**C语言项目报告**

**项目名称：\_\_蚂蚁回家小游戏\_\_\_\_\_\_\_\_\_**

**项目成员：\_\_白子豪 刘禹辰 杨楚萌\_\_\_\_\_\_**

**填写日期：\_\_2020.9.9\_\_\_\_\_\_\_\_\_**

1. **摘要（Abstract）**

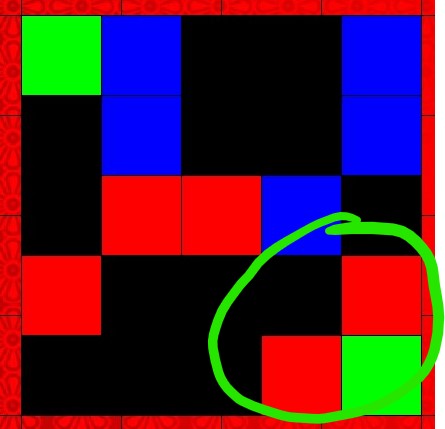
**本项目的最初灵感来源于一款记忆力小游戏。本游戏的基本设定为一只小蚂蚁需要避开所有的蜘蛛，并且吃掉所有的食物，最终回到家中，即判定为游戏胜利。该游戏中玩家共有三条生命，每遇到一次蜘蛛便会减少一条生命，生命值为0时即判定失败。**

**由于这是一款记忆力小游戏，在设定中我们增加了迷雾功能，即玩家需要开局记住随机地图的信息，随后在移动第一步之后地图会被迷雾遮盖，玩家需要凭借记忆完成游戏。为降低游戏难度，我们会适当给予玩家提示，即离玩家最近蜘蛛和食物的距离。**

**本作品的另一特色为体感控制，在脑力开发的同时兼顾体育锻炼。本游戏采用人体关键点识别AI技术，通过人体动作检测控制游戏。**

**本项目的创新点在于：**

**1.随机地图的生成，其中运用了并查集的数据结构去除了如下图所示的包围圈。**



**2.广度优先搜索算法（bfs）的应用，用来求算最短路径。**

**3.与计算机视觉结合，采用多线程处理，通过前置摄像头捕获玩家的运动状态来操控游戏。**

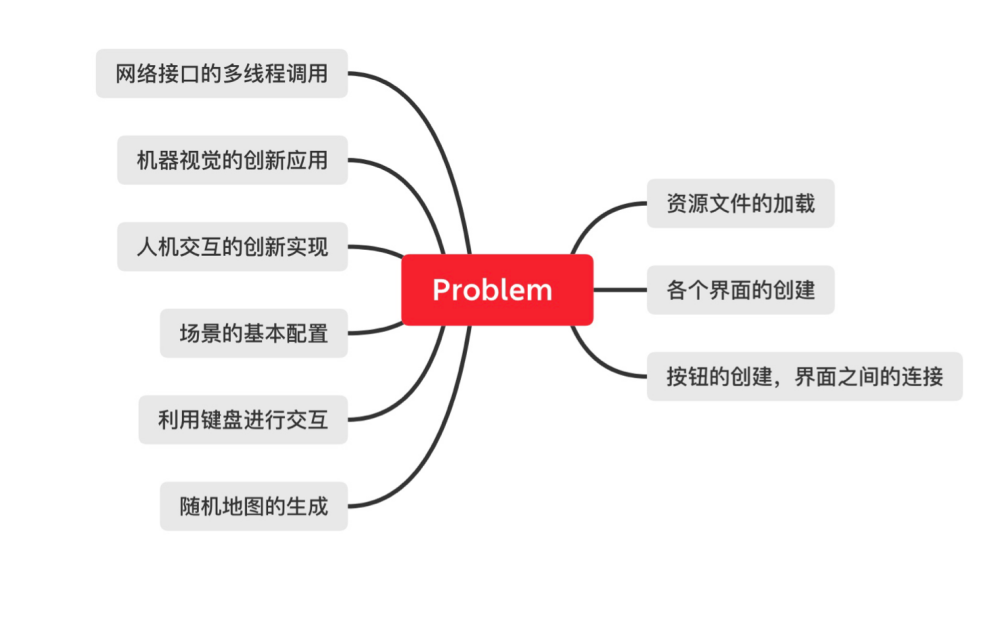
**4.** **整个体感模块的设计符合工程开发原则，符合鲁棒性，健壮性，可维护性，做了相当多的优化设计，在流程图中大致有所体现。**

