

系统建模报告

这篇报告通过用例图、类图、序列图 and 状态图等多个方面，介绍“基于 OpenGL 的三维峡谷探险迷宫设计”项目的系统建模。

用例图将展示游戏中的各种用户行为和系统响应，从而明确定义游戏功能和用户互动。类图将详细描述游戏中的关键对象及其之间的静态关系，包括迷宫结构、玩家角色、道具和怪物等。序列图将逐步展示玩家如何与游戏中的元素交互，揭示游戏事件的时序和逻辑流程。最后，状态图将分析玩家在不同游戏状态下的行为及状态转换条件，帮助设计游戏的状态管理和动态展示。通过这些系统建模工具的深入分析，我们旨在为项目的设计和实现提供清晰的指导，确保游戏系统能够高效、稳定地实现预期的功能和用户体验。

【用例图】

系统的功能需求主要为以下四大方面：

一、绘制画面：通过玩家摄像头的视角观察三维迷宫，绘制二维画面。

具体包括：墙体装饰、机关陷阱、光照渲染、小地图。

二、位置移动：玩家可以控制人物的移动，从而实现对迷宫的探索。

具体包括：水平移动、跳跃下坠。

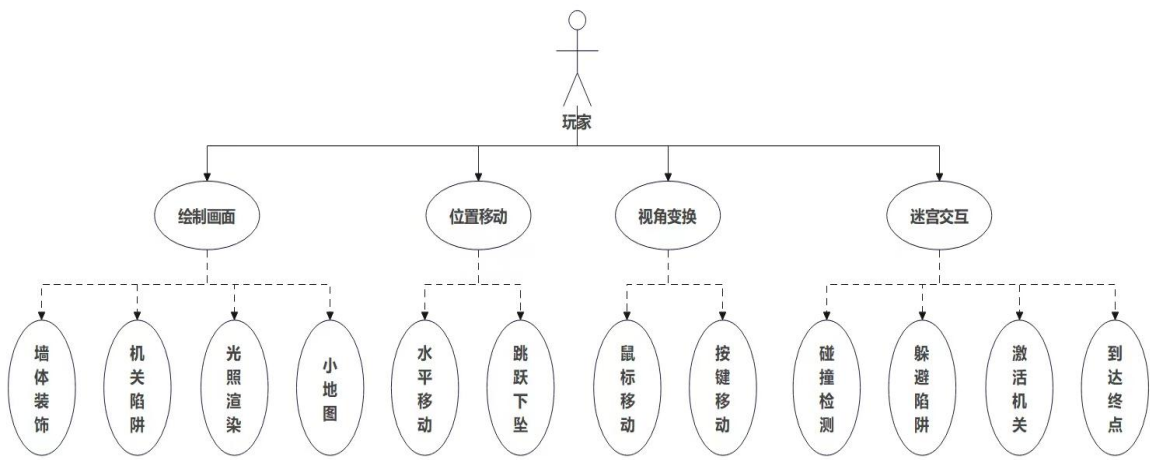
三、视角变换：玩家可以通过按键或鼠标来调整移动人物的视角。

具体包括：鼠标移动、按键移动。

四、迷宫交互：玩家与三维峡谷探险迷宫环境的互动。

具体包括：碰撞检测、躲避陷阱、激活机关、到达终点。

以下用例图描述系统的功能需求：



（图 1：系统的功能需求）

本次设计的 3D 迷宫游戏主要通过二维矩阵所表示的迷宫图，来构建三维立体场景。我们为系统设计了一架摄像机，来模拟用户在迷宫中的第一视角，用户可以通过鼠标、键盘控制自己在 3D 迷宫中的移动。

以下用例图描述用户与系统之间的交互：

按键事件	事件响应
W	向前方移动
S	向后方移动
A	向左方移动
D	向右方移动
I	视角向上移动
K	视角向下移动
J	视角向左移动
L	视角向右移动
Space	垂直方向跳跃
F	飞到高处

（图 2：用户与系统的交互）

【类图】以下类图描述系统中的静态结构，

即系统中的类、类描述、类成员，以及类之间的关系。

类名	Maze
类描述	用于表示和操作迷宫信息
类成员	Height, Width: 迷宫的高度和宽度。 xOrigin, yOrigin: 起点坐标。 xEnd, yEnd: 终点坐标。 maze[100][100]: 迷宫地图。 blockHeight[100][100]: 迷宫中每个元素块的高度。 isDynamicWall[100][100]: 标记墙壁是否是动态的。 dynamicWallProportion: 动态墙壁的比例。 wallsize: 墙体大小。 textures: 纹理对象。
关系	调用 Textures 类，获取纹理材质

