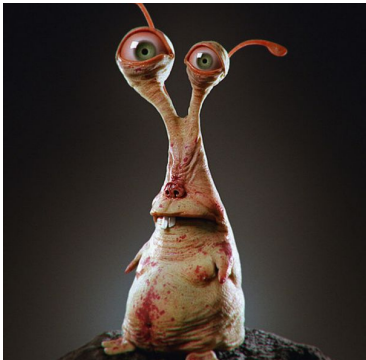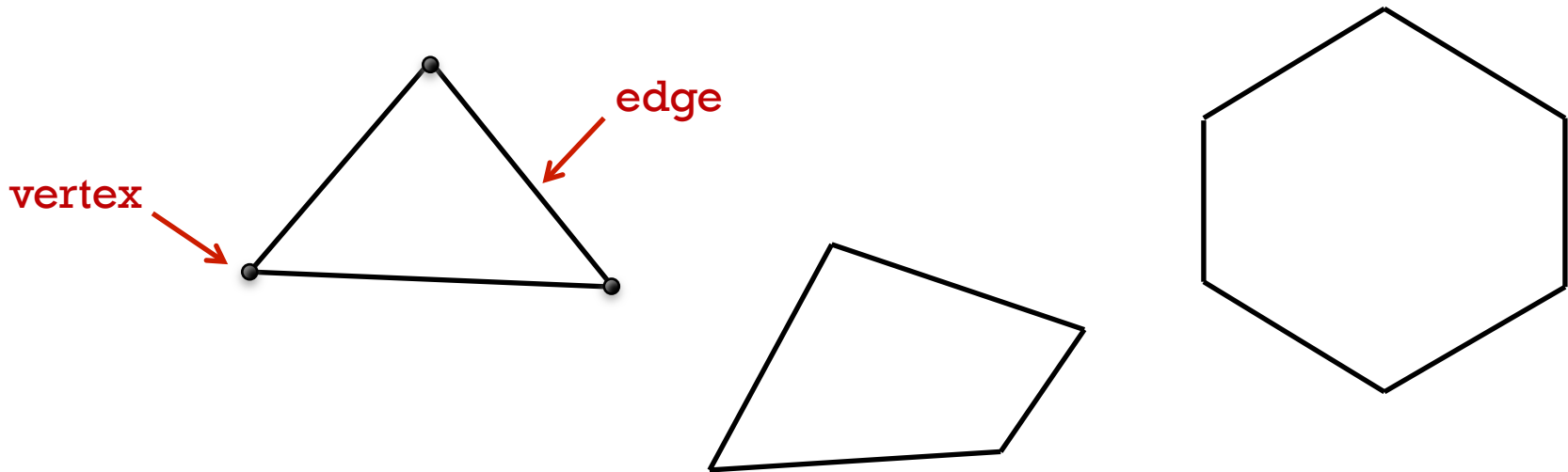# 3D Modelling

3D Computer Graphics (Lab 2)
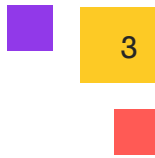
Polygon

- A polygon is a plane figure that is bounded by a closed path, composed of a finite sequence of straight line segments.
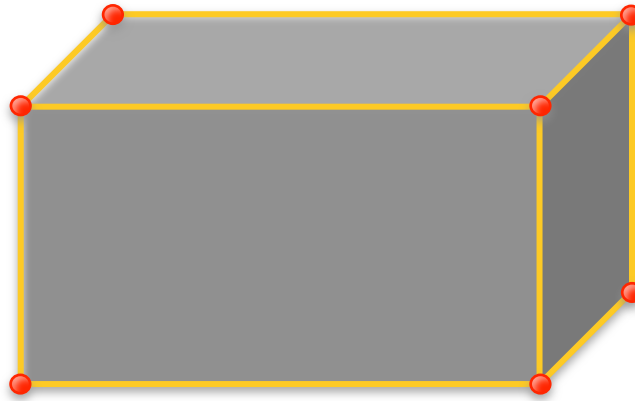
edge

vertex

- These straight line segments are called edges.

- The points where two edges meet are called vertices.

