VS+QT环境配置

1、安装vs 2017和qt, 下载下面这个,完成在VS中安装qt插件。 Qt Visual Studio Tools(for vs 2017)链接: https://pan.baidu.com/s/1w15AGfPG5JUWd6jnNd80JQ,提取码: 8x4z

2、在VS中进行qt的配置,如下图

点击后选择 options 或 qt versions, 🪽

选项							?	>
搜索选项(Ctrl+E)	P	Default	Version	Host	Path	Compiler		
<ul> <li>● 漢武</li> <li>● 法第三員</li> <li>● Azure 服務券份验证</li> <li>● Azure 服務券份验证</li> <li>● Acure 服務券份验证</li> <li>● FF Tools</li> <li>● FF Tools</li> <li>● Nucles 100%</li> <li>● Ceneral</li> <li>Ceneral</li> <li>Ceneral</li> <li>Versions</li> <li>Legacy Project Format</li> <li>&gt; SQL Server 工具</li> <li>&gt; VisualSVN</li> <li>&gt; Web</li> <li>&gt; Web</li> <li>&gt; Web 短期税工具</li> <li>&gt; Web 短期税工具</li> <li>&gt; Wich 短期税工具</li> <li>&gt; Aune 100%</li> </ul>		X	<ul> <li>5.142_mevc2017_64</li> <li>add new Qt version&gt;</li> </ul>	Windows ¥	Diqt5.142/msv2017_64	meve		
						确定	R	9 <b>6</b>

选择 add new,进行添加。在选择 path 选择添加文件的路径。

Default	Version	Host	Path	Compiler
$\checkmark$	<b>5.14.2_msvc2017_64</b>	Windows ~	🐂 D:\qt\5.14.2\msvc2017_64	msvc
	●/////////////////////////////////////	Windows ~	<b>1</b>	msvc
	😌 <add new="" qt="" version=""></add>			

#### 找到安装 qt 的文件的位置, 找到下面这个程序。

> lt	t电脑 > D (D:) > qt > 5.14.2 > msvc2	017_64 → bin		✓ ひ 搜索"bin"
挟				
^	名称	修改日期	类型	大小
	📧 qmake 👝	2020/3/27 21:18	应用程序	2,799 KB

添加成功后,可完成 qt 在 VS 中的配置。--



### 每个库的结构如下 Bin=dll文件, (此文中该文件夹也用作vs的工作目录) Include=头文件 Lib=lib文件

n
nake
clude
ugins
are

# 在lib文件夹下批处理文件输入代码: "DIR \*.\* /B >list.TXT",并运行,得到lib 的名称。后面用。

cmake	15/07/2021 20:49	File folder	
💿 a.bat	18/07/2021 20:02	Windows Batch File	1 KB
😓 list.TXT	18/07/2021 20:03	TXT File	3 KB
💵 vtkalglib-8.0.lib	15/07/2021 20:49	Object File Library	47 KB
🏙 vtkChartsCore-8.0.lib	15/07/2021 20:49	Object File Library	580 KB

### 步骤1 1.1双击打开VS



## 1.2出现空白的软件界面,新建项目



1.3弹出窗口选择QT,并修改名称,路径



步骤1.3

📢 Qt Widgets Application Wizard

 $\times$ 

Welcome to the Qt Widgets Application Wizard

This wizard generates a Qt Widgets application project. The application derives from QApplication and includes an empty widget.

To continue, click Next.



### 单击Next->next 有需要就修改类名称 单击finish

Welcome to the Qt Widgets Application Wizard

This wizard generates a Qt Widgets application project. The application derives from QApplication and includes an empty widget.

