**Visualization Toolkit (VTK)**

# 简介

## 概述

卫星、超级计算机、激光数字化系统和数字数据采集系统以惊人的速度采集、生成和传输数据。可视化为我们可以提取隐藏在数据中的重要信息提供了一些希望。以图像（或其他感官形式）的形式探索、转换和查看数据以理解和洞察数据的过程。

Visualization Toolkit (VTK)，开源、免费提供的软件系统，主要用于 3D 计算机图形学、建模、图像处理、体积渲染、科学可视化和 2D 绘图。它支持多种可视化算法和先进的建模技术，并利用线程和分布式内存并行处理技术，分别提升了处理速度和可扩展性。VTK 旨在实现平台无关性。这意味着它几乎可以在任何地方运行，包括 Linux、Windows 和 Mac 操作系统；以及在 Web 和移动设备上。

Kitware ，VTK 的核心功能使用 C++ 编写，以最大化效率。这些功能被封装到其他语言绑定中，以便更广泛的用户群体能够使用。与 Python 的互操作性特别完善。作为开源软件，VTK 可免费用于任何目的。

## 历史

**1993 - Origin：**Will Schroeder、Ken Martin 和 Bill Lorensen，他们从1993年12月开始在个人时间内编写该书和相关软件。教材《The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics》。VTK 的发展源于作者们在 GE 的工作经验。1998 年离开 GE，创办了 Kitware 公司。

**2014 - Transition from OpenGL to OpenGL2。**

**2016 - Rendering Backend in ParaView 5.0。**

## 特征

* **滤波器 (Filters)**

VTK 应用程序通过过滤器来操作数据。每个过滤器会检查它接收到的数据，并生成派生数据。多个连接的过滤器形成一个数据流网络。这个可配置的网络将原始数据转化为更加易于理解的视觉格式。

* **图形系统 (Graphics System)**

VTK 在底层图形库（主要是 OpenGL-面向过程的C函数库）之上添加了一个渲染抽象层。这个高层次的抽象简化了创建引人注目的可视化的任务。

* **数据模型 (Data Model)**

VTK 的核心数据模型能够表示几乎所有与物理科学相关的实际问题。用于表示各种类型的数据，包括点、线、多边形、体素和网格等。这些数据模型可以用于存储和处理数据，并被可视化算法使用。其基本数据结构特别适用于医学成像和涉及有限差分与有限元解法的工程工作。

* **数据交互 (Data Interaction)**

数据交互有助于理解数据的内容、形状和意义。在 VTK 中，3D 小部件、交互器以及与 Qt 等 2D 小部件库的接口使得您能够向程序中添加全面的用户交互功能。包括鼠标点击、平移和缩放等。

* **二维图形和图表 (2D Plots and Charts)**

VTK 提供了一整套用于处理表格数据的二维图形和图表类型。VTK 的选择和查询能力帮助您与数据进行交互，此外，VTK 还与 Python 和 Matplotlib 具有良好的互操作性。

* **并行处理 (Parallel Processing)**

VTK 对可扩展的分布式并行处理（通过 MPI）提供了优秀的支持。此外，许多 VTK 过滤器通过 vtkSMP（用于粗粒度线程）和 vtk-m（用于多核和 GPU 架构上的精细粒度处理）实现了更细粒度的并行化处理。

## 资料

* VTK官网：https://vtk.org/
* VTK官方例子：https://kitware.github.io/vtk-examples/site/
* VTK官方代码库：https://gitlab.kitware.com/vtk/vtk
* VTK官方参考手册：https://vtk.org/doc/nightly/html/index.html
* VTK官方论坛：<https://discourse.vtk.org/>

# 基本概念

## 面向对象的程序设计

与传统的程序系统相比，面向对象的系统更加模块化，更易于维护和描述。面向对象的设计过程尝试将复杂的任务划分为称为对象的小而简单的部分。面向对象的系统将数据和操作保存在一个称为对象的编程结构中。数据和操作共同构成了对象的属性。将操作应用于对象时，编程语言的动态绑定机制将执行适用于该对象的过程。面向过程的系统中，程序员必须提供逻辑来决定要调用哪个过程。

类：Class

对象：类(Class)的一个实例化

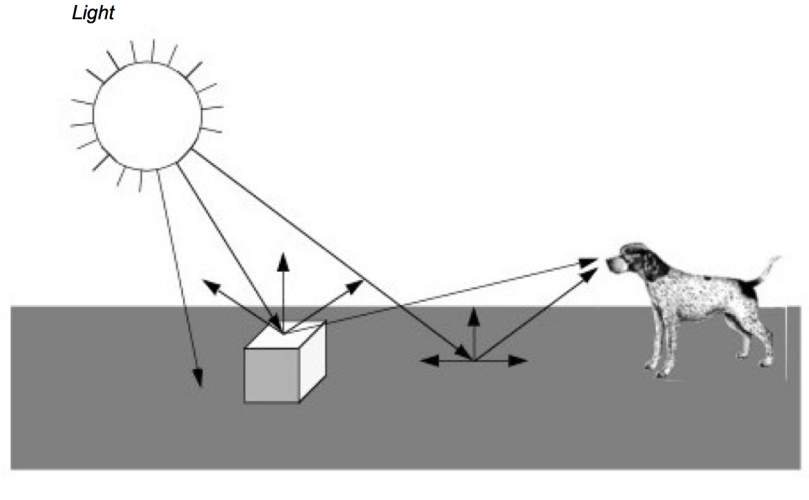
继承：

## 计算机图形学

1. **渲染的物理过程**

计算机图形学是数据可视化的基础。可视化是将数据转换为一组图元(点、线、面)的过程。然后使用计算机图形学的方法将这些基元转换为图片或动画。

计算机图形学是使用计算机生成图像的过程。我们称这个过程为 rendering。将图形数据转换为图像的过程。渲染过程有多种类型，从 2D 绘制程序到复杂的 3D 技术。