Polygonal mesh

- A (polygonal) mesh is a collection of polygons that approximate the surface of a 3D object.

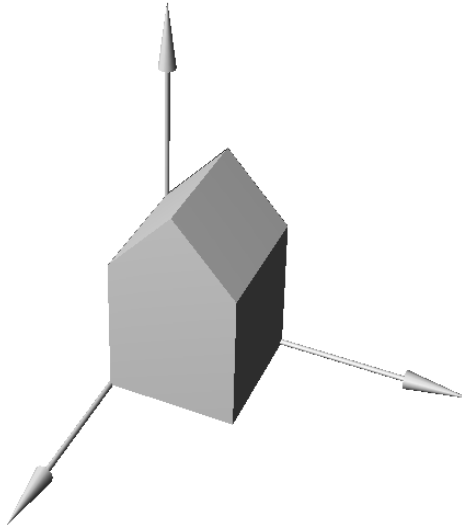
vertices

edges

faces

- How many faces does this mesh have?   6

- How many vertices does this mesh have?   8

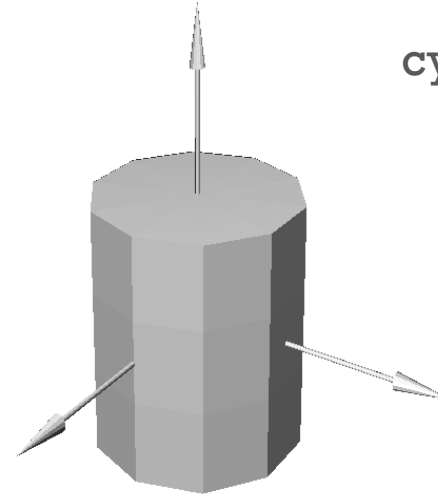- How many edges does this mesh have?   12

Polygonal mesh

barn

cylinder

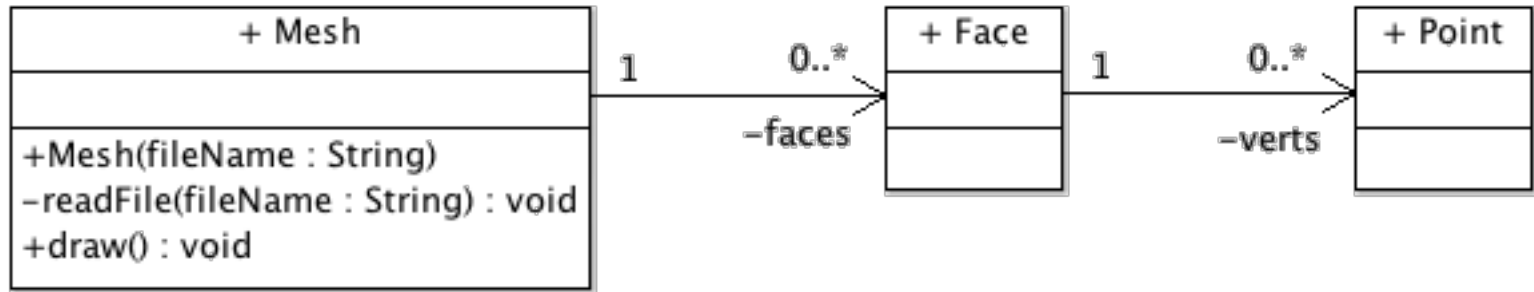Accurate representation

Approximate representation

Meshes can approximate the surface to any degree of accuracy by making the mesh finer (more faces) or coarser (less faces).

How can we represent a polygonal mesh in software?

Polygonal mesh

- A mesh is a list of faces.
- Each face is a list of vertices.
- Each vertex is a point which can be represented by its three coordinates.

```
+----------------------------+           +--------+           +---------+
|          + Mesh            |           | + Face |           | + Point |
+----------------------------+  1   0..* +--------+  1   0..* +---------+
|                            |---------->|        |---------->|         |
+----------------------------+    -faces |        |    -verts |         |
| +Mesh(fileName : String)   |           +--------+           +---------+
| -readFile(fileName : String) : void |
| +draw() : void             |
+----------------------------+
```
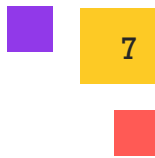
The list of vertices of a face contains the vertices in a particular order.

