
Gravity:

Arduino 编程积木

互动媒体教程



DFROBOT
DRIVE THE FUTURE

目录

目录.....	I
揭秘互动媒体.....	6
互动媒体.....	6
互动媒体技术.....	6
初识 Arduino.....	7
Arduino 是什么?	7
Arduino 的诞生.....	8
Arduino 名称由来.....	8
认识 Arduino UNO	9
IO 传感器扩展板 V7.1	10
Mind+编程平台.....	10
Mind+平台介绍	10
Mind+下载与安装.....	11
下载 Mind+编程软件	11
安装驱动.....	11
Mind+界面介绍	12
Mind+程序上传	14
第一章 星光剧场.....	17
第一节 小麦变身.....	18
任务分析.....	18

知识要点	18
动手实践	19
探究思考	24
第二节 空空走钢丝	24
任务分析	24
知识要点	25
动手实践	25
探究思考	30
第三节 点点踩球	30
任务分析	31
知识要点	31
动手实践	31
探究思考	34
第二章 魔法小屋	35
第一节 彩虹琴	35
任务分析	35
知识要点	35
硬件连接	36
动手实践	38
探究思考	40
第二节 吹动风车	40
任务分析	41

知识要点	41
硬件连接	41
动手实践	42
探究思考	45
第三节 点亮星灯	45
任务分析	45
知识要点	46
硬件连接	46
动手实践	47
探究思考	50
第三章 造梦空间	50
第一节 防盗报警器	50
任务分析	50
知识要点	51
硬件连接	52
动手实践	53
探究思考	56
第二节 调光台灯	56
任务分析	56
知识要点	57
硬件连接	58
动手实践	59

探究思考	62
第三节 智能门铃	62
任务分析	62
知识要点	62
硬件连接	63
动手实践	64
探究思考	66
第四节 感应走廊灯	66
任务分析	66
知识要点	67
硬件连接	67
动手实践	68
探究思考	71
第四章 奇想部落	72
第一节 接香蕉 1.0	72
任务分析	72
知识要点	72
硬件连接	73
动手实践	74
探究思考	78
第二节 接香蕉 2.0	78
任务分析	78

知识要点	78
硬件连接	79
动手实践	80
探究思考	84
第三节 吹气球	84
任务分析	84
知识要点	84
硬件连接	85
动手实践	86
探究思考	89

*点击页面左上角 DF 创客社区，有意外惊喜哦

揭秘互动媒体

互动媒体

你有没有看过这样一个视频？一个公益广告牌安静伫立在人来人往的车站，远远看去它只是一个普通的海报，只是中间有一条黑线。但当人们将信用卡从黑线的顶端插入缓缓的往下移动，2 欧元随即从信用卡中扣除，与此同时，厚实的面包片被切下取走（如图 A-1）或者捆绑着双手被释放（如图 A-2），原来你轻轻一刷的 2 欧元代表了给第三世界的孩子食粮与自由。



图 A-1 “切” 面包



图 A-2 “切” 绳索

原本简单的捐款行为在这种方式下变得意义非凡，这就是**互动媒体**的魅力。

互动媒体（Interactive Media）又称互动多媒体、互动式多媒体。它是在传统媒体的基础上加入了互动功能，通过交互行为并以多种感官来呈现信息的一种崭新的媒体形式。¹

互动媒体技术

互动媒体技术也称多媒体互动技术，其关注的是计算机和外界环境的信息互动。科技馆展厅中常见的电子鱼缸、互动投影（如图 A-3）、虚拟翻书和 4D 影院（如图 A-4）等科普项目，正是典型的互动媒体作品，体现了科技和艺术相结合的神奇魅力。²

¹ <https://baike.baidu.com/item/互动媒体/4019538?fr=aladdin>

² 谢作如. S4A 和互动媒体技术[M]. 清华大学出版社, 2014.



图 A-3 互动投影



图 A-4 4D 影院

一个典型的互动媒体作品的运行，一般分为动作采集、智能处理、显示输出三个关键环节，如下图所示。

3

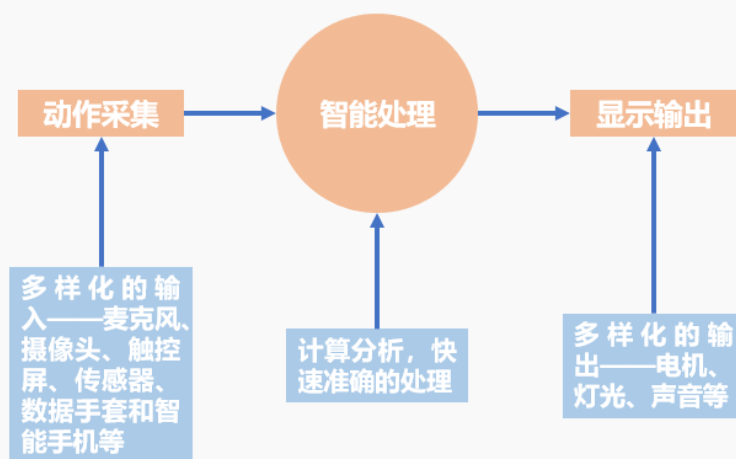


图 A-5 互动媒体作品运行流程

初识 Arduino

Arduino 是什么?

Arduino 是一个开放源码电子原型平台，拥有灵活、易用的硬件和软件。Arduino 专为设计师，工艺美术人员，业余爱好者，以及对开发互动装置或互动式开发环境感兴趣的人而设的。

Arduino 可以接收来自各种传感器的输入信号从而检测出运行环境，并通过控制光源，电机以及其他驱动器来影响其周围环境。板上的微控制器编程使用 Arduino 编程语言（基于 Wiring）和 Arduino 开发环境（以

³ 谢作如. S4A 和互动媒体技术[M]. 清华大学出版社, 2014.

Processing 为基础)。Arduino 可以独立运行,也可以与计算机上运行的软件(例如,Flash, Processing, MaxMSP)进行通信。Arduino 开发 IDE 接口基于开放源代码,可以让您免费下载使用开发出更多令人惊艳的互动作品。

Arduino 是人们连接各种任务的粘合剂。要给 Arduino 下一个最准确的定义,最好用一些实例来描述。

- ◆ 您想当咖啡煮好时,咖啡壶就发出“吱吱”声提醒您吗?
- ◆ 您想当邮箱有新邮件时,电话会发出警报通知您吗?
- ◆ 想要一件闪闪发光的绒毛玩具吗?
- ◆ 想要一款具备语音和酒水配送功能的 X 教授蒸汽朋克风格轮椅吗?
- ◆ 想要一套按下快捷键就可以进行实验测试蜂音器吗?
- ◆ 想为您的儿子自制一个《银河战士》手臂炮吗?
- ◆ 想自制一个心率监测器,将每次骑脚踏车的记录存进存储卡吗?
- ◆ 想过自制一个能在地面上绘图,能在雪中驰骋的机器人吗?

Arduino 都可以为您实现

Arduino 的诞生

这个最经典的开源硬件项目,诞生于意大利的一间设计学校。Arduino 的核心开发团队成员包括: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis 和 Nicholas Zambetti。

据说 Massimo Banzi 的学生们经常抱怨找不到便宜好用的微控制器,2005 年冬天,Massimo Banzi 跟朋友 David Cuartielles 讨论了这个问题,David Cuartielles 是一个西班牙籍晶片工程师,当时在这所学校做访问学者。两人决定设计自己的电路板,并引入了 Banzi 的学生 David Mellis 为电路板设计编程语言。两天以后,David Mellis 就写出了程式码。又过了三天,电路板就完工了。这块电路板被命名为 Arduino。几乎任何人,即使不懂电脑编程,也能用 Arduino 做出很酷的东西,比如对感测器作出回应,闪烁灯光,还能控制马达。

Arduino 名称由来

意大利北部一个如诗如画的小镇「Ivrea」,横跨过蓝绿色 Dora Baltea 河,它最著名的事迹是关于一位受压

迫的国王。公元 1002 年，国王 Arduin 成为国家的统治者，不幸的是两年后即被德国亨利二世国王给废掉了。今日，在这位无法成为新国王的出生地，cobblestone 街上有家叫「di Re Arduino」的酒吧纪念了这位国王。Massimo Banzi 经常光临这家酒吧，而他将这个电子产品计划命名为 Arduino 以纪念这个地方。

认识 Arduino UNO

Gravity: Arduino 编程积木套件中使用的 Arduino 主板是 Arduino UNO，如下图 B-1 所示。

图中有标识的部分为常用部分。图中标出的数字口和模拟口，即为常说的 I/O。数字口有 0~13，模拟口有 0~5。除了最重要的 I/O 口外，还有电源部分。UNO 可以通过两种方式供电方式，一种通过 USB 供电，另一种是通过外接 6~12V 的 DC 电源。除此之外，还有 4 个 LED 灯和复位按键，4 个 LED 中 ON 是电源指示灯，通电就会亮了。L 是接在数字口 13 上的一个 LED。TX、RX 是串口通讯指示灯，比如我们在上传程序的过程中，这两个灯就会不停闪烁。

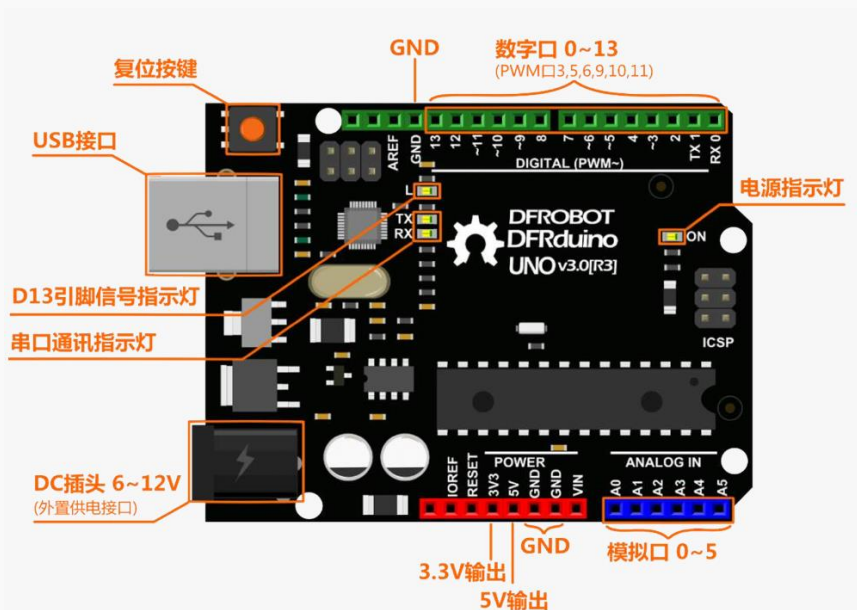


图 B-1 Arduino UNO

Arduino-智能处理设备

智能处理设备就是 Arduino 的控制器。我们这里选用的是 Arduino UNO。控制器好比人的大脑的作用，用来处理事情。

小贴士

IO 传感器扩展板 V7.1

在 Gravity: Arduino 编程积木套件中提供了一个传感器扩展板，用来连接传感器，IO 传感器扩展板功能图如下。

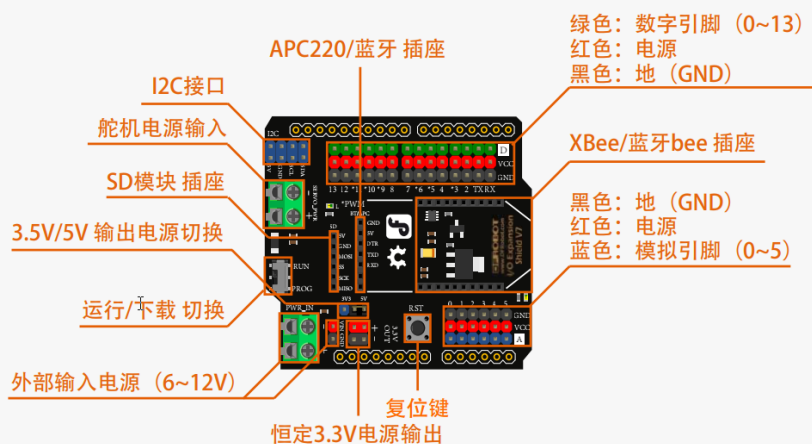


图 B-2 IO 传感器扩展板功能图

传感器-动作采集设备

小贴士

动作采集设备包括传感器。传感器是一种物理装置或生物器官，能够探测、感受外界的信号、物理条件（如光、热、湿度）或化学组成（如烟雾），并将探知的信息传递给其他装置或器官。传感器的作用是将一种能量转换成另一种能量形式，所以不少学者也用“换能器 - Transducer”来称谓“传感器 - Sensor”。

IO 扩展板的最大的好处之一，就是相对于控制板上的有限的几个电源接口，扩展板大大增加了电源口和 GND 接口，不用担心如果连接多个传感器时，会出现电源接口不够用的情况。

Mind+编程平台

Mind+平台介绍

模块化编程软件有不少，硬件也多的让人感觉迷茫。可是有些图形化编程只能在电脑屏幕上操作，或者只能支持某些特定的硬件和机器人。如果有一款软件不仅能图形化编程，还能兼容各种硬件，那就节省了不少的麻烦啦！

Mind+就是这样的软件!

Mind+不仅仅适用于中小学生的学习,还可以为想提高自身技能的“创客”提供 python/c/c++等高级编程语言的学习环境。它支持 Arduino、 micro: bit 等各种开源硬件,只需要拖动图形化程序块即可完成编程,让大家轻松体验创造的乐趣。

如果你什么都不懂,希望能一步一步跟着学,可以用 Mind+;

如果你想学习代码式编程可以用 Mind+;如果你觉得上传程序太麻烦,想实时执行程序,可以用 Mind+;

如果你想要物联网,可以用 Mind+;如果你想找社区,还可以 Mind+!

Mind+ 下载与安装

下载 Mind+ 编程软件

打开网页下载: <http://mindplus.cc>



图 C-1 Mind+ 下载界面

安装驱动

下载成功之后点击“学习”菜单下的“视频教程”按钮打开教程。



图 C-2 打开入门教程

点击“**安装驱动**”教程视频，打开课程视频悬浮窗，根据课程视频一步一步完成驱动安装，**只需安装一次**，

再次打开 Mind+无需安装。



图 C-3 打开课程悬浮窗

Mind+界面介绍



双击 **Mind+** 图标，启动 Mind+ 软件，一起来看看神奇的 Mind+ 的真面目吧！

在 Mind+ 软件界面上，我们可以看到不同的分区，主要由菜单栏、指令区、脚本区、舞台区、角色区组成，

如图 C-4。

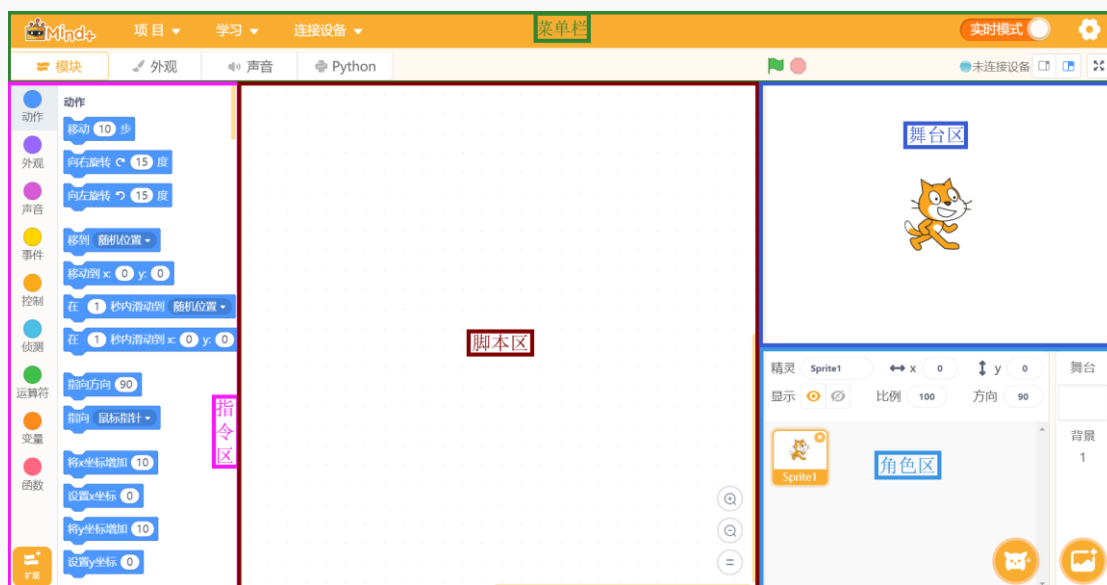


图 C-4 Mind+ 软件界面

菜单栏: 用来设置软件的区域，这里就是“舞台”的幕后啦，没有菜单栏的帮助，连上台表演的机会都没有。“舞台”的幕后有什么呢？

“项目”菜单 可以新建项目、打开项目、保存项目。

“学习”菜单 可以找到想要的教程和示例程序，后期学习过程中还可以通过官方论坛寻求帮助，或者分享自己的作品。

“连接设备”菜单 检测到连接的设备，并且可以选择连接或是断开设备。

“实时模式/上传模式”按钮 切换程序执行的模式，“实时模式”是将脚本区可执行的程序在硬件和 Mind+ 舞台中实时执行，“上传模式”是将程序上传到硬件设备中执行。本教材都是在“实时模式”下操作的。

