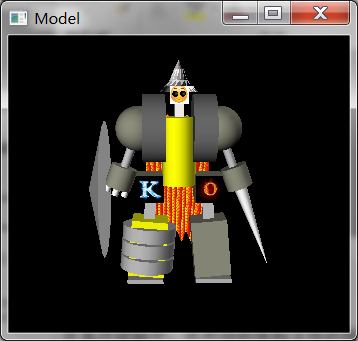
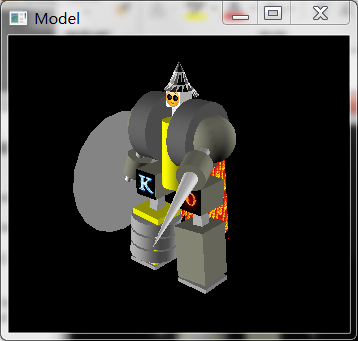
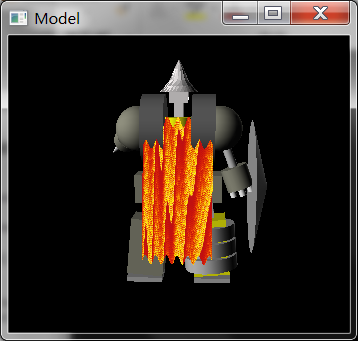
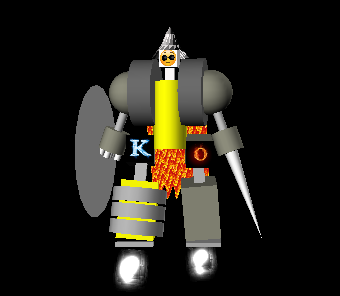
一.artifact (即你绘制模型的截图)：