 $\times$ 

Class Name:	Base class:	
QtWidgetsApplication1	QMainWindow	*
Header (.h) file:	Source (.cpp) file:	
QtWidgetsApplication1.h	QtWidgetsApplication1.cpp	<u>ر</u>
User Interface (.ui) file:	Resource (.qrc) file:	
QtWidgetsApplication1.ui	QtWidgetsApplication1.qrc	
Lower case file names  Precompiled header  Add default application icon		
< Pr	vious Next > Finish	Cancel

步骤1.4

出现如下文件,ui=qt界面(双击进入),h,cpp文件(单击进入) 双击.ui文件,进入GUI设计界面Qt Designer。单击"detach",可让窗口弹出,视野更好





### 步骤1.5 拖动一个"widget"到窗口中; 在widget上右击,单击"提升为"; 手动输入QVTKWidget(大小写一致)



### 步骤1.6 发现原来的widget已经变成了QVTKWidget

01 OtWidgetsApplication1 - OtWidgetsApplication1 ui*
lype mere

### 单击保存,数据会传送回VS

Qt Designer	
File Edit Form View Settings Window Help	
D 🥖 🔒 🔚 🖵   📑 🎭 🖏   (	II = H I II II II II II II
Widget Box 🗗	QtWidgetsApplication1 - QtWidgetsApplication1.ui*
Filter	Type Here
✓ Layouts	∧
Vertical Layout	
IIII Horizontal Layout	
Grid Layout	
🛱 Form Layout	

### 步骤1.7 返回vs,在项目上右击-》属性。调试-》改工作目录为vtk的"bin"



### 步骤1.8 VC++目录-》包含目录=vtk的"include下面出现h文件的路径"; 库目录=vtk的"lib"文件夹。

#### QtWidgetsApplication1 属性页

?

配置(C): Debug	~ 平台(P): 活动(x64)	~ 配置管理器(O)
▲ 配置属性 ^	✓ 常規	
常规	可执行文件目录	\$(VC_ExecutablePath_x64);\$(WindowsSDK_ExecutablePath);\$(VS_Execut
调试	包含目录	D:\Work\VS\CMCsMech\External\VTK\vtk8.0 debug\include\vtl
VC++ 目录	引用目录	\$(VC ReferencesPath x64);
Qt Project Settings	库目录	D:\Work\VS\CMCsMech\External\VTK\vtk8.0 debug\lib:\$(LibraryPa
Qt Meta-Object Compi	Windows 运行库目录	\$(WindowsSDK MetadataPath);
Qt Resource Compiler	源目录	\$(VC SourcePath):
Qt User Interface Comp	排除目录	\$(VC IncludePath):\$(WindowsSDK IncludePath):\$(VC ExecutablePath x
▷ C/C++	100.000	*(*
▲ 链接器		
常规		
输入		
清单文件		
调试		
系统		
优化		
嵌入的 IDL		
Windows 元数据		
高级		
所有选项		
命令行		
▷ 清单工具		
▷ XML 文档生成器	包含目录	
▷ 浏览信息	生成 VC++ 项目期间,搜索包含文件部	付使用的路径。 与环境变量 INCLUDE 相对应。
< + str = //+ >		
		OK Cancel Apply

### 步骤1.9 C/C++->常规-》附加包含目录=vtk的"include下面出现h文件的路径";

QtWidgetsApplication1 属性页		? ×
配置(C): Debug	~ 平台(P): 活动(x64)	~ 配置管理器(0)
▲ 配置属性 常规 调试 VC++目录 Qt Project Settings ▷ Qt Meta-Object Compi ▷ Qt Resource Compiler ▷ Qt User Interface Comp ▲ C/C++ 常规 优化 预处理器 代码生成 语言 预编译头 输出文件 浏览信息 高级 所有选项 命令行 ▷ 链接器 ▷ 清单工具 ▷ XML 文档生成器	附加包含目录 其他 #using 指令 调试信息格式 支持仅我的代码调试 公共语言运行时支持 使用 Windows 运行时扩展 取消显示启动版权标志 警告等级 将警告视为错误 警告版本 诊断格式 SDL 检查 多处理器编译	D:\Work\VS\CMCsMech\External\VTK\vtk8.0_debug\include\vtk-8. 程序数据库 (/Zi) 否 是 (/nologo) 等级 1 (/W1) 否 (/WX-) 传统型 (/diagnostics:classic) 是 (/MP)
▶ 浏觉信息 ▶ # dī 畫/# < >	皆定—个或多个要添加到包含路径中的目录 	表;当目录不止一个时,请用分号分隔。 (川路径])
		OK Cancel Apply

