# VS2017 配置 CG/OpenGL环境

## 1.安装VS2017-勾选C++

<https://visualstudio.microsoft.com/zh-hans/>

## 2.安装Cg Toolkit

<http://developer.nvidia.com/object/cg_toolkit.html>

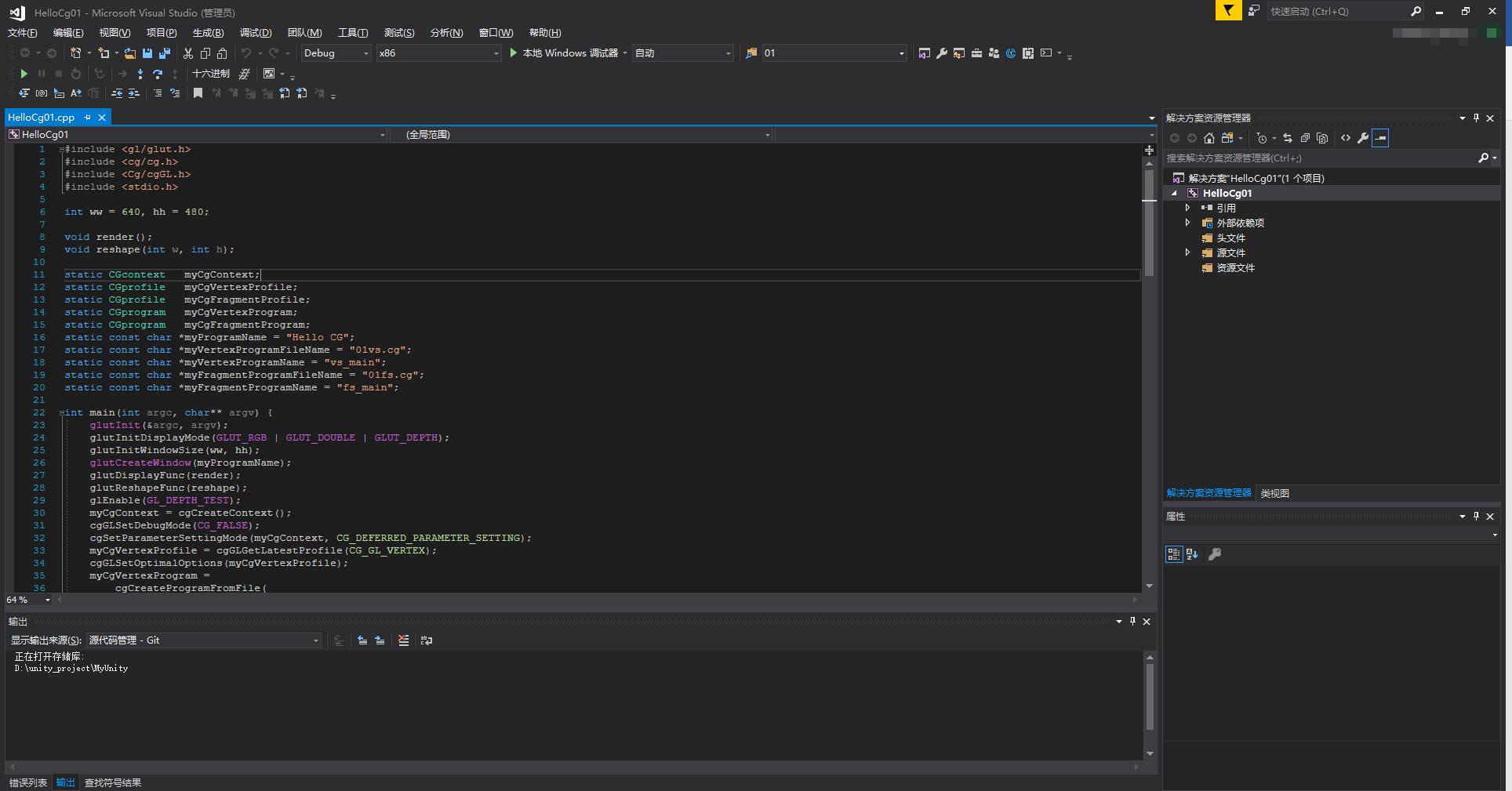
安装后，就会生成三个环境变量,CG\_BIN\_PATH,CG\_INC\_PATH,CG\_LIB\_PATH

