





끞
<u>></u> к

第一课你好,行空板	2
第二课 旅游打卡路牌	19
第三课 西游记舞台剧	31
第四课 情绪卡片	49
第五课 密室逃脱游戏	62
第六课 实景星空	
第七课 名画互动博物馆	90
第八课 智慧钢琴	
第九课 健身打卡追踪器	121
第十课 肺活量测量仪	134
第十一课 loT 数据助手	151
第十二课 肺活量数据可视化报告	172
第十三课 loT 课堂互动答题系统	192
第十四课 语音翻译机	214
第十五课 天气助手	

第一课 你好, 行空板

拿起行空板,它的正面有一块屏幕,可以用来显示文字、图表、视频等,实现丰富的视觉互动效果。 这节课,就让我们一起来用屏幕显示多彩的文字,开启行空板的图形化学习之旅吧!



任务目标

在行空板屏幕上显示多彩的中英文。



<u>知识点</u>

- 1. 掌握在行空板上显示文字的方法
- 2. 理解行空板屏幕坐标知识

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



软件基础使用

如何在行空板屏幕上显示文字呢,那就需要编写程序啦!这里我们使用 Mind+编程软件。Mind+是 一款集成了包括行空板在内的大量开源硬件的编程软件,它拥有非常多的图形化基础指令和扩展库,可以 让大家轻松体验创造的乐趣。下面我们就基于 Mind+来学习基础的软件操作,为接下来的项目任务做好准 备工作。

1. 打开 Mind+

双击启动 mind+软件 👼 , 看到如下画面:

🚔 Mind+	V1.7.2 RC3.0	Eda:					- 🗆 X
ist the						ST HIMS SHOW I	LANGS Princette 🙆 🔮
植物						1 上传到设备	🙆 (Gille) 🖉 ardulno C 👻
a	1284	草			自动生成。手动偏振		
0 12701	SH D D				1 * /*! 2 * MindPlus 3 * noBoard		
● 出時行					4 5 1/ 6		
	minutery 10 ct				8 // 1 HDP H 20 9 - void setup() {		
通数					10 11 } 12 - void loop() {		
					13 14 15		
				2	_		
	NUT CONTRACT			C.			<u>(</u>)
	-			1			
	tille Misseli			0			
	-			Q			
63				Ξ			
			5		ф 💉 🙆		发送

点击右上角的"python 模式",

)意见反馈	实时模式	上传模式	Python模式	6
_	▶运行	O ft) Oxf	4.A
文件目录上	ī 击此处	选择编	同程模式	,

接下来点击左上角选择编程方式, "模块"表示图形化编程, "代码"表示 python 代码编程, 这里选择"模块",使用图形化指令完成我们的项目。

-	p.
👑 Mind+ \	/1.7.2 RC3.0 新建项目-2022/8/
Mind	₽ 項目 ● 教程 ●
模块	代码法探图形化
a	■ 编程方式
•	1945 1 et

进入之后会看到下面这样的画面,这就是主要的编程操作界面。

d Mind+ V1.7.2 RC3.0 新建项目			– 🗆 X
and the set a set	- 菜单栏	快捷工具区 ***** ****	Camerica 🎱 🗘
Re10 (CBS			(@:?####
(A) #22446 19 × 및		district of	
22 12 12 12 12 12 12 12 12 12 12 12 12 1		1 # -*- coding: UTF-8 -*-	
		2 3 # MindPlus	
	Python±8899788	a Python 自动生成	
	MADAN'S	· Python代征	9
X4.		8 while True: 9 pass	
	1	10	
CO ROMO	75377(1277)	l	
	现现1749	⊘ 終端	(1) 消除地出
	0	Python 3.6.5 [MSC v.1900 64 bit AMD64)]] on win32
Real Contraction of the second	1	Type "help", "copyright", "credits" or " more information.	'license" for
		>2	
Han .	脚本区	终端区	
	Q		
	Ξ		
一 展库			

注意:关于 Mind+界面中每个区域的详细介绍,参见本课"知识园地"部分。

2. 加载行空板

基于前面的操作,只需要点击"扩展库",找到"官方库"下的"行空板"模块点击完成添加,点击返回后,就可以在"指令区"找到行空板,完成行空板加载。



需要注意的是行空板加载完成后,原来的菜单栏会多出一个用来连接行空板的选项——"连接远程终端"。

Mind+		(程 -	连接远程终端,	
模块	代码			

3. 连接行空板

首先, 你需要通过 USB 连接线将行空板连接到计算机;



然后,等待并确认行空板处于连接且开机状态(如下图),开始远程连接行空板;



最后,点击连接远程终端后,点击 10.1.2.3 连接行空板。

-	至接远程终端 -	点	击	此	处·i	车	妾	ΞĒ	≥村	Ā
	10.1.2.3									
	手动输入 打开网络中心 帮助文档									

注意: "10.1.2.3"为 USB 直连时的固定 IP 地址。

连接成功后会弹出提示,终端会显示行空板即表示连接成功,并检测行空板上的依赖库版本,等待检测完成即可。

	Diana C	(e 20-00)
● 提示 已wine#\$/12123	0 # -*- coding: UTF-8 -*-	
● 提示 出於行空经上的问题则但工中,诸别性。	W MindPlus W Python	
		~
0	 // 按編 (行空版) >>> 已成功连接到10.1.2.3 	(a) 139981

连接成功后,点击运行 🕒 🔤 , Mind+会将 Python 代码发送到行空板上运行,你就可以在行空板

上看到代码运行效果了。

注意: 连接不成功可以参考下面解决办法。

(1)点击"连接远程终端",选择"打开网络中心"。(2)在"网络连接"窗口中找到 Remote NDISCompatible Devi..,检查连接状况。

(3)若显示"正在识别",等待一会重新连接。
(4)若没有找到"Remote NDIS Compatible
Devi..",检查行空板屏幕是否显示行空板 logo;
或者检查 USB 线是否正常插入电脑 USB 口 (不要使用扩展坞或延长线)

● 网络连接	连接远程终端
(2) 出现 "网络连接"窗口	(1) 点击
← → ~ ↑	10.1.2.3
组织 • WLAN	- 手动输入
DFRobot-office	打开网络中心
Intel(R) Wi-Fi 6 AX201 160MHz	帮助文档
以太网 2 正在识别… (3)观察连拐 Remote NDIS Compatible Devi	情况

如果还是无法解决可进入行空板官网(https://wiki.unihiker.com)查看"FAQ常见问题"。

4. 保存文件

点击"菜单栏"里的"项目",并在出现的下拉菜单中点击"保存项目"。



然后在弹出的界面中选择保存位置,输入你自己的文件名,保存类型"Mind+",点击保存即可。

751575			(1)选择保存位置	.0
$\in \neg$	Ť	> 此电脑 > 文档 >	✓ ひ 𝒫 捜索"文档"	
Г	文件名()	y): 001、初识行空板		-
		The minut ((t shall)		

动手实践

接下来一起动手完成今天的课程任务吧。我们将从第一个行空板指令开始,逐步学习如何在行空板屏 幕上显示多个彩色的文字。

任务一:运行第一条指令

首先我们将学习显示文字的方法,并在行空板屏幕上显示预设文字"行空板"。

任务二:显示有颜色的"你"

在学习完显示文字方法后,进一步加深对于文字显示指令中坐标和颜色的应用,进而在行空板屏幕上 显示红色的"你"字。

任务三:显示不同颜色的一串文字

通过对于文字显示指令中坐标的修改,学习如何调整测试多个文字的距离,在行空板屏幕中间显示一 串彩色文字。

任务一:运行第一条指令

1. 硬件搭建

确认使用 USB 连接线将行空板连接到计算机。



2. 软件准备

确认已打开 mind+,在 Python 图形化模式下出现预设代码,成功加载行空板。接下来就可以编写项

目程序了。

Mind+	V1.7.2 RC3.0			- 0	X
Min	flg,项目 → 数程 → 10.1.2.3 →		② 意识反馈 same.	t 1-68080 Pythontifet	۵ ک
模块	代語 (5)名	行空板连接成功			3
控制	母亲 显示	示行空板IP	自动生地		
 ご算符 近算符 数学 算量 文本 副 引表 元祖 (6) 	 ・ 新展展示 (2)切換到图形化编程模式 	Python主题注册:	1 # -*- coding: UTF-8 3 # MindPlus 4 # Python 5 6 7 8 while True: 9 pass 10)切换至Pythor	模式
字典	XIERE RUMBER . IEE. EX	ANAGAN	授助 (行空板)	🔊 清晰	输出
第合	20 x12x5 minute 120 x 11		>>> 已成功连接到10.1.2	.3	
安量 函数 Python 行空板	 Nach <l< td=""><td>(3)出现预设代码 来 ②</td><td>Last login: Mon Sep 5 10.1.2.121 root@unihiker:~# root@unihiker:~# root@unihiker:~#_</td><td>17:58:41 2022 f</td><td>rom</td></l<>	(3)出现预设代码 来 ②	Last login: Mon Sep 5 10.1.2.121 root@unihiker:~# root@unihiker:~# root@unihiker:~#_	17:58:41 2022 f	rom

3. 编写程序

我们可以先在行空板分类下找到 对象名 显示文字"行空板"在 X0Y0 字号 20 颜色蓝 指令,然后将 它拖拽,放在预设代码 python 主程序开始 的下面。



完整代码如下图:

Python主程序	研始	1	2	1	1		2	8	20		2		1
◎ 対象名	显示	文字	• 行	空板 "	在	x 🕡) v	0	字号	20	颜色	•	
循环执行	1.0			~	9				100	282	e.	0	

4. 程序运行

STEP1: 检查并保证行空板已连接 Mind+, 即在"菜单栏"部分显示 IP;



STEP2: 点击界面右上方的运行按钮;



STEP3: 观察行空板屏幕显示效果(参考下图)。



任务二:显示有颜色的"你"

1. 编写程序

通过上一个任务中"行空板"字样的显示,可以知道指令对象名显示文字"行空板"在 XOYO 字号 20颜色蓝的功能是:显示蓝色 20 号"行空板"文字内容在屏幕左上角,也就是坐标原点(0,0)。将显示内容修改为"你",点击蓝色颜色框调整颜色,在出现的颜色选择框中选择红色即可。



完整代码如下:



2. 程序运行

运行并观察代码效果:



3. 试一试

现在你已经知道如何使用指令去显示文字了,不妨试试调整坐标和字号,在行空板屏幕中间显示一个 比较大的"你"字。



任务三:显示不同颜色的一串文字

1. 编写程序

(1) 确定文字颜色数量

由于最终要显示的文字是"你好,行空板"和"Hello, UNIHIKER!",每个字和标点符号都有不同的 颜色,不能只使用一个指令,在本项目中一共有9个颜色,就需要9个指令,可以右键点击指令,选择复制,完成多个指令的搭建。



(2) 调试字之间的位置

每个汉字或符号有不同的位置,在调试字之间的位置时,你可以不必频繁观察行空板屏幕显示情况, 直接使用断开连接后的 Python 运行弹窗快速进行坐标调试。如何操作呢?

a) 在"菜单栏"点击行空板 IP,选择下拉菜单中的"断开连接",



b) 直接点击运行,程序会直接在电脑上运行,弹出一个和行空板屏幕一样大的窗口。



注意:关于弹窗的更多问题,参见本课"知识园地"部分。

接下来我们就可以利用这个弹窗完成字之间距离的调试。多个汉字间的距离,一般只需要确定横向或 者纵向两个汉字或符号的坐标,记住它们的坐标差就可以通过计算获得所有汉字或符号的位置。

以"你好"字样显示为例,设置字号都为20,"你"字的位置在X:30Y:80,"好"字在右侧,保证 Y坐标不变,X坐标向右增加,尝试调整几个数值找到合适的坐标(调试过程参照下图)。



上图中最后为合适的坐标情况, "你" "好"两字 X 坐标的差值为 30, 那么后面汉字或符号的 X 坐标依次增加 30, Y 坐标不变即可完成。完整示例程序如下:



- 2. 程序运行
- STEP1: 连接行空板
- STEP2: 点击运行
- STEP3: 观察效果

观察行空板,可以看到在屏幕上显示"你好,行空板!"以及"Hello, UNIHIKER!"的字样。



知识园地

1. Mind+相关界面介绍

前面我们已经了解过 Mind+与行空板相关的操作界面,接下来详细学习一下各个区域的功能。

Mind+ V1.7.2 RC3.0		- 🗆 X
₩Minds 项目• ## · i##in#### · 莱单栏	快捷工具区(2	Nedit showed lawer pythonest 🔮 🧿
機快 代码		> 福行 (の代制区) (の文件系统)
	1 # -*- coding: UTF-8 -*- 3 # MindPlus 4 # Python 5 日力生成 6 Python代码 9 pass 10 Python 3.6.5 [MSC v.1900 64 bit AMD64)] on vin32 Type "help", "copyright", "cr edits" or "license" for more information. >>> _ 经端区	文件目录 ○ □ □ □ ■ <

