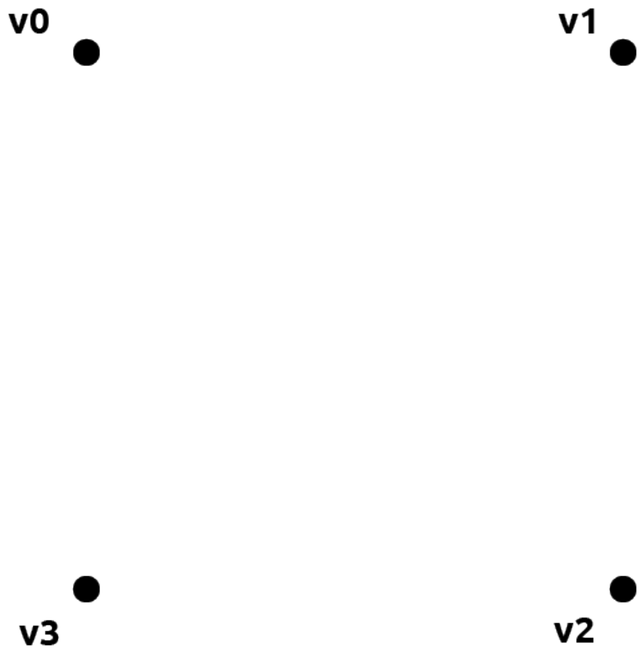
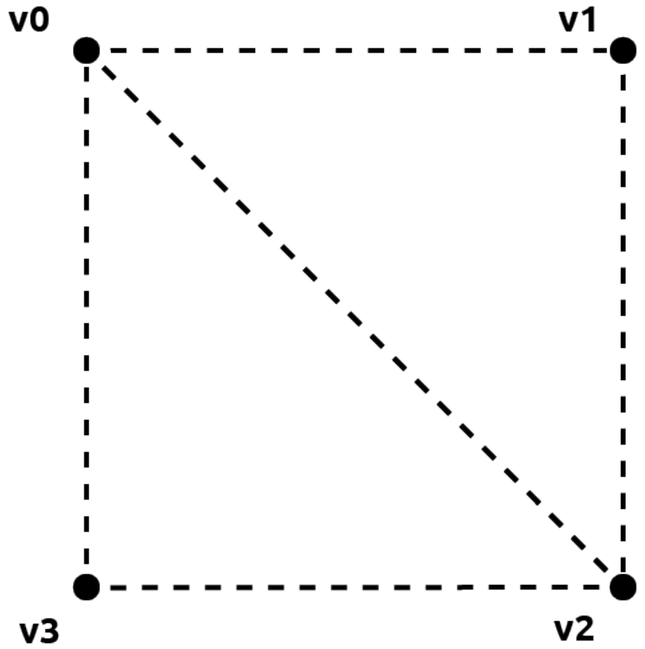
Mesh basics

Carl Emil Carlsen, January 2012



Vertices

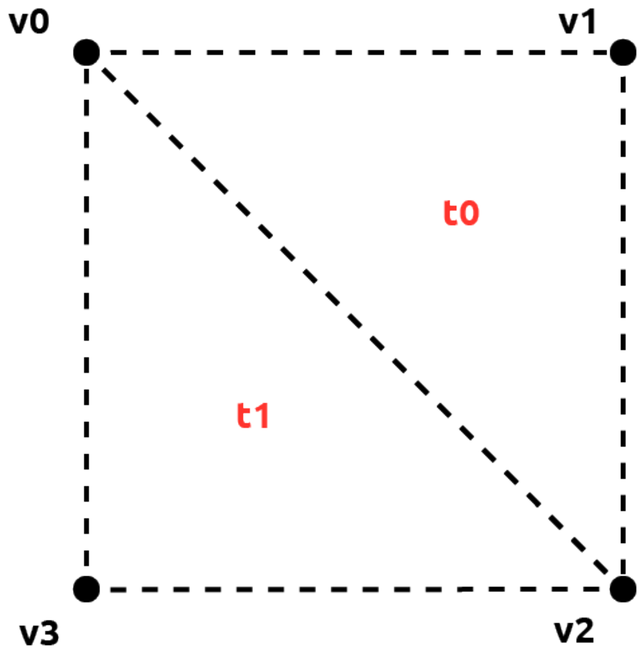
```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);
```



