

计算机图形学 作业 1 作业报告

一、 要求

1. 调用 OpenGL 库函数绘制一个球面；
2. 鼠标或音频控制小球弹跳运动，要求物理仿真；
3. 写一个 Loader 读入三维模型 (.max、.3ds 或.obj 格式) 并绘制，三维模型可替换球面做弹跳运动。

二、 实现

1. 绘制球面

根据题目的要求，查阅了 OpenGL 与 GLUT 相关的文档，最终使用了 *glutSolidSphere* 函数进行球面的绘制。此时并未进行材质与光照相关代码的编写，因此得到的结果仅仅是一个看似平面的圆形。

2. 控制小球弹跳运动

我采用的方法是使用鼠标控制小球弹跳运动，之后为了试验的便利性我另外绑定了空格键 (space) 触发小球的弹跳运动。

执行程序后，小球呈现静止状态 (状态 0，在程序中对应 $status = 0$) 出现在屏幕中央偏下的位置，如图 2-1 所示，点击屏幕任意位置后小球会得到一个向上的初速度 ($v_0 = 10$)，即向上运动 (状态 1，在程序中对应 $status = 1$)，

对应公式为
$$\begin{cases} v = v_0 - gt (g = 9.8) \\ height = v_0 t - \frac{1}{2}gt^2 \end{cases}$$
 (t 为当前帧与初始帧的时间差)，运动过程中不考虑空气阻力，当小球到达顶点后记录最大高度 (max_height)，随后落下 (状态 2，在程序中对应 $status = 2$)，对应公式为

$$\begin{cases} v = gt \\ height = max_height - \frac{1}{2}gt^2 \end{cases}$$
 再次撞击底面时，假设这个碰撞过程会使小球损失 20% 的速度，随后再次跳起，直至小球的跳起高度小于一定值时

($max_height \leq 0.001$ ，近似为静止状态)，小球会被重置为静止状态 (状态 0)，弹跳运动完成。

如果在小球运动过程中 (状态 1 与状态 2) 点击鼠标，小球的速度将被重置为 10，方向向上。我使用了全局变量 *pos* 来记录点击时小球所在高度，当事件被触发时，*pos* 会记录该数据，因此上升时有关高度的公式应变为 $height = v_0 t - \frac{1}{2}gt^2 + pos$ 。

基于上述公式，我认为小球的运动基本达成了物理仿真的要求。



图 2 - 1

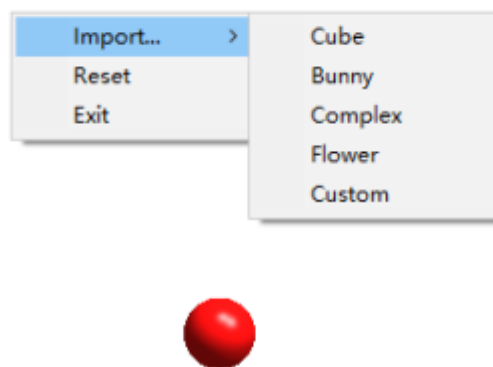


图 2 - 2

图 2-2 显示的是基于 GLUT 的 *glutCreateMenu* 函数制作的菜单

“Import”项的作用为利用 loader 进行.obj 文件的导入，在下文会有所提及；

“Reset”项的作用为重制小球为静止状态，在使用 loader 之后也可以使用该按钮将图形重置为小球；“Exit”项的作用为退出，esc 键也可达到相同效果。

3. 读入三维模型 (.obj 文件)

我认为 loader 的编写是这次作业中最为复杂的部分，其中要考虑到的事情较为复杂，我单独使用了一个类来编写 loader。

对于.obj 文件，我仅处理了“#”（comment，即注释）、“v”（geometric vertex，即几何体顶点）、“vn”（vertex normal，即顶点法向量）、“f”（face，即面）。

在读取文件的每一行之后，通过首位字符判断分类，如果首位字符为 v 再通过第二位字符判断为顶点或是法向量。在处理“f”的时候我分成了两种情况，其一为 Vertex Indices，其二为 Vertex Normal Indices Without Texture Coordinate Indices。在第二种情况下，三个参数使用“/”分开来，由于 C++ 标准库没有内置 *split* 函数，我自行编写了一个，第一项为顶点索引，第二项为空，第三项为法向量索引。

起初我作为标准的.obj 文件过于简单，一度以为 f 的参数竟有三项，实际上是不定长的，因此我一开始编写的函数仅能应付三项的情况，后来我对处理的函数进行改写，将多边形分割成多个三角形进行绘制，此时就能应付多项参数的情况。

图 3-1 为在 macOS 下预览.obj 文件的结果，图 3-2 为使用 loader 进行读取后的结果，可以看出 loader 的结果还是令人满意的。且使用 loader 读取之后运

动状态并不会受到影响，如果在运动过程中进行读取，读取进来的物体会继续保持上一个物体的运动状态。



图 3 - 1



图 3 - 2

三、 总结

我认为本次作业还是十分具有挑战性的，尤其是在 loader 以及小球弹跳运动过程的处理上，虽然我基本按照要求完成了本次作业，但是 loader 我自己处理得还是相对较为简单的，仅仅考虑了一些必要的部分。根据我所查阅的资料，OpenGL Mathematics（即 GLM）库有处理读取.obj 文件的函数，其中要考虑的参数十分复杂，我只处理了对于形成图形而言较为重要的几个，对于模型中没有定点法向量的文件的光照效果的处理也有待提高。