Mind+与行空板相关的操作界面功能介绍

区域名称	功能
菜单栏	包含项目文件操作选项、教程示例和连接设备选项(连接行空板后出现)
指令区	放置所有图形化指令和已加载扩展库中的图形化指令
扩展库	除基础功能外的扩展功能库
脚本区	图形化编程区域,指令需要连接到 Python 主程序开始 或其他帽子型指令后 才会被执行 (帽子型指令是指形如 Python 主程序开始 的指令,只能作为第 一个指令执行的指令,例如下图)
快捷工具区	包含代码运行按钮,开启或关闭"代码区"和"文件系统"的区域
自动生成 Python 代码	点击快捷工具区里的"代码区"(如图 < (代码区) 开启, 会显示自动生成
	图形化代码对应的 Python 代码;再次点击 2018 关闭
终端区	点击快捷工具区里的"代码区"(如图 < (代码区) 开启,显示代码运行的
	输出信息、错误信息等;再次点击 《《四》关闭
立件日之区	点击快捷工具区里的"文件系统"(如图 (@ 文件系统))开启,显示项目中的
XITHWC	文件和已添加的计算机文件;再次点击 ② 24556 关闭

2. 电脑端运行 Python 程序

当行空板正常连接电脑时,点击运行,代码在行空板上完成运行,显示效果也会呈现在行空板屏幕上; 但是当行空板和电脑断开连接,点击运行,代码则是在电脑端完成运行的,此时显示效果会出现在一个弹 窗上。这个弹窗的大小和行空板一致,它可以很方便的帮助你检查或调试自己的行空板代码。

🖋 tk	1		×
你好	行	空机	反!
Hello,	UNI	HIKE	ERI

关于弹窗的操作:

(1) 弹出的窗口是可以最小化和关闭的,但是这个窗口是不允许改变大小的,所以你可以看到最大化

的按钮是灰色的。



(2) 关闭弹出窗口的方式有两种:直接点击窗口上的关闭 "X"或者点击停止运行。



(3) 在开始学习的时候,如果你遇到点击运行看不到这个弹窗,有可能是因为下面三种情况:

a) 代码中未使用行空板"屏幕显示"、"鼠标键盘"以及"多线程"分类下的指令,窗口不会 弹出;

行空板		
日本日二	▶ 鼠标键盘事件	▶ 多线程
▶ 肝福业小		

b)代码运行出现问题,这会显示在终端区,你需要根据报错信息,排查错误原因;



c)运行完代码后,窗口没有停靠在电脑上,这是因为代码运行结束后 Mind+会认为没有需要执行的任务,自动结束窗口运行,关闭窗口,此时你只需要在代码的后面加上一个"循环执行", 让运行完代码后,窗口一直停靠在电脑上。



3. 行空板屏幕坐标

行空板屏幕分辨率为 240 x 320,也就是说行空板宽 240 像素,高 320 像素。可以用坐标系来描述整 个行空板位置,坐标原点为屏幕左上角,向右为 x 轴正方向,向下为 y 轴正方向。为了便于使文字等屏幕 元素能够显示在指定位置,编程时可以通过设定 x 和 y 的坐标来实现。



4. 指令学习

本项目主要使用了文字显示指令,以及图形化 python 代码预设指令。



挑战自我

- 1. 你见过"颜文字"吗, 试一试用行空板绘制一个表情, 表示你当前的心情吧!
- 2. 除了上述内容,你还有什么话想向行空板说呢,也将它添加到屏幕上吧!



第二课 旅游打卡路牌

在旅途中拍照打卡一处风景,用一个文艺的路牌为景色增添浪漫气息,同时也能留下一段有趣的影像 与朋友分享。但不是所有地方都存在有趣的路牌,本节课我们就用行空板来制作一个旅游打卡路牌,只要 通上电就可以用它来完成一次独一无二的旅游打卡啦。



任务目标

在行空板屏幕上显示打卡路牌。



<u>知识点</u>

- 1. 认识程序中的顺序结构
- 2. 掌握文字坐标设置方法
- 3. 掌握在行空板上显示矩形和数码管数字的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

接下来一起动手在行空板上显示旅游打卡路牌,我们将分为路牌外部轮廓显示和路牌文字细节显示两 个任务。

任务一:显示路牌的轮廓

在此任务中,我们通过学习行空板显示矩形的方法,结合行空板屏幕坐标和路牌轮廓形状的组成结构, 完成路牌的轮廓的显示。

任务二:显示路牌的文字细节

在完成路牌轮廓的显示后,我们将学习如何计算文字坐标和数码管数字的显示方法,完成路牌的上下 两部分文字显示。

任务一:显示路牌的轮廓

1. 硬件搭建

确认使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+,选择 Python 图形化编程模式,点击"扩展库"加载行空板,然后连接行空板,等待出现以下画面。然后,保存好当前项目,就可以开始编写项目程序了。



3. 编写程序

(1) 观察轮廓构成

为了更清楚地知道轮廓的组成,需要对原路牌进行拆分(如下图),可以拆分为橙色圆角矩形、白色 实心矩形、空心矩形以及黑色实心矩形。



根据拆分情况,在指令区"行空板"分类下的"屏幕显示"里寻找含有"矩形"关键字的指令,拖出 对应类型的矩形指令: 对象名显示矩形在 X0Y0 高 100 宽 200 线宽 1 边框颜色蓝、显示填充矩形在 X0Y0 高 100 宽 200 填充颜色蓝 (2个)、显示圆角填充矩形在 X0Y0 高 100 宽 200 圆角半径 5 填充颜 色蓝。



注意: 行空板 "屏幕显示" 分类下的指令有很多,寻找的时候不是很方便,Mind+提供了一个快捷查 找指令的工具。它在指令区的最上面,具体使用方法参考下图:



找到指令后,按下图顺序搭好并放在预设代码 python 主程序开始 的下面。



(2) 路牌轮廓尺寸和坐标设置

基于上一步的代码,现在要根据路牌轮廓上矩形的位置关系,填写各矩形的宽、高以及坐标。 **注意**:矩形的坐标默认是在矩形的左上角位置。



路牌轮廓尺寸方面,由于路牌是出现在行空板屏幕的正中间,需要根据行空板屏幕的尺寸 (240x320) 来设置。我们可以先设定整个路牌略小于行空板尺寸,比如,路牌整体宽为 210 像素,高为 280 像素。至 于轮廓中各矩形宽高的设置,根据路牌内各矩形的的位置关系设置即可,下图展示了一个路牌尺寸设置示 例以供参考。需要说明的是圆角矩形下半部分圆角被白色矩形遮挡,所以它的的实际高度要比看到的高度 数值大一点。



确定了路牌的尺寸以后,现在只需要结合路牌轮廓的尺寸以及各矩形的宽和高,计算矩形坐标即可, 以橙色圆角矩形坐标(**x**,**y**)为例:



然后根据计算结果填写橙色圆角矩形显示指令的具体内容。

接下来你可以参考橙色圆角矩形的坐标计算方法,分别计算白色实心矩形、空心矩形以及黑色实心矩形的坐标,需要说明的是白色实心矩形和黑色空心矩形的宽度、高度和坐标都是一样的。完整代码如下图:



4. 程序运行

点击运行,观察行空板屏幕显示效果(如下图)。



任务二:显示路牌的文字细节

1. 编写程序

通过上一个任务,路牌的轮廓已经显示完成,接下来就可以通过轮廓中矩形和文字的位置关系设置显示文字坐标。你可以断开行空板,使用 Python 运行弹窗对文字坐标进行调试。

(1) 白色文字部分显示

先来完成路牌上部白色文字显示,字号设置为 15。"我在 DF 星球等你"八个字,出现在第一个矩形内,也就是在上文提到的点 A(15,20)右下方,所以这些字显示的 x 坐标要大于 15, y 坐标要大于 20。



y 坐标向下移动 10 像素即设为 30, x 坐标需要根据字号和像素的对照关系计算。 注意:关于汉字字号和像素关系的详细介绍,参见"附录一"。

白色文字中有汉字 6 个,字母 2 个。汉字所占宽度相同,查询可知 15 号汉字占 20 个像素,字母的宽 度约为汉字的一半,那么白色文字所占宽度为 140 个像素,计算可得 x=(240-140)/2=50。

注意: 文字的坐标默认是指文字左上角的位置。



那么,填写好的指令如下:



(2) 黑色文字部分显示

白色矩形里的黑色文字可以使用类似的计算方法,不同的是有一个特殊的数字显示。y坐标依然比 B(15,60)点 y 坐标大 10,即 y 坐标为 70。



"行空板第2站"一共有6个字,根据文字特点可以分成三段来显示:"行空板第"、"2"和"站"。 第一段文字坐标的计算要从整段文字入手,设置字号为12,每个汉字占16像素,6个字占96个像素,可 以算出 x=(240-96)/2=72。后面的两段文字可以根据每个字所占的像素情况计算,三段文字间距可以使用 Python 运行弹窗测试并进行微调(参考下图完成坐标设置)。



最后, "南"和"北"两汉字可以使用更小的字体,使用相似的计算方法完成坐标设置。完整的项目 代码如下:



- 2. 程序运行
- STEP1: 连接行空板
- STEP2: 点击运行
- STEP3: 观察效果

观察行空板屏幕,可以看到路牌显示在屏幕上。(参考下图)



<u>知识园地</u>

1. 顺序结构

回顾第一课和第二课,我们在编写代码的时候都是按照显示内容顺序,逐步完成。像这样按照步骤顺 序编写的程序结构称为顺序结构。顺序结构是最简单的程序结构,它的执行顺序是自上而下,依次执行, 以本项目显示路牌为例,可以被描述成如下程序流程:



2. 指令学习

本项目主要使用了矩形显示指令,以及数码管显示指令,接下来我们就详细学习一下指令。



置数字等内容,同时可对内容的 位置、字号以及颜色进行调整。

挑战自我

- 1、你已经学习过矩形的显示了,不妨使用 等待几秒 和矩形显示指令设计一个不断变大的正方形的隧道效
- 果,显示在你的行空板上呀!



运算符

<u>附录一</u>

字体大小对照换算

汉字属于方块字,也就是说汉字所占的空间是正方形,因此在描述大小的时候也就是在描述汉字的宽度或高度。我们在使用行空板文字显示指令的时候,通过修改字号实现对于字体大小也就是宽度的设置,这里的字号单位为磅(pt),然而行空板屏幕坐标系的单位是像素(px)。为了更好的在行空板上对文字进行排版,需要将这两个单位进行换算,换算公式是:**1pt = 4/3px**。需要说明的是像素是屏幕显示的最小单位,必须是整数,当换算过程中出现了小数,一般要对计算结果进行向下取整。如计算字号 6.5pt 所占的像素(px):

