
浙江大学

计算机图形学作业报告

作业名称:	课程大作业 AE-86
组 长:	李鸿屹
组 员:	贝晨杰、陈越尧、徐俊豪、Alon Kagan
电子邮箱:	858519155@qq.com
联系电话:	17326083793
指导老师:	张宏鑫

20019 年 1 月 17 日

计算机图形学期末大程（AE-86）

一、 作业已实现的功能简述及运行简要说明

作业已实现的功能：我们的课程大作业做的是一款游戏，名字为 AE-86，取之于电影《头文字 D》。这是一款赛车游戏，构建一个完整的 CG 世界。不仅实现了项目的基本要求：建模、存储、编辑、变换、光照、漫游、记录，同时做了项目的高级要求：NURBS 曲面建模，漫游时碰撞检测，光照明模型细化，复杂材质效果，以及具有可玩性的三维游戏。

运行简要说明：“wsad”分别控制着汽车的前移后移左转右转；“z”加速前进；“v”控制着视角，共有“远”“较远”“近”“正常”四种视角；“c”是屏幕截取功能，截屏保留在工程文件夹下的 record.bmp 文件中。鼠标的滚动键可以小幅度调整视角的远近。

由于在实际工程中存在修正、完善和覆盖的过程，因此我们的报告按照每个人的工作实现情况来对项目中的难点进行阐述。

二、 作业的开发与运行环境

Windows10 下在 Visual Studio 2019 中利用 OpenGL 开发

三、 程序具体实现要点与组员分工

李鸿屹 3170106234

完成模块：场景建模、纹理贴图、碰撞检测改进、截图保存改进

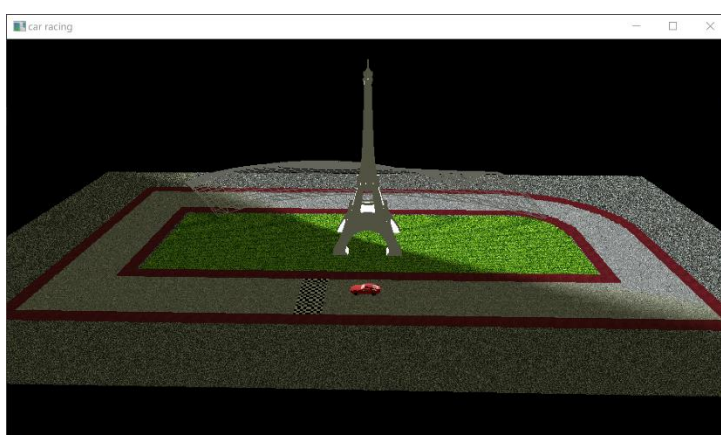
涉及到的技术点：

1. 具有基本体素建模与 NURBS 曲面建模

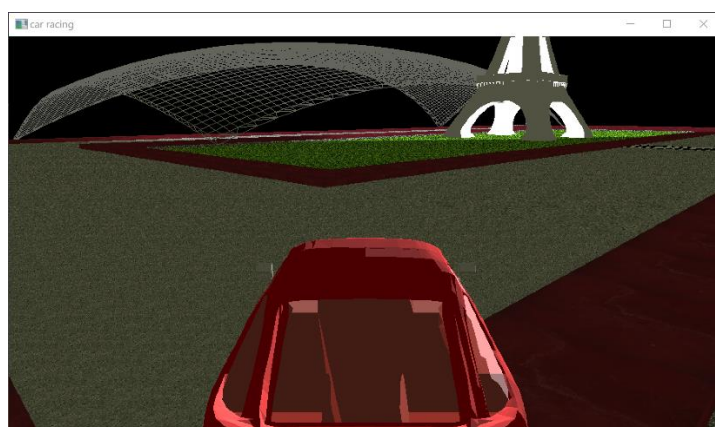
建模在本项目中主要体现在场景建模上，最基本的要素为跑道边缘。为了达到约束赛车行驶在跑道槽内的视觉效果，这里的路基部分采用多个长方体堆叠的方式进行表达，实际上与实际生活中的路基搭建是相一致的。



