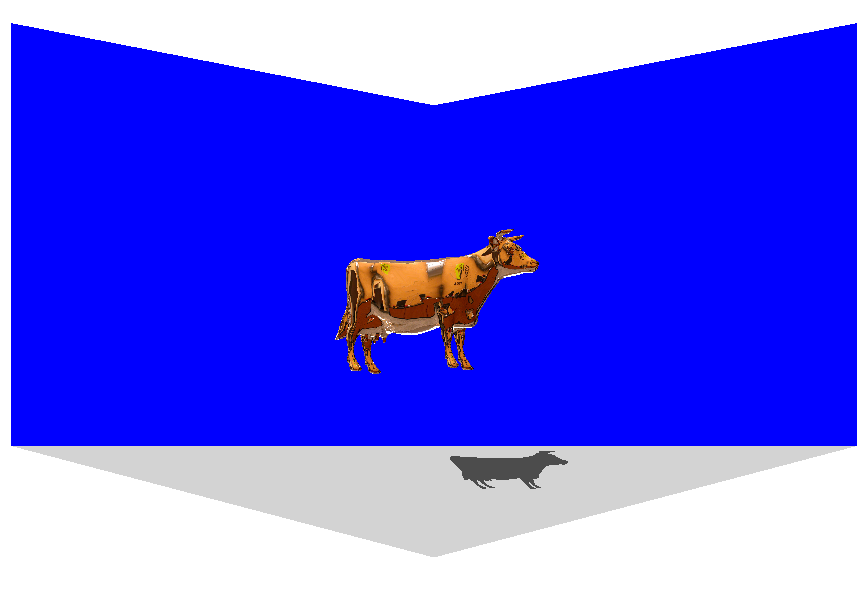
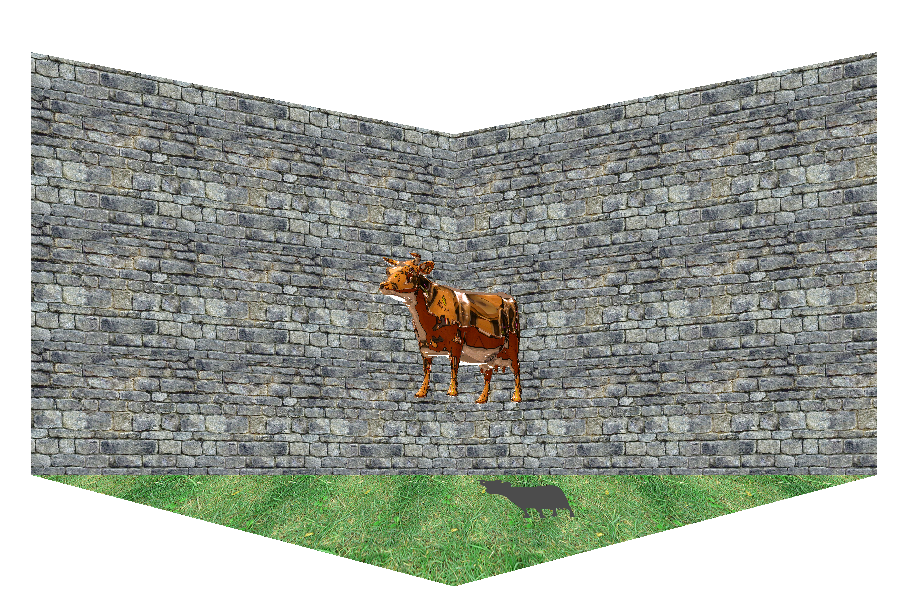
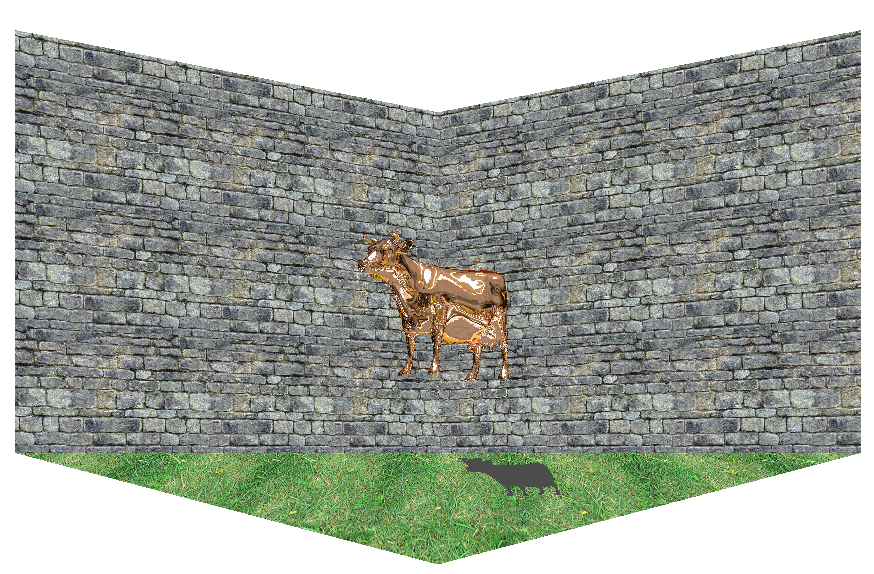
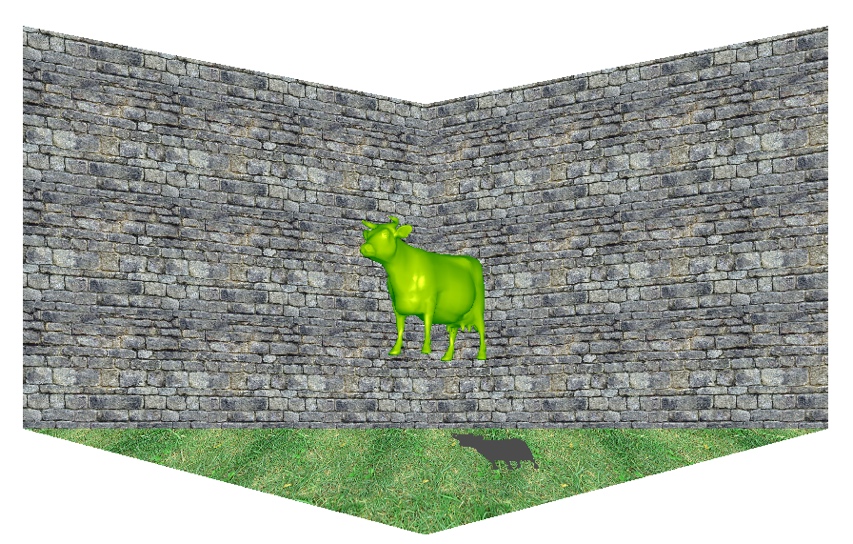
**This project demonstrates:**

