

AnyCAD 三维控件用户手册

v2015.UR1

www.anycad.net

目录

1. 简介	4
1.1. 授权	4
1.2. 下载	4
1.3. 技术支持	4
2. 功能介绍	5
2.1. 数学运算	5
2.2. 几何造型	5
2.3. 文件交换	6
2.4. 三维显示	6
3. .Net 入门教程	7
3.1. 添加 DLL 模块引用	7
3.2. 添加显示控件	8
3.3. 视频教程	8
3.4. 三维建模	9
3.4.1. 建模 API	9
3.4.2. 模型显示	9
3.4.3. 练习	10
4. C++入门教程	10
4.1. 项目设置	10
4.2. MFC 相关基类修改	11
4.3. 第一个模型	11
5. 深入浅出	12
5.1. 几何建模	12
5.1.1. 拓扑结构	12
5.1.2. 几何属性	12
5.2. 显示引擎	13
5.2.1. 显示设置	13
5.3. 场景管理	15
5.3.1. 显示几何对象	15

5.3.2.	设置物体的显示样式	16
5.4.	鼠标拾取.....	16
5.4.1.	拾取命令	16
5.4.2.	选择对象查询	16
6.	示例	17
7.	参考资料.....	23

1. 简介

AnyCAD 三维图形控件是一个功能强大，简单易用的建模和显示控件，包括三维实体建模、三维可视化和数据交换模块，支持 .Net Framework 和 C++。 .Net 版本提供继承于 .Net 的 User Control 的三维显示控件，可以方便地与 WinForm/WPF 的应用程序集成。 C++ 版本支持 MFC/Qt/wxWidgets 等界面库。

1.1. 授权

- 试用

AnyCAD .Net SDK 提供 30 天免费试用。试用期满，请自行删除 SDK。

试用期间，您可以在我们的合作问答论坛提问问题：

<http://www.opencascade.net/ask/?/topic/anycad>

- 授权

若您的企业有 3 个程序员使用 AnyCAD.Net SDK，则您需要购买 3 个授权。

- 服务

购买正式版后，您可以得到半年免费的技术支持。

- 程序发布

您购买之后，发布 EXE 程序不收取任何费用，但若发布的是 SDK，则每出售一份 SDK 都需要购买授权。

购买咨询：

QQ: 1323461552

E-Mail: anycad@anycad.net

1.2. 下载

.Net/C++ SDK 试用版下载: <http://www.anycad.net/sdk/>

1.3. 技术支持

技术问答: <http://www.opencascade.net/ask/?/topic/anycad>

API 帮助文档: <http://www.anycad.net/help>

2. 功能介绍

AnyCAD SDK 提供的功能按照模块从底层到上层分为以下几大模块：

- Common：基本数学库
- Geometry：几何造型、模型读写
- Render：显示引擎相关
- View：视图，命令等
- Document：文档管理

注：Document 模块不包含在专业版中。

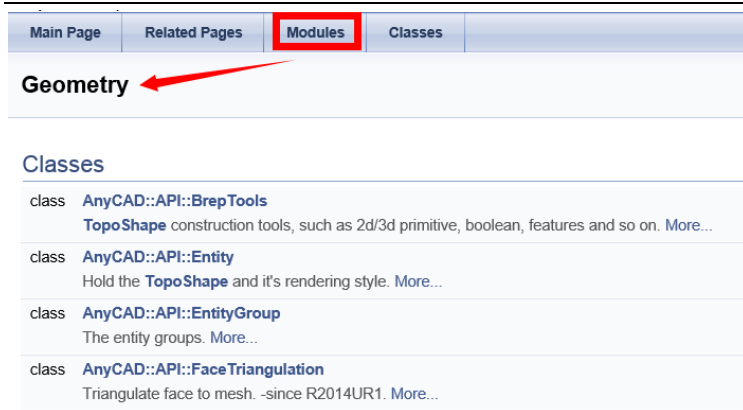
2.1. 数学运算

包含在 API 文档中的 Common 模块下。包括向量、矩阵等基本的数学工具。

Main Page	Related Pages	Modules	Classes
Common			
Classes			
class AnyCAD::API::Vector2 Point or vector. More...			
class AnyCAD::API::Vector3 Point or vector. More...			
class AnyCAD::API::Vector3List Point of vector list. More...			
class AnyCAD::API::ColorValue Color Value. More...			
class AnyCAD::API::Matrix3 Matrix3 . More...			
class AnyCAD::API::Matrix4 Matrix4 . More...			

2.2. 几何造型

包含在 API 文档中的 Geometry 模块下。



三维建模有以下功能：

- 三维基本体，如点、线、多折线、Spline 曲线、圆、椭圆、圆弧、球、长方体（立方体）、圆柱、圆锥等
- 复合体：复合线(wire)、复合体等
- 三维特征操作，如拉伸、扫掠、回转、放样、蒙面等
- 布尔操作：交、叉、并