1. **问题描述（Problem Statement）**

**考虑到唐班同学学习繁忙，又缺少体育锻炼。我们希望能够实现唐班同学盼望已久的愿望：在体育锻炼的同时又能开发脑力。**

**本项目采用C++和QT开源框架进行开发，采用物联网/嵌入式等作为辅助开发。**

**经过分析，本项目可被拆解为以下几个子问题。**



**其中，右边三个问题可归结为QT开发平台的应用，我们可以利用QT中自带的库来进行游戏各个界面的初始化创建。**

**左边前三个问题可归结为人机交互设计。**

**左边后三个问题可归结为游戏逻辑的设计，即游戏主场景算法的设计。**

1. **组内分工（Group Division）**

**依照以上三个大问题，本组具体项目分工如下：**



**其中，**

**白子豪代码量1105行**

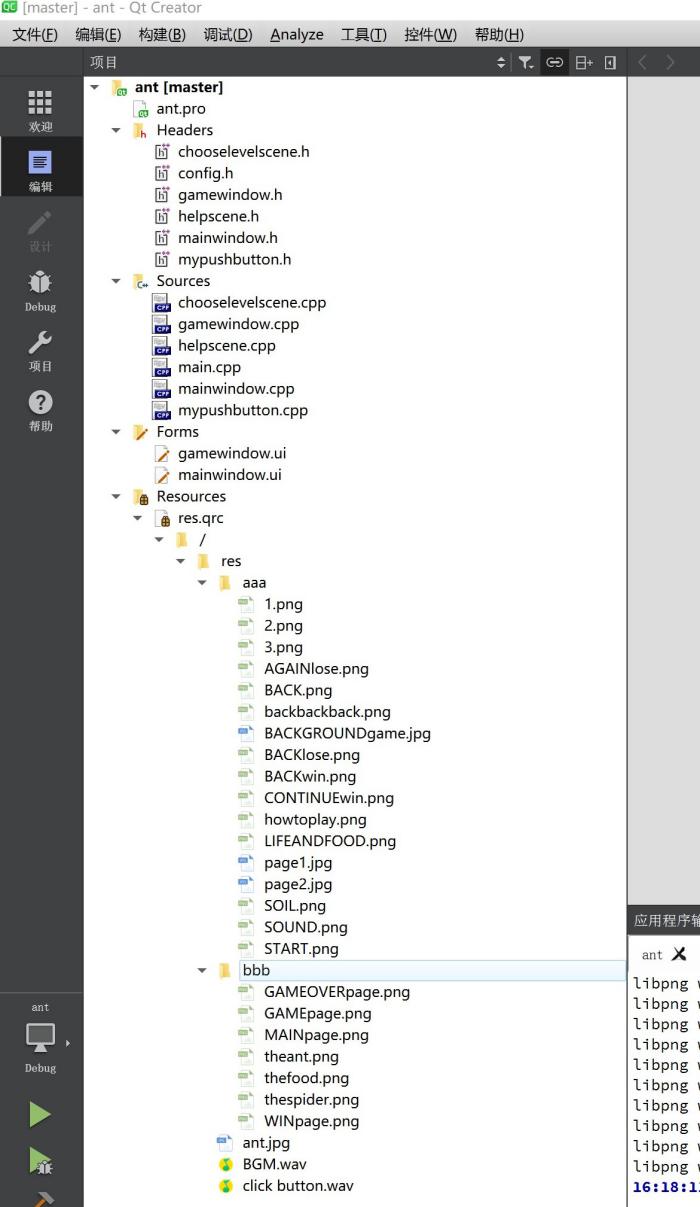
**刘禹辰代码量523行**

**杨楚萌代码量502行**

1. **分析（Analysis）**

**本项目共分为两个大部分进行，即游戏本体的设计以及体感交互模块的设计。**

**在游戏本体部分，该游戏由四个界面构成，各个部分的代码均在各自的界面进行完成，每个界面均有一个.h 头文件和一个.cpp文件。**



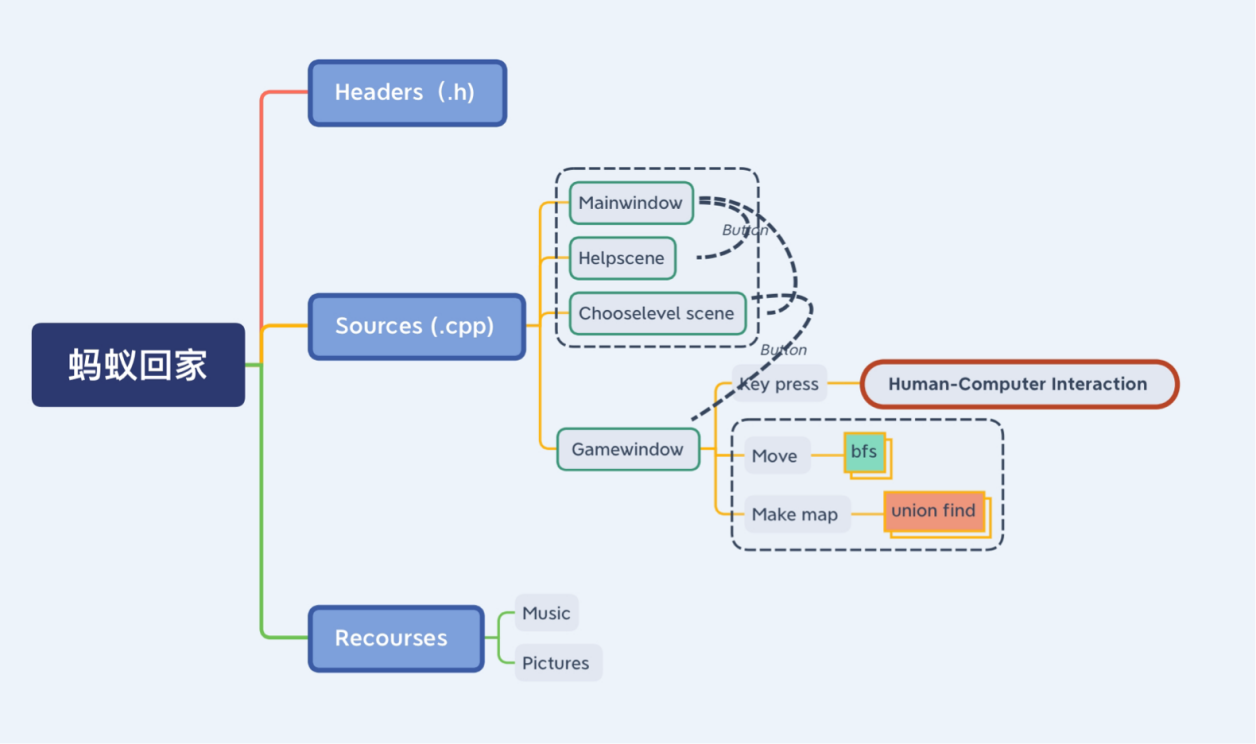
**我们利用了QT自带的信号槽工具，结合精美的图片，创建了功能按钮，实现了各个界面之间的连接。**



