**Micro:bit作品蛇舞龙腾开发实习报告**

朱信霖、韩克晋

**摘要**：本学期的课外拓展作业是python开源硬件，在micro:bit微型控制板运行代码。经过小组讨论，我们这次实习项目是使用micro:bit开发两个游戏，分别为贪吃蛇枪林游走和小恐龙弹雨远征。

**一、选题及创意介绍**

我们选择了利用micro:bit开发游戏的项目，旨在通过实际编程操作和调试来提高我们的编程能力。微控制器具有小巧、便携、易学易用、低功耗等特点，用micro:bit作为载体，非常有助于利用所学编程知识解决实际问题的能力，而且相较于在电脑上运行更能够体会程序对空间开销的要求。我们小组本着简单有趣同时能训练编程能力的原则进行选题，经过讨论，我们最后选择了开发简单的游戏：贪吃蛇枪林游走和小恐龙弹雨远征，创意源自于经典的贪吃蛇游戏与Chrome断网的小恐龙游戏。一方面这两款游戏的实现相对简单，可以用来检验我们的编程水平，体验编程的乐趣；另一方面，能让我们在进行程序设计的同时初步了解游戏开发的过程。

**二、设计方案和硬件连接**

**1、设计方案**

贪吃蛇枪林游走与传统贪吃蛇游戏玩法相同。在5×5LED灯上随机生成食物，通过按键或体感控制贪吃蛇移动方向，超越边界或碰到蛇身游戏失败，吃到食物长度增加并获得分数。小恐龙弹雨远征中，LED屏幕的最底层为地面，小恐龙位于屏幕左侧，通过按键或体感控制小恐龙跳跃或下蹲，以躲避从右侧出现的障碍物。两个游戏中的体感控制均通过加速度传感器监测micro:bit硬件板姿态实现。

**2、硬件连接**

这段程序需要使用micro:bit硬件，并对其进行如下连接

贪吃蛇枪林游走：

Pin0 - 左按钮，用于控制蛇头方向向左

Pin1 – 右按钮，用于控制蛇头方向向右

Pin2 – 上按钮，用于控制蛇头方向向上

Pin3 – 下按钮，用于控制蛇头方向向下

accelerometer – 加速度传感器，直接控制蛇头方向

小恐龙弹雨远征：

button\_a – 用于跳跃

button\_b – 用于下蹲

accelerometer – 加速度传感器，直接控制小恐龙动作

**三、实现方案及代码分析**

游戏开始，玩家首先通过触摸pin\_logo来改变是否使用加速度传感器来控制贪吃蛇以及小恐龙的运动，默认为不使用，界面显示开心表情，触摸后界面显示惊讶表情，再次触摸可取消使用加速度计，用accel变量表示是否使用加速度计。随后按下button\_a，则进入贪吃蛇枪林游走；按下button\_b，则进入小恐龙弹雨远征。

若玩家选择贪吃蛇游戏，游戏开始时，贪吃蛇位于屏幕正中央，长度为1，蛇头方向向右，贪吃蛇蛇头移动方向可以通过使用有关按键或加速度传感器来控制，在程序中限制了蛇头无法直接反向。在蛇身外随机位置生成食物，食物亮度由3-6随机变化，对应其分值，吃下食物后得分增加，生成新的食物。如果玩家控制的贪吃蛇头部碰到了自己的身体或者屏幕边缘，则游戏结束，最终显示分数。可修改bmp参数改变贪吃蛇的移动速度。

各个函数的分下如下：

1.new\_food函数：用于生成食物的初始位置，避免与蛇身重合。随机生成两个整数x和y表示食物位置坐标，判断它是否与蛇身重合，如果是，则重新生成。最后在显示屏上设置该位置的亮度值为3-6范围内的随机值，并返回食物位置与分值。

2.move\_snake函数：控制蛇的移动和状态更新，根据不同的方向来改变蛇头的位置，判断蛇是否撞墙或者撞到自己，如果蛇撞到了食物，则在蛇尾加入一个新的位置，并重新生成下一个食物。如果蛇没撞到食物，则蛇尾位置改变，并将蛇尾从显示屏上擦除掉。最后返回游戏是否结束、移动后的蛇身、食物位置和当前分数等参数。

3.reset\_game函数：重置游戏状态，包括初始化蛇身、生成第一个食物、设置初始方向、清空屏幕并显示蛇头位置亮度值为9等。当通过按钮控制时，通过pin0.is\_touched()等语句监听按键事件，当按键被触发时更新方向；当使用重力传感器时，通过accelerometer.get\_x()和accelerometer.get\_y()函数获取当前设备的位置，并根据不同方向来改变蛇头移动的方向。每次重置游戏状态时，都会播放音效并显示当前分数，最后清空屏幕。

