

**软件设计文档**

课程：软件设计综合实验

课题：3D迷宫漫游游戏

专业：2014级数字媒体技术

组名：free to play

组长：汪凯威

组员：庄建平，何文震，张昱婷，徐莹钶

文档管理信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 更新时间 | 更新人 | 主要更新内容 |
| 1 | 2017.4.30 | 汪凯威 | 创建文档，添加基本资料 |
| 2 | 2017.5.24 | 汪凯威 | 编写开发概况 |
| 3 | 2017.6.10 | 张昱婷 | 编写软件设计与接口规范 |
| 4 | 2017.6.23 | 汪凯威 | 编写模块设计 |
| 5 | 2017.6.24 | 汪凯威 | 最后的功能补充 |

目录

一、项目开发概况 …………………………………………………………………………………………...... 3

1、开发人员 …………………………………………………………………………………………........................ 3

2、开发计划 …………………………………………………………………………………………........................ 3

3、开发环境和工具 …………………………………………………………..…………………........................ 3

二、软件设计 …………………………………………………………………………………………………...... 3

1、概念术语 …………………………………………………………………………………………........................ 3

1.1、OpenGL ………………………………………………………………………………………….......................... 3

1.2、obj文件 ………………………………………………………………………………..……….......................... 3

2、基本设计 …………………………………………………………………………………………........................ 4

3、主界面流程 …………………………………………..…………………………………………........................ 4

三、接口规范 …………………………………………………………………………………………………...... 6

四、模块设计 …………………………………………………………………………………………………...... 7

五、软件设计技术 ………………………………………………………………………………………………. 8

一、项目开发概况

1、开发人员

|  |  |  |  |
| --- | --- | --- | --- |
| 角色 | 职责 | 负责内容 | 人员 |
| 项目经理 | 项目设计  进度控制 | 项目框架编写  软件设计文档编写 | 汪凯威 |
| 产品经理 | 定义需求  产品监督 | 软件需求规格说明书编写  软件设计文档编写 | 张昱婷 |
| 技术经理 | 技术规划  核心功能实现 | 主程序编写  纹理映射、摄像机漫游 | 庄建平 |
| 程序员 | 软件编码  额外功能实现 | 光照功能编写  法向量与材质 | 何文震 |
| 测试工程师 | 监控软件质量  测试与改进建议 | 用户手册编写  安装部署说明编写 | 徐莹钶 |

2、开发计划

|  |  |  |
| --- | --- | --- |
| 项目总体计划与阶段 | | |
| 开始日期 | 结束日期 | 项目计划内容 |
| 4月30日 | 5月24日 | 实现基本元素：墙壁生成、摄像机放置、运动与碰撞 |
| 5月25日 | 6月14日 | 加入附加功能：纹理映射、光照、天空盒、视角切换 |
| 6月15日 | 6月23日 | 尝试额外功能：模型管理，显示数据列表，碰撞检测 |

3、开发环境和工具

开发工具

|  |  |
| --- | --- |
| 工具 | 功能 |
| Microsoft Visual Studio | 代码编写，基本环境配置，项目生成 |
| Photoshop | 贴图和纹理编辑 |
|  |  |

二、软件设计

1、概念术语

1.1、OpenGL

OpenGL（全写Open Graphics Library）是指定义了一个跨编程语言、跨平台的编程接口规格的专业的图形程序接口。它用于三维图像（二维的亦可），是一个功能强大，调用方便的底层图形库。

1.2、obj文件

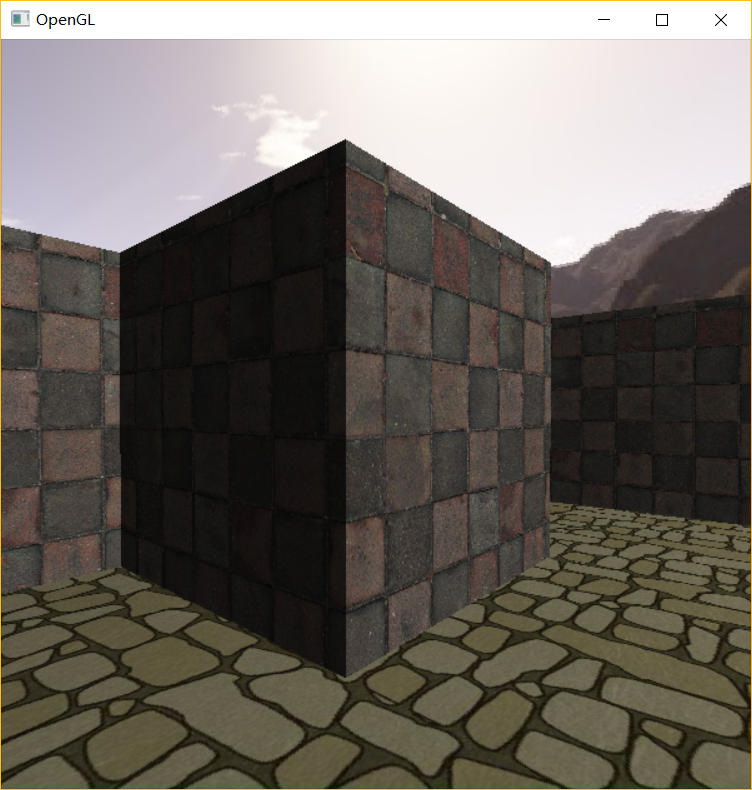
3D模型文件格式。由Alias|Wavefront公司为3D建模和动画软件"Advanced Visualizer"开发的一种标准，用于3D软件模型之间的互导，也可以通过Maya读写。