项目的总体框架设计逻辑

**在体感模块中，我们采集相机图像，采用多线程并发调用网络人工智能接口，从而返回人体骨骼关键点。在本地进行人体动作计算，触发键盘控制游戏。**

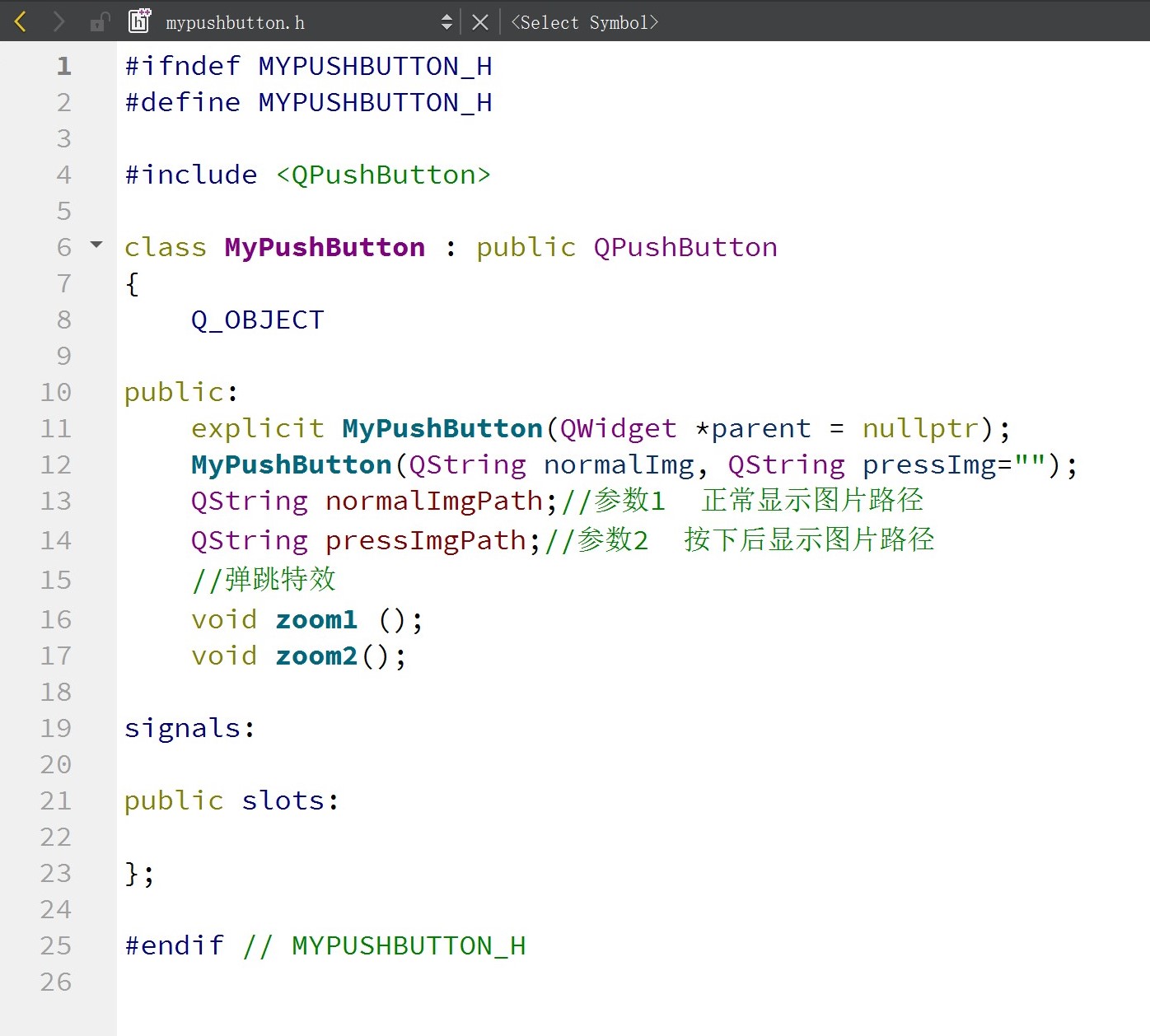
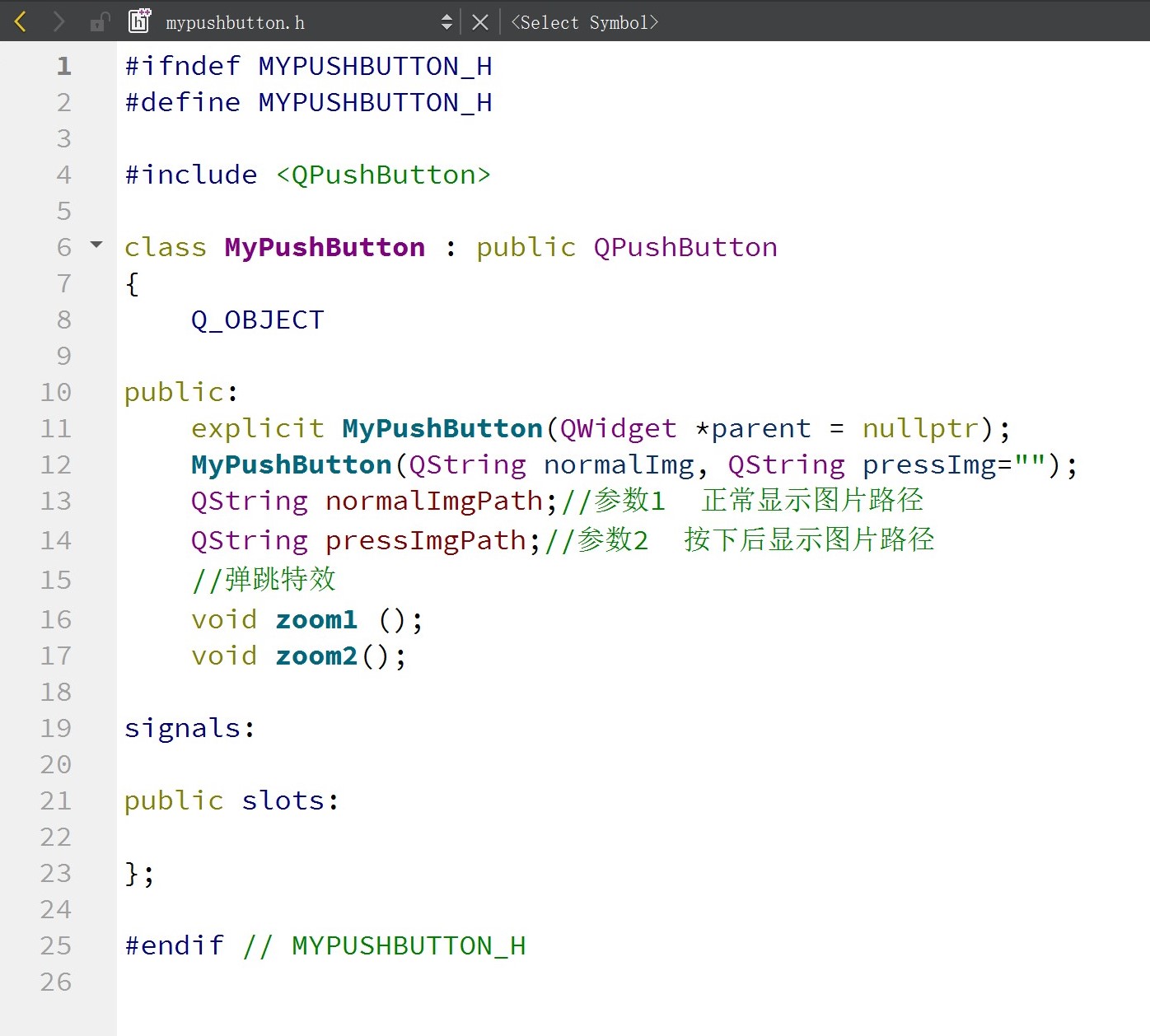
1. **设计（Design）**