$$6.5\,pt = 6.5 \times \frac{4}{3}\,px = 8.6\,px \approx 8\,px$$

当然你也可以直接在下面的"字体大小对照换算表"中查找需要的字号磅数对应的像素值。

注意:行空板上显示的汉字字体是汉仪旗黑,不可修改。由于汉字属于方块字的特性,查找数据时,可以不用关注字体,只使用红色框内的数据。

字号 磅数 毫米 英寸 像素 宋体 黑体 黑体 楷体 明弱 42 14.8 0.58 56 宋体初 黑体初 楷体 小別 36 12.7 0.50 48 宋体小初 黑体小初 楷体小初										
148 14.8 0.58 56 宋体初 黑体初 楷体初 小別 36 12.7 0.50 48 宋体小初 黑体小初 楷体小初 -8 26 9.2 0.36 34 宋体一号 黑体一号 楷体一号 小一 24 8.5 0.33 32 宋体小一 黑体小一 楷体小一 -28 22 7.8 0.31 29 宋体小一 黑体小一 楷体小一 -28 22 7.8 0.31 29 宋体小一 黑体小 楷体小 -29 16 5.6 0.22 21 宋体小二 黒体 楷体小二 -29 16 5.6 0.22 21 宋体小三 黒体 橋本小 -29 16 5.6 0.22 21 宋体 黒体 黒体 橋本 -29 12 2.0 7.8 18 第.3 0.21 20 宋体小三 -29 3.2 0.13 12 宋体小三 黒体 黒体 橋本 -29 3.2 0.13 12 寒* 黒 黒 <th>字号</th> <th>磅数</th> <th>毫米</th> <th>英寸</th> <th>像素</th> <th>宋体</th> <th>黑体</th> <th>楷体</th>	字号	磅数	毫米	英寸	像素	宋体	黑体	楷体		
小別 36 12.7 0.50 48 宋体小初 黑体小初 橋体小初 -号 26 9.2 0.36 34 宋体一号 黑体一号 楷体一号 小- 24 8.5 0.33 32 宋体小一 黑体小一 楷体小号 二号 22 7.8 0.31 29 宋体小一 黑体小一 楷体小二 二号 18 6.3 0.25 24 宋体小二 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体小三 黒体小二 楷体小二 三号 16 5.6 0.22 21 宋体小 黒体小三 기三 15 5.3 0.21 20 宋体小三 黒体小三 기三 14 4.9 0.19 18 宋体四号 黒体四号 기三 12 4.2 0.17 16 宋体小四 黒体小四 五号 10.5 3.7 0.15 14 宋雄小四 </th <th>初号</th> <th>42</th> <th>14.8</th> <th>0.58</th> <th>56</th> <th>宋体初</th> <th>黑体初</th> <th>楷体初</th>	初号	42	14.8	0.58	56	宋体初	黑体初	楷体初		
-号 26 9.2 0.36 34 宋体一号 黑体一号 楷体一号 小- 24 8.5 0.33 32 宋体小一 黑体小一 楷体小一 二号 22 7.8 0.31 29 宋体二号 黑体二号 楷体小二 六 18 6.3 0.25 24 宋体小二 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体三号 黑体小三 楷体小二 기 18 6.3 0.25 24 宋体小 黑体小二 18 6.3 0.25 24 宋体小 黒体小二 19 16 5.6 0.22 21 宋体三号 15 5.3 0.21 20 宋体小 19 14 4.9 0.19 18 宋体四号 <	小初	36	12.7	0.50	48	宋体小初	黑体小初	楷体小初		
小- 24 8.5 0.33 32 宋体小一 黑体小一 楷体小一 二号 22 7.8 0.31 29 宋体二号 黑体二号 楷体二号 小二 18 6.3 0.25 24 宋体小二 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体三号 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体三号 黒体二号 楷体小二 □号 14 4.9 0.19 18 宋体四号 黑体小三 楷体小三 □号 14 4.9 0.19 18 宋体四号 黑体小目 楷体小三 □号 14 4.9 0.19 18 宋体四号 黑体小目 楷体小三 □号 14 4.9 0.19 18 宋体四号 黑体小目 #体小目 □号 14 4.9 0.19 18 宋体四号 黒体小目 #体小日 □号 12 4.2 0.17 16 宋体小三 黒体小目 ボ ボ ボ ボ ボ ボ ボ ボ ボ ボ	—号	26	9.2	0.36	34	宋体一号	黑体一号	楷体一号		
二号 22 7.8 0.31 29 宋体二号 黑体二号 楷体二号 小二 18 6.3 0.25 24 宋体小二 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体三号 黑体二号 楷体小二 파三 15 5.3 0.21 20 宋体小三 黑体小三 楷体小三 四号 14 4.9 0.19 18 宋体四号 黑体小三 楷体小三 12 4.2 0.17 16 宋体小四 黑体小四 楷体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体小四 楷体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体小四 楷体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体小四 標本4 五号 10.5 3.7 0.15 14 宋体五号 黒体本5 #本4 小五 9 3.2 0.13 12 宋体本5 黒体4 #本4 六方 7.5 2.6 0.10 10 寒本5 黒 #**** </th <th>小—</th> <th>24</th> <th>8.5</th> <th>0.33</th> <th>32</th> <th>宋体小一</th> <th>黑体小一</th> <th>楷体小一</th>	小—	24	8.5	0.33	32	宋体小一	黑体小一	楷体小一		
小二 18 6.3 0.25 24 宋体小二 黑体小二 楷体小二 三号 16 5.6 0.22 21 宋体三号 黑体三号 楷体三号 小三 15 5.3 0.21 20 宋体小三 黑体小三 楷体小三 四号 14 4.9 0.19 18 宋体四号 黑体四号 楷体四号 小四 12 4.2 0.17 16 宋体四号 黑体小四 楷体五号 五号 10.5 3.7 0.15 14 宋体五号 黒体小三 #体本号 方号 7.5 2.6 0.10 10 寒母大五 黒体小五 #本サ 小六 6.5 2.3 0.09 8 瑞士号 第本日 小六 6.5 2.3 0.09 8 瑞士号 水六 5 1.8 0.07 6 第849	二号	22	7.8	0.31	29	宋体二号	黑体二号	楷体二号		
三号 16 5.6 0.22 21 宋体三号 黑体三号 橋体三号 小三 15 5.3 0.21 20 宋体小三 黑体小三 橋体小三 四号 14 4.9 0.19 18 宋体四号 黑体四号 橋体四号 小四 12 4.2 0.17 16 宋体小四 黒体四号 橋体小町 五号 10.5 3.7 0.15 14 宋体万号 黒体小五 橋体小手 五号 10.5 3.7 0.15 14 宋体万号 黒体赤号 橋体小手 大合 7.5 2.6 0.10 10 東保赤号 黒体小五 橋本小香 小六 6.5 2.3 0.09 8 黒い小本 黒田小香 黒本本号 小六 6.5 1.9 0.08 7 細い母 黒田小香 黒本本号 小長 5 1.8 0.07 6 細い母 黒田小母 雪田小母	小二	18	6.3	0.25	24	宋体小二	黑体小二	楷体小二		
小三 15 5.3 0.21 20 宋体小三 黑体小三 楷体小三 四号 14 4.9 0.19 18 宋体四号 黑体四号 楷体四号 小四 12 4.2 0.17 16 宋体小四 黒体小四 楷体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体五号 楷体ふ子 小五 9 3.2 0.13 12 寒体小五 黒体五号 楷体小五 六号 7.5 2.6 0.10 10 寒体六 黒体六母 #**** 小六 6.5 2.3 0.09 8 第***** 黒金4 #**** 七号 5.5 1.9 0.08 7 #**** 黒金4 #**** 八号 5 1.8 0.07 6 #**** #****	三号	16	5.6	0.22	21	宋体三号	黑体三号	楷体三号		
四号 14 4.9 0.19 18 宋体四号 黑体四号 楷体四号 小四 12 4.2 0.17 16 宋体小四 黒体小四 楷体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体五号 檜体五号 小五 9 3.2 0.13 12 宋体子 黒体小五 神体小五 六号 7.5 2.6 0.10 10 寒洋子 黒体小五 神体小五 六六 6.5 2.3 0.09 8 雨時小 黒母子 博士中 七号 5.5 1.9 0.08 7 無市小 黒母子 神士中 八号 5 1.8 0.07 6 第四八母 黒母小母 神士中	小三	15	5.3	0.21	20	宋体小三	黑体小三	楷体小三		
小四 12 4.2 0.17 16 宋体小四 黑体小四 潜体小四 五号 10.5 3.7 0.15 14 宋体五号 黒体五号 潜体五号 小五 9 3.2 0.13 12 宋体小五 黒体五号 潜体小五 六号 7.5 2.6 0.10 10 寒洋大き 黒体小石 潜体小子 六六 6.5 2.3 0.09 8 雨ト小 黒田介六 潮々ホテ 七号 5.5 1.9 0.08 7 油市小 黒田介六 オキキ 八号 5 1.8 0.07 6 第市小 黒田介 オキャキ	四号	14	4.9	0.19	18	宋体四号	黑体四号	楷体四号		
五号 10.5 3.7 0.15 14 宋体五号 黒体五号 檜体五号 小五 9 3.2 0.13 12 宋体小五 黒体小五 神体小五 六号 7.5 2.6 0.10 10 宋年六寺 黒体小石 神体小子 六合 5.5 1.9 0.08 第時小小 黒田介六 神母キャキ 七号 5.5 1.9 0.08 7 第時小句 黒田介石 神母キャキ 八号 5 1.8 0.07 6 第8小母 湾田小母 湾田小母	小四	12	4.2	0.17	16	宋体小四	黑体小四	楷体小四		
小五 9 3.2 0.13 12 栄体小五 媒体小五 六号 7.5 2.6 0.10 10 突然大き 運像大き 端体小支 小六 6.5 2.3 0.09 8 半日本・ 運像小支 増産小支 七号 5.5 1.9 0.08 7 物目小 第四小章 増産小支 八号 5 1.8 0.07 6 第四小章 第四小章 増産小車	五号	10.5	3.7	0.15	14	宋体五号	黑体五号	槽体五号		
六号 7.5 2.6 0.10 10 東保大寺 美術大寺 美術大寺 美術大寺 学校大寺 小六 6.5 2.3 0.09 8 第64/1 第金/1 第金/1 七号 5.5 1.9 0.08 7 第四日 第四日 第四日 第四日 八号 5 1.8 0.07 6 第四日 第四日 第四日 第四日	小五	9	3.2	0.13	12	果体小五	黑体小五	稽体小五		
小六 6.5 2.3 0.09 8 第64点 第金点 第金点 第金点 七号 5.5 1.9 0.08 7 第金点 #金点 # # # # # # # # # # # # # # # #	六号	7.5	2.6	0.10	10	· 朱伴六号	黒体大号	增保共考		
七号 5.5 1.9 0.08 7 またり Re44 Haure 八号 5 1.8 0.07 6 Re14 Re14 Haure	小六	6.5	2.3	0.09	8	安林中六	繁华小六	增值公开		
八号 5 1.8 0.07 6 第8.49 発電小等 神秘小寺	七号	5.5	1.9	0.08	7	来非七号	五体七号	4449		
	八号	5	1.8	0.07	6	来自人号	20/15	特别人奇		

字体大小对照换算表

第三课 西游记舞台剧

《西游记》是中国神魔小说的经典之作,讲述了慈悲啰嗦的唐僧、神通广大的孙悟空、憨厚直率的猪 八戒以及善良老实的沙僧,师徒四人西行取经的故事。本节课我们就以西游记为主题,在行空板上展示一 个师徒四人的小舞台剧。



任务目标

在行空板屏幕上显示动态角色唐僧和孙悟空以及台词,按下行空板按键,猪八戒和沙僧及对应台词分 别出现。



<u>知识点</u>

- 1. 认识程序中的循环结构
- 2. 理解程序中对象的概念和命名规则
- 3. 掌握行空板 A/B 按键操作方法
- 4. 掌握行空板屏幕显示图片的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



<u>动手实践</u>

西游记舞台剧主要包括背景画面、动态的师徒四人和有趣的台词,当然还可以使用按键手动控制角色 出现,接下来我们就把舞台剧拆分成三个任务来制作吧。

任务一:布置舞台初始画面

在此任务中,我们通过学习行空板图片显示方法,根据唐僧和孙悟空出现的位置,完成舞台剧初始场 景的显示。

任务二: 让角色动起来

在完成舞台剧初始场景显示后,我们将通过对图片的切换和更新图片位置的方法的学习,来实现角色动画效果。

任务三:使用按键让角色逐个出现

这一步我们会学习使用行空板上的按键操作方法,控制猪八戒和沙僧出现。

任务一: 布置舞台初始画面

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。

Mind + 1	V1.7.2 RC3.0			– 🗆 X
i dud	lo 项目 ● 教育 ● 10.1.2.3 ●		() MIRKE SHREE 169	ert Dytweiget 🏟 🖨
极快	代码	、(5)行空板连接成功		
10:37	1	显示行空板IP	ALL MERINE	/
 通算符 証算符 数字 算量 文本 2本 列表 元由 2番 	##200 (2)切換到图形化编 前該方<	程模式 Python主起的开始:	(1)切掉 1 # -*- coding: UTF-8 -*- 3 # MindPlus 4 # Python 5 6 7 8 while True: 9 pass 10	è至Python模式
学供 (3) 集合			後端 (行空板)	(金) 清除给出
の重 の重 形成 行之物 に	メはな 記示単の約462× (20) Y おはな 記示二単時 外回、 (4)行空 が除た 記示法的 かたべ が除た 記示法の たべ ジェにたち 記示法の たべ	(3)出现预设(代码 恢模块 成功 (3)出现预设(代码 (2) (3) (3) (3) (3) (3) (4) (4) (4) (4) (4) (4) (4) (4) (4) (4	>>> 己成功连接到10.1.2.3 Last login: Mon Sep 5 17:5 10.1.2.121 root@unihiker:~# root@unihiker:~#	8:41 2022 from

3. 编写程序

观察一下,舞台剧初始场景中包含背景图片、唐僧、孙悟空,接下来开始学习如何把它们显示到行空板屏幕上。



首先,背景图片在行空板屏幕上的显示需要三个步骤:找到图片、将图片加载进项目、使用指令显示 图片。



第一步: 找到图片, 在附件资料中的"第三课 图片素材"文件夹中找到"bg.png"图片文件;

第二步: 将图片加载进项目, 在 Mind+的 "快捷工具区" 点击 "文件系统" 打开文件目录, 将 "bg.png" 图片拖入 "项目中的文件" 里。



拖入后,就已经成功将图片加入项目了,文件目录中会显示图片文件的完整名称(如下图):