### 步骤1.20 链接器-》常规-》附加库目录=vtk的"lib"文件夹。

#### QtWidgetsApplication1 属性页

? X

配置(C	:):	Debug	~	平台(P):	活动(x64)				$\sim$	配置管	理器(0)
	) The factor is a second secon	214-	. tou	177/4			\$(OutDia)\$/TerretNierre	) \$ (Tttt)			
	こにに	与11 <u>十</u> 401		1又1十 2)#1度			3(OutDir)3( largetivame 土:八四	e)\$(largetExt)			
	市	/%: \f	ر/ <u>نین</u> د برخ <u>م</u> ند	1.550			不反互				
	VC		版4								
	0+	Project Settings	后开	1911年19日 1911日 - 1911日	-+17+						
Ь В	Ot	Meta-Object Compi	4X/F	驱水后动脉	风仪你心		走(/NOLOGO)				
	Ot	Resource Compiler	心明	诗人库			(1)				
	Ot	User Interface Comr	注册	摘出			<u> 日</u>				
	C/0	C++	逐用	沢 王 定 同			音				
	锚	- 接器	<u></u>	<u> </u>			D:\Work\VS\CMCsMe	ch\External\VTK\vtk	(8.0_de	ebug\lib	%(Addition
		常规	链接	好依赖项			是				
		输入	使用	作依赖项籍	入		合				
		清单文件	链接	狱态							
		调试	KH TI	_ DII 绑定							
		系统	将锁	接器警告视	的错误						
		优化	强制	1文件输出							
		嵌入的 IDL	创建	回热修补明	像						
		Windows 元数据	指定	节特性							
		高级									
		所有选项									
		命令行									
Þ	清	单工具									
⊳	XM	NL 文档生成器	附加库	目录							
⊳	浏	览信息	、 允许用	□重写环境	库路径。(/LIE	3PATH:folder)					
<	4				1						
								OK	Cano	el 🛛	Apply

### 步骤1.21 链接器-》输入-》附加依赖项=vtk的lib文件夹下所有lib文件名称。第2页ppt得到的。

QtWidgetsApplication1 属性页					? X
配置(C): Debug	~ 平台(P): 活动(x64)			~ 配置	管理器(0)
▲ 配置属性 常规 调试 VC++目录 Qt Project Settings ▷ Qt Meta-Object Compi ▷ Qt Resource Compiler ▷ Qt User Interface Compi ▷ Qt User Interface Compi ▷ C/C++ ④ 链接器 常规 输入 输入 有單文件 调试 系统 优化 嵌入的 IDL Windows 元数据 高级 所有选项 命令行 ▷ 清单工具 ▷ XML 文档生成器 ▷ 浏览信息 ▷ 4+ rdt=//4	附加依赖项 忽略所有默认库 忽略特定默认库 模块定义文件 将模块添加到程序集 嵌入托管资源文件 强制符号引用 延迟加载的 DLL 程序集链接资源	. [例如 kernel32.lib]	iCore-8.0.lib;vt	kCommonColo	r-8.0.lib;vtkCo
			ОК	Cancel	Apply

上述操作冗余了。

### 步骤**1.22** 头文件增加如下代码。

main.	срр	QtWidgetsApplication1.h 👳 🗙 QtWidgetsApplication1.cpp	
♣ Qt	QtWidgetsApplication1 - 🕈 QtWidgetsApplication1		
	1	#pragma•once	
	2		
	3	_⊟#include• <qtwidgets qmainwindow=""></qtwidgets>	
	4	<pre>#include ''ui_QtWidgetsApplication1.h"</pre>	
	5		
	6	#include• <vtkautoinit.h></vtkautoinit.h>	
	7	₽#ifndef•INITIAL_OPENGL	
	8	#define INITIAL_OPENGL	
	9	VTK_MODULE_INIT(vtkRenderingOpenGL);	
	10	<pre>VTK_MODULE_INIT(vtkInteractionStyle);</pre>	
	11	<pre>VTK_MODULE_INIT(vtkRenderingFreeType);</pre>	
	12	#endif	
	13		
	14	¤#include• <vtkspheresource. h=""></vtkspheresource.>	
	15	<pre>#include.<vtkpolydata.h></vtkpolydata.h></pre>	
	16	<pre>#include &lt;&lt; vtkSmartPointer. h&gt;</pre>	
	17	<pre>#include.<vtkpolydatamapper.h></vtkpolydatamapper.h></pre>	
	18	<pre>#include &lt;&lt; vtkActor. h&gt;</pre>	
	19	<pre>#include.<vtkrenderwindow.h></vtkrenderwindow.h></pre>	
	20	<pre>#include &lt;&lt; vtkRenderer. h&gt;</pre>	
	21	[#include• <vtkrenderwindowinteractor.h></vtkrenderwindowinteractor.h>	
	22		
	23		
	24	<pre>class.QtWidgetsApplication1public.QMainWindow</pre>	
	25	{	