“设置”按钮 用于设置软件主题、语言、学习基本案例，在线或加群寻求帮助。

指令区: 这里是“舞台”的“道具”区，为了完成各种眼花缭乱的动作，需要很多不同的道具组合。在“扩展”里，可以选择更多额外的道具，如“音乐”、“画笔”等，支持各种硬件编程。

脚本区: 这里就是“舞台表演”的核心啦，所有的“表演”都会按照“脚本区”的指令行动，这里是大家都能看得懂的图形化编程。拖拽指令区的指令就能在此编写程序。

角色区: 在这里你可以选择自己需要的角色/背景，软件中有一个角色库/背景库，你可以在库里选择，也

可以发挥自己动笔上色，绘制新的角色/背景。

舞台区：这里就是角色们“表演”的地方啦，所有的“表演”都是按照“脚本区”的指令行动的。

Mind+程序上传

1、开始之前，请确认有一台运行 Windows 操作系统的电脑和如下物品，如图 C-5 和 C-6:

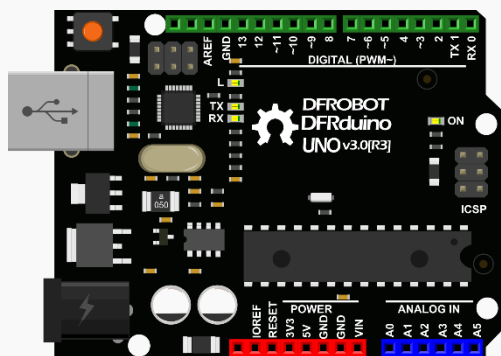


图 C-5 DFRduino UNO R3 (配套方口 USB 线)

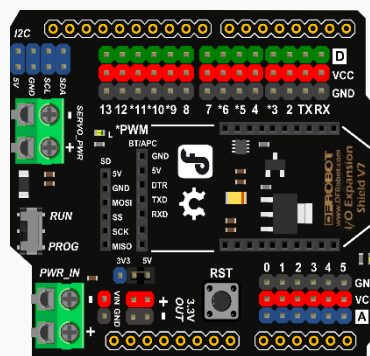


图 C-6 IO 传感器扩展板

将 DFRduino UNO R3 和 IO 传感器扩展板按图 C-7 方式组合，使用方口 USB 线连接电脑。

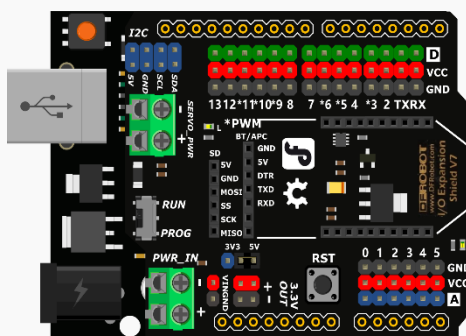


图 C-7 将 UNO 板与扩展板组合

2、点击菜单栏“项目”菜单——新建项目



图 C-8 新建项目

3、选择界面左下角“扩展”——“主控板”——“Arduino Uno”



图 C-9 选择“扩展”



图 C-10 添加“Arduino Uno”

4、将 UNO 板连接计算机，点击菜单栏上“连接设备”，找到并且点击串口号“COM-Uno”，连接成功后，菜单栏显示“连接设备成功”。



图 C-11 选择串口



图 C-12 连接设备成功

5、编写程序——小猫说话，点亮 UNO 板。

选择触发事件，在指令区选择“事件”——“当绿旗被点击”鼠标左键拖拽至脚本区。



图 C-13 触发事件

在指令区选择“外观”——“说 hello! 2 秒”，鼠标左键拖拽至脚本区中“当绿旗被点击”下。

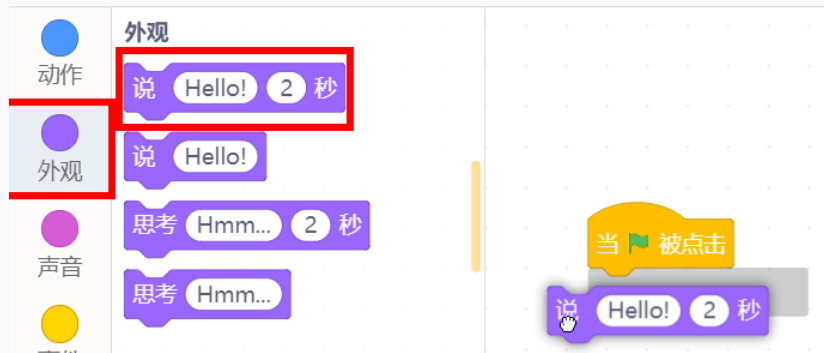


图 C-14 小猫说话

在指令区选择“Arduino”——“设置数字引脚输出”，引脚改为“13”，鼠标左键拖拽至脚本区中“说 hello! 2 秒”模块下。




图 C-15 设置数字引脚输出

完整程序，如下图：



图 C-16 小猫说话，点亮 UNO 板程序

6、程序编写完成后点击菜单栏的  运行，可以看到小猫说“Hello!”后，UNO 板和扩展板上的 D13 引脚信号指示灯亮起。

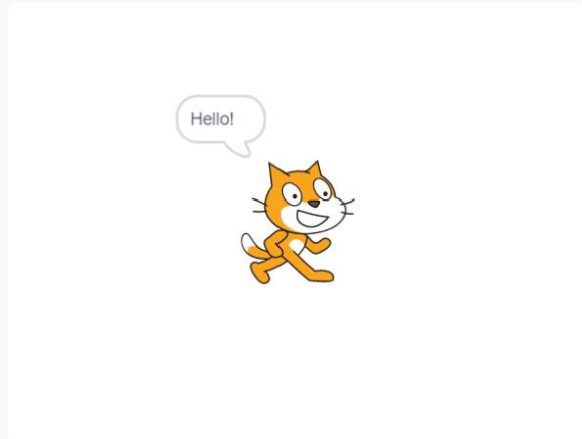


图 C-17 舞台区效果

执行器-显示输出设备

小贴士

这里的 D13 引脚信号指示灯就是一种执行器，可作为显示输出设备。执行器也有很多种，最常见的是“动”。好比人的动作，任何动作我们需要借助电机来完成。有了电机才能让东西“动”起来。其他的还有“声音”，“光”表现形式。蜂鸣器和喇叭就可以实现声音的输出。

恭喜你成功上传了你的第一个程序哦!

第一章 星光剧场

星光剧场的演出即将开始，各位演员已经准备就绪，让我们和他们一起来感受 Mind+ 带来的乐趣吧!



图 1 星光剧场




第一节 小麦变身

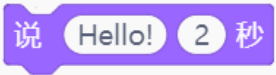

猫咪小麦可是剧场的大明星呢，它有一个变身的绝活，让我们一起来看看吧。

任务分析

角色	猫咪
背景	剧场
实现效果	1、小猫从舞台一侧移动到舞台中央； 2、说“Hello!” 3、说“我会变身！变！” 4、表演变身。

知识要点


指令	功能描述	所属模块
 当  被点击	触发事件。当绿旗被点击，执行下面的指令。	事件
 移动 10 步	使角色向面向方向移动 10 步，数字可更改。	动作

	角色说“Hello!”并保持 2 秒，说话内容和保持时间均可更改。	外观
	将角色颜色特效增加 25。除了颜色外还可选择：超广角镜头、旋转、像素化、马赛克、像素和虚像；数字也可更改。	外观

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“剧场”。

◇ 关于背景

背景库图标  位于软件右下角，将鼠标悬浮在背景库图标上，显示以下 4 个图标。舞台的背景可以通过这四种方法新建或修改，即选择背景库、画笔、随机、上传背景。

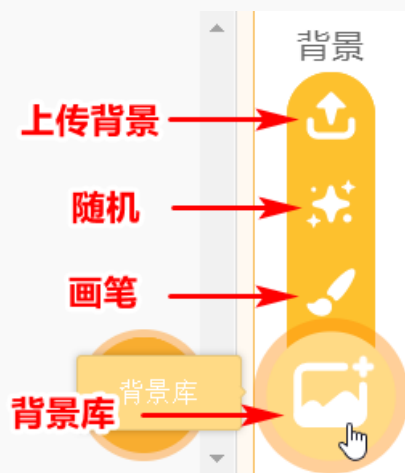



图 1.1.1 添加背景

选择背景库 鼠标左键单击  “背景库”，Mind+软件自己有很多库存，鼠标单击选择你喜欢的图片，舞台区的背景就变成我们选的图片啦。

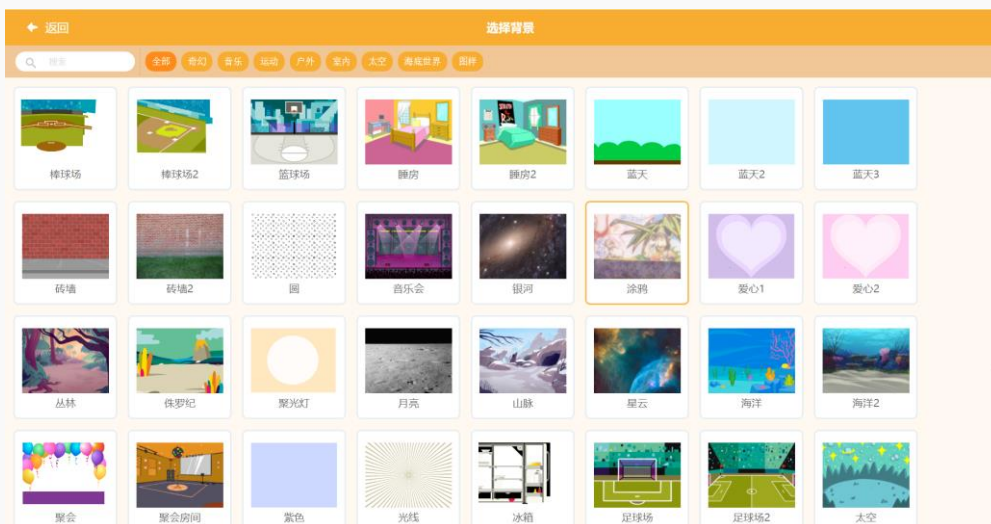


图 1.1.2 背景库

画笔 可以通过画笔绘制背景，鼠标单击  “画笔”，指令区和脚本区变成画图区，在画布上大家可以动手自己画画啦。

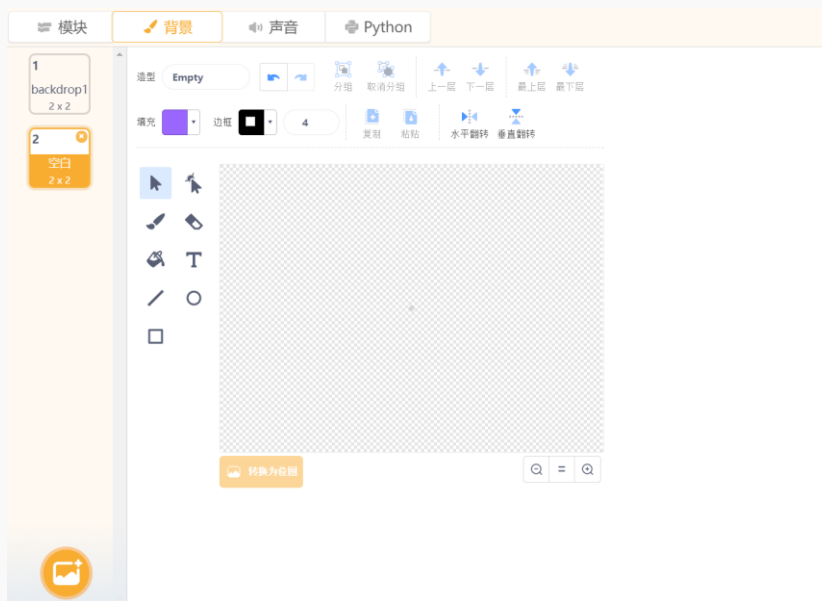




图 1.1.3 绘制背景

我们不仅可以在空白的画布上画画，还可以在原本的背景图片上画画哦，只要单击选中右侧的其中一个背景，就可以在原来的背景图片上做修改。

随机 当单击  “随机”时，我们会发现，舞台背景变了，背景图片不是由我们选择的，而是背景库中的任意一张，这是不是很神奇呢。

上传背景 当背景库中的图片无法满足我们的需求时，我们可以点击 “上传背景”，从计算机本地文件中选择图片，然后点击“打开”。

3. 新建角色。舞台默认出现的角色就是我们需要的角色，所以不需要添加，但需要将**角色命名**为小麦。

◇ 关于角色



在软件右下角找到，角色库的图标和背景库的图标相邻，他们的新建和修改方法也是一样的。将鼠标悬浮在角色库图标上，显示如图 1.1.4 所示 4 个图标，分别是选择角色库、画笔、随机、上传。每个方法的具体使用参考新建背景的四种方法。角色区其他各部分功能如图 1.1.5 所示。



图 1.1.4 添加角色



图 1.1.5 角色区部分功能




4. 选择触发事件。在**指令区**选择“事件”——“当绿旗被点击”，鼠标拖拽指令至脚本区。



图 1.1.6 选择触发事件

小贴士

删除指令有如下两种方法:

1、将鼠标指针移动到想要删除的指令上，鼠标指针变为“”；按下鼠标左键选中该模块，鼠标指针变为“”，此时拖拽模块至指令区，当鼠标指针变为“”，松开鼠标左键，这样就可以轻松删除啦！

2、将鼠标指针放置在程序中要删除的指令上，右击选择“删除积木块”即可。



5. 小麦从舞台一侧移动至舞台中央。选择“动作”——“移动 10 步”，将数字“10”改为“160”，如图 1.1.7。同时，将角色小麦移动至舞台一侧，这样才能使程序开始执行后角色移动到舞台中央，如图 1.1.8。



图 1.1.7 角色移动



图 1.1.8 移动角色至舞台一侧

6. 小麦说“Hello!”和“我会变身! 变!”。选择“外观”——“说 Hello! 2 秒”，鼠标拖拽指令至脚本区与其他指令组合。让小麦说“我会变身! 变!”的方法同上，只需将“Hello!”改为“我会变身! 变!”。程序如图 1.1.9。



图 1.1.9 添加小麦说话指令

小贴士

当需要重复使用同一指令时有两种方法：

1、重新在指令区拖拽所需指令，一个指令可重复使用多次。

2、将鼠标指针放置在程序中所需指令上，右击选择“复制”即可。



7. 小麦变身。选择“外观”——“将颜色特效增加 25”，将“颜色”改为“马赛克”，如图 1.1.10。



图 1.1.10 添加马赛克特效指令

8. 保存项目，命名为“小麦变身”。选择菜单栏“项目”——“保存项目”；选择保存位置；命名项目；点击保存，如图 1.1.11 和图 1.1.12。



图 1.1.11 点击“保存项目”

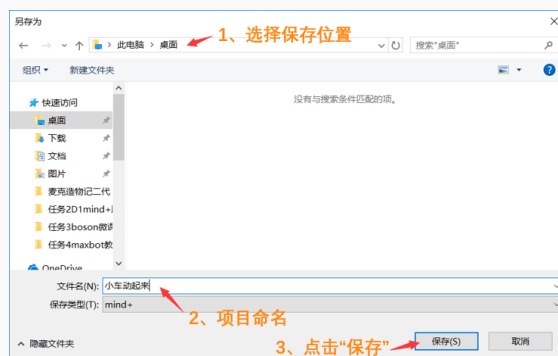


图 1.1.12 保存项目


完整程序如图 1.1.13 所示，点击菜单栏的  欣赏下小麦的变身秀吧！



图 1.1.13 完整程序

探究思考

小麦上台还没有做自我介绍呢，大家来帮助小麦做一个自我介绍吧！在设计图纸中写下你的想法吧。




第二节 空空走钢丝

小麦的变身秀结束啦，是不是很精彩呢。小猴子空空上台啦，快来看看他有什么绝活吧。

任务分析

角色	猴子空空
背景	剧场
实现效果	<ol style="list-style-type: none"> 空空在舞台中打招呼说“大家好！我来给大家表演走钢丝！”。 空空移动到钢丝上。 空空在钢丝上左右行走。

