

Java3D 鼠标交互功能的扩展与应用

魏 东¹ 刘 童² 余冠华² (1.合肥工业大学 计算机学院 安徽 合肥 230009;
2.沈阳工业大学 信息学院 辽宁 沈阳 110178)

摘 要: 分析了 Java3D 提供的鼠标交互功能, 针对场景比较复杂, 待拾取物体相对距离近的虚拟环境中用户拾取不准的问题提出了一种解决思路: 在鼠标移动过程中实时创建一条连接空间中的视点与投影平面上的鼠标点的射线, 检测沿射线的方向是否有物体与之相交, 若有则返回距离视点最近的与射线相交物体所在的节点, 并在该节点上添加一个动画效果提示用户该节点是预选节点。重点讲解了具体的实现过程, 在最后的部分结合实际应用背景验证了该方法可以有效地方便用户的拾取操作。

关键字: 计算机图形学; 计算机仿真; 虚拟现实; 人机交互; 拾取

Extension and Application of Enhanced Interactive Mouse Functions in Java3D

WEI Dong¹, LIU Tong², YU Guan-Hua²

(1. Hefei University of Technology, Hefei 230009, China; 2. Shenyang University of Technology, Shenyang 110178, China)

Abstract: This paper has analyzed the build-in mouse interactive function of Java3D to solve the problem that user cannot pick the objects which are too closed to each other accurately. A solution is proposed. First, a ray can be built that connects the viewpoint and the mouse point in the imageplate as the mouse is moved. Then, the ray can be checked if it intersects with any object. If yes, we can return to the object intersecting the ray and closest to the viewpoint. Finally, we can add an animation effect on the node to remind the user that this node is a pre-selected node. This paper focuses on explaining the realizing process of this method. In the final part, a practical application verifies that this method can facilitate the user in mouse picking effectively.

Keywords: computer graphics; computer simulation; virtual reality; human-computer interaction; picking

1 引言

Java3D 是 Java 语言在三维图形领域的扩展, 利用 Java3D 提供的 API, 可以编写出三维动画、三维游戏等等。Java3D 本身已经提供了一些鼠标交互功能, 可以根据用户的操作进行实时的场景变换^[1]。但是现有的交互手段还比较有限, 通常需要进行扩展。邓文 0 A 生、马王俊美在参考文献^[1]中提出并实现了使用鼠标点取同一区域上的多个目标物体并自由选择具体操作对象的方案, Fabio.R.Mirand 等人开发的 VMMV 系统^[2]能够使用弹出式菜单完成对目标对象的操作。然而, 在实际应用中特别是在虚拟环境中存在

大量物体的情况下, 用户不仅面临着在纵向(垂直于显示屏幕)上相同区域存在多个重叠物体时产生的拾取不到的问题, 而且面临着在横向平面(平行于显示屏幕)上由于多个物体距离较近带来拾取误差而产生的拾取不准的问题。这里提出了一种应对方法, 即在用户发生点取操作之前提示用户将要选中的对象是哪个, 方便了用户的拾取操作。

2 拾取提示功能的总体思路和具体实现

2.1 拾取不准问题的产生与应对方法

如图 1 是对相邻的几个小立方体进行拾取的实

基金项目: 沈阳市科技局项目(1032033-1-03)

收稿时间: 2009-07-21; 收到修改稿时间: 2009-08-29

验。小立方体被选中后会变成绿色，用户希望选中右上角的小立方体，点击鼠标之后选中的却是相邻的另一个小立方体。这个问题是由 Java3D 的拾取误差导致的。在 Java3D 中拾取是这样一个过程：发生点击事件的时建立一条连接空间中的视点和鼠标在投影平面上的点击点的拾取射线并需要返回与射线相交的可点取对象^[3]。系统建立拾取射线时需要连接视点和鼠标点击点，视点的坐标是基于虚拟世界中原点的三维坐标，而从系统中获得的鼠标指针坐标是基于窗口右上角的 AWT 事件坐标，两个坐标处于不同的坐标系，这就需要将鼠标指针基于屏幕右上角的坐标转化为基于虚拟世界中原点的坐标，在这个坐标系转换中会不可避免地产生视觉上的误差和数值上的误差，于是带来了用户拾取不准的问题。