	(3)		×
⑦ 意见反馈 实时模式 上标模式	Python模	e 😫	٥
▶ 运行 (◎代码)	3	文件系统)
文件目录			0
项目中的文件		G	১ ≡
bg.png			÷
电脑中的文件		9	5 ≡

第三步:使用指令显示图片,在指令区"行空板"分类下的"屏幕显示"里寻找含有"图片"关键字的指令,拖出对应显示图片的指令: 对象名显示图片在X0Y0,然后修改"显示图片"为背景图片名称"bg.png",将指令放在预设代码 python 主程序开始的下面。



现在,你可以使用同样的方法,将"西游记-唐僧 1.png"和"西游记-孙悟空 1.png"两张图片显示 在行空板屏幕上。关于唐僧和孙悟空的坐标设置,可以结合行空板屏幕坐标以及他们的位置关系进行设置。 注意:图片的坐标默认是图片左上角位置。



完整的代码如下:


4. 程序运行

点击运行,观察行空板屏幕显示效果(如下图)。



任务二: 让角色动起来

1. 编写程序

通过上一个任务,舞台初始画面已经显示完成,接下来我们学习如何让角色动起来。

(1) 唐僧走动

如果想让初始画面中的唐僧动起来,则需要多张唐僧走动动作的图片,按顺序不断切换。也就是说需 要将在"第三课图片素材"文件夹中所有有关唐僧的图片加载进程序,然后使用指令在程序中不断更换需 要显示的图片。



更换图片的指令在指令区"行空板"分类下的"屏幕显示"里,你可以寻找含有"更新图片"关键词的指令,拖出更新对象图片源为。



使用此指令时,需要指明需要更换哪张图片,例如上面已经显示的唐僧和孙悟空图片。通常的处理方 式是给图片设定一个对象名,比如给唐僧的图片设定为"tang",这样就可以修改这张图片的图片源了。 注意:关于对象知识的详细介绍,参见"知识园地"。

Python主程庁	研始	- 6	-	0	2					X	寸多	名-	致										
S MRA		* 💽	bg.pn	ng "	ŧΧ	0	v C		指	定更	换	的是	唐僧	图片	ŧ								
🕞 <u>제유용</u> tai	ng) Jurk	ar (* 西流	纪-唐	僧1.p	ng *) ÆX	0) v (150	Η	Ð	更象	13)	对象(tang) ਬ	(渡)	• Ē	够记	-唐僧1	.png	•
〇 対象名) G (8)	+ 💽	西游证	3-孙悟	空1.p	ng ") ÆX	60) v (170										1			
循环执行				C.			2	×		-	-							更把	與的	图片	名称		

与唐僧走动相关的图片一共有4张,所以我们需要4个更换图片指令分别对应它们。



然后按顺序完成指令搭建,为了让唐僧以合适的速度持续运动,可以在每个更换图片的指令前面放等 待 0.2 秒 并将完成的代码放在循环执行 里。

ę	Python主程序开始
e	対象名 - 显示图片 ・ bg.png ・ 在X 0 Y 0
e	対象名 tang 显示图片 * 西游记-唐僧1.png * 在X 0 Y 150
e	対象名 ● 显示图片 * 西游记-孙悟空1.png * 在X 60 Y 170
循环	
	更新图片对象 tang 图片 游为 · 西游记-唐僧1.png ·
	第 0.2 秒 ···································
	更新图片对象 tang 图片 资为 · 西游记-唐僧2.png ·
	更新图片对象 tang 图片游为 ·西游记-唐僧3.png ·
	更新图片对象 tang 图片 添为 * 西游记- 唐僧4.png *
注意 : 等待 秒 指令和	循环执行 都在"控制"分类下,关于 <mark>循环执行</mark> 的更多知识,参见"知识园
地"。	
括	制
运	

(2) 孙悟空腾云

孙悟空腾云的效果和唐僧走路非常相似,不同的是,孙悟空不止有动作的变化,还需要向上移动,改 变Y坐标。

此时我们要给孙悟空图片设定对象名为"sun",修改坐标要使用到 更新对象名 的数字参数 为 指 令,你可以在"行空板"指令分类下,寻找含有"数字参数"关键词的指令,拖出并修改对应内容。



为了让孙悟空向上飞行慢一点你可以多次改变孙悟空图片位置。参考完整代码如下:

Python主程序开始
○ 対象名 ● 显示器片 ● bg.png ● 在X ● Y ●
○ 対象名 tang 显示图片 "西游记-唐僧1.png" 在X 0 Y 150
○ 对象名 sun 显示图片 * 西游记-孙悟空1.png * 在X 60 Y 170
等待 0.2 秒
○ 更新対象名 sun 的数字参数 y ▼ 为 130 極
等待 0.2 秒 空空
● 更新对象名 sun 的数字参数 y → 为 90 <mark>6</mark> 起
等待 0.2 秒
● 更新对象名 sun 的数字参数 y ▼ 为 60
循环执行
等待 0.2 秒
○ 更新图片对象 tang 图片源为 "西游记-唐僧1.png "
等待 0.2 眇
⑤ 更新图片对象 tang 图片源为 "西游记-唐僧2.png"
等待 0.2 秒
● 更新图片对象 tang 图片源为 "西游记-唐僧3.png "
等待 0.2 秒
更新图片対象 tang 图片源为 "西游记-唐僧4.png"

2. 程序运行

检查行空板连接,点击运行,观察行空板屏幕,可以看到孙悟空飞起,然后唐僧原地走动。



3. 试一试

(1) 现在孙悟空已经可以上升了,但是孙悟空飞起的动作还很单调,请你利用"第三课 图片素材"文件 夹中孙悟空的图片素材,让他能够腾云上升;



(2) 现在师徒四人还差沙僧和猪八戒,请你把猪八戒和沙僧也添加进舞台,并让他们和唐僧一起走动。



任务三:使用按键让角色逐个出现

1. 编写程序

在上一个任务中,运行代码后,师徒四人已经出现,现在我们可以使用行空板自带的 A/B 按键,手动 控制角色的出现。处理的大致思路是:先让角色移出行空板屏幕,然后使用行空板 A/B 按键,控制他们移 动到合适的位置。

以猪八戒为例,移出行空板屏幕的操作只需要修改猪八戒开始出现的 X 坐标。



接下来,我们来学习按键的操作。按键操作指令在"行空板"分类下的"鼠标键盘事件"中,拖出当



注意:在行空板屏幕正对你时,行空板 A/B 键位于行空板左侧,上为 A 键,下为 B 键。关于行空板按 键相关知识,参见知识园地。



接下来,就可以很方便的完成当按下A键,猪八戒的X坐标更新到合适的位置。



最后,完善你的代码,参考上节课的知识,为唐僧、孙悟空、猪八戒和沙僧四个角色添加对应的台词。 完整的程序请参考以下代码:



等待	0.2 秒
$\overline{\bigcirc}$	更新图片对象 tang 图片源为 "西游记-唐僧3.png "
\bigcirc	更新图片对象 zhu 图片源为 "西游记-猪八戒3.png "
\bigcirc	更新图片对象 sha 图片源为 "西游记-沙僧3.png "
等待	0.2 秒
$\overline{\bigcirc}$	更新图片对象 tang 图片源为 "西游记-唐僧4.png"
00	更新图片对象 tang 图片源为 "西游记-唐僧4.png" 更新图片对象 zhu 图片源为 "西游记-猪八戒4.png"
000	更新图片对象 tang 图片源为 "西游记-唐僧4.png" 更新图片对象 zhu 图片源为 "西游记-猪八戒4.png" 更新图片对象 sha 图片源为 "西游记-沙僧4.png"

2. 运行程序

点击运行,唐僧和在空中的孙悟空头顶出现台词,按下行空板右侧 A 键,猪八戒和对应台词出现。 按下 A 键:



3. 试一试

你现在已经学会按键操作了,不妨试一试使用行空板 B 键,控制沙僧和对应台词出现吧。



知识园地

1. 循环结构

循环结构是指在程序中需要反复执行某个功能而设置的一种程序结构,它通常可以被描述为如下执行 流程:



在今天的项目中,为了让唐僧、猪八戒和沙僧在屏幕中一直走动,我们使用了循环执行指令,它表示将某些操作一直循环重复的执行。循环执行 是循环结构中最简单的指令,又叫做"死循环",它的执行 过程是循环往复,不会停止。另外,在循环结构中还会有条件循环、有限次循环等指令,后续的课程中用 到会再详细介绍。

2. 程序中的对象

(1) 理解程序中的对象

在程序世界的"对象"是程序中具体的操作目标,比如在舞台剧中,孙悟空图片、孙悟空台词等等这些具体的操作目标,都是对象。



有了对象,给对象起好名字,就可以对对象的位置、颜色、文字内容等属性进行修改。修改的时候通 常需要在建立好对象的基础上,搭配使用含有"更新对象"关键词的指令来完成操作,例如,要修改文字 对象 t 的文本内容为"你好",需要如下两个指令即可完成:

$\overline{\bigcirc}$	对象名 🕇 🗔	文字・行空板・ 石	< 0 Y	0 字号	20 颜色
\bigcirc	更新对象名 t	的文本内容参数为 🗖	好・		

我们现在已经学过四种对象:文字、数码管字体、矩形和图片对象,需要说明的是对于不同的对象可以修改的属性又略有不同,使用时你可以参考下表查找可修改属性。

	文字、数码管字体、矩形和图片对象可修改属性表
文字对象	x 坐标、y 坐标、宽度、文字内容、文字颜色、字体大小
数码管字体对象	x 坐标、y 坐标、文字内容、文字颜色、字体大小
矩形对象	x 坐标、y 坐标、宽度、高度、线宽、颜色,圆角半径 (圆角矩形可改)
填充矩形对象	x 坐标、y 坐标、宽度、高度、颜色,圆角半径(圆角矩形可改)
图片对象	x 坐标、y 坐标、宽度、高度、图片源

注意:更详细的属性内容介绍,请进入行空板官网的"基础控件"页面查看

(https://wiki.unihiker.com/unihiker_python_lib_2) 。

(2) 对象的命名规则

为了方便对对象进行操作,需要给对象命名,它的命名规则如下:

a) 一般由数字、字母、下划线构成;

b)不能由数字开头;

c)命名不能使用 Python 的关键字,即 Python 中已经有特殊含义的词,如 True、False、def、if、elif、else、import 等;

d) 建议命名不要太长,尽量有意义,不要使用中文。

3. 行空板按键

行空板上一共有三个按键,它们分别是 A 键、B 键和 Home 键。当行空板屏幕正对你的时候,A 键位于右侧上面,B 键位于 A 键下方,而 Home 键位于左侧。



关于行空板 A/B 键的相关指令都位于"行空板"分类指令下的"鼠标键盘事件"。另外,在断开行空板,使用 Python 运行弹窗测试代码时, A/B 按键其实对应的就是键盘上的 A 和 B 按键。使用时,记得切换至英文输入法。



Home 键不可以使用指令进行控制,但如果行空板正在运行代码,长按 Home 键,会终止代码的运行,并直接回到行空板开机画面。

另外, Home 键常被用来对主菜单进行操作。在主菜单操作部分,长按 Home 按键可进入主菜单;短 按 Home 按键表示确认或进入选项;此时,AB 键可分别表示光标的上下移动。

注意:行空板上运行了一个 Home 主菜单,关于主菜单的详细介绍,可进入行空板官网"行空板 Home 菜单"页面查看(https://wiki.unihiker.com/homemenu)。

4. 指令学习

本项目主要使用了图片显示、更新对象数字参数以及 A/B 按键操作指令,接下来我们就详细学习一下 指令。



该指令用于在行空板屏幕显示一个图 片对象,在指令中可以设置对象名、图 片文件名和图片位置。

	该指令用于改变指定对象的数字参数,
○ 更新对象名 的数字参数 x ▼ 为 1	在指令中需要写明改变的对象名、参
	数、以及修改参数值。
v v v v v v v v v v v v v v v v v v v	
and the second	
service service and the service se	
线窗	
起始点x	
超始点y	
终止点x	
终止点y	
字体大小	
	该指令用于改变指定图片显示对象的
受到國方対象 图方源为 "Img2.png"	图片内容,在指令中需要与明改变的对
-	象名以及要修改的图片文件。
A - Atto	
	该指令表示当行空板 A 键被按下,执行
	下面放置的程序,下拉参数包括A和B。
	该指令用于延时等待,当执行到这条指
等待 1 秒	令时,会等待指定的时间,然后继续执
	行下面的程序。

挑战自我

1. 你已经学会如何显示、切换图片以及按键操作,现在不妨使用"第三课挑战自我素材"文件夹内的图 片,制作一个使用按键打扰行空板屏幕精灵休息的小案例:

