1. myglwidget.cpp void MyGLWidget::initializeGL() 91-135行【phong光照模型】



1. 对于maze.h，增加blockHeight，记录每个块的上表面的高度

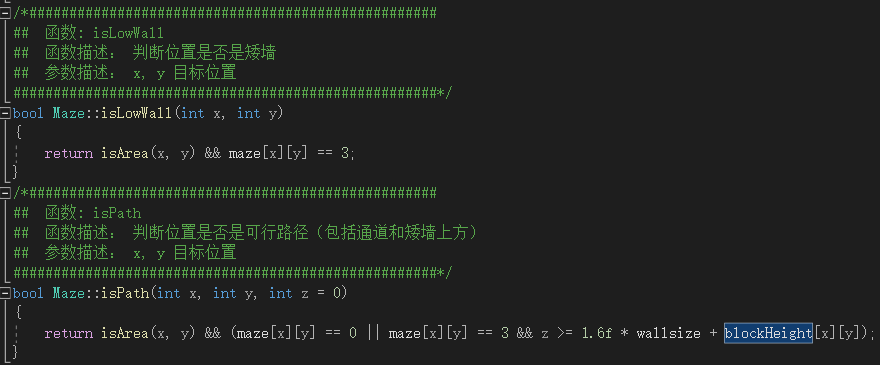


增加bool isLowWall(int x, int y) 矮墙的判断（3表示矮墙）

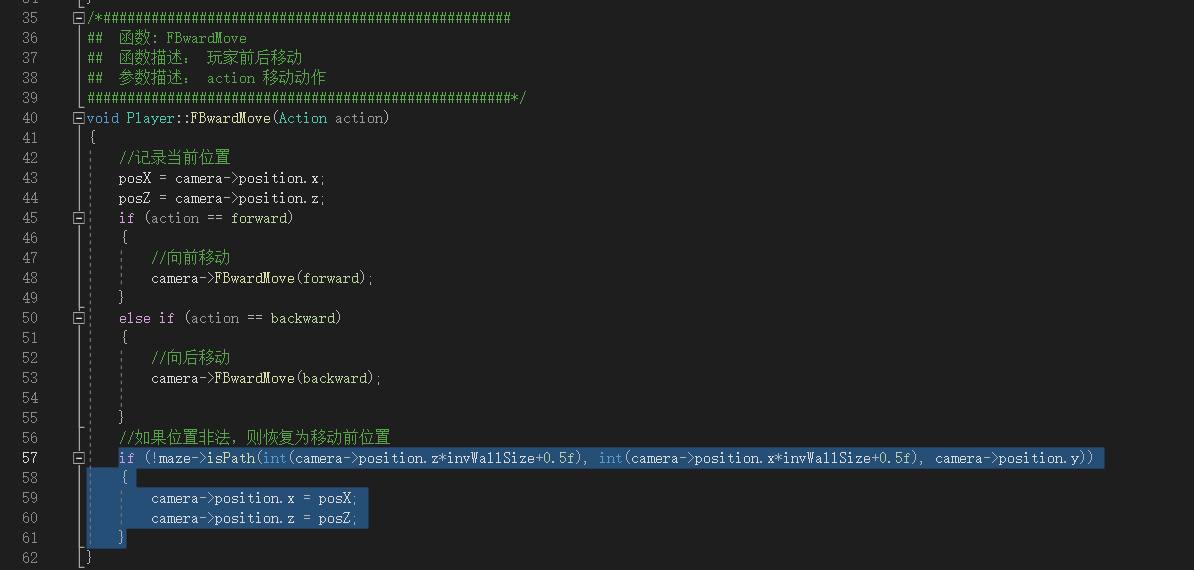
修改bool isPath(int x, int y); 为 bool isPath(int x, int y, int z);



对于maze.cpp，实现上述的修改。其中修改isPath，目的是为了实现**合法移动位置的检测**（即**z轴碰撞检测的一部分**）：当且仅当 （目标位置是通道） 或 （目标位置是矮墙 且 目标位置的高度 > 矮墙的高度）时，位置合法



