Programming 3: Lecture 3

**<u>Vertices, Primitives and Texturing</u>**

A vertex is a coordinate in a 2D or a 3D space. They are the building blocks of "Primitives" in 3D Geometry and are used to create things from a simple cube to a complicated scene constituting of a series of meshes. Primitives are one- or two-dimensional entities or surfaces such as points, lines, and polygons (a flat multishaded shape) that are assembled in a 3D space to create 3D objects.

For example a 3D-cube consists of six two-dimensional squares, each placed on a separate face. Each corner of the square (or any primitive) is called a vertex. Throughout Direct3D, vertices describe position(x, y, z coordinates) and orientation. Each vertex in a primitive is described by a vector that gives its position, colour, texture coordinates, and a normal vector that gives its orientation.

Vertex use
Basically we use Vertices to define shapes. A vertex is a point that defines where two edges of a shape come together. The most common graphics primitive is a triangle.
A triangle for instance has 3 vertices:
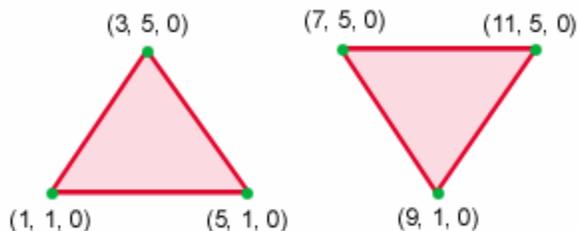
As you can see each of the vertices above have 3 numbers next to them. This is because a vertex is a point it is defined like any other point with X, Y, Z coordinates.

A vertex buffer is therefore a block of memory that contains the vertices needed to draw a shape. For instance to create a triangle you would have to create a vertex buffer that holds the three vertices necessary for the triangle.

In Direct3D though a vertex buffer doesn't just contain the information necessary to define a shape, it can also contain the information necessary for colour, blending, texture coordinates and more.

These values are managed in Direct3D using what is called Flexible Vertex Format (FVF).

We can therefore define these concepts:

Flexible Vertex Format (FVF)
A Flexible Vertex Format or FVF is a format for describing attributes of a vertex. We've already seen three attributes: x value, y value and z value. There are other attributes that we can specify for a vertex, such as, colour and shininess. Using FVFs we can configure which attributes we want to specify for a vertex. When you specify a polygon in Direct3D, the polygon can be filled based on attributes of the vertices. The fill of the polygon is interpolated (blended) between vertices.

Vertex Buffers
A Vertex Buffer is a memory buffer for storing vertices. A vertex buffer can store vertices of any format. Once your vertices are stored in a vertex buffer you can perform operations such as: rendering, transforming and clipping.

Colours
We have been through this before, but just to recap….To represent colours in Direct X, we use the D3DCOLOR_XRGB macro. There are three parameters: Red, Green and Blue. These parameters are integer values between 0 and 255. By specifying different red, green and blue values (mixing colours) you can make any colour you need.

For example:
D3DCOLOR_XRGB(0, 0, 0) is black (no colour)
D3DCOLOR_XRGB(255, 255, 255) is white (full colour)
D3DCOLOR_XRGB(0, 255, 0) is bright green (no red, full green, no blue)
D3DCOLOR_XRGB(100, 20, 100) is dark purple (100 red, 20 green, 100 blue)

Representing our Vertex
In your code we can represent a vertex as a structure, this means that you can customise and alter the structure to have member data based on what is required. Will represent a vertex as the following structure:

```
struct CUSTOMVERTEX
{
    FLOAT x, y, z, rhw;
    DWORD colour;
};
```

This structure defines a data type for our vertices. To actually create a set of vertices that describe a shape we would most likely write something that looks like this:

```
//Store each point of the triangle together with it's colour
CUSTOMVERTEX triangleVertices[] =
{
    {250.0f, 100.0f, 0.5f, 1.0f, D3DCOLOR_XRGB(255, 0, 0),}, //Vertex
    1 - Red (250, 100)
```

```
    {400.0f, 350.0f, 0.5f, 1.0f, D3DCOLOR_XRGB(0, 255, 0),}, //Vertex
    2 - Green (400, 350)
    {100.0f, 350.0f, 0.5f, 1.0f, D3DCOLOR_XRGB(0, 0, 255),}, //Vertex
    3 - Blue (100, 350)
};
```

The numbers represent screen coordinates and are known as transformed
coordinates as they are values that tell Direct3D the exact screen cords where at
which you want the shape drawn.
Don't worry about the 4th value in the structure yet, it is something called the
"reciprocal of homogenous W". It is usually set to one and is used for matrix
mathematics.

What is a Texture
A texture is an image (bitmap usually in our cases) that is applied much like a
decal shape in a 3D world in order to give the shape greater graphical detail.

Transparency
We covered transparency with the Win32 GDI in programming 2.
With Direct3D transparency is a little more complex than telling the computer
what colour we wish to be set to transparent. We now have alpha-values; they
tell Direct3D just how transparent a colour is.

An Alpha value ranges between 0 and 255 just like colours.
The greater the number the greater the opaqueness of the colour, for instance a
pixel of a colour with an alpha-value of 255 will completely overlay any pixel on
which it is drawn.
Any pixels of a colour with alpha value 0 are completely transparent and wont be
drawn at all.

Primitives
A graphics primitive in computer graphics, is a basic element, such as an arc or a
line, that is not made up of smaller parts and that is used to create diagrams and
pictures.
Basically a texture can be applied to any object no matter what the objects size.
This is known as mapping and the objects they are mapped to primitives.

Mapping
Mapping is the process of converting from one type of value to another. An
example is converting texture coordinates to a destination primitive's coordinates.

If you have a large rectangle and you want it completely covered by a texture
then Direct3D will stretch the texture to fit the rectangle. For example the 0th pixel
of the texture is mapped to the 0th pixel of the rectangle and the 512th pixel of the
texture is mapped to the 1024th pixel of the rectangle, the rest of pixel fall
somewhere in between.

Texture coordinates are often expressed as decimal percentages that indicate the portion of the texture to be mapped to the primitive.