****

**物理世界的可视化映射**

3D 计算机图形学的一种常见且有效的技术称为光线跟踪或光线投射。渲染过程可以分为两类：image-order 和 object-order。光线追踪是一个图像顺序过程。

1. **表面渲染与体积渲染**

面渲染：当我们渲染一个物体时，我们正在观察物体的表面以及它们与光的相互作用。如桌子、椅子，我们会使用曲面描述（如点、线、三角形、多边形或 2D 和 3D 样条）对对象进行数学建模。对象的内部没有被描述。

体渲染：但是，云、水和雾等常见对象是半透明的，或者会散射穿过它们的光线。体积渲染技术使我们能够看到对象内部的不均匀性。光线不仅与物体表面相互作用，还与内部相互作用。

1. **颜色**

人类可见的电磁波谱包含约 400 至 700 纳米的波长。我们用来描述颜色的两个简化组件系统是 RGB 和 HSV 颜色系统。

* RGB 系统根据颜色的红色、绿色和蓝色强度表示颜色。
* HSV 系统根据颜色的色调、饱和度和值来表示颜色。

1. **灯光**

控制渲染过程的主要因素之一是光线与场景中角色的交互。如果没有灯光，则生成的图像将是黑色的，并且相当没有信息。

1. **表面属性**

光线与 actor 的表面相互作用以产生颜色。这种结果颜色的一部分实际上不是由于直射光，而是由于从其他对象反射或散射的环境光。局部光源与无限点光源。RGB 颜色和 alpha 值。

1. **相机**

我们有发射光线的光源和具有表面属性的 actor。在 actor 表面的每个点上，这种交互都会产生一些复合颜色（即，来自光、对象表面、镜面反射和环境效果的组合颜色）。好比人的眼睛，将三维场景投影到二维的平面上。

## 可视化管线

使用简单的数学模型创建了图形图像，用于照明、查看和几何图形。照明模型包括环境、漫反射和镜面反射效果。光线从光源传播到actor，在那里被反射和散射。其中一些光线被相机捕获并产生 2D 图像。

vtkProp、vtkAbstractMapper、vtkProperty、vtkCamera、vtkLight、vtkRenderer、vtkRendererWindow、vtkRenderWindowInteractor、vtkTransform、vtkLookupTable等等类都是与数据显示或渲染相关的，它们构成了**VTK的渲染引擎(Rendering Engine)。**

可视化管线是指用于获取或创建数据、处理数据以及把数据写入文件或者把数据传递给渲染引擎进行显示，这样的一种结构在VTK里就称为可视化管线，可视化管线(Visualization Pipeline)。

Renderwindow

Render

Actor

Mapper

Filter

Source

显示

显示

控制调节可视化数据的属性：  
颜色，透明度...