2.3. 文件交换

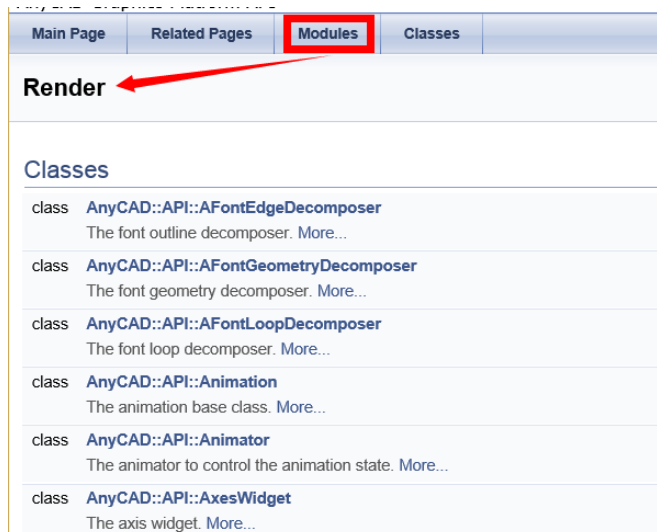
包含在 API 文档中的 Geometry 模块下。

标准 SDK 支持以下格式的读取和保存：

- STEP
- IGES
- STL

2.4. 三维显示

包含在 API 文档中的 Render 模块下。



SDK 提供 WinForm 控件，可以方便的添加到窗体中。

主要功能：

- 标准视图操作：缩放、旋转、平移
- 显示节点管理：添加、删除、根据 Id 查询
- 实体模型显示及其显示风格设置
- 鼠标拾取
- 选择对象过滤

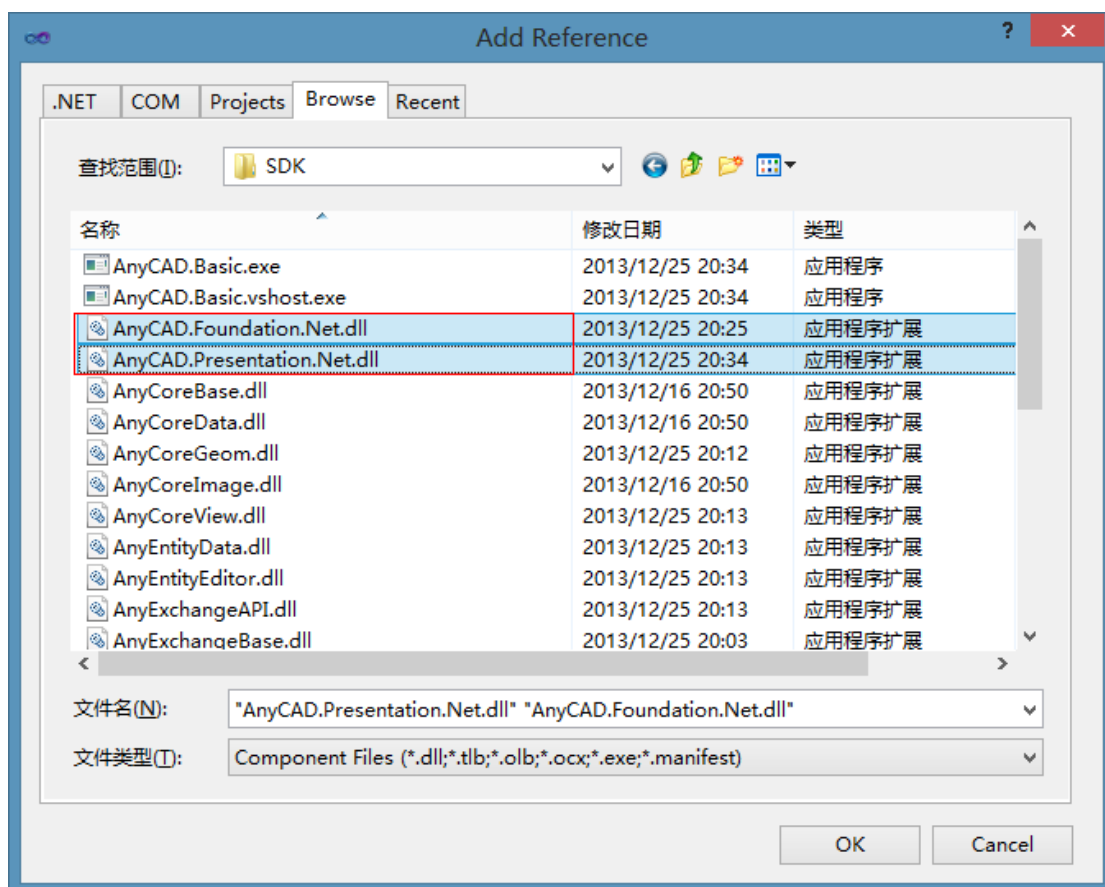
3. .Net 入门教程

3.1. 添加 DLL 模块引用

新建或在已有的 C# 工程中加载 DLL 引用。

选择加载引用 AnyCAD.SDK bin 目录下的程序集：

- AnyCAD.Foundation.Net.dll
- AnyCAD.Presentation.Net.dll:



注意：请把 SDK 下所有的文件复制到 exe 程序的目录下。

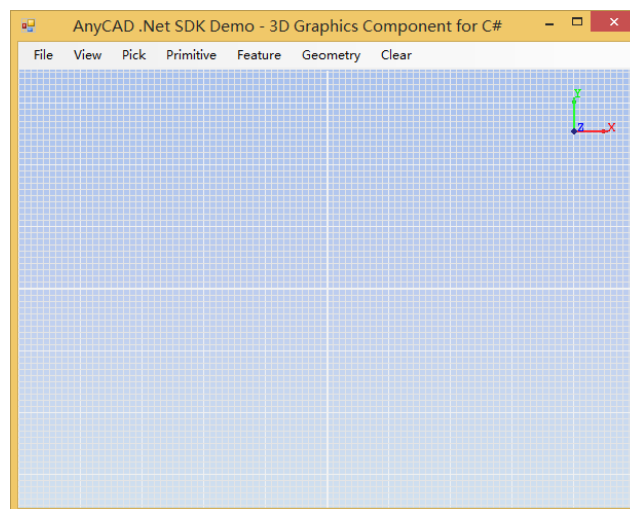
3.2. 添加显示控件

```
public partial class FormMain : Form
{
    // Render Control
    Private AnyCAD.Presentation.RenderWindow3d renderView;

    public FormMain()
    {
        InitializeComponent();

        //
        // Create renderView
        //
        this.renderView = new AnyCAD.Presentation.RenderWindow3d();
        this.renderView.Location = new System.Drawing.Point(0, 27);
        this.renderView.Size = this.Size;
        this.renderView.TabIndex = 1;
        this.Controls.Add(this.renderView);
    }
}
```

运行程序，显示类似下图的三维窗口：



3.3. 视频教程

参考在线视频教程引用 AnyCAD .Net SDK 程序集：

http://v.youku.com/v_show/id_XNzExNDg3ODg4.html

3.4. 三维建模

3.4.1. 建模 API

三维建模 API 由 BrepTools 类实现，每个方法都返回 **TopoShape** 对象。
详细的 API 参考 AnyCAD API 文档。

TopoShape	表示一个几何对象的拓扑结构。
BrepTools	构造 TopoShape 的方法类。汇集各种造型算法，如拉伸、扫略、2D/3D 基本体等。

Main Page
Related Pages
Modules
Classes

Class List
Class Index
Class Hierarchy
Class Members

AnyCAD::API::BrepTools Class Reference
Public Me

Geometry

TopoShape construction tools, such as 2d/3d primitive, boolean, features and so on. [More...](#)

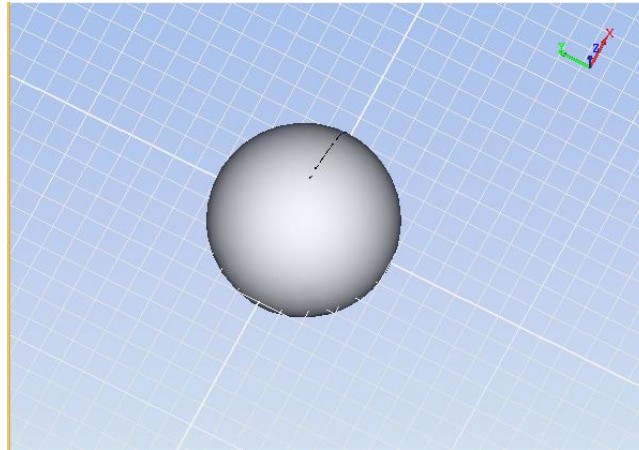
Public Member Functions

TopoShape	LoadFile (const String &strFile)
bool	SaveFile (const TopoShapeGroup &shapeGroup, const String &strFileName)
String	ToJson (const TopoShapeGroup &meshGroup, Real precision)
TopoShape	MakePoint (const Vector3 &position)
TopoShape	MakeLine (const Vector3 &start, const Vector3 &end)
TopoShape	MakeCircle (const Vector3 ¢er, double radius, const Vector3 &dir)
TopoShape	MakeEllipse (const Vector3 ¢er, double majorRadius, double minorRadius, const Vector3 &dir)
TopoShape	MakeArc (const Vector3 &start, const Vector3 &end, const Vector3 ¢er, const Vector3 &dir)
TopoShape	MakeArc (const Vector3 ¢er, Real radius, Real startAngle, Real endAngle, const Vector3 &dir)
TopoShape	MakeArc3Pts (const Vector3 &start, const Vector3 &end, const Vector3 &middle)