1. Calculating face and vertex normals.
2. Phong shading.
3. Reflection.
4. Refraction.
5. Cube maps.
6. Texture maps.
7. Solid textures.
8. Projection shadows.

**Screenshots:**

1. Background with solid textures.
2. Cow model with Phong Shading.
3. Cow model with Reflection of cube map.
4. Cow model with Refraction of cube map.

**Files to view:**

For core OpenGL functionality see following files

modelDraw.cpp

for the model:

fshader1.glsl

vshader1.glsl

for the background:

fshader2.glsl

vshader2.glsl

**Instructions on how to run:**

Open a GLSLExperiment.sln in Visual Studio 18

Please run multiple times if it crashes

(Angel's library makes shader compilers crash)

Key 'a': Toggle shadows ON/OFF.

Shadows are shown using projection.

The toggle is passed to fragment shader for the second object drawn.

That decides the color of the shadow.

The code exists in fragment and vertex shader 1.

Key 'b': Toggle ON/OFF between a scene with grass texture on floor and stone texture on the walls and a plain wall (no texturing).

uses program 2

fragment shader and vertex shader 2 have this code.

I create textures from bitmap and then in fragmentshader 2 I apply texture.

I scale the textures by diving by 3 in vertex shader 2.

TextCoord = vPosition.xy/3;

vPosition is used for s,t

Key 'c': Toggle reflection ON/OFF. When ON, the hierarchy/cow PLY is drawn with reflection.

When OFF, the hierarchy/cow PLY objects is drawn with no reflection

(rendered as a solid model(s) with per-fragment lighting).

reflection code exists in vertex shader 1.

It checks a toggle

if(reflectionToggle == 1.0f)

R = reflect(pos, N);

else if(reflectionToggle == 2.0f)

R = refract(pos, N, 0.011);

else

R = reflect(pos, N);

proper R is then given to fragment shader.

Key 'd': Toggle refraction ON/OFF. When ON, the hierarchy/cow PLY is drawn with refraction.

When OFF, the hierarchy/cow PLY file is drawn with no refraction

( rendered as a solid model(s) with per-fragment lighting).

same as above

R = reflect(pos, N);

else if(reflectionToggle == 2.0f)

R = refract(pos, N, 0.011);

else

R = reflect(pos, N);

The lower refraction ratio draws cow well.