知识要点

指令	功能描述	所属模块
	触发事件。当键盘上空格键被按下时，执行下面程序。除空格键外也可以选择其他按键触发。	事件
	角色移动到指定位置 (x, y)。	动作
	设置角色的旋转模式。有三种模式：左右翻转、不能旋转、任意旋转。	动作
	当角色移动至舞台区边缘时自动返回。	动作
	重复执行循环体指令。循环体为填入指令“循环执行”下空白处的程序块。	控制

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“剧场”。
3. 添加角色空空，删除默认角色。选择“角色库”——“猴子”，将角色命名为“空空”，比例为 70，方向为 90。删除软件默认的小猫角色。
4. 绘制角色钢丝。
 - (1) 选择角色下画笔工具，打开如图 1.2.1 所示界面。



图 1.2.1 绘制新角色


小贴士

位图和矢量图：

位图：位图图像也称为点阵图像，位图使用我们称为像素的一格一格的小点来描述图像。

矢量图：矢量图是根据几何特性来绘制图形，是用线段和曲线描述图像，矢量可以是一个点或一条线，矢量图只能靠软件生成，矢量图文件占用内在空间较小，因为这种类型的图像文件包含独立的分离图像，可以自由无限制的重新组合。



(2) 使用  直线工具，在绘图区画一条直线，边框为黑色，无填充颜色，如图 1.2.2。

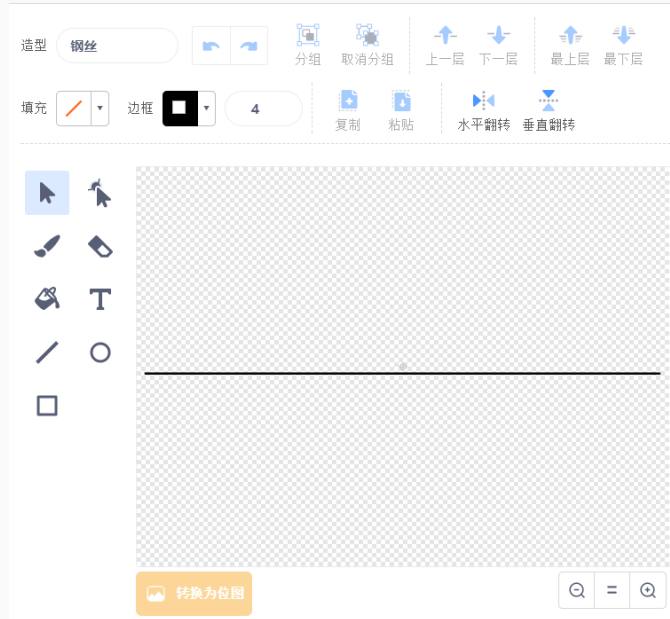


图 1.2.2 绘制钢丝

- (3) 鼠标选择舞台区中的角色“钢丝”，左键点击拖拽移动，将其移动到舞台合适位置，如图 1.2.3。



图 1.2.3 将“钢丝”放置在适当位置

5. 猴子空空移动到舞台中央。

- (1) 在角色区选择角色空空，只有选中角色才能移动以及为该角色编写程序。将空空移动到舞台中央适当位置，如图 1.2.4 所示。确定位置后，我们可以看到空空所在位置 x 坐标为-2，y 坐标为-90。



图 1.2.4 空空移动到舞台中央适当位置

◇ 关于 x 和 y

(x, y) 是一种坐标表示方式, x 坐标即横坐标, y 坐标即纵坐标。在体育课站队过程中, 老师经常会说“第几行第几列的某某同学”, 由此指定到具体的某一个同学。x, y 和行列的作用是一样的, 都是具体到舞台中的一个位置, 如图 1.2.6 所示。在角色区中可以查看当前角色的坐标值, 如图 1.2.7。

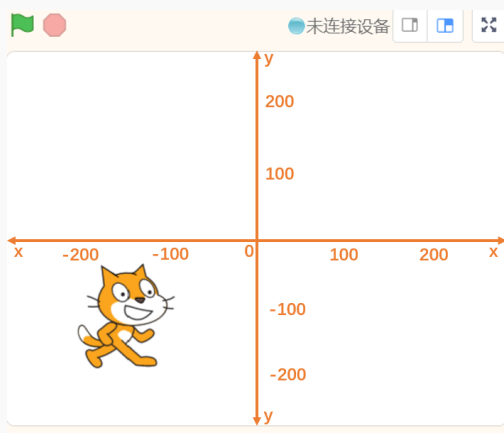


图 1.2.6 舞台坐标



图 1.2.7 查看角色坐标

(2) 选择“动作”——“移到 (x, y)”, 依据刚才的数值修改指令, 拖拽至脚本区, 如图 1.2.5 所示。



图 1.2.5 移到 (-2, -90)

- 6. 空空打招呼。选择“外观”——“说 Hello! 2 秒”，将“Hello!” 改为“大家好! 我来给大家表演走钢丝!”。



图 1.2.8 空空打招呼

- 7. 空空移动到钢丝上。方法参考步骤 5，完成空空走钢丝初始位置的选择，确定 x 坐标和 y 坐标，选择“移到 (x, y)”，如图 1.2.9 所示。



图 1.2.9 空空移动到钢丝上

- 8. 空空在钢丝上左右移动。

- (1) 要实现空空一直在钢丝上左右移动，需要用到“控制”模块下的“循环执行”，如图 1.2.10 所示。



图 1.2.10 循环执行

- (2) 选择“动作”——“移动 10 步”、“设置旋转模式左右翻转”和“碰到边缘就反弹”组合放入循

环体位置。试试改变移动步数，看看效果有什么不同吧。

9. 保存项目，命名为“空空走钢丝”。

完整程序如图 1.2.11 所示，按下空格键执行程序，来看猴子空空走钢丝吧！

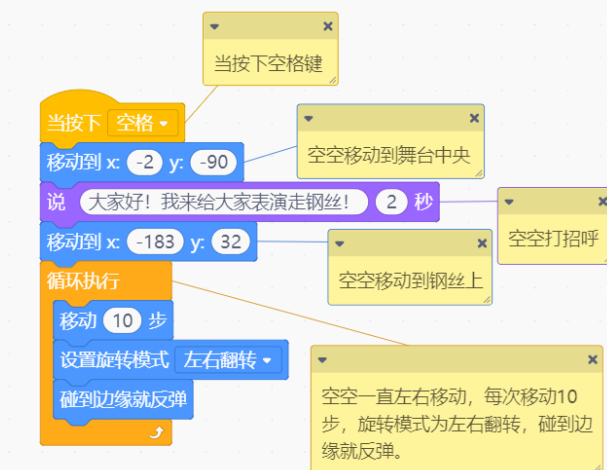


图 1.2.11 空空走钢丝完整程序

探究思考

猴子空空一直以一个造型表演走钢丝，可不可以使用指令区的指令让空空换一个造型呢？在设计图纸中写下你的想法吧。

提示：点击菜单栏的“外观”可以查看不同的造型哦！

设计图纸




第三节 点点踩球

压轴大秀要来啦！小狗点点的踩球表演可以一等的棒哦！

任务分析

角色	点点
背景	剧场
实现效果	1、点点站在球上说“我可以在球上保持平衡哦!”。 2、点点在球上不断切换造型。 3、球不断转动。

知识要点

指令	功能描述	所属模块
	角色顺时针旋转 15 度，角度可更改。	动作
	角色切换为下一个造型。	外观
	持续上一条指令状态 1 秒，持续时间可更改。	控制

动手实践

1. 打开 Mind+ 软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“剧场”。
3. 添加角色，删除默认角色。删除软件默认的小猫角色；选择“角色库”——“点点”，将角色命名为“点点”；选择“角色库”——“球”，将角色命名为“球”，比例改为 230。
4. 将角色点点和球放置在舞台中合适的位置，如图 1.3.1 所示。

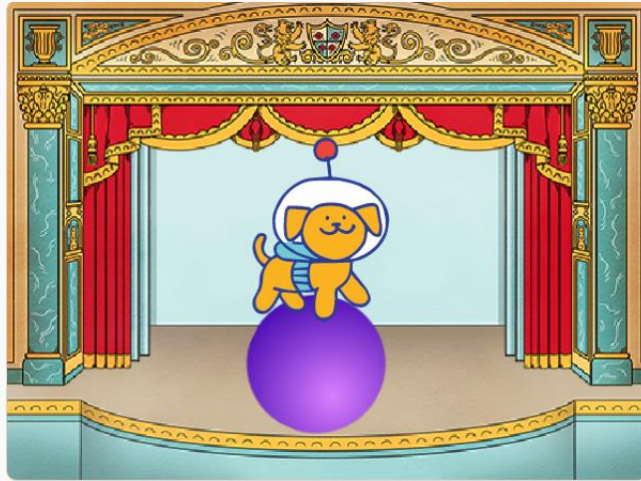


图 1.3.1 点点和球位置

5. 为角色点点编写程序，实现点点说话后在球上不断变换造型。

- (1) 在角色区选择角色点点，选择“事件”——“当绿旗被点击”。
- (2) 选择“外观”——“说 Hello! 2 秒”，将“Hello!”改为“我可以在球上保持平衡哦!”。
- (3) 选择“控制”——“循环执行”。
- (4) 循环体。选择“外观”——“下一个造型”，“控制”——“等待 1 秒”，将时间改为 0.2 秒。

试试不同的等待时间下效果有什么不同吧。

◇ 关于造型

一个角色可以有多个造型，Mind+中如何查看角色造型呢？

在角色区选择想要查看的角色，点击菜单栏中的“外观”，界面如图 1.3.2 所示。

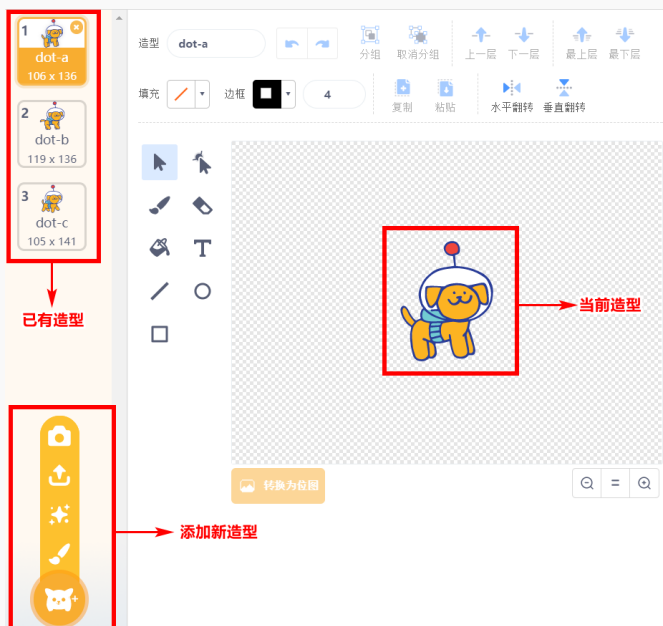


图 1.3.2 查看角色造型

只有当角色的造型个数大于等于两个时，才可以运用指令区“外观”下的“下一个造型”哦。

角色点点完整程序如图 1.3.3 所示：

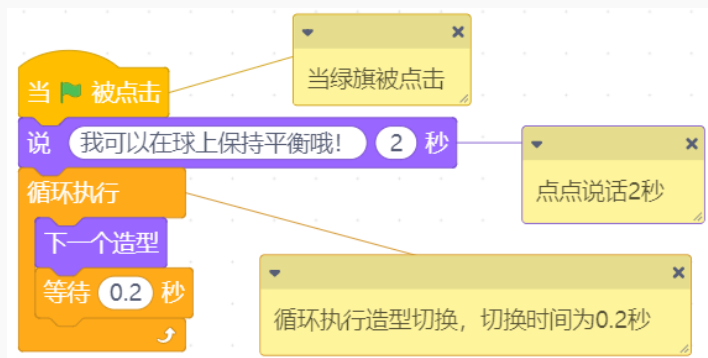


图 1.3.3 角色点点完整程序

6. 为角色球编写程序，实现角色点点说话后，球上不断旋转。

- (1) 在角色区旋转角色球，选择“事件”——“当绿旗被点击”。
- (2) 选择“控制”——“等待 1 秒”，将时间改为 2 秒。

为什么是等待 2 秒呢？

在程序开始时，角色点点说话时间为 2 秒，之后才开始切换造型，模仿在球上保持平衡的状态，此时才需要角色球开始旋转。等待 2 秒是为了等待角色点点说话 2 秒的时间。

- (3) 选择“控制”——“循环执行”。
- (4) 循环体。选择“动作”——“向右旋转 15 度”，将角度大小改为 5 度。

角色球的完整程序如图 1.3.4 所示：



图 1.3.4 角色球完整程序

- 7. 保存项目，命名为“点点踩球”，点击 查看效果。

探究思考

点点一直在进行原地踩球保持平衡，能不能让点点和球一起在舞台上动起来，实现边移动边保持平衡呢？

大家来试试吧！在设计图纸中写下你的想法吧。

提示：可以使用多个触发事件。

设计图纸

第二章 魔法小屋

星光剧场的表演有没有让你印象深刻呢？然而，仅仅作为旁观者来观看，是不是完全不够尽兴，快走进魔法小屋，这里有很多有趣的互动小玩意儿，带上小伙伴儿一探究竟吧！



图 2 魔法小屋

第一节 彩虹琴

听说魔法小屋的琴不仅声音动人悦耳，而且里面还有彩虹哦！快去看看吧。

任务分析

角色	彩虹琴七个琴键
背景	聚光灯
实现效果	按下不同颜色的琴键能够发出不同的声音。

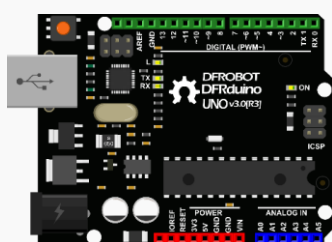
知识要点

名称	电子元件	功能描述
数字蜂鸣器		数字蜂鸣器是 Arduino 传感器模块中最简单的发声装置，只要简单的高低电平信号就能够驱动，可以通过频率来控制音调。它结构简单、应用丰富，能够模拟我们生活中许多声音和音乐。

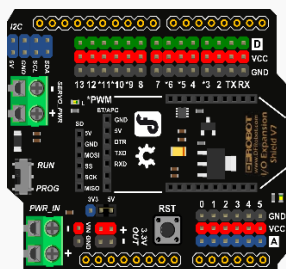
指令	功能描述	所属模块
	触发事件，当角色被点击执行下面程序。	控制
	设置蜂鸣器连接的引脚、音调和节拍。	Arduino

硬件连接

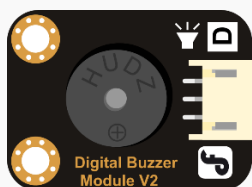
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字蜂鸣器模块 X1

连接图

数字蜂鸣器模块连接在数字引脚 7 处，UNO 板连接计算机。

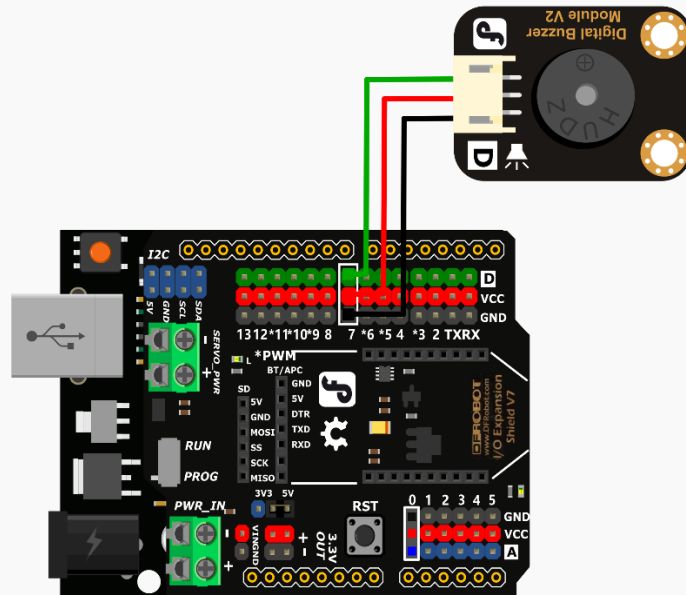


图 2.1.1 硬件连接图

◇ 关于数字和模拟

数字信号与模拟信号的区别

电子世界有两种“语言”——数字信号与模拟信号。电子世界的数字与模拟与我们平常说的数字与模拟不同。这里的数字，并不是代表的阿拉伯数字的意思。这里的模拟，也不是我们日常认为的真实事物的虚拟。这里需要你颠覆对数字与模拟原有的概念，电子世界将给你一个全新的诠释。

数字信号和模拟信号

数字(Digital Signal): 只有 2 个值(0V 和 5V)。运用在 Arduino 中, 就是高(HIGH)或者低(LOW), “HIGH”是“1”, 对应为 5V。“LOW”是“0”, 对应为 0V。

模拟(Analog Signal): 在一定范围内, 有无限值。在 Arduino 中模拟口中, 已经将 0V 到 5V 之间的值映射为 0~1023 范围内的值。比如, 0 对应为 0V, 1023 对应为 5V, 512 对应为 2.5V。

小贴士

如何区分 DF 套件中的传感器为数字还是模拟?






