

《八年级第五单元第 22 课-指尖平衡解压游戏》学生学习手册

学校名称：_____ 班级：_____ 姓名：_____

模块一：课程导入

真实情境

你有没有过这样的体验：上课时间长了，或者写作业久了，感觉头昏脑胀，注意力很难集中？这时候，如果能做一个快速、有趣的小游戏来放松一下大脑，重新找回专注力，是不是很棒？

这个小游戏就像一个“专注力挑战”：你需要通过精确地倾斜主控板，来控制屏幕上的一个小光点，去“吃掉”随机出现的“奖励豆”。这不仅考验你的反应速度，更能训练你的手眼协调能力和专注力。让我们一起用编程，为自己打造一个课间休息的“神器”吧！

核心问题

我们能否利用主控板的加速度传感器，制作一个“指尖平衡”游戏，通过控制光点移动来锻炼我们的专注力？

学习目标

通过本课程的学习，你将能够：

1. 理解加速度传感器的工作原理，并用它来控制角色移动。
2. 学会使用“随机数”积木块，在屏幕的任意位置生成物体。
3. 掌握“碰撞检测”的编程逻辑，并实现得分功能。
4. 学会使用变量来存储和更新游戏分数。

5. 设计并实现一个完整的、可玩的“专注力训练”小游戏。

模块二：概念学习

1. 核心元件：加速度传感器

想一想：你的手机屏幕为什么能自动旋转？当你把手机横过来时，屏幕内容也会跟着横过来。这是怎么做到的呢？

答案就是**加速度传感器**。它就像主控板里的一个“小弹珠”，能感受到主控板在不同方向上的晃动和倾斜。它主要测量三个方向的加速度：

- **X 轴**: 左右方向
- **Y 轴**: 前后方向
- **Z 轴**: 上下方向

在我们的游戏中，我们主要使用 X 轴和 Y 轴。通过读取这两个轴的数值，我们就能知道主控板的倾斜方向和幅度，从而控制光点的移动方向和速度。

2. 核心逻辑：碰撞检测

想一想：我们怎么判断光点是否“吃掉”了奖励豆呢？

在编程世界里，“吃掉”这个动作，其实就是判断两个物体的坐标是否完全重合。这个过程就叫做**碰撞检测**。

逻辑非常简单：

如果 光点的 X 坐标 等于 奖励豆的 X 坐标，**并且** 光点的 Y 坐标 等于 奖励豆的 Y 坐标，**那么**，我们就认为它们发生了碰撞。

模块三：逻辑分析

1. 系统功能拆解

一个完整的系统可以分为输入、处理、输出三个部分。让我们一起来分析一下我们的“指尖平衡”游戏吧！

系统部分	具体功能	实现方式（图形化积木块）
输入	感知主控板的倾斜	使用“读取加速度(X/Y)轴”积木块
	1. 控制光点移动	1. 将加速度传感器的值赋给角色的坐标
	2. 生成随机奖励豆	2. 使用“在...到...之间取随机数”块设置坐标
	3. 判断是否碰撞 4. 更新分数	3. 使用“如果...那么...”和“等于”积木块进行判断 4. 创建“分数”变量，碰撞时“将分数增加 1”
处理	1. 在屏幕上显示光点	1. 使用“绘制圆形”或“显示角色”积木块
	2. 显示当前得分	2. 使用“显示字符串”积木块，将“分数”变量的值显示出来
	和奖励豆	
输出		

2. 逻辑可视化：碰撞检测真值表

填一填：请根据每一行的情况，判断是否发生碰撞。

条件 1:		条件 2:		最终结果:
$Gx == Dx$ $Gy == Dy$		两个条件是否“同时”满足？		是否碰撞？
是	是	是	是	是
是	否	否	否	否
否	是	否	否	否
否	否	否	否	否