模型截图如下：









二．画出你模型的Hierarchical结构图

机器人

左腿

右躯干

主躯干

披风

左躯干

脖子

右腿人

左腿上肢

右腿上肢

左肩

右肩

脑袋

左腿下肢

左臂上肢

右臂上肢

右腿下肢

头盔

左脚

左臂下肢

右臂下肢

右脚

左腿圆柱1

喷气装置

炮筒1

长剑

左腿圆柱2

炮筒2

左腿圆柱3

炮筒3

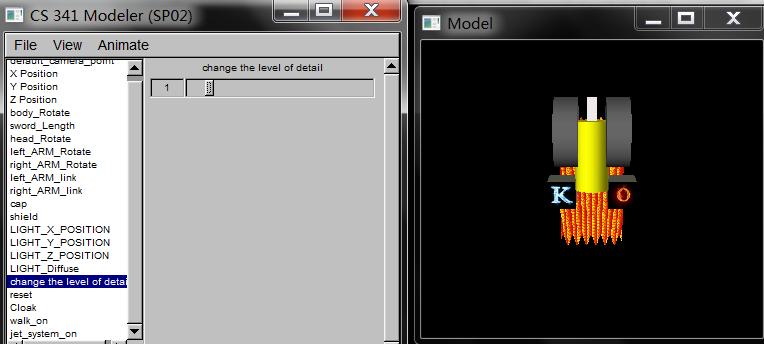
喷气装置

炮筒4

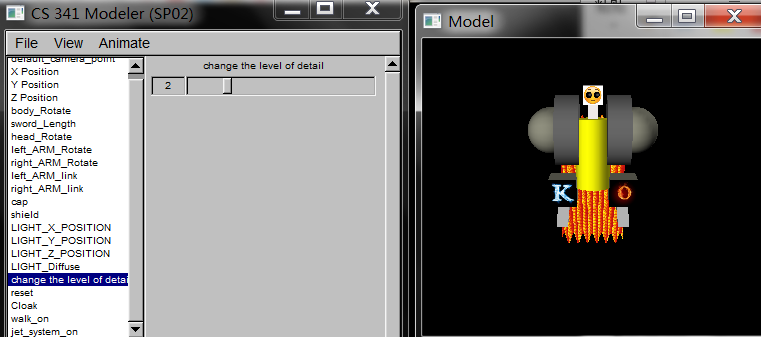
炮筒5

盾牌

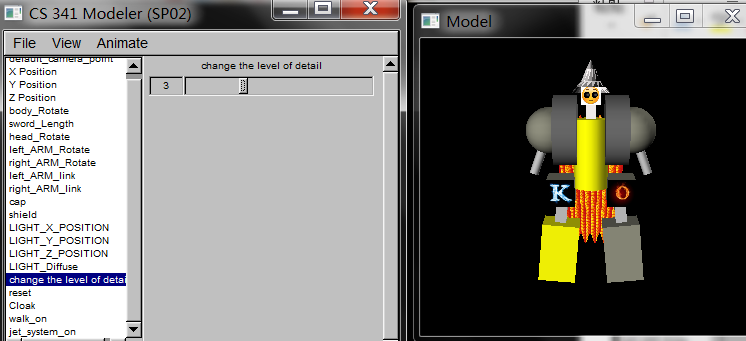
Level1：



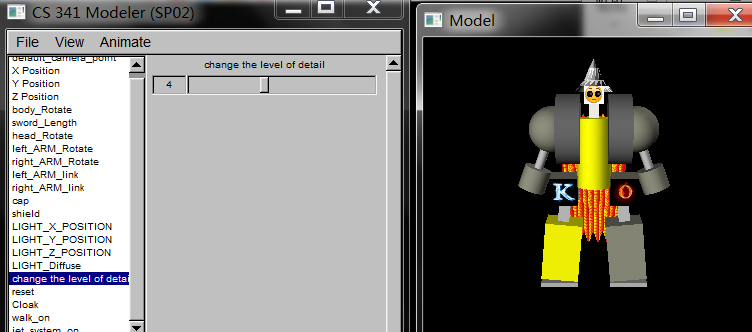
Level2：



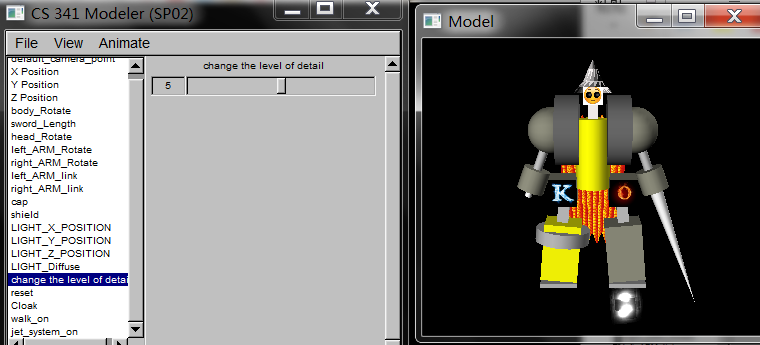
Level3：



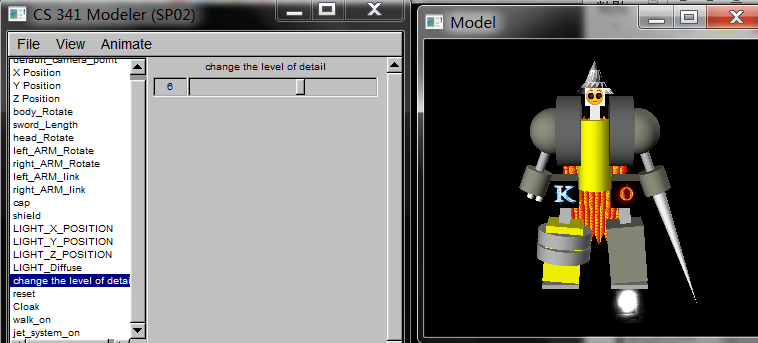
Level4：



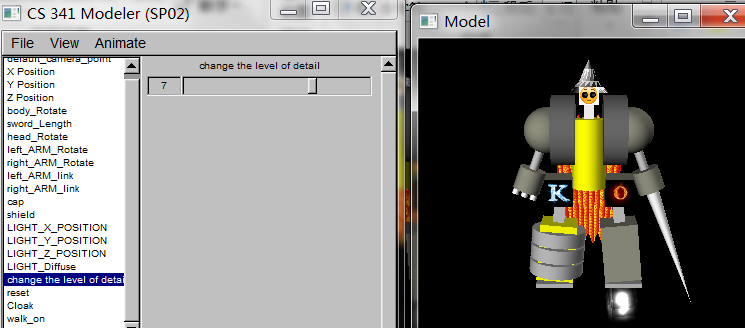
Level5:



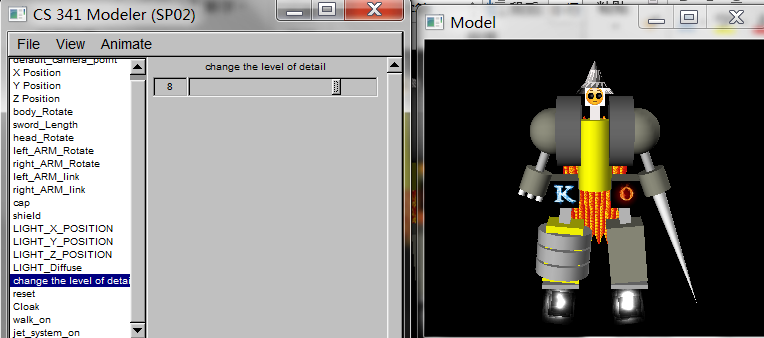
Level6:



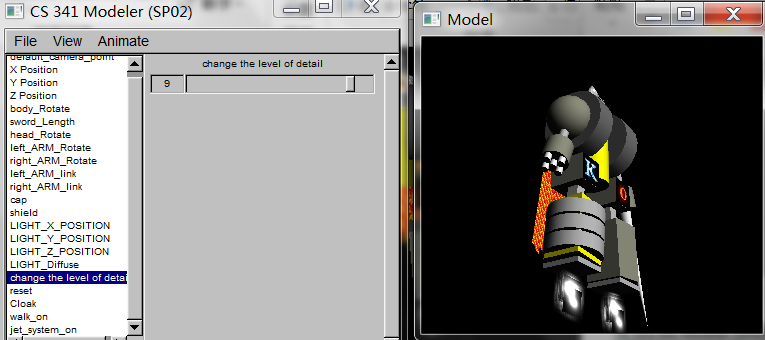
Level7:



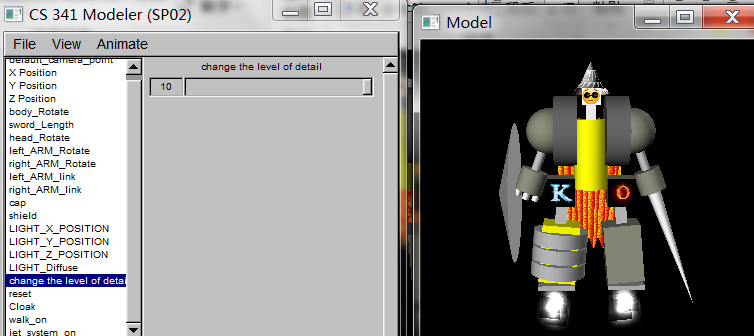
Level8:



Level9:



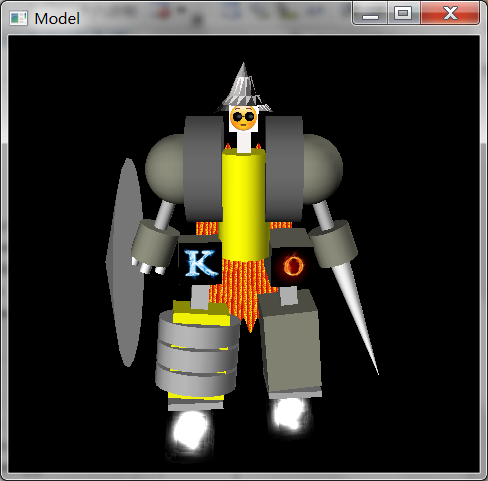
Level10:



三．说明你的代码完成了Project requirements里面的哪些功能

3.完成要求的至少10个基元，4个层次建模，使用glTranslate()，glRotate()，glScale()函数，使用glPushMatrix() and glPopMatrix().

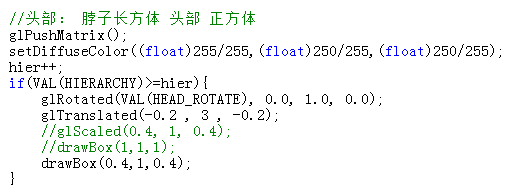
用drawBox()函数画立方体,drawCylinder()函数画广义的圆柱体，drawSphere()函数球体，用drawTriangle()函数画三角形，绘制过程需要用到glTranslate()平移坐标系，glRotate()旋转坐标系，glScale()放大缩小坐标系。一开始先画一个圆柱体作为主躯干，再在此基础上进行层次建模，分成5个方向进行，头部，左手，右手，左脚和右脚共5个方向。其中头部为脖子，头部，帽子三个层次，左手为左躯干，左肩，左臂，炮座，5个炮筒，防护盾10个层次，右手为右躯干，右肩（披风并列同一层次），右臂，基座，长剑5个层次，左脚为7个层次（加上隐藏的喷气装置有8个层次），右脚为4个层次（加上隐藏的喷气装置为5个层次）。以上为模型的构造思想。最终完整模型图如下所示：