3.4.2. 模型显示

由 BrepTools 创建的 TopoShape 对象，转换为 SceneNode 就可以在三维窗口中显示。

```
//在原点创建一个半径为40的球
TopoShape sphere = GlobalInstance.BrepTools.MakeSphere(new Vector3(0, 0, 0), 40);
//显示物体， shapeId可以用来在场景中查找对象
SceneNode sphereNode = renderView.ShowGeometry(sphere, ++shapeId);
```



SceneNode	三维场景可以显示节点的基类。从SceneNode继承了若干的子类
RenderStyle	设置物体显示样式的基类。利用其子类，可以分别设置SceneNode的点、线、面的显示效果，如线的粗细、颜色等。

3.4.3. 练习

使用 BrepTools 分别创建立方体、圆柱体，进行布尔运算并显示出来。

4. C++入门教程

以 MFC 中的 Document-View 应用使用 AnyCAD C++ SDK 为例。

4.1. 项目设置

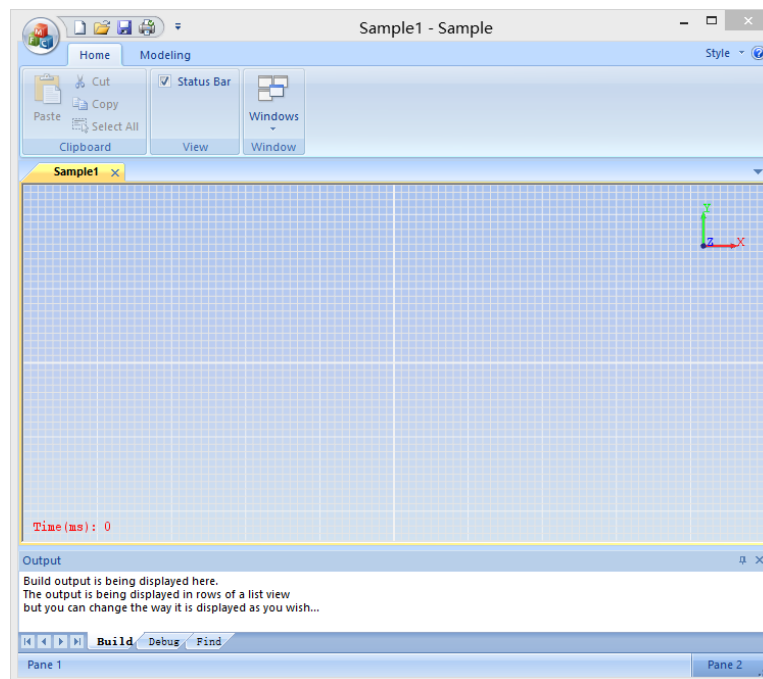
\$(AnyCADSDK)替换为您的工程相对于 AnyCAD C++ SDK 的位置，您也可以把 AnyCADSDK 加到环境变量中。

- 头文件
\$(AnyCADSDK)\inc
- Lib 查找目录
\$(AnyCADSDK)\lib\\$(Configuration)\
- 依赖的 lib
AnyPlatformAPI.lib; AnyPlatformMFC.lib
- 在stdafx.h中包含
`#include "View/AnySDK.h"`

4.2.MFC 相关基类修改

- CWinApp/CWinAppEx
把以上的类替换为 AuWinApp
- CView
把需要三维显示的类替换为AuRenderView3d
通过访问AuRenderView3d的m_View3d成员可以获取显示引擎相关的对象。

编译运行程序，您可以得到类似如下的三维视图，并且具备旋转、移动、缩放等基本操作：



4.3. 第一个模型

创建球体对应的处理函数：

```
void CSampleView::OnButtonSphere()
{
    using namespace AnyCAD::API;
    auto sphere = BrepTools().MakeSphere(Vector3(0,0,0),100);
    ShowGeometry(sphere, 101, 1);
}
```

5. 深入浅出

5.1. 几何建模

5.1.1. 拓扑结构

TopoShape 用来统一表示拓扑结构，可以是点、边、面、体等。通过 TopoShape::GetShapeType 可以得到具体的类型：

```
enum EnumTopoShapeType
{
    Topo_COMPOUND,
    Topo_COMPSOLID,
    Topo_SOLID,
    Topo_SHELL,
    Topo_FACE,
    Topo_WIRE,
    Topo_EDGE,
    Topo_VERTEX,
    Topo_SHAPE
};
```

使用类TopoExplor可以遍历拓扑的子结构，如包含的点、边、面等。

5.1.2. 几何属性

TopoShapeProperty 用来得到几何的属性，如：

- 边的长度
- 面的面积
- 体的体积

计算曲线的长度

```
TopoShapeProperty property = new TopoShapeProperty();
property.SetShape(arc);
float len = property.EdgeLength();
```

GeomCurve 用来得到曲线上的信息，如曲线参数、点、曲率等。

计算曲线上对应点的法线

```
GeomCurve curve = new GeomCurve();
curve.Initialize(arc);
```

```
float paramStart = curve.FirstParameter();
float paramEnd = curve.LastParameter();

float step = (paramEnd - paramStart) * 0.1f;

for (float uu = paramStart; uu <= paramEnd; uu += step)
{
    Vector3 dir = curve.DN(uu, 1);
    Vector3 pos = curve.Value(uu);
}
```

GeomSurface 用来得到曲面上的信息，如 uv 参数、点、法线、曲率等。

计算曲面上参数域点与该点的法线

```
GeomSurface surface = new GeomSurface();
surface.Initialize(face);
float ufirst = surface.FirstUParameter();
float uLarst = surface.LastUParameter();
float vfirst = surface.FirstVParameter();
float vLast = surface.LastVParameter();

float ustep = (uLarst - ufirst) * 0.1f;
float vstep = (vLast - vfirst) * 0.1f;
for(float ii=ufirst; ii<=uLarst; ii+= ustep)
    for (float jj = vfirst; jj <= vLast; jj += vstep)
    {
        Vector3List data = surface.Dl(ii, jj);

        Vector3 pos = data.Get(0);
        Vector3 dirU = data.Get(1);
        Vector3 dirV = data.Get(2);
        Vector3 dir = dirV.CrossProduct(dirU);
        dir.Normalize();
    }
```

5.2. 显示引擎

5.2.1. 显示设置

通过 Renderer 可以设置显示模式、网格平面控制、标准视图控制等。

Renderer 对象在创建显示控件的时候自动创建，通过控件的 View3d 属性可以得到。

获取 Renderer 对象
View3d::GetRenderer()