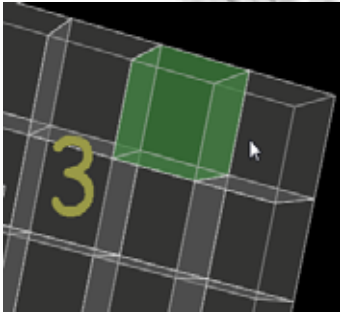


图1 错误地拾取到相邻的立方体

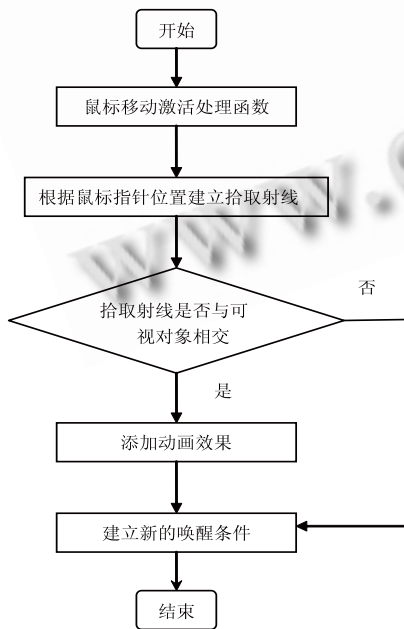


图2 拾取提示功能的流程图

应对方法是通过对预选对象进行提示避免用户选取出错的问题。具体来说就是：以鼠标移动事件作为激励信号(此时用户尚未发生点取操作)创建一条连接空间中的视点与投影平面上的鼠标点的判定射线，返回距离视点最近的可点取对象所在的节点。然后向该节点上添加一个动画效果以提示用户该对象为预选对象。以上过程如图 2。

2.2 鼠标交互功能扩展的实现原理

Java3D 提供的主要鼠标交互功能如表 1。已有的交互类均设置了 MouseCallBack 或者 Picking-Callback 为回调接口，为的是更新场景的位置、角度和尺度(即场景的 Transform3D 对象，因为位置、角度、尺度信息均包含在 Transform3D 对象中)。这里提出的方法是在鼠标的移动过程中对鼠标指针与虚拟场景中物体的位置关系进行实时的判断，需要对鼠标移动事件进行处理，但是与原有的鼠标交互类不同的是该方法不需要更新场景的位置、角度、尺度等信息，即无需设置 MouseCallBack 或者 PickingCallback 为回调接口，所以选择了对 Behavior 类直接进行扩展，扩展后的类关系如图 3。

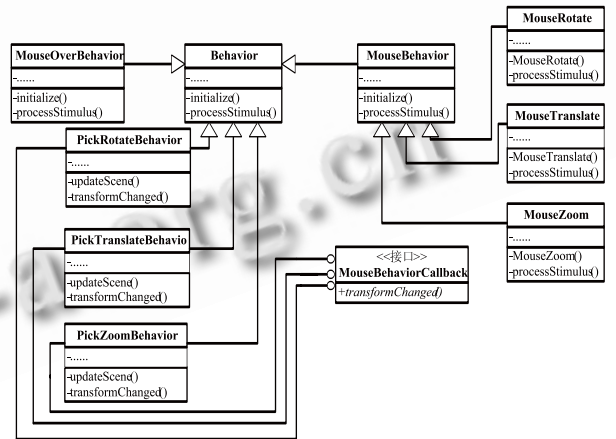


图3 扩展后各类之间的关系

表1 Java3D 原有的几个提供鼠标交互功能的类

类名	实现功能描述	回调接口
MouseRotate	左键控制对象旋转	MouseBehaviorCallback
MouseTranslate	控制对象在平行于图像板的平面上移动	MouseBehaviorCallback
MouseZoom	控制对象在与图像板正交的平面上移动	MouseBehaviorCallback
PickRotateBehavior	点取并旋转对象	PickingCallback
PickTranslateBehavior	点取并平移对象	PickingCallback
PickZoomBehavior	点取并缩放对象	PickingCallback