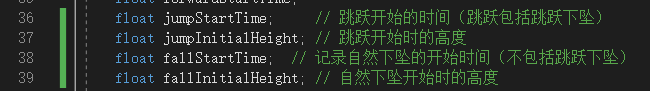
对应地，在player.cpp中，修改【判断移动是否合法】的逻辑

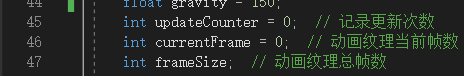


修改maze.cpp 154-163行，在可行通道上【随机生成矮墙】



1. 对于myglwidget.h，增加【自然下坠】和【动态纹理】的变量和函数。

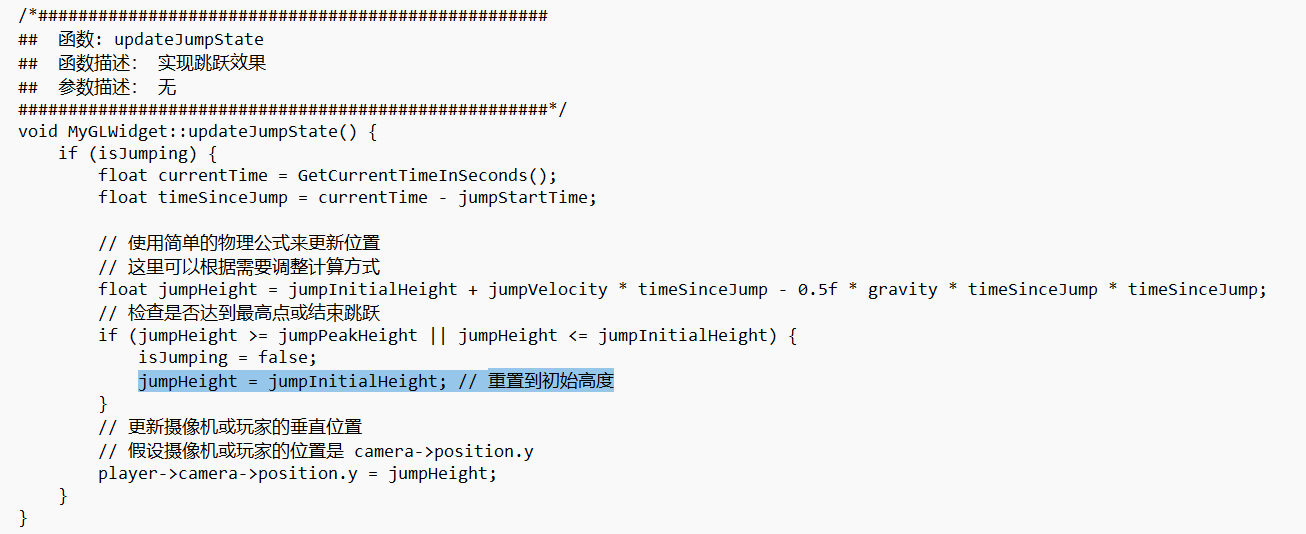




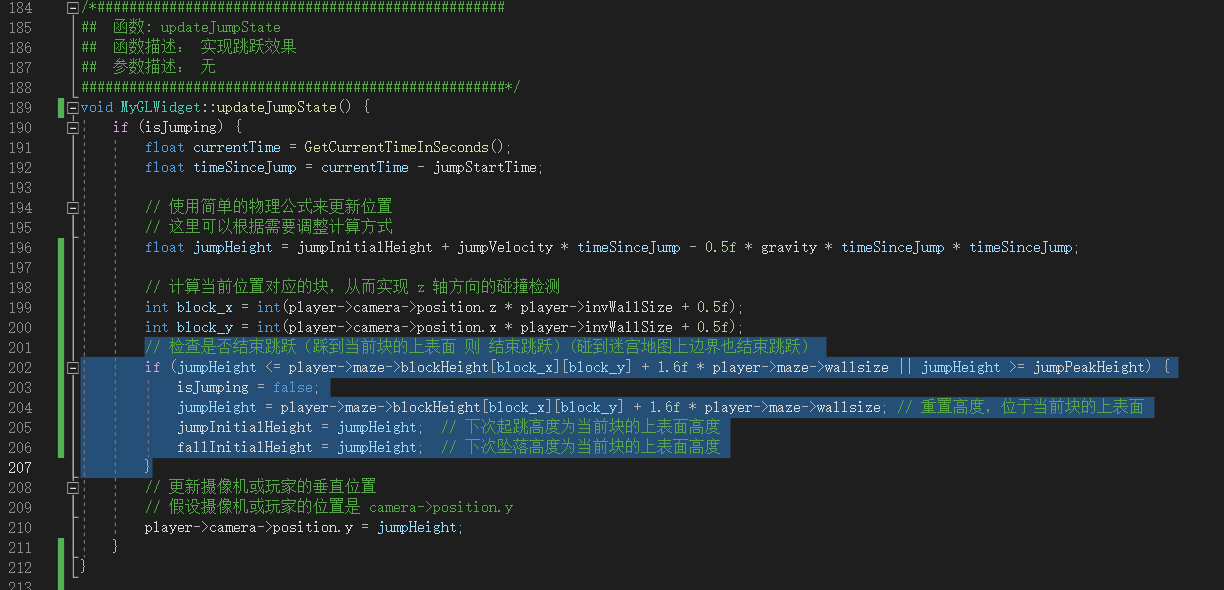