注：设置后调用 `Renderer::RequestDraw(1)`刷新视图。

- 标准视图设置

`Renderer::SetStandardView(int viewType)`

标准视图

```
enum EnumStandardView
{
    SV_Invalid,
    SV_Front, /**< Front View. */
    SV_Back, /**< Back View. */
    SV_Top, /**< Top View. */
    SV_Bottom, /**< Bottom View. */
    SV_Left, /**< Left View. */
    SV_Right, /**< Right View. */
    SV_ISO /**< XYZ View. */
};
```

- 平面网格显示控制

`Renderer::ShowWorkingGrid(bool)`

- 显示模式控制

`Renderer::SetDisplayMode(int mode)`。通过组合以下枚举，可以控制显示效果。

```
enum EnumDisplayStyle
{
    DS_Face          = 0x0001,
    DS_Wireframe     = 0x0002,
    DS_Realistic     = 0x0004,

    DS_Vertex        = 0x0008,
    DS_Edge          = 0x0010,

    DS_ShadeEdge     = DS_Face | DS_Edge,
};
```

- 选择器过滤

`Renderer::SetPickMode(int mode)`

可以控制选择模式，分别选择点、边、面或者节点

```
enum EnumRenderFlags
```

```
{
    RF_None      = 0,
    RF_Face      = 0x0001,
    RF_Edge      = 0x0002,
    RF_Vertex    = 0x0004,
    RF_Text      = 0x0008,
    RF_BBox      = 0x0010,
    RF_HlrEdge   = 0x0020,
    RF_SceneNode = 0x0100,
    RF_GroupSceneNode = 0x0200,
    RF_Default   = RF_Face | RF_Edge | RF_Vertex | RF_Text,
};
```

5.3. 场景管理

SceneManager 用来管理 SceneNode 对象，通过 Renderer 可以得到 SceneManager 对象。只有加入到 SceneManager 中的 SceneNode 才能在显示。显示控件中的 RenderWindow3d::ShowGeometry 和 RenderWindow3d::ShowSceneNode 方法封装了显示几何对象和场景节点对象。

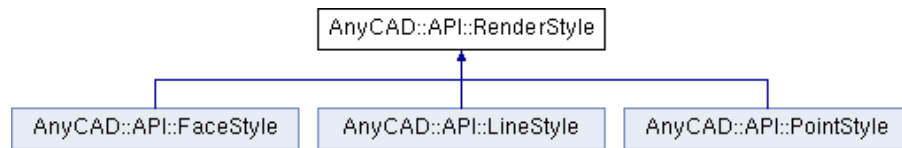
5.3.1. 显示几何对象

TopoShape 本身不能显示，需要转化成 SceneNode 才能在场景中显示。TopoShapeConvert 是 TopoShape 对象和 SceneNode 对象之间转换的桥梁。在转换过程中需要制定显示精度。数值越小，精度越高。如模型比较大，建议精度设置的低些，以提高显示效率。

C#	<pre>TopoShape cylinder = GlobalInstance.BrepTools.MakeCylinder(Vector3.ZERO, Vector3.UNIT_Z, 50, 100, 0); // 0.1f is the precision for visualization. EntitySceneNode sceneNode = GlobalInstance.TopoShapeConvert.ToEntityNode(cylinder, 0.1f)</pre>
C++	<pre>TopoShape geometry = ... static TopoShapeConvert tool; EntitySceneNode sceneNode = tool.ToEntityNode(geometry, accurate);</pre>

5.3.2. 设置物体的显示样式

每个显示对象支持单独设置点、线、面的样式



给面设置透明颜色（C#）

```
// Set transparent face
FaceStyle style = new FaceStyle();
style.SetColor(new ColorValue(0.5f, 0.3f, 0, 0.5f));
style.SetTransparent(true);
sceneNode.SetFaceStyle(style);
```

5.4. 鼠标拾取

5.4.1. 拾取命令

鼠标拾取分为点选和框选两个命令

启动点选：View3d::ExecuteCommand("Pick")

启动框选：View3d::ExecuteCommand("RectPick");

注：按住Shift键，实现多选。

5.4.2. 选择对象查询

获取选择的对象有两种方法：

- 被动获取
当选择命令选择物体的时候，发出通知。
- 主动获取
主动从场景中查询选择的物体。

5.4.2.1. 被动获取实现

C++

从EventListener继承，重载其OnSelectElement(ElementIdSet ids)即可收到选择通知。

```
void CSampleView::OnSelectElement(const AnyCAD::API::ElementIdSet& ids)
{
```



```
if(ids.size() > 0)
{
    CString str;
    str.Format(_T("Selected: %d"), *ids.begin());
    AfxMessageBox(str);
}
}
```

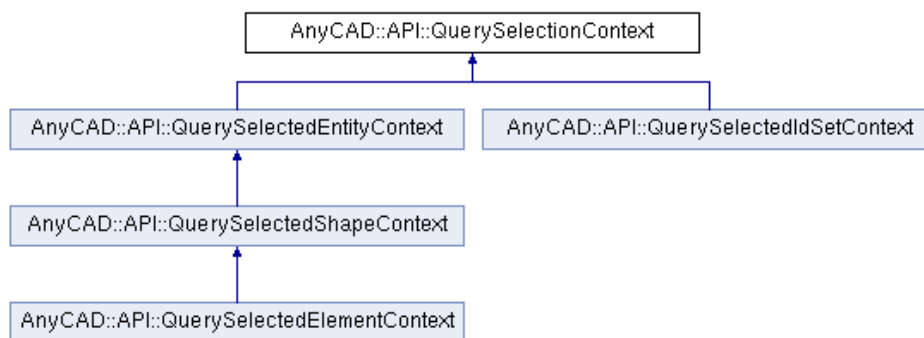
C#

利用C#的事件机制，可以接受选择通知

```
GlobalInstance.EventListener.OnSelectElement += OnSelectionEvent;
```