2.3 拾取提示功能的具体实现方法

Behavior 类是一个抽象类,对它进行扩展主要工作就是指定激励条件和编写针对该激励的反应函数。具体来说就是指定鼠标的移动事件为激励条件编写鼠标移动事件的处理函数并建立判定射线,在满足条件的节点上添加提示效果。

2.3.1 设置激励处理函数的唤醒条件

initialize 方法允许 Behavior 对象初始化其内部状态和指定其唤醒条件^[3]。包含 Behavior 对象的节点被创建的时候, initialize 方法被调用。在 initialize 方法中需要指定唤醒条件,否则激励事件的处理方法即 processStimulus 方法不会被执行。本方法以鼠标的移动事件作为激励,所以在 initialize 方法中给出唤醒条件为鼠标移动即 MouseEvent.MouseMoved 代码如下:

```
public void initialize(){
    this.wakeupOn(newWakeupOnAWTEvent(MouseEvent.MOUSE_MOVED));}
```

2.3.2 处理鼠标移动事件与添加动画节点

激励处理函数即 processStimulus 完成了该方法的核心工作。它接收并处理 Behavior 的激励信息并执行响应函数,建立新的唤醒条件,最后退出该方法^[4]。按照建立射线,返回相交物体所在的节点的总体思路,这个方法被设计为主要完成三部分工作:

激励事件的接收,即获取鼠标移动事件。

激励事件的响应,即建立射线,返回距视点最近且与射线相交的对象所在的 TransformGroup 并在对这个节点下添加一个动画效果证明该节点是预选节点。

建立新的唤醒条件,即下一次鼠标移动的时候再次触发这个处理过程。

为了获取鼠标指针的坐标,首先要获取鼠标事件。当有输入事件发生时激励响应函数 processStimulus 会自动记录这些事件并存入一个叫 criteria 的枚举类型变量中^[5]。要从中得到鼠标事件需要进行一系列的转换,转换过程如图 4。首先从枚举类型中得到一个对象将其转化成 WakeupOnAWTEvent 类型并赋值给该类型的对象,调用该对象的 getAWTEvent()方法返回包含一连串 AWT 事件的数组,最后从数组中取出第一个元素赋值给鼠标事件对象,至此获取鼠标移动事件的工作完成。



图 4 激励事件的处理过程

得到本次鼠标事件后既可以得到鼠标的坐标,建立判定射线,返回符合条件的节点。实现这样的功能需要建立一个拾取对象并调用它的 pickNode(SceneGraphPath sgPath,int node_type)方法,该方法根据节点类型 node_type 中指定的类型得到一个可点取的节点对象的引用^[5]。本方法需要获取被选中物体所在的 TransformGroup 所以类型指定为 PickObject.TRANSFORM_GROUP。pickNode 方法的第一个参数使用 PickObject 的另一个方法 pickClosest(int xpos,int ypos)的返回值传入,该方法以视点为起点建立一条从视点出发以沿鼠标坐标方向的射线并返回距视点最近的可拾取物体,其中鼠标指针坐标 xpos 和 ypos 从之前获得的鼠标事件对象中获得。最后把 pickNode 方法返回的值强制转化成 TransformGroup 类型并在该节点下添加一个动画效果表明这个物体是预选物体。最后在程序的末尾指定新的激励条件为鼠标移动事件。激励响应中主要函数说明如表 2。