对于myglwidget.cpp，修改【跳跃】的逻辑，实现【自然下坠】和【动态纹理】的逻辑。

为什么修改【跳跃】的逻辑：原逻辑只适用于起点终点高度相同的跳跃

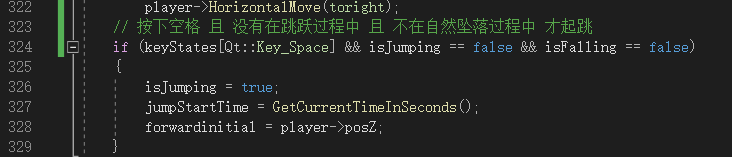


以下是修改后的逻辑：踩到当前块的上表面 则 结束跳跃，并重置起跳高度和坠落高度

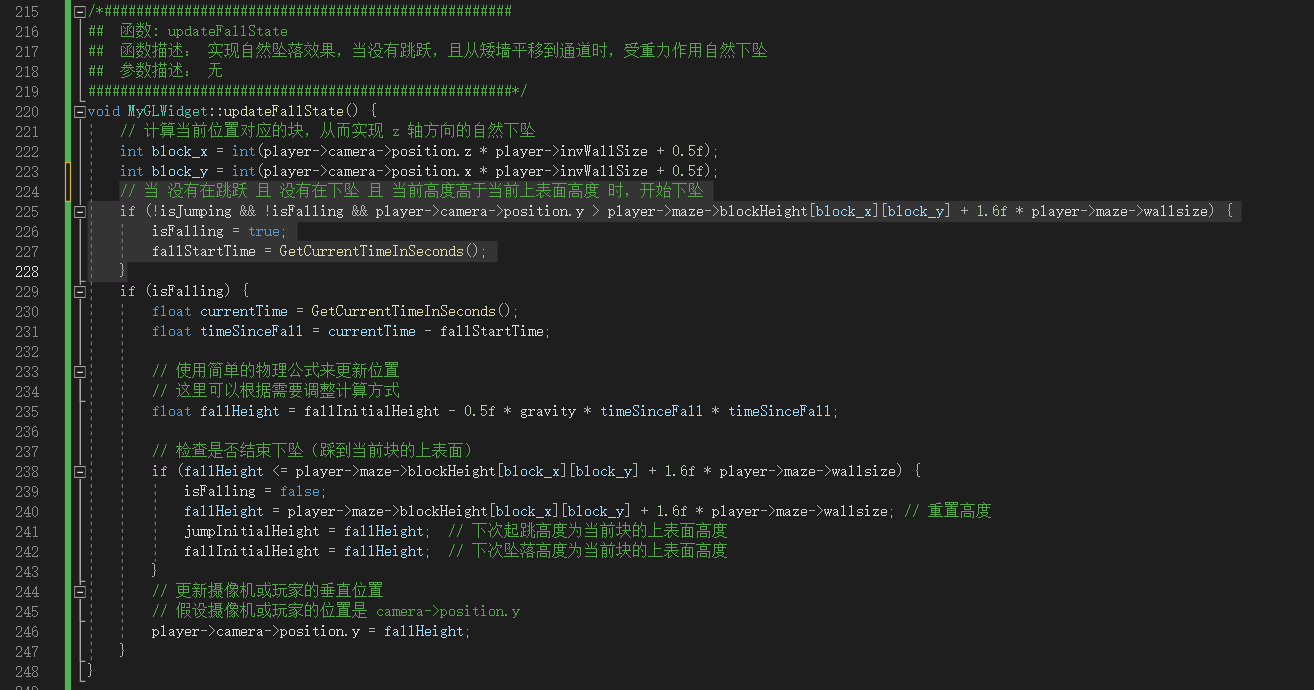


【跳跃】还有一个问题是重复跳跃的问题，key\_press那里修改为以下逻辑

按下空格 且 没有在跳跃过程中 且 不在自然坠落过程中 才起跳