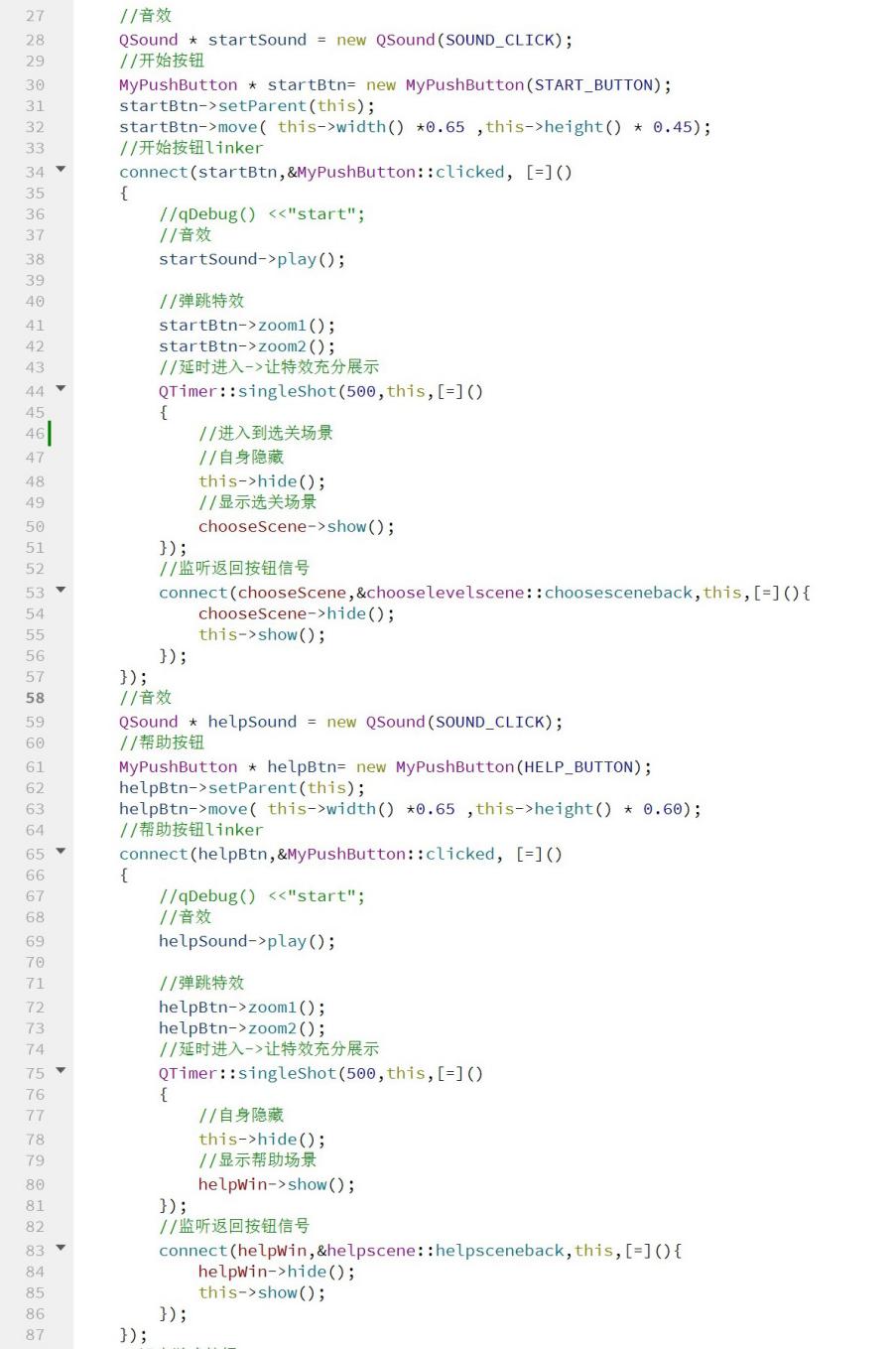




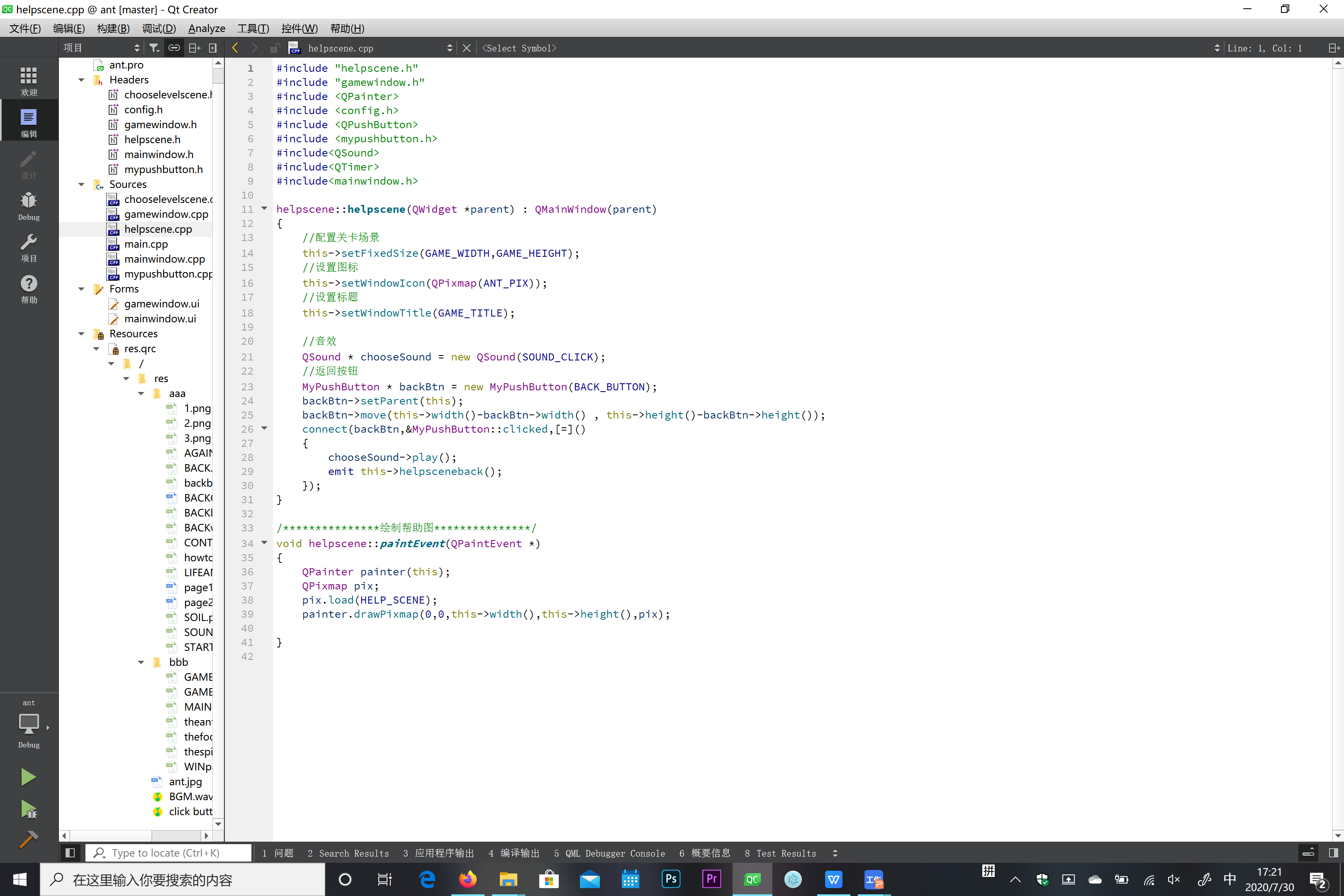
1. **实施（Implementation）**
2. **可视化界面的创建**

**1.1按钮的封装**

**为了代码的简洁美观，我们创建了mypushbutton.h和mypushbutton.cpp这两个文件，以封装游戏中出现的所有按钮，我们定义了自己的按钮的一个类，它继承于QPushbutton,同时定义了自己的成员函数来实现上下弹跳的功能，从而在需要使用按钮挂件时直接使用mypushbutton这个类来执行操作。**

**1.2 各个界面的连接**

**在界面中我们设置了几个按钮，用信号和槽来进行窗口与窗口之间的连接，connect函数可以将点击按钮的信号和隐藏这一界面、显示下一界面联系到一起，我们看到在开始和帮助按钮的connect函数的槽函数中还有一个connect函数，它把监听到的返回信号和槽函数联系到一起，退出游戏按钮的的槽函数是close，可以将点击按钮和关闭窗口联系到一起。**



