

计算机图形学作业 1 报告

一、 题目描述

作业 1: 基于 OpenGL 编程实现

调用 OpenGL 库函数绘制一个球面 (10 分)

鼠标或音频控制小球弹跳运动, 要求物理仿真 (50 分)

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 (30 分)

作业报告 (10 分)

*加分项:

(1) 球面可在弹跳过程或受力过程中塑形变形

(2) 用音频控制代替鼠标控制

(3) 其他任何你们的创新设计

二、 基本设定

小球或模型在 900*400 的窗口中弹跳。弹跳范围限定在 xOz 平面, 并始终受到“重力”的吸引。每次接触地面 (x 轴) 会反弹, 同时各方向的速度缩减为 90% (模拟弹跳过程的能量损失)。(物理仿真)

物体弹跳过程中会沿着速度方向拉伸变形, 在靠近地面时会挤压缩小。
(塑形变形)

外界声音达到一定大小时物体加速, 鼠标在窗口内拖动时物体减速。(鼠标和音频控制)

三、 实现简述

1、main.cpp: 包含主函数、物体运动的描述和图形绘制。

其中的 Model 类保存当前时刻物体的三个坐标分量、速度分量、加速度分量。通过 glutTimerFunc 函数定时调用 move 方法, 根据时间间隔, 计算下一时刻的坐标和速度, 如此循环。

2、objLoader.cpp(h): 从文件读入 obj 模型

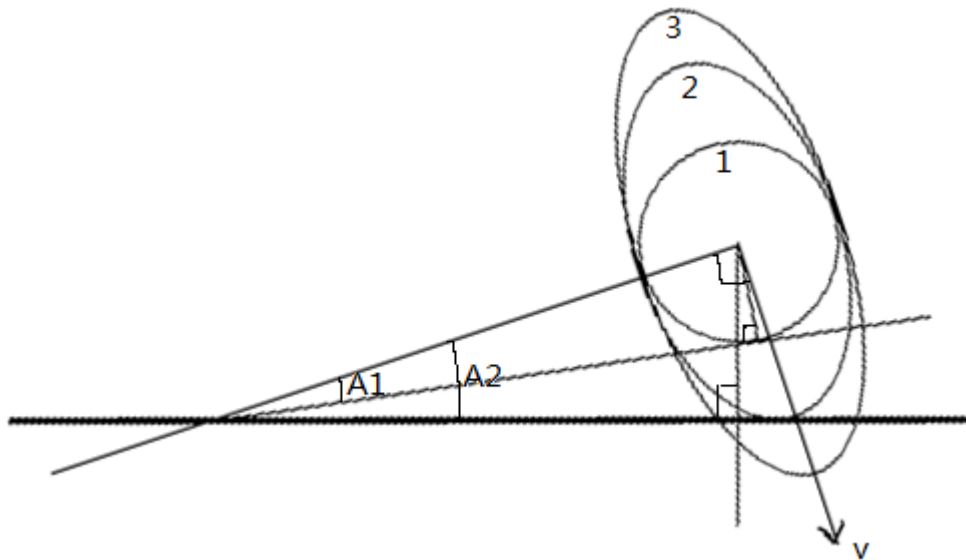
程序默认读取当前目录下的 model.obj 文件并载入。如果载入成功, 则绘制该模型; 否则调用 glutSolidSphere 库函数绘制一个球面。

3、waveIn.cpp(.h): 处理音频输入

程序定时采集声音数据并计算音量，根据音量改变小球的状态

四、 创新设计

- 1、 物体的运动：为了更好的物理仿真，采用实时计算的方式确定物体的位置而不是利用公式计算每个时刻的坐标。由于运动形式为抛物线，在一个时间间隔内，用中点的速度作为平均速度计算位移可以完全消除时间分割上的误差。
- 2、 物体的形变：物体运动时沿运动方向拉伸，拉伸比例与速度成正比（但不会超过上限）。需要注意的是物体靠近边界时的特殊处理。



如上图，圆 1 是运动的物体，速度方向如 v 表示；如果按照速度的大小计算，其拉伸程度应达到椭圆 3 的大小，但受地面的限制，拉伸程度事实上只能达到椭圆 2 的大小。main.cpp 中的 maxScaleRatio 函数用来计算此种情况下从圆 1 拉伸到椭圆 2 的比例（在途中，该值为 $\tan A2 / \tan A1$ ）。如果圆 1 距离地面小于半径，则不再沿运动方向压缩，而是改为沿垂直方向压缩。

经上述处理后，物体弹跳过程中的形变更加真实。

（程序中为了简化，只处理了下边界的情况，对于左、右、上三个边界未作处理）

五、 其他说明

- 1、 程序中关于物体的运动、形变等核心代码完全原创，其他部分（opengl 作图、obj 模型读入、音频采集等）参考了网上的实现，在此向各位原作者表示感谢。
- 2、 程序省略了很多繁琐的细节处理，如窗口大小变换的处理（直接在代码中固定数据）、光照处理（使用了 `glLightfv` 简单处理）、视角变换（通过 `gluLookAt` 处理，数据直接在代码中固定，而不是根据用户鼠标的移动等来决定）、obj 文件读入（直接在指定目录寻找，不允许用户指定）、物体颜色（直接使用灰色）等。
- 3、 当声音控制使小球加速时会在控制台打印“volume %d”（可能需要较大的声音才能触发），当鼠标在界面中拖动使小球减速时会在控制台打印“MouseMotion”
- 4、 需要 `glut.h`、`glut.lib`、`glut32.lib`、`glut.dll`、`glut32.dll` 以支持编译和运行，编译环境为 visual studio 2015。直接打开 `hw1.vcxproj`
- 5、 如果在 visual studio 中通过 F5 或 Ctrl+F5 执行，则默认模型文件 `model.obj` 在 `hw1` 文件夹下（与代码文件在一起）；如果单独执行生成的 `exe` 文件，则默认模型文件应与 `exe` 文件处于同一目录。（也可以直接运行演示/模型/`hw1.exe` 和演示/小球/`hw1.exe` 查看效果）