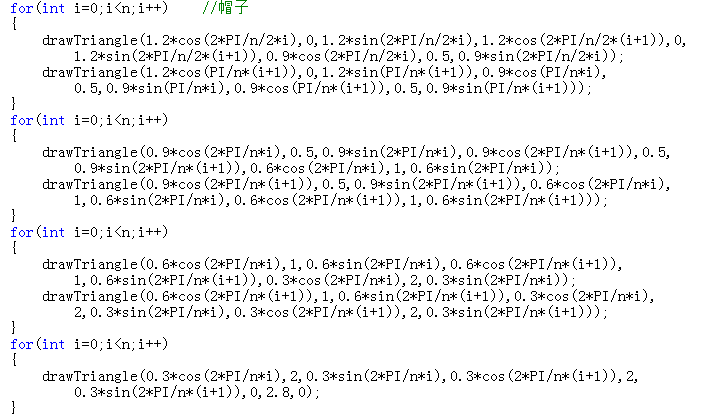


4.各部分详细过程：

头部方向为：

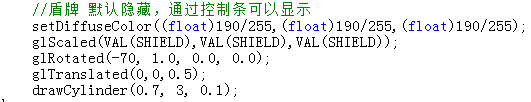
在圆柱的主躯干上，先画立方体的脖子，再画正方体的脑袋，最后用drawTriangle()画三角形由底向上分四层拼出来一个复杂图形作为帽子。主要代码如下：



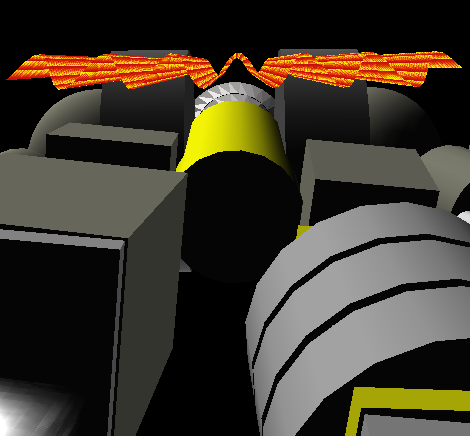


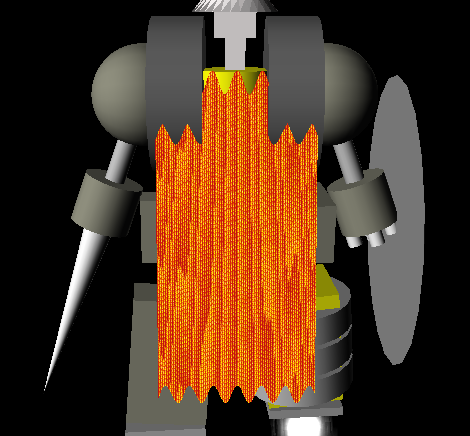
左手方向为：

先画圆柱左躯干，再画球体左肩，再画左臂圆柱，再画炮座圆柱，再画5个炮筒圆柱，最后画类似圆锥的圆柱体盾牌。其中盾牌代码如下：

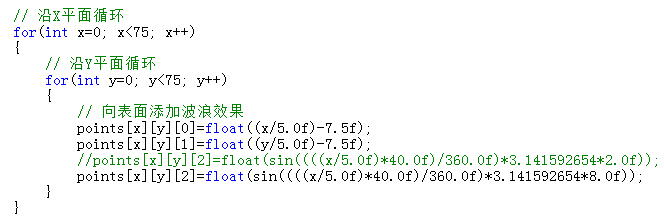


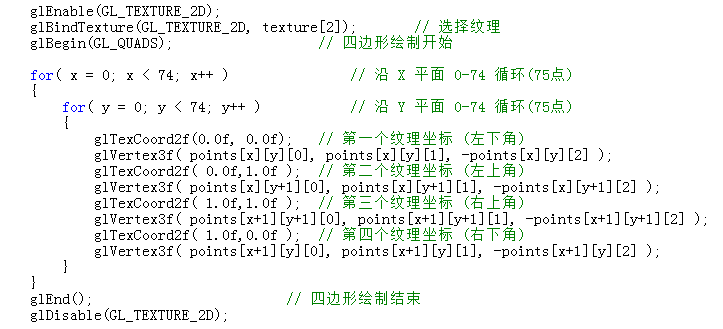
左手部分还有一个是披风。它是一个波浪形的图形，由75\*75个小四边形拼接而来，通过每次坐标的轻微变换得到波浪形的披风。图示如下红色部分：





主要代码如下:





