Modeling Objects with Polygons

**Polygons**

A polygon is a closed sequence of edges. They can only approximate a curved object.

A polygon mesh is a collection of polygons connected such that each edge is shared by at most two polygons.

**Polygon mesh representation:**

1) explicit list of pointers
2) pointers to vertex list
3) pointers to edge list

**Rep.#1:** Each polygon is rep. by a list of vertex coords

\[
P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n))
\]

stored in order of traversal around polygon

**Advantage:** space-efficient for a single polygon

**Disadvantage:** space-inefficient for polygon mesh because coords of shared vertices are duplicated. No explicit rep of shared edges + vertices (to save time in clipping, drawing, interactive manipulation,...)
Rep#2: Each polygon is rep. by a list of ptrs to a vertex list.

\[ V = ((x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)) \]
\[ P = (3, 5, 7, 10) \quad \text{polyon made up of vertices 3, 5, 7, 10} \]

Ex: \[ V = (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)) \]
\[ P_1 = (1, 2, 4) \quad P_2 = (4, 2, 3) \]

Note that \( V_2 \) and \( V_4 \) are not duplicated.

Advantage: Vertices are stored once; saves space; coords of a vertex can be changed easily (visit vertex list only)

Disadvantage: Difficult to find polygons that share edges; shared edges are drawn twice
Rep #3: Each polygon is rep. by a list of ptrs to an edge list.

Each edge in the edge list occurs once, points to 2 vertices in vertex list, and also to 1 or 2 polygons to which the edge belongs.

\[ P = (E_1, E_2, \ldots, E_n) \]
\[ E_i = (V_1, V_2, P_1, P_2) \]

**Ex:**

\[ V = (V_1, V_2, V_3, V_4) = ((x_1, y_1, z_1), \ldots, (x_4, y_4, z_4)) \]
\[ E_1 = (V_1, V_2, P_1, \emptyset) \]
\[ E_2 = (V_2, V_3, P_2, \emptyset) \]
\[ E_3 = (V_3, V_4, P_2, \emptyset) \]
\[ E_4 = (V_4, V_2, P_1, P_2) \]
\[ E_5 = (V_4, V_1, P_1, \emptyset) \]
\[ P_1 = (E_1, E_4, E_5) \]
\[ P_2 = (E_2, E_3, E_4) \]

**Advantage:** Knowing shared edges lets us avoid redundant clipping, drawing, transformations, + scan conversion.

**Disadvantage:** Still not easy to determine which edges are incident to a vertex.