## 3.打开需要配置CG的项目(新建一个也行)

例如新建一个项目,参考文章<https://blog.csdn.net/zhangci226/article/details/5540153>

|  |
| --- |
| #include <gl/glut.h>  #include <cg/cg.h>  #include <Cg/cgGL.h>  #include <stdio.h>  int ww = 640, hh = 480;  void render();  void reshape(int w, int h);  static CGcontext   myCgContext;  static CGprofile   myCgVertexProfile;  static CGprofile   myCgFragmentProfile;  static CGprogram   myCgVertexProgram;  static CGprogram   myCgFragmentProgram;  static const char \*myProgramName = "Hello CG";  static const char \*myVertexProgramFileName = "01vs.cg";  static const char \*myVertexProgramName = "vs\_main";  static const char \*myFragmentProgramFileName = "01fs.cg";  static const char \*myFragmentProgramName = "fs\_main";  int main(int argc, char\*\* argv) {      glutInit(&argc, argv);      glutInitDisplayMode(GLUT\_RGB | GLUT\_DOUBLE | GLUT\_DEPTH);      glutInitWindowSize(ww, hh);      glutCreateWindow(myProgramName);      glutDisplayFunc(render);      glutReshapeFunc(reshape);      glEnable(GL\_DEPTH\_TEST);      myCgContext = cgCreateContext();      cgGLSetDebugMode(CG\_FALSE);      cgSetParameterSettingMode(myCgContext, CG\_DEFERRED\_PARAMETER\_SETTING);      myCgVertexProfile = cgGLGetLatestProfile(CG\_GL\_VERTEX);      cgGLSetOptimalOptions(myCgVertexProfile);      myCgVertexProgram =          cgCreateProgramFromFile(              myCgContext,              /\* Cg runtime context \*/              CG\_SOURCE,              /\* Program in human-readable form \*/              myVertexProgramFileName,              /\* Name of file containing program \*/              myCgVertexProfile,              /\* Profile: OpenGL ARB vertex program \*/              myVertexProgramName,              /\* Entry function name \*/              NULL);      /\* No extra compiler options \*/      cgGLLoadProgram(myCgVertexProgram);      myCgFragmentProfile = cgGLGetLatestProfile(CG\_GL\_FRAGMENT);      cgGLSetOptimalOptions(myCgFragmentProfile);      myCgFragmentProgram =          cgCreateProgramFromFile(              myCgContext,              /\* Cg runtime context \*/              CG\_SOURCE,              /\* Program in human-readable form \*/              myFragmentProgramFileName,              /\* Name of file containing program \*/              myCgFragmentProfile,              /\* Profile: OpenGL ARB vertex program \*/              myFragmentProgramName,              /\* Entry function name \*/              NULL);      /\* No extra compiler options \*/      cgGLLoadProgram(myCgFragmentProgram);      glutMainLoop();      return 0;  }  void reshape(int w, int h) {      glMatrixMode(GL\_PROJECTION);      glLoadIdentity();      gluPerspective(45, (float)w / (float)h, 0.1, 100);      glViewport(0, 0, w, h);      ww = w;      hh = h;  }  void render() {      glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);      glClearColor(.0f, .0f, .2f, 1.0f);      glMatrixMode(GL\_MODELVIEW);      glLoadIdentity();      gluLookAt(.0, .0, 5.0, .0, .0, .0, .0, 1.0, .0);      static float angle;      glRotatef(angle, 0.0, 1.0, 0.0);      cgGLBindProgram(myCgVertexProgram);      cgGLEnableProfile(myCgVertexProfile);      cgGLBindProgram(myCgFragmentProgram);      cgGLEnableProfile(myCgFragmentProfile);      //将ModelViewProjection矩阵传入shader      CGparameter mvp = cgGetNamedParameter(myCgVertexProgram, "MVP");      cgGLSetStateMatrixParameter(mvp, CG\_GL\_MODELVIEW\_PROJECTION\_MATRIX, CG\_GL\_MATRIX\_IDENTITY);      glutSolidTorus(0.3, 1.0, 30, 30);      cgGLDisableProfile(myCgVertexProfile);      cgGLDisableProfile(myCgFragmentProfile);      angle += 0.5;      if (angle >= 360) angle = 0.0f;      glutSwapBuffers();      glutPostRedisplay();  } |

复制代码:



还有两个.cg文件:

01fs.cg

|  |
| --- |
| 01fs.cg |
| float4 fs\_main(float4 color: COLOR): COLOR  {      return color;  } |