5.4.2.2. 主动获取实现

通过构造 QuerySelectionContext 的子类对象查询场景中选择物体。

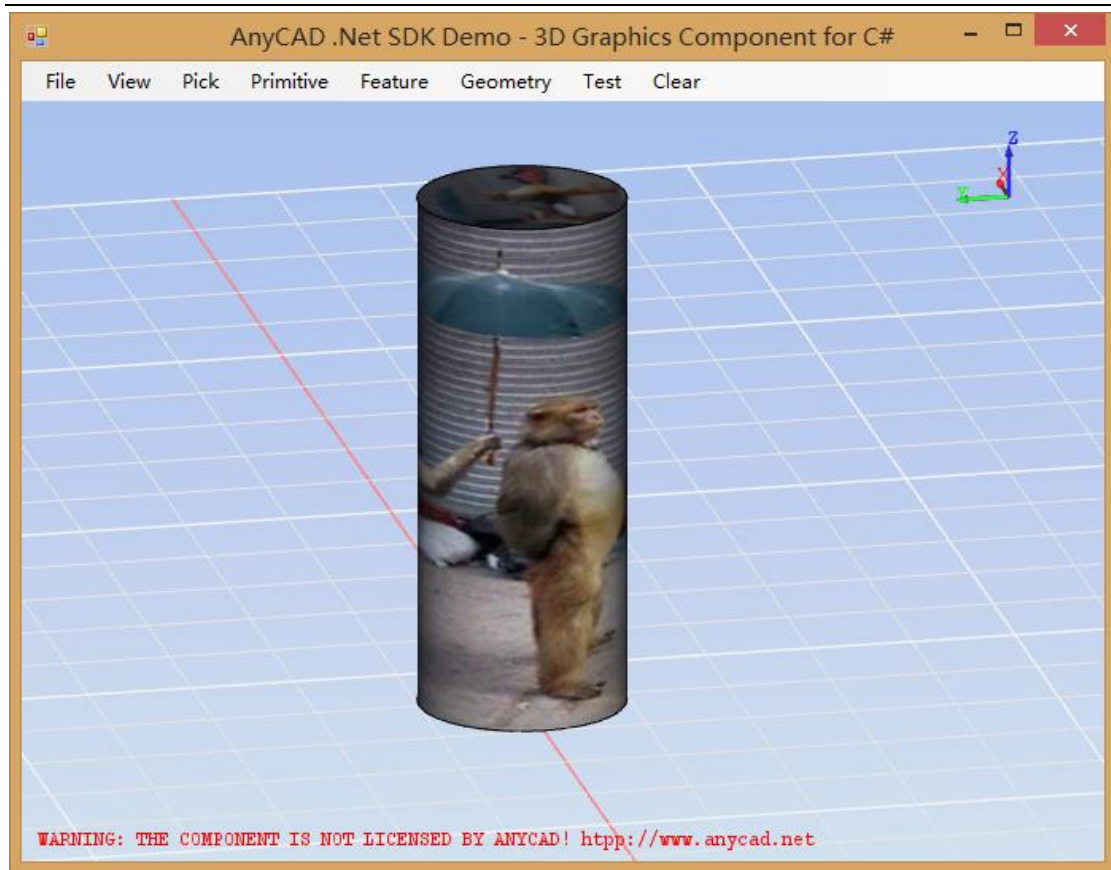


使用 Renderer::QuerySelection 可以获取选择的物体

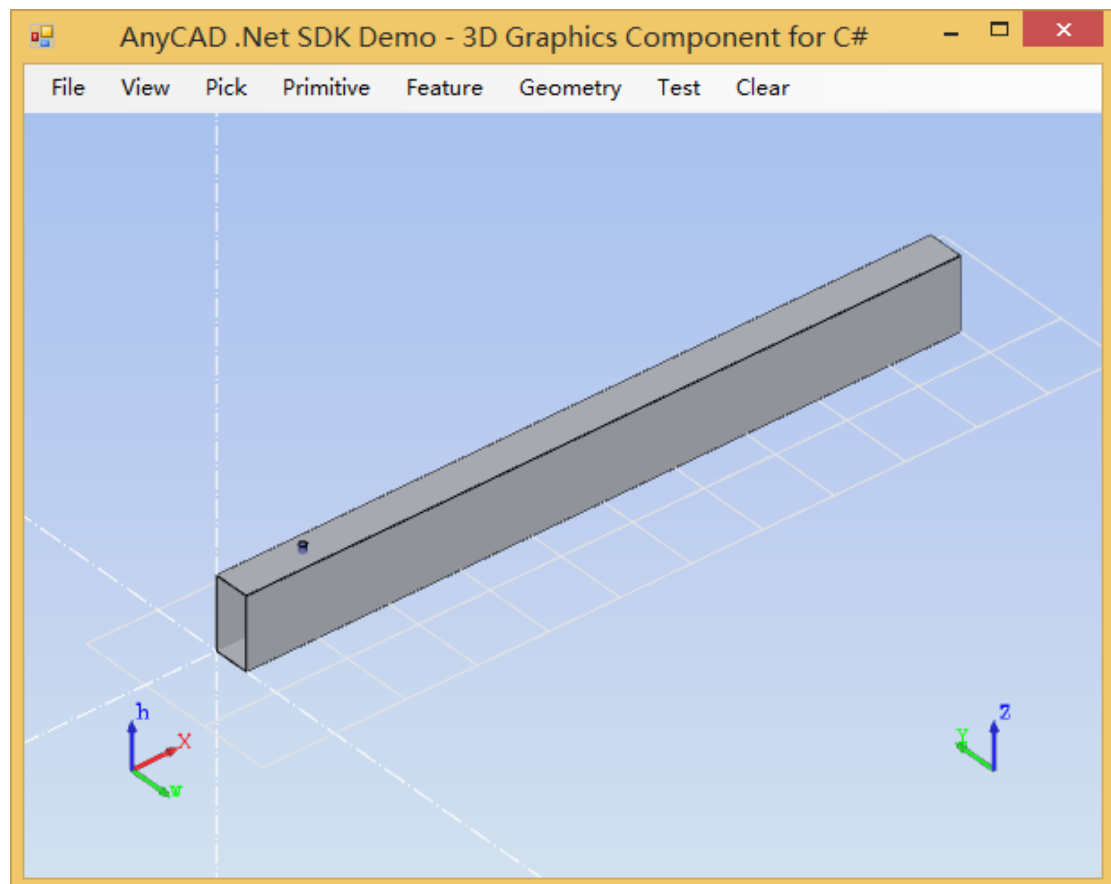
```
QuerySelectedShapeContext context = new QuerySelectedShapeContext();
renderer.QuerySelection(context);
int id = context.GetNodeId();
MessageBox.Show(id.ToString());
```