Convention:
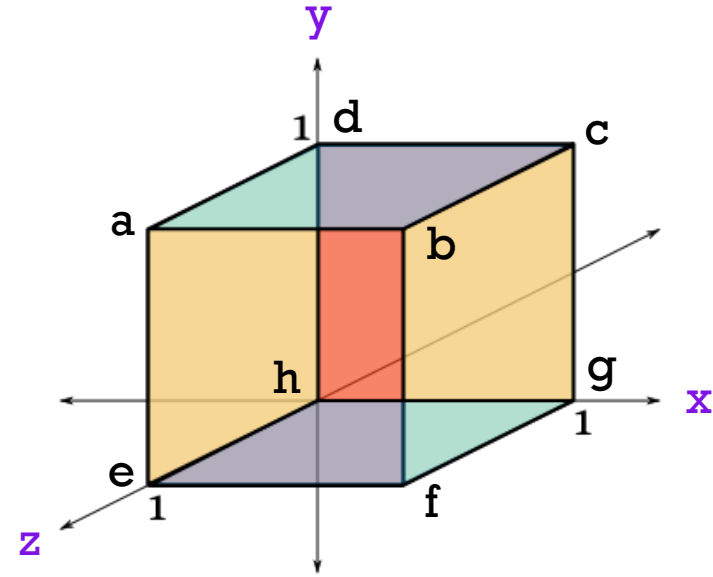
Traverse the face counterclockwise as seen from outside the object.

Using this convention allows to carry out some operations faster later on.

Example

4 a e f b	(front)
4 h d c g	(back)
4 a b c d	(top)
4 a d h e	(left)
4 e h g f	(bottom)
4 f g c b	(right)

a = (0,1,1)
b = (1,1,1)
c = (1,1,0)
d = (0,1,0)
e = (0,0,1)
f = (1,0,1)
g = (1,0,0)
h = (0,0,0)

```
4  0 1 1   0 0 1   1 0 1   1 1 1
4  0 0 0   0 1 0   1 1 0   1 0 0
4  0 1 1   1 1 1   1 1 0   0 1 0
4  0 1 1   0 1 0   0 0 0   0 0 1
4  0 0 1   0 0 0   1 0 0   1 0 1
4   1 0 1   1 0 0   1 1 0   1 1 1
```
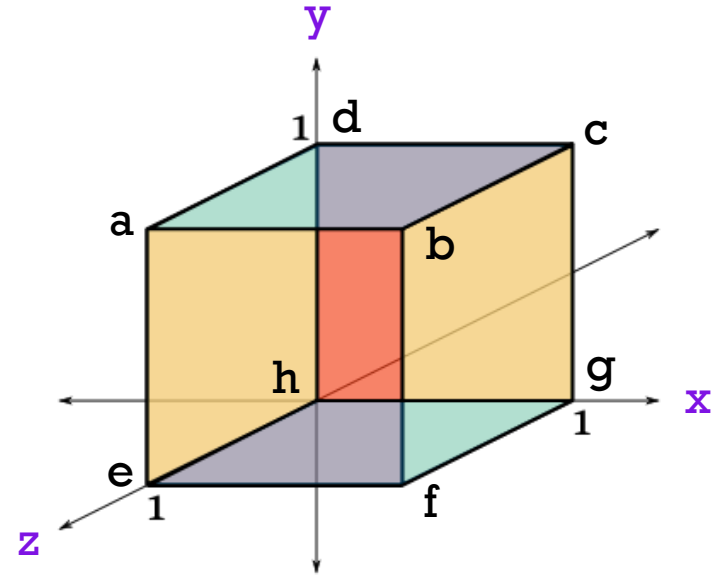
File format ok?

Data duplication

Solution?

