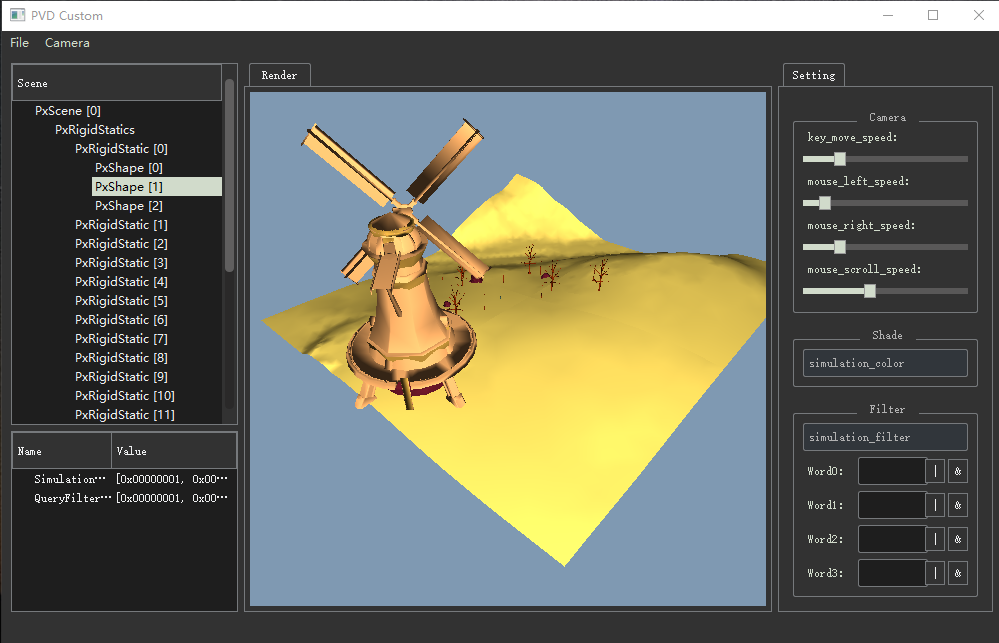
pvd\_custom文档

## 背景

服务端场景会用到physx库来进行碰撞检测，在physx中，场景由许多个actor组成，而每个actor又是由多个shape组成，physx提供了一种配置：可以将shape设置为共享模式来达到节省内存的目的，而这样会产生一个副作用：在pvd里面显示的时候，所有的RigidStatic都会显示为红色，这样会严重影响在通过可视化调式时观察两个shape是否发生碰撞的过程，所以有必要在shape设为共享模式的时候，同样也能将这些shape进行区分显示。由于pvd未开源，因此需要单独开发一个physx相关的可视化工具，该工具的核心就是能将这些被设置未共享模式的shape显示成自定义的颜色。

## 简介

该工具可以作为pvd的辅助调式工具，主要用来解决一些在physx中需要对不同shape进行不同的着色显示来调式他们之间的碰撞关系的问题。采用的第三方库为phyx3.4和qt5.12，主要用到physx库中的序列化api（用于加载服务端dump出的.col\_xml场景文件）和射线检测api（用于鼠标点选物体），qt则作为界面显示。



如上图，该工具的布局参考了pvd的界面，左边栏为加载出来的场景树，点选树中的shape节点，可视化窗口中的相机会自动定位到相应的shape位置，此外也会显示当前shape的相应的一些属性，而右边栏为一些自定义的相机控制参数，以及自定义filer后与shape中的filter（simulationFilter和queryFilter）进行位操作，将得到的结果进行颜色查表，从而对shape进行相应的着色。

## 实现

shape数据结构中定义了一些可以表示其几何信息的字段数据，其中geometry类型字段定义了该shape所对应的几何体类型以及相应的大小，而geometry类型分为box、sphere、capsure、triangleMesh、convexMesh、heightField以及plane。这里实现了除heightField和plan类型外的其它geometry类型（geometry对应一个mesh，包含vertices数据以及相应的indices数据）生成算法，并采用blinn\_phong光照模型对其进行渲染着色。

程序模块上分为相机模块、渲染模块、物理模块、场景模块以及UI模块。

相机模块：可以在第一人称以及第三人称之间进行切换的相机（按住鼠标左键和右键，分别会滑动触发第三人称和第一人称的视角移动），此外还能通过WSAD键来控制相机位置的移动。

渲染模块：采用qt封装的opengl函数，定义了一个渲染引擎单例类来提供一些opengl相关的api接口，抽象出着色器程序类并继承该类后实现blinn-phong着色程序类，定义一个渲染对象类来包含一些渲染需要的缓冲数据（vertexBuffer和indexBuffer，以及对应的color和transform）。

物理模块：定义了一个物理引擎单例类来提供一些physx相关的api接口，而且定义了一个物理对象类来包含相应的shape指针及其对应的父actor和filterData数据

场景模块：定义一个场景对象类，该对象包含渲染数据（渲染对象）和物体数据（物理对象），加载场景文件时生成包围盒用于初始的相机定位，并提供一些用于渲染shape和射线检测shape的方法。

UI模块：定义一些场景数据和属性显示控件以及一些参数设置控件。