

# microbit 创意作品实习报告

2100011367 任勇钢 2000017727 邓绶豪

摘要：探索童芯派上的 event，cyberpi 库以及 sprite 类；在童芯派上复现经典游戏重力平衡球，并利用 scratch 和 python 实现。

## 一、 选题及创意介绍

我们选用了童芯派这个板子用来进行创意开发，原因是它具有较为高清的屏幕，有大量灵敏的传感器可供选用。而重力平衡球是一款经典的小游戏，玩家通过设备的重力感应来控制小球的移动，躲开洞口等障碍物，滚向终点，此游戏看来容易，亲自操作时极为细节，易使人焦躁。我们决定在童芯派上，利用编程写游戏重现这一经典游戏。

## 二、 设计方案和硬件连接

我们打算重现重力平衡球中几个重要的元素：

墙：球碰到墙之后向反方向弹开

洞：球的重心在洞口时球落入洞中，游戏失败并重新开始

炮：炮会发射炮弹，如果炮和炮弹碰到球，游戏失败并重新开始

硬件连接：只使用童芯派主板，重力传感器是其自带的，通过 Type-C--USB 线与电脑相连接，通过电脑进行编程控制，将代码传输至童芯派上运行。

### 三、 实现方案及代码分析

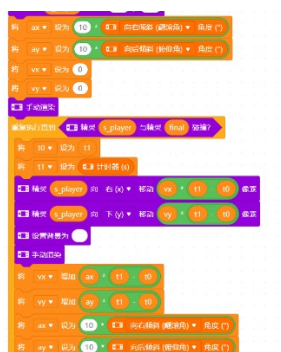
利用童芯派企业开发的软件 mblock，结合 scratch 编程与 Python 语言编程进行书写。

代码分析:

关键部分：



创建终点 `final` 和小球 `s_player`，绘制图案并把他们初始化到合适的位置



初始化运动学参量：这是最重要的一部分。首先在小球运动前初始化沿 x 和 y 方向速度与加速度，随后进入循环，记录每次循环开始的时刻，与前一次循环做差得到时间差，每次循环重新根据重力倾斜角度设定加速度，速度的改变为加速度乘时间间隔，小球的位移为速度乘时间。循环结束条件为小球和重点相碰，此时停止循环。对小球碰撞和死亡重新开始的判断都在循环内部。

```
if s_player.get_x() > 120 or s_player.get_x() > 44 and 58 > s_player.get_x() and 96 > s_player.get_y() or s_player.get_x() > 80 and 96 > s_player.get_x() and s_player.get_y() > 24:
    vx = -0.5 * vx
    s_player.move_x(-1)

if 8 > s_player.get_x() or 40 > s_player.get_x() and 104 > s_player.get_y() or s_player.get_x() > 58 and 72 > s_player.get_x() and 96 > s_player.get_y() or s_player.get_x() > 96 and
    vx = -0.5 * vx
    s_player.move_x(1)

if 8 > s_player.get_y() or s_player.get_y() < 104 and s_player.get_x() < 40 or s_player.get_y() < 96 and 44 < s_player.get_x() and s_player.get_x() < 72:
    vy = -0.5 * vy
    s_player.move_y(1)

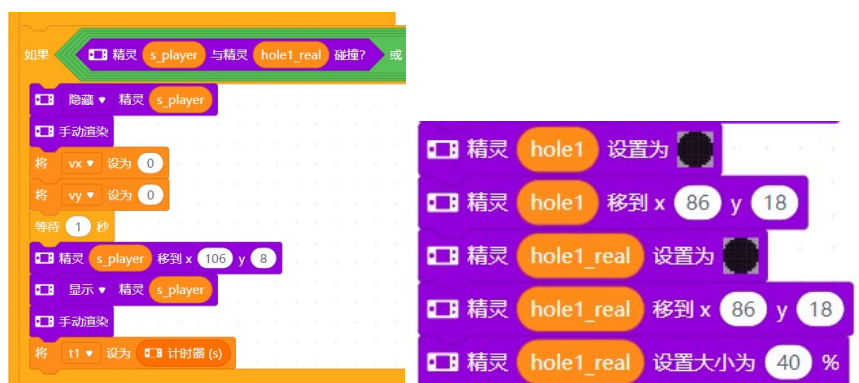
if s_player.get_y() > 120 or s_player.get_y() > 24 and 80 < s_player.get_x() and s_player.get_x() < 112:
    vy = -0.5 * vy
    s_player.move_y(-1)
```

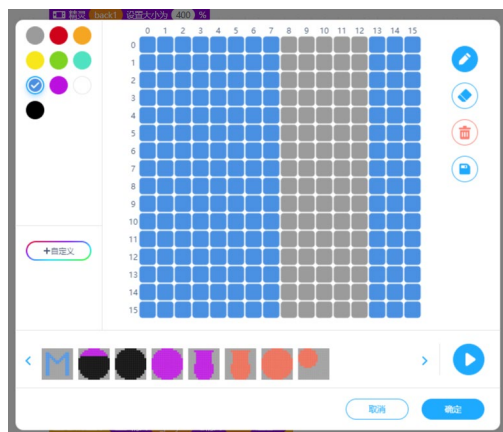
(scratch 转 python)

球与墙（和边界）进行碰撞判断：童芯派的横纵编辑单位为 128\*128，将墙分为四个朝向，写出墙的解析式，如果小球的坐标在墙内（进入墙内）就改变这个方向的速度，速度的恢复系数设为 0.5，沿墙外弹出 1 像素（主要为防止产生 bug 把球卡进墙里）



显示墙：墙通过背景精灵来判断（在图层最底层，不参与与小球运动的判断）童芯派每个精灵的初始大小为 16\*16 像素，适当地将精灵放大，通过计算可以得出墙的解析式和作为墙的背景板精灵像素点的位置之间的关系，在 scratch 上涂鸦并移到相应解析式的位置即可





球与洞中心（炮，炮弹）进行判断：sprite 类中的判断两个精灵碰撞判断的是边界像素点，建立精灵‘真洞’，并调整其大小为原来的 40%（目的是使球心靠近洞时，球才掉下去），在真洞的中心建立精灵‘洞’作为背景精灵，如果球与真洞发生碰撞，则隐藏球，重置球的速度，等待 1 秒后在原始位置上出现。



炮弹的运动：每次循环移动一个像素，当移动到边界时，移到原位置再次开始移动



游戏结束：当球与终点发生碰撞，则游戏结束，在屏幕中心生成带有祝贺的话语意思的精灵，球消失

设计了由简单到困难的三个关卡

第一关只有墙的碰撞

第二关加入了洞的元素

第三关加入了炮弹的元素

注：由于童芯派内存限制，我们不得不将三个关卡独立开来，因此源代码是三个关卡各自的内容。

#### 四、 后续工作展望

首先，我们的代码具有高度可扩展性，可以用于设计更多关卡  
其次，可以在关卡中加入更多元素（例如反弹块——小球碰到之后反向增大速度）  
最后，我们希望向更多的玩家推广该游戏

## 五、 小组分工合作

邓绶豪：建立小球和洞的碰撞，小球和大炮的碰撞以及炮弹的运动，完成第 2，3 关的设计和编程

任勇钢：建立小球运动的物理引擎，建立小球与墙的碰撞以及墙的显示，完成第 1 关的设计和编程，完成视频和报告的编辑。

共同完成：研究童芯派的基本控制语言，学习文档，研究有关 sprite 库的语法使用