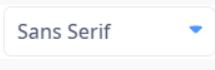
- 1、绿色线为数字信号的传感器, 蓝色线为模拟信号的传感器。
- 2、板上会印有 “ D ” 或者 “ A ” 的字样, “D” 代表 “数字”, “A” 代表 “模拟”。

DF 套件中各种颜色代表了什么?

绿色: 数字信号(Digital Signal) 蓝色: 模拟信号(Analog Signal) 红色: 电源 黑色: GND

动手实践

- 1. 打开 Mind+软件, 新建一个项目。
- 2. 添加舞台背景。选择 “背景库” —— “聚光灯”。
- 3. 删除角色区的默认小猫角色。
- 4. 绘制彩虹琴。

- (1) 角色区选择 “画笔”, 使用  矩形工具绘制琴键, 填充为 , 边框为  2。
- (2) 使用  文字工具在琴键上输入 “C”, 填充为 , 字体选择  Sans Serif。将角色命名为 “C”, 效果如图 2.1.2 所示。

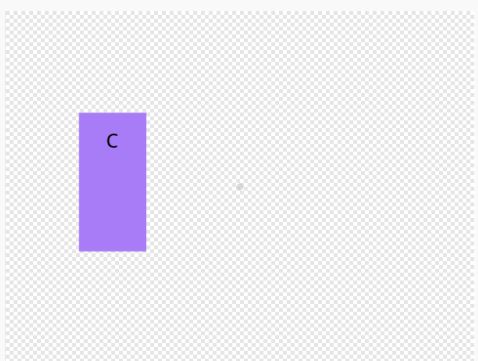


图 2.1.2 琴键 C

- (3) 按上述方法依次绘制琴键 D、E、F、G、A 和 B, 并且调整它们在舞台中的位置, 如图 2.1.3

所示。同学们可以自己选择琴键的颜色哦。

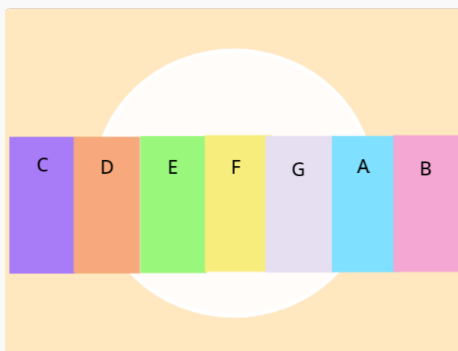


图 2.1.3 彩虹琴

5. 为不同颜色的琴键编写程序。

- (1) 在角色区选择角色“C”，在指令区选择“事件”——“当角色被点击”，拖拽至脚本区。
- (2) 选择“Arduino”——“设置喇叭”指令，引脚为“7”，音调为“中 C/C4”，节拍为“1/2”，放置在“当角色被点击”下。这样琴键 C 的程序就编写完成了，如图 2.1.4。

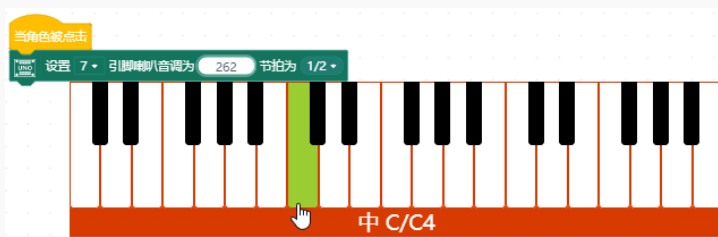


图 2.1.4 琴键 C 程序

小贴士

“C4” 和 “1/2” 表示什么？

“C” 是音符 Do(即音符 1)，字母后面的数字是音符的音高，即“4”表示中音，那么“C4”就表示中音 Do；“1/2”是节拍，节拍关系到曲子的快慢和节奏。对应关系如下图：

C	D	E	F	G	A	B
1(Do)	2(Re)	3(Mi)	4(Fa)	5(So)	6(La)	7(Si)

- (3) 其他琴键的程序编写与琴键 C 类似，只需要更改音调为 D4、E4、F4、G4、A4 和 B4，如图

2.1.4~2.1.9 所示。注意，编程前一定要在角色区选中要进行编程的角色哦。

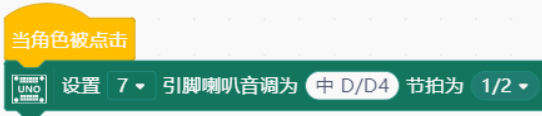


图 2.1.4 琴键 D 程序



图 2.1.5 琴键 E 程序



图 2.1.6 琴键 F 程序



图 2.1.7 琴键 G 程序



图 2.1.8 琴键 A 程序



图 2.1.9 琴键 B 程序

6. 保存项目，命名为“彩虹琴”，连接设备，选择串口，点击不同颜色的琴键，下一个音乐家就是你啦！

探究思考

试一试可不可以使用指令实现识别不同颜色演奏不同音符呢？写下你的想法吧！

提示：



第二节 吹动风车

听说在魔法小屋竟然可以吹动屏幕里面的风车哦，不信就来看看吧！

任务分析

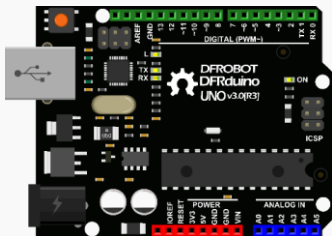
角色	风车
背景	蓝天 2
实现效果	通过向声音传感器吹气，吹动风车。

知识要点

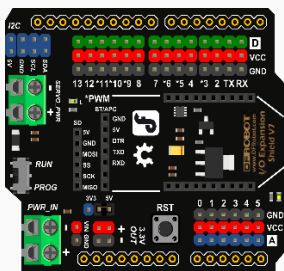
名称	电子元件	功能描述	
模拟声音传感器		模拟声音传感器能感知声音的大小，并转化为模拟信号，即通过反馈的电压值来体现声音的大小。	
指令	功能描述	所属模块	
	如果满足判断条件，那么执行指令内程序，否则执行指令外程序。	控制	
	空白处条件 1 大于条件 2。	运算符	
	读取模拟引脚 A0 的值。引脚范围为 A0~A7，但是扩展版上只有引脚 A0~A5。	Arduino	

硬件连接

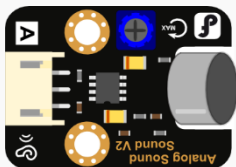
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



模拟声音传感器 X1

连接图

模拟声音传感器连接在模拟引脚 A0 处，UNO 板连接计算机。

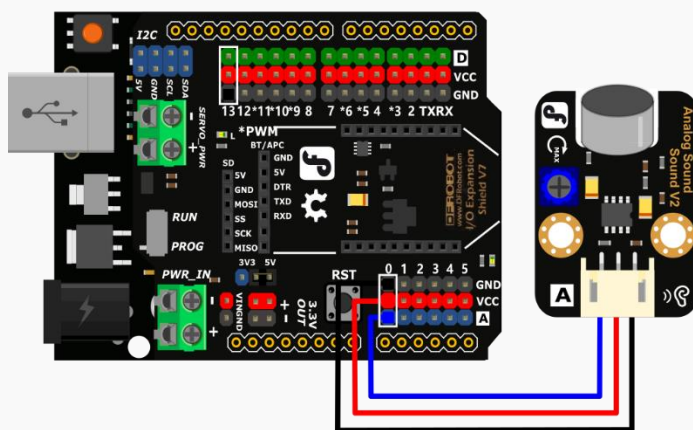


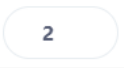




图 2.2.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“蓝天 2”。
3. 删除角色区的默认小猫角色。
4. 绘制风车。

(1) 角色区选择“画笔”，使用  直线工具绘制风车轮廓，边框为  2 .

- (2) 选择  选择工具，选中全部线条，点击“分组”，这样所有线条就被组合在一起了。
- (3) 选择组合后的图案，更改填充颜色为 ，将左下角放在绘图区中心点上，效果如图 2.2.2 所示。

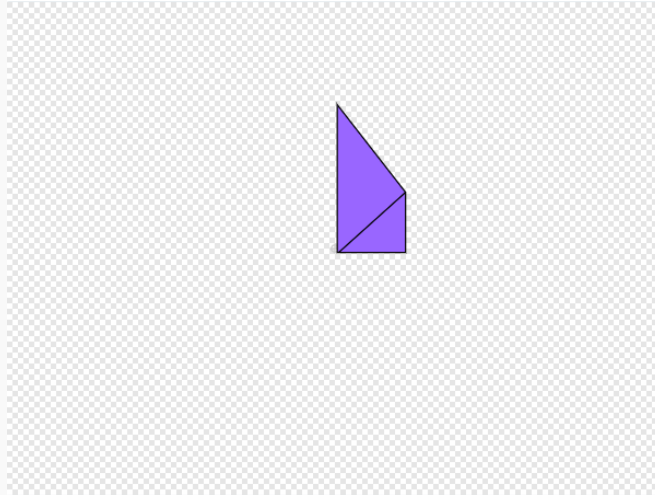


图 2.2.2 部分风车

- (4) 依据步骤 (1) ~ (3) 绘制风车的其他部分，整体组合。注意，风车的中心要在绘图区的中心点上，这是因为角色的旋转是以中心点为中心的。将角色命名为“风车”，如图 2.2.3 所示。

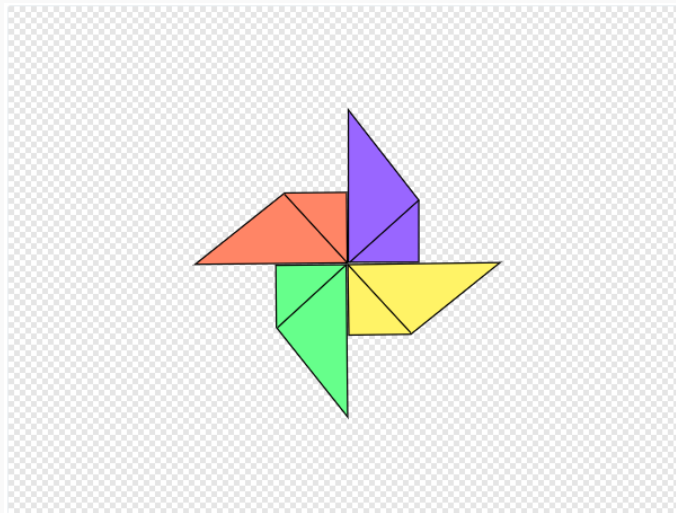




图 2.2.3 风车

- 5. 绘制木棒。角色区选择“画笔”，使用  直线工具绘制木棒，边框为  2。将角色命名为“木棒”，如图 2.2.4 所示。调整木棒和风车在舞台中的位置，如图 2.2.5 所示。

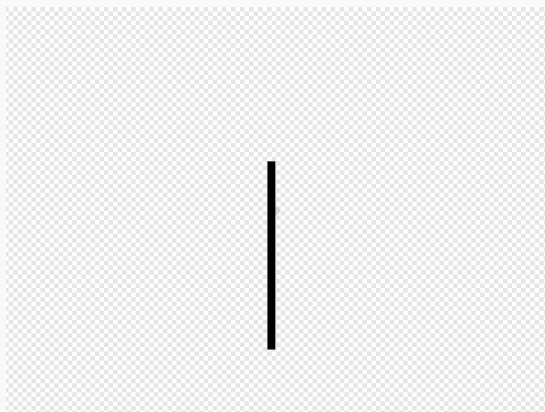



图 2.2.4 木棒



图 2.2.5 舞台区效果

6. 编写程序实现当向声音传感器吹气时，风车转动。风车完整程序如图 2.2.6 所示。

- (1) 在角色区选择角色风车，选择“事件”——“当绿旗被点击”，拖拽至脚本区。
- (2) 选择“控制”——“如果那么执行”指令，拖拽至脚本区与“当绿旗被点击”组合。
- (3) 当声音传感器接收到的值足够大时，风车才转动，因此，判断条件为当声音传感器接收到的值大于临界值 50。选择“运算符”——“大于 100”指令，将“100”改为“50”；选择“Arduino”——“读取模拟引脚 A0”，放入“大于 50”指令中的空白处，即 。

小贴士

如何确定声音传感器的临界值？

在实际生活中，如果吹气的力度太小，风车是不能转动的，只有力度足够大时风车才能转动。因此，为了实现吹动风车的效果，需要设定一个临界值，使得声音传感器接收到的值大于临界值时，风车才转动。

声音传感器接收到的值可以通过编写程序查看，程序如下，这种方法可以帮助我们获取到当前声音传感器接收到的值，这种方法对于其他传感器也适用哦！



- (4) 当满足判断条件时，风车转动。选择“动作”——“向右旋转 15 度”指令，放入“如果那

么执行”指令内。

- (5) 要使风车产生持续转动的效果，需要加入“循环执行”指令。选择“控制”——“循环执行”指令，将步骤(2)~(4)所编写的程序放在循环体位置。

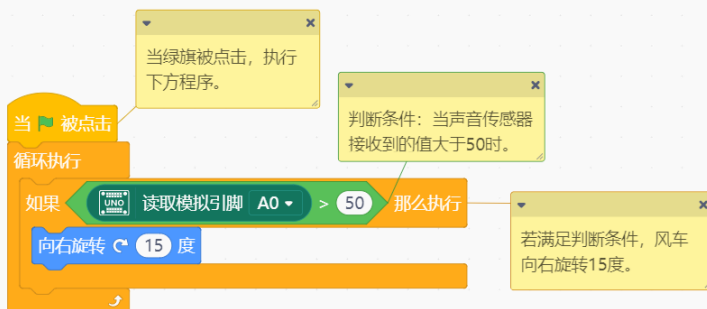



图 2.2.6 风车完整程序

- 7. 保存项目，命名为“吹动风车”，连接设备，选择串口。点击 ，吹口气，让风车转起来吧。

探究思考

可不可以制作一个变速风扇呢？声音越大风车转动的越快。写下你的想法吧！



第三节 点亮星灯

魔法小屋里面有一片星空哦，一起来看看吧！

任务分析

角色	多个星星
背景	群星

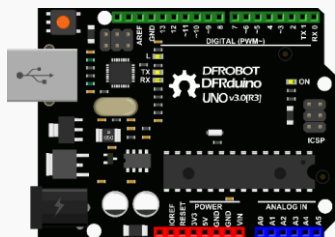
实现效果	按下按钮可以点亮夜空中的星星。
-------------	-----------------

知识要点

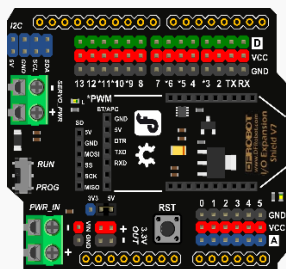
名称	电子元件	功能描述	
数字按钮模块		按压式的开关数字输入模块,与 Arduino 专用传感器扩展板结合使用,按钮模块使用大按钮加优质按键帽手感一流,使用方便可以做到“即插即用”。	
指令		功能描述	所属模块
		角色移动到舞台中随机位置。“随机位置”可更改为鼠标指针或其他角色位置。	动作
		显示/隐藏舞台中的角色。	外观
		如果满足判断条件,则执行“那么执行”下的程序;如果不满足判断条件,则执行“否则执行”下的程序。	控制
		读取数字引脚的值。	Arduino

硬件连接

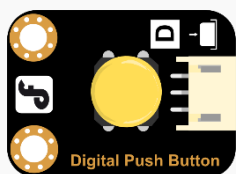
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字按钮模块 X1