数据映射成可视化的数据

修改数据：  
差值，

下采样…

数据源：

球体

圆柱体

# 环境的部署

## 环境

操作系统：ubuntu 24.04

QT版本：6.7.3

VTK版本：VTK-9.3.1

编译工具：cmake

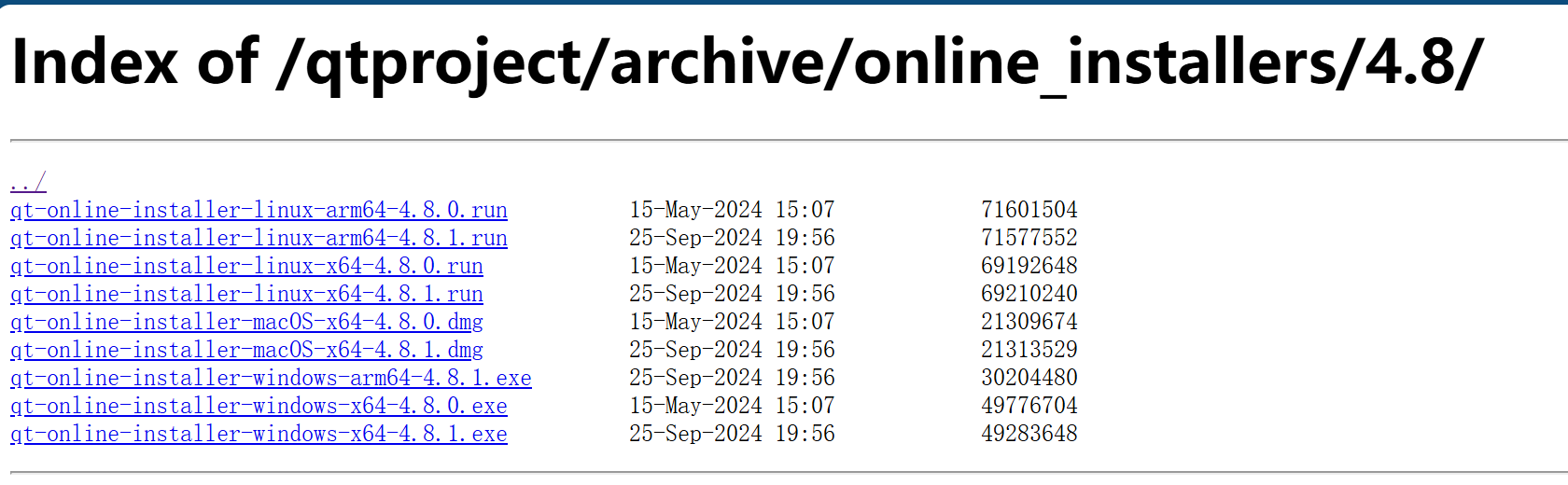
## 步骤

### 虚拟机安装操作系统

略

### 下载并安装QT

1. **下载QT在线安装包（以最新版本为例）：**<http://mirrors.ustc.edu.cn/qtproject/archive/online_installers>



1. **安装依赖(除了下面的依赖，安装过程中如果还有缺失，就补充安装！)**

*安装 libxcb-xinerama0 库，这个库提供了与 X11 服务器相关的 Xinerama 扩展支持。Xinerama 是一个允许将多个显示器连接并扩展到一个虚拟桌面的 X11 扩展。Qt 在多显示器环境下运行时可能需要这个库，特别是在多显示器显示、窗口管理以及多屏显示相关的操作中。没有这个库，Qt的某些图形界面和窗口可能不能正确显示在多个屏幕上。*

sudo apt-get install libxcb-xinerama0

*安装 libxkbcommon-x11-0 库，这是一个为 X11 系统提供键盘输入相关功能的库。它是 XKB（X Keyboard Extension）协议的实现之一，主要用于处理键盘布局和按键事件。在 Qt 应用程序中，处理用户的键盘输入事件时，可能会用到这个库，尤其是在支持多种键盘布局和按键映射的情况下。这个库对于确保输入事件的正确处理非常重要。*

sudo apt install libxkbcommon-x11-0

安装 libxcb-icccm4 库，提供了与 ICCCM（Inter-Client Communication Conventions Manual）协议的支持。ICCCM 是一个用于 X11 客户端和窗口管理器之间通信的协议，负责窗口的属性、行为等管理。Qt 使用这个库来进行窗口管理，特别是在窗口的操作（如关闭、最小化、最大化、对齐等）以及与 X11 服务器和窗口管理器的交互时。没有这个库，Qt可能无法正常管理或响应窗口操作。

sudo apt install libxcb-icccm4

安装 libxcb-image0 库，这个库提供了对图像扩展的支持，允许应用程序处理图像数据和 X11 服务器的图像接口。Qt 使用这个库来加载和显示图像，特别是在使用 X11 显示系统时，图像的加载、处理和显示都可能依赖此库。例如，Qt的图形界面显示涉及到许多图像元素，包括按钮、背景、图标等。

sudo apt install libxcb-image0

* build-essential：这是一个包含了构建软件所需的基本工具和库的元包。它包括 GCC 编译器、make 工具、以及其他一些编译和构建过程中的必要工具。用于安装 Qt 及其依赖项时，通常需要编译某些源代码，build-essential 包含的工具集为这一过程提供支持。*

* libgl1-mesa-dev：这是一个用于开发与 OpenGL 相关的图形应用程序的开发包，包含了 Mesa 实现的 OpenGL 库的头文件和静态链接库。Mesa 是一个开源的 OpenGL 实现。Qt 中的一些模块，尤其是与图形相关的模块，可能需要 OpenGL 来实现硬件加速渲染。这个库提供了对 OpenGL 图形加速和渲染的支持，确保 Qt 应用可以高效地使用图形硬件来渲染界面。*

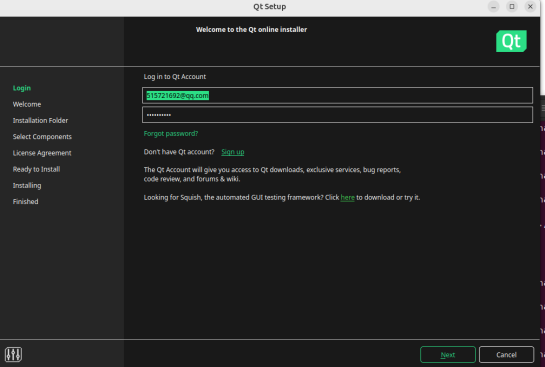
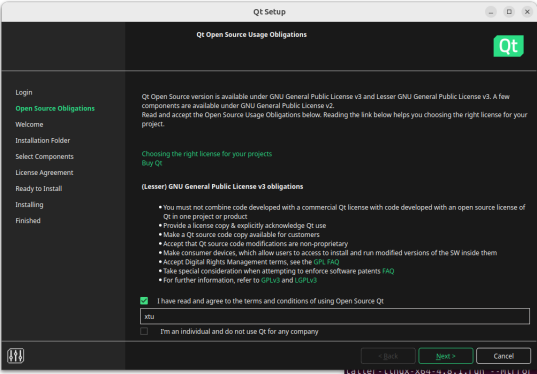
sudo apt-get install build-essential libgl1-mesa-dev

