Outline: 11/9

- Stacks revisited
- Shading (using normal vectors)
- Texture Mapping
- Final quiz
- If time: robot with two arms
Matrix Stack

Current Matrix

Matrix Stack
Matrix Stack

Add a transformation: matrix multiplication (right)

Current Matrix

\[ \times \]

Matrix Stack
Matrix Stack

Replace current matrix with result

Current Matrix

Matrix Stack
Matrix Stack

**PUSH:** Add current matrix to the matrix stack
Matrix Stack

Add another transformation: matrix multiplication (right)
Matrix Stack

Replace current matrix with result

Current Matrix

Matrix Stack
Matrix Stack

**PUSH:** Add current matrix to the matrix stack
Matrix Stack

Add another transformation: matrix multiplication (right)
Matrix Stack

Replace current matrix with result

Current Matrix

Matrix Stack
Matrix Stack

POP: Completely get rid of current matrix and pop from the top of the stack
Matrix Stack

Back to where we were before

Current Matrix

Matrix Stack
Matrix Stack

Add another transformation: matrix multiplication (right)

Current Matrix

Matrix Stack
Matrix Stack

Replace current matrix with result
Matrix Stack

**POP:** Completely get rid of current matrix and pop from the top of the stack.
Matrix Stack

Back to the beginning

Current Matrix

Matrix Stack
Shading in OpenGL
Lighting is determined by the surface’s angle to the light source
Lighting is determined by the surface’s angle to the light source

(as defined by the *surface normal*)
Lighting is determined by the surface’s angle to the light source

(as defined by the surface normal)
Lighting is determined by the surface’s angle to the light source

(as defined by the *surface normal*)
Lighting is determined by the surface’s angle to the light source

(as defined by the *surface normal*)
Lighting is determined by the surface’s angle to the light source

(as defined by the surface normal)
Suppose Poly based Normals
Vertex based Normals
Interpolated Surface Normals

glShadeModel(GL_SMOOTH)
glShadeModel(GL_FLAT)  \hspace{2cm}  glShadeModel(GL_SMOOTH)
Begin: Texture Mapping

(next time: texture mapping lab)
Texture Mapping

$\beta$ is how far along AB you are.
$\gamma$ is how far along AC you are.
$\beta, \gamma$ are $[0,1]$
Texture Mapping

Texture Coordinates
Texture Mapping

Texture Coordinates
Texture Mapping

Texture Coordinates
Exercise: what image does this code produce?

```c
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_TEXTURE_2D);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    glBindTexture(GL_TEXTURE_2D, texName);

    glBegin(GL_QUADS);
    glTexCoord2f(0.0, 0.0); glVertex3f(-2.0, -1.0, 0.0);
    glTexCoord2f(0.0, 1.0); glVertex3f(-2.0, 1.0, 0.0);
    glTexCoord2f(1.0, 1.0); glVertex3f(0.0, 1.0, 0.0);
    glTexCoord2f(1.0, 0.0); glVertex3f(0.0, -1.0, 0.0);
    glTexCoord2f(0.0, 0.0); glVertex3f(1.0, -1.0, 0.0);
    glTexCoord2f(0.0, 1.0); glVertex3f(1.0, 1.0, 0.0);
    glTexCoord2f(1.0, 1.0); glVertex3f(2.41421, 1.0, -1.41421);
    glTexCoord2f(1.0, 0.0); glVertex3f(2.41421, -1.0, -1.41421);
    glEnd();

    glFlush();
    glDisable(GL_TEXTURE_2D);
}
```
Image Produced

OpenGL Redbook, Chapter 9, Figure 9-2
Bump Mapping

"Bump-map-demo-full" https://commons.wikimedia.org/
Final quiz

- In-class on Monday 12/14
- One study sheet (hand-written, front and back)
- Focus on the theoretical aspects of the second half of the semester

Reasons:
- Quiz earlier so there is time for final project
- Self-scheduled exam period is very late
- Helps you study in a focused manner