此后就采用延伸堆叠的方式绕着整个跑道完成路基的搭建，整个跑道由三个直角弯以及一个圆弧弯道构成，起点处用黑白格表示开始。整个场景的大小与赛车的 OBJ 模型大小相适配，包括三个组分：中央草地、赛道以及赛道外的沙石区域。



除了基础场景外，在场景中还设置了两个装饰性建筑：利用 NURBS 曲面建模完成的弧形隧道，以及中心的艾菲尔铁塔，如下图所示。

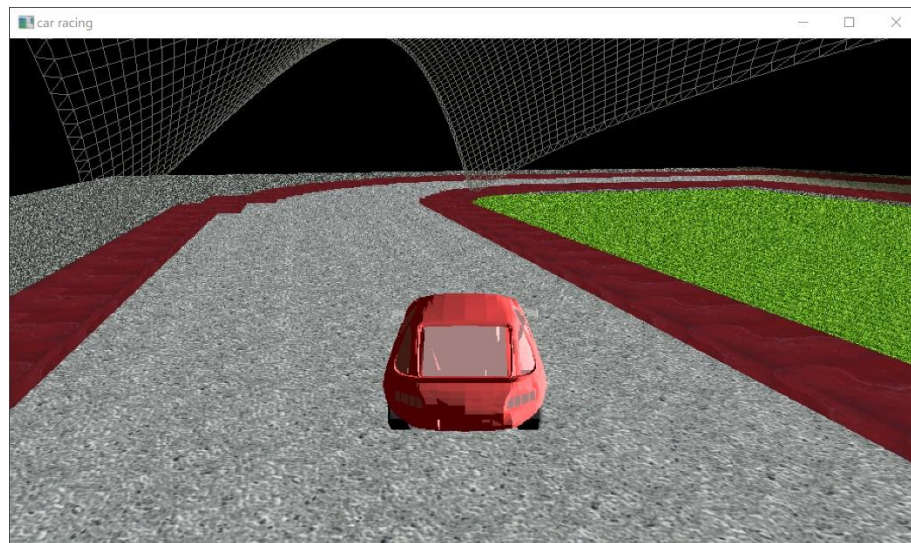


以上的弧面隧道采用 OpenGL 内置的 NURBS 曲面绘制函数 `gluNurbsSurface` 完成，希图实现的就是能够在直道上展现出隧道的效果。场景中心的艾菲尔铁塔模型是直接利用现成的 Obj 模型进行导入完成的，因此不再单独介绍

但由于 NURBS 在我们测试的过程中发现因为绘制所占用的资源较多，在部分组员的电脑上运行情况不够稳定，帧率出现明显的下降，为保证该游戏的流畅运行，所以仅在答辩展示，以及该报告中展示了这部分的内容。

2. 编辑：具有基本材质、纹理的显示和编辑能力；

整个场景的着色部分全部采用纹理贴图进行。包括：草地的部分采用了草地的纹理，路面部分采用沥青作为纹理，路基部分为经过着色加工的大理石纹理，曲面隧道则采用金属材质纹理，而场外用粗沙砾作为纹理进行贴图。