4.重力传感器控制贪吃蛇运动：该代码通过不断监测accelerometer传感器获取到的x轴和y轴的加速度值，并计算其与初始值之间的差值，从而判断当前的倾斜方向。y轴方向加速度差值乘了修正系数1.2，原因为手持屏幕时，屏幕平面并不完全平行于水平面，而是有一定倾角，各方向转动所对应加速度变化快慢不同，1.2只是一个参考的修正系数。

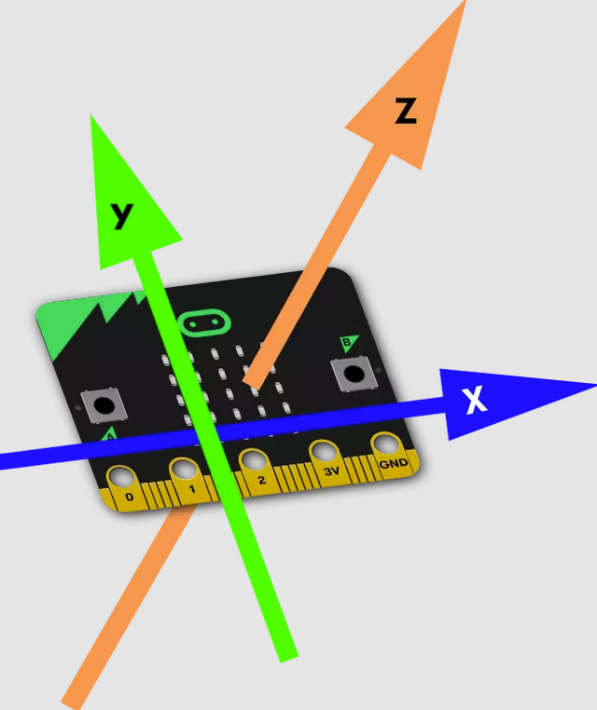


图1 加速度计方向示意图

如果玩家选择小恐龙跳跃游戏，该游戏主要包含两个元素：小恐龙和障碍物。玩家需要通过控制小恐龙跳过障碍物并获得尽可能高的分数。下面是该代码的主要实现方式：

1.通过初始化方法传入 accel参数表示是否启用加速度传感器。

2.casechange(self, case1, case2, stren)函数用于不同状态的图像显示切换,实现亮度为stren的图像case1到图像case2切换。小恐龙动作产生的图形变化与障碍物移动均通过该函数实现。

3.block(self, contin, blockstep, blockcase, dinodic, dinocase)函数用于生成障碍物，并控制障碍物移动与消失，同时检测小恐龙是否与障碍物发生碰撞。共有三种障碍物，包括两种需要跳越躲避的障碍物和一种需要下蹲躲避的障碍物。contin为判断游戏结束的参量，blockstep表示障碍物的帧，blockcase表示障碍物种类，dinocase表示恐龙当前状态。

4.move(self, bmp, dinocase, dinodic, contin, blockstep, blockcase, y0)函数用于控制小恐龙的状态，调用block函数实现动画的逐帧进行。bmp控制游戏速度，障碍物移动速度和小恐龙动作速度随着分数增加逐渐增大。

5.inmain(self)函数是该小游戏的主要逻辑。对游戏进行初始化并控制游戏循环，触发游戏结束的分数结算。

**四、后续工作展望**

在本次实习项目中，我们成功地使用micro:bit开发了两个游戏，积累了编程经验和实践经验。后续我们会进一步优化代码，改进游戏画面和玩法，增加更多的功能和关卡等，以提高游戏品质和可玩性。同时，我们还可以尝试利用其他传感器和模块来扩展游戏的功能，如声音传感器、温湿度传感器等，从而探索更为有趣和实用的微控制器应用场景。

游戏中的体感控制部分实机效果可能较差，可以考虑将加速度计获取的信息先转化为方位角再进行处理，其次还可以完善体感操作的相关逻辑，使体感操作更人性化。

**五、小组分工合作**

在本次实习项目中，我们两人合作完成了项目设计、代码编写、调试等工作。其中，韩克晋同学编写了贪吃蛇游戏代码的最初版本，并负责撰写实习报告，朱信霖同学负责Chrome浏览器小恐龙游戏的代码编写，修改完善贪吃蛇游戏的代码，并将两段代码进行后续的加工以及拼接等工作。在编写完代码之后，我们一起进行硬件连接和整合测试等工作。通过互相协作，充分发挥各自的特长，共同完成了这个游戏开发项目。

**六、总结**

本次实习项目是我们第一次尝试使用micro:bit进行游戏开发，通过实际操作和不断调试，我们逐渐熟悉了micro:bit的硬件连接与编程，掌握了常见的Python编程语法和开发技巧。同时，实习项目也启发我们对游戏开发的思考和理解，深入探索了游戏开发与程序设计之间的关系和联系。值得一提的是，通过本次实习项目，我们还培养了团队合作和沟通能力，提高了自身综合素质和工作效率。