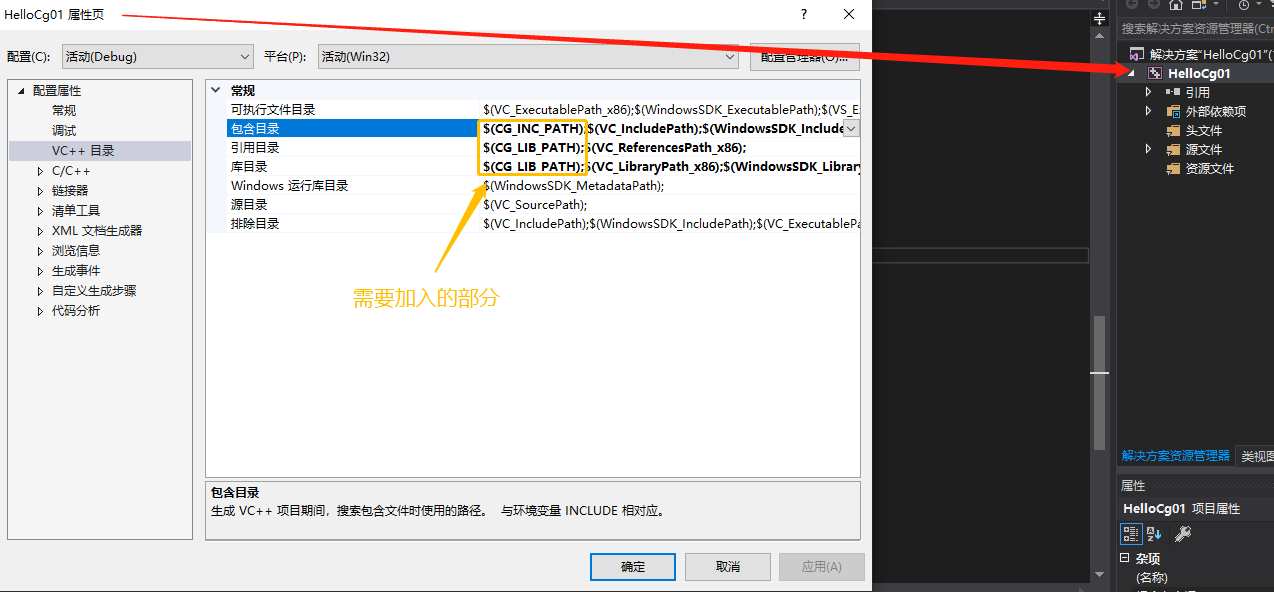
01vs.cg

|  |
| --- |
| 01vs.cg |
| struct output  {      float4 position: POSITION;      float4 color: COLOR;  };  output vs\_main(float4 position: POSITION,float4 color: COLOR,uniform float4x4 MVP){      output OUT;      OUT.position = mul(MVP, position);      OUT.color = position;      return OUT;  } |