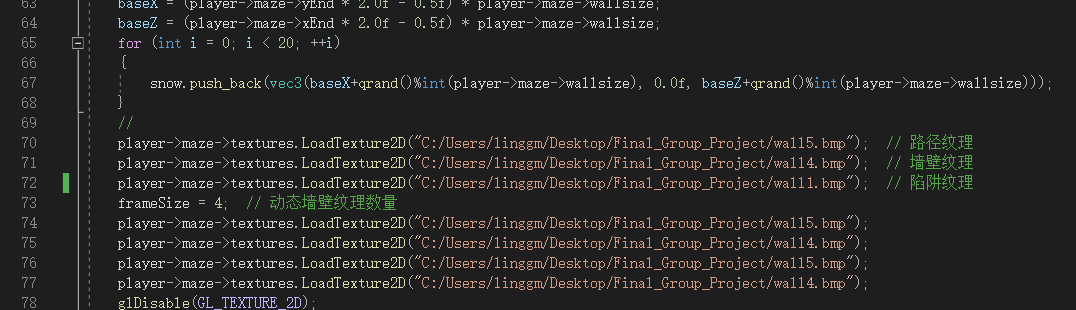
【自然下坠】：当从 矮墙所在块 移动到 通道所在块 时，应当受到重力作用而自然下坠到地面，从 较高矮墙 移动到 较矮的矮墙 时，也应下坠



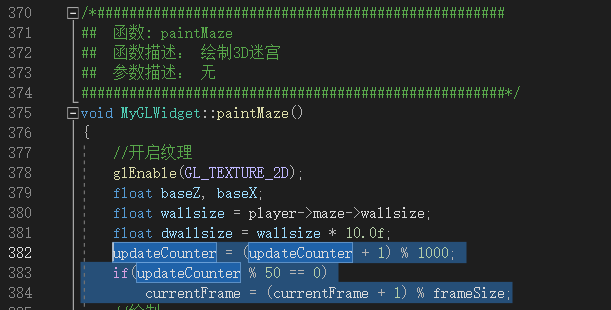
在 PaintGL函数中调用updateFallState，从而每16ms调用一次这个函数，更新下坠状态。



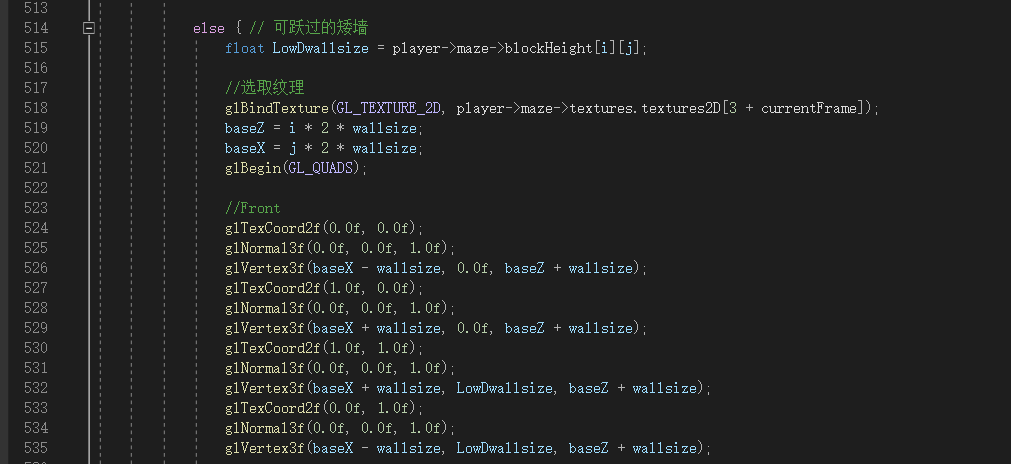
1. 动态纹理 void MyGLWidget::initializeGL() 中读取动态纹理对应的纹理