Vertices + Triangles

```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);

// the triangle array defines the draw order of the vertex indexes
int[] triangles = new int[]{ 0, 1, 2, 2, 3, 0 };
```

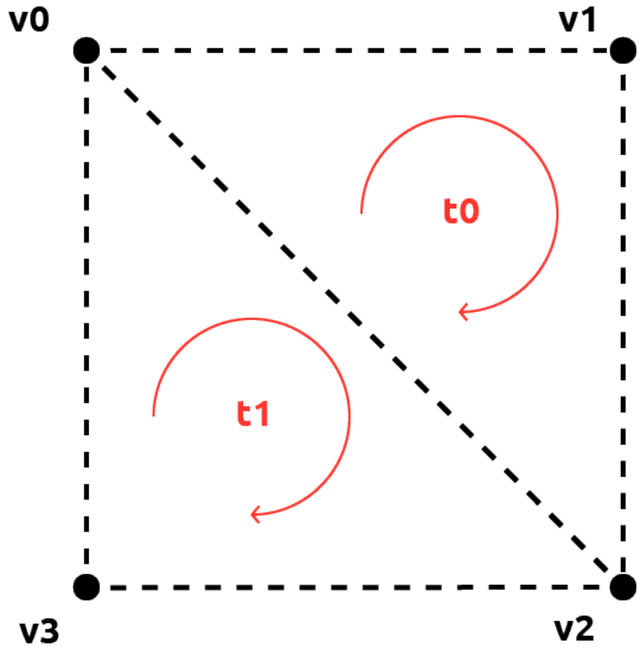


Vertices + Triangles

```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);

// the triangle array defines the draw order of the vertex indexes
int[] triangles = new int[]{ 0, 1, 2, 2, 3, 0 };
```

t0 t1

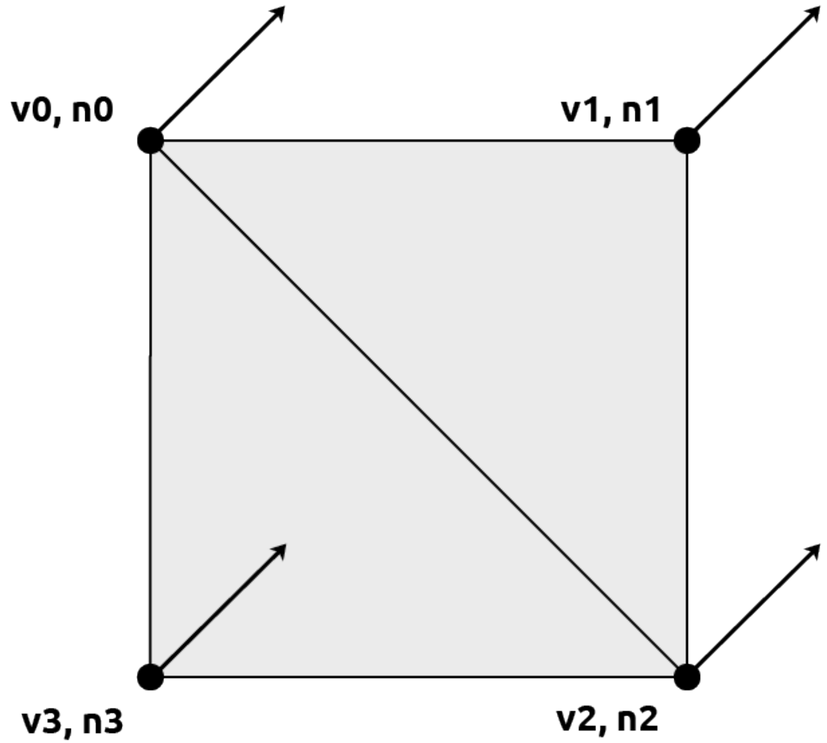


Vertices + Triangles

```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);

// the triangle array defines the draw order of the vertex indexes
int[] triangles = new int[]{ 0, 1, 2, 2, 3, 0 };
                               t0    t1
```