右手方向：

基本雷同左手，只是炮部分变成了长剑。

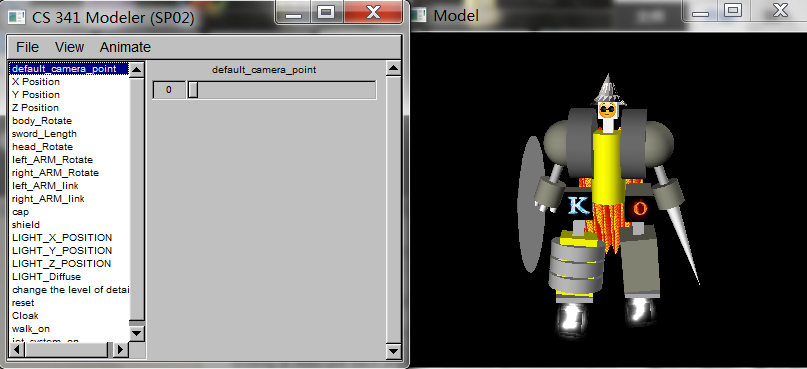
左脚方向：

先画一个立方体的轴点，再画左腿，再画一个立方体，接着脚底画一块立方板，接着在腿上画三个圆柱，最后画喷气装置。

右脚方向：基本雷同左脚，只是少了三个圆柱。

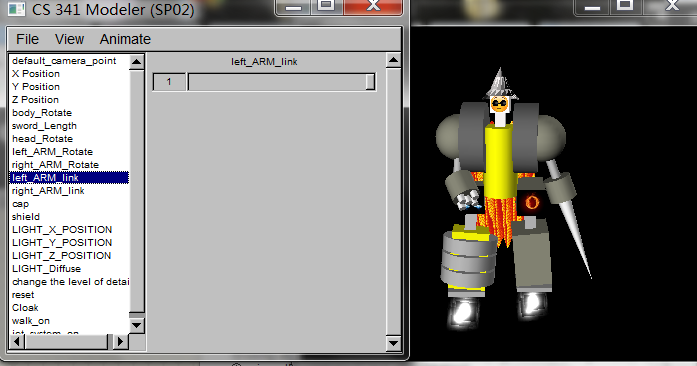
5.加控制条

本部分加了转换相机位置控制条default\_camera\_point，X,Y,Z控制条XYZ Position，上半身转动控制条body\_Rotate，右手长剑长度控制条，左右手转动控制条，灯光控制条等等。



6．完成一个控制条可以控制一个以上的关节或组件

控制条left\_ARM\_link,可以控制左肩转动，同时控制炮座转动，图示如下：

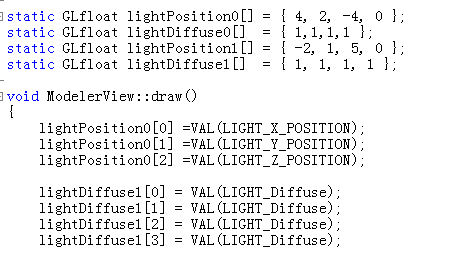


控制条right\_ARM\_link,可以控制右肩转动，同时控制右臂转动，图示如下：



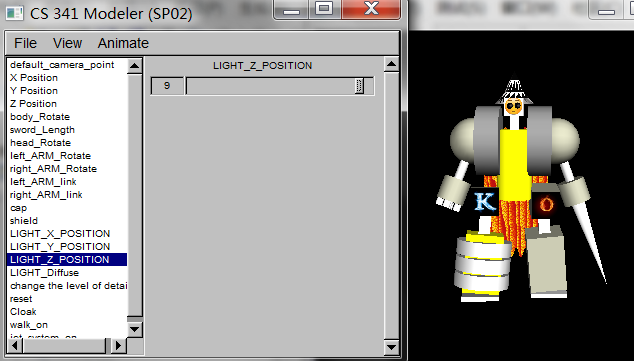
7.完成改变光源位置和光强控制的whistle

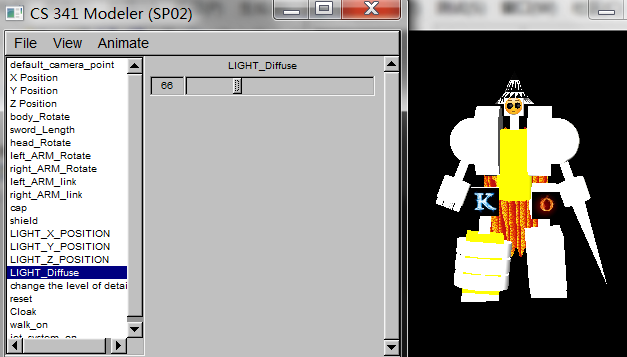
修改在modelerview.cpp进行，主要代码如下，其中lightPosition0[]存储光源位置，lightDiffuse1[]存储光强参数：