6. 示例

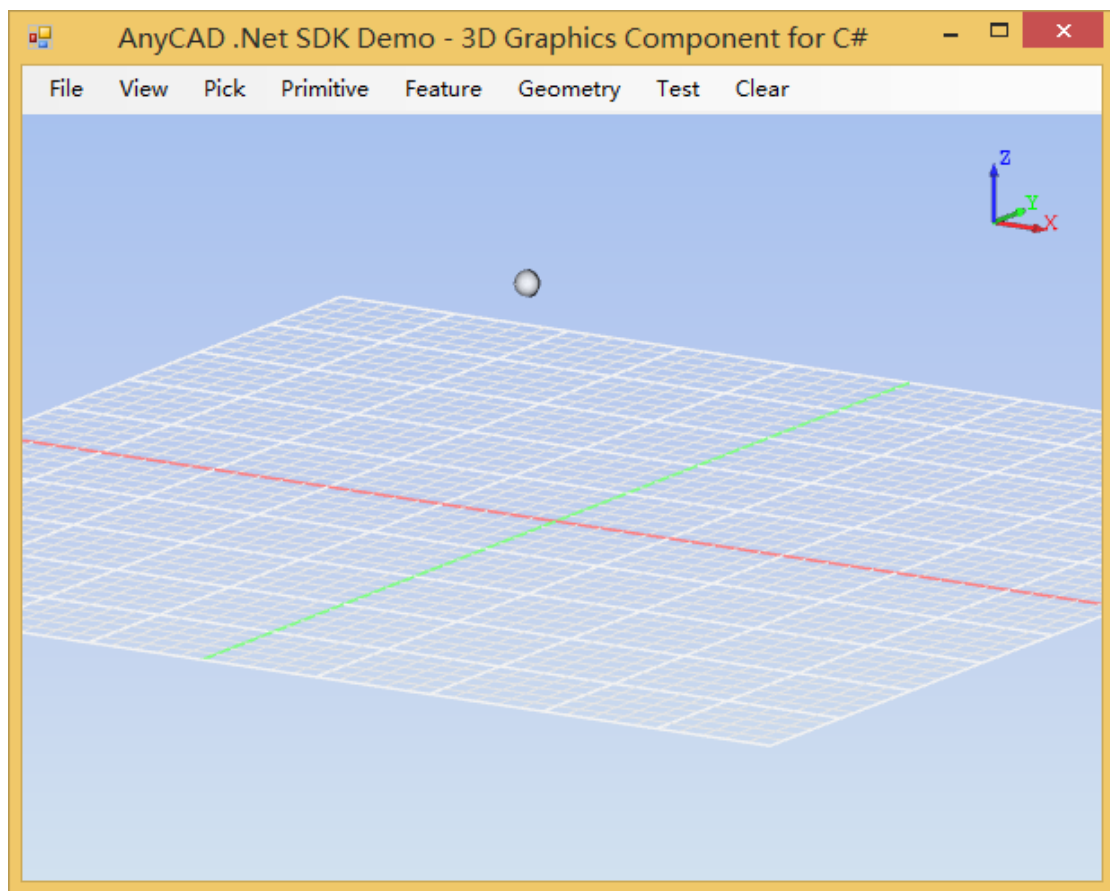
详细代码请参考附带的例子程序



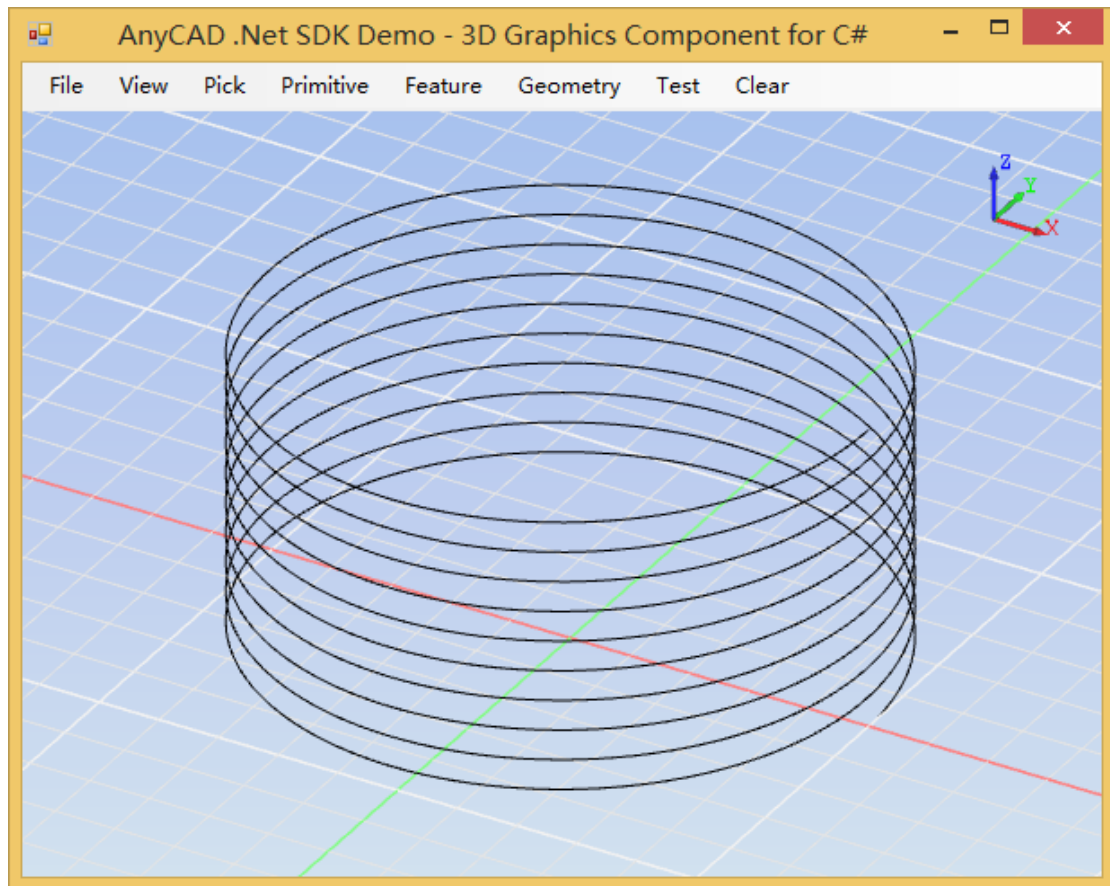
纹理



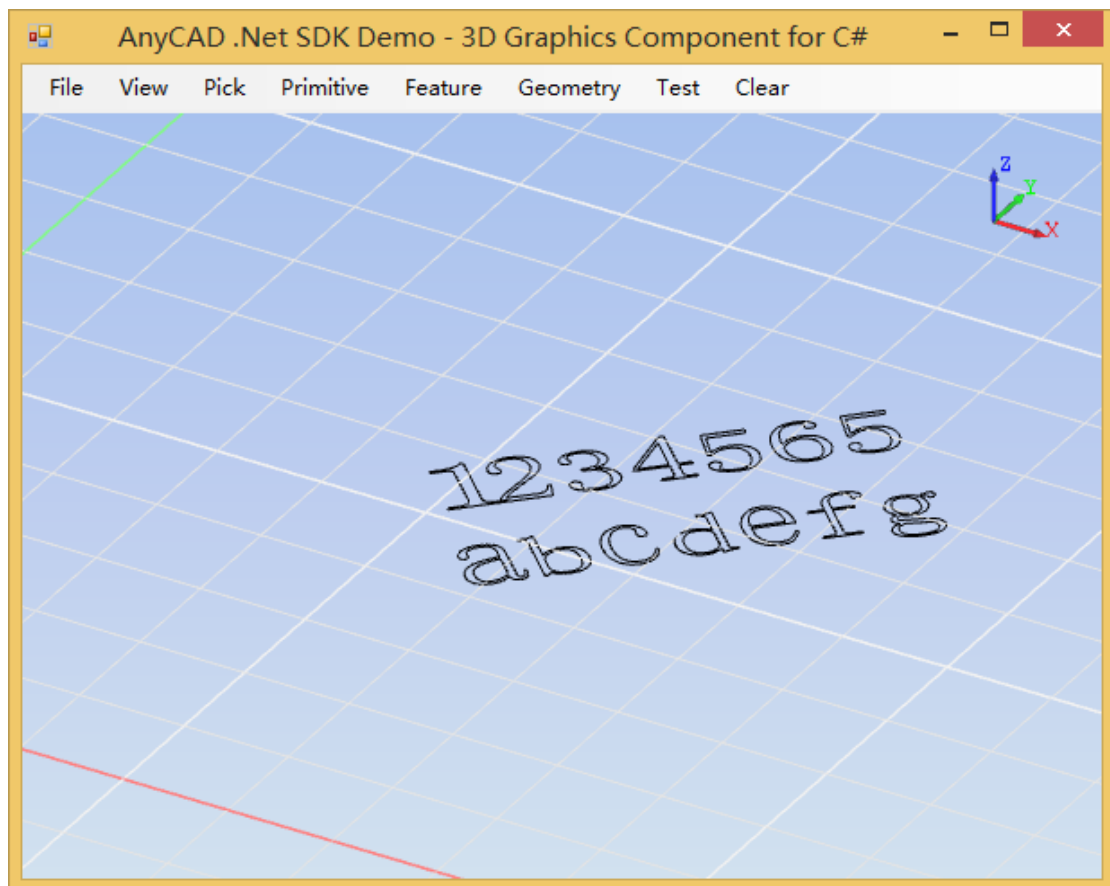
演示自定义坐标轴、网格



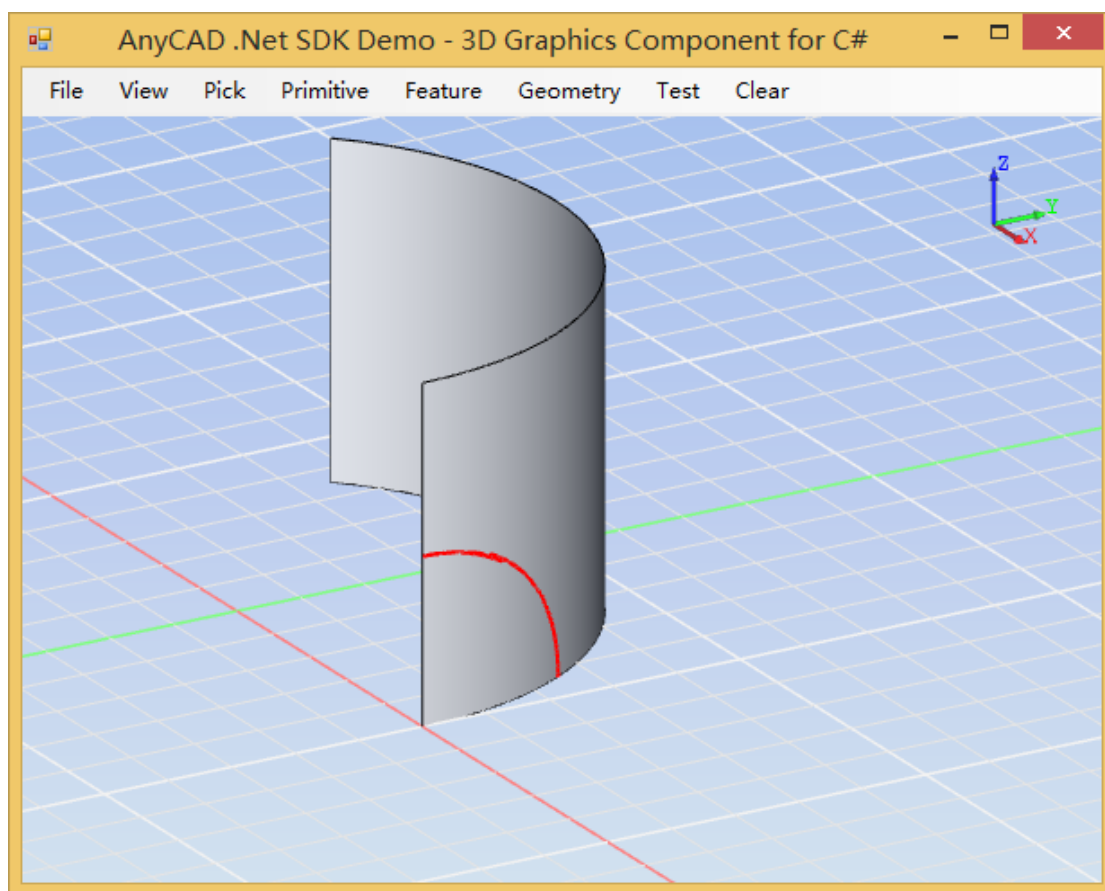