连接图

数字按钮模块连接在数字引脚 2 处，UNO 板连接计算机。

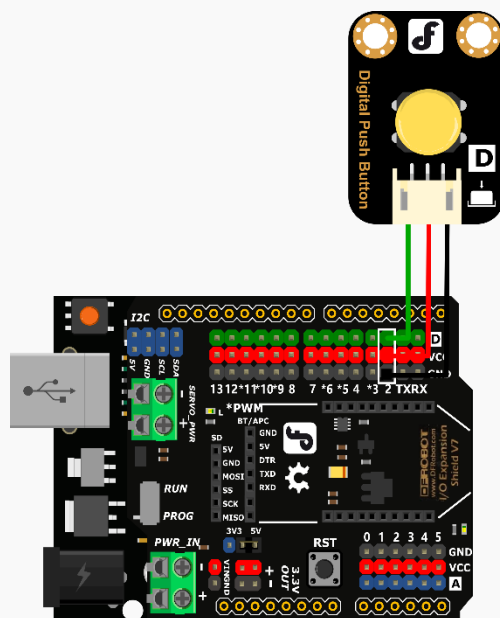





图 2.3.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“群星”。
3. 删除角色区的默认小猫角色。
4. 绘制星星。

- (1) 角色区选择“画笔”，使用  矩形工具绘制正方形，填充颜色为 ，无边框。
- (2) 选择  变形工具，鼠标点击正方形的其中一边的中心点向下拖拽，如图所示 2.3.2，其他三条边操作同上。

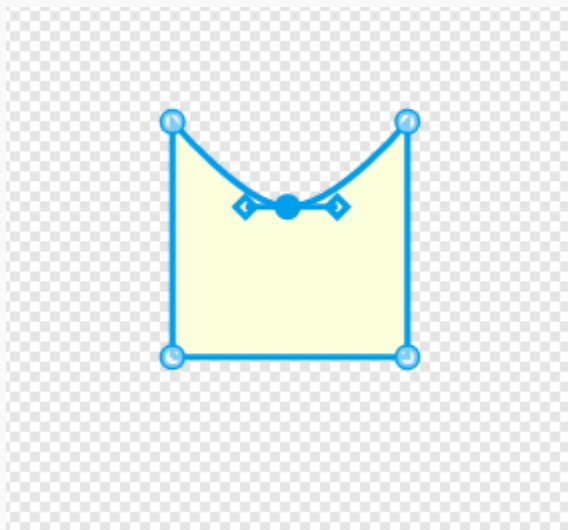



图 2.3.2 正方形变形



图 2.3.3 角色星星造型 1

- (3) 选择  选择工具，调整角色大小，旋转整个图案，如图 2.3.3 所示。

5. 编写程序，实现点亮星灯：按下按钮，星星出现在舞台的随机位置。完整程序如图 2.3.4 所示。

- (1) 在角色区选择角色 star，选择“事件”——“当绿旗被点击”，拖拽至脚本区。
- (2) 选择“控制”——“如果那么执行否则执行”指令，拖拽至脚本区与“当绿旗被点击”组合。
- (3) 判断条件为按下按钮，即读取数字引脚值为 1。选择“Arduino”——“读取数字引脚值”，放入判断条件位置。
- (4) 按下按钮，点亮星灯。选择“外观”——“显示”指令，“动作”——“移到随机位置”指令，按顺序组合。在“移动到随机位置”指令后加入延时指令，即“等待 0.5 秒”，实现星星出现的效果。将组合后的程序放入“如果那么执行否则执行”指令“那么执行”下。选择“外观”——“隐藏”指令放入“否则执行”下。
- (5) 要使每次按下按钮都能实现点亮星灯的效果，需要加入“循环执行”指令。选择“控制”——

“循环执行”指令，将步骤（2）~（4）所编写的程序放在循环体位置。

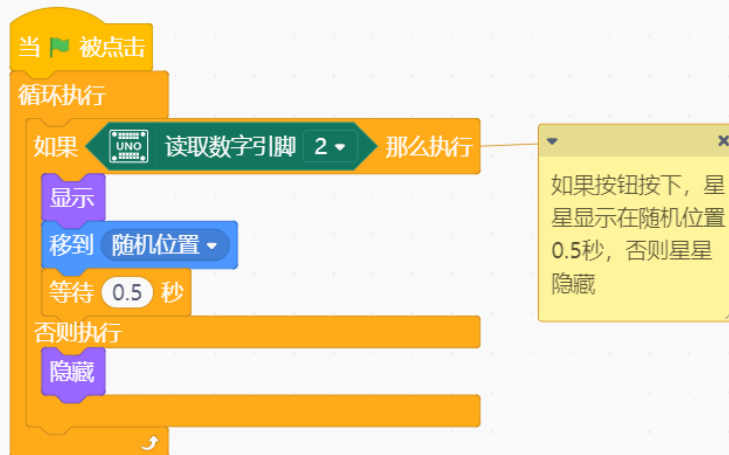



图 2.3.4 点亮星灯程序

6. 在角色区复制角色 star 增加星星个数，此时角色的程序和造型会一并复制，可复制多个，效果更好。

如图 2.3.7.



图 2.3.7 复制角色

7. 保存项目，命名为“点亮星灯”，连接设备，选择串口。点击 ，按下按钮，点亮星灯吧。实现效果如下图 2.3.8 所示。

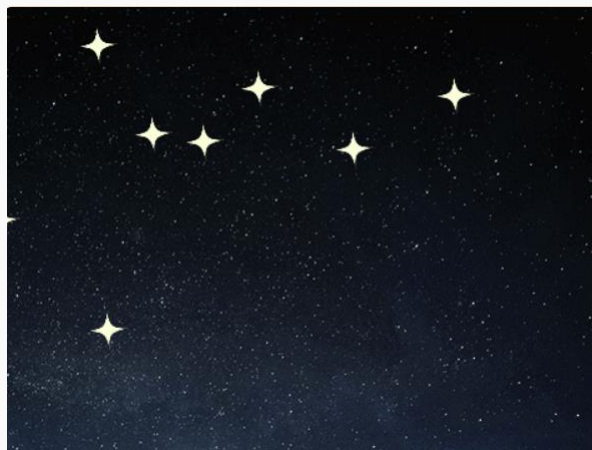


图 2.3.8 点亮星灯效果

探究思考

试一试在星星随机出现时切换不同造型。写下你的思路吧。



第三章 造梦空间

魔法小屋有没有让你收获满满呢？各种互动媒体小东西是不是很有趣呢？其实有些互动媒体作品是非常有实际应用价值的哦，我们一起去造梦空间看看吧！



图 3 造梦空间

第一节 防盗报警器


造梦空间里有很多珍贵的仪器，防盗报警器是非常必要的，小心一些，不要触发警报啦。

任务分析

角色	美人鱼、钻石、摆台
----	-----------

背景	海洋 2
实现效果	红外数字避障传感器感知到有小偷经过，宝石被偷，喇叭报警，美人鱼出现。

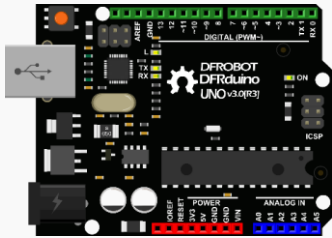
知识要点

名称	电子元件	功能描述
红外数字避障传感器		一种集发射与接收于一体的光电开关传感器。若检测到障碍，灯亮，输出信号“1”，若未检测到障碍，灯灭，输出信号“0”。检测距离可以根据要求进行调节，可调范围 3-80cm。

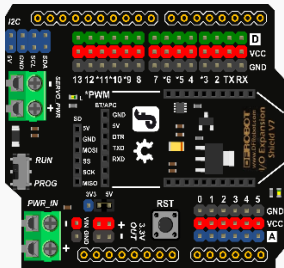
指令	功能描述	所属模块
	角色在某段时间内滑动到某一坐标位置，坐标和事件均可更改。	动作
	实现角色之间的通信，广播信息的名称可更改。	事件
	触发事件，当接收到广播信息时执行下方程序。	事件
	有限次重复执行循环体。	控制
	停止全部程序。可选项还包括：当前程序和角色中的程序。	控制

硬件连接

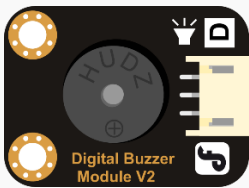
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字蜂鸣器模块 X1



红外数字避障传感器 X1

连接图

红外数字避障传感器连接数字引脚 2，蜂鸣器连接数字引脚 7，UNO 板连接计算机。

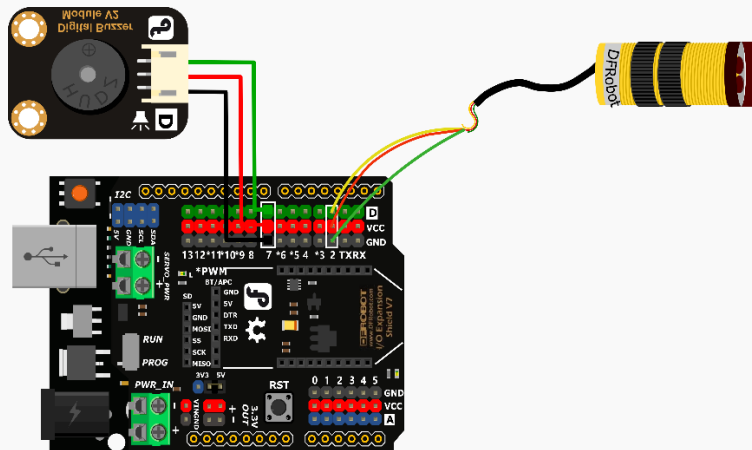


图 3.1.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“海洋2”。
3. 删除角色区的默认小猫角色。选择“角色库”——“美人鱼”和“钻石”两个角色。另外还需要使用



“画笔”—— 矩形工具和  圆形工具，绘制一个钻石展台角色。如图 2.3.2 所示。



图 3.1.2 舞台区

4. 角色钻石效果：当红外数字避障传感器感知到有人经过时，钻石隐藏，蜂鸣器发出声音报警，广播“宝石被偷”；若无人经过，钻石显示。流程图如下。

程序流程图是什么？

小贴士

程序流程图又称程序框图，是用统一规定的标准符号描述程序运行具体步骤的图形表示。程序框图的设计是在处理流程图的基础上，通过对输入输出数据和处理过程的详细分析，将计算机的主要运行步骤和内容标识出来。程序框图是进行程序设计的最基本依据，因此它的质量直接关系到程序设计的质量。

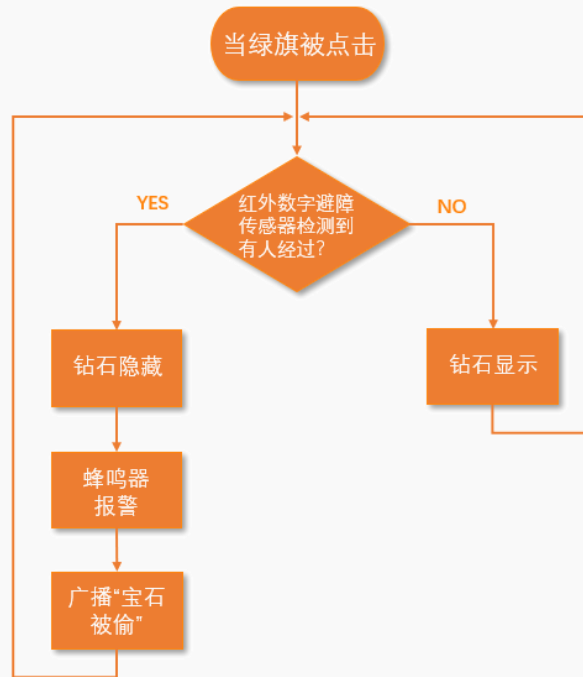


图 3.1.3 角色钻石流程图

根据流程图编写程序，角色钻石完整程序如图 3.1.4 所示。



图 3.1.4 角色钻石程序

- 5. 角色美人鱼效果：当绿旗被点击，即程序开始，美人鱼隐藏，移动到 (-294, 287)；当接收到广播“宝石被偷”，美人鱼显示，在 1 秒时间内移动到舞台中，说“宝石被偷啦”！流程图如图 3.1.5 所示。

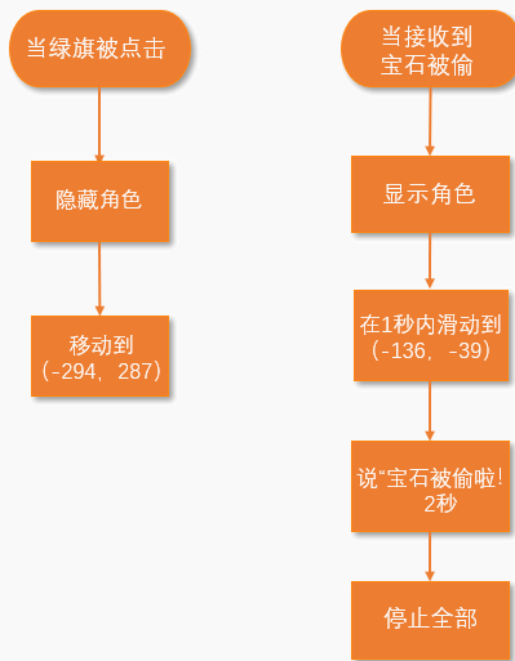


图 3.1.5 角色钻石流程图

根据流程图编写程序，角色钻石完整程序如图 3.1.6 所示。



图 3.1.6 角色美人鱼程序

6. 保存项目，命名为“防盗报警器”，连接设备，选择串口。效果图如下图所示。

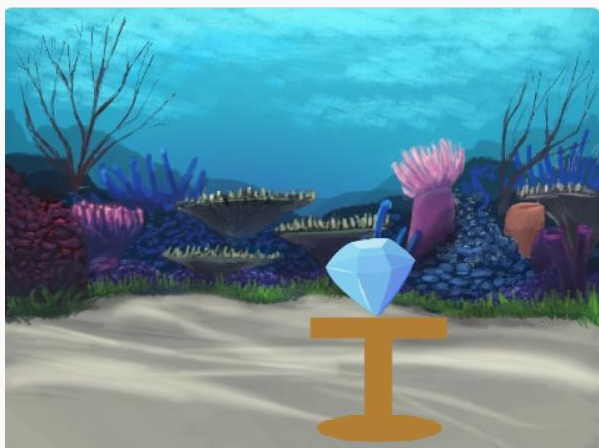


图 3.1.7 宝石被偷前效果图



图 3.1.8 宝石被偷后效果图

探究思考

尝试加入一个小偷角色，模拟宝石被偷走的过程吧！写下你的思路。

第二节 调光台灯

造梦空间里有一台调光台灯，旋转旋钮，点亮造梦空间和猫咪小麦的家吧。

任务分析

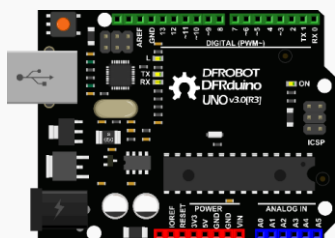
角色	猫咪
背景	睡房
实现效果	调节旋钮，灯变亮，房间也逐渐变亮。

知识要点

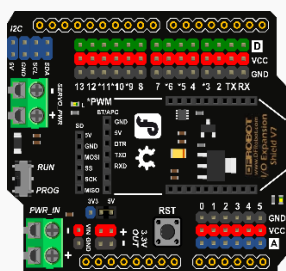
名称	电子元件	功能描述	
模拟角度传感器		基于电位器的旋转角度传感器，旋转角度从 0 到 300 度，取值范围为 0~1023，与 Arduino 传感器扩展板结合使用，可以非常容易地实现与旋转位置相关的互动效果。	
红色 LED 发光模块		LED 发光模块是入门玩家必备的电子元件，取值范围为 0~255。可以用数字端口控制灯的亮灭，也可以用模拟口控制它的亮度。输入高电平灯亮，低电平则灯灭。	
指令		功能描述	所属模块
		设置角色亮度为 0（正常），亮度范围为 -200~200。除了亮度外还可选择：超广角镜头、旋转、像素化、马赛克、像素和虚像；数字也可更改。	外观
		将某个值从 0~1023 映射到 0~255。	运算符
		空白处条件 1 和条件 2 相等。	运算符
		设置 PWM 引脚 3 输出值为 200，引脚和输出值均可更改。	Arduino

硬件连接

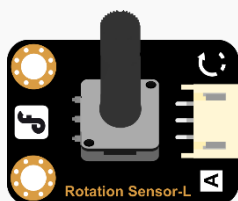
硬件清单



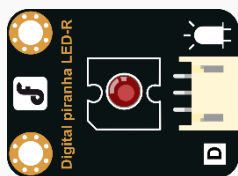
DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