前后对比图：



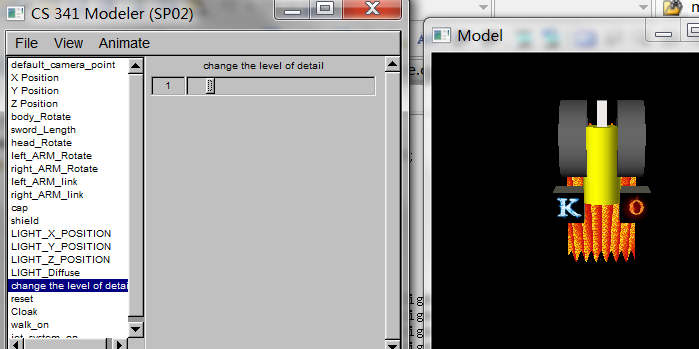


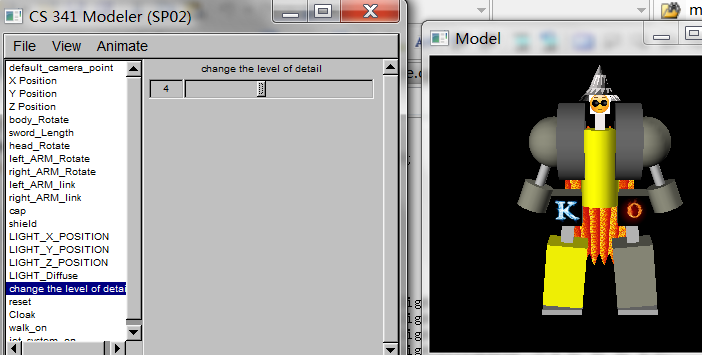


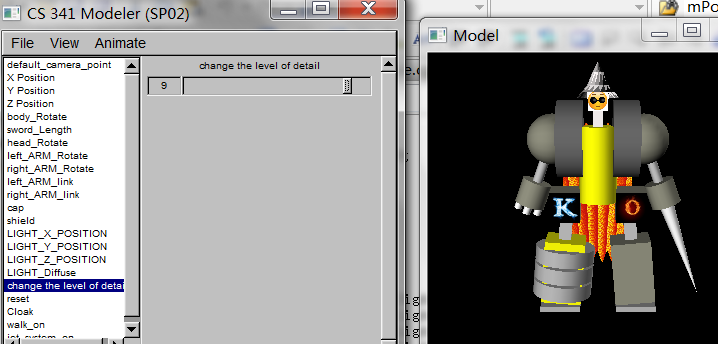
8.完成用控制条显示建模层次的whistle。

增加控制变量HIERARCHY，在sample.cpp中增加全局变量hier=0,修改draw 函数，一个glPushMatrix()和glPopMatrix()之间每一次有绘图函数画图进行hier加一，最后在glPopMatrix()前hier减去增加的层次数，两者之间都写在if(VAL(HIERARCHY)>=hier){}里面,对全文档进行修改后，即可对模型进行层次展示。

