****

**图形学作业2:**

**层次模型和骨架子空间变换(SSD)**

学 号： 20171002157

班级序号： 111172

姓 名： 杨杰

指导教师： 罗忠文

**中国地质大学信息工程学院软件工程系**

**2019年 10月**

目录

[作业简介 2](#_Toc23442992)

[具体实现 2](#_Toc23442993)

[实现矩阵栈 2](#_Toc23442994)

[层次骨架：输入文件 2](#_Toc23442995)

[层次骨架：画杆状图形 2](#_Toc23442996)

[层次骨架：用户接口 4](#_Toc23442997)

[骨架子空间变换：文件输入：绑定姿态网格 4](#_Toc23442998)

[骨架子空间变换：网格渲染 5](#_Toc23442999)

[骨架子空间变换：文件输入：附加权重 5](#_Toc23443000)

# 作业简介

在本作业中, 需要构造层次角色模型并能与用户接口交互和控制。层次模型可以表示人形角色 (如普通人类, 机器人和外星人), 动物 (狗, 猫, 蜘蛛), 机械装置(手表，三轮车)等。你需要实现骨架子空间变换，在层次骨架上的简单蒙皮，并且当转动骨架的关节角时蒙皮会自然的变形。

# 具体实现

## 实现矩阵栈

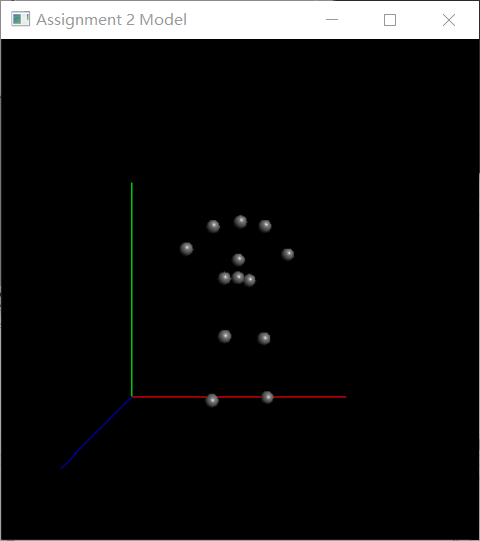
矩阵栈类有一个数据vector< Matrix4f > m\_matrices的矩阵栈；有五个函数：用单位矩阵初始化矩阵堆栈的构造函数，恢复为仅包含单位矩阵的栈的clear()函数，返回栈顶的top()函数，将矩阵放入栈的push()函数，从堆栈中删除顶部元素的pop()函数。根据提示对vector< Matrix4f > m\_matrices进行操作实现。

## 层次骨架：输入文件

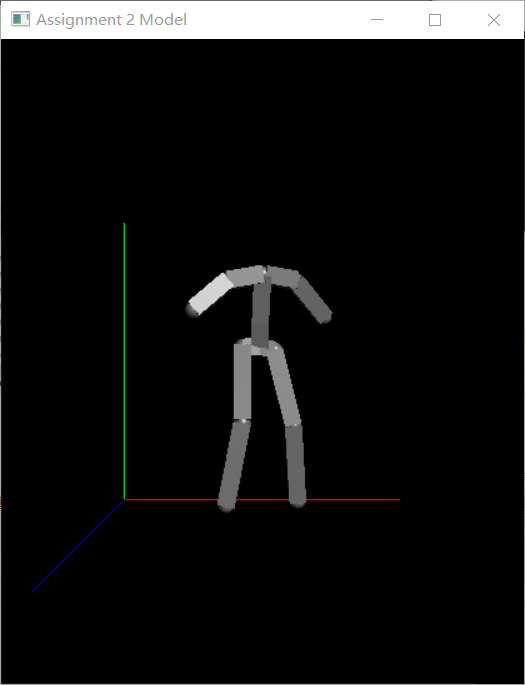
读取骨架文件(.skel)，解析骨架。每行包含4个用空格分隔开的字段，除第一行是角色在世界中全局位置外，其他行前三个字段是浮点数给出关节相对其父关节的平移，第四个字段是其父关节的标号。用指针Joint \*joint = new Joint装入关节，并对每个关节的children根据读取到的值进行赋值。

## 层次骨架：画杆状图形

创建一个递归函数，在drawJoints中调用，是从根出发遍历关节层次，使矩阵栈在每个关节处画一个球。后续遍历整个joint->children[i]将joint->transform装入MatrixStack中，绘图调前使用glLoadMatrix()，使用glutSolidSphere(0.025f, 12, 12)绘制图片。