2、基本设计

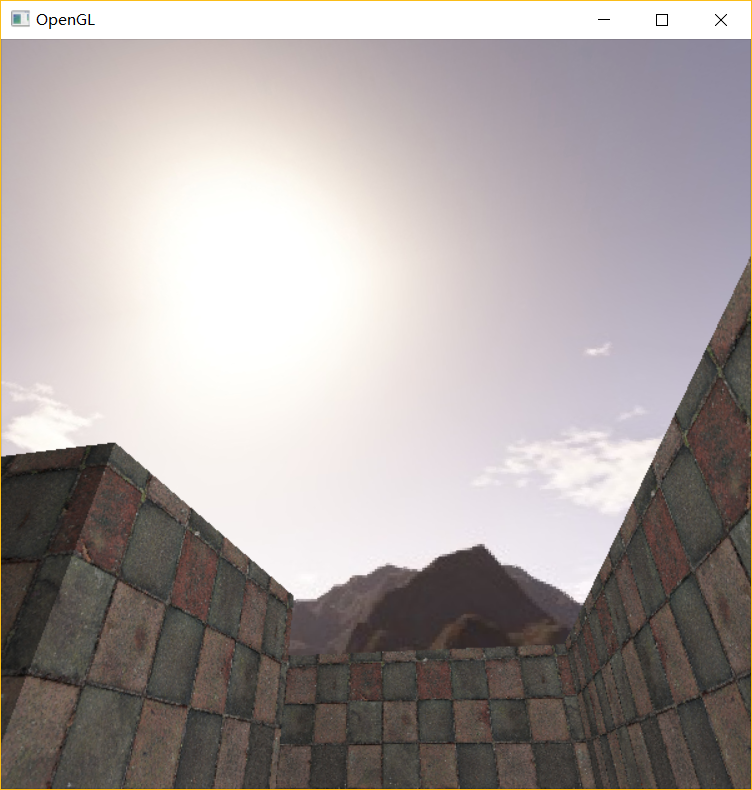
使用基于C++的Opengl库创建图形绘制窗口，绘制立体迷宫模型，为模型墙壁、地面创建纹理映射，在迷宫外创建天空盒，实现第一人称视角漫游，将摄像机置入迷宫内，用鼠标和键盘进行操作和运动，走出迷宫。为迷宫加上光源和obj模型，使环境更加逼真。