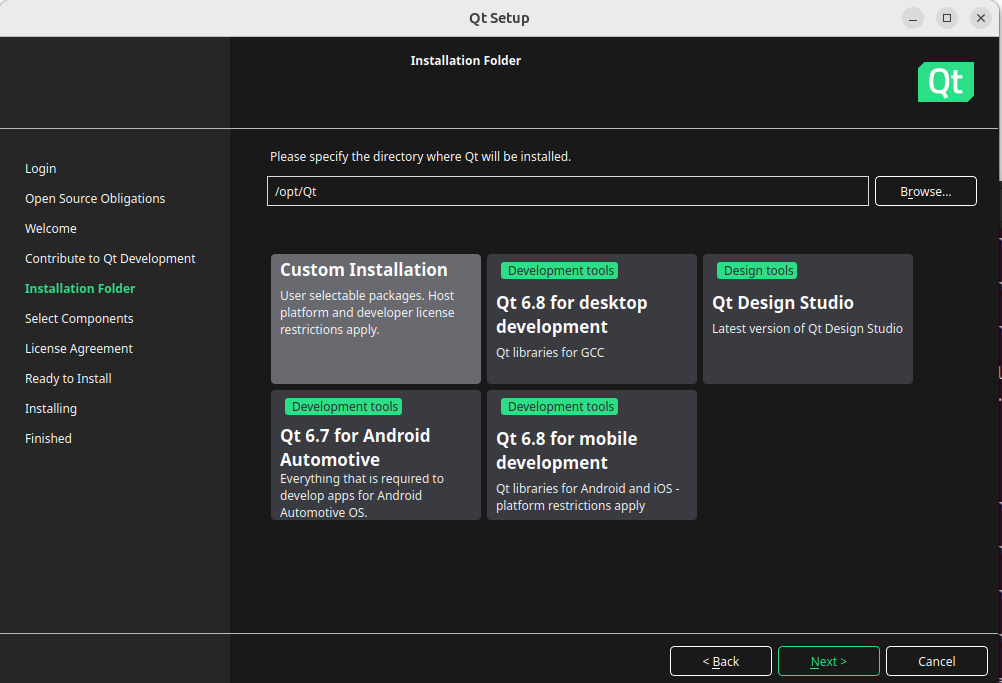
1. **修改安装包操作权限，+x**：给文件添加执行权限

sudo chmod +x qt-online-installer-linux-x64-4.8.1.run

1. **运行在线安装程序，等待安装完成。注意，国内安装一定要指定其他的源，默认源会非常慢！**

sudo ./ qt-online-installer-linux-x64-4.8.1.run --mirror *http://qt.mirror.constant.com*

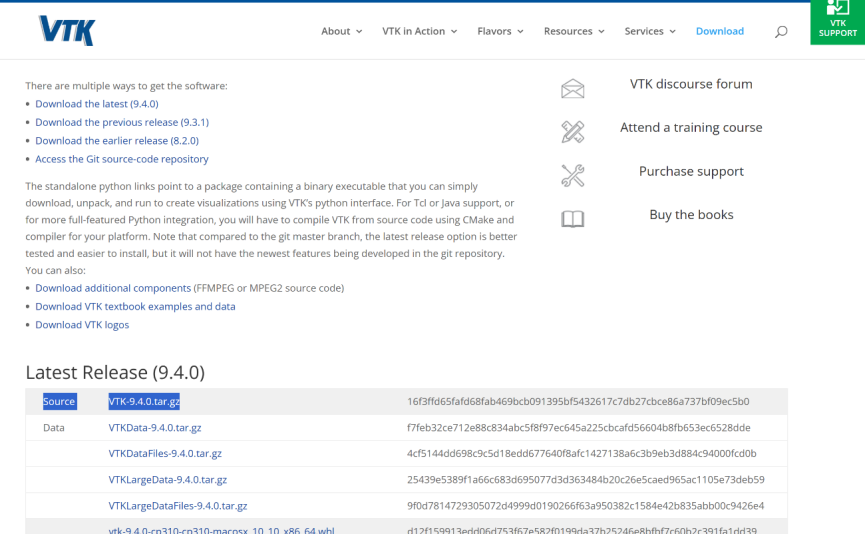
 



### 下载并安装VTK

1. **下载VTK，这里我下载的是最新版本9.3.1**

**官方下载地址：**[Download | VTK](https://vtk.org/download/) https://vtk.org/download/



1. **解压VTK源码**

tar -xzvf VTK-9.3.1.tar.gz

1. **安装ccmake和VTK依赖库**

sudo apt update

sudo apt install build-essential cmake cmake-curses-gui mesa-common-dev mesa-utils freeglut3-dev g++ qtbase5-dev libgl1-mesa-dev