模块四：算法设计

让我们用流程图来梳理整个游戏的运行逻辑。这就像是盖房子之前的设计图纸。

读一读，理一理：下面的流程图描述了游戏的核心逻辑，请重点理解程序是如何实现光点移动、随机生成和碰撞检测的。

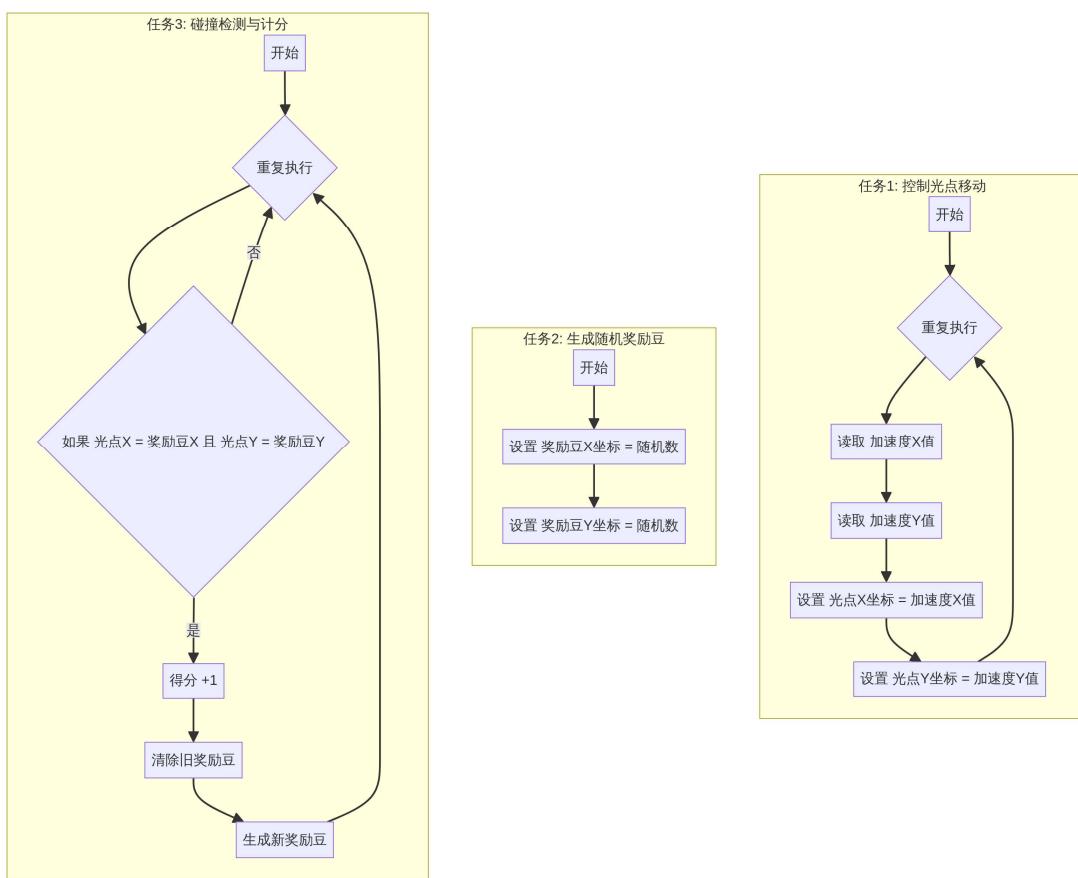


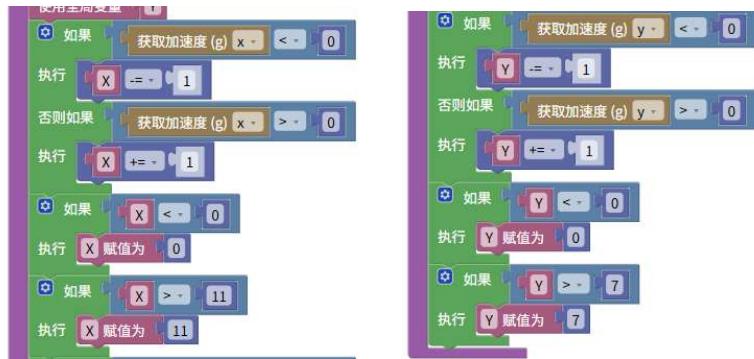
图1：游戏核心逻辑流程图

模块五：搭建指导

我们将通过三个核心任务，一步步构建出功能强大的“指尖平衡”游戏。

任务 1：让光点动起来

目标：通过倾斜主控板，控制光点在屏幕上移动。

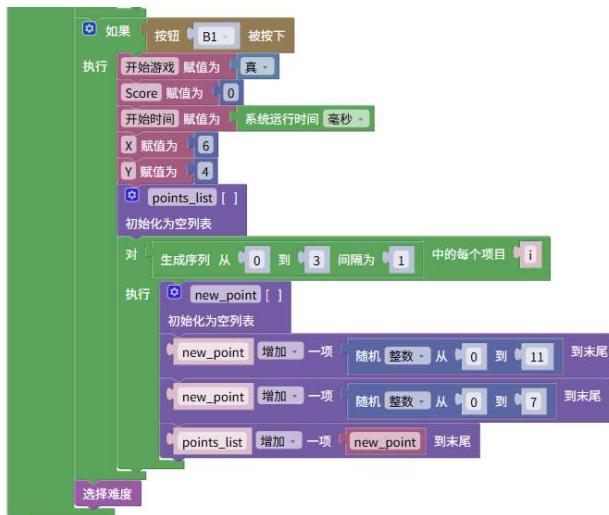


动手试一试：

1. 在一个“重复执行”积木块内部。
2. 拖入“设置角色 X 坐标”积木块，将“读取加速度 X 轴”的值连接上去。
3. 拖入“设置角色 Y 坐标”积木块，将“读取加速度 Y 轴”的值连接上去。
4. 运行程序，观察光点是否能跟随主控板的倾斜而移动。