#include <vtkAutoInit.h>
#ifndef INITIAL\_OPENGL
#define INITIAL\_OPENGL
VTK\_MODULE\_INIT(vtkRenderingOpenGL);
VTK\_MODULE\_INIT(vtkInteractionStyle);
VTK\_MODULE\_INIT(vtkRenderingFreeType);
#endif

#include <vtkSphereSource.h>
#include <vtkPolyData.h>
#include <vtkSmartPointer.h>
#include <vtkPolyDataMapper.h>
#include <vtkActor.h>
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkRenderer.h>

### 步骤1.23

### Cpp文件构造函数增加如下代码。 代码中Ui.widget根据qt designer中的 object名称适应性修改。

main.cpp	QtWidgetsApplication1.h QtWidgetsApplication1.cpp 🗦 🗙		
S QtWidgetsApplication1 → QtWidgetsApplication1			
1	#include•"QtWidgetsApplication1.h"		
2			
3	₽QtWidgetsApplication1::QtWidgetsApplication1(QWidget·*parent)		
4	••••:•QMainWindow(parent)		
5	{		
6	••••ui.setupUi(this);		
7			
8	vtkSmartPointer <vtkspheresource>・sphereSource・=</vtkspheresource>		
9	→ vtkSmartPointer <vtkspheresource>::New();</vtkspheresource>		
10	→ sphereSource->SetCenter(0.0, •0.0, •0.0);		
11	→ sphereSource->SetRadius(5.0);		
12			
13	→ //mapper		
14	vtkSmartPointer <vtkpolydatamapper>·mapper·=</vtkpolydatamapper>		
15	→ vtkSmartPointer <vtkpolydatamapper>::New();</vtkpolydatamapper>		
16	mapper->SetInputConnection(sphereSource->GetOutputPort());		
17			
18	→ //actor		
10			

vtkSmartPointer<vtkSphereSource> sphereSource =
vtkSmartPointer<vtkSphereSource>::New();
sphereSource->SetCenter(0.0, 0.0, 0.0);
sphereSource->SetRadius(5.0);

#### //mapper

vtkSmartPointer<vtkPolyDataMapper> mapper =
vtkSmartPointer<vtkPolyDataMapper>::New();
mapper->SetInputConnection(sphereSource->GetOutputPort());

#### //actor

vtkSmartPointer<vtkActor> actor =
vtkSmartPointer<vtkActor>::New();
actor->SetMapper(mapper);

//renderer ,renderWindow, renderWindowInteractor.
vtkSmartPointer<vtkRenderer> renderer =
vtkSmartPointer<vtkRenderer>::New();
vtkSmartPointer<vtkRenderWindow> renderWindow =
vtkSmartPointer<vtkRenderWindow>::New();

ui.widget->GetRenderWindow()->AddRenderer(renderer);//加入renderer显示

vtkSmartPointer<vtkRenderWindowInteractor> renderWindowInteractor =
vtkSmartPointer<vtkRenderWindowInteractor>::New();
renderWindowInteractor = ui.widget->GetInteractor();

renderWindow = ui.widget->GetRenderWindow();

renderer->AddActor(actor); renderer->SetBackground(1, 1, 1); // Background color green

ui.widget->update();

步骤**1.24** 运行。