1. **进入VTK源码目录配置VTK的编译环境**

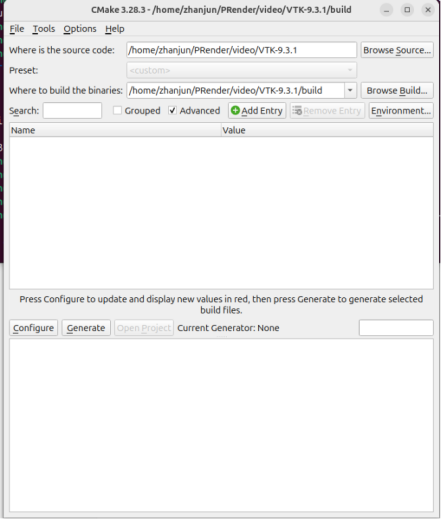
**在源码目录打开命令行终端，创建build文件夹，并进入文件夹。**

mkdir build

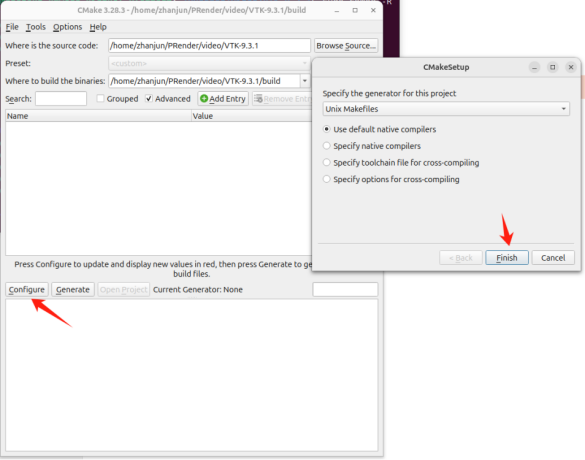
cd build

1. **启动cmake-gui配置编译选项**

cmake-gui

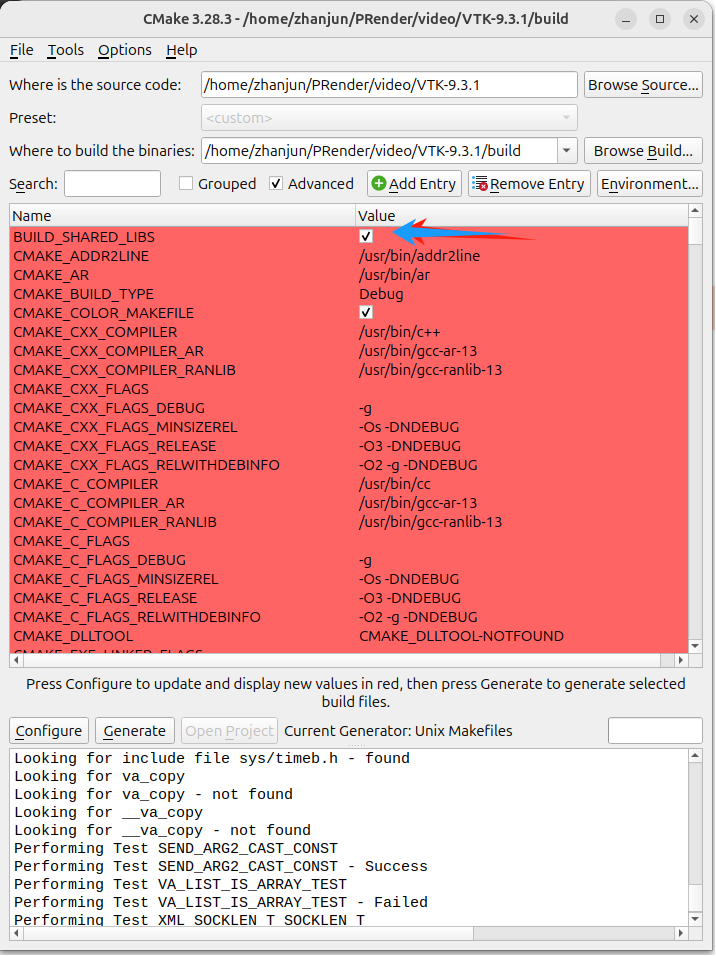
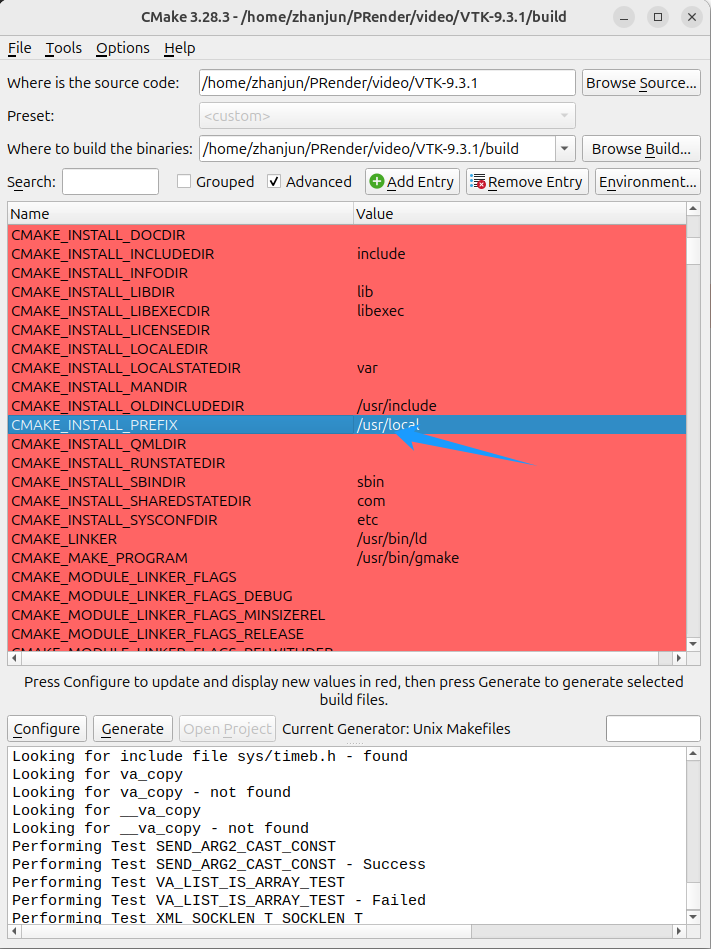