1. **游戏逻辑的设计**

**2.1游戏基本场景的配置——iniscene函数的使用**



**我们通过iniscene函数设置了界面的大小，图标，音效标题等基本配置，我们还设置了一个返回按钮。在原有配置的基础上添加了一个清空地图数组的函数，并且通过QPainter绘图事件设计了动画曲线，设置了胜利界面以及失败界面的动画。**

**2.2 随机地图的生成——makemap函数**



**这里提到的随机地图其实并不准确，因为这个函数并没有将地图画出来，我们实际上是通过构建一个二维数组，并在二维数组中添加随机数，生成了随机地图的坐标。**

**在上一个界面中我们传入了一个level数据，我们拿他去规定生成矩阵的大小，我们设置一个随机种子，调用系统时间，进行迭代生成了随机矩阵，她由0，1，2，3，4组成，分别代表背景食物蜘蛛蚂蚁以及终点。**



**但为了防止地图中出现包围圈的问题，在此我们借助了一种并查集算法，通过队列实现了包围圈的查找以及排除，具体过程如下：**

**1. 将所有出度为0的点删除**

**2. 将所有出度为1的点推进队列**

**循环以下操作：**

**1. 取队列头（此点出度一定为1）**

**2. 删除该点和所有该点所连的边**

**3. 删除操作二后出度为0的点**

**4. 将操作二后出度为1的点推进队列**

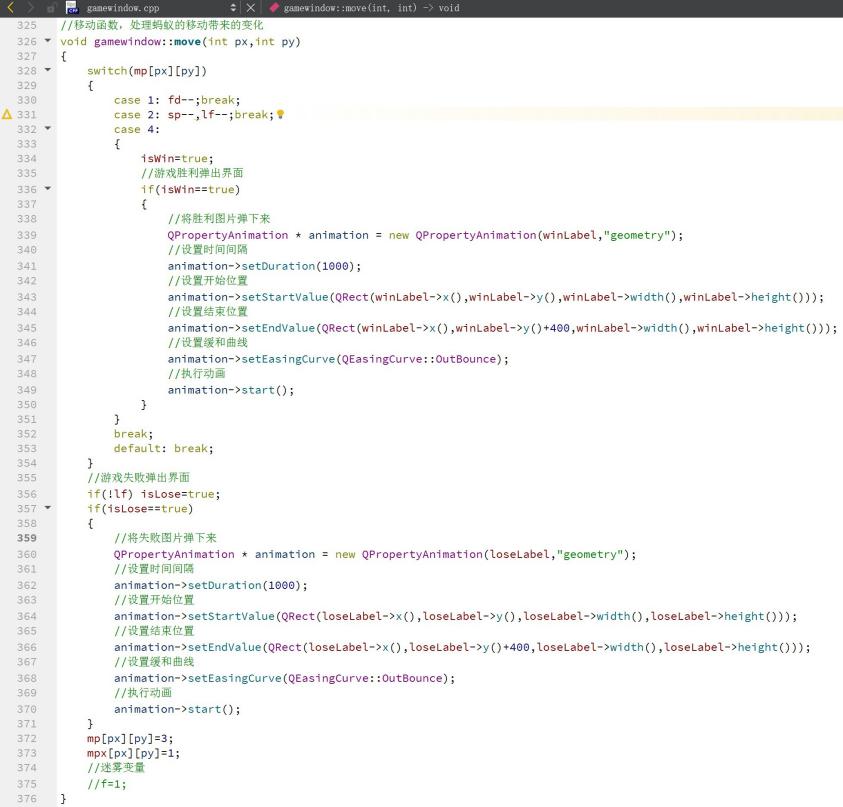
**直到队列为空**

**如果结束循环还有点没被删除，则说明有环。**

**2.3绘制地图——QPainter绘图事件的应用**

**在这个函数中，我们利用了QT自带的QPainter绘图事件按照之前生成的随机地图数组绘制我们的随机地图。但由于我们添加了迷雾功能，为了实现它我们设置了一个迷雾变量f，并为其设置了一个初值0。当游戏初始打开时，f=0，绘制去掉迷雾的全地图，随后当我们按下键盘时，将f的值改为1。在执行这一函数之前，我们用一个if语句判断一下f是否为0。若为0，则绘制全地图，若为1，则绘制带迷雾的地图。**

**2.4处理操作对象移动带来的变化——move函数**



**我们构造了一个move函数，当我们的蚂蚁移动到一个位置时，我们先判断地图矩阵中该位置有没有道具如食物，蜘蛛之类，如果有，我们对这些道具变量进行操作，如果没有我们会更新地图数组。**

**2.5利用键盘操作对象——keypress键盘操作事件**



**这里，我们采用QT中自带的kepress键盘按下事件，通过WSAD来实现其上下左右的移动，并在每次移动后重新更新地图。**

**由于我们最高难度使用了10\*10的大地图，为了提高玩家的游戏体验，我们会适当地在旁边一栏给出提示，即剩余食物数量，剩余蜘蛛数量，离蚂蚁最近的食物和蜘蛛的距离分别是多少。为了实现这样的功能，我们在此使用了一种求最短路径的bfs广度优先搜索算法，并且借助队列，实现了最短路径的计算，**

**2.5.1 bfs广度优先搜索算法简介**

**宽度优先搜索算法（又称广度优先搜索）是最简便的图的搜索算法之一，这一算法也是很多重要的图的算法的原型。其别名又叫BFS，属于一种盲目搜寻法，目的是系统地展开并检查图中的所有节点，以找寻结果。换句话说，它并不考虑结果的可能位置，彻底地搜索整张图，直到找到结果为止。**