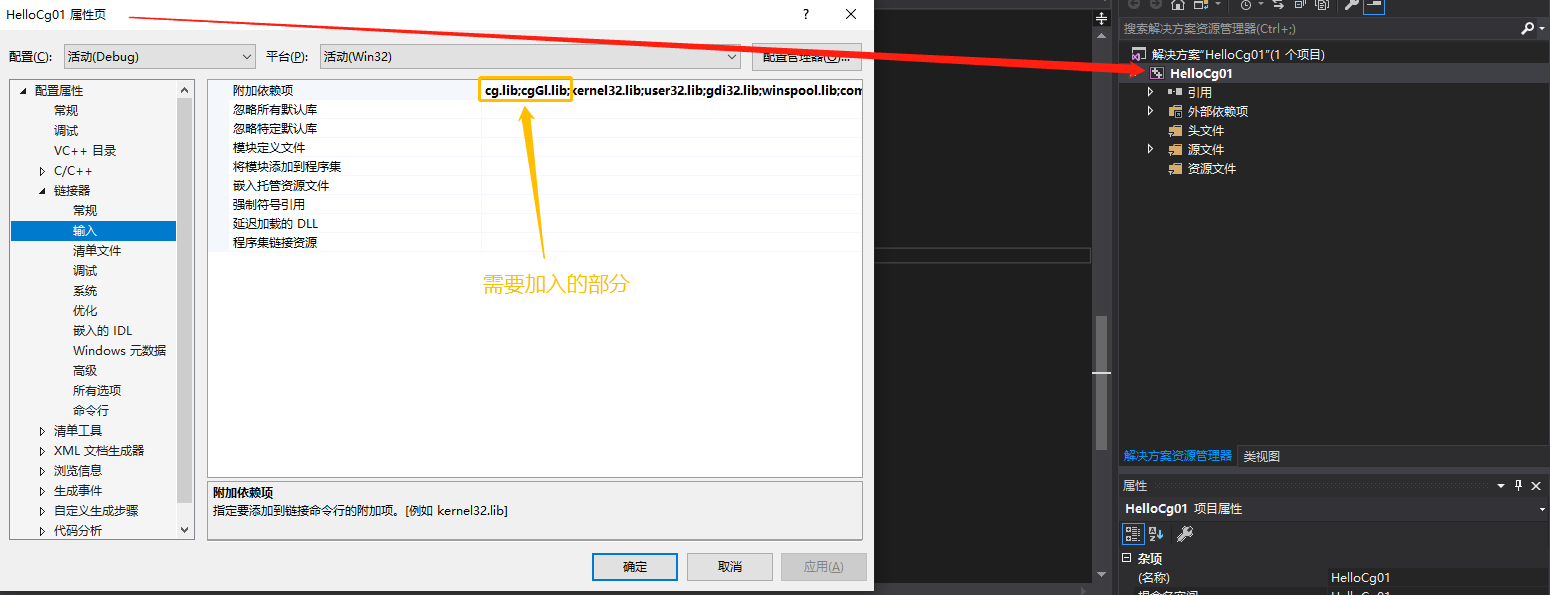
## 4.配置VS2017+CG环境

$(CG\_INC\_PATH);

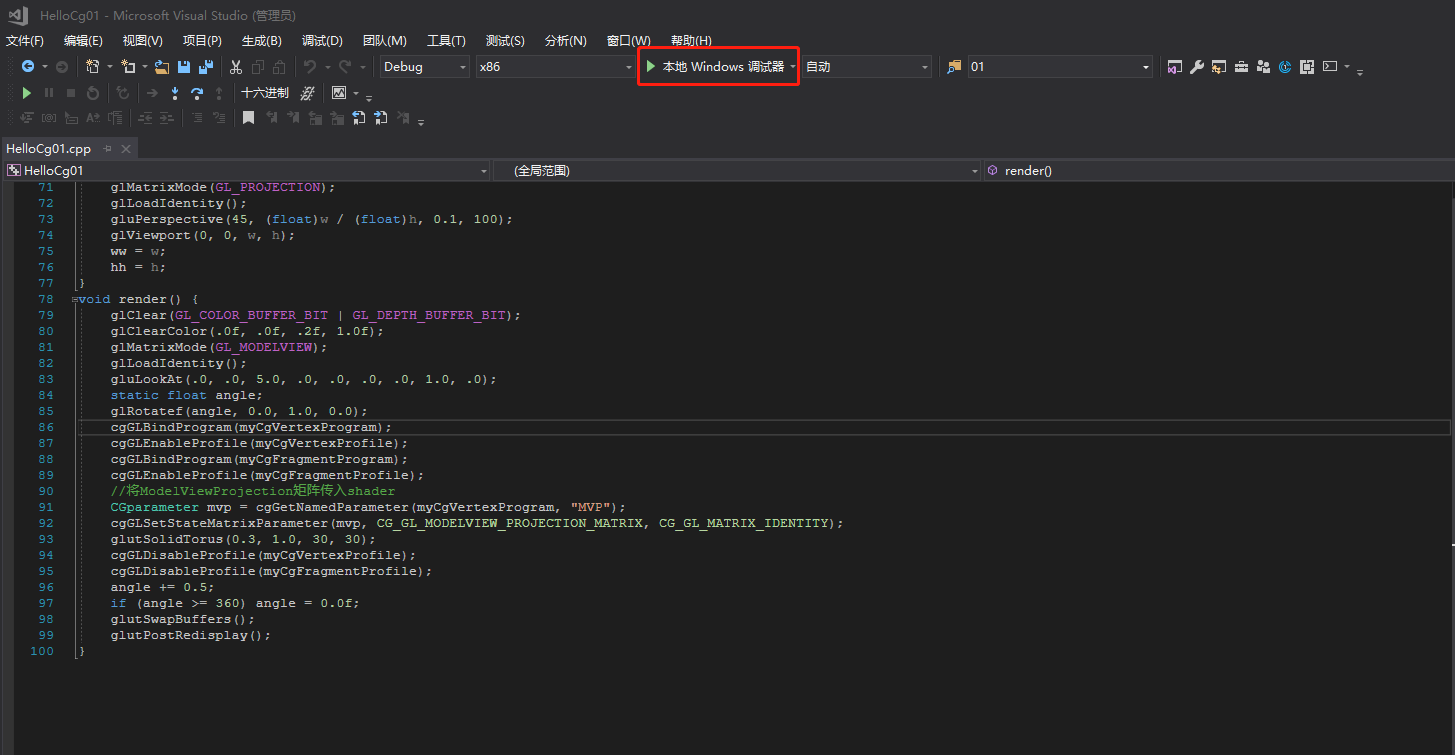
$(CG\_LIB\_PATH);