Example

4 a e f b	(front)
4 h d c g	(back)
4 a b c d	(top)
4 a d h e	(left)
4 e h g f	(bottom)
4 f g c b	(right)

a = (0,1,1)
b = (1,1,1)
c = (1,1,0)
d = (0,1,0)
e = (0,0,1)
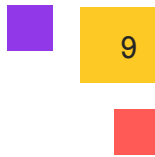f = (1,0,1)
g = (1,0,0)
h = (0,0,0)

```
0 1 1   1 1 1   1 1 0   0 1 0
0 0 1   1 0 1   1 0 0   0 0 0
4 0 4 5 1
4 7 3 2 6
4 0 1 2 3
4 0 3 7 4
4 4 7 6 5
4 5 6 2 1
```
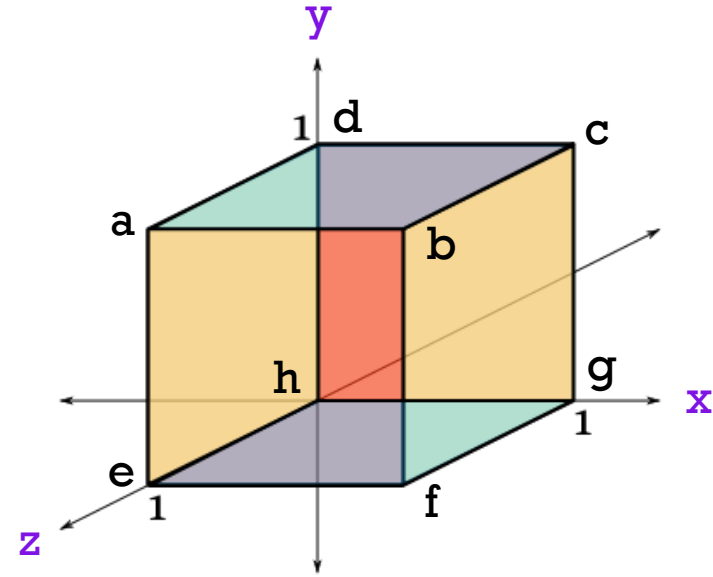
File format ok?

How does the file parser know when the list of coordinates ends?

Solution?

Example

```
4 a e f b      (front)
4 h d c g      (back)
4 a b c d      (top)
4 a d h e      (left)
4 e h g f      (bottom)
4 f g c b      (right)
```

```
a = (0,1,1)
b = (1,1,1)
c = (1,1,0)
d = (0,1,0)
e = (0,0,1)
f = (1,0,1)
g = (1,0,0)
h = (0,0,0)
```

```
8
0 1 1   1 1 1   1 1 0   0 1 0
0 0 1   1 0 1   1 0 0   0 0 0
4 0 4 5 1
4 7 3 2 6
4 0 1 2 3
4 0 3 7 4
4 4 7 6 5
4 5 6 2 1
```

number of vertices

Example

```
4 a e f b    (front)
4 h d c g    (back)
4 a b c d    (top)
4 a d h e    (left)
4 e h g f    (bottom)
4 f g c b    (right)
```

```
a = (0,1,1)
b = (1,1,1)
c = (1,1,0)
d = (0,1,0)
e = (0,0,1)
f = (1,0,1)
g = (1,0,0)
h = (0,0,0)
```

Why do we add the number of faces?

- Extra check for the validity of the data.

- Allows to allocate enough memory at once to store all the faces. (time efficiency)

```
8 6
0 1 1   1 1 1   1 1 0   0 1 0
0 0 1   1 0 1   1 0 0   0 0 0
4 0 4 5 1
4 7 3 2 6
4 0 1 2 3
4 0 3 7 4
4 4 7 6 5
4 5 6 2 1
```
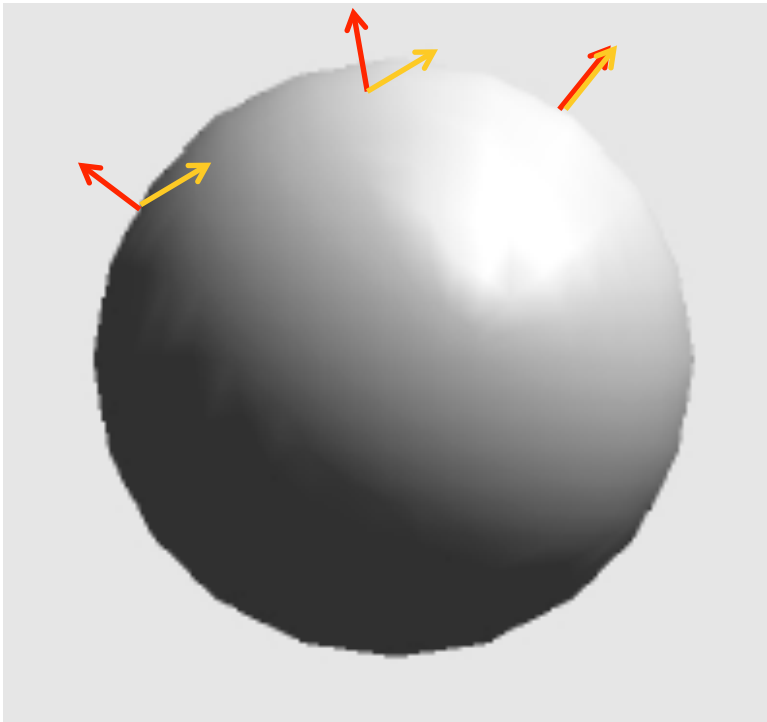
number of faces