void MyGLWidget::paintMaze()中每1000次更新（每1.6s），切换纹理的一帧

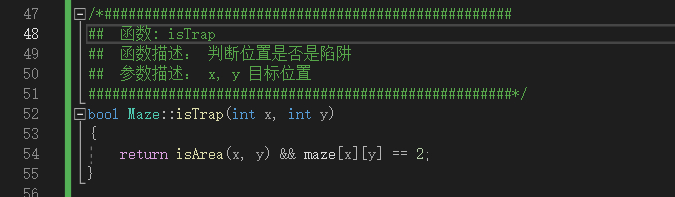


然后绘制矮墙

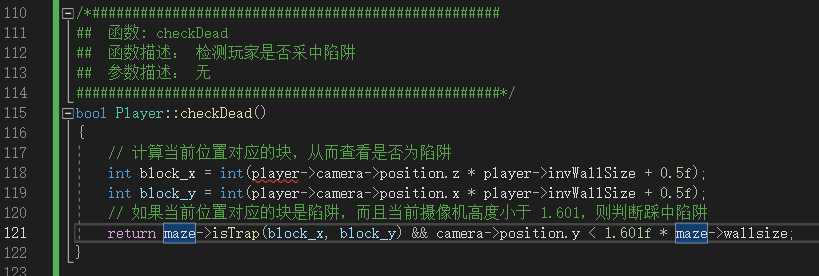


1. 实现【踩中陷阱即死亡】

Maze.h增加 bool isTrap(int x, int y); 并在cpp中实现



Player.h增加 bool checkDead(); 并在cpp中实现



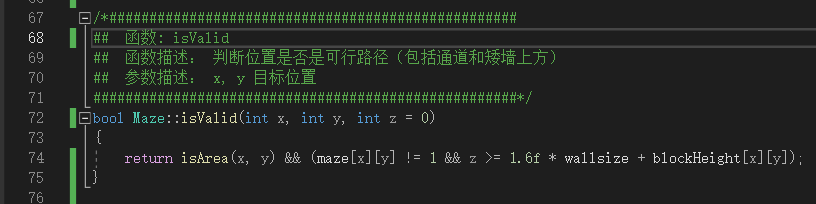
myglwidght.cpp的void MyGLWidget::paintGL()cpp每次更新，检查是否踩陷阱



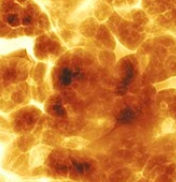
修改maze.cpp中的bool Maze::isPath(int x, int y, int z = 0)，

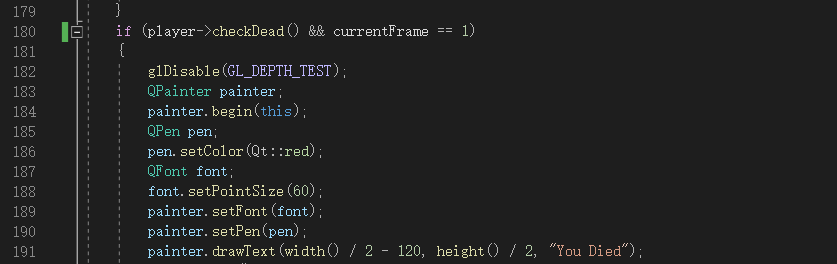
重命名为 bool Maze::isValid(int x, int y, int z = 0)

只要 当前块不是墙 且 脚高度高于上表面高度，则判断移动合法

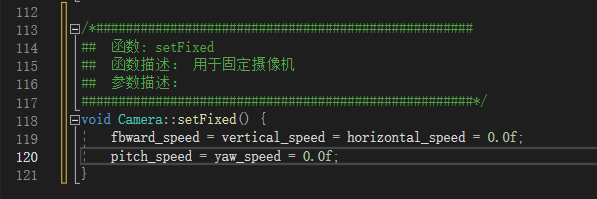


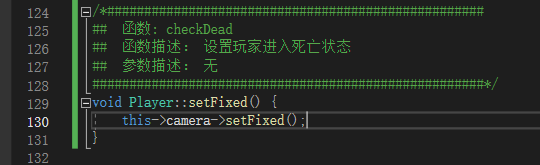
修改：实现动态陷阱，当岩浆溢出且踩上去，则死亡；当岩浆没有溢出，踩上去不会死亡。左图没有溢出，右图溢出





