Project1说明

518021910273

刘慎恒

1. 原理简述

这个贪吃蛇project的核心原理是在mainwindow中利用timerEvent类，在m\_timerEvent的时间间隔中的不断刷新snake的位置，以及判断是否吃到了食物，同时刷新计分。

1. 操作说明
2. 普通模式下，键盘方向键操控蛇的运动。
3. 双人模式下，键盘上下左右操控snake1的运动，wasd操控snake2 的运动。
4. Ai模式下，没什么要操作的。

3.各个类的简介

1. snakenode类、wallnode类、sfood1node类、sfood2node类是最底层部分，分别为构成蛇的身体、墙、特殊食物1、特殊食物2的最小模块。在这些类的构造函数中进行了初始化，并直接显示在了屏幕上，同时配有一些setPos()、getPos()等工具函数，每个模块的大小为10\*10.
2. snake类的主体是一个以snakenode为数组元素的list数组body[ ],构成了蛇的身体。同时配有一些对于蛇对象的操作，比如加头，去尾，吃食物等等。
3. food类仅包含一个构造函数。构造函数的作用是生成两个随机数（必须是10的倍数），分别作为foodnode 的横纵坐标生成新的食物并显示在窗口中。
4. wall类的主题是一个以wallnode为数组元素的list数组ww。ww的构造函数随机生成了30个位置不同的墙的方块，用来阻挡蛇的行动。
5. sfood1、sfood2类是特殊食物，生成方式和food类似。
6. mainwindow是最顶层。它画出了蛇可以运动的区域，并且画出了得分和等级区域。在每次刷新的时候，首先检查sf1，sf2的计时器是否启动，如果启动了，检查是否该消失。之后检查是否吃到特殊食物1，特殊食物2，以及普通食物，同时更新分数和等级。
7. 特殊功能的实现
8. 双蛇功能的实现是构建了snake1类，snake2类，再分别在mainwindow中生成相应的对象。两条蛇互不干扰，但是需要一起争夺食物。当其中一条蛇先达到一定分数就获胜。
9. 自动蛇的实现，是判断食物与蛇的相对位置，沿着最短路径去吃食物。
10. 目前还存在的一些问题
11. 当玩家操作得太快，以至于两次连续操作快于刷新帧率的时候，可能会出现直接死亡的情况。
12. 由于墙和食物都是随机生成的，可能出现墙与食物重合的情况。但是出现这种情况时，会优先考虑吃到食物而不是撞到墙死亡。
13. 还有可能食物被墙完全围住，导致食物没法吃到。
14. 墙更新的时候有可能出现更新在蛇的身体上的情况，不过这不会导致蛇的死亡。