3、主界面流程

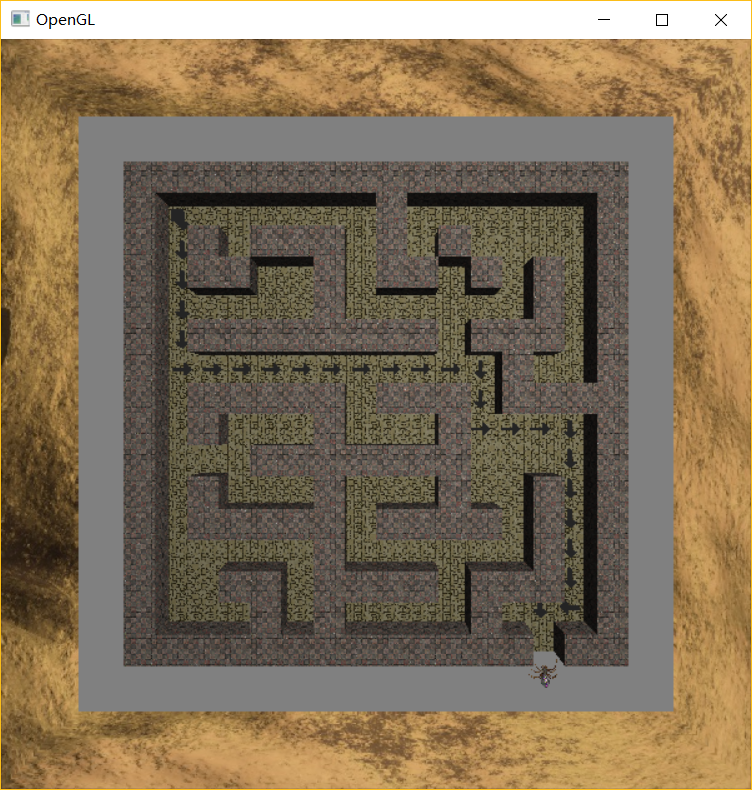
游戏内没有文字提示。进入游戏程序后，使用方向键控制前、后、左、右移动。



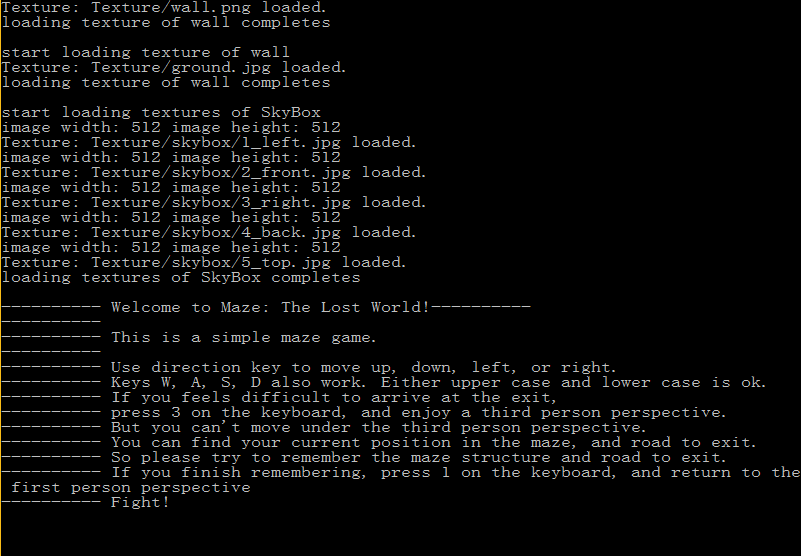
鼠标左键拖动旋转视角，改变运动方向。



数字键1、3切换第一人称运动视角和第三人称俯瞰作弊视角。



命令行窗口输出资源加载情况信息与简单操作指令。



迷宫出口有一个3D蜘蛛模型。走出迷宫，游戏结束。

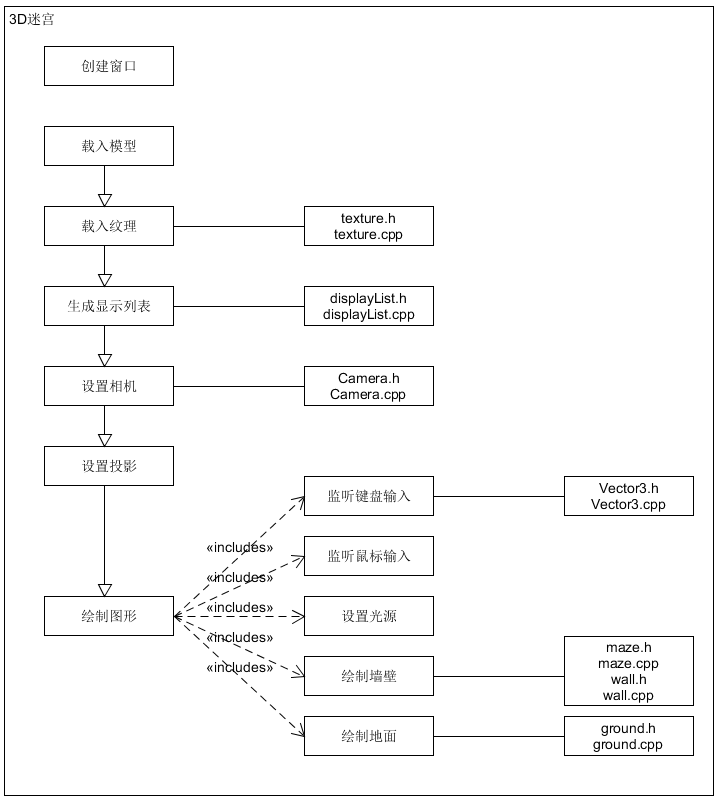


三、接口规范

|  |  |  |
| --- | --- | --- |
| 模块 | 接口 | 描述 |
| Assimpload | std::string AssimpLoad::getBasePath |  |
| void AssimpLoad::color4\_to\_float4 |  |
| void AssimpLoad::set\_float4 |  |
| void AssimpLoad::get\_bounding\_box\_for\_node |  |
| void AssimpLoad::get\_bounding\_box |  |
| void AssimpLoad::apply\_material | 设置材质 |
| void AssimpLoad::recursive\_render | 设置法向量和位置 |
| void AssimpLoad::do\_motion | 每帧结束后操作 |
| bool AssimpLoad::LoadModel | 读取模型 |
| bool AssimpLoad::LoadTextures | 读取纹理 |
| void AssimpLoad::Display | 每帧显示 |
| bool AssimpLoad::Load | 调用模型和纹理 |
| void AssimpLoad::SetMaterialColor |  |
| bool AssimpLoad::LoadWithoutText |  |
| void AssimpLoad::Clear |  |
| AssimpLoad::~AssimpLoad |  |
| Camera | Camera::Camera |  |
| Camera::~Camera |  |
| void Camera::setCamera |  |
| void Camera::moveW |  |
| void Camera::moveS |  |
| void Camera::moveA |  |
| void Camera::moveD |  |
| void Camera::moveMouse |  |
| void Camera::moveSkyBox | 保持天空盒静止 |
| Vector3 Camera::getPostion |  |
| void Camera::setPosition |  |
| Vector3 Camera::getCenter |  |
| void Camera::setCenter |  |
| Vector3 Camera::getUp |  |
| void Camera::setUp |  |
| Vector3 Camera::getTranslate |  |
| void Camera::disableCollision |  |
| bool Camera::collision | 碰撞检测 |
| displayList | void generateDisplayList |  |
| ground | void drawGround |  |
| maze | char mazeData[MAZE\_HEIGHT][MAZE\_WIDTH] |  |
| skyBox | void drawSkyBox |  |
| texture | void loadTexture |  |
