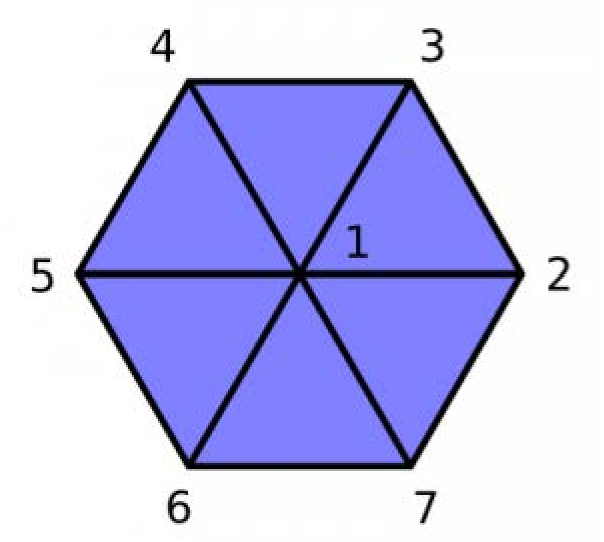
CS 418: Interactive Computer Graphics Spring 2019
In-class Worksheet 4

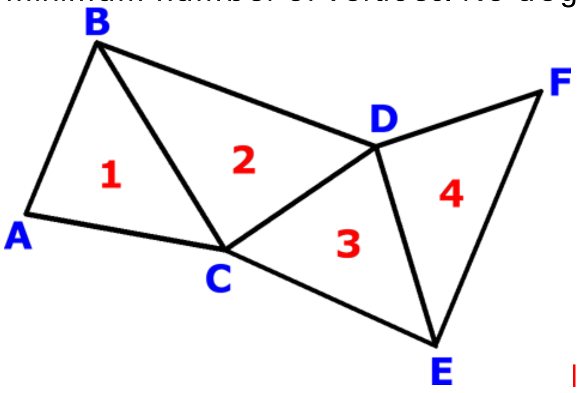
3D Geometry and Geometric Primitives

1. Draw a picture of a triangulated polygon that can be drawn using a single triangle fan but not a single triangle strip. You must use only the minimum number of vertices. No degenerate triangles can be used.



In a strip, no vertex can appear in more than 3 triangles.

2. Draw a picture of a triangulated polygon that can be drawn using a single triangle strip but not a single triangle fan. You must use only the minimum number of vertices. No degenerate triangles can be used.



In fan, one vertex appears in all triangles.

3. Suppose a triangle has a normal vector of $\langle 1,1,0 \rangle$ and that the vector for the view direction is $\langle 1,-2,0 \rangle$. Is the triangle front-facing or back-facing?
   $\langle 1,-2,0 \rangle \cdot \langle 1,1,0 \rangle = -1$
   Assuming the view vector is expressed as running from the eyepoint to the surface, the triangle is front-facing.

4. The following vertex buffer is suitable for drawing 3 triangles using gl.TRIANGLES and gl.DRAW_ARRAYS. Convert the buffer to one suitable for drawing the same triangles using gl.TRIANGLE_STRIP and gl.DRAW_ARRAYS. Assume we are using a CCW winding order.

V1
V2
V3
V3
V4
V1
V4
V3
V5

We can express the mesh as a strip with this buffer:

V5
V4
V3
V1
V2

5.

The Euler Characteristic

The Euler Characteristic states the following relationship for the elements of a closed and connected surface mesh:

$$V-E+F=2(1-G)$$

V is the number of vertices
E is the number of edges
F is the number of faces
G is the genus of the surface (how holes/handles it has)

Show that for a triangle mesh with no holes we have $F \approx 2V$. Hint: each face has 3 edges and each edge is shared by 2 faces.

$V-E+F=2$
$V-(3/2)F+F=2$
$V-(1/2)F=2$
$V=2+(1/2)F$
$2V=2+F$

Memory Requirements

Using the fact that $F \approx 2V$, compare the storage requirements for an indexed face mesh and a triangle soup (in WebGL this corresponds to using `gl.drawElements` versus `gl.drawArrays`). Assume the mesh has V vertices and a number requires 4 bytes of space. Derive functions for the number of bytes the mesh will require as a function of V.

drawArrays: F x 3 vertices per face x 3 coords per vertex x 4B per number = 2Vx36 = 72 bytes per vertex

drawElements: F x 3 vertex indices per face x 4 bytes +
V x 3 coords per vertex x 4 bytes
= 2V x 12 + 12V = 36V = 36 bytes per vertex