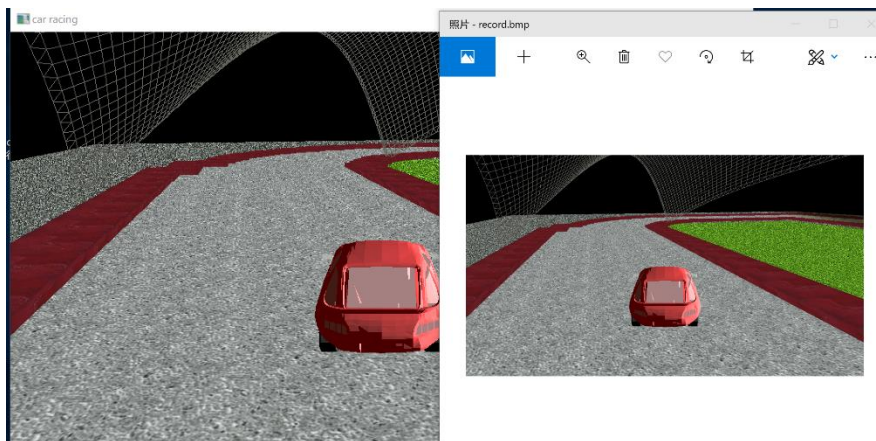


贴图过程的实现通过 `load_texture` 函数将纹理加载到缓冲区中，每次调用时，通过激活 2D 纹理将纹理绑定到对应的区域上。为了方便起见，本项目中的纹理除了路基外均以 10×10 的小方格作为基本单位进行贴图，除了路基刻意追求的分块效果外，草地、跑道、场外部分，都尽力避免了明显的拼接缝的出现。

加载的纹理均通过转换 jpg 图像到 bmp，再裁剪成 2 的幂次边长的正方形完成。再路基的部分除了原先的纹理之外，还单独再其上进行了第二次着色，从而显示出红色的大理石路基效果。

3. 记录：提供屏幕截取/保存功能。

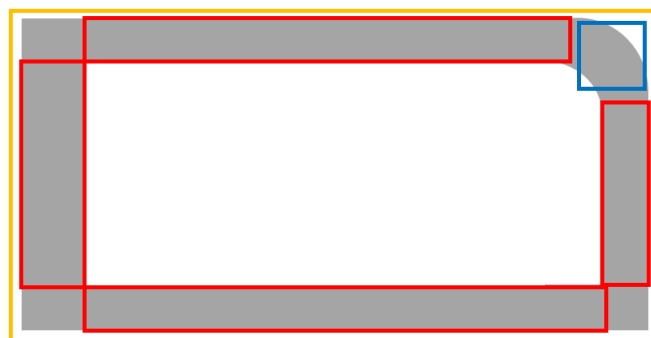
屏幕截取的基本思路可以类比于 Lab6 的图像处理实验，即直接将片元着色器的扫描结果读入文件，总而实现截屏的效果。下图展示的即为运行过程中的实时截屏效果，截屏保留在工程文件夹下的 `record.bmp` 文件中。



具体的实现上，首先先读取用于贴图的成型纹理的数据，复制文件头作为新 bmp 文件的文件头，然后采用 `glReadPixels` 对整个窗口的绘制结果进行扫描，将每一个像素点扫描得到的颜色信息直接写进新的文件即可完成截屏的工作在本实验中，截屏直接通过 C 键进行触发。这一块改进主要将代码的实现过程进行了简化，减少了对外接库的依赖。

4. 漫游时可实时碰撞检测

在此前对于整个场景的碰撞检测的实现基础上，答辩后我们对于碰撞检测部分进行了细化。由于后期加入了计时机制，为了保证游戏的公平性，我们将车体的运动限制在了整个跑道内，实现的具体思路如下图所示。



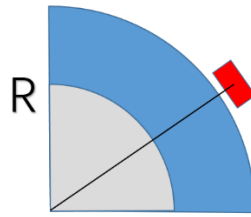
依据跑道的结构特点（灰色部分）将碰撞检测分成三个层次实现。

第一个层次为整个跑道的包围盒收束，即 1 上图的黄色部分，通过控制赛车的位置不能超出黄色矩形边界达到控制赛车不超出外边界的目的。

第二层次为上图红色部分，在跑到中存在四个全直线赛段，在上一层次中已经保证了赛车不超出外边界在这里，如果赛车位于上述四个红色区域内，就用类似的方法控制赛车的移动不能超出跑道的内边界。对于三个直角弯道，四个区域的收束恰好也完成了对直角弯出的收束，因此，下面只差对弧形弯道的碰撞检测处理。