类名	Textures
类描述	记录和管理纹理信息的类。
类成员	GLuint textures2D[20]: 存储 2D 纹理的数组。 GLuint texturesCube[20]: 存储立方体纹理的数组。 int count2D: 记录当前加载的 2D 纹理的个数。 int countCube: 记录当前加载的立方体纹理的个数。
关系	调用 OpenGL 库

类名	Player
类描述	存储游戏中玩家的相关信息和状态。
类成员	Maze* maze: 指向迷宫地图信息的指针，用于与迷宫交互。 Camera* camera: 指向摄像机对象的指针，控制玩家视角。 float posX, posZ: 玩家当前位置的 X 和 Z 坐标。 float targetXL, targetXR, targetZF, targetZB: 终点范围的边界值，用于检测玩家是否到达终点。 float invWallSize: 墙体倒数，用于计算与墙体的交互。 bool isJumping: 标志玩家是否正在跳跃。 float jumpStartTime: 记录跳跃开始的时间。 float currentTime: 当前时间。 bool isEnded: 标志游戏是否结束。 bool money: 标志玩家是否拥有金钱或其他可收集物品。
关系	Player 类持有 Maze 类和 Camera 类

类名	Camera
类描述	用于管理摄像机在游戏中的位置、方向和视角等信息。
类成员	<p>float mouse_sensitivity: 鼠标灵敏度, 用于控制鼠标移动时视角变化的速度。</p> <p>float fbward_speed、float vertical_speed、float horizontal_speed: 前后、垂直和水平移动的速度。</p> <p>float pitch_speed、float yaw_speed: 俯仰角和偏航角的变化速度。</p> <p>float pitch、float yaw: 俯仰角和偏航角的当前角度。</p> <p>vec3 worldUp: 世界坐标系中的上向量, 通常是指示世界中的上方向。</p> <p>float zoom: 视角大小, 用于控制摄像机的视野范围。</p> <p>vec3 position、vec3 in、vec3 right、vec3 front、vec3 up: 摄像机的位置和方向向量, 用于描述摄像机在世界空间中的位置和朝向。</p>
关系	调用 Vec3 向量类

类名	Vec3
类描述	表示三维向量, 包含 x、y、z 分量。
类成员	<p>float x, y, z: 三维向量的坐标分量。</p> <p>vec3(): 默认构造函数, 初始化向量为零向量。</p> <p>vec3(float xx, float yy, float zz): 构造函数, 初始化向量。</p> <p>vec3 operator-() const: 一元负号运算符重载, 返回向量的负向量。</p> <p>vec3& operator+=(const vec3& v)、vec3& operator-=(const vec3& v)、vec3& operator*=(const vec3& v)、vec3& operator/=(const vec3& v): 复合赋值运算符重载, 用于向量的加减乘除。</p> <p>vec3& operator*=(const float t)、vec3& operator/=(const float t): 与标量的乘除复合赋值运算符重载。</p> <p>float length() const: 计算向量的长度。</p> <p>float squared_length() const: 计算向量长度的平方。</p>
关系	工具类

类名	MyGLWidget
类描述	继承自 QOpenGLWidget, 创建一个基于 OpenGL 的图形窗口部件。
类成员	<p>bool keyStates[256]: 记录键盘按键状态的数组, 假设支持 256 个可能的键值。</p> <p>bool isJumping、isFalling、isFlying、isforwarding: 标记玩家当前是否在跳跃、自然下落、飞行、前进等状态。</p> <p>float forwardStartTime、jumpStartTime、fallStartTime、jumpInitial Height、fallInitialHeight、forwardinitial、jumpPeakHeight、jumpVelocity、moveVelocity: 记录跳跃、自然下落、前进等状态的时间和速度参数。</p> <p>float gravity: 重力加速度常量。</p> <p>int updateCounter、currentFrameForTrap、frameSizeForTrap、current FrameForWall、frameSizeForWall:</p>

	记录更新次数和动画纹理的当前帧数、总帧数等。 GLfloat lightPosition[4]: 光源的初始位置。 const float frameTime: 帧时间, 用于动画更新。 float skyboxAngle: 天空盒的旋转角度。
关系	游戏窗口