→ ~ ↑	📁 🕨 第三课 挑战自	1我图片素材	∨ C	.₽ 在第三课线	35白我图片素材 中
🔸 快速访问 📕 東面 🎤	 _	50	50	v v	ųυ
⊻ 下荻 ★ Ⅲ	Sleep-1.png	Sleep-2.png	Sleep-3.png	Sleep-4.png	Sleep-5.png
📕 图片 🖈 🍋 第二课	••			60	•••
<mark>22 第三連</mark> 22 受哈惊项()	Sweat-1.png	Sweat-2.png	Sweat-3.png	Sweat-4.png	Sweat-5.png
■ 行空板機 OneDrive	00	00	00	00	00
- unreat	Think-1.png	Think-2.png	Think-3.png	Think-4.png	Think-5.png

(1) 屏幕中间显示打瞌睡的动态表情;



(2) 按下行空板 A 键, 动态表情变为思考, 并显示"你找我有什么事?";



(3) 按下行空板 B 键, 动态表情变为汗颜, 并显示"没事你别按它"。



第四课 情绪卡片

你是否经历过情绪的高低起伏,情绪低落的时候不想被人打扰,情绪激昂渴望有人能够分享,但遗憾 的是很少有人能看出你的情绪,此时,情绪卡片就能很好的帮你表达当前的情绪。本节课我们就使用行空 板来制作一个情绪卡片,帮助分享你的情绪。



任务目标

通过行空板屏幕按钮来改变情绪卡片的内容。 点击"换表情"按钮,更换情绪表情,并显示对应的说明文字; 点击"确定"按钮,按钮消失,生成最终的情绪卡片。



<u>知识点</u>

- 1. 了解程序中的回调函数知识
- 2. 掌握行空板基准点知识
- 3. 掌握行空板删除对象方法

4. 掌握在行空板上显示屏幕按钮和圆形的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

本课项目可以使用屏幕按钮修改卡片内容,并通过确认按钮,生成最终情绪卡片。接下来我们将分为 初始情绪卡片界面显示以及修改表情两个任务来完成。

任务一:显示情绪卡片

在此任务中,我们通过学习行空板动态表情和圆形的显示以及基准点操作方法,完成情绪卡片的显示。 任务二:个性化修改情绪卡片

在此任务中,我们将学习行空板屏幕按钮的相关操作、对象的修改及删除方法,通过按钮的功能实现, 完成表情的更换并生成最终的情绪卡片。

任务一:显示情绪卡片

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

观察一下,在程序开始的时候,行空板上会显示一个初始情绪卡片,它由圆形表情框、动态表情和文字构成,接下来我们逐个完成他们的显示吧。



(1) 显示表情框

表情框是橙色圆形,圆形的显示指令在指令区"行空板"分类下的"屏幕显示"里,你可以寻找含有 "圆形"关键词的指令,拖出 对象名显示圆形在 X120 Y100 半径 50 线宽 1 边框颜色蓝,然后设置对 象名、线宽和颜色,将指令放在预设代码 python 主程序开始 的下面。

注意:圆形的坐标,表示的是圆心的位置。



(2) 加入动态表情

上节课我们已经知道, 动态表情的显示是由多张图片切换完成的, 但是行空板提供了更方便的添加指 令——<mark>对象名 显示内置表情 微笑 在 X0Y0 间隔 0.2 秒</mark>, 它提供了 9 个内置动态表情, 你可以查找 "动 态表情"关键词, 找到并拖出对应的指令。





找到指令后,需要将表情要放在表情框里,即需要设置表情的位置和大小。 设置表情位置,表情要和圆形表情框中心对齐,也就是说表情的中心要和圆形表情框的圆心位置一致, 此时只要修改表情的基准点,即修改代表图片位置的点,也为中心即可。



修改基准点指令为 更新对象名 的基准点为中心,你可以寻找含有"基准点"关键词的指令,拖出放在动态表情显示指令的下面,并完成代码的修改。

注意:关于基准点知识的详细介绍,参见"知识园地"。



度或高度。完整参考代码如下:

÷	Python主程序	开始	н р 4 р	•		*	, e	v s	ħ	ч Б 4 д	•	•
$\overline{\bigcirc}$	对象名 yu	an 显	示圆形在X	120) y (140	样径 50	线宽	3	边框颜色		
\bigcirc	对象名(em	y)显示	内置动态	表情	乏眼 🔻	ÆX	120	Y 14		局 0.2	秒	
\bigcirc	更新对象名	emj	的基准点	为中心	۰ – ۱		* *					
\bigcirc	更新对象名	emj	的数字参	数宽	- ₹₹	9 100						
循环执	机宁		e e	÷			<i>*</i> 0					

4. 程序运行

点击运行,行空板屏幕显示表情框和动态表情(如下图)。



5. 试一试

你已经学会了对象基准点的设置指令,一段文字可以看成是一个整体,现在请你试试使用修改基准点的方法,让"情绪卡片"文字的横坐标与圆形表情框对齐。



任务二:个性化修改情绪卡片

4. 编写程序

通过上一个任务,初始情绪卡片显示完成,接下来我们学习屏幕按钮的设置。

(1) 添加屏幕按钮





(2) 设置按钮功能

按钮功能是通过按钮显示指令中的"回调函数"实现的,具体实现需要用到回调函数指令—— 当点 击回调函数 button_click1 被触发 ,你可以通过"回调函数"关键词寻找指令,拖出并修改回调函数名。 回调函数需与已添加按钮指令中的回调函数名保持一致。以"换表情"为例有:



注意:在程序中,行空板按钮指令和回调函数指令一定要成对出现。另外,回调函数一般用来实现由 某操作触发的功能,关于"回调函数"知识的详细介绍,参见"知识园地"。

完成对应回调函数指令修改后,就可以在它们下方逐个编写按钮的功能了。接下来我们就学习实现"换表情"和"确定"按钮的功能。

(a) "换表情"按钮功能

"换表情"按钮,修改情绪表情,要使用到的是 更换表情对象 表情源为 指令,寻找含有"表情源" 关键词指令,并修改指令中的"表情对象名"和"表情源",然后放在对应按钮的回调函数指令下。



当然,你也可以给对应的表情添加文字说明,只要使用 更新对象名 的文本内容参数为"你好" 指令,修改文字对象的文字内容即可。



(b) "确定"按钮功能

点击确定"按钮,所有按钮消失,生成最终的情绪卡片。按钮消失要使用到的是 删除对象 指令,寻找含有"删除对象"关键词指令,并写明要删除的对象。如"换颜色"按钮,对象名为 button1,则修改为 删除对象 button1 ,然后放在对应按钮的回调函数指令下。



关于"确定"按钮的删除方法相同。下面是"换表情"和"确定"按钮显示和功能实现的参考代码:

Python主程序开始
○ 对象名 yuan 显示圆形在X 120 Y 140 半径 50 线宽 3 边框颜色
○ 对象名 emj 显示内置动态表情 眨眼 ▼ 在X 120 Y 140 间隔 0.2 秒
● 更新对象名 emj 的基准点为 中心 ▼
⑤ 更新对象名 emj 的数字参数 宽 ▼ 为 100
○ 对象名 text 显示文字 "情绪卡片 " 在X 120 Y 210 字号 15 颜色 ●
● 更新对象名 text 的基准点为 上 ▼
対象名 button1 増加按钮 * 換表情 在X 80 Y 20 宽 80 高 40 点击回调函数 button_click1 短
○ 対象名 button2 增加按钮 "确定" 在X 80 Y 260 宽 80 高 40 点击回调函数 button_click2
循环执行
当点击回调函数 button_click2 被触发 当点击回调函数 button_click1 被触发 田
◎ 删除对象 button2 更新对象名 text 的文本内容参数为 "我好生气" 置

5. 程序运行

检查行空板连接,点击运行,观察行空板屏幕,可以看到初始情绪卡片以及两个按钮。 按下"换表情",动态表情发生变化;按下"确定",生成最终情绪卡片。

	() ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
按下"换表情":	按下"确定":
(・ ●■・語・■ ※ 接表情 ● 我好生气 ● 確定	(○ ● ● ● 書 ● 書 ● ● ● ● ● ● ● ● ● ● ● ● ●

知识园地

1. 回调函数

回调函数你可以把它理解为由一个操作触发的一系列操作或变化,就像本课项目中点击按钮触发的表 情变化一样。使用时,触发操作和回调函数指令必须成对出现,回调函数指令一般作为某段程序的第一个 指令出现。代码执行时,操作代码触发回调函数执行,回调函数执行结束后,会回到刚才操作代码的部分 继续执行,整个代码执行流程如下:



2. 行空板基准点

基准点概念的出现是由于要显示在行空板上的元素基本上都是有面积的形状,如果要用坐标点描述位置,必须确认坐标点代表的是元素形状上的哪个位置,这个位置点就是"基准点"。行空板屏幕上显示的文字、按钮、图片等元素都可以修改基准点的,而矩形、圆形等形状的基准点是不可修改的。



行空板一共设定了9个基准点,即上、下、左、右、中、左上、左下、右上、右下。也就是说,如果 要在行空板屏幕的中间位置显示一张图片,即下图中的红点位置显示,坐标为(120,160),根据基准点 的不同,会出现以下9种位置情况。



3. 指令学习

本项目主要使用了按钮显示、回调函数、动态表情、删除对象等指令,接下来我们就详细学习一下它们。



更新对象名 的文本内容参数为 · 你好 ·	该指令用于修改文字对象的文 本内容。使用时,需写明要修 改的对象名称和修改的文字。
	该指令用于删除行空板屏幕上 的指定对象。使用时,需写明 要删除的对象名称。
	该指令用于删除在行空板屏幕 上的所有对象。

挑战自我

我们已经实现了情绪卡片项目,但是项目中可以添加的表情还不够丰富,请你使用今天学习的内容, 动手试一试丰富一下你的"情绪卡片"项目

- (1) 增加其他按钮切换更多的动态表情;
- (2) 增加"复原"按钮,点击后使卡片回到初始状态。



第五课 密室逃脱游戏

密室逃脱——顾名思义就是从一间封闭的房子里面逃出来,玩家在游戏过程中需要保持头脑清醒,根 据提示找到钥匙,打开房门让自己脱离险境。怎么样,是不是迫不及待地想要试试这个奇妙的游戏了。本 节课,我们就来学习一下如何制作一款密室逃脱的游戏,体验不一样的游戏旅程。



任务目标

游戏场景设置:利用各种图片,布置密室场景,并将钥匙藏在地毯下面。 游戏机制设置:需要找到藏在家具中的钥匙,将钥匙移动到门锁位置,房门打开。



<u>知识点</u>

- 1. 掌握比较运算符与逻辑运算符的使用方法
- 2. 理解程序中的变量和全局变量概念
- 3. 认识程序中的条件语句结构
- 4. 认识行空板屏幕鼠标
- 5. 掌握使用鼠标控制图片移动的方法

材料清单

硬件清单:



动手实践

密室逃脱的游戏看起来很难,但只要将逻辑理顺了其实很容易实现的。游戏的实现主要是设置游戏场 景,根据不同的游戏场景设定一些游戏机制。接下来,根据布置游戏场景与设定游戏机制这两个任务,实 现一个简单的密室逃脱游戏吧!

任务一: 布置游戏场景

在这个任务中,需要将不同背景、门、地毯等图片在指定位置显示,并将钥匙藏在地毯下。

任务二:设定游戏机制

学习行空板触控移动图片的方法,找到钥匙后,将钥匙移动到门锁位置,打开房门完成游戏。

任务一: 布置游戏场景

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。

Hind+	V1.7.2 RC3.0			- 🗆 X
Mine	£6 项目 ≠ 收程 ≠ 10.1.2.3 ÷		() RUUER SHARE LIGA	et Pythonest 🏟 🍥
极快	代码	(5)行空板连接成功	•	
一 控制	HTT I	显示行空板IP	III ANTENE	/
 □ □ □ ○ ○	 → HAERFA (2)切換到图形化组 → NARA ● BACT ● FREE ● G → NARA ● BACT ● FREE ● G → NARA ● BACH ● INTERNA → NARA ● BACH ● INTERNA → NARA ● BACH ● INTERNA 	扁程模式	(1)切換 1	至Python模式
字典			後端 (行空板)	 清除输出
集合 受量 通数 Python 管理版 管理版		(3)出现预设代码 空板模块 线成功	>>> 己成功连接到10.1.2.3 Last login: Mon Sep 5 17:56 10.1.2.121 root@unihiker:~# root@unihiker:~# root@unihiker:~#	3:41 2022 from

3. 编写程序

(1) 观察游戏场景

为了便于理解,这里对游戏场景进行了拆分(如下图),游戏场景可以拆分为密室逃脱背景、门、地毯、钥匙。



接下来对图片在屏幕上的位置进行分析,为了方便理解,会用虚线的矩形框将图片框出来,标注红点的位置就是图片在行空板上的(X,Y)坐标位置。



将游戏场景中要用到的图片,从图片素材文件中加载进项目。



拆分后发现,游戏场景布置中,主要是设置图片在不同的位置进行显示,因此这个任务是对图片的操作。你还记得在行空板的屏幕上显示图片的指令吗? 对象名显示图片 在 X0Y0。

○ 対象名 ● 显示图片 **img.png * 在X ● Y ●
首先在屏幕上显示密室逃脱的背景图片,在 Python 主程序 开始 下面增加显示图片指令。
Python主程序开始
○ 対象名 bg 显示图片 "背景密至逃睨.png" 在X 0 Y 0

注意:程序中,显示图片的名称要与项目文件中的图片名保持一致。

(2) 调节图片比例

运行上面的程序,发现背景并没有完全显示屏幕上,只能看到图片的一部分。因为行空板屏幕尺寸为 240*320,图片的尺寸为 896*1109,行空板屏的显示比图片尺寸小,所以只能显示部分图片。



	((+	₽∎	- 15	• = ě	ł	
L	-	-		-		
	•	1	2	31	G	ND

(896,1109)

让行空板完整地显示图片有两种方法: 1. 在图片处理软件中,将图片的尺寸处理为 240*320(这种 方法需要图片处理软件)。2. 在显示图片指令下,添加指令<mark>更新对象名的数字参数高为</mark>修改图片的显示 比例。



注意观察原图的右侧与行空板上显示图片的右侧有什么区别,为什么按照上面的方式修改了图片的显示比例后,行空板上还是不能显示完整的图片呢?