File format ok?

In theory, yes.

*But in practice,
we need more data.*

Lighting calculations (preview)

The larger the angle between the red and yellow vector at a point of the surface, the less light this point receives from the light source.
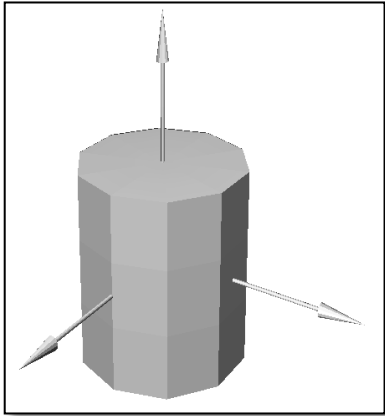
direction to the light source

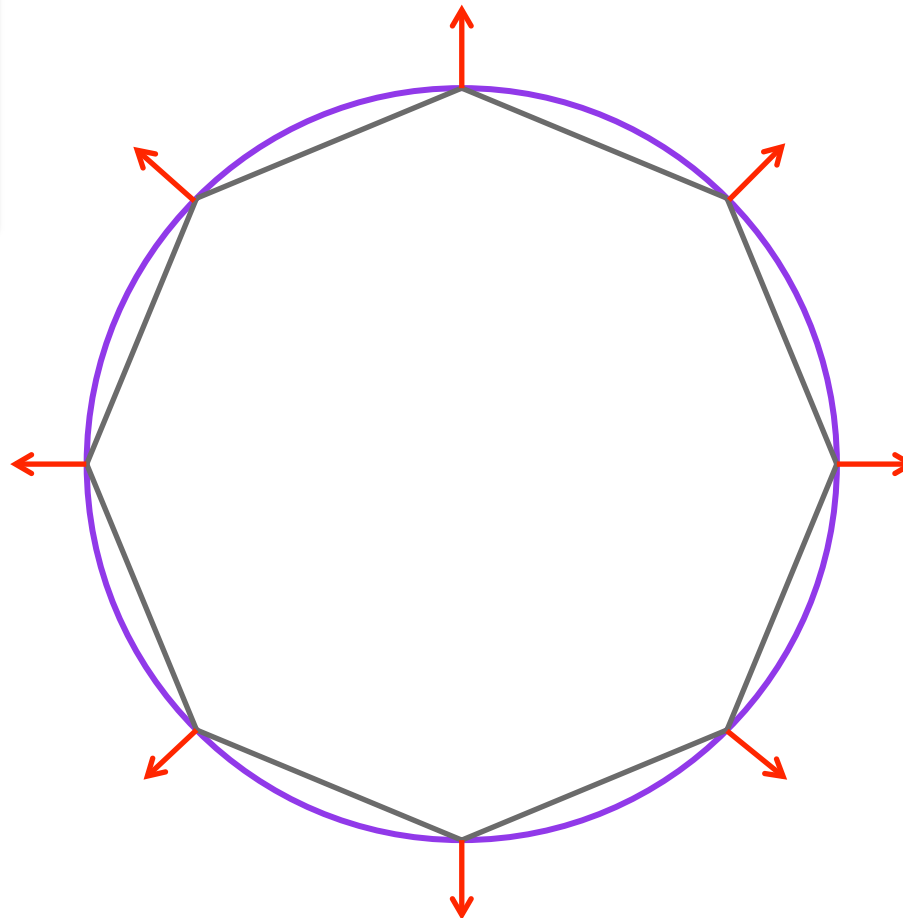normal vector to the surface, a <u>unit</u> vector which is <u>perpendicular</u> to the surface and is pointing <u>outward</u>.