**所谓广度，就是一层一层的，向下遍历，层层堵截。**

**1、访问顶点vi ；**

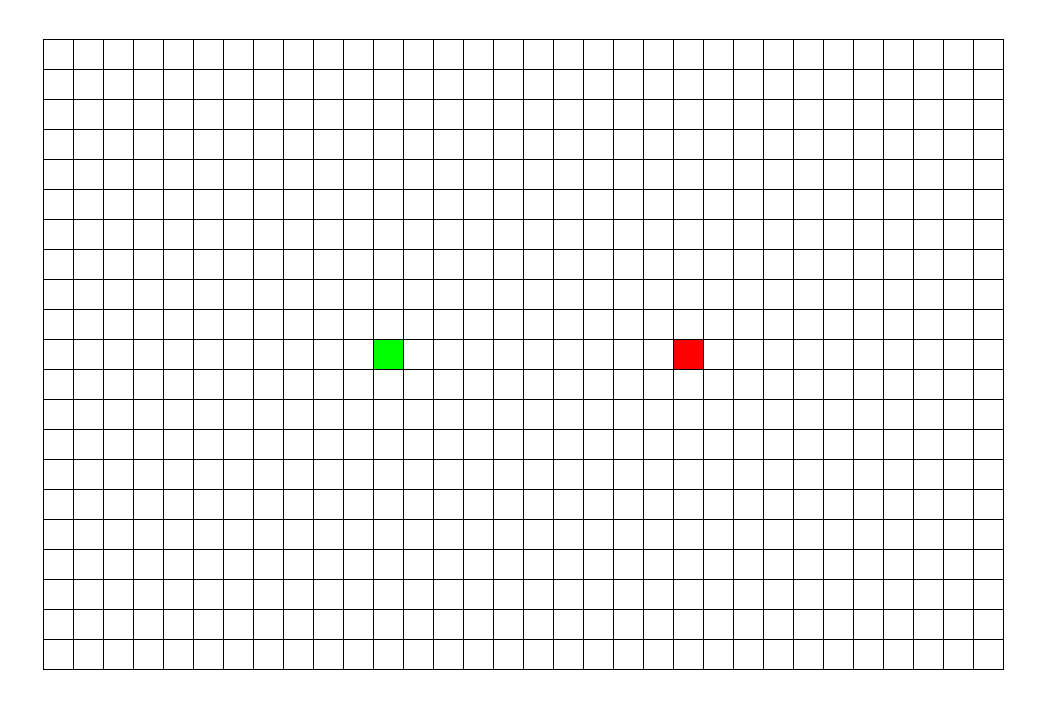
**2、访问vi 的所有未被访问的邻接点w1 ,w2 , …wk ；**

**3、依次从这些邻接点（在步骤②中访问的顶点）出发，访问它们的所有未被访问的邻接点; 依此类推，直到图中所有访问过的顶点的邻接点都被访问；**

**2.5.2 bfs算法的基本思想**

**主要借助一个队列、一个布尔类型数组、邻接矩阵完成（判断一个点是否查看过，用于避免重复到达同一个点，造成死循环等），先将各点以及各点的关系存入邻接矩阵。**

**再从第一个点开始，将一个点存入队列，然后在邻接表中找到他的相邻点，存入队列，每次pop出队列头部并将其打印出来，整个过程有点像往水中投入石子水花散开。**



**2.5.3 bfs算法在本项目中的应用**

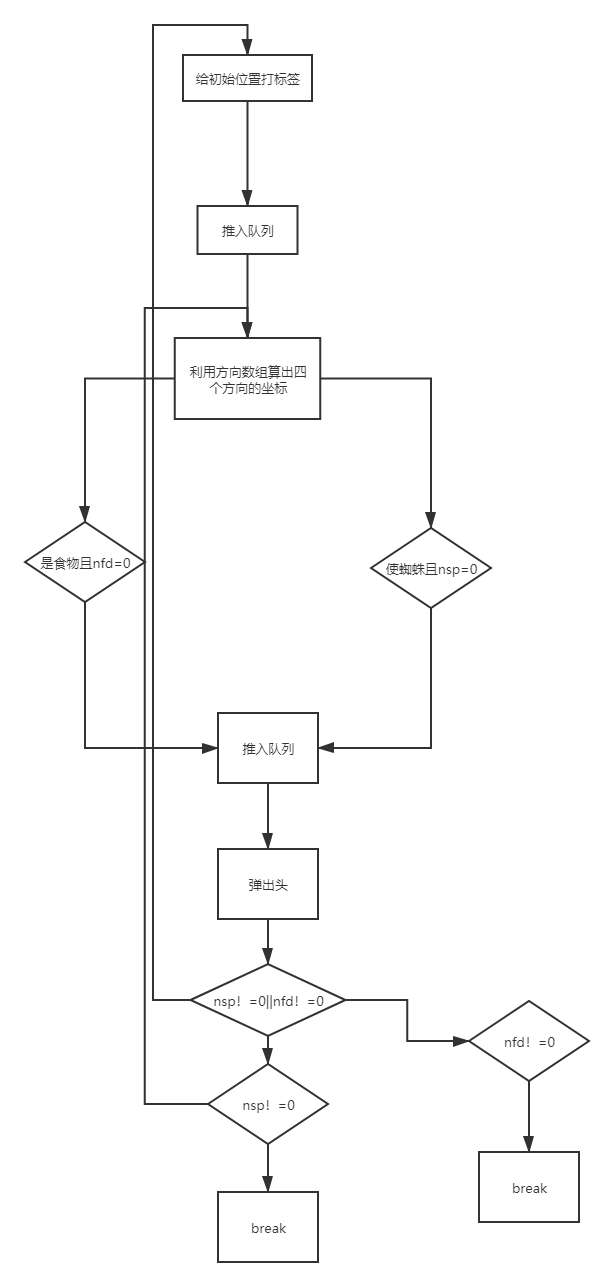


