

详解软件中的拾取(选择)功能

原创 邓子平 [多物理场仿真技术](#)

收录于合集

#工业软件 2

#CAD 9

#CAE 9

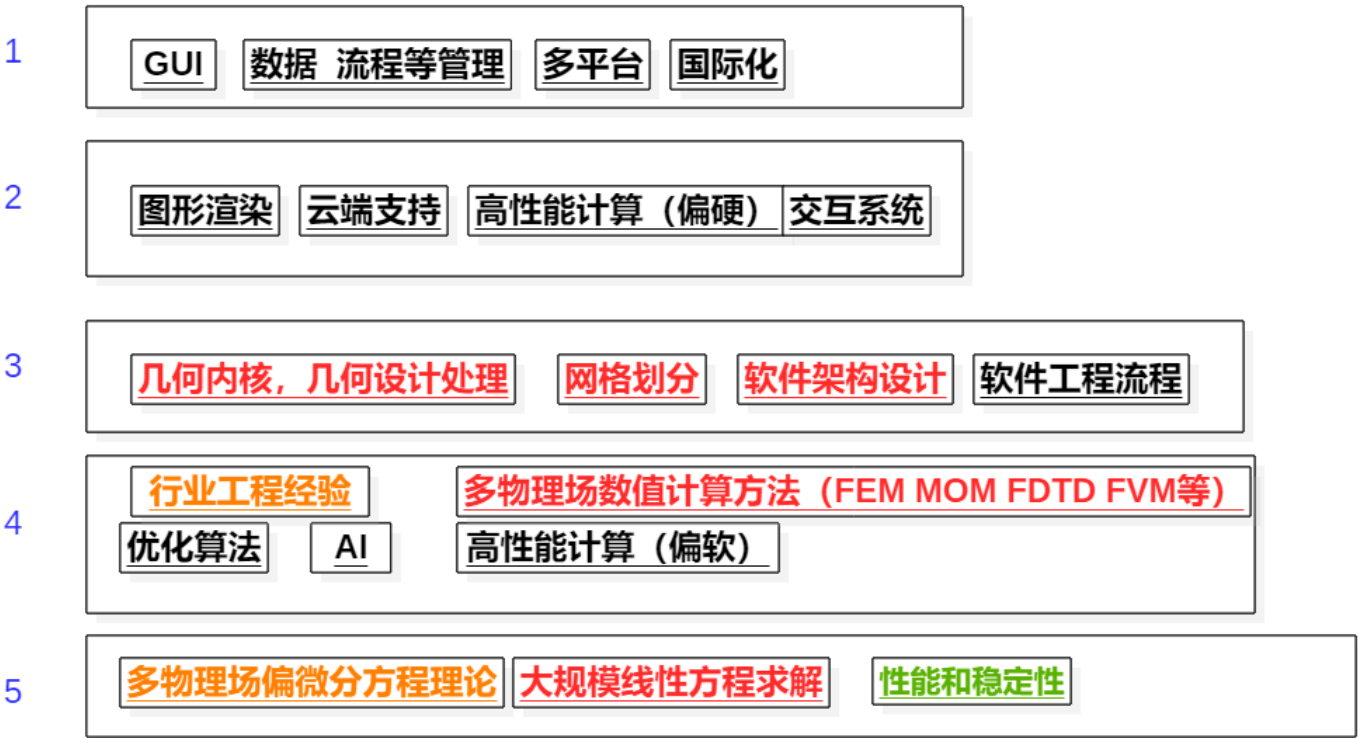
#软件架构设计 2

#多物理场仿真 2



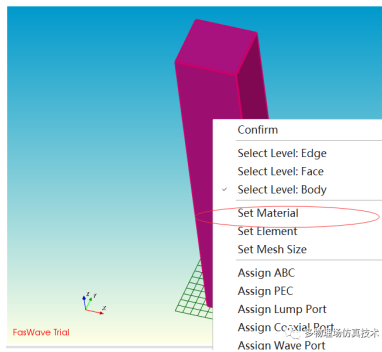
注：本文针对软件研发，全文约2500字

模型视图里的拾取(选择)操作，是工业软件(设计仿真)里上层应用的一个基本功能，属于下图中“交互系统”的一部分。



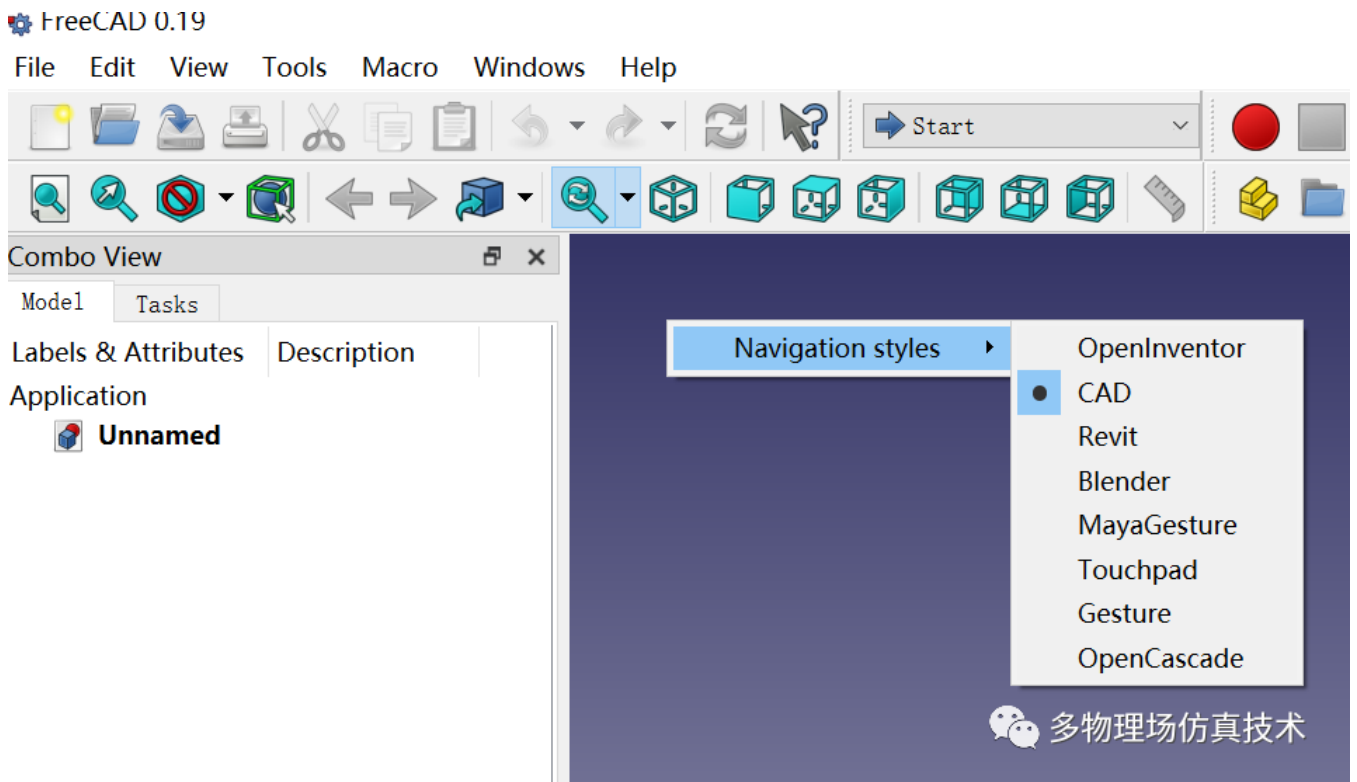
多物理场仿真技术

比如在三维模型视图里，我们可以选中一个对象赋材料；选中面或边，赋荷载和边界条件；选中两个对象执行布尔运算；编辑一条NURBS曲线时，选择一个控制点进行拖动等等。



拿到一款软件，首先体验拾取交互功能，每种软件的交互操作不太一样，好的拾取交互能简化操作，提高工作效率。

开源CAD软件FreeCAD里提供了对多种软件交互模式的切换。如下图：



1.基础功能

从研发角度看，功能完全的拾取通常包括如下内容：

1. 拾取一个三维对象(二维对象可以视为三维的一种特殊情况)
2. 拾取三维几何对象上的面，线和点
3. 点选(单选)，即鼠标点一下选中对象
4. 点选(多选)，一般按住Ctrl+单选，可以选中多个对象
5. 框选，按住鼠标拖动画出一个方框，方框内的选中。框选又可以分为部分框选选中，和全部框选才算选中两种
6. 反选，即物体选中后，再次执行选中操作即设置为不选中
7. 拾取通常不会单独发生，会有其它事件触发，比如模型创建，对象高亮显示
8. 当模型有前后遮挡，需要选中被挡模型
