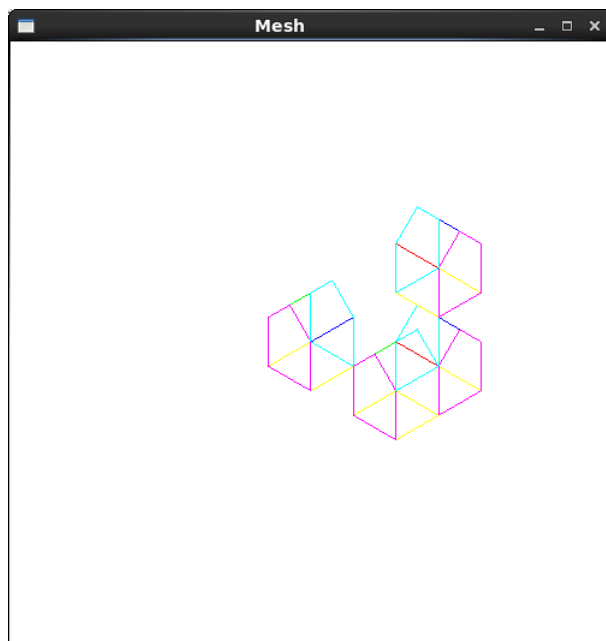
Now add or remove the statements

```
glEnable( GL_CULL_FACE );  
glCullFace ( GL_BACK );
```

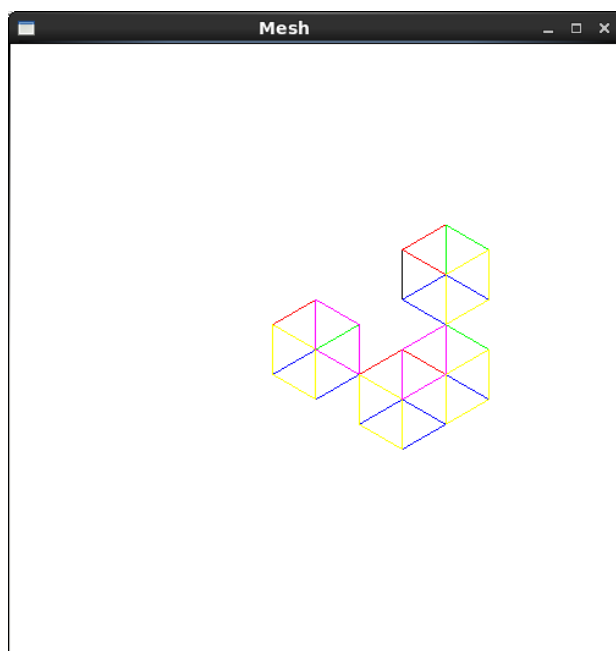
to see the effect of culling back faces.



Remove the cull statments and use "GL_LINE_LOOP" instead of "GL_POLYGON" in the drawing.



Modify the data to draw some other interesting objects.



```
//mesh.cpp
...
void Mesh::drawMesh()    // use OpenGL to draw this mesh
{
```

```
// draw each face of this mesh using OpenGL: draw each polygon.  
if( isEmpty() ) return; // mesh is empty
```

```
glEnable( GL_CULL_FACE );
```

```
glCullFace ( GL_BACK );
```

```
for(int f = 0; f < numFaces; f++) // draw each face
```

```
//for(int f = 6; f < numFaces; f++) // draw each face
```

```
{
```

```
glBegin(GL_LINE_LOOP);
```

```
cout << endl;
```

```
setColor( f );
```

```
for(int v = 0; v < face[f].nVerts; v++) // for each vertex
```

```
{
```

```
    int in = face[f].vert[v].normIndex ; // index of this normal
```

```
    int iv = face[f].vert[v].vertIndex ; // index of this vertex
```

```
    glNormal3f(norm[in].x, norm[in].y, norm[in].z);
```

```
    cout << "[" << norm[in].x << "," << norm[in].y << "," <<
```

```
        norm[in].z << "]" << "  ";
```

```
    glVertex3f(pt[iv].x, pt[iv].y, pt[iv].z);
```

```
    cout << "(" << pt[iv].x << "," << pt[iv].y << "," <<
```

```
        pt[iv].z << ")" << "  ";
```

```
}
```

```
glEnd();
```

```
SDL_Delay ( 2000 );
```

```
glFlush ();
```

```
cout << endl;
```

```
}
```

```
} //drawMesh
```

```
...
```

```
//data.txt
```

```
10 7 7
```

```
0 0 0 1 0 0 1 1 0 0.5 1.5 0 0 1 0
```

```
0 0 1 1 0 1 1 1 1 0.5 1.5 1 0 1 1
```

-1 0 0 -0.707 0.707 0 0.707 0.707 0
1 0 0 0 -1 0 0 0 1 0 0 -1
4 0 5 9 4 0 0 0 0
4 2 4 9 7 1 1 1 1
4 1 2 7 6 2 2 2 2
4 0 1 6 5 3 3 3 3
4 5 6 7 9 4 4 4 4
4 0 4 2 1 5 5 5 5

Report:

I successfully completed all parts of this lab.