效果如下：



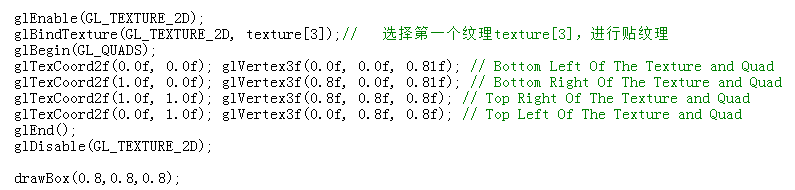




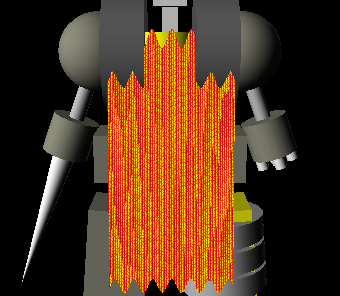
9.完成贴图bell

在左右脚贴了字母KO，头部贴脸图，披风贴75\*75的红黄色图

主要代码如下：

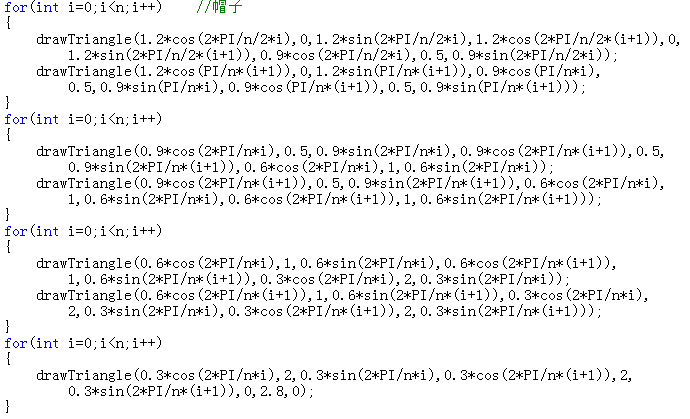


效果如下：

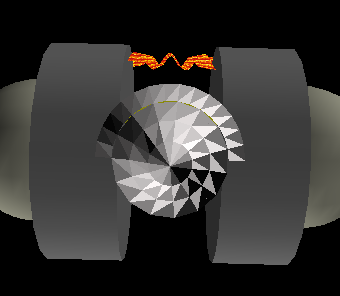


10.完成用三角形画复杂图形bell

主要代码如下：



效果如下：

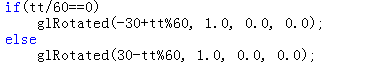


11.完成动画bell，需配合walk\_on控制条和打开Animate enable使用

增设全局变量tt，每次调用draw函数时tt增加相应的值，然后把TT放到脚步和上身的参数中，,这种情况下会有人物在场景中上身左右摆动30度，下身前后走动30度的效果。