关于图片显示的缩小比例有一个换算公式:缩小比例=原图片(高)/行空板屏幕尺寸(高),下面就 用这个公式来探究为什么图片按比例缩小后还是显示不完整的问题。

将图片的高 1109 与行空板屏幕的高 320, 代入公式进行换算:缩小比例=1109/320,算出来的缩小 比例约为 3.5 倍,也就是图片整体缩小 3.5 倍。缩小 3.5 倍后,图片宽变为 896/3.5,缩小后的图片尺寸约 为 256*320。行空板显示尺寸 240*320,因此图片右侧会有一小部分不能显示。



门、锁、地毯图片以同样的方式进行设置,完整代码如下:



4. 程序运行

点击运行,程序运行成功后,行空板屏幕显示效果(如下图)。



任务二:设定游戏机制

1. 编写程序

上个任务,已经将游戏场景布置好了,并且钥匙也藏在了地毯下。游戏规则是,找到钥匙,将钥匙移 动到门锁位置,房门打开。

(1) 钥匙出现

钥匙被我们藏在了地毯下,点击地毯出现钥匙。触发的对象就是地毯,需要给地毯图片增加一个点击的回调函数。图片触发回调函数的方式与按钮回调函数的方式一样,都是通过点击屏幕。不同的是图片触发回调函数需要为图片对象创建一个回调函数名,使用指令<mark>对象名 的点击回调函数为 button_click1</mark>。

C)对象名	的点击回调函数为	button_click1		
回调函数对象名	/			回调函数名	

点击地毯图片后,触发的指令是<mark>更新对象名 tan 的数字参数 x 为 50</mark>,让地毯向右移动 50,将钥匙 漏出来。

Python主程序开始	
○ 对象名 bg 显示图片 *背景密室逃脱,png * 在X (0) Y (0) ○ 再新対象名 bg 的数字条数 高・カ (320)	
○ 対象名 door 显示图片 ●门.png ● 在X ① Y 55	((• • • • • • • • • • • • • • • • • • •
○ 更新対象名 door 的数字参数 高 • 为 200	
○ 对象名 (key)显示图片 "钥匙png" 在X 10 Y 280	
○ 対象名 tan 显示图片 "地毯.png" 在X ① Y 270	
一 更新対象名 tan 的数字参数 高・ 为 45	
対象名 (tan) 的点击回调函数为 (button_click1) 勤休祝(r	
	R- KO
⊖ 当点击回调函数 button_click1 被触发	0 1 2 3V GND
● 更新对象名 (tan) 的数字参数 x • 为 (50)	

注意:对象名 的点击回调函数为 button_click1需要和<mark>当点击回调函数 button_click1</mark>搭配使用, 并且回调函数名要保持一致。

(2) 行空板屏幕鼠标

怎么才能移动钥匙呢?其实在行空板屏幕上点击图片和点击按钮,使用的都是行空板屏幕鼠标功能。 当然移动钥匙也可以通过行空板屏幕上的鼠标进行移动,控制鼠标的图形化指令为<mark>当接收到鼠标移动事件</mark> 返回坐标 XY,指令中的参数如下图。



既然想到了用鼠标移动钥匙,那应该怎样去控制呢?可以让钥匙跟随鼠标一起移动,将钥匙的坐标更 改为鼠标的坐标就可以,使用指令<mark>更新对象名的数字参数 y 为</mark>,具体实现方法如下。



(3) 判断游戏胜利

游戏胜利有一个判断的标准,在密室逃脱这个游戏中,游戏胜利的标准是当钥匙移动到门锁位置时, 房门打开。关键是钥匙移动到什么位置,门才能打开呢?下面来分析一下门锁对应在行空板上的位置。



门锁对应行空板 y 轴的坐标大约为 150-200,因此需要判断钥匙的 y 坐标是否在 150-200 之间,如 果在 y 坐标 150-200 之间,<mark>执行删除对象 door</mark>,<mark>删除对象 key</mark>。

判断钥匙的位置需要用到运算符中的<mark>比较运算符">"和</mark>比较运算符"<"<mark>以及</mark>逻辑运算符"与"。 注意:关于"运算符"的介绍见知识园地。



(4) 完善游戏

游戏的功能已逐个实现,但在游戏过程中,出现了游戏漏洞。游戏中,点击屏幕任一位置都会出现钥 匙,是因为<mark>鼠标移动事件的程序</mark>与点击回调函数程序</mark>的执行没有先后顺序,即使游戏过程中没有触发地毯, 只要点击屏幕,鼠标移动事件都会执行。

要让这两个程序的执行有先后顺序,可以通过新建<mark>变量</mark>来决定该程序是否执行。接下来,就先新建一 个名称为 flag 的变量。

注意:关于"变量"的介绍见本课知识园地。

横块	代码)#1 (0 (12938) (0 x##
9	安量 新達安量	2	新建变		
控制 运算符	将安重 安量 · 飘值为 ●		变量名称: flag		
1 取字	函数 自定义模块	Python±型井 ○ 24223 bg 5 ○ 24223 bg 5 ○ 24574323 bg 5		取消降症	: (button_click1) #398232 1) 防教子参数 x ・ カ 50
文本	Python主程序开始) and the contract of the contract on the contract of the contract on the cont	显示相片 *1.png * 在X 0 Y 55 500 的数字参数 高 - 为 200		
013k 元组	 ・ (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	이 제작은 (key) please 이 변화자성은 (ke 이 지장은 Tan J	記元代H (1819年,png) (AX 10) V (280) (19) 10数字参数 高・方 30 記元代H (11828,png) (AX 0) V (270)		iFS:30中41:18回坐雨 X Y key 10(数字参数 y・ カ Y 150) 与 Y < 200) 再244
字典	▶ 文件功能 → 以 ● UTF8 ● 株式 该和) स्वातंत्रका क । () ह () स्वतंत्रका क	n)的数字参数 高十 为 (45) 的点击回避泊数为 (button_click1)		door zey
5	完整程序如下:				
			· · · · · · · · · · · · · · · · · · ·	定义标志变	量 flag 为全局变
÷	Python主程序开始 对象名 bg 显示图片	*背景密室逃脱.png * 在X 0	Y ① 当点 raliged 全局global 交量 (数(button_click1) flag	km 使发



2. 程序运行

点击运行,程序运行成功后,就可以开始玩游戏了。只有当你点击地毯后,才会出现钥匙,然后将钥 匙拖动到门锁位置,房间门打开,游戏完成。


3. 试一试

学习了如何移动钥匙后,程序中实现的是移动钥匙 y 坐标,钥匙只能上下移动,不能左右移动。接下 来就一起修改程序,实现钥匙随鼠标触发实现上下左右随意移动。

<u>知识园地</u>

1. 运算符

运算符是用于执行程序代码运算,会针对一个以上的操作数项目来进行运算。例如:3+2,其中操作数是3和2,运算符则是"+"。在 Mind+中,运算符大致可分为6种类型:算术运算符、位运算符、比较运算符、逻辑运算符、求字节数运算符、条件运算符。

这节课中用到了比较运算符和逻辑运算符,因此重点介绍一下这两种运算符,如果后续使用到其他运算符,课程中会对该运算符进行详解的介绍。



比较运算符在代码中是用作判断的,用于对常量、变量或表达式的结果进行大小比较。比较运算符返回的结果只有两种, true 和 false。



逻辑运算符用于对程序中的逻辑值进行运算,逻辑值也只有两种, true 和 false。逻辑运算符有三个, 分别为与、或、非,对应的程序模块如下图:



从模块可以看出来,逻辑运算符"与"和逻辑运算符"或",必须要有两个操作数才能进行运算,因此"逻辑与和逻辑或"又称为双目运算符。逻辑运算符"非"称为单目运算符,只要求一个操作数就可以进行运算。

逻辑运算符"与",只有当两个操作数的值都为真时,运算结果为 true(真),否则为 false(假)。 例如:当 Y=180 时,Y>150 并且 Y<200,那下面整个逻辑表达式的逻辑值为 true(真)。这个"与"就 相当于日常生活中表达的"并且"。



逻辑运算符"或",只要其中一个操作数的值为真时,运算结果就为 true(真),当两个操作数的值都为假时,运算结果为 false(假)。逻辑运算符"或",就相当于生活中我们说的"或者"。

逻辑运算符 "非",只对一个操作数的值进行运算,当操作数的值为真时,逻辑运算结果为 false (假)。 当操作数的值为假时,逻辑运算结果为 true。

2. 变量与全局变量的概念

什么是**变量**?为了便于理解,我们可以将变量看作是一个盒子,给变量赋值就相当于往这个盒子里放 东西。变量可以重复的赋值,就是说盒子里的东西是可以拿出来的,拿出来后再往盒子里放另一个东西。



这个程序就是给 flag 变量重复赋值,第一次赋值,相当于在 flag 这个空盒子里面放入了一个数字 0,那么 flag 的值就为 0。第二次赋值时,将 0 拿出来,再将数字 1 放入盒子, flag 的值为 1。



注意:新建变量时,需要给这个变量进行命名,命名时严格遵守变量的命名规则。

(1) 一般由数字、字母、下划线构成;

(2) 命名不能使用 Python 的关键字, 即 Python 中已经有特殊含义的词, 如 True、False、def、if、 elif、else、import 等;

(3) 名称中不能包含特殊字符(\`~!@#\$%^&*()+<>?:,./; '[]);

(4) 命名时最好能做到见名知意;

什么是全局变量?简单地说只要定义一次该变量为全局变量,程序中的其他函数或者循环中都可以使用该变量的值,一直到程序结束。但是如果要修改变量的值,就需要在修改的函数下再次定义为全局变量。

变量又称为局部变量,只能在它所在的函数或循环中使用,并且随函数或循环结束而结束。将变量放入 入<mark>全局 global</mark>指令中,就定义该变量为全局变量。

全局global 💮

任务二的程序中,分别在 Python 主程序开始、<mark>当点击回调函数 被触发、</mark>当接收到鼠标移动事件 返 回坐标 XY 这三处地方都出现了<mark>变量 flag</mark>,为什么是在<mark>当点击回调函数 button_click 被触发</mark>中定义变量 flag 为全局变量呢?

