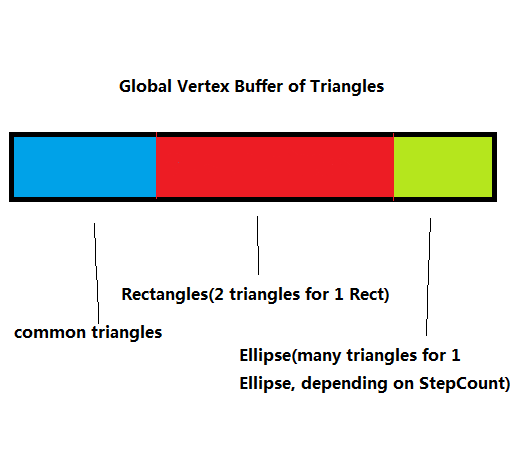
* **NoiseGraphicObject**

1. (2015.8.2)这是一个专门存线条的buffer，本来是打算直接在renderer那存个两顶点的vertex buffer，每次调用render 的时候就更新数据到GPU。 有AddToRenderList，在这个函数里会检查是否需要更新到GPU缓冲区，并调整GPU缓冲区的大小。
2. (2015.8.30)现在简单基本的几何对象都用这个类来储存（避免即时更新到GPU的速度下降）。暂时有几种类型，Line3D/2D，Point3D/2D，Triangle2D，Rectangle，每种类型都有自己的状态变量和Vertex Buffers。
3. (2015.8.30)Triangle2D可以用来装圆，粗线条，切角长方形，正方形，贴图长方形...等复杂一点点的的对象，这就要开辟多一些对象信息列表来给Triangle2d的Vertex buffer分区。
4. (2015.9.5)现在有了实心矩形和椭圆的渲染（还没贴上图）。有一些特殊的列表，专门记录line / triangle buffer特殊的部分，因为例如长方形或者椭圆这种由一堆三角形组成的图形，需要知道对应的三角形id段，暂时用startID和elementCount来表示。普通三角形的ID会出现在一个vector<UINT>里面。
5. (2015.9.5)在AddToRenderList的时候，需要更新才进行updatesubresouce()；
6. (2015.9.6)现在有了矩形和椭圆线框。要注意Line2D和Triangle2D的vertex buffer比较特殊，有一定的分区，如下图。
7. (2015.9.16)Vertices的更新都在renderer的AddToRenderList()里面。
8. (2015.9.16)普通(Common) Triangles是不带贴图的，长方形和椭圆可以带贴图。
9. (2015.11.10)作为文本材质的载体，Rectangle可以设置纹理坐标（专门为动态文本的渲染而设置）
10. (2015.12.1)Graphic Object类最近进行了一次重构，删掉了没什么用的图形(Ellipse,Outlines)，吧Add/Set/Delete Vertices抽象了出来，用了std::initializer\_list避免了同样逻辑的功能因为参数个数不同导致的代码复制。
11. (2015.12.1)因为经过结构调整以后，支持的图形只有solid基本图形和textured rect。对于textureRect，有一个专门列表储存每个Rect对应的(UINT)texID，这个在AddToRenderList的时候可以生成一个SubsetList，跟mesh的subset一样，一个subset对应一个draw call，这样就不用一个矩形一个draw call，那样极为浪费。