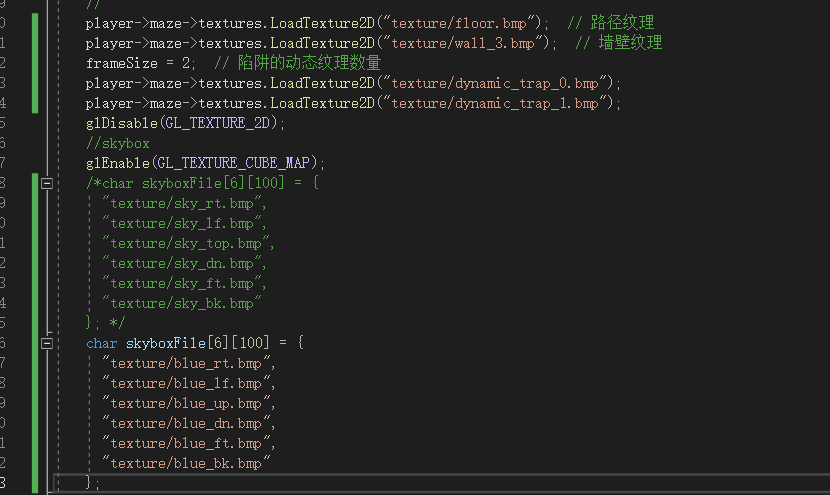
增加【死亡】或者【到达终点】后，玩家不能移动的逻辑



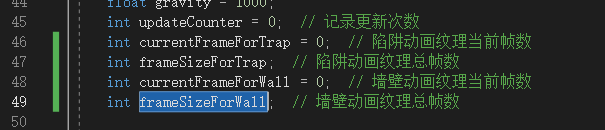


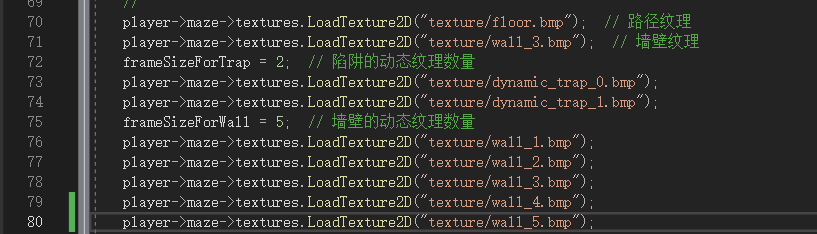
1. 天空盒的设置：代码中按照右，左，上，下，前，后的顺序加载



