1 Administrivia

Announcements

Assignment

Hand-out final project.

From Last Time

Exam.

Outline

1. Introduction to texture mapping.

2. A few issues/problems.

3. Example program: checker.c

Coming Up

Texture lab.
2 Introduction

1. What is a texture?

2. Why texture map? Complexity from simplicity.

3. Where can I get textures? Fixed patterns (polygon fills), texture generation functions, digitized images.

4. Do Web browsers texture map?

5. 1-D and 2-D textures.

3 A Little More Detail

1. Texture pattern: \( T(s, t) \). Stored in texture memory as array of texels. (Texture memory and the rise of AGP.)

2. Texture mapping: mapping from \( T \) to object’s geometric coordinates. (Must then map to screen coordinates.)

3. Possible problems: curved surfaces, closed surfaces, pixels not inverse mapping onto texels, aliasing.

4. How is a texture mapped to an object?
   
   (a) texture and vertex coordinates.

   (b) Range of texture coordinates.

   (c) Wrapping modes: repeat, clamp.

5. How is it applied to an object?

   (a) Modulation, decal.

4  checker.c

Numbers in parentheses refer to line numbers in handouts.

   Interpretation?

2. Initializing: `init()` (39).
   (a) Storage formats, texture handles and binding.
   (b) Wrapping modes.
   (c) Minification, magnification filters.
   (d) `glTexImage2D()` — ugh.

3. Rendering: `display()` (73).
   (a) Enabling textures and specifying mapping mode.
   (b) Specifying which texture.
   (c) Binding texture and vertex coordinates.

4. Comparison of code with rendered results.

Experiments:

1. Is the lower right result a fluke? Rotate the lower left polygon and see what happens.

2. Specify a color for the polygons. Did anything change? Change the mode from `GL_DECAL` to `GL_MODULATE` and try again.

3. Individually change the `GL_REPEAT`s to `GL_CLAMP`s and see what happens.

Brief demos of other programs, time permitting.