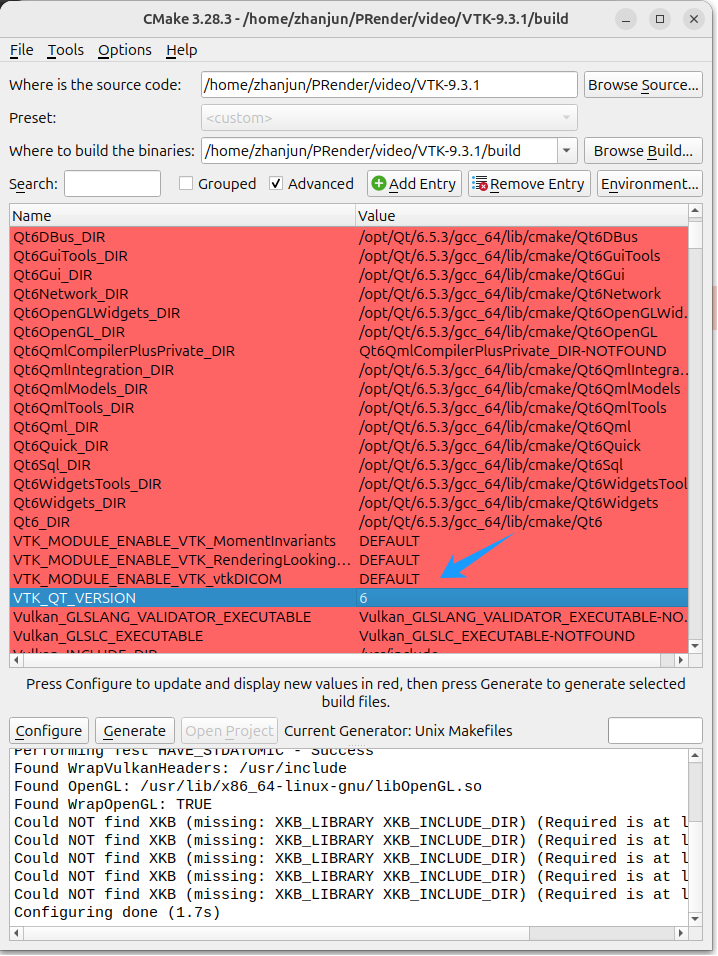
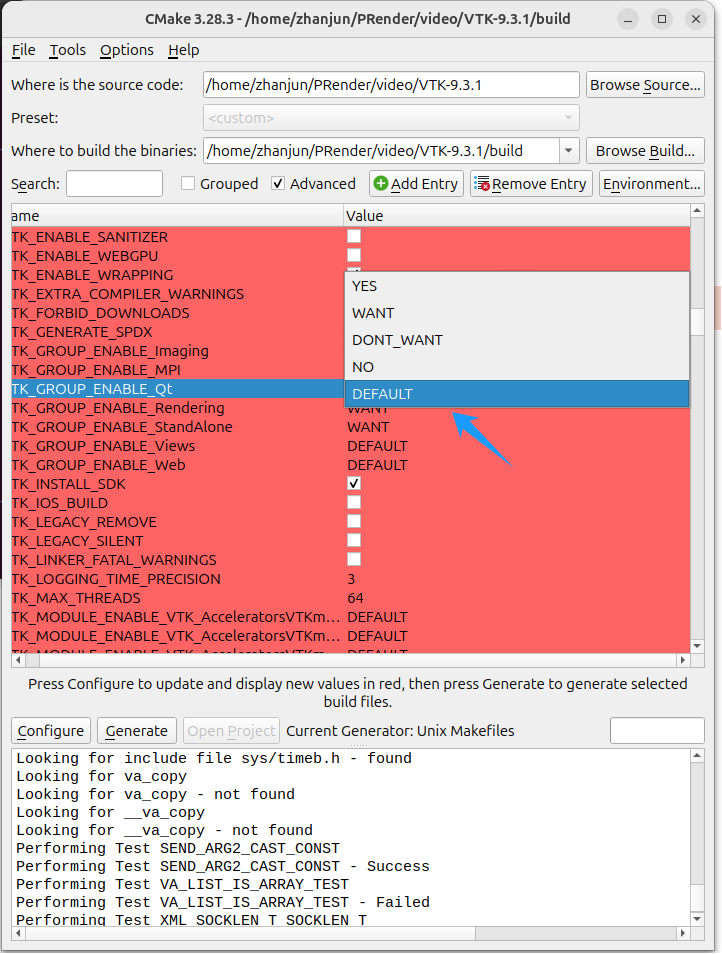
**点击configure**



