

XNA Drawing

305890
Spring 2011
3/21/2011
Kyoung Shin Park

Drawing

- Vertex Buffer & Index Buffer
 - Creating vertex buffer & index buffer
 - Accessing vertex & index buffer memory
 - Getting vertex buffer & index buffer information
- Render State
- Drawing Preparations
 - Vertex buffer drawing
 - Vertex buffer & index buffer drawing
 - Example
- Geometry Object

Vertex Buffer / Index Buffer

- Vertex buffer & Index buffer
 - Vertex buffer is simply a chunk of contiguous memory that contains vertex data
 - Index buffer is a chunk of contiguous memory that contains index data
- Creating a vertex buffer
 - This example creates a static vertex buffer that has enough memory to hold 8 vertices of Vertex type:

```
VertexPositionColor[] vertices = new VertexPositionColor[8];
vertices[0] = new VertexPositionColor(new Vector3(), Color.White);
//...
VertexBuffer vertexBuffer = new VertexBuffer(GraphicsDevice,
    VertexPositionColor.SizeInBytes*(vertices.Length), BufferUsage.None);
vertexBuffer.SetData<VertexPositionColor>(vertices);
```

Vertex Buffer / Index Buffer

- Creating an index buffer
 - This example shows how to create a dynamic index buffer that has enough memory to hold 6 short-type indices:

```
short[] indices = new short[6];
for (int i=0; i<6; i++)
{
    indices[i * 2] = (short)(i);
    indices[(i * 2) + 1] = (short)(i + 1);
}
```

Render State

- Render state
 - "SetRenderState" is used to specify rendering states other than default value
 - Enum of many state variables about 100

```
GraphicsDevice.SetRenderState.PointSize = 10;
```

```
// to draw wireframe mode rendering  
GraphicsDevice.SetRenderState.FillMode = FillMode.Wireframe;
```

```
// to draw solid fill mode rendering  
GraphicsDevice.SetRenderState.FillMode = FillMode.Solid;
```

Vertex Declaration

- Vertex Declarations
 - We need to create a vertex declaration to describe the format of the vertex we are using.

```
VertexDeclaration vertexDeclaration = new VertexDeclaration(  
    GraphicsDevice,  
    VertexPositionColor.VertexElements  
);
```

Vertex Buffer Drawing

- DrawUserPrimitives
 - This method is used to draw primitives that do not use index
- ```
GraphicsDevice.DrawUserPrimitives<T>(
 PrimitiveType primitiveType, // primitive type
 T[] vertexData, // vertex data
 int startVertex, // index to an element in the vertex
 // buffer for starting point
 int primitiveCount // number of primitives to draw
);

// draw 4 triangles
GraphicsDevice.DrawUserPrimitives<VertexPositionColor>(
 PrimitiveType.TriangleList, vertices, 0, 4);
```

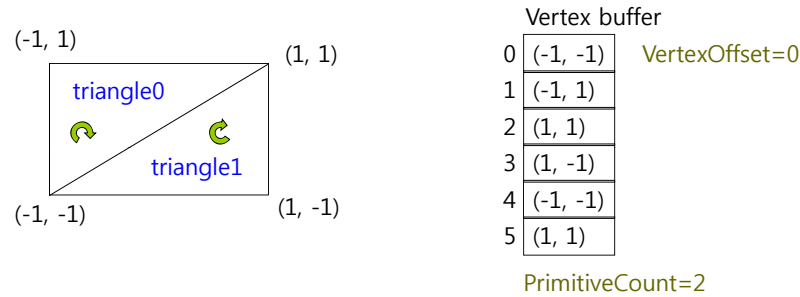
## Vertex/Index Buffer Drawing

---

- DrawUserIndexedPrimitives
- ```
GraphicsDevice.DrawUserIndexedPrimitives<T>(  
    PrimitiveType primitiveType,    // primitive type  
    T[] vertexData,                // vertex data  
    int vertexOffset,              // offset (in bytes) from the beginning of the vertex  
                                    // buffer to the first vertex to draw  
    int numVertices,              // number of vertices to draw  
    short[] indexData,            // index data  
    int indexOffset,              // offset (in bytes) from the beginning of the first  
                                    // index to use  
    int primitiveCount            // number of primitives to draw  
);  
  
// draw a geometry consisting of 12 triangles and 8 vertices  
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(  
    PrimitiveType.TriangleList, vertices, 0, 8, indices, 0, 12);
```

Drawing Example

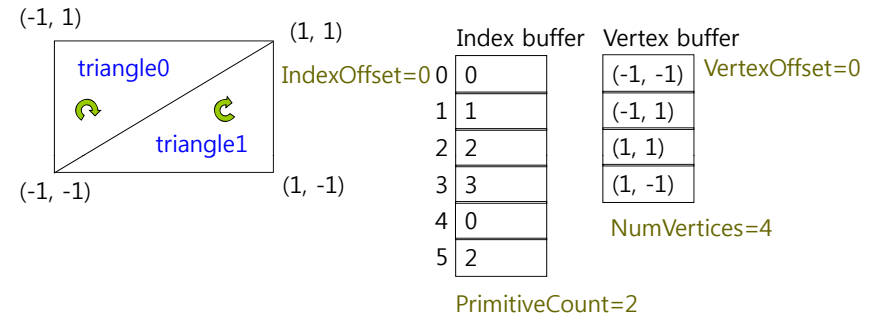
- Draw 2 triangles using DrawUserPrimitives



```
GraphicsDevice.DrawUserPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleList, vertices, 0, 2);
```

