abbreviations expressed as macros.
you will, an object and header file for the (trunk) function and the exam.
OpenGl programming question if you do it on the computer. I will supply
Midterm 2 catch-up: Up to 15 points (half credit) will be used for the
user interaction, and lighting/viewing.
One more OpenGl project... Will involve 3d objects' viewing/projection
Now.
If you haven't already, also read HB 10-1, 10-2, 10-3, 10-20 pages 637-647.
10-9; Read more carefully 10-10.
READING IN HB: Finish 10-3, Read quickly 10-4, 10-5, 10-6, 10-7, 10-8.
As you read it, play with my Blender session... Click on the link.
of color index mode are optional.
READING: Chapters 4 (Color) and 5 (Lighting) in the RedBook. (Details
CS1 422/502: Lecture 29
MAT372, because bounded convex functions on them take on their maximum
determined by hyperplanes (which are possibly unbounded volumes whose faces are
convex polyhedra) and bounded by hyperplanes (which are possibly unbounded volumes whose faces are
not convex polyhedra, which no “corners” or “hardcorners” or “indentations”.

A convex polygon has no “corners” or “hardcorners” or “indentations”.

A polygon is part of the polygon.
and boundary of the polygon, the entire line segment between $P_1$ and $P_2$ are inside or on the
boundary of the polygon, the entire line segment between $P_1$ and $P_2$.
and convex: For every pair of points $P_1$, $P_2$ inside or on the
their endpoints, and only one connected component.
simple: edges never cross or touch except for adjacent edges at
planar: all vertices in one plane.

The OpenGL supports drawing only 3D polygons (non-degenerate
The practical, OpenG solution to the problem of messy meshes:

University at Albany Computer Science Dept.
polygons or straight lines for OpenGL to approximate them.

Also, GLU support for defining curved surfaces and lines (like circles) by
quadric formulas (degree 2 polynomials) and then generating lots of
OpenGL (OpenGL standard utility library) on top of, but not inside

See Redbook Chapter 11:
OGL must tessellate each non-convex face before asking OpenGL to render it.
If an application generates models with non-convex polygons as faces, it
4. Plane is part of a model for light reflection, scattering, color, and more.

OpenGL requires at least 6 bit acocompatsions, allowing implementations of Good for section views.

Redbook.

3. Plane is one of OpenGL optional clipping planes, see HB 7-12 or

\[ 0 = m + z \quad 0 = m + z \quad 0 = m + h \quad 0 = m + h \quad 0 = m + x \quad 0 = m + x \]

2. Plane is one of the 6 NDC clipping planes:

1. Plane is a projection surface for perspective or parallel projection.

\[ 0 = m \mathbf{D} + z \mathbf{C} + h \mathbf{B} + x \mathbf{A} = \begin{bmatrix} m \\ z \\ h \\ x \end{bmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{C} & \mathbf{B} & \mathbf{D} \end{pmatrix} \]

Equation of a plane (homogeneous coods):

University at Albany Computer Science Dept.
operations) is the \((A', B', C')\) of the plane equation.

RGBA color, to each OpenGL vertex (generated by `glVertex()`)

Each OpenGL normal vector (attached, like 3D coordinates and

emission, etc.)
The normal vectors.

The vector cross product and length calculation is used to calculate

Building an Icosahedron Example.

See Redbook's Building an Icosahedron Example.

normal vector and vertex coordinates for each.

2. Might generate the polygons for a surface one by one, calculating the

3D world of solid objects, lights, cameras, etc.

1. Mesh data structures, and hierarchical world models using them, for a

Higher level models.
How do you distinguish the front side from the back side of one face

Blender training HW shows this.

But backface culling should not be done if we sometimes want to see

Saving work this way is called backface culling.

So, all rendering work can be avoided if the outside side faces away

the outside.