模拟角度传感器 X1



红色 LED 发光模块 X1

连接图

红色 LED 发光模块连接在 PWM 引脚 3 处，模拟角度传感器连接在模拟引脚 A1 处，UNO 板连接计算机。

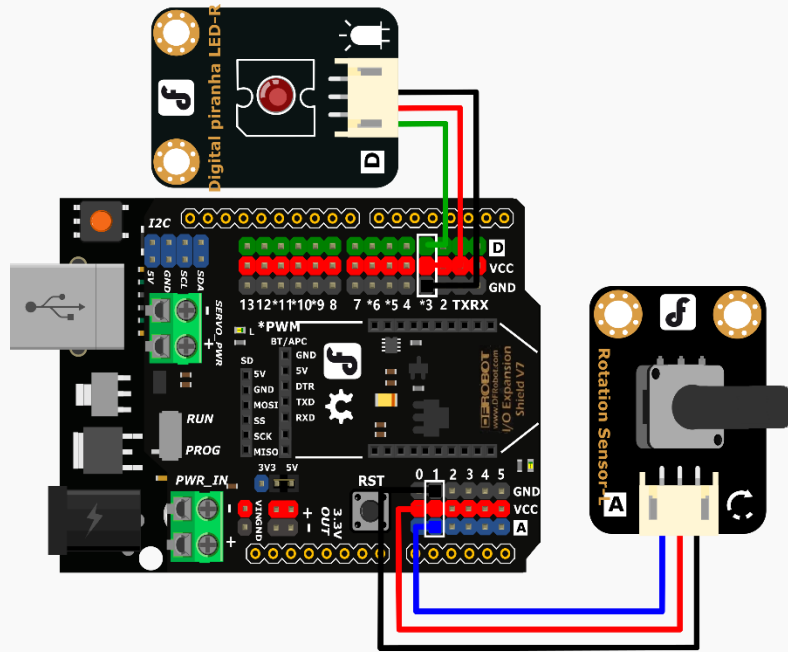


图 3.2.1 硬件连接图

小贴士

PWM 是什么？

脉冲宽度调制(PWM)，是英文 “ Pulse Width Modulation” 的缩写，简称脉宽调制，是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术。

在 UNO 板和扩展板上，PWM 引脚为数字引脚 3、5、6、9、10 和 11。

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择 “背景库” —— “睡房”。
3. 保留默认角色猫咪，将角色重命名为 “小麦”。
4. 背景睡房效果：旋转模拟角度传感器背景的亮度不断变化，同时，LED 的亮度也随之变化。流程图

如图 3.2.2 所示。



图 3.2.2 背景睡房效果流程图

根据流程图编写程序，背景睡房完整程序如图 3.2.3 所示。



图 3.2.3 背景睡房程序

小贴士

映射是什么？

在数学里，映射是个术语，指两个元素的集之间元素相互“对应”的关系，这是一种函数关系。

由于舞台背景亮度由暗到正常的取值范围为“-200~0”，而模拟角度传感器的取值范围是“0~1023”，为了实现旋转模拟角度传感器时舞台背景亮度不断变化，则需要让两个值一一对应，这时就需要用到映射，即，“0”对应“-200”，后面数值依此类推，直到“1023”对应“0”。

- 5. 角色小麦效果：当房间漆黑一片时说“好黑啊！把灯光调亮一些吧！”；当房间变亮时说“房间变亮

啦!”。流程图如图 3.2.4 所示。根据流程图编写程序，角色小麦完整程序如图 3.2.5 所示。

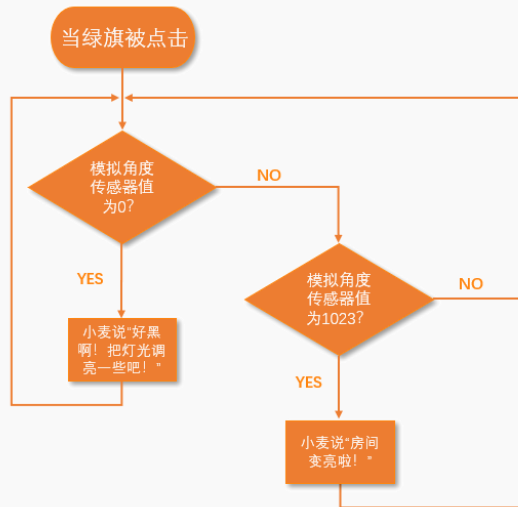


图 3.2.4 角色小麦效果流程图



图 3.2.5 角色小麦完整程序

6. 保存项目，命名为“调光台灯”，连接设备，选择串口。效果图如下图所示。



图 3.2.6 模拟角度传感器值为 0 时效果图



图 3.2.7 模拟角度传感器值为 1023 时效果图

探究思考

模拟角度传感器不只可以调节亮度，还可以用来切换不同背景，带着小麦去全世界各地旅行吧。写下你的设计思路。



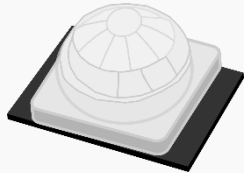
第三节 智能门铃


小麦非常喜欢朋友到家里玩，但是他在房间里时经常听不到门铃响，朋友只好离开。造梦空间里有一个可以放在房间里的智能门铃，正好能帮到小麦，我们一起来看看吧。

任务分析

角色	猫咪
背景	睡房
实现效果	无人经过，灯不亮，小猫说“今天天气真好”；有人经过，灯闪烁，小猫说“来客人啦”。

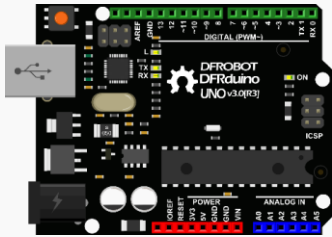
知识要点

名称	电子元件	功能描述	
人体红外热释电运动传感器		红外热释电运动传感器能检测运动的人或动物身上发出的红外线，输出开关信号，可以应用于各种需要检测运动人体的场合。	
指令		功能描述	所属模块

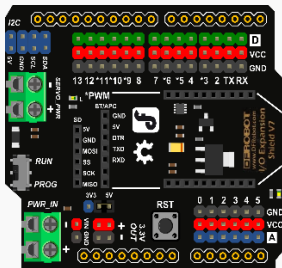
	<p>重复执行循环体直到满足判断条件。</p>	<p>控制</p>
---	-------------------------	-----------

硬件连接

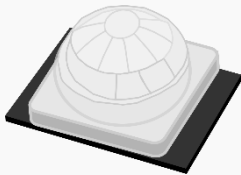
硬件清单



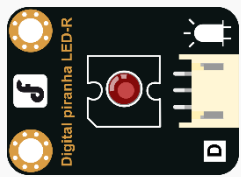
DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



人体红外热释电运动传感器 X1



红色 LED 发光模块 X1

连接图

红色 LED 发光模块连接在数字引脚 13 处，人体红外热释电运动传感器连接在数字引脚 2 处，UNO 板连接计算机。

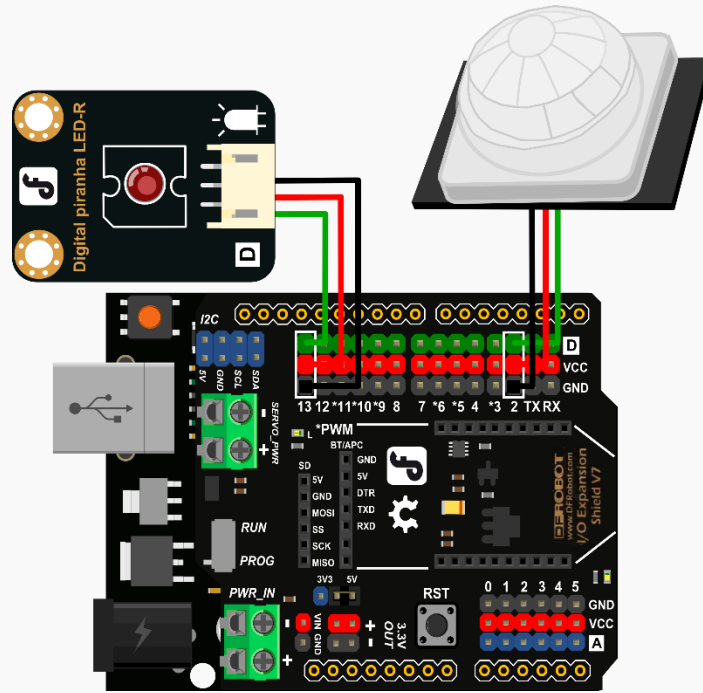


图 3.3.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“睡房”。
3. 保留默认角色猫咪，将角色重命名为“小麦”。
4. 角色小麦效果：灯不亮，小麦说“今天天气真好！”，直到运动传感器感知到有人经过，灯闪烁，小麦说“来客人啦！”。流程图如图 3.3.2 所示。

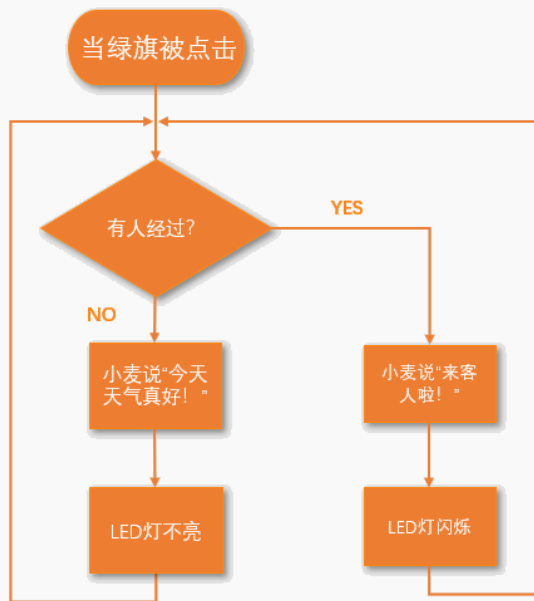


图 3.3.2 角色小麦效果流程图

根据流程图编写程序，角色小麦完整程序如图 3.3.3 所示。



图 3.3.3 角色小麦完整程序

5. 保存项目，命名为“智能门铃”，连接设备，选择串口。效果图如下图所示。



图 3.3.4 无人经过效果图



图 3.3.5 有人经过效果图

探究思考

尝试加入蜂鸣器，实现当有人经过时，灯闪烁，蜂鸣器发声的效果。写下你的思路吧！



设计图纸

第四节 感应走廊灯

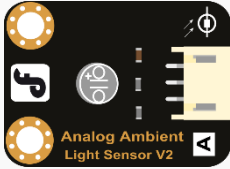

在造梦空间里，每一条走廊都有一个智能感应走廊灯，想知道它是怎么工作的嘛？让我们一起来一探究竟吧。

任务分析

角色	Abby
背景	走廊
实现效果	白天：有无人经过，走廊灯均不亮（背景走廊亮度为 0）。

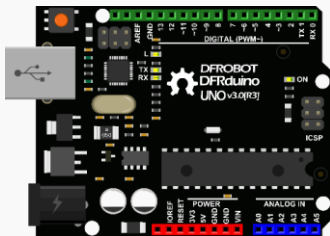
	晚上：有人经过，走廊灯亮（背景走廊亮度为 0）；无人经过，走廊灯不亮（背景走廊亮度为-200）。
--	--

知识要点

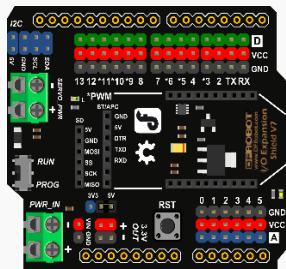
名称	电子元件	功能描述	
模拟环境光传感器		模拟环境光传感器，可以用来对环境光线的强度进行检测。通常用来制作随光线强度变化产生特殊效果的互动作品。	
	指令	功能描述	所属模块
		空白处条件 1 小于条件 2。	运算符

硬件连接

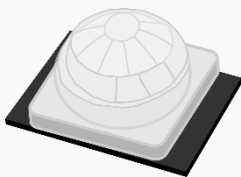
硬件清单



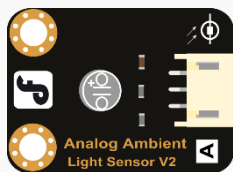
DFRduino UNO R3（以及配套 USB 数据线） X1



IO 传感器扩展板 V7.1 X1



人体红外热释电运动传感器 X1



模拟环境光传感器 X1

连接图

模拟环境光传感器连接在模拟引脚 A0 处，人体红外热释电运动传感器连接在数字引脚 3 处，UNO 板连接计算机。

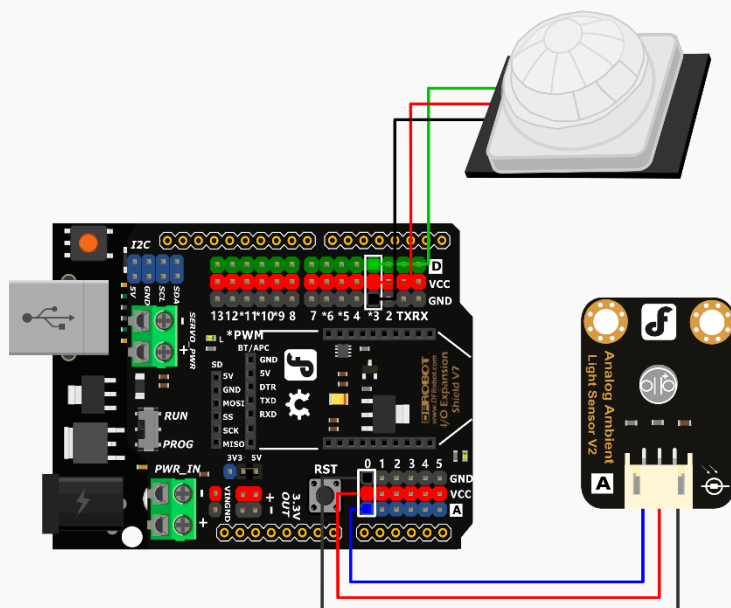


图 3.3.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“上传背景”，找到需要的背景图片。
3. 删除默认角色猫咪，添加角色 Abby。选择“角色库”——“艾比”。
4. 背景走廊效果

白天 (光线强度>10)：背景走廊亮度正常。晚上 (光线强度<10)：有人经过，背景走廊亮度正常；无人经过，背景走廊亮度变暗。流程图如图 3.4.2 所示。

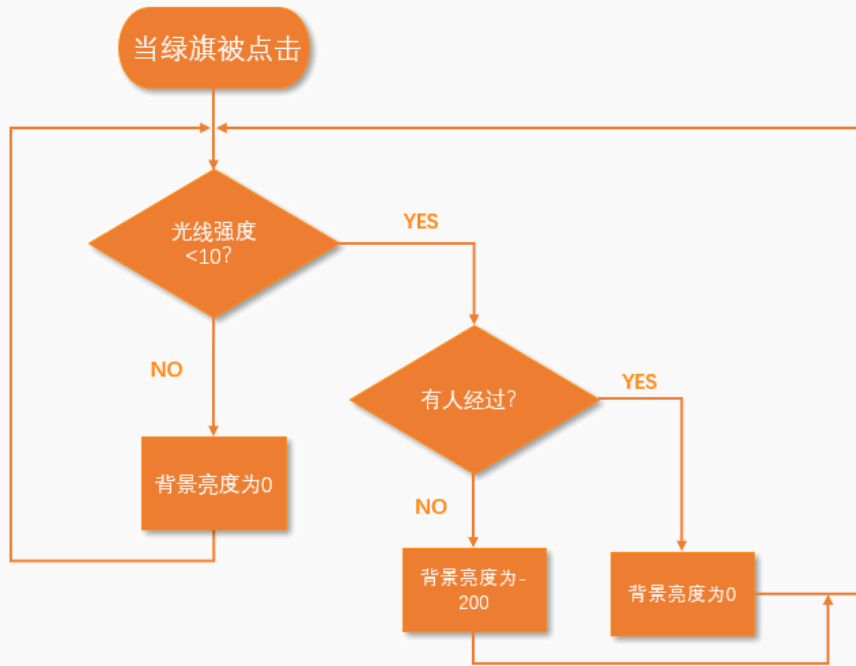


图 3.4.2 背景走廊效果流程图

根据流程图编写程序，背景走廊完整程序如图 3.4.3 所示。



图 3.4.3 背景走廊完整程序

5. 角色 Abby 效果

白天 (光线强度>10): Abby 说: “现在是白天哦!”。晚上 (光线强度<10): 有人经过, Abby 说: “灯亮啦!”; 无人经过, Abby 说: “现在是晚上哦!”。流程图如图 3.4.4 所示。