Drawing Example

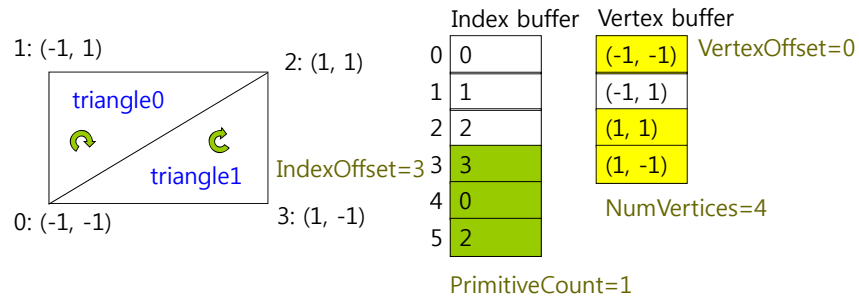
- Draw 2 triangles using DrawUserIndexedPrimitives



```
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleList, vertices, 0, 4, indices, 0, 2);
```

Drawing Example

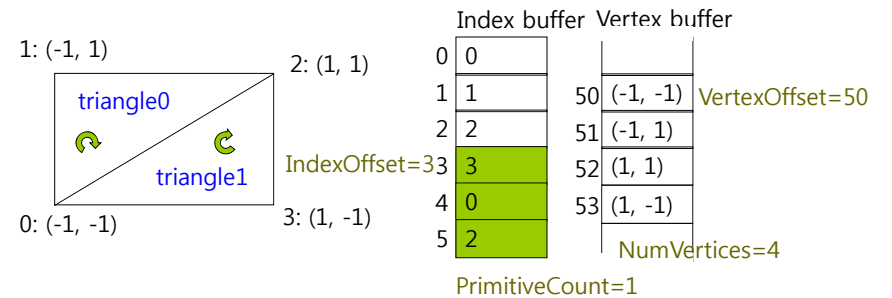
- Draw 1 triangle (i.e., 2nd one) specifying IndexOffset in DrawUserIndexedPrimitives



```
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleList, vertices, 0, 4, indices, 3, 1);
```

Drawing Example

- Draw 1 triangle specifying VertexOffset in DrawUserIndexedPrimitives



```
GraphicsDevice.DrawUserIndexedPrimitives<VertexPositionColor>(
    PrimitiveType.TriangleList, vertices, 50, 4, indices, 3, 1);
```

BeginScene / EndScene

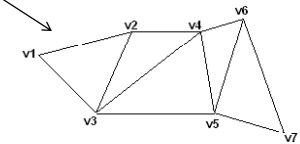
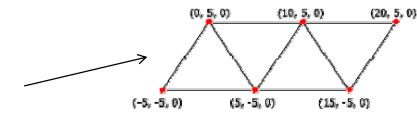
- Drawing methods must always be called inside effect Begin and End pair.

```
effect.Begin();
foreach(EffectPass pass in effect.CurrentTechnique.Passes)
{
    pass.Begin();
    ...
    GraphicsDevice.DrawUserPrimitives<VertexPositionColor>( ... );
    ...
    pass.End();
}
effect.End();
```

Primitive Types

- Some primitive types are

- PointList
- LineList
- LineStrip
- **TriangleList**
- **TriangleStrip**
- TriangleFan



v1 v2 v3 v4 v5 v6 v7
DrawUserPrimitives(TriangleStrip, vertices, 0, 5)

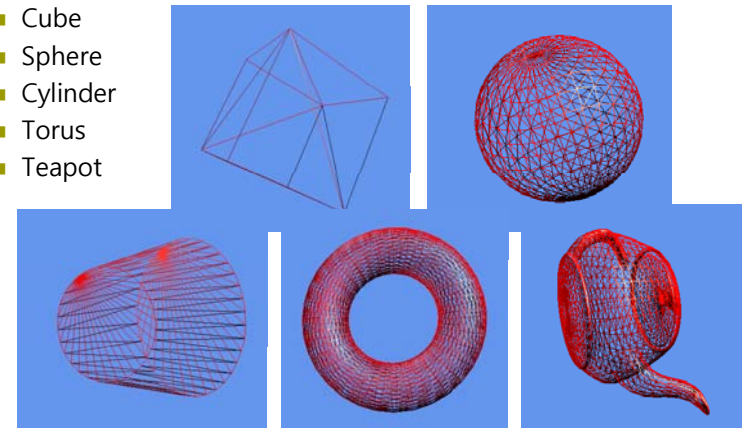
Primitive Types

```
namespace Microsoft.Xna.Framework.Graphics
{
    public enum PrimitiveType {
        PointList = 1,
        LineList = 2,
        LineStrip = 3,
        TriangleList = 4,
        TriangleStrip = 5,
        TriangleFan = 6,
    }
}
```

3D Geometry Object

- XNA basic 3D geometric primitive objects:

- Cube
- Sphere
- Cylinder
- Torus
- Teapot



http://create.msdn.com/en-US/education/catalog/sample/primitives_3d