主要代码如下：





效果图：

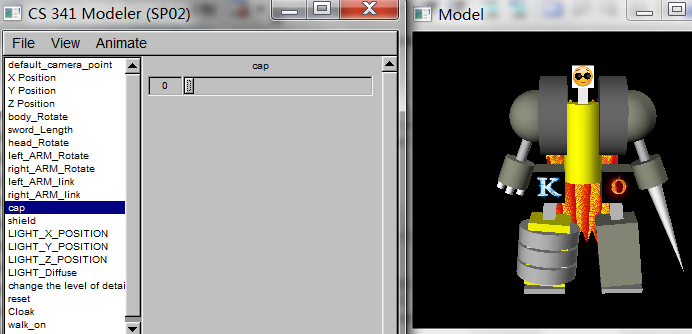


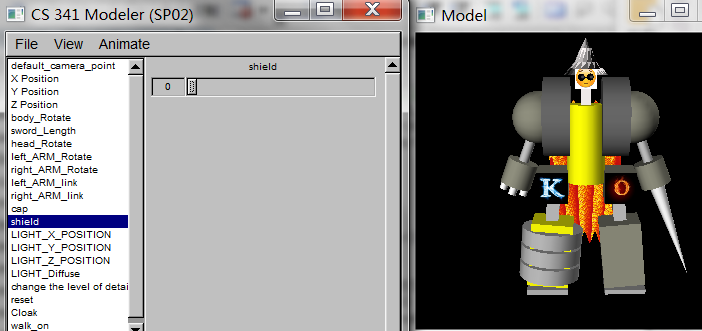


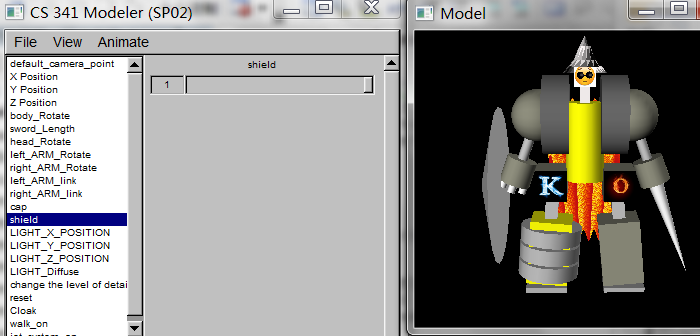
12.完成创建不同外观的bell

在角色中添加了盾牌和帽子。通过控制条，把它打开/关闭。

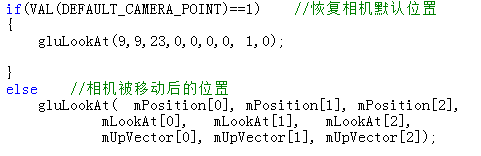
效果如下：



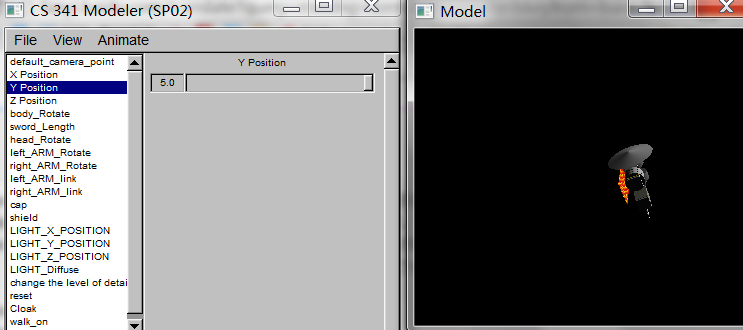


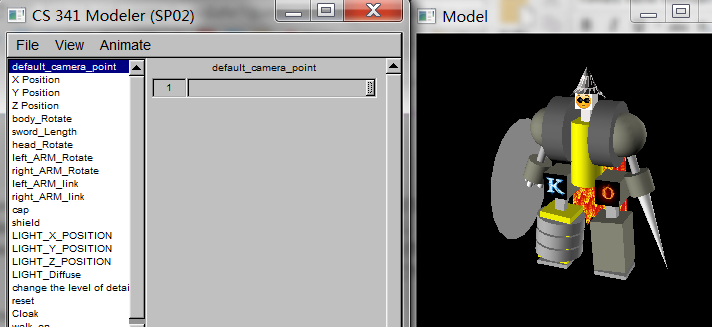


13.完成添加恢复相机默认位置重置控制条bell，使得模型在窗口是完全可见的，无论姿态和位置。本过程修改camera.cpp，修改最后一个applyViewingTransform()函数里面的glulookat函数即可，主要代码如下：



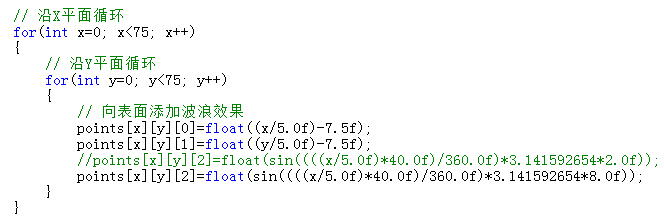
效果如下：

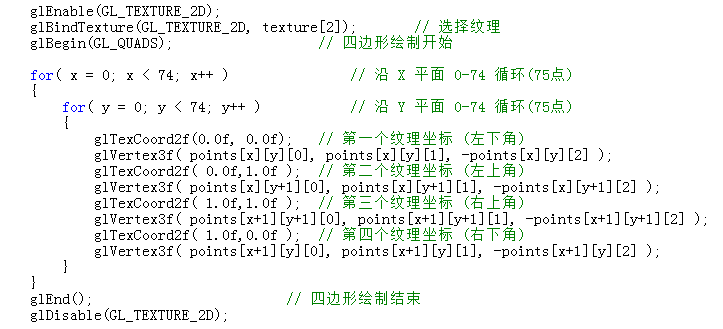




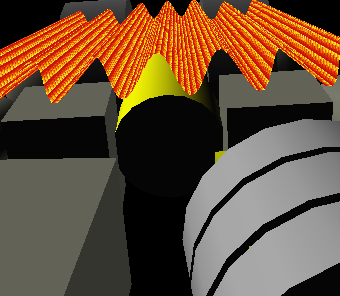
14．完成自己定义一个原始绘图函数的bell

它是一个波浪形的图形，由75\*75个小四边形拼接而来，通过每次坐标的轻微变换得到波浪形，主要代码如下：



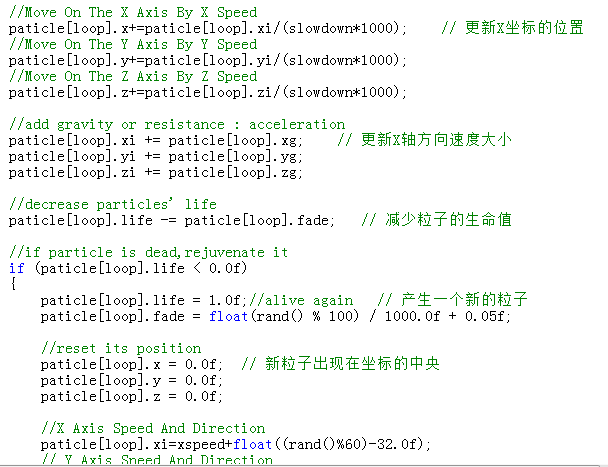


效果图：



15.增加炫酷的粒子系统—喷气装置，需配合jet\_system\_on控制条和打开Animate enable使用.

定义1000粒子，每个粒子都有生命周期，以不同速度往负Y轴移动，消亡后回到产生的坐标继续重复，粒子都贴图，使喷射效果更明显。主要代码如下;



效果如下：