If the polygon is part of a "closed surface" (actually a 2D surface

without any 3D boundary edges), none of the inside will be visible from

I. The two different sides might have different colors.

Reasons:

2. Faces the outside of a (mesh represented) surface which the polygon is

part of.

2. Faces the outside of a (mesh represented) surface which the polygon is

I. Faces the viewing direction?

Important question: "Issue" in 3D: Which side of a (planar) polygon
When you look at the face from the outside, the vertices are listed in clockwise (mathematically positive) order.

Answer: The order of listing the vertices in each face is chosen so—
COUNTER-CLOCKWISE order.

A, B, and C are vertices of the triangle in positive or counterclockwise order.

The resulting vector will point TOWARD a viewer who sees the sequence ABC.

\[ \frac{|\mathbf{y}|}{N} = N \]

and dividing (3 times) to get \( N \).

Then calculate \( \mathbf{y} \) by calculating \( \| \mathbf{y} \| \) and normalizing it.

\( \mathbf{y} = (\mathbf{A} - \mathbf{C}) \times (\mathbf{A} - \mathbf{B}) \)

Calculate \( \| \mathbf{y} \| \) and normalizing it.

\[ \mathbf{y} = \mathbf{A} - \mathbf{C} \]

\[ \mathbf{y} = \mathbf{A} - \mathbf{B} \]

and \( \mathbf{y} \) not all in one line.

Given coordinates of 3 points \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) on a plane surface:

How to calculate the normal of a plane surface:
This is one scalar equation

The equation of the plane determined by 3 non-coplanar points $A', B', C'$ is

Way 2: (Really cool way to use Projective Geometry to remember the

\[
0 = \begin{vmatrix}
1 & 1 & 1 & 1 \\
\varphi & \varphi & \varphi & \varphi \\
\varphi & \varphi & \varphi & \varphi \\
\varphi & \varphi & \varphi & \varphi \\
\end{vmatrix}
\]
OpenGL shading:

Calculation of normals to be attached to OpenGL vertices is necessary for projections of \( p \) on the \( yz, \) \( xx \) and \( xy \) planes.

Practice book says: \( (a', b', c') \) are proportional to the areas of the polygons. 

Poly, Van Dam, Reiner, Hughes, the Computer Graphics Principles and
they are not quite all in the same plane?

\( W \)ay (3):
\( \text{What if the mesh's polygon} \ p \ has 4 or more vertices, and they
polygons. Calculated by interpolation of vertex colors.

smooth (or Gouraud) shading: color varies smoothly over the filled

triangle or quad vertex.

Hat shading: color is equal to the color of last polygon vertex and last

point, line and filled polygon colors.

Simple 3D coloring (Chapter 4 of Redbook): Vertex color(s) determine
Please, but this is "broken" in LC-3, 4.

Animation: Blender can generate and sequentially display a series of images.

Image output to see complete results.

Sad experience: You have to push the RENDER button and wait for a full render.

Textured provides approximate preview of light/materials model rendering.

Shaded, Textured.

Blender 3D "preview" rendering options: Bounding Box, Wire Frame, Flat.
(c) `enable(GL_LIGHTING)`

(b) `enable(GL_DEPTH_TEST)`

(a) Display modes must include RGB and DEPTH.

- Function calls to enable GL features or display modes:

- Lighting/reflecton model: ambient, diffuse, specular, and emission

Different material properties can be specified for the four different kinds of

- Material properties for each vertex and kind of face it's in (FRONT, BACK)

3. 1 or more light sources:

2. a Lighting model: Ambient Light and Location of viewpoint.

1. a normal vector at each vertex (or be lazy and let `gl_normal3d`).

What you (OpenGL programmer) must provide:

- Rendeing (or "shading") with a Lighting and Materials Model.
multiple lights,

(e) 

and others (LIGHT1, LIGHT2, etc.) for

Whenever clearing the screen,

(d) 

equire (GL-COLOR-BUFFER-BIT, GL-DEPTH-BUFFER-BIT),