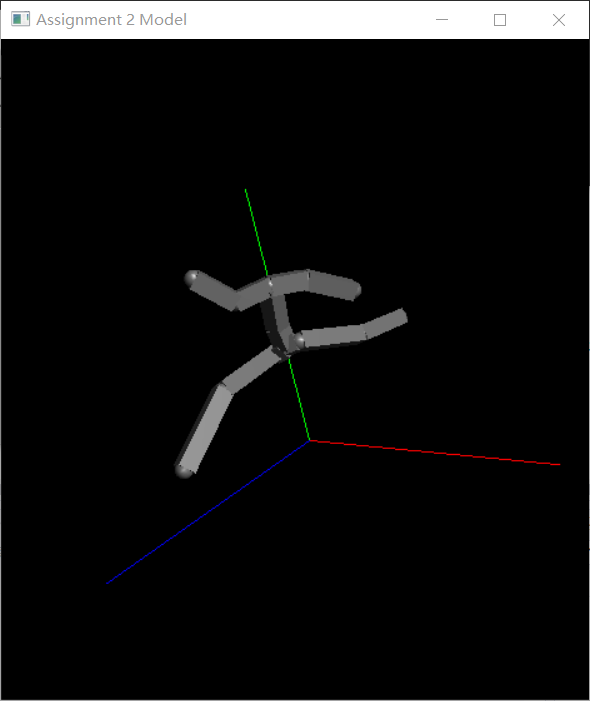


绘制骨骼，定义一个递归函数，该函数将遍历关节层次。除根节点外，在每个关节，画一个盒子，该盒子连接关节和其父关节。调用glutSolidCube( 1.0f )绘制边长1的正方体，沿z平移，范围从[−0.5, −0.5, 0]T到 [0.5, 0.5, 1]T。进行缩放变换，将范围从[−0.025, −0.025, 0]T 到[0.025, 0.025,L]T。使用parentOffset.normalized()旋转z轴，使与其父关节的方向对齐。



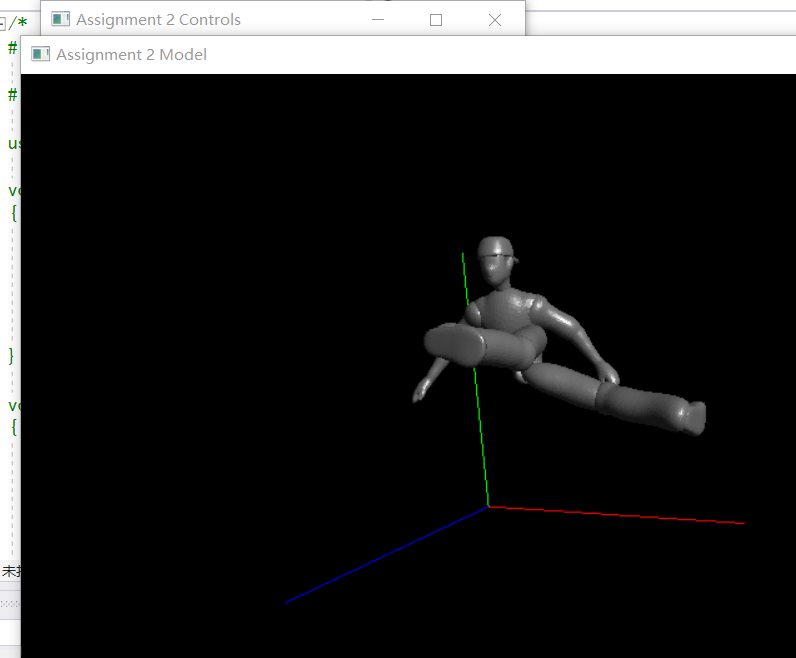
## 层次骨架：用户接口

当拖动关节旋转滑块时，调用setJointTransform, 传入需要更新的关节的序号和用户设置的欧拉角。根据传入的rX、rY、rZ使用rotateX(rX)更函数初始化Matrix4f，相乘后使用getSubmatrix3x3(0, 0)得到Matrix3f。最后使用m\_joints[jointIndex] -> transform. setSubmatrix3x3(0, 0, upperLeftR) 将当前变换旋转矩阵设置为整体旋转矩阵的旋转部分。



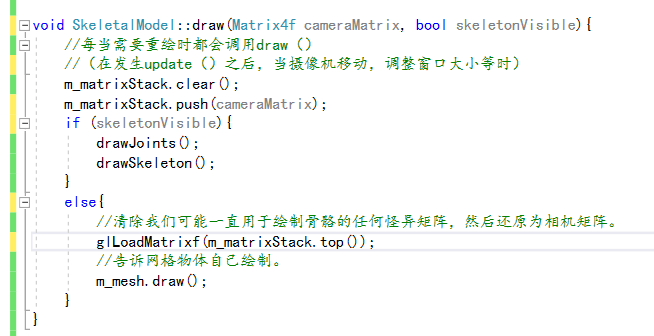
## 骨架子空间变换：文件输入：绑定姿态网格

类似作业0中的读取0bj文件，读取v开头的顶点装入vector< Vector3f > bindVertices，读取f开头的面装入vector< Tuple3u > faces。



## 骨架子空间变换：网格渲染

通过渲染网格来验证网格加载程序的正确性。调用Mesh::draw。一要用m\_mesh.currentVertices。



## 骨架子空间变换：文件输入：附加权重

完成Mesh::loadAttachments，.attach包含多行文本，每行是在网络中的一个结点。每一行包含所有关节，-1，字段用空格分隔。每个字段是一个浮点数，表示结点附着到第（i+1）个关节的强度。第0个关节(根关节)的权重，假定为零。填充网格的attachments字段。