| void loadSkyBoxTexture |  |
| Vector3 | Vector3::Vector3 |  |
| void Vector3::normalize |  |
| Vector3 Vector3::normalVector |  |
| Vector3 Vector3::operator - |  |
| Vector3 Vector3::operator + |  |
| Vector3 Vector3::operator \* |  |
| wall | void drawWalls |  |
| void drawWallUnit |  |
| void drawWallFace |  |
| arrow | void drawArrow  void drawRoadToExit |  |

四、模块设计

|  |  |
| --- | --- |
| 模块名称 | 设计与描述 |
| Assimpload | 获取纹理映射，材质，光照等参数，存储纹理ID |
| Camera | 设置摄像机运动接口，监控鼠标和键盘操作，进行碰撞检测 |
| displayList | 创建显示列表 |
| ground | 绘制地面，创建纹理映射 |
| maze | 存储迷宫地图信息。可修改迷宫结构 |
| skybox | 绘制天空盒，创建天空盒纹理 |
| texture | 加载纹理贴图 |
| Vector3 | 创建坐标操作 |
| wall | 读取迷宫数据，绘制墙壁，进行贴图 |
| arrow | 绘制第三人称视角下的作弊导航箭头 |
| main | 加载图形，载入模型、纹理、相机、光源等，调用绘制函数 |



五、软件设计技术

Structure Programming：

在整个代码设计中应用较多。尤其在main程序中，适应OpenGL本身特性，使用了大量结构化代码支持主绘图程序运行。

glutDisplayFunc(display);

glutMouseFunc(mouse);

glutMotionFunc(motion);

glutKeyboardFunc(keyboard);

glutSpecialFunc(specialKeyboard);

glutSpecialUpFunc(upSpecialKeyboard);

glutIdleFunc(idle);

glutReshapeFunc(reshape);

SetupLights();

glutMainLoop();

在wall模块中，墙壁绘制程序读取地图信息循环绘制，而绘制过程分成三个部分，分别负责绘制单个墙壁、绘制单个墙面纹理、循环绘制完成所有墙面，以提高可读性和可维护性。

void drawWalls()

void drawWallUnit()

void drawWallFace()

Object-Oriented Programming：

在Vector3模块中，为了更好的处理坐标与法向量，对Vector3对象进行了一系列定义，比较好地体现了面向对象的思想。Vector3是一个具有多态性的类，并重载了一些运算符。

class Vector3{}

Vector3::Vector3

Vector3::Vector3(float a, float b, float c)

Vector3::Vector3(const Vector3& vector)

Vector3 Vector3::normalVector

Vector3 Vector3::operator - (const Vector3& v)

Vector3 Vector3::operator + (const Vector3& v)

Vector3 Vector3::operator \* (float scale)

Design Patterns：

编码比较好地应用到了合成复用与接口隔离。源码根据接口实现功能分为多个头文件和实现文件，分别支持不同功能，可以分别修改实现。连续的处理过程和绘制，调用多个接口进行，没有过多使用继承方式。