**点击finish**

* BUILD\_SHARED\_LIBS：True；
* CMAKE\_BUILD\_TYPE：Release；
* CMAKE\_INSTALL\_PREFIX：/usr/local；
* VTK\_GROUP\_ENABLE\_QT：Yes；

**修改**VTK\_QT\_VERSION**，将其调整为**6**即可，即默认使用QT6版本。完成后，点击configure，configure成功后，再点击Generate，至此，**完成VTK9.3的编译环境的配置**。**

1. **启动编译安装**

sudo make -j8

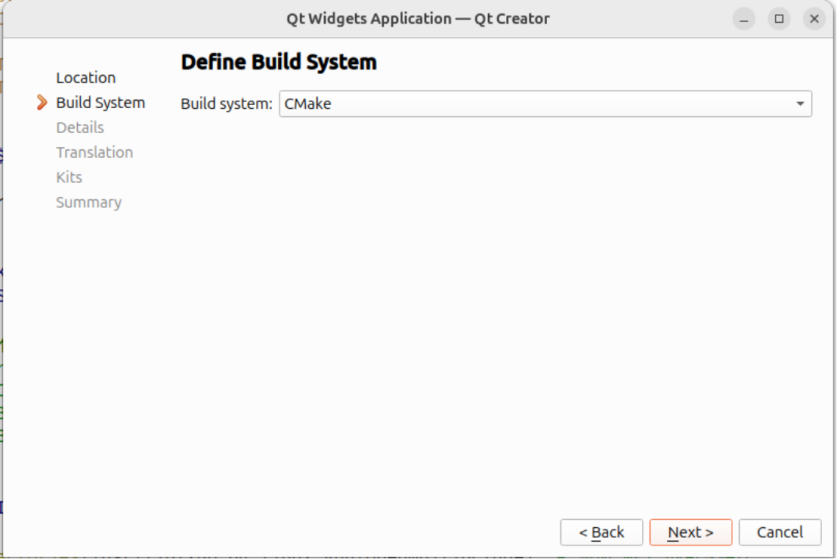
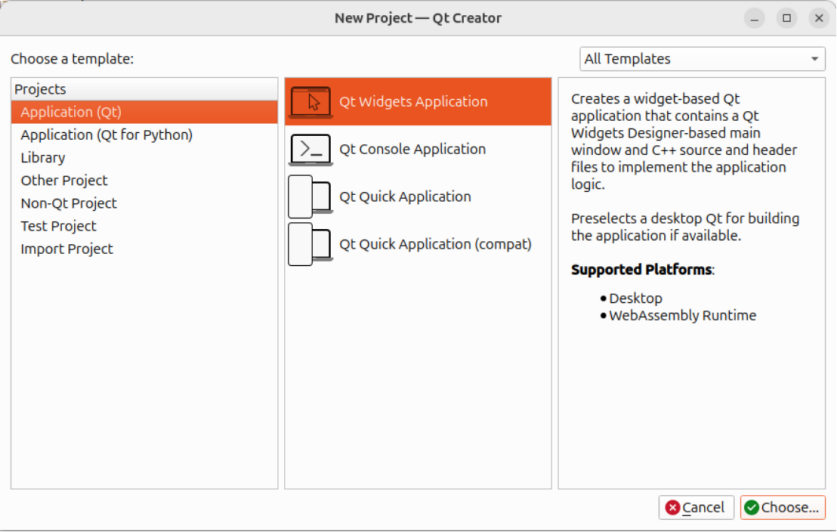
sudo make install