We will add the normal vectors to the surface of the 3D object for each vertex of the mesh.

Polygonal mesh

Top view of cylinder

The normal vector is a vector which is perpendicular to the true surface!

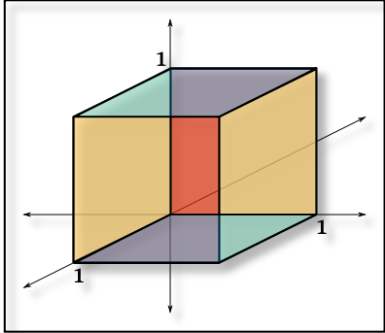The vertices of a face do not necessarily have the same normal vector!
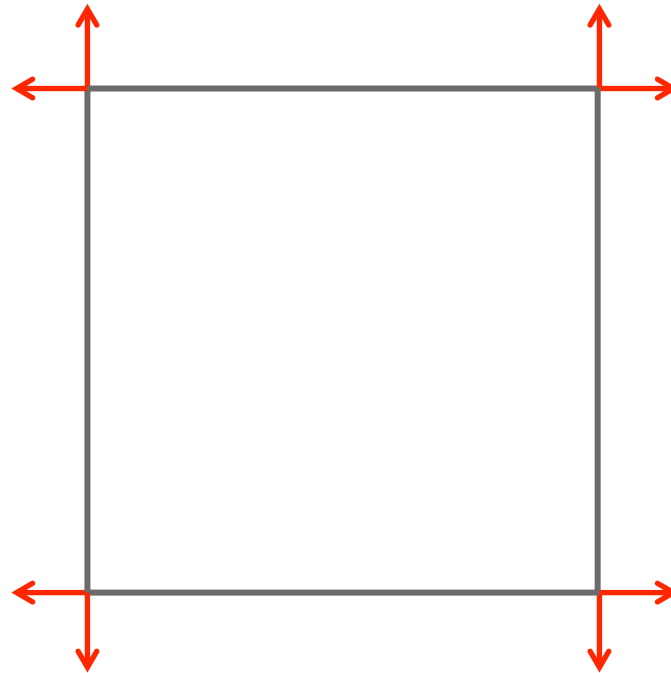
true surface

polygonal mesh

normal vector

Polygonal mesh

Top view of cube

More than one normal vector can be associated with a point of the mesh depending on the face it is part of.
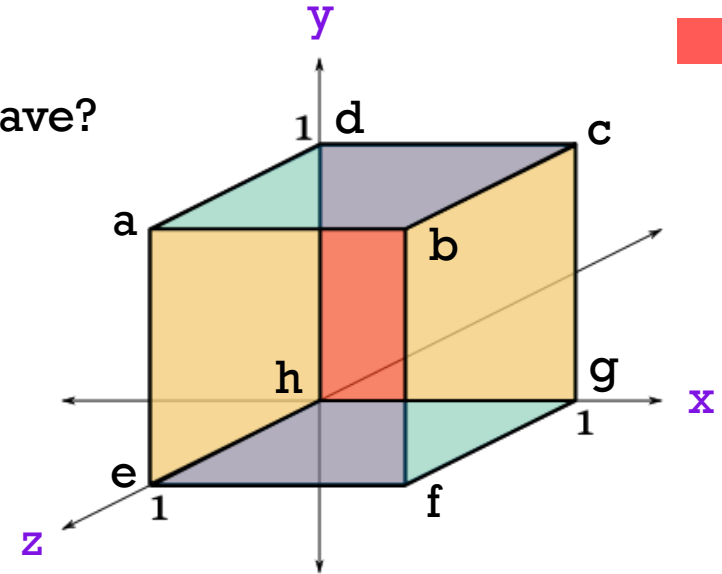
polygonal mesh
= true surface

normal vector

Example

How many distinct normal vectors does this mesh have?