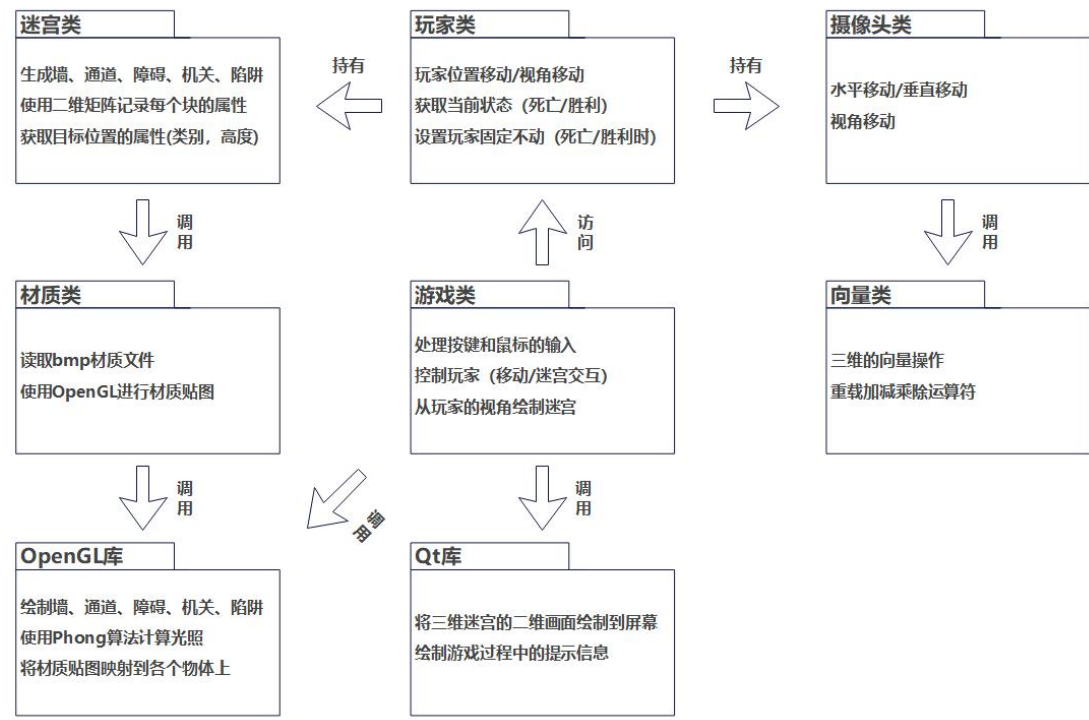
类关系总览:



(图 3: 类关系总览)

【序列图】

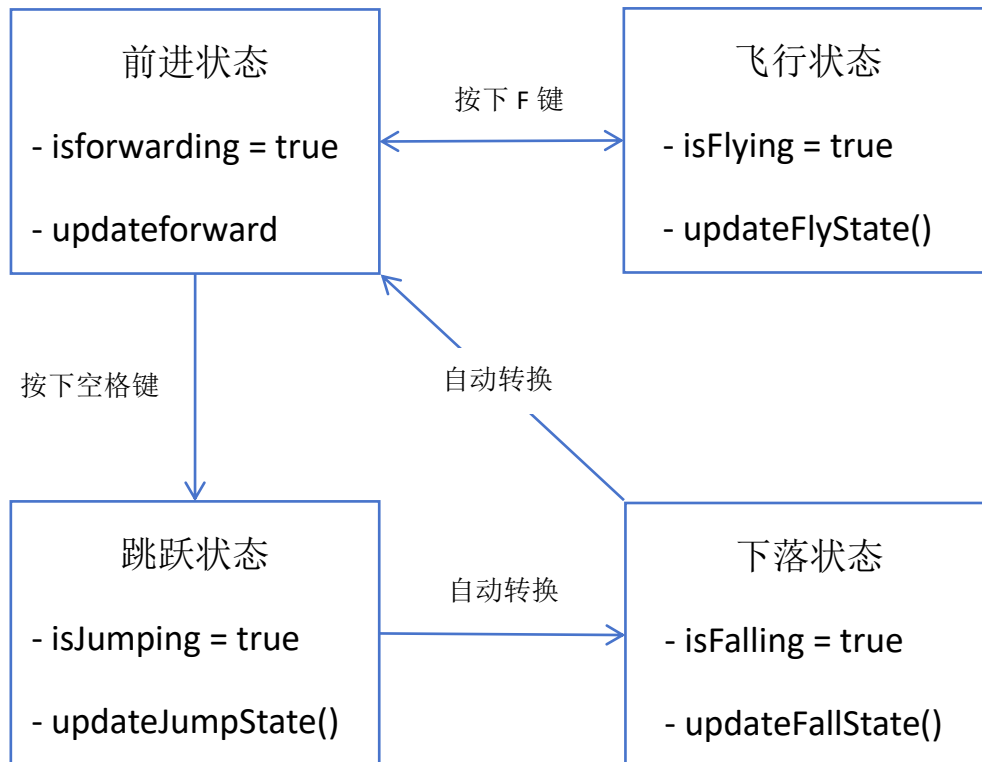
序列图帮助我们理解系统中不同类对象如何协作完成特定功能。



(图 3: 类对象之间的动态交互)

【状态图】

展示系统中对象的状态管理和事件触发机制。



(图 4: 状态转换和事件触发机制)

事件:

- `keyPressEvent(QKeyEvent* e)`
- `keyReleaseEvent(QKeyEvent* e)`
- `mouseMoveEvent(QMouseEvent* e)`