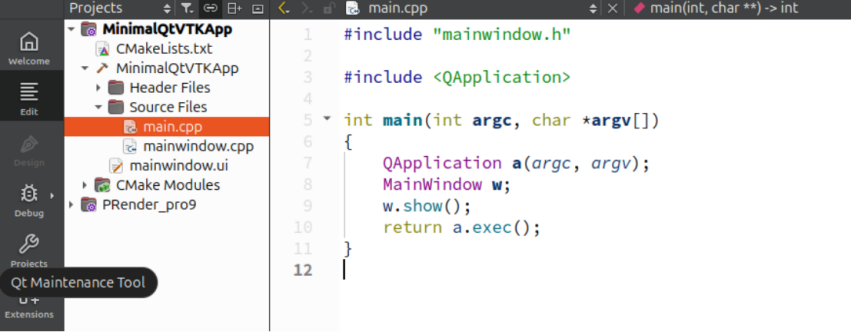
### 测试安装

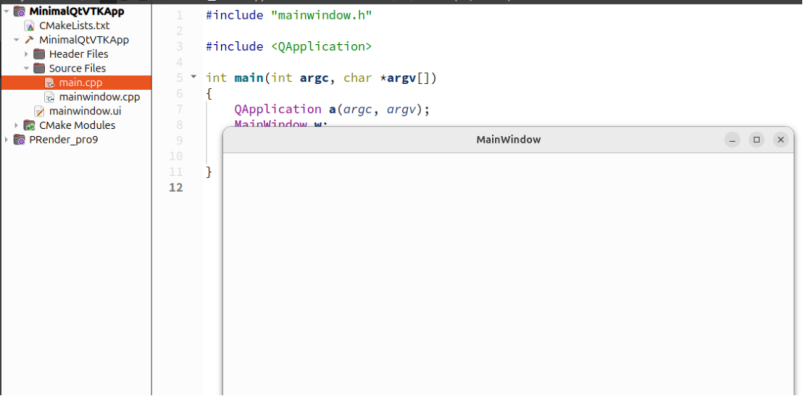
1. **下载测试代码**

<https://examples.vtk.org/site/Cxx/Qt/MinimalQtVTKApp/>

1. **打开QT创建工程**







1. **修改cmakelist.txt配置文件**

* LANGUAGES CXX C 指定项目使用的编程语言是 C++（CXX）和 C（C）；

project(MinimalQtVTKApp VERSION 0.1 LANGUAGES CXX C)

* 并查找必要的组件

find\_package(Qt${QT\_VERSION\_MAJOR} ...) 自动根据当前的 Qt 版本（QT\_VERSION\_MAJOR）来选择合适的 Qt 版本，并查找必要的组件：Widgets、Core、Gui、OpenGL 和 OpenGLWidgets。

* 设置 VTK 的路径，设置 VTK 库的安装路径。这里是一个示例路径 /usr/local/include/vtk-9.3，可能需要根据系统的实际安装位置进行修改。

set(VTK\_DIR /usr/local/include/vtk-9.3)

* 查找 VTK 包

# 查找 VTK

find\_package(VTK REQUIRED)

include(${VTK\_USE\_FILE})

* 查找 MPI（消息传递接口）库

# 查找 MPI

find\_package(MPI REQUIRED)

if(MPI\_FOUND)

include\_directories(/usr/lib/x86\_64-linux-gnu/openmpi/include) # 添加 MPI 的包含路径

# 这里可以根据你的需求显式指定库路径

link\_directories(/usr/lib/x86\_64-linux-gnu/openmpi/lib)

endif()

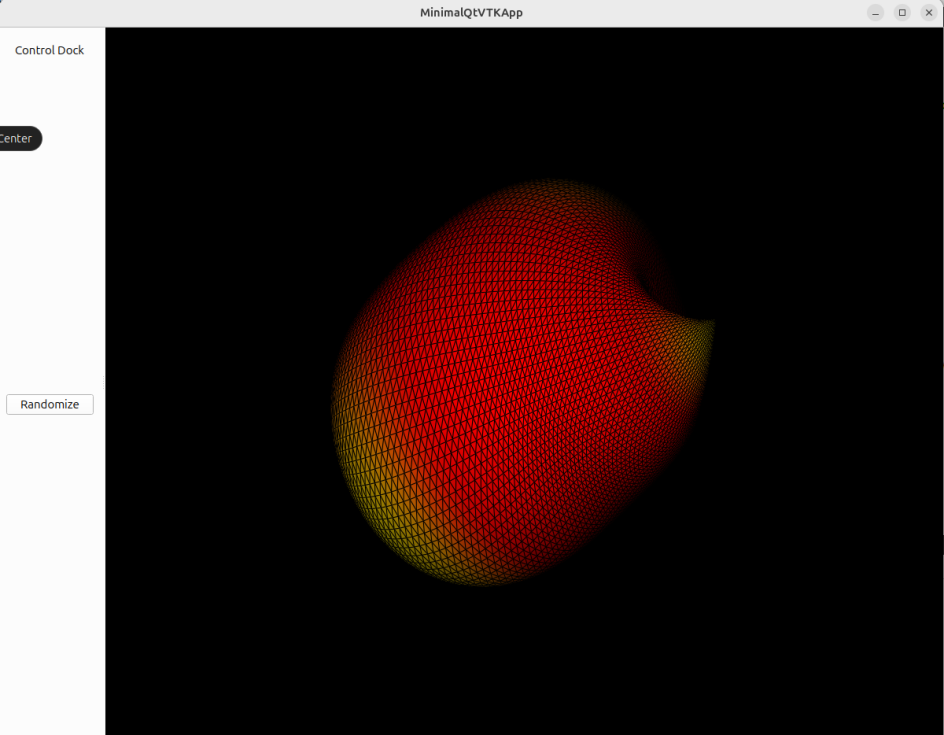
* 链接 Qt 和 VTK 库，target\_link\_libraries() 命令将 Qt 的 Widgets 模块和 VTK 库链接到目标 MinimalQtVTKApp。

target\_link\_libraries(MinimalQtVTKApp PRIVATE Qt${QT\_VERSION\_MAJOR}::Widgets ${VTK\_LIBRARIES})

1. **创建测试代码（见视频）**

**MinimalQtVTKApp.cxx**

1. **编译运行**



# 编程详解