任务 2：生成随机的奖励豆

目标：在屏幕的随机位置生成一个奖励豆。

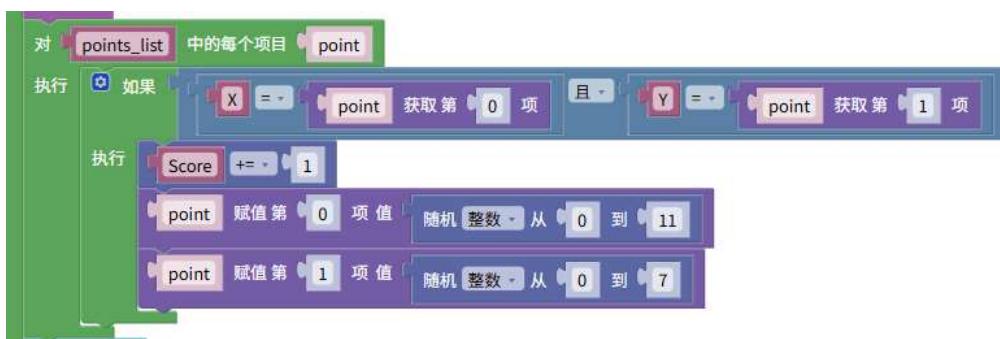


动手试一试：

- 在程序的“初始化”或“当启动时”部分。
- 拖入“设置角色 X 坐标”积木块，在里面连接一个“在...到...之间取随机数”积木块。
- 对 Y 坐标也做同样的操作。

任务 3：实现碰撞检测和计分

目标：当光点碰到奖励豆时，分数加 1，并生成一个新的奖励豆。



动手试一试：

- 在“重复执行”积木块内部。
- 拖入一个“如果...那么...”积木块。

3. 在“如果”的条件框里，放入一个“逻辑与”积木块。
4. “逻辑与”的左边，放入“光点 X 坐标 = 奖励豆 X 坐标”的判断。
5. “逻辑与”的右边，放入“光点 Y 坐标 = 奖励豆 Y 坐标”的判断。
6. 在“那么”的区域里，依次放入： - “将分数增加 1”积木块。 - “隐藏当前奖励豆”积木块。 - 重新生成随机坐标并“显示新奖励豆”的积木块组合。

模块六：调试指导

程序出错了？别担心，这是每个程序员的必经之路！学会调试，是比学会写代码更重要的能力。

技巧 1：“分段测试”与“最小化原则”

千万不要一次性写完所有积木块再运行！

1. 先测移动：只写任务 1 的代码，确保光点能动。
2. 再测生成：加入任务 2 的代码，看奖励豆是否随机出现。
3. 最后测碰撞：加入任务 3 的代码，看是否能正常计分。

技巧 2：利用好电脑端的“眼睛”——打印模块

在代码的关键位置加入 打印 模块，可以帮你洞察程序的每一步动态。

技巧 3：常见问题排查清单

错误现象	可能原因	解决方法
光点不动或乱动	传感器读取错误 误	1. 检查传感器是否正常连接 2. 单独测试传感器读数

奖励豆不出	随机数范围设	检查随机数范围是否在屏幕内
现	置错误	
无法计分	碰撞检测逻辑 错误	1. 使用打印模块确认光点和奖励豆的坐标 2. 检查“逻辑与”是否正确使用

模块七：拓展挑战

挑战 1：增加游戏难度

- **限时奖励豆：**让奖励豆显示的时间缩短，是不是更难“吃”到了？
- **限时挑战：**增加一个倒计时功能，看看在 30 秒内最多能得多少分？

挑战 2：升级视觉和听觉效果

- **庆祝动画：**每次“吃”到奖励豆时，播放一个短暂的庆祝动画或让屏幕闪烁一下。
- **背景音乐和音效：**为游戏添加背景音乐，并在吃到豆子时播放一个“叮”的音效。

模块八：学习反思

1. 我的知识清单

请用自己的话，简要描述你对以下概念的理解：

- **加速度传感器：** _____
- **碰撞检测：** _____
- **变量：** _____

2. 我的“拦路虎”日志

我遇到的问题 我是如何解决的（或仍未解决） 我从中学到了什么

3. 我的创意思考

除了“指尖平衡”，你还能想到用加速度传感器来制作哪些其他有趣的游戏或应用吗？请画出你的创意草图或写下你的想法。

恭喜你，完成了一款属于自己的专注力训练游戏！继续探索，用编程创造更多乐趣！