9. 有过滤器，可以使用过滤器选中特定属性的对象

10. 对外提供统一的拾取接口

11. 处理大模型时，要解决性能瓶颈问题，参考[工业软件研发中处理超大模型\(4\)](#)(点击链接查看)

2.拾取原理:

根据软件技术选型，拾取原理有以下几种：

1. 在创建模型的时候，将模型和屏幕像素建立映射关系，当进行拾取时，计算鼠标所在像素位置，再查找映射关系。这种操作偏底层，一般是渲染引擎做的事情，比如OPENGL里的模板测试。
2. 如果仅使用渲染数据，在视图上，从鼠标位置垂直视图发射一条射线，计算射线和视图对象的位置关系，相交则代表拾取到。
3. 操作系统处理鼠标事件时，鼠标的坐标位置一般是屏幕坐标(局部坐标)，也就是像素值，当使用了三维几何内核时，可以将位置坐标转为世界坐标，也就是真实坐标，从鼠标所在世界坐标位置发出垂直视图的射线，计算和真实模型位置的相对位置关系，如果相交则代表拾取到。
4. 框选逻辑和单选类似，只不过计算的是长方形和物体的位置关系。由于框选一般是针对多个对象，所以计算量偏大，需要进行简化或者必要的过滤操作，比如将三维投影到二维，对象进行包围盒过滤等。

其它：

1. 可以设置计算容差，即射线和物体即使没有真实相交，但距离在一定容差范围内，也可以认为相交。
2. 以上拾取原理不仅适用于三维CAD建模，对于纯曲面建模，前处理，后处理，网格编辑等操作也都适用。


3. 鼠标响应

鼠标事件由操作系统发出，在一般的UI开发工具中都有鼠标响应事件，以QT为例，QT的QWidget提供了事件入口函数：

```
virtual void event(QEvent *event);
```

event事件中又提供了各种鼠标事件处理函数，包括鼠标按下，释放，移动，双击，滚动中轮：

```
virtual void mousePressEvent(QMouseEvent *event);
virtual void mouseReleaseEvent(QMouseEvent *event);
virtual void mouseDoubleClickEvent(QMouseEvent *event);
virtual void mouseMoveEvent(QMouseEvent *event);
#ifdef QT_NO_WHEELEVENT
virtual void wheelEvent(QWheelEvent *event);
#endif
```

 多物理场仿真技术

QT部分源码：

```

bool QWidget::event(QEvent *event)
{
    Q_D(QWidget);

    // ignore mouse and key events when disabled
    if (!isEnabled()) { ... }
    switch (event->type()) {
    case QEvent::MouseMove:
        mouseMoveEvent((QMouseEvent*)event);
        break;

    case QEvent::MouseButtonPress:
        mousePressEvent((QMouseEvent*)event);
        break;

    case QEvent::MouseButtonRelease:
        mouseReleaseEvent((QMouseEvent*)event);
    }
}