在实现算法上以上两个模块的实现方式是类似的，如果赛车的位置处于相应的分段，且横坐标或者纵坐标超出了允许的界限，就将对应坐标重新赋值为相应的临界值，就可以完成对于水平/竖直便捷的碰撞收束。

第三层次就是蓝色标出的圆弧形区域。圆弧区域的碰撞检测算法如下图所示。



对于弧形区域,所谓的赛车在边界内指的是赛车到圆弧圆心的距离在内外边界的半径的闭区间内,由于内外边界的实现思路相近,这里就以外边界为例说明。设相对于圆心,小车的坐标为 (C_x, C_y) , 如果该点到圆心的距离大于外边界半径 R , 则更新 C_x 和 C_y 的值。

$$\begin{cases} C_x' = C_x \frac{R}{\sqrt{C_x^2 + C_y^2}} \\ C_y' = C_y \frac{R}{\sqrt{C_x^2 + C_y^2}} \end{cases}$$

以上,就完成了对于赛车在场地内部的碰撞检测的改进。

贝晨杰 3170100677

完成模块: 赛车移动的键盘响应、基本光照明模型、加入计时器

涉及的技术要点:

1.变换: 赛车运行键盘响应

本项目中通过键盘控制赛车的旋转、平移,具体实现是借助 x,y,z 坐标与车身角度 $frotate$ 来确定场景中的赛车,每次“wsad”或“z”触发键盘响应,即对这四个量做相应的改变。