(0,0,1)	(front)
(0,0,-1)	(back)
(0,1,0)	(top)
(-1,0,0)	(left)
(0,-1,0)	(bottom)
(1,0,0)	(right)

```
8 6
0 1 1  1 1 1  1 1 0  0 1 0
0 0 1  1 0 1  1 0 0  0 0 0
4 0 4 5 1
4 7 3 2 6
4 0 1 2 3
4 0 3 7 4
4 4 7 6 5
4 5 6 2 1
```

```
8 6 6
0 1 1  1 1 1  1 1 0  0 1 0
0 0 1  1 0 1  1 0 0  0 0 0
0 0 1  0 0 -1  0 1 0
-1 0 0  0 -1 0  1 0 0
4 0 4 5 1  0 0 0 0
4 7 3 2 6  1 1 1 1
4 0 1 2 3  2 2 2 2
4 0 3 7 4  3 3 3 3
4 4 7 6 5  4 4 4 4
4 5 6 2 1  5 5 5 5
```

number of
normal vectors

coordinates of
normal vectors

indices of the
normal vectors
at the vertices
of the faces

Final file format to store a mesh

number of vertices

number of normal vectors
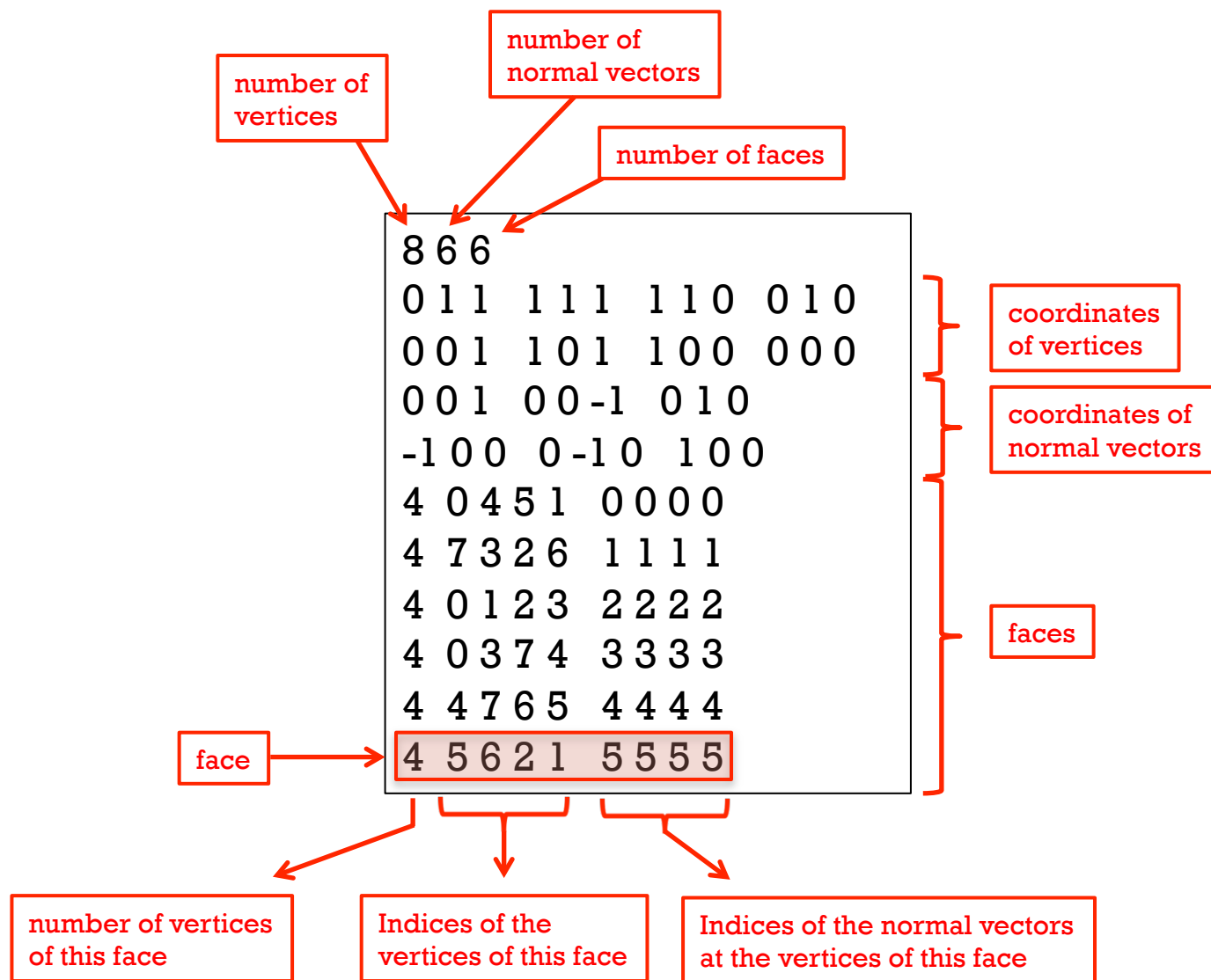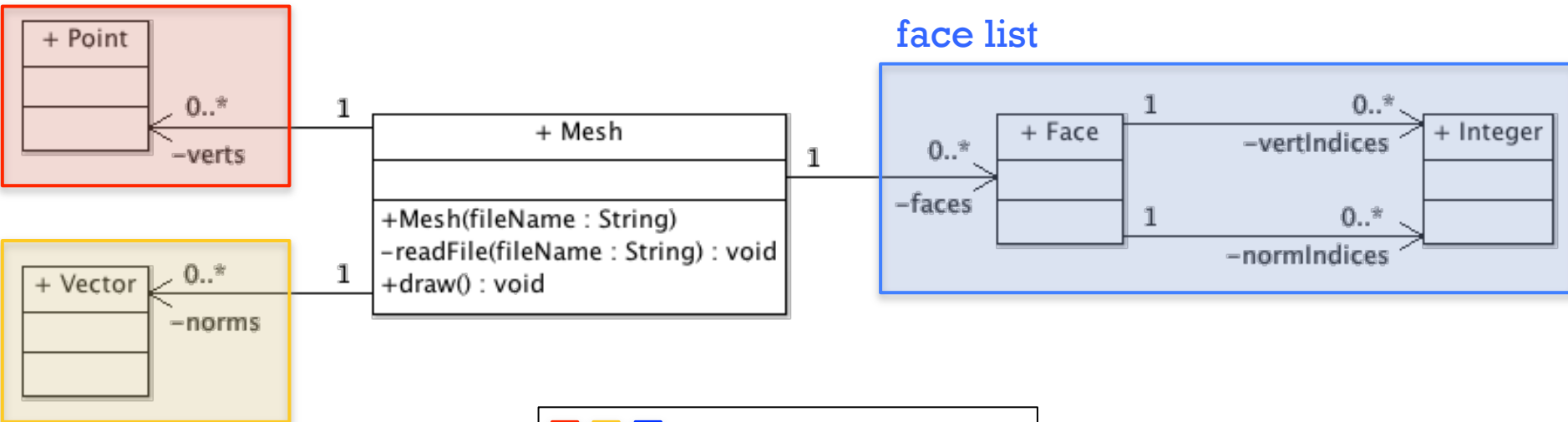
number of faces

```
8 6 6
0 1 1   1 1 1   1 1 0   0 1 0
0 0 1   1 0 1   1 0 0   0 0 0
0 0 1   0 0 -1   0 1 0
-1 0 0   0 -1 0   1 0 0
4 0 4 5 1   0 0 0 0
4 7 3 2 6   1 1 1 1
4 0 1 2 3   2 2 2 2
4 0 3 7 4   3 3 3 3
4 4 7 6 5   4 4 4 4
4 5 6 2 1   5 5 5 5
```

coordinates of vertices

coordinates of normal vectors

faces

face

number of vertices of this face

Indices of the vertices of this face

Indices of the normal vectors at the vertices of this face

vertex list

Final classdiagram

face list

normal list

Final file format

Questions?