```

我们可以将选择逻辑放到对应函数中处理，针对不同的视图可以重写不同的选择逻辑。为了提供统一接口，可以将模块设计成QWidget和待选对象作为传入参数。

4.业务区分

虽然不同技术选型的拾取原理大同小异，但是针对不同业务，拾取的设计上还是有很大区别的：

1. 针对CAD软件，尤其是三维CAD软件，三维几何模型和模型渲染单元之间需要一直存在映射关系，**也就是通过几何对象可以找到渲染对象，通过渲染对象可以找到几何对象**。所以在设计拾取的时候，选中拾取一个即可；但是如果拾取渲染对象，需要映射两次，即渲染对象映射到几何，在几何修改后，再次更新渲染对象；拾取几何的话，更新渲染对象即可；CAD设计软件涉及到大量几何对象创建，编辑，参数化，约束求解，模型更新驱动，**设计上偏向于拾取几何对象**；
2. 仿真软件一般建模功能较弱，主要在前处理上用以生成求解模型。所以拾取设计上没有太多讲究，设计实现上怎么方便怎么来；
3. 前处理软件大部分交互操作在**面和线**上，尤其是曲线曲面有精度要求，几何精度使用double型，而渲染数据一般使用float型，所以使用几何拾取更准确；
4. 网格编辑一般是编辑面片，面片只有几何信息，没有BREP拓扑数据，针对渲染数据拾取更容易操作。

5.过滤器设计

过滤器是选择操作中一个基本功能，过滤器有几何上的，功能上的，也有业务上的：

比如模型中有多个对象，其中有三维，二维和一维对象，我们希望仅选择三维实体，不参与仿真和不可见的对象不参与拾取；同一个对象，如果是一个三维实体，仅希望选择面用以赋边界条件；多个物体被选择而且相互遮挡，希望选择最后面的一个；实体显示模式为线框模式，但是希望单选到面上的时候也被选中。

过滤器可以单独设计成一个类，执行相关过滤操作，枚举出所有可能的选项和模式，也可以作为参数传入到拾取模块中。

6.性能问题

拾取操作能有效检验一个软件性能。

拾取操作需要确定射线和对象几何的关系，**一般是线线求交，线面求交**，对于非解析参数几何需要进行大量迭代计算，这种计算本身是比较耗时的，如果模型对象较多，则会存在性能瓶颈。

1. 当模型或者场景视图比较固定，我们可以将三维模型投影到二维，二维模型的位置计算量要远小于三维；
2. 当对象较多，而且比较分散，则可以使用树结构，常用的八叉树，BSP树，R树，参见[深入剖析三维几何内核\(7\)--一般技术](#)(点击链接查看)，通过树结构，可以把拾取的候选对象保持在一个较低的数目；
3. 包围盒计算效率很高，在实际计算中，可以通过包围盒对对象进行过滤；**除了常用的AABB包围盒，有条件可以使用紧包围盒**；
4. 对于复杂大对象的拾取，比如汽车CFD分析中表面NURBS曲面的拾取，可以通过降维取得面上所有参数点，构造树结构。实体也可以通过分割等操作变成小模型，**这也是为什么在模型操作中，要尽量避免不必要的合并**。

5.拾取是一个实时性很强的操作，响应时间一般小于200毫秒，用户才不会觉得卡顿，要达到这个目的一方面计算内容不能太多，另一方面通信开销要尽可能小，所以计算中不太推荐多线程；多线程可以用来监听，更新处理各种状态。

7.关键字

可以看出，拾取功能的设计还是比较依赖技术选型。

最后总结一下拾取涉及到的一些关键字

向量 射线 几何投影 图元对象 几何对象

求交运算 线面求交 线线求交 拾取模块

点选 框选 反选 深度计算 模型分割

几何映射图元 过滤 鼠标事件 包围盒 容差

局部坐标 世界坐标 坐标转换 八叉树 BSP树 R树

阅读: null

在看: null