因为 Python 主程序开始下面给变量 flag 赋值为 0,是在初始化变量,不需要定义为全局变量。在 <mark>当点击回调函数 button_click 被触发</mark>中,需要修改变量,给变量 flag 赋一个新的值 1,因此这个回调函 数中需要定义变量 flag 为全局变量。在<mark>鼠标移动事件</mark>中,也用到了变量 flag,但是只用了它的值,没有对 变量 flag 进行修改,所以也不需要定义为全局变量。



3. 条件语句

条件语句是给定一个判断条件,并在程序执行过程中判断条件是否成立,根据判断结果(真或假)执行不同的操作,从而改变程序的执行顺序,实现更多的功能。其中判断条件也可称为表达式,而判断条件 既可以是比较运算也可以是逻辑运算。



条件语句嵌套,如果条件语句内部还需要再次应用条件语句时,就可以使用条件语句嵌套。用到了条件语句嵌套时,只有当外层的判断语句结果为真时,才会执行内层的条件判断语句。



4. 指令学习

S 更新对象名 S 的数字参数 高•为 1	该指令用于改变指定对象的数字参数,指令中 需要写明对象名、参数、以及修改参数值。 当参数选择为高或宽时,可以用来修改图片的 显示比例。
○ 対象名 ○ 的点击回调函数为 button_click1	该指令用于给指定对象增加一个点击的回调 函数,指令中需要写明对象名、回调函数名。
〇 当点击回调函数 button_click1 被触发	该指令需要和设置回调函数指令搭配使用,并 且函数名需要保持一致。
〇 当接收到鼠标移动事件 返回坐标 X Y	该指令用于获取鼠标移动过程中返回的XY坐标,鼠标移动是指点击行空板屏幕,就会触发鼠标事件。
	比较运算符">",需要两个操作数,当前一 个操作数大于后一个操作数时,返回值为 true,否则为 false。
	比较运算符"<",需要两个操作数,当前一 个操作数小于后一个操作数时,返回值为 true,否则为 false。
	比较运算符"=",需要两个操作数,当两个 操作数的值相等时,返回值为 true,否则为 false。

r	
	逻辑运算符"与",需要两个操作数,只有当
5	两个操作数的值都为真时,运算结果为 true,
	否则为 false。
	全局变量,将变量放入这个模块中,就定义该
全局global	变量为全局变量。

挑战自我

游戏场景的设置很简单,就是在行空板屏幕的不同位置显示不同的图片,并更改图片的显示大小。接下来,大家将图片素材文件中的时钟和钥匙串这两张图片也应用于游戏场景的布置吧。在之前的游戏基础 上增加游戏难度,比如墙上放置一个时钟,将真钥匙藏在时钟后面,在门边的柜子上放一串假钥匙,用来 迷惑玩家。





第六课 实景星空

夜幕降临,华灯初上,在城市中的你有多久没有看到过璀璨的星空了。其实,你看不到星星是因为人 类越来越丰富的夜生活导致夜晚环境非常亮,而环境越亮就越看不到星星。本节课我们就使用行空板模拟 一个星空,根据环境的亮度,控制星空的出现。



任务目标

根据环境的亮度情况,模拟显示星空。环境亮,看不到星星;环境暗,看到星星不断闪烁。



<u>知识点</u>

- 5. 强化变量的运用
- 6. 认识程序中的双分支条件语句
- 7. 理解三原色混色原理
- 8. 掌握行空板光线传感器的使用方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

实景星空的实现需要先获取环境亮度,然后根据亮度控制星星闪烁,接下来我们就分两个任务来完成 实景星空的制作吧。

任务一: 获取并显示环境光强度

在此任务中,我们将学习获取环境亮度和数字转字符串方法,完成环境亮度在行空板屏幕的显示。

任务二:设置星星闪烁

在此任务中,我们将学习使用三原色调色方法,并进一步应用变量,实现由环境亮度控制星星的闪烁。

任务一: 获取并显示环境光强度

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。

Mind+ V1.7.2	RC3.0	-	o x
Minde "	页目 ★ 数程 ★ 10.1.2.3 ★	@ BULLER SHEET LOBER Syllio	hillite 🛎 🖸
模块 代	码 (5)行空板连接成功		9 文件系统)
控制 行空	显 显示行空板IP	自动主线	
 通貨符 数字 数字 支本 刷 列表 元但 	MBERF (2)切换到图形化编程模式 21988	(1)切换至Py 1 # -*- coding: UTF-8 -*- 2 3 # MindPlus 4 # Python 5 6 7 8 while True: 9 pass 10 HTRHM	thon模式
 芥共 芥共 () <li< th=""><th>対象を 間は時間 振道・ 使花 対象を 同方均1時40.X 120 Y 10</th><th>授 接端 (行空板)</th><th>🕑 清除输出</th></li<>	対象を 間は時間 振道・ 使花 対象を 同方均1時40.X 120 Y 10	授 接端 (行空板)	🕑 清除输出
安島	→ ##265 ● 日示明示明HHATX 120 Y (3)出刊研研	>>>> 已成功连接到10.1.2.3	
函数 Python 行空板	(5)山场历 对数: Gan ##1 中音 * https:/ (4)行空板模块 对数: Gan ##1 #20 / 和载成功	Last login: Mon Sep 5 17:58:41 20 10.1.2.121 coot@unihiker:~# root@unihiker:~# root@unihiker:~#	022 from

3. 编写程序

首先我们要使用到行空板自带的光线传感器获取环境光强度,获取指令为 读取环境光强度,你可以 在指令区"行空板"分类下的"板载传感器"里寻找该指令。



为了方便理解显示数值的含义,可以给环境光强度数值增加一个说明,如显示"环境光线强度:233", 现在只需要适当调整文字位置即可。



为了实时获取环境光线强度数值,别忘了在 循环执行 里不断更新显示文字对象内容。参考完整代码 如下:



4. 程序运行

点击运行,并尝试用手指捂住光线传感器位置,观察行空板屏幕显示环境光线强度数值(如下图)。 捂住光线传感器,环境暗,光线强度较低; 不捂住光线传感器,环境亮,光线强度较高。 **注意**:关于光线传感器的介绍,参见"知识园地"。



任务二:设置星星闪烁

1. 编写程序

任务一已经完成了环境光强度的获取和显示,接下来我们就从星星亮起到星星闪烁,再到由环境亮度 控制星星,逐步实现实景星空的模拟。

(1) 让星星亮起来

星星亮起是利用镂空图片透过不同的颜色实现的,所以项目中需要显示可以占满屏幕的白色实心矩形、 镂空图片以及文字,图层关系如下:



接下来根据图层情况依次在行空板上显示内容即可,图片素材依然在"第六课图片素材"文件夹中, 别忘了调整图片大小。

$\overline{\bigcirc}$	x 象名 bg 显示填 短形 在X 0 Y 0 宽 240 高 320 填充颜在	百名沃汗继续传来
Θ	ス 象名 img 显示 片 ・ 夜空背景1.png ・ 在X 0 Y 0 ・ ・ ・ ・ ・ ・	▲日巴迈过接 <u>空位直</u> 星星亮起
Θ	夏新对象名 img f 数字参数 高 • 为 320 注意出现顺序	
Θ	2 象名 text1 显示文字 ・环境光线强度: ・ 在X 0 Y 0 字号 15 颜色	
\bigcirc	x 象名 text2 显示文字 🕞 读取环境光强度在X 135 Y 0 字号 15 颜	

(2) 星星闪烁

6

星星的闪烁,是星星逐渐亮起又变暗的过程,接下来我们就看看是如何实现的吧。

(a) 改变星星亮度

星星的亮度变化是通过最底层实心矩形的颜色在黑色和白色间逐渐变化的,如何改变星星亮度呢?有 两种方法:

方法一:使用 更新对象名 的颜色为蓝 指令,让底层实心矩形的颜色多次变化,但是它可选的颜色有限,无法让星星亮度连续变化;



方法二,使用"三原色调色"指令去改变矩形"对象 bg"的颜色。你可以通过"红"关键字找到 更 新对象名的颜色红 255 绿 255 蓝 255 指令。

注意:三原色混色知识,参见"知识园地"。



"三原色调色"指令可以通过数字的连续变化控制颜色从黑色到灰色再到白色均匀变化。下图展示了指令设置和对应的颜色变化。



(b)实现星星闪烁

星星的闪烁,我们可以利用"控制"分类下的 重复执行几次 控制 亮度变量 的多次变化,进而完成 最底层实心矩形颜色的连续变化。首先将 亮度变量 设为 0,然后让变量每次增加 51,并通过变量控制颜 色从黑向灰色过度,增加 5 次直到 亮度变量 变为 255,颜色即显示为白色,每次变化的时候可以设置等 待时间,让颜色变化慢一点,实现参考代码如下:



(3) 用环境亮度控制星星闪烁

项目中,环境亮不显示星星,环境暗星星闪烁。这里需要根据当前环境光强度去做判断,在任务一中 我们已经知道环境越亮,环境光强度数值越大,也就是说如果环境光强度数值大于某数值,那么不显示 星星,即底层矩形显示黑色, 否则 星星闪烁。找到"控制"分类下的如果那么否则,根据刚才的描述

来完成代码。			
注意:如	II果那么否则 指令的	详细介绍,参见"知识园地"。	
 Q 型調 ご算符 近算符 型 数字 (1) (1)	理 那 么执行 理 那 么执行 则	如果 ② 读取研境光器度 > 100 那么與行 ② 更新对接名 bg 的酸色为红 ① 绿 ① 盖 否则 将变量 亮度 • 赋值为 ① 重要执行 5 次 将变量 亮度 • 赋值为 ② 重要执行 5 次 将变量 亮度 • 赋值为 ② 重要执行 5 次 将变量 亮度 • 赋值为 ② 重新对象名 bg 的颜色为红 变量 亮度 绿 氧倍 ① 秒 	 ○ 底层矩形黑色 着不到星星 星星闪烁 变呈 亮度 蓝 变呈 亮度

最后,为了保证行空板可以实时根据环境光强控制星星闪烁,需要将刚才实现的代码放进 循环执行 里, 完整项目代码参考下图:

や Python主程序开始
○ 对象名 bg 显示填充矩形 在X 0 Y 0 宽 240 高 320 填充颜色
○ 对象名 img 显示图片 "夜空背景1.png" 在X 0 Y 0
● 更新对象名 img 的数字参数 高 マ 为 320
○ 对象名 text1 显示文字 "环境光线强度: " 在X 0 Y 0 字号 15 颜色
○ 对象名 text2 显示文字 ○ 读取环境光强度 在X 135 Y 0 字号 15 颜色
循环执行
⑤ 更新对象名 text2 的文本内容参数为 ⑤ 读取环境光强度
如果 💿 读取环境光强度 > 100 那么执行
⑤ 更新对象名 bg 的颜色为红 0 绿 0 蓝 0
否则
将变量 亮度 - 赋值为 0
重复执行 5 次
将变量 亮度 • 赋值为 变量 亮度 + 51
○ 更新对象名 bg 的颜色为红 变量 亮度 绿 变量 亮度 蓝 变量 亮度
等待 0.1 秒

2. 程序运行





<u>知识园地</u>

1. 双分支条件语句

双分支语句指的是 如果那么否则 语句,是条件判断语句中的一个,它的执行过程是,判断条件是否 成立,如果成立就执行"那么"后面的语句,否则就执行"否则"后面的语句,它的代码执行流程可以被 描述成下面右边的图片。



2. 三原色调色原理

三原色指色彩中不能再分解的三种基本颜色,本节课我们使用的三原色是光的三原色,即红、绿、蓝,简称 RGB,通过这三种颜色的混合你可以得到不同的颜色。



在三原色指令中,红绿蓝三个颜色都可以使用数值去调整,不同数值混合出的颜色也不相同,调色的 时候你可以根据需要,调整红绿蓝数值获得想要的颜色。需要说明的是,这三个颜色数值的有效变化范围 是 0-255。这里的数值表示的是每种颜色的亮暗程度,以红色为例,0表示最暗的红即黑色,而 255 则表 示正红色。



比较特殊的混色是白色, 白色是由红绿蓝三个颜色值都为 255 混合而成, 本节课项目中黑色到灰色再 到到白色的过度, 红绿蓝三个颜色需要同步从 0 到 255 变化。



3. 行空板板载光线传感器

当行空板屏幕正对你的时候,光线传感器位于上方右侧。它是由 PT0603 光敏三极管构成,你可以使用读取环境光强度指令来获得当前环境的光线强度。环境光强度是一个连续变化的数值,数值小表示当前环境暗;数值大则表示,当前环境亮。



另外,需要特别说明的是,如果代码中用到了行空板板载传感器,如光线传感器,当行空板断开时运 行代码,程序会报错,Python运行弹窗不会出现。

 · · ·	▶ 清除输出
程序报错	
/	
////_//////////////////////////	
[01] Python3.6.5 Windows-10-10.0.22000-SP selected -> board: serial: None	0
[32] Firmata ID: 0.0	
Traceback (most recent call last):	
File "C:\Users\Zhimi.Wang\Documents\min	dplus-py\us
er\2022-09-27-13-25-27\.cache-file.py", 1	ine 10, in
<module></module>	
BOALD(), Degin()	4 - A - A

此时,若你需要在电脑屏幕展示项目效果,你可以使用"远程桌面连接"工具,通过行空板 IP,将行

空板屏幕显示在电脑上。详细操作方法介绍,可以进入行空板官网的"vnc 屏幕共享"页面 (https://wiki.unihiker.com/vnc) 查看。

4. 指令学习

本项目主要使用了光线传感器、三原色调色等指令,接下来我们就详细学习一下它们。

这 取环境光强度	该指令用于获取光线传感器的 数值。
● 更新对综合 的颜色为红 255 梁 255 蓝 255	该指令用于修改指定形状、文 字对象的颜色。使用时,需写 明要修改的对象名称和红绿蓝 三个颜色的数值,有效数值范 围是 0-255.
	该指令用于将两个数相加。
重复执行 10 次	该指令用于将某段代码重复执 行指定次数。属于循环结构中 的指令。