2.光照: 建立了基础光照明模型,实现了基本光源编辑

加入了环境光、辐射光等光照,调整了光源、光强等参数。

3. 构建了基于此引擎的完整三维游戏,具有可玩性: 添加了游戏里的计时器。



当程序开始运行时，计时器变启动计时，直到程序退出。程序具体代码是在 `gettime()` 函数中实现的，其计时是按游戏中的时间计算，并非现实秒数。

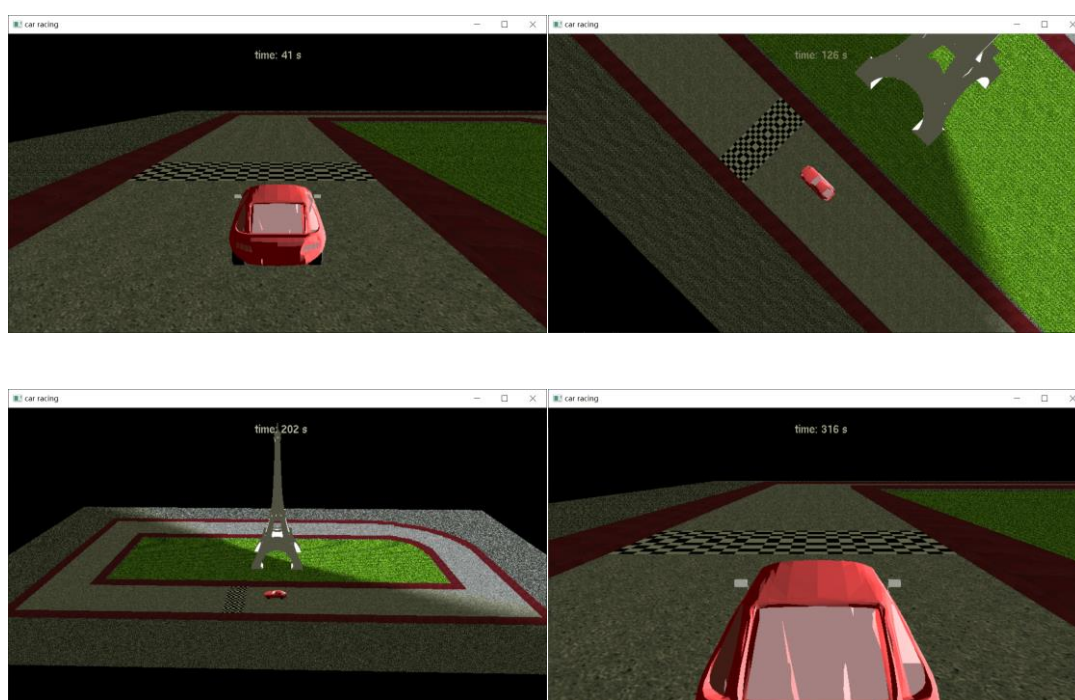
陈越尧 3160101919

完成模块：多视角及切换、镜头拉伸、汽车运行控制优化、场景布局优化

涉及到的技术点：

1. 漫游：实现 4 个视角及切换

根据虚拟相机、小车和场景的相对位置关系的计算，得出四个视角的位置参数；然后通过视角函数的设置实现多视角，通过对键盘事件的响应实现视角的切换。



2. 漫游：实现镜头拉伸

通过对鼠标滚轮事件的响应实现对镜头的拉伸。

3. 漫游时可实时碰撞检测：实现矩形区域的碰撞检测

根据小车和矩形场景的相对位置关系实现碰撞检测。

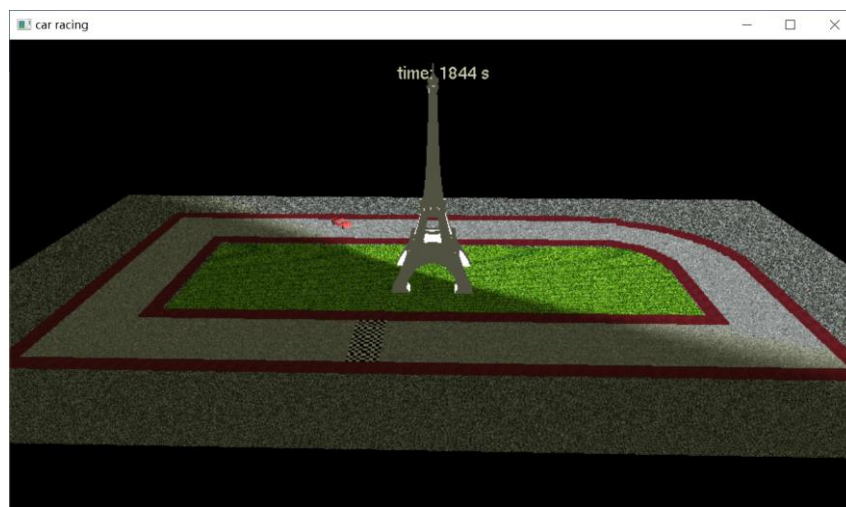
4. 变换：汽车运行控制优化

通过状态机编程，实现小车在转向的同时前进。即将每一次键盘事件记录下来但不立即处理，在绘制更新的空档期同意完成键盘回调事件的刷新。

5. 建模：场景优化布局

初期的场景建模将整个直道的宽度均设置为 500，但在实际运行时发现这一大小的视觉效果不佳，因此将长直道的宽度从 500 调节成 600，从而实现了更好的游戏体验。

