

实习报告

程思博 李茂

一、选题及创意介绍

选题为传统的贪吃蛇，灵感来源于小时候在诺基亚玩的贪吃蛇游戏，算是我接触最早游戏之一，可以说是我童年的回忆之一。

主要实现的功能为蛇的自主向前移动，按键控制转向，可视化，食物的随机生成，吃食物时蛇会变长，碰到障碍或蛇身游戏结束。

附加功能（已实现）为可以穿过地图的边界，更大的地图和与之配套的视角随蛇头移动，食物的定时消失，地图选择功能和地图的设计，吃食物和游戏结束的音效，计分并在结束后展示，蛇的加速功能。

计划实现的附加功能（未实现）为食物将要消失的闪动，游戏结束时蛇身的闪动，用按键选择然后中心键确定和开始的标题展示（已实现但因为内存原因放弃），食物消失后变为障碍（已实现但参数没时间调整了）

二、设计方案和硬件连接

主体为两个循环，第一个循环实现开始界面的功能，用于音画展示（画还比较奢侈）和接收玩家的一些参数设置，主要为地图的选择。

然后设置一些参数，进入第二个循环也是主体部分。主要思路是维护一个地图的数据对象，其中包括地图上每个位置的所有信息。每个循环蛇进行前方物体的检测然后向前移动一格，其间通过检测按钮的信息改变蛇的方向或者加速，然后输出以蛇头为中心 5×5 的位置上的画面到显示屏上并停留一段时间。设置并记录一个虚拟的时间，循环中定时生成食物。游戏结束即跳出循环。

没有除 microbit 和数据线外的硬件

三、实现方案及代码分析

开始界面：第一个循环 while True，每个循环中播放字符串，然后 if 条件检测 A 或者 B 按钮是否被按下并赋值参数，然后跳出循环

主要数据对象有字典的 map 和 foods，和列表的 snake。map 中以坐标为键，字符串表示的亮度和类型为值。foods 中以坐标为键，食物生成时刻为值。snake 中为从蛇尾到蛇头的坐标。

两个循环间进行初始化和参数设置。生成长两个位置的蛇记录到 snake 和 map 中

方向：用 0123 分别表示上右下左四个方向，用整数的 direction 记录。每个循环中用 if 语句检测按钮，左转 direction 减 1 余 4，右转类似。

移动：先通过 snake 得到蛇头的坐标，和 direction 一起获得前方一格的坐标。若这一格不在 map 中，将这一格设为蛇头，记到 snake 和 map 上，通过 snake 得到蛇尾的位置，删去 map 中和 snake 中对应的元素；若这一格在 map 中的类型为食物，将这一格设为蛇头，维护 snake 和 map，对蛇尾不做处理；若这一格在 map 中的类型为障碍或蛇尾，跳出循环游戏结束。

可视化：得到蛇头的坐标，简单加减得到视野的上下左右的坐标边界，其中包括蛇头靠近地图边界时的特殊判断。利用坐标边界进行双循环的遍历得到视野中每个坐标的亮度，形成可直接用 display 语句输出的形式然后输出。然后用 sleep 语句停留 gap 时间

虚拟时间：相关数据对象有每个循环的虚拟时间 gap，总时间 game_time，都是整数形式。

每个循环中 game_time 自加 gap

食物的生成：相关数据对象有食物生成的间隔时间 food_gap，食物生成计时 food_count，

都为整数类型。food_count 初始值为 0，每过一个循环检测 food_count 是否等于 0，若等于 0，用双重循环的生成器得到每个空着的地图坐标，然后用 random 模块随机从中选出一个，记录到 foods 和 map 中。然后 food_count 自加 gap，然后检测 food_count 是否大于 food_gap，若大于 food_count 赋值为 1。

食物消失：相关数据对象有整数的食物停留时间 remain_time。每个循环中检测 foods 中每个值+remain_time 是否小于 game_time，若小于删去 foods 和 map 中的对应元素。

蛇的加速：相关数据对象有每个循环的时间 gap，较快的时间 fast_gap，正常的时间 classic_gap，都为整数类型。由于预设包里的语句只能检测一次某按钮是否被按，每个循环中先用 a, b 两个布尔值记录按钮按下的情况。然后判断 A 被按还是 B 被按还是都被按。若两个按钮都被按，gap 赋值为 fast_gap。若某个被按就是转向的处理，且 gap 赋值回 classic_gap。若两按钮都没被按过，进一步判断两按钮是否都正被按（持续按的时间超过一个循环），否则 gap 赋值回 classic_gap，表示加速结束。

设地图行数为 r，地图列数为 c。 $O(r,c)=r*c$ ，主体为生成食物时对空白位置的遍历。但现有地图 r, c 最大都为 10。

四、后续工作展望

要实现闪动的效果就要抛弃每个循环走一步的框架。后续的方案有两个，一个是继续用虚拟时间，只不过每两步间有一个类似食物生成的 gap，同时转向也要防止转 180° 的情况，这时就可以通过相邻循环的可视化展示与否实现闪动，于是 gap 就成为一种类似帧率的东西。另一个方案与之类似，不过用到实际运行时间，好处是可通过对 sleep 的休眠时间进行多退少补减少代码运行导致的画面卡顿。

调整参数完善食物消失变障碍的游戏模式

闪动效果的引入使我们辨认物体种类的方式除了亮度外多了一种，故可引入新的类型星星，吃掉后可将障碍变为食物一段时间或全屏食物一段时间，类似于奖励模式，提供玩家改变障碍的能力。

可探索下 microbit 的数据传输功能，目标是对战模式。

五、小组分工合作

程思博主要负责代码的主体部分，即参数初始化和第二个循环。李茂负责初始界面的第一个循环和一些地图内容的编写（写的部分后来因为爆内存被大幅简化），音效部分，游戏参数的调试。

编代码主体时主要是商量好各自部分代码的封装然后各写各的。后来的调整添功能 debug 阶段，程思博负责代码编写，李茂负责实机调试。提交阶段程思博负责实习报告，李茂负责其它所有。