跳动的小球（动画演示）



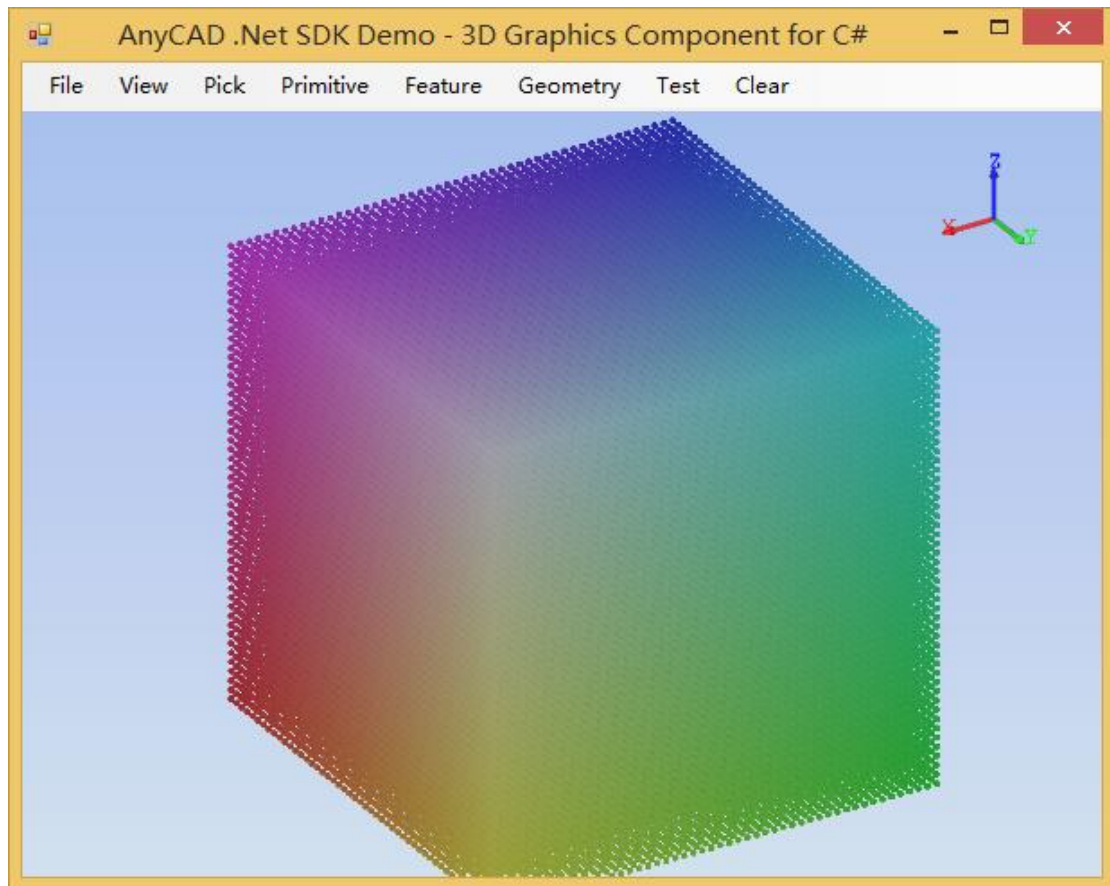
弹簧线



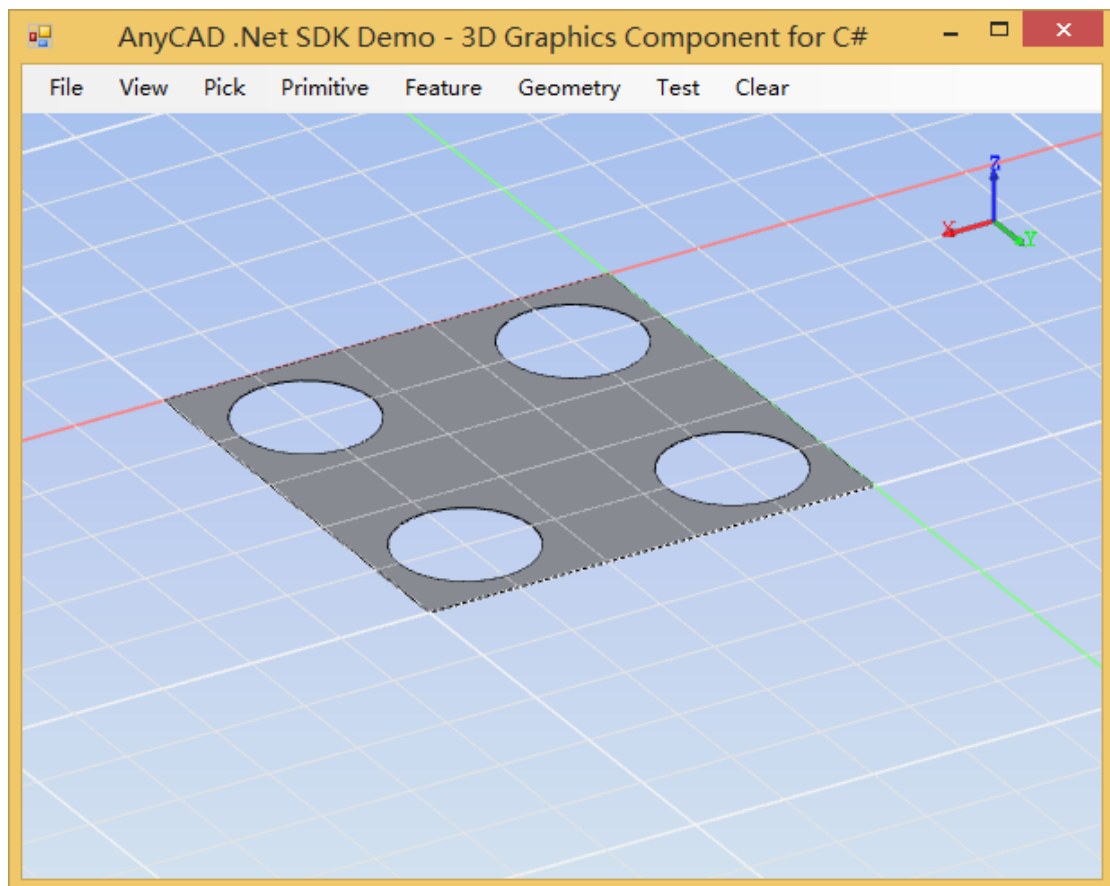
三维文字



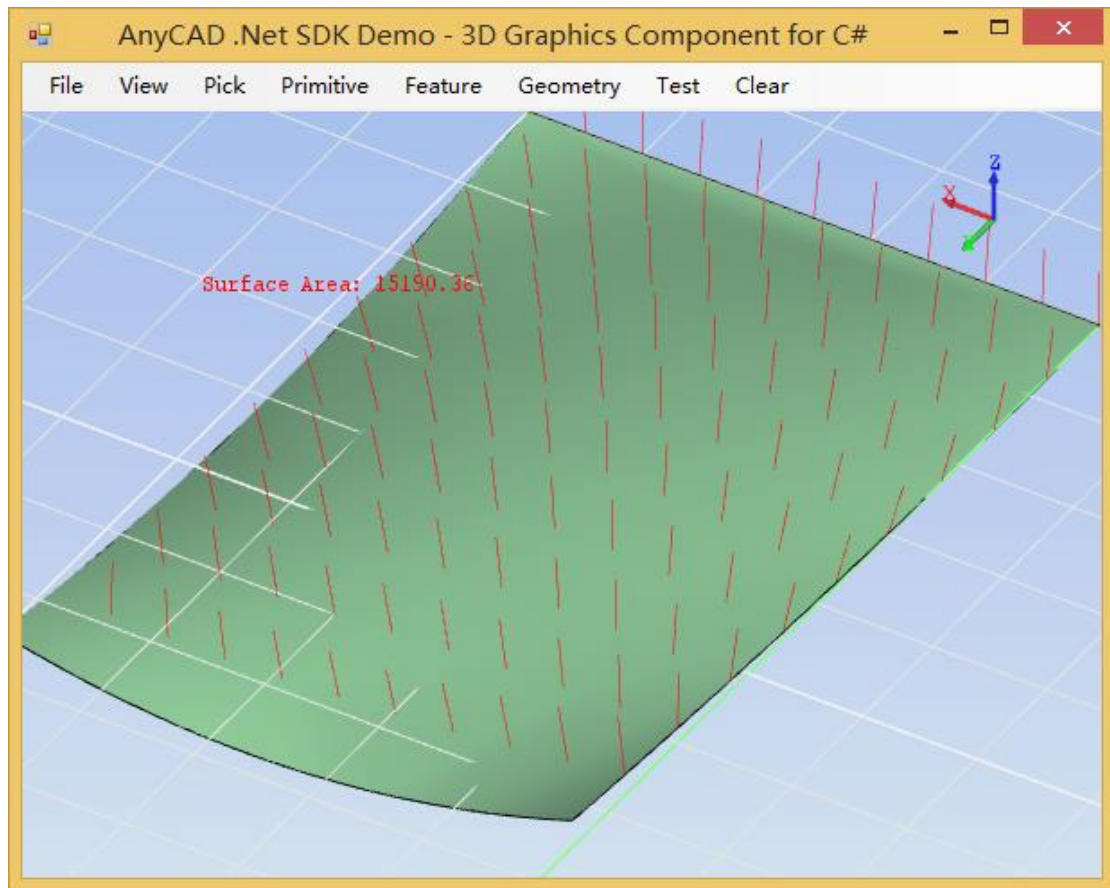
曲面相交线



点云显示



智能生成带孔的面



曲面信息提取

7. 参考资料

- 在线帮助文档: <http://www.anycad.net/help/>
- 示例程序代码:
 - <https://github.com/anycad/anycadnetsdkpro>
 - <https://github.com/anycad/StpViewer>
 - <https://github.com/anycad/dxfViewer>
- 视频教程
 - <http://i.youku.com/u/UMjm1OTAwNTg4>