**首先我们定义了一个data结构体，用来储存x坐标，y坐标，以及离对象的距离。**

**接下来定义了一个方向数组dirc用于遍历上下左右的位置。**

**随后构建了一个队列。**

**以及一个bool类型的数组inf用来给已经遍历过的点打标签。**

****

1. **体感部分**

**主要代码类及代码库：**

**Thread类 并发设计**

**Mutex类 保障全局变量 线程安全。**

**Math类 动作计算**

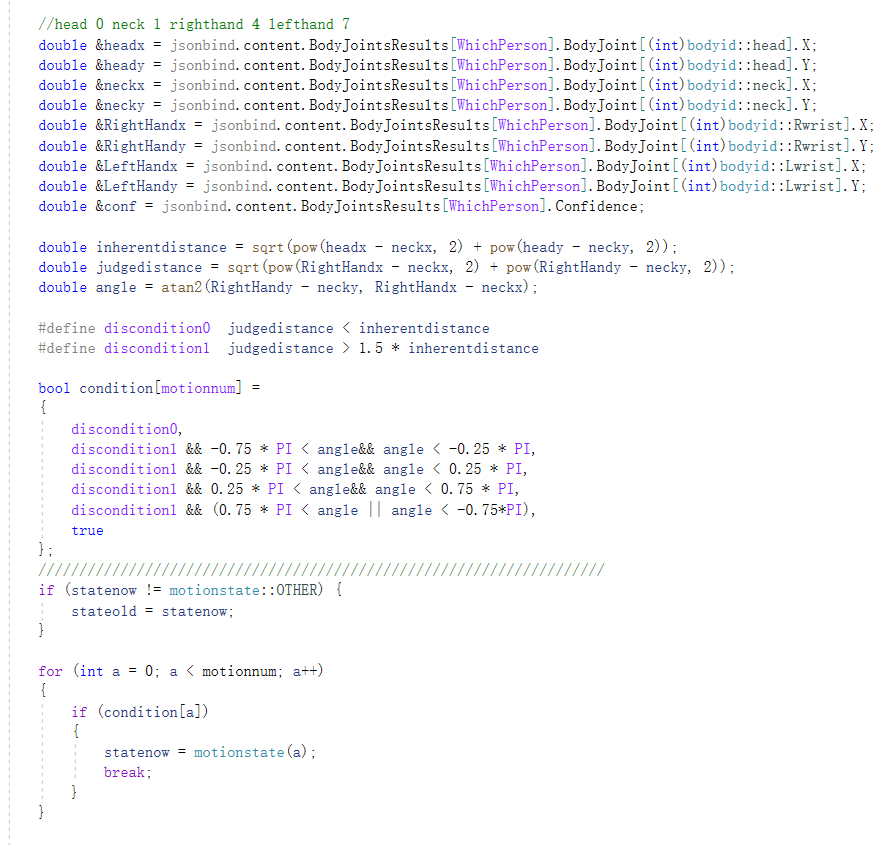
**Opencv库，图像处理。**

**C\_Python Api双语言调用**

**以下抽样两个核心功能展示**

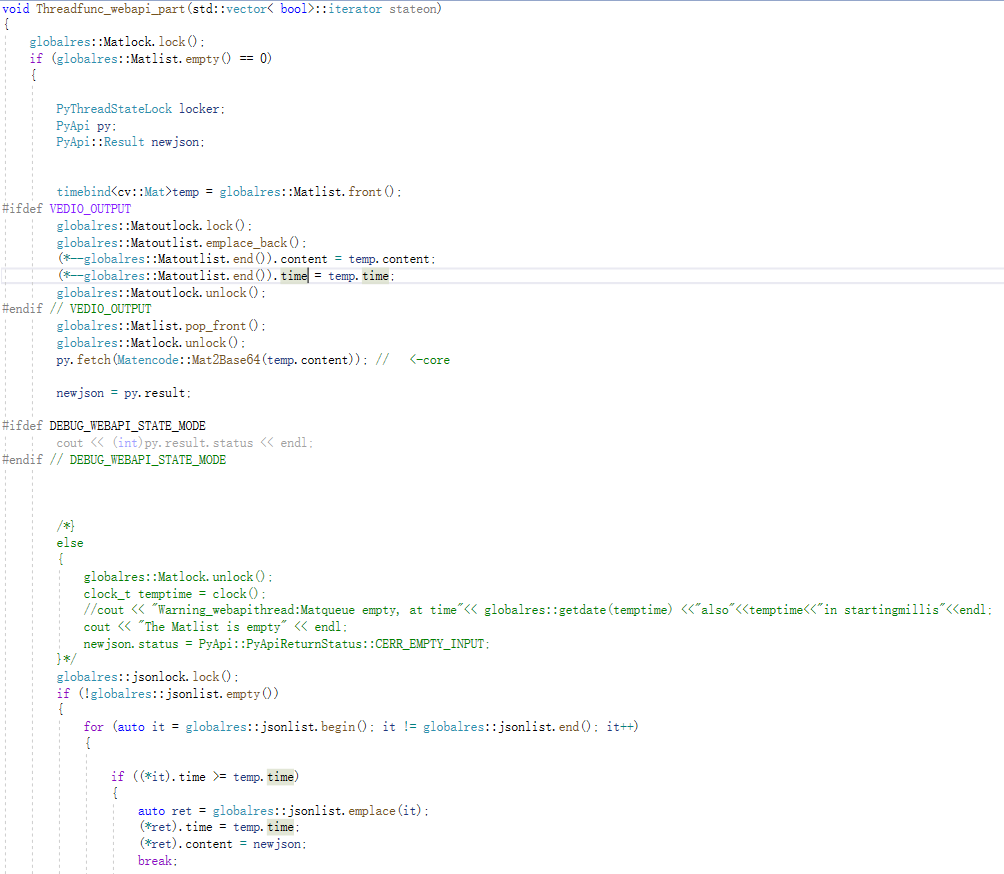
**整个体感模块的设计符合工程开发原则，符合鲁棒性，健壮性，可维护性。做了相当多的优化设计，众多细节在此不再赘述。**