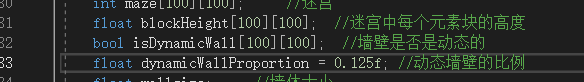


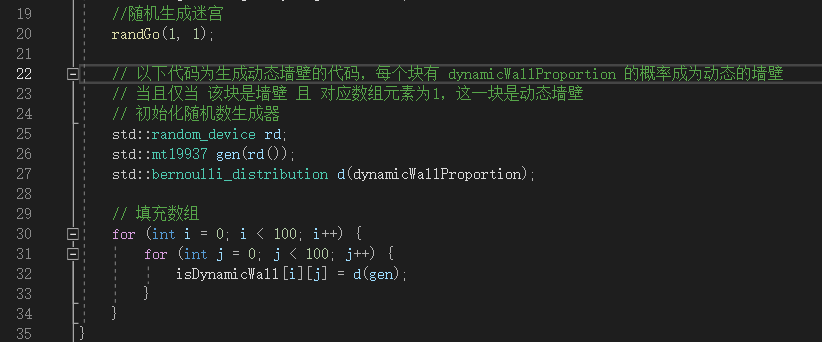
1. 墙壁动态纹理 myglwidght中，读取





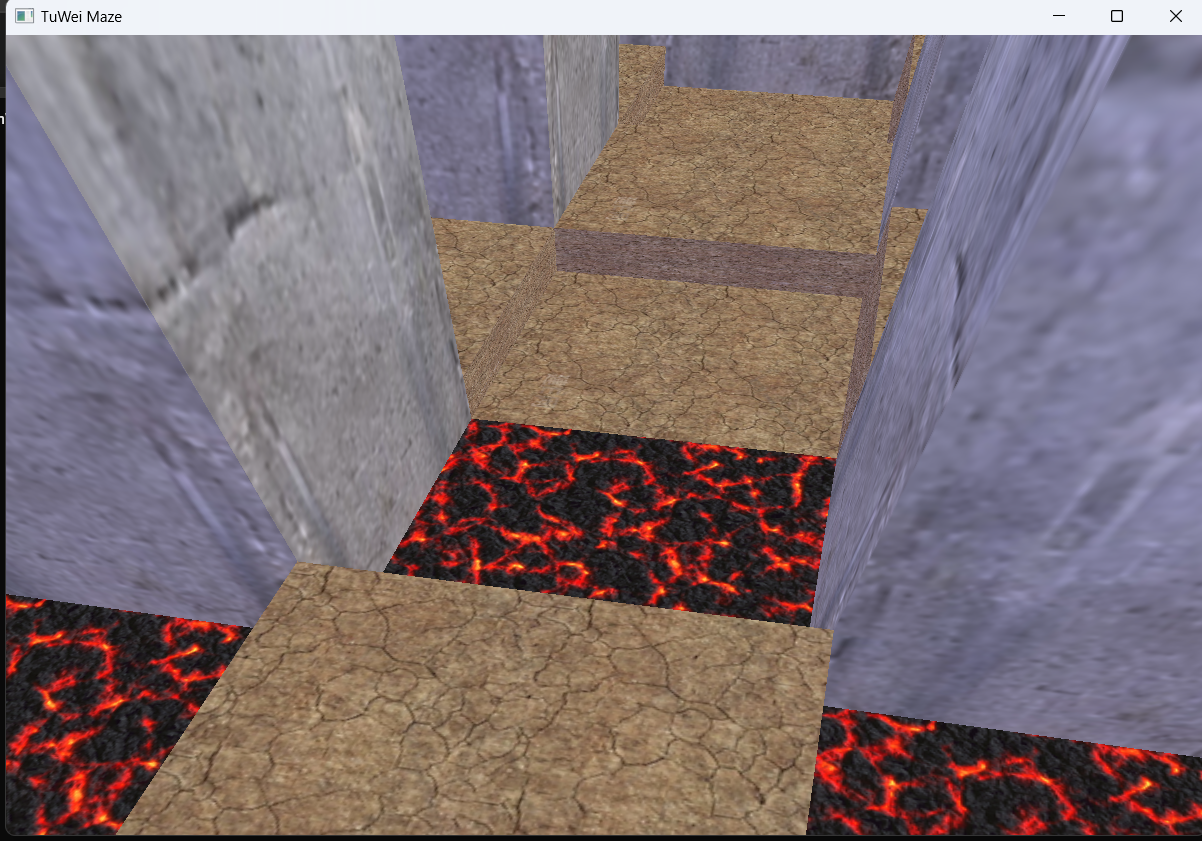
Maze中







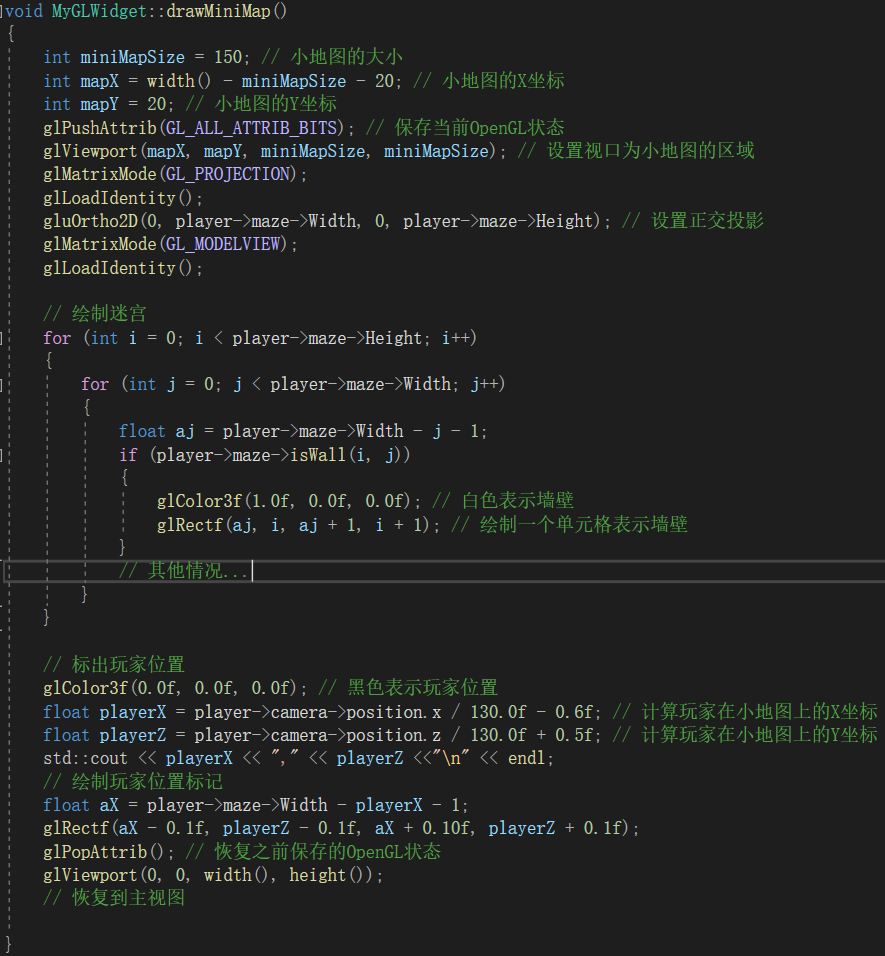