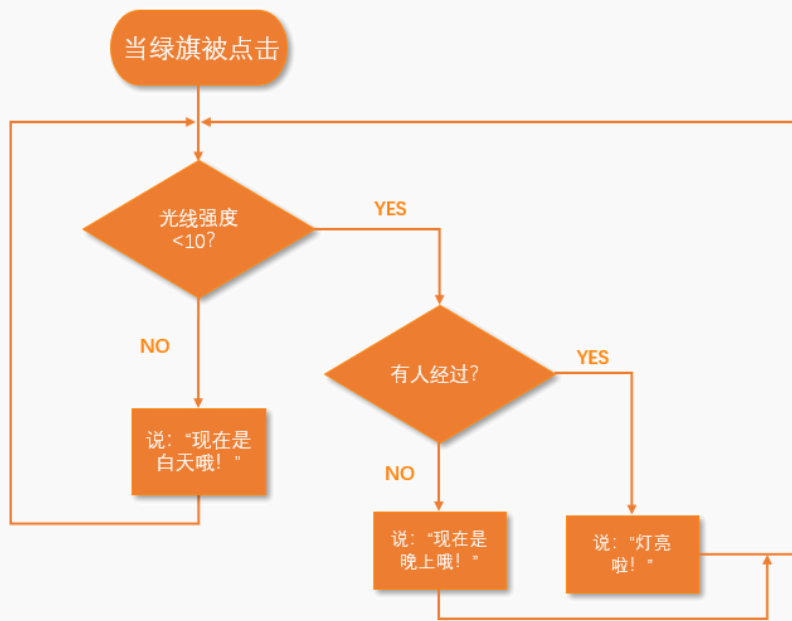


图 3.4.4 角色 Abby 效果流程图

根据流程图编写程序，角色 Abby 完整程序如图 3.4.5 所示。

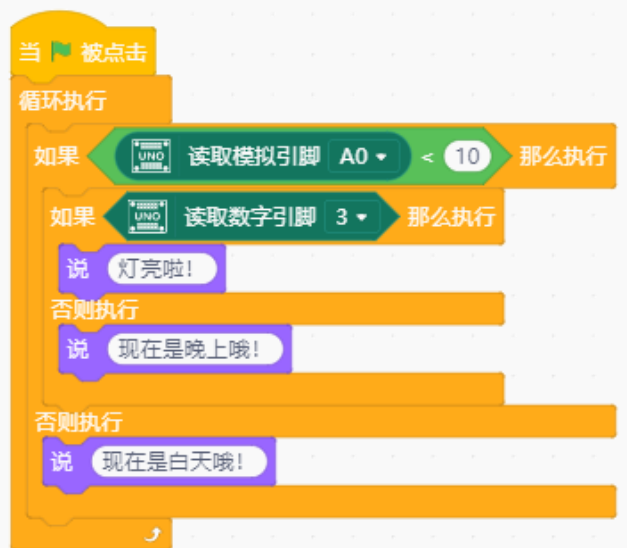


图 3.4.5 角色 Abby 完整程序

6. 保存项目，命名为“感应走廊灯”，连接设备，选择串口。效果图如下图所示。



图 3.4.6 光线强度>10 效果图



图 3.4.7 光线强度<10 且无人经过效果图



图 3.4.7 光线强度<10 且有人经过效果图

探究思考

尝试增加一个 LED 小灯，小灯被点亮的同时背景亮起。写下你的思路吧!

设计图纸

第四章 奇想部落

造梦空间的各种智能设备有没有惊艳到你呢？若是还没尽兴，那就到奇想部落来吧，这里有趣的互动游戏绝对不会让你失望。



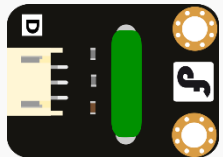

第一节 接香蕉 1.0

晚餐时间到了，让我们一起帮助小猴子接到香蕉吧，不然，它可要饿肚子啦。

任务分析

角色	猴子、香蕉
背景	蓝天
实现效果	香蕉从随机位置落下，使用数字钢球倾角传感器控制小猴子左右移动接香蕉，小猴子只有接到香蕉才不会饿肚子哦。

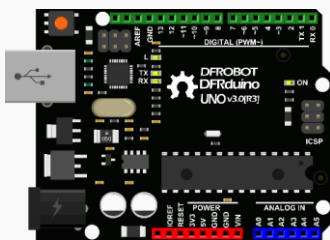
知识要点

名称	电子元件	功能描述
数字钢球倾角传感器		基于钢球开关的数字模块，利用钢球的特性，通过重力作用使钢球向低处滚动，从而使开关闭合或断开，因此也可以作为简单的倾角传感器使用。手持传感器徽标方向为  时，钢球左倾输出为“1”，钢球右倾输出为“0”。

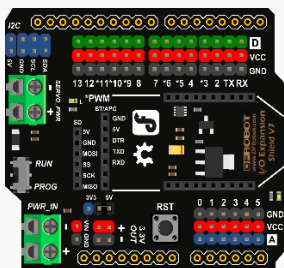
指令	功能描述	所属模块
	将 y 坐标增加 10。数字可更改。	动作
	角色面向 90 度方向，即面向舞台右侧。指向角度可通过键盘输入或者圆盘 选择。	动作
	在 1~10 范围内选取随机数。数字可更改。	运算符
	判断角色是否碰到某种颜色。可使用 吸管工具吸取颜色。	侦测
	判断角色是否碰到鼠标指针。其他可选项为：边缘和其他角色。	侦测

硬件连接

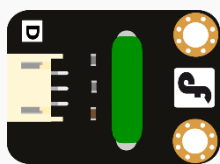
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字钢球倾角传感器 X1

连接图

数字钢球倾角传感器连接在数字引脚 7 处，UNO 板连接计算机。

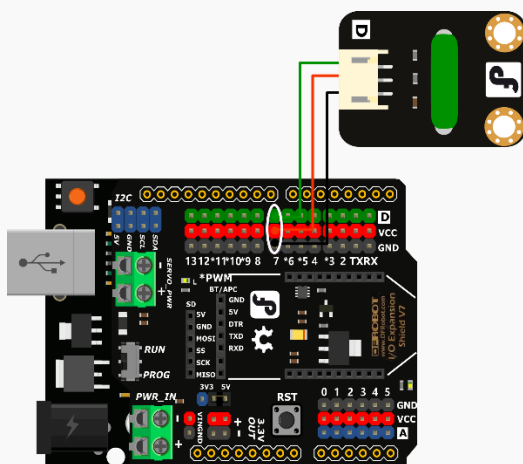


图 4.1.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“蓝天”。
3. 删除默认角色猫咪，添加角色猴子和香蕉。选择“角色库”——“猴子”和“香蕉”。
4. 角色香蕉效果：在舞台上随机位置出现香蕉，并逐渐下落。下落过程中，香蕉若被猴子接到则广播“接到香蕉”，香蕉若是落地则广播“没有接到”。流程图如下图 4.1.2 所示。

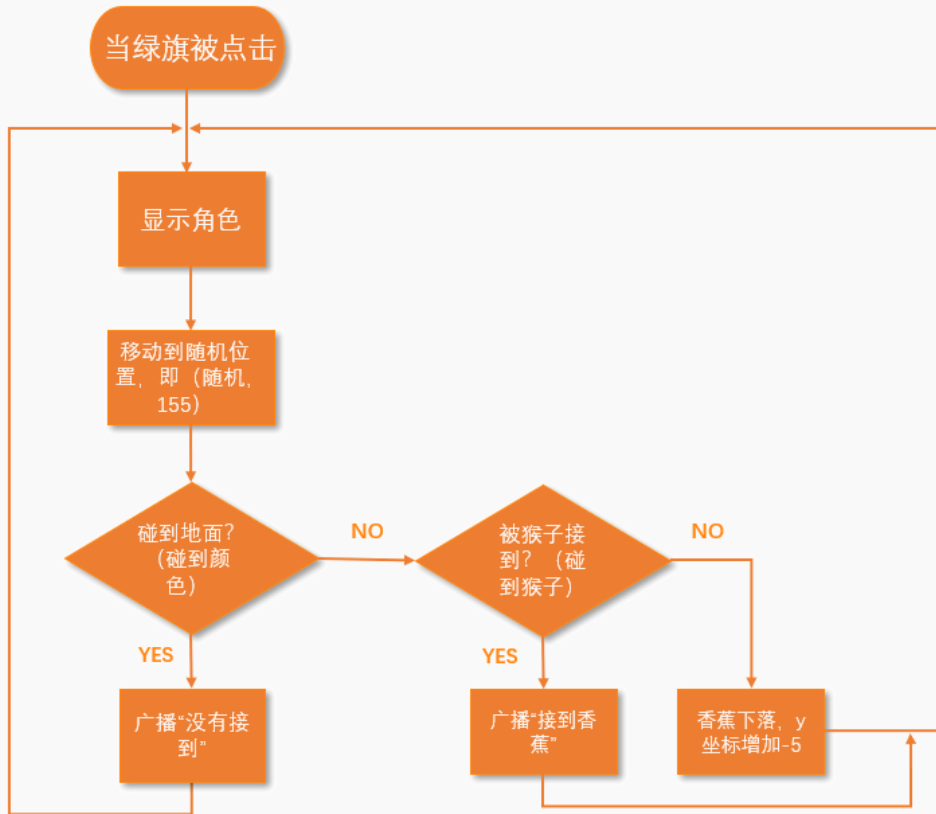


图 4.1.2 角色香蕉效果流程图

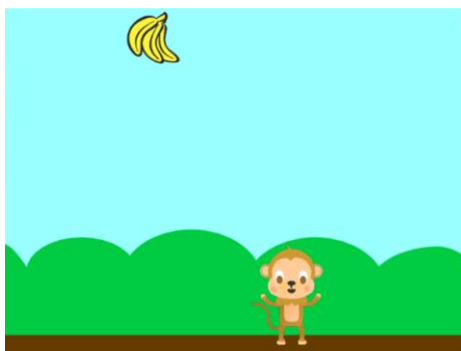
根据流程图编写程序，角色香蕉完整程序如图 4.1.3 所示。




图 4.1.3 角色香蕉完整程序

角色香蕉如何出现在随机位置?

要使角色香蕉出现在随机位置，需要设置 x 坐标和 y 坐标。可以设置 y 坐标固定（在舞台上移动角色至合适位置，确定 y 坐标），保证角色出现在同一条直线上，舞台上 x 坐标范围为-211~211，在这个范围随机选取值，即可实现角色香蕉出现在随机位置。



- 5. 角色猴子效果：当手持传感器徽标方向为  时，数字钢球倾角传感器左倾，猴子向左移动；数字钢球倾角传感器右倾，猴子向右移动。当猴子接到香蕉时说：“有香蕉吃啦！”；当香蕉落地时说“今天要饿肚子了”。流程图如下图 4.1.4 所示。

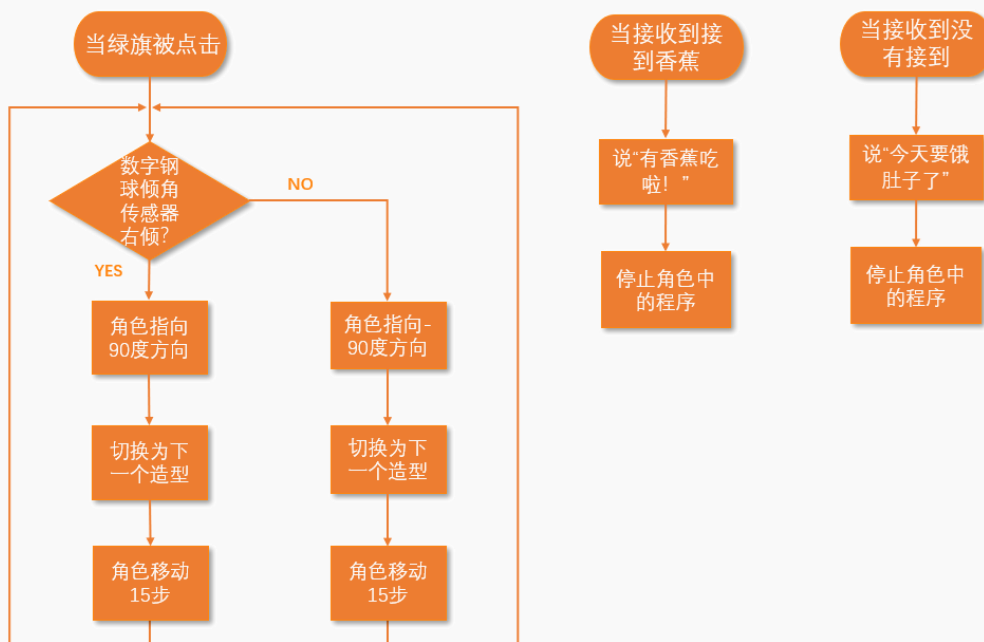


图 4.1.4 角色猴子效果流程图

根据流程图编写程序，角色猴子完整程序如图 4.1.5 所示。



图 4.1.5 角色猴子完整程序

6. 保存项目，命名为“接香蕉 1.0”，连接设备，选择串口。实现效果如下图所示



图 4.1.6 接到香蕉效果图



图 4.1.7 没有接到香蕉效果图

探究思考

尝试增加不同的水果，使猴子接到不同的水果有不同的反应吧，例如，猴子接到苹果说：“今天有苹果吃啦！”；猴子接到香蕉说：“今天有香蕉吃啦！”。写下你的思路吧！



第二节 接香蕉 2.0

接香蕉游戏 2.0 来啦！来和小伙伴比比吧，谁能接到更多的香蕉！

任务分析

角色	猴子、香蕉、GAME OVER
背景	蓝天
实现效果	香蕉不断从随机位置落下，使用数字钢球倾角传感器控制小猴子左右移动接香蕉，香蕉落地游戏结束。

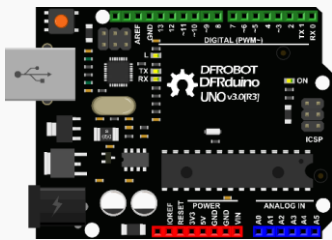
知识要点

指令	功能描述	所属模块
	当作为克隆体启动时，执行下面程序。克隆，即复制。	控制
	克隆当前角色或其他角色。	控制
	删除克隆体。	控制

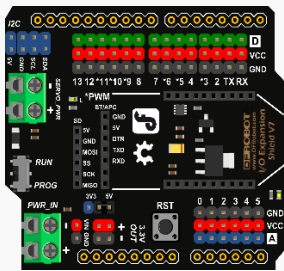
	<p>变量即可变的量。“my variable” 是变量名，可更改。</p> <p>选择 <input checked="" type="checkbox"/> 在舞台中出现 。</p>	<p>变量</p>
	<p>设置变量值为 0，数字可更改。</p>	<p>变量</p>
	<p>将变量值增加 1，数字可更改。</p>	<p>变量</p>
	<p>在舞台区显示/隐藏 。</p>	<p>变量</p>

硬件连接

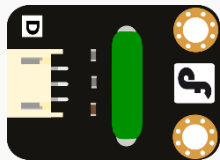
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字钢球倾角传感器 X1

连接图

数字钢球倾角传感器连接在数字引脚 7 处，UNO 板连接计算机。

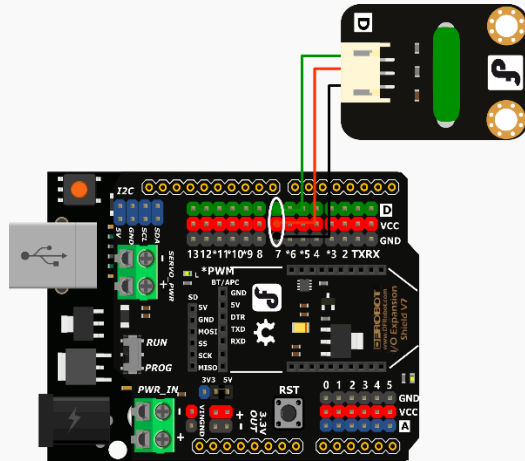


图 4.2.1 硬件连接图

动手实践



1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“蓝天”。
3. 删除默认角色猫咪，添加角色猴子和香蕉。选择“角色库”——“猴子”和“香蕉”。
4. 绘制角色“GAME OVER”。选择“画笔”下的  文字工具，在画布中输入“GAME OVER”，填充为 ，无边框，调整大小并放置在舞台中。如图 4.2.2 所示。