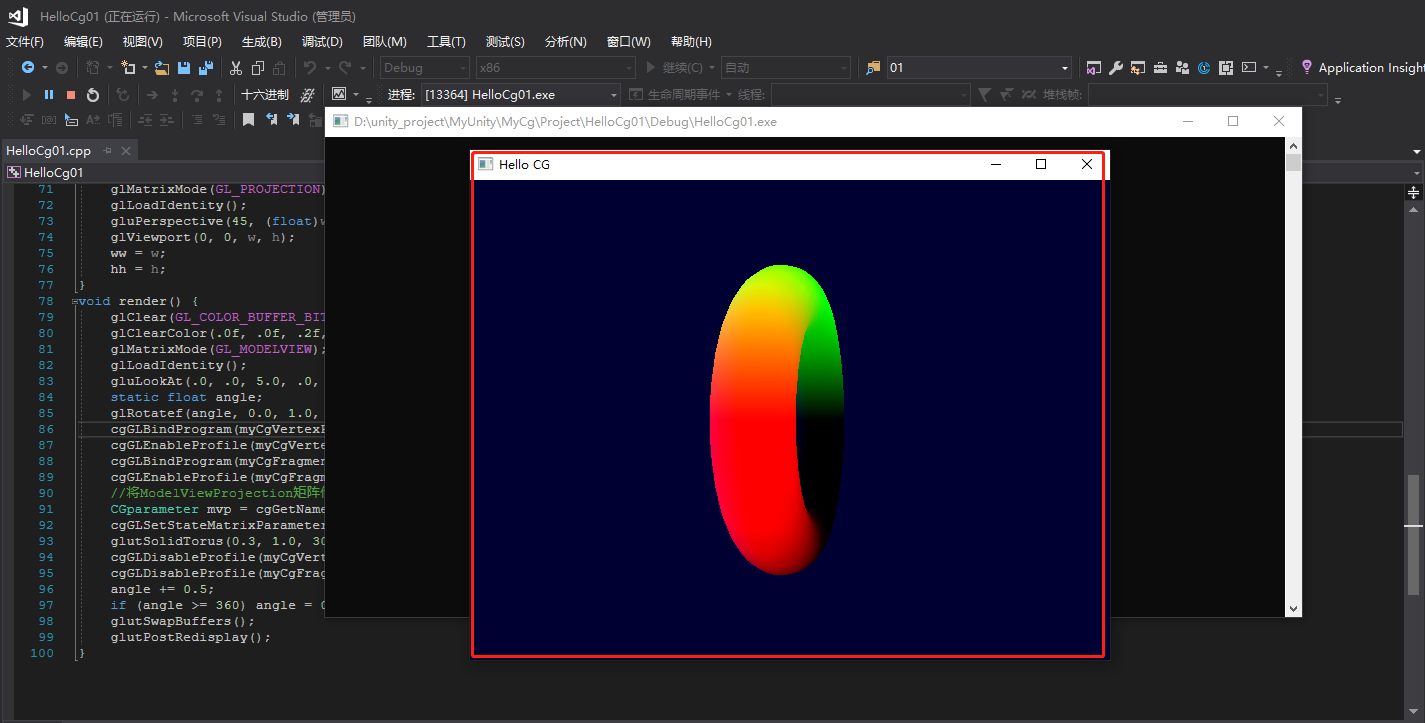


cg.lib;cgGl.lib;



## 5.点击本地Windows调试器





## 6.不推荐方案-VS2017-Nuget包-nupengl插件