6. 光照明模型细化：复杂光照明模型



徐俊豪 3170100183

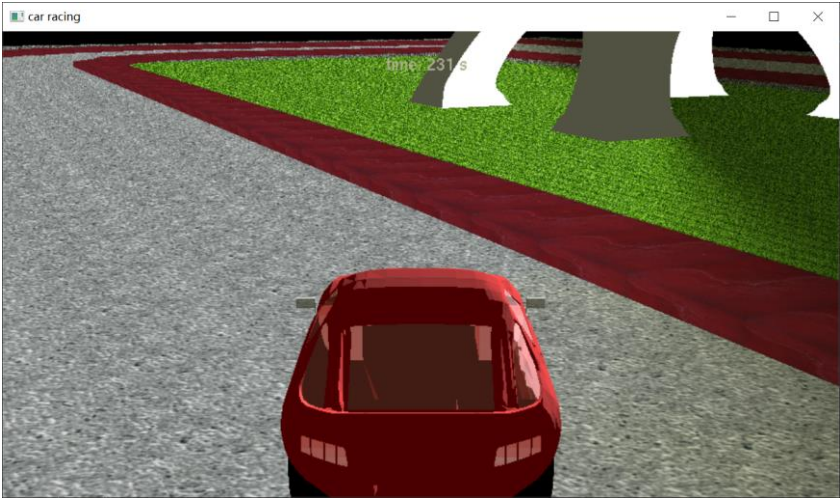
完成模块：OBJ 格式文件的寻找和读取，纹理材质，场景优化

涉及到的技术点：

1. 存储：OBJ 格式的读取

首先要寻找到合适的 OBJ 模型，由于画图时使用的是三角形面片，有些 OBJ 格式模型中部分面片是四边形的，要先进行三角化。随后按照 OBJ 的格式读取点的坐标和面的顶点

索引至内存中，这个步骤在初始化时进行。绘图时对每个面片按照索引取出点的坐标。
最终的游戏使用 obj 添加了埃菲尔铁塔和小车。



2. 复杂材质效果：材质颜色处理

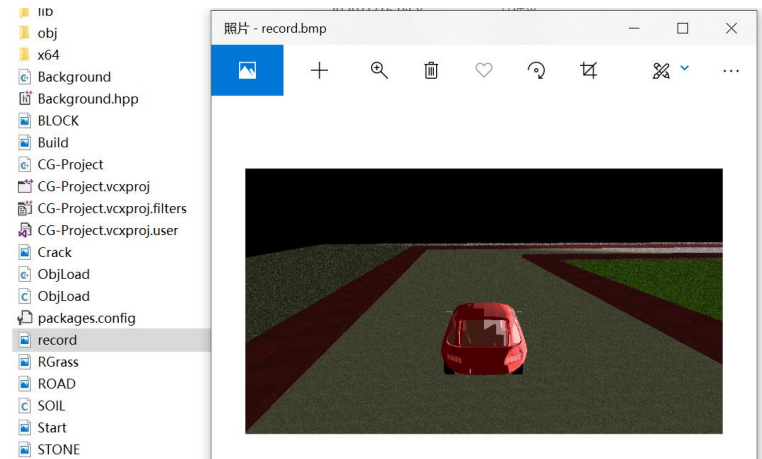
将小车分为了车身、玻璃、其他三个部分，分别使用不同的材质和颜色参数。玻璃是半透明的且有较大的镜面反光成分；车身是红色，以漫反射和环境颜色为主；其他部分是黑色。

Alon Kagan

实现模块：提供屏幕截取/保存功能

涉及的技术要点：

1. 存储:提供了实时截屏功能，将截屏文件保存在文件夹中。



总体来说，除了 7 条基本要求外，我们的项目细节中还涉及 NURBS、碰撞检测、复杂光照明模型、复杂材质这几个高级要求，并在答辩之后，根据反馈对当时提出的问题逐一进行了修正。在分工比例上，我们对内的五名同学参与讨论与任务完成都非常积极，对于项目代码也各有相应的贡献，并且共同完成了报告部分的撰写。

四、 结论与心得体会

本次课程大作业,我们运用了之前所学的知识,实现了三维场景建模与真实的 CG 世界,往里面加入了各种元素,不仅达到了项目的所有基本要求,也做了 5 个项目高级要求,显得很充实。组员间也分工明确,组中有外国小哥哥加入,虽然存在一定的沟通难度,但大家最终都完成了各自的任务,团队的完成了本次课程大作业。

我们也非常喜欢这次的课程大作业形式,自由度很高,我们选择做了一款赛车游戏。这也印证了计算机图形学课程的理论难度较高,但趣味性很足,这是我们做的第一款游戏,相信以后能运用所学知识做出更好的游戏。

五、 参考文献

[1] Computer Graphics with OpenGL