In Unity, draw the corners of each triangles in CLOCKWISE order to make them face you. (the backside is not drawn unless you specfy it in the shader)

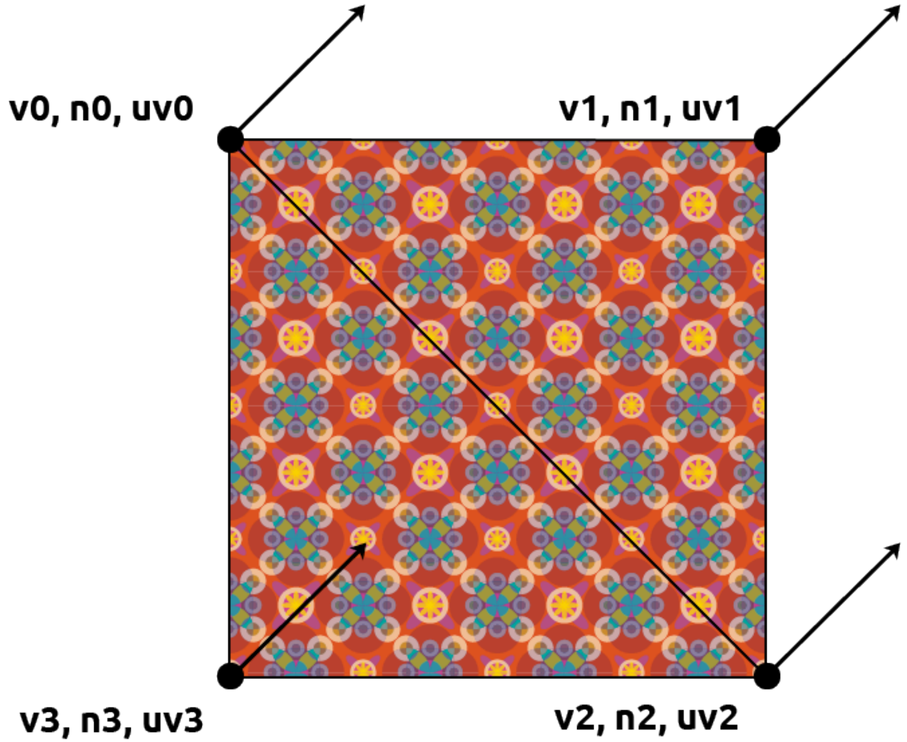


Vertices + Triangles + Normals

```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);

// the triangle array defines the draw order of the vertex indexes
int[] triangles = new int[]{ 0, 1, 2, 2, 3, 0 };

// the normal array defines the forward direction of each point
Vector3[] normals = new Vector3[4];
normals[0] = -Vector3.forward;
normals[1] = -Vector3.forward;
normals[2] = -Vector3.forward;
normals[3] = -Vector3.forward;
```



Vertices + Triangles + Normals + UVs

```
// the vertex array defines the points
Vector3[] vertices = new Vector3[4];
vertices[0] = new Vector3(0,1,0);
vertices[1] = new Vector3(1,1,0);
vertices[2] = new Vector3(1,0,0);
vertices[3] = new Vector3(0,0,0);

// the triangle array defines the draw order of the vertex indexes
int[] triangles = new int[]{ 0, 1, 2, 2, 3, 0 };

// the normal array defines the forward direction of each point
Vector3[] normals = new Vector3[4];
normals[0] = -Vector3.forward;
normals[1] = -Vector3.forward;
normals[2] = -Vector3.forward;
normals[3] = -Vector3.forward;

// the uv array defines the texture coordinate for each point
Vector2[] uvs = new Vector2[4];
uvs[0] = new Vector2(0,1);
uvs[1] = new Vector2(1,1);
uvs[2] = new Vector2(1,0);
uvs[3] = new Vector2(0,0);
```