表 2 主要函数功能说明

函数名	功能
Enumeration.nextElement()	从鼠标事件对象 criteria 枚举取出一个对象
MouseEvent.getAWTEvent()	从 WakeupOnAWTEvent 对象中反对 AWTEvent 数组
PickObject.pickNode()	返回与拾取射线相交一个可拾取的节点
PickObject.pickClosest()	返回与拾取射线相交距视点最近的 SceneGraphPath 对象
BranchGroup.detach()	去除 BranchGroup 节点下所有子节点
TransformGroup.addChild()	在 TransformGroup 节点下添加一个对象
MouseEvent.getX(),MouseEvent.getY()	分别获取鼠标指针基于屏幕右上角的 X 和 Y 坐标

3 在三维数独游戏中实际应用的效果

三维数独游戏是一种填数字游戏,由 9*9*9 个小立方体组成,在每个小立方体中填入数字并保证在 x,

y, z 三个方向上的每列中均没有相同的数值, 同时保证每宫中的数值不能相同。其三维布局如图 5。由于要在每个立方体中填入数字而且立方体距离较近, 所以不可避免地出现用户点取不便的现象。使用上述的解决方法, 用户把鼠标指针移动到某个可点取的小立方体时该立方体显示一个渐变的动画效果告知用户哪个小立方体会将会被选中, 方便了用户的操作。

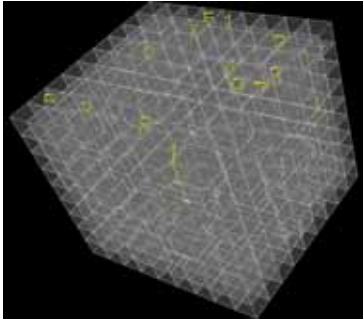


图 5 三维数独游戏程序

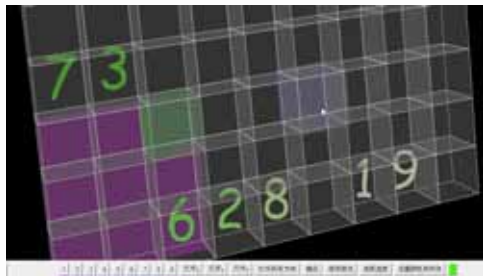


图 6 拾取提示功能在三位数独程序中的应用效果

如图 6 是被放大的三维数独 z 轴方向上 9 个平面中的一个平面。未被选中的立方体为白色半透明状, 被选中的立方体变成绿色, 用户单击下方的数字按钮

可以向绿色的立方体中填入对应的数字。用户移动鼠标到另外一个位置时鼠标指针对应的预选立方体表现为不断变化的半透明蓝色, 若用户此时点击鼠标则该蓝色立方体将被选中变成绿色, 与此同时原来的绿色立方体变回白色半透明装。实际操作中这种方法可以方便用户进行拾取, 提高拾取的正确率。

4 结语

分析了 Java3D 现有鼠标交互功能, 在此基础上对虚拟场景中鼠标的交互功能做出了扩展, 提出了一种能够在用户发生点取操作之前提示用户哪个物体将被选中的方法并给出了实现过程。在三维数独程序中的应用证明这种方法提高了用户同场景交互的灵活性, 方便了用户的拾取操作, 有一定的参考价值。但是, 这种方法没有从根本上解决 Java3D 产生拾取误差的问题, 未来研究工作的重点是对 Java3D 的拾取机制进行进一步的研究, 从根本上减少误差的产生。

参考文献

- 1 邓文生, 马王俊美. Java3D 扩展鼠标交互功能的研究与实现. 计算机仿真, 2007, 24(3): 176 - 179.
- 2 Miranda FR, Santos CSD, Joao E Kogler Jr. Graphic simulation for Virtual Prototyping with Java3D. Computer Society, 2003. 83 - 90.
- 3 梅泽高, 俞涛, 王栋, 朱文华. 虚拟装配仿真系统相关技术的研究. 计算机仿真, 2007, 24(11): 231 - 234.
- 4 唐秀祯. 基于 Web 的虚拟装配系统的设计及实现 [硕士学位论文]. 成都: 西南交通大学, 2006.
- 5 The Java3D API Specification Version 1.3, sun Microsystems, 2002.