U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Definition of Vertices, Indices and Normals

Extra support document

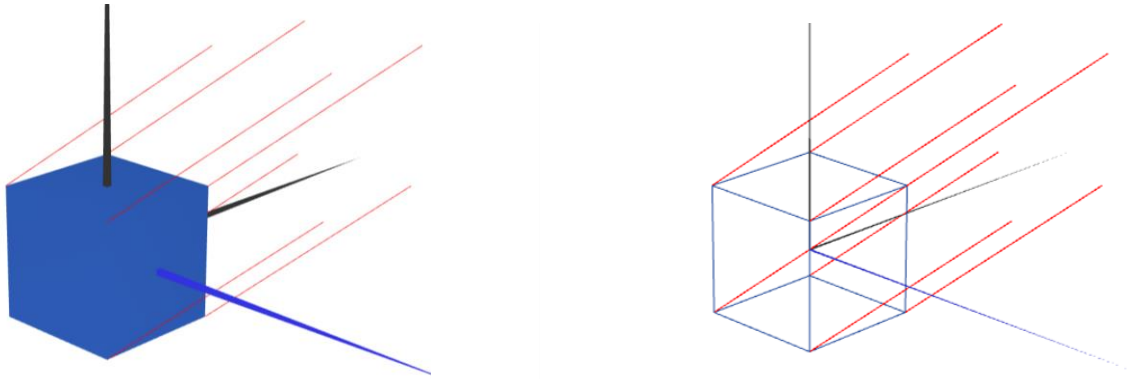
Teresa Matos
3 March 2020

MyUnitCube – without normals

For TP2, the **MyUnitCube** class was created to represent a unitary cube

Without defining normal vectors in **initBuffers()** function, default vectors are used

For each vertex (8 in total), a normal vector $\vec{N} = (1,1,1)$ is defined automatically



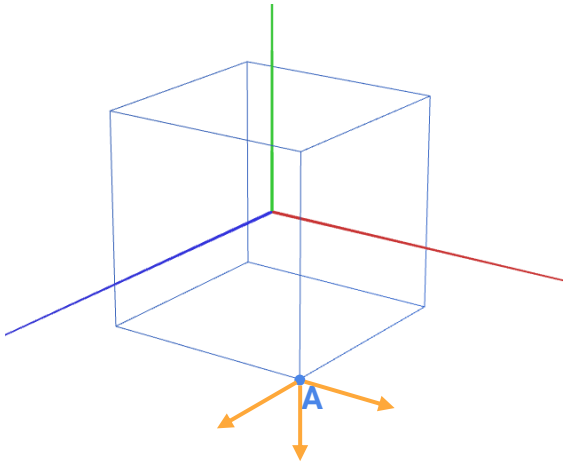
MyUnitCube – With normals

With the default normal vectors, the lighting of MyUnitCube is **not very accurate**

We need to define the normal vectors according to the **different cube faces**

Each vertex is part of **three faces**:

- Vertex A, for example, belongs to the right (X+), front (Z+), and bottom (Y-) face



MyUnitCube – Repeating vertices

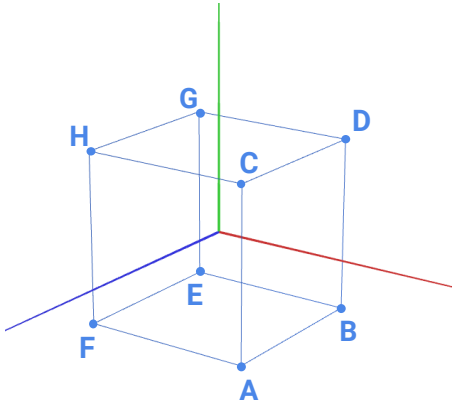
Each vertex must be **defined *X* times** in the `vertices[]` array:

- *X* is the number of faces the vertex appears in

You can **choose any order** to define the vertices

Take into consideration that **this order affects** how indices and normals are defined

In the case of the cube, each vertex must be repeated 3 times



MyUnitCube – Repeating vertices

Some order strategies make it easier/harder to define the other attributes

Here are examples of possible orders to define the vertices of the cube

Group by vertices:

Vertex A

```
vertices = [
  A_right,
  A_front,
  A_bottom,
  B_right,
  B_front,
  …
]
```

Group by faces:

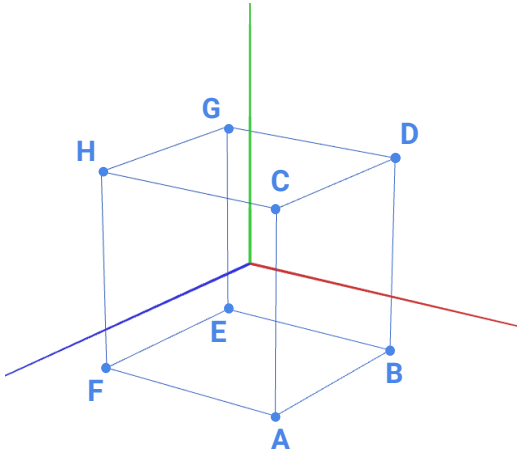
Right Face

```
vertices = [
  A_right,
  B_right,
  C_right,
  D_right,
  B_back,
  ...
]
```

Define all vertices and repeat:

All vertices (A to H)

```
vertices = [
  A_right
  B_right,
  ...,
  H_left,
  A_right,
  B_back,
  ...
]
```



Note: Not all of these orders are very efficient, and there are other possible strategies

MyUnitCube – Redefining indices

The indices are used to define triangles that compose the different faces

The position of each vertex in the array indicates its corresponding index

Using one of the examples from the previous slide:

Group by faces:

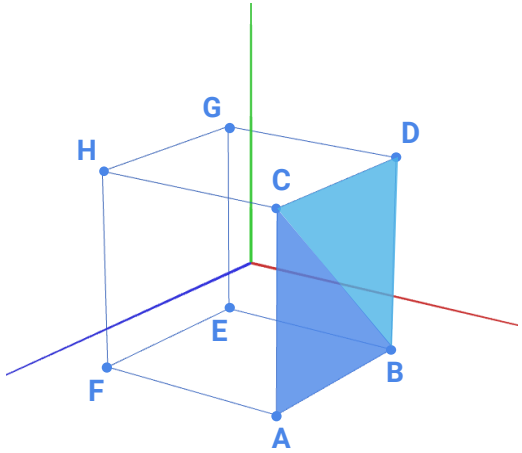
```
vertices = [
0 A_right,
1 B_right,
2 C_right,
3 D_right,
4 B_back,
5 E_back,
6 D_back,
7 G_back,
…
]
```

Indices:

```
indices = [
  [0, 1, 2,
  1, 3, 2,
  [4, 5, 6,
  5, 7, 6,
   …
]
```

Right Face

Back Face



MyUnitCube – Defining normal vectors

The normal vectors are defined in the same order as the vertices

Their value depends on what face the corresponding vertex belongs to

Group by faces:

```
vertices = [
0 A_right,
1 B_right,
2 C_right,
3 D_right,
4 B_back,
5 E_back,
6 D_back,
7 G_back,
…
]
```

Vertex 0 belong to the right face…

Right Face

Back Face

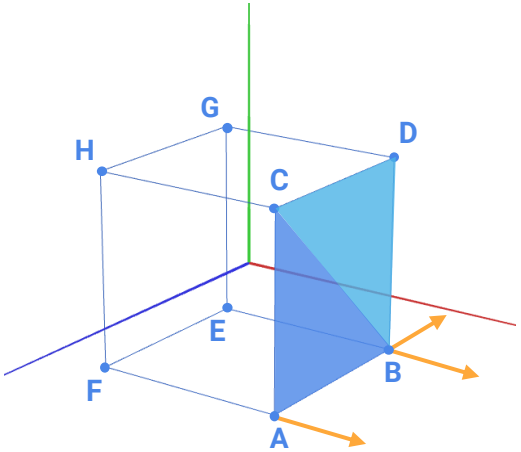
Indices:

```
indices = [
  [0, 1, 2,
  1, 3, 2,
  [4, 5, 6,
  5, 7, 6,
   …
]
```

So normal vector 0 is the right face's normal

Normal vectors

```
normals = [
0 1, 0, 0,
1 1, 0, 0,
2 1, 0, 0,
3 1, 0, 0,
4 0, 0,-1,
5 0, 0,-1,
6 0, 0,-1,
7 0, 0,-1,
  …
]
```



MyUnitCube – Improving code

As you create more complex geometries, the different arrays become longer

There are more efficient ways of filling the arrays without “hardcoding” the values

This is also useful for geometries with variable attributes (e.g.: number of sides)

Try to find patterns in the values, so you can use algorithms to fill the arrays

For example, for the **`indices`** array:

Indices:

```
indices = [
  0, 1, 2,
  1, 3, 2,

  4, 5, 6,
  5, 7, 6,
  …
]
```

Right Face

Back Face

Starting at the first index of each face, and going forward, you may find patterns between the indices different faces.

Try to replace one of the numbers with a variable **`i`** and define the others from that variable (e.g.: **`i+1`**)