图 4.2.2 角色 GAME OVER

5. 角色香蕉效果：通过克隆，在舞台上随机位置不断出现香蕉，并逐渐下落。下落过程中，香蕉若被

猴子接到，香蕉个数增加 1，香蕉若是落地则广播“没有接到”。流程图如下图 4.2.3 所示。

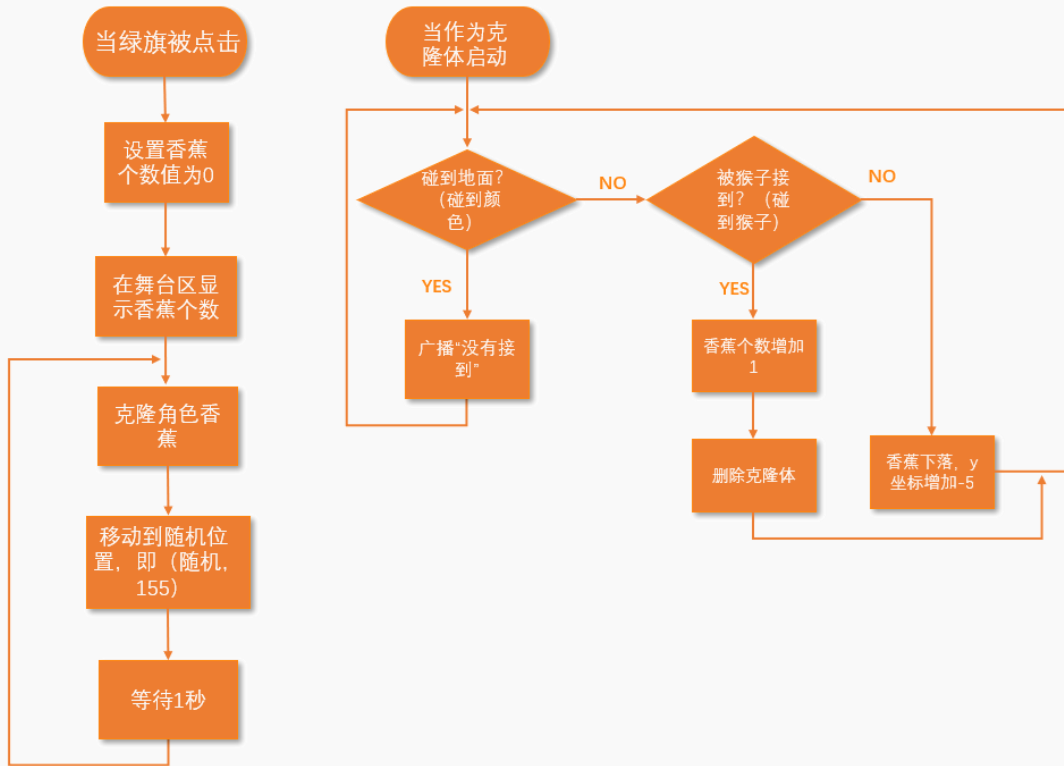


图 4.2.3 角色香蕉效果流程图

根据流程图编写程序，角色香蕉完整程序如图 4.2.4 所示。



图 4.2.4 角色香蕉完整程序

小贴士

如何新建变量？

在指令区**变量模块**下选择 **新建变量...**，输入变量名称，点击确定即可。

如何重命名变量和删除变量？

在指令区**变量模块**下右击 **变量 my variable** 指令，选择“重命名变量”和“删除变量”。



- 6. 角色猴子效果：当手持传感器徽标方向为 **f** 时，数字钢球倾角传感器左倾，猴子向左移动；数字钢球倾角传感器右倾，猴子向右移动。流程图如下图 4.2.5 所示。

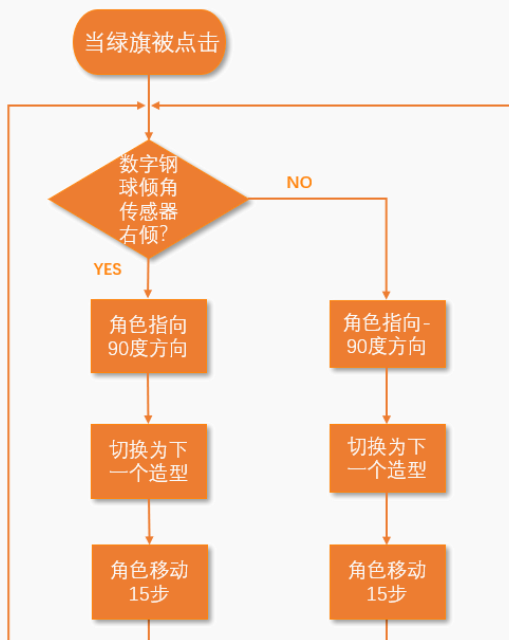


图 4.2.5 角色猴子效果流程图

根据流程图编写程序，角色猴子完整程序如图 4.2.6 所示。



图 4.2.6 角色猴子完整程序

- 7. 角色 GAME OVER 效果：游戏进行中不显示，当香蕉落地，接收到广播“没有接到”时，显示。完整程序如下图 4.2.7 所示。



图 4.2.7 角色 GAME OVER 完整程序

- 8. 保存项目，命名为“接香蕉 2.0”，连接设备，选择串口。实现效果如下图所示

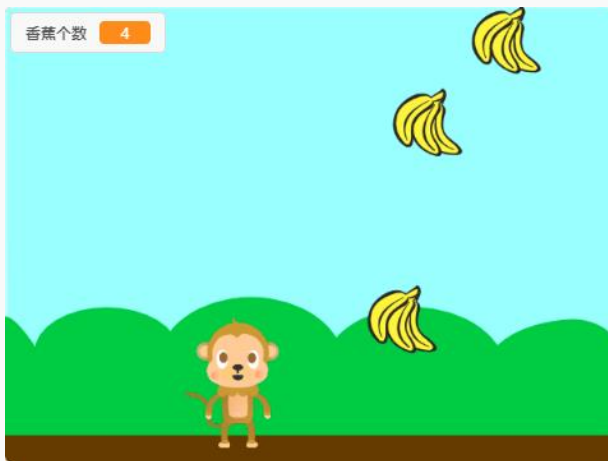


图 4.2.8 接香蕉 2.0 效果图

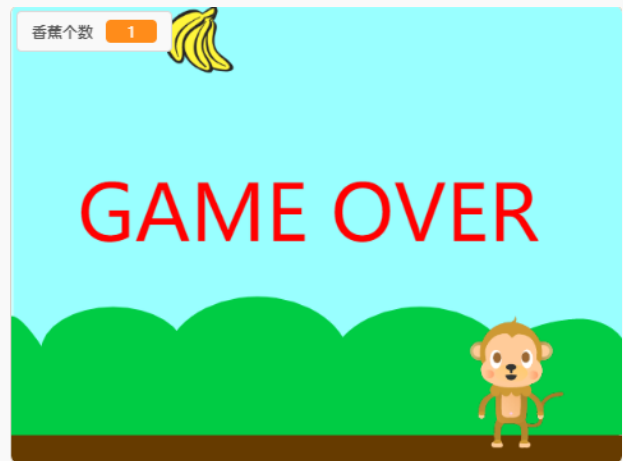


图 4.2.9 游戏结束效果图

探究思考

尝试着将游戏规则改成：10 秒钟之内谁接的香蕉多，谁就是赢家。写下你的思路吧！



第三节 吹气球

接香蕉游戏不符合你的胃口，那快来挑战吹气球游戏吧，不要以为只是把气球吹大哦！

任务分析

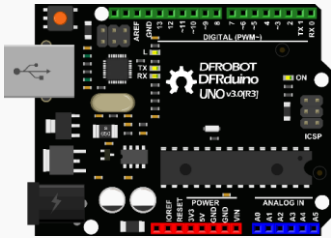
角色	男巫 1、气球、YOU LOSE 和 YOU WIN
背景	蓝天
实现效果	通过吹声音传感器使气球升高，坚持 10 秒，游戏胜利。若是不吹，气球则缓慢下落，落到地上气球破碎，游戏失败。

知识要点

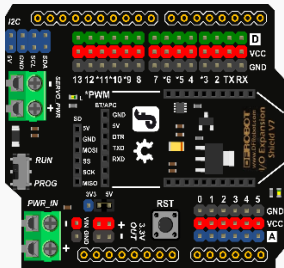
指令	功能描述	所属模块
	切换角色造型为造型 1。	外观
	计时器用来计算时间。选择 <input checked="" type="checkbox"/> 在舞台中出现 	侦测
	计时器归零。	侦测

硬件连接

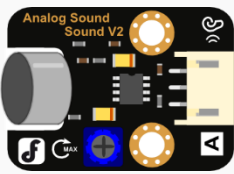
硬件清单



DFRduino UNO R3 (以及配套 USB 数据线) X1



IO 传感器扩展板 V7.1 X1



数字钢球倾角传感器 X1

连接图

模拟声音传感器连接在模拟引脚 A0 处，UNO 板连接计算机。

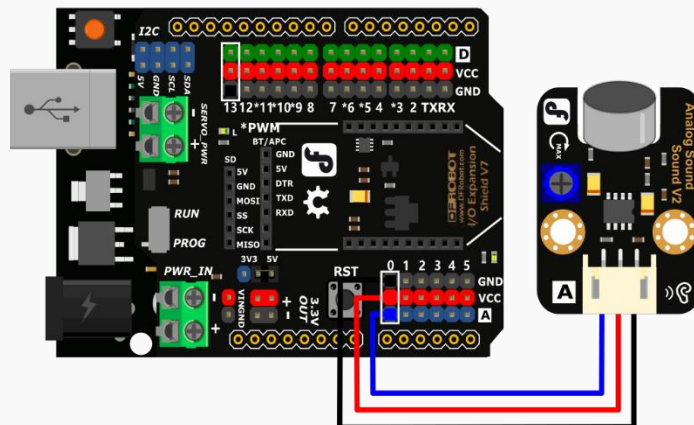


图 4.3.1 硬件连接图

动手实践

1. 打开 Mind+软件，新建一个项目。
2. 添加舞台背景。选择“背景库”——“蓝天”。
3. 删除默认角色猫咪，添加角色男巫和气球。选择“角色库”——“男巫 1”和“气球”。
4. 添加气球落地破碎造型。在角色区选择角色气球，在菜单栏中选择“外观”，删除“气球-蓝”和“气球-黄”两个造型，重命名“气球-紫”造型为“气球 1”，复制“气球-紫”造型，命名为“气球 2”。


使用  变形工具拉伸“气球 2”，使其呈现气球破碎的状态，如图 4.3.2 所示。



图 4.3.2 气球 2 造型



5. 绘制角色“YOU WIN”和“YOU LOSE”。选择“画笔”下的  文字工具，在画布中输入“YOU WIN”，填充为 ，无边框，调整大小并放置在舞台中。按上述方法绘制角色“YOU LOSE”。效果如图 4.3.3 和 4.3.4 所示。



图 4.3.3 角色 YOU WIN

图 4.3.4 角色 YOU LOSE

- 6. 角色男巫 1 效果：游戏开始前男巫 1 说：“一起来玩吹气球吧！”“小心不要掉在地上啦！”，广播“吹气球”游戏开始。程序如下图 4.3.5 所示。

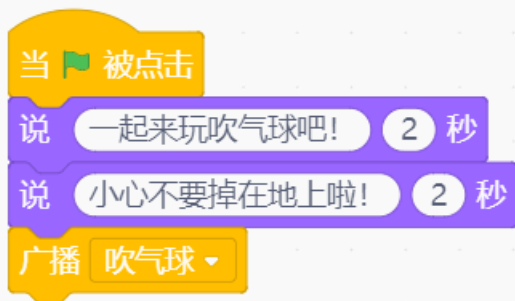


图 4.3.5 角色男巫 1 完整程序

- 7. 角色气球效果：游戏开始，计时器开始计时，声音传感器接收到的声音大于临界值 50 时，气球上升，否则下落。若气球落地破碎，游戏失败；若保持气球不落地 10 秒，游戏胜利。流程图如下图 4.3.6 所示。

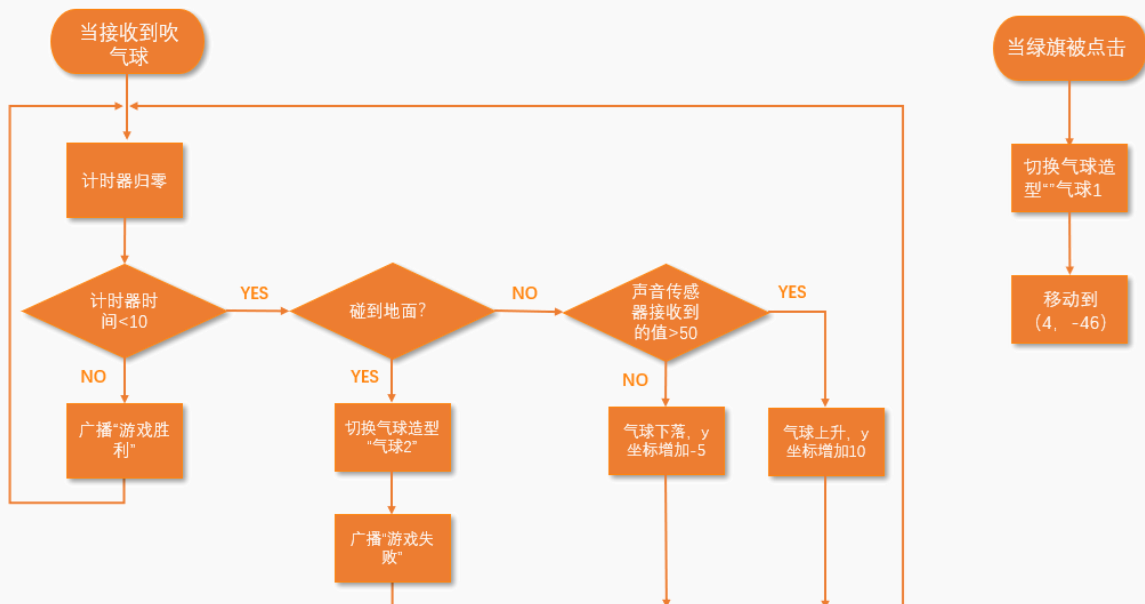


图 4.3.6 角色气球效果流程图

根据流程图编写程序，角色气球完整程序如图 4.3.7 所示。



图 4.3.7 角色气球完整程序

- 8. 角色 YOU LOSE 效果：游戏进行中不显示，当 10 秒之内气球落地，接收到广播“游戏失败”时，显示。完整程序如下图 4.3.8 所示。



图 4.3.8 角色 YOU LOSE 完整程序

- 9. 角色 YOU WIN 效果：游戏进行中不显示，当气球保持 10 秒不落地，接收到广播“游戏胜利”时，显示。完整程序如下图 4.3.9 所示。



图 4.3.9 角色 YOU WIN 完整程序

- 10. 保存项目，命名为“吹气球”，连接设备，选择串口。实现效果如下图所示

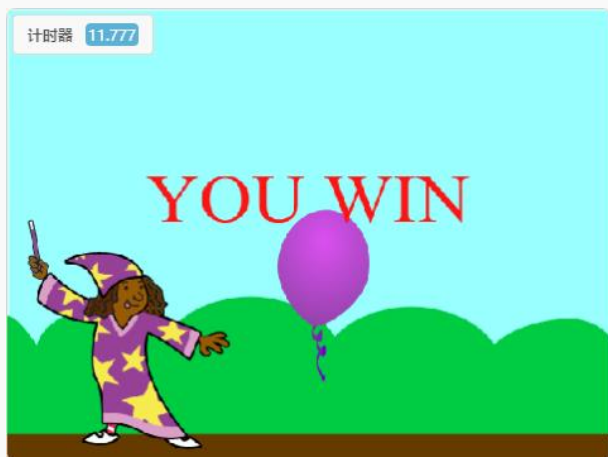


图 4.3.10 游戏胜利效果图

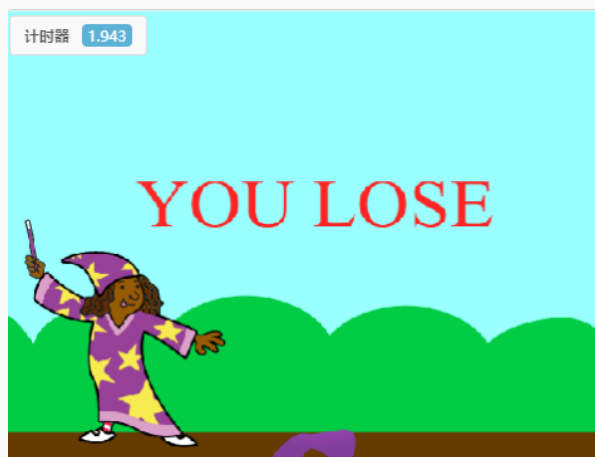


图 4.3.11 游戏失败效果图

探究思考

尝试在此基础上增加若气球飞出舞台则游戏失败。写下你的思路吧！

设计图纸

希望你的机器人探索兴趣一直延续，用你的奇思妙想，玩出更多新颖有创意的作品。如果你愿意与我们分享的话，也可以直接登陆我们的论坛，让我们的社区论坛记录下你的点点滴滴！

欢迎登陆 DFRobot 创客社区!

DFRobot 创客社区: www.dfrobot.com.cn