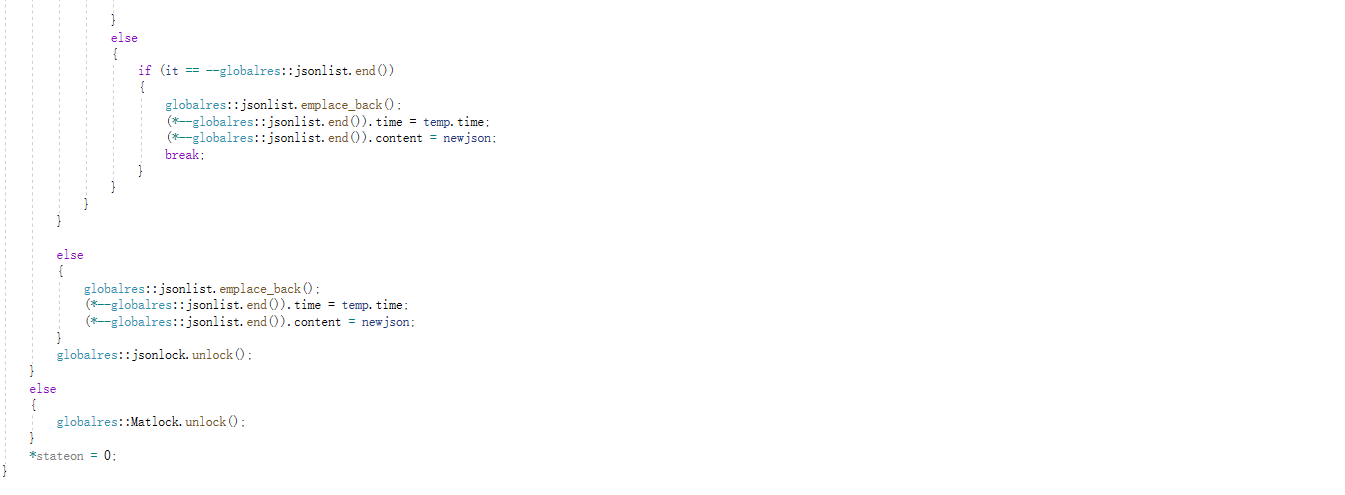
**3.1动作计算和触发：**



**3.2 网络并发线程收发、排序**

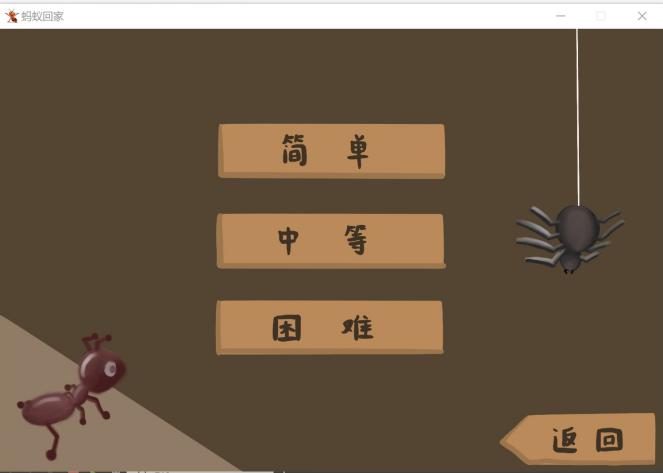
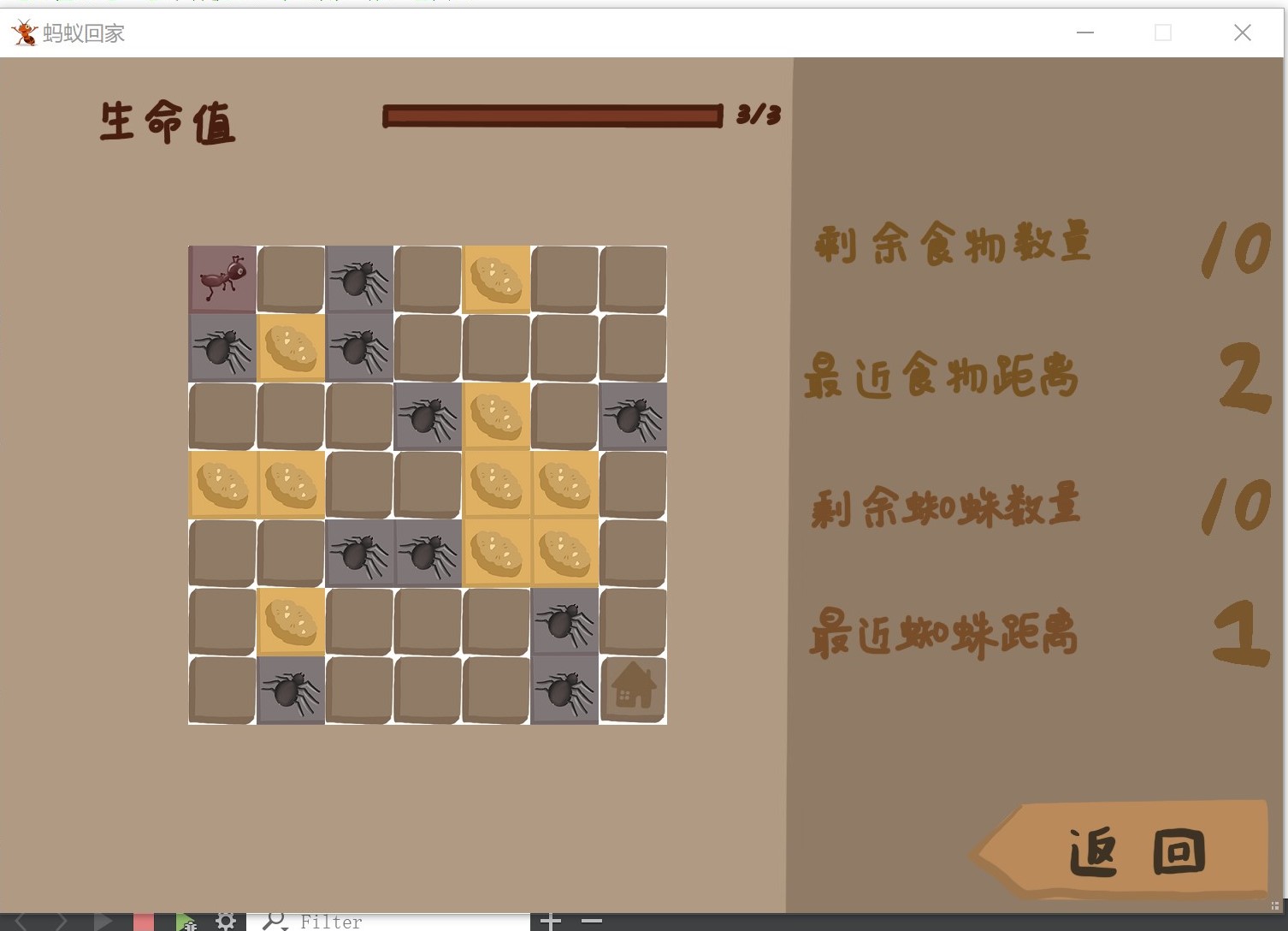


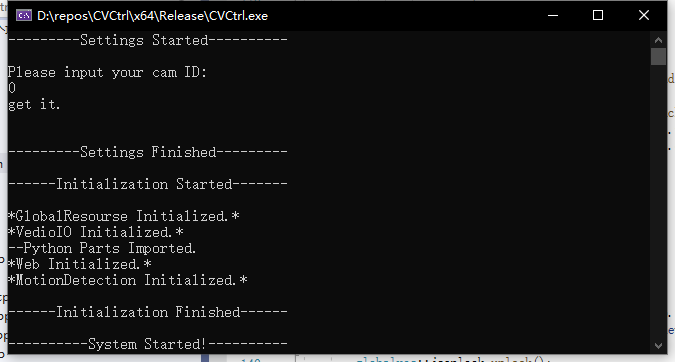




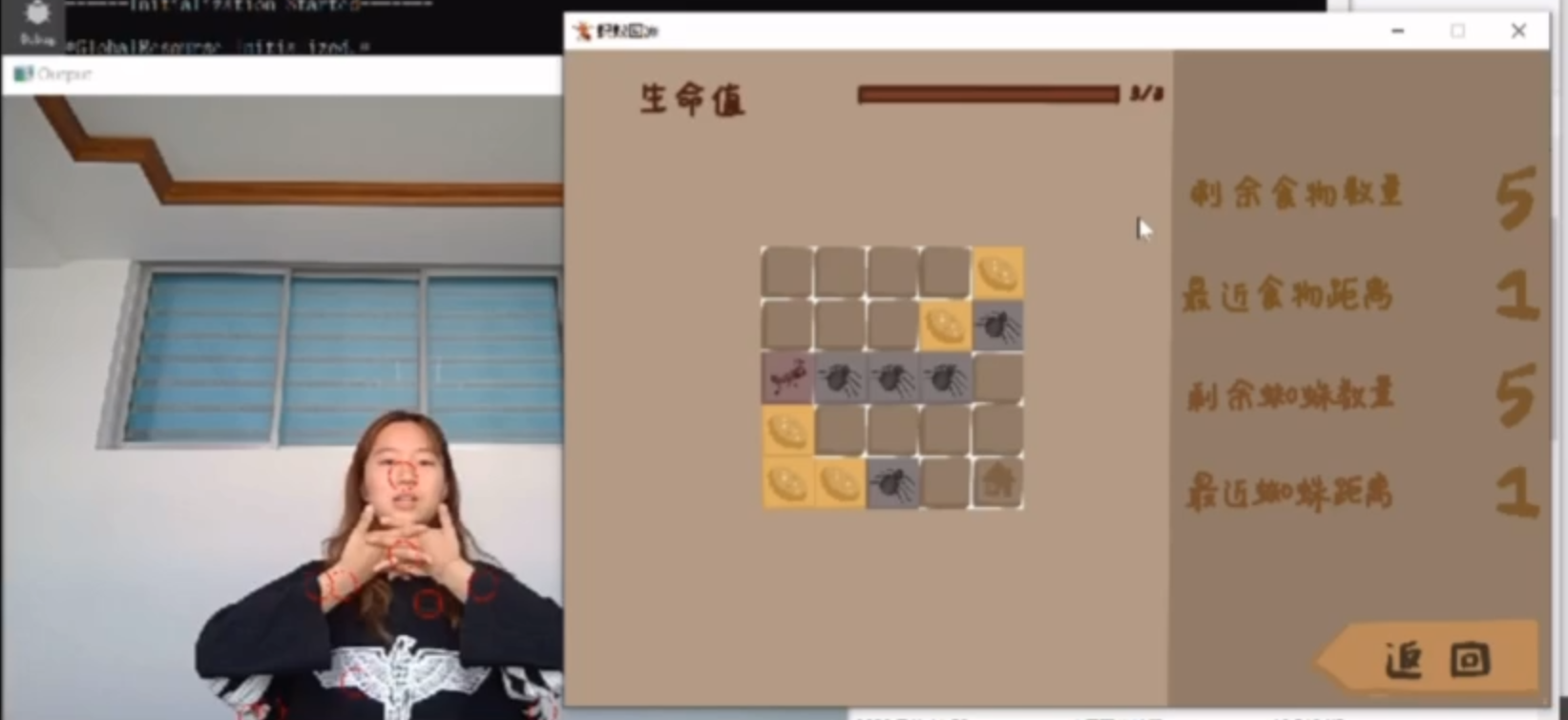
1. **测试（Test）**

**界面展示：**





**体感系统控制台截图**



**体感系统图像处理及输出截图，图中红色圆圈为识别的关键点**

**功能演示：**