1. 增加一个飞行的彩蛋可以更好地看到天空盒，按F触发：

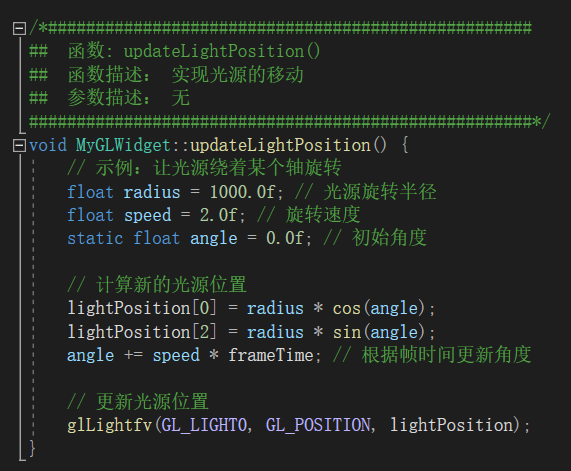


1. 增加了小地图更方便移动：

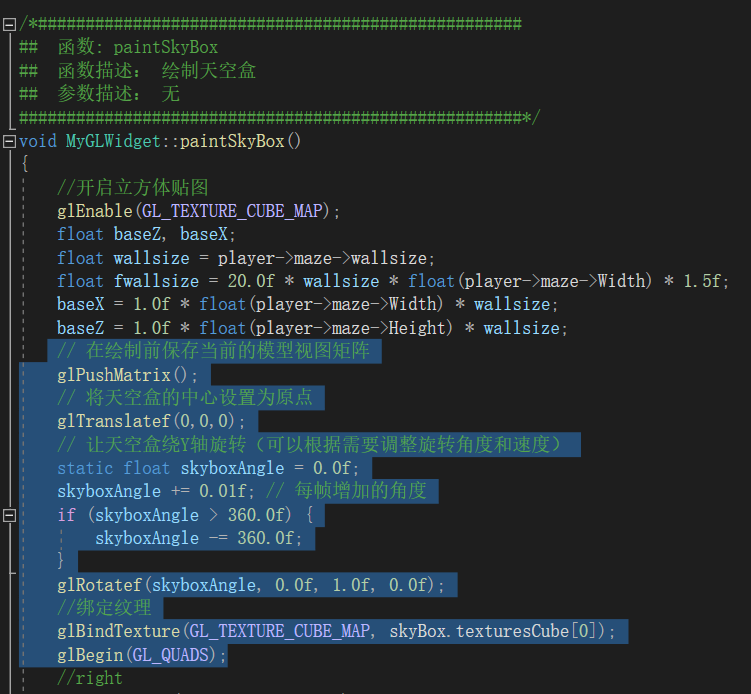


1. 增加了天空盒的旋转与光源的移动，优化了光影的效果

光源的移动



天空盒的旋转：



1. 添加机关，只有激活了机关到达终点才可以结束；否则到达终点会提示请激活机关