挑战自我

漆黑的房间里有一只黑色的小猫,你能找到它在哪里吗?请你结合本节课知识,以图片素材中的"隐藏黑猫图片"为背景,设计一个"寻找黑猫"的小游戏。

提示: 1、利用光线传感器来控制行空板屏幕亮度,若环境亮,行空板屏幕暗,你只能看到一双猫的眼睛; 2、环境暗或用手指捂住光线传感器,屏幕亮起,房间里的黑猫立马就显示出来了。

屏幕暗,快来找黑猫~ 屏幕亮,黑猫显示出来了~





第七课 名画互动博物馆

如果名画不只能用双眼欣赏,还可以通过点赞互动,推荐给更多的人,你是否会为喜爱的名画驻足点 赞呢?通过网络上流行的点赞方式,不仅可以与作品进行互动,还能增添名画的趣味性,让名画得到更多 人的关注和喜爱。

《清明上河图》是一幅举世闻名的现实主义风俗画卷,向人们展示了北宋京城繁华热闹的景象和优美的自然风光,《清明上河图》画卷长 528.7cm,宽 24.8cm。这节课以《清明上河图》为例,教大家制作一个可以点赞互动的名画博物馆。



任务目标

在行空板屏幕上显示原图与缩略图,原图可以通过上下左右移动查看详细的画面,缩略图上可以查看 当前画面在图上的位置,还可以通过屏幕上的点赞按钮与名画进行互动。



<u>知识点</u>

- 1. 掌握算术运算符"+、-、/"的使用方法
- 2. 掌握图片连续移动的方法

3. 掌握行空板二维码显示的方法

材料清单

硬件清单:



行空板 x1

软件使用: Mind+编程软件 x1 下载地址: <u>https://mindplus.cc/</u>



动手实践

要和《清明上河图》实现名画互动,需要先设计一个便于交互的界面,然后实现移动鼠标查看详细画 面和缩略图,最后进行点赞互动。可分为下面三个小任务执行。

任务一: 首页显示

这个任务主要是做一个简单的首页介绍,说明互动的名画是《清明上河图》。

任务二:图片移动

使用行空板屏幕鼠标进行画卷上下左右移动,查看完整的画卷。由于画卷过大,查看过程中无法准确 地知道自己当前看到的画面对应在画卷中的位置,采用缩略图加定位框的形式帮助我们快速定位。

任务三: 点赞互动

在行空板屏幕底部设置一个互动按钮,按下按钮屏幕上出现点赞的大拇指。

任务一: 首页显示

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+,按照下面图示完成软件准备过程。



3. 编写程序

开始编写程序之前,先来分析这个任务中具体要实现哪些功能。图片移动这个任务可将功能大致拆分为显示三种不同的对象:文字、二维码、按钮。



(1) 文字显示

首页会有三行不同的文字显示,每一行文字都是一个新的文字对象。

Python	主程序开	始	œ	8	81 - 6	9 9)	- 24	- 20	1	œ	8	99	2	3)	×	200	20	Œ	8	100	3
🕞 对象名	wenzi1) 5	床文	字 (" 欢迎	四来到	名画	互动	博物	馆") Æ	< (2	25	v (1	0	字号	(12) ø	ie (
◎ 対象名	wenzi2) 5	示文	字(-	- 《清	明上	河图	» •) 在	x (6	5	Y (4	0	枵	12) 🕅	铯			
	wenzi		示文	字 ("扫码	马查看	画卷	详细	介绍	•) (ŦX (35) Y	70	字	8 (12	颜色			
循环执行	1	1	14	2	(i)	a - 41	12	16	2	14	1	10	1	¥1	1	26	1	14	1	- 60	

(2) 二维码显示

如何才能在行空板上显示一个二维码呢?两种方法: 1.保存一个二维码图片,以显示图片的方式显示 二维码。2. 使用显示二维码指令,具体用法如下。

🔘 对象名 🔵 显示二维码 内容	・https://unihiker.com 7 7 70 Y 100 边长 100	
显示二维码内容, 可以是具体的 文字、数字、或者是网址链接。	二维码的边长(大小)	



运行上面的程序,试一下<mark>显示二维码</mark>指令是否能将文字内容"清明上河图简介"生成二维码,并显示 在行空板上。



(3) 进入按钮

在行空板坐标为(60,255)的位置,增加一个宽为120,高为30的按钮,并且按钮上显示文本"点击按钮进入"。



4. 程序运行

点击运行,上面的程序运行成功后,行空板屏幕显示效果(如下图)。



5. 试一试

首页显示中的二维码,通过微信扫一扫,扫出来的是文字"清明上河图简介"。接下来完善一下程序 功能,让参观者扫描首页的二维码后,出现的是对清明上河图详细的介绍(详细的介绍可以是《清明上河 图》的百度百科内容)。手机扫码后的界面如图所示:

提示:将显示二维码中的内容,换成《清明上河图》百度百科的网址。

网址链接:

https://baike.baidu.com/item/%E6%B8%85%E6%98%8E%E4%B8%8A%E6%B2%B3%E5%9B%BE/ 102?fr=kg_general



任务二:图片移动

1. 编写程序

上一个任务中,实现了一个简单的首页显示。接下来的任务就是实现,点击按钮后,会进入一个新的 页面,在新的页面中需要用到的对象如下图。



(2) 缩略图、缩略框以及定位框显示

原图宽为 1833, 高为 500, 如何才能让图片缩小 10 倍后, 显示在行空板的左上侧呢? 使用<mark>更新对象</mark> <mark>的数字参数</mark>指令, 设置图片的数字参数高为 50。



为了让缩略图和背景原图有一个区分, 给缩略图绘制一个矩形框, 将图片框起来。原图 1833*500 宽 与高等比例缩小 10 倍后, 尺寸变为 183.3*50, 因此需要绘制一个宽为 183.3, 高为 50 的矩形框。

			areas a	/月明	上門图	.png ") #X	0	0					
	余名(suc	lue) 显示	图片("清明	上河图.	png ") 在X (0 Y	0					
⊝ ₹	新对象名	suolue	的数字	参数	高•	为 (50		4	2	- 3	45	2		\$ -
0 x	家名 (kua	ing) 显示	D 形在	x ()	y ()	宽(183.3	高	50)线	宽 🕻) 边	柜飾	通	

定位框用于缩略图上,可以通过定位框在缩略图上的位置,来判断行空板当前界面上显示的画面在原 图中的位置。

缩略图与原图的尺寸关系是 1:10,那么定位框与行空板界面的尺寸关系一样,也是 1:10。行空板界面的尺寸为 240*320,因此定位框的尺寸就为 24*32。



(3) 图片左右移动

如何才能判断参观者是想要查看左侧画面还是右侧画面呢?使用指令<mark>鼠标移动事件</mark>,获取鼠标返回的 X坐标,再使用<mark>条件分支</mark>语句指令,判断鼠标点击的是行空板屏幕的左侧还是右侧。如果>=160,说明鼠 标在行空板屏幕右侧;如果 X<=80,说明鼠标在屏幕左侧。



现在已经可以判断参观者是想要查看右侧还是左侧的画面了,那要怎样才能实现图片移动呢? 控制图 片左右移动,可以通过改变图片在行空板上显示的 X 坐标来实现。具体分析如下:



原图的坐标为(0,0)怎样才可以连续的改变 X 的坐标值呢?新建一个<mark>变量 img_x</mark>用来改变图片的 X 坐标值。



(4) 定位框移动

原图的左右移动已经实现了,但是定位框还不能跟随原图左右移动。这里使用一个小技巧,让定位框 跟随图片移动而移动,定位框是行空板的屏幕缩小10倍,原图在行空板上移动img_x,定位框在缩略图 上应该移动img_x/-10(定位框的移动是图片移动的相反方向,原图左移,查看右侧画面,定位框右移,因此img x/-10)。使用更新数字参数指令,更新定位框的X坐标位置。

○ 更新对象名 dingwei 的数字参数 x ・ 为 安量 img_x / -10

程序示例如下:



(5) 页面切换

运行上面的程序,实现了图片左右移动,并且定位框也跟随图片在缩略图上移动。你有没有发现,任 务一和任务二是两个独立的程序,两个程序只能独自运行,导致首页显示和图片移动没有任何关联。接下 来,将这两个独立的程序合并成一个,实现当点击首页显示页面中的进入按钮后,界面切换到任务二中的 图片移动页面的功能。完整程序如下:

注意:使用多个页面,所有页面中包含的对象,都必须在 python 主程序开始下新建。不属于第一个 页面的对象,都要设置为不可见。设置为不可见的方法,将该对象的 X 坐标设置为 240,即显示在行空板 屏幕坐标外,这样在行空板屏幕上就看不见它了。进入第二个页面时,如果第一个页面的对象不会再使用 到时,删除第一个页面中的对象即可。



2. 程序运行

点击运行,程序运行成功后,首先进入第一个页面的首页显示,点击最下方的进入按钮,进入图片移动界面,点击屏幕左右两侧,实现图片左右移动,定位框也会跟随图片移动而移动。



3. 试一试

你可以在上面程序的基础上,增加一个图片上下移动的功能吗?实现图片上下左右都可以移动,同样的定位框也要实现上下左右移动。

提示:新建一个变量img_y,通过改变img_y的值实现图片上下移动。

任务三: 点赞互动

1. 编写程序

再开始编写程序前,先来分析一下,这个任务中具体要实现一个怎样的功能。首先要有一个点赞的按 钮,按下这个按钮后,让大拇指的图片从按钮位置从下到上的移动。



(1) 主程序下设置对象隐藏

这个任务的程序是在任务二完整程序上进行添加,在 Python 主程序开始下,设置赞按钮、赞图片为不可见状态。

注意: 赞1的图片需要先从图片素材文件中, 加载进项目中。



(2) 设置按钮对象显示界面上

点击首页的"点击按钮进入"按钮后,在行空板第二页(100,280)的位置显示按钮,更新赞按钮的数字参数 x 为 100。

		5.8	S	- P	8	3		
Θ	当点击回调图	buttor	_click	1) 被	触发			
全局	global 变重 f	lag		11	1			
将强	重 flag • 赋	直为 1						
\bigcirc	删除对象(we	nzi1						
Θ	删除对象(we	nzi2						
0	删除对象(we	nzi3						
0	删除对象 (erv	veima						
Θ	删除对象(an	niu	F	π		-	2	
Θ	更新对象名 🔇	/uantu) 🕅	欧宁主	敞		为 (0	
	更新对象名	suolue) 🕅	数字章	数)		为 🤇	0	
0	更新对象名	cuang) (M	数字多	数 x		њ <u>(</u>	0)	
0	更新对象名	dingwei) f	的数字	参数	x -	为	0	
0	更新对象名	an) anac	参数		为(100		

(3) 赞图片向上移动

实现赞图片向上移动,同样还是使用指令<mark>更新图片数字参数 y</mark> 的值来实现向上移动。新建一个<mark>变量</mark> zan_y,当赞按钮按下,赞图片向上移动。由于都是一些学过的知识,这里就不做详细地介绍了,程序中 做了简单的注释,完整程序示例如下:



2. 程序运行

点击运行,程序运行成功后,行空板上显示首页的介绍页面。点击进入按钮,进入图片移动界面,可 以通过点击行空板上下左右不同的位置查看更多画面。看到你觉得有意思的地方时,点屏幕上点赞按钮, 显示与名画互动。



<u>知识园地</u>

1. 算术运算符

算术运算符就是进行数学运算的运算符, Mind+中主要的算术运算符有+(加)、-(减)、*(乘)、 /(除)。例如: 3+2, 操作数为 3 和 2, 算术运算符为 "+", 返回的结果就是 3+2 的值, 结果为 5。 算术运算符 "+", 求和。假设变量 img_x 的值为 10, 返回求和的结果为 11。



程序中的算术运算符的操作数为两个,返回的结果就是该算术运算符的计算结果。其中操作数可以是 常量、变量以及运算式,并且运算规则与数学中的规则一致。

2. 图片连续移动方法

你肯定会疑惑为什么连续改变<mark>变量 img_x</mark>的值,就能实现在行空板上左右移动图片? 接下来分析一下,图片是如何实现左右移动的。

(1) 开始时, 将图片设置在行空板 (0, 0) 的位置进行显示, 行空板上显示的画面如下图所示:



(2) 新建一个变量 img_x, 并且作为图片的 X 坐标, 改变 img_x 的值, 就是改变图片 X 的坐标值。



(3) 要是图片向左移动一个单位, 让变量 img_x - 1, 图片的坐标变为 (img_x-1, 0) 。



(4) 要实现图片连续向左移动, 让变量 img_x 连续减 1 即可, 就用到了赋值语句。

按量 img_x ◆ 赋值为 变量 img_x - 1	- 将 img_x-1 计算后的结果,赋给变量
-----------------------------	-------------------------

变量赋值语句,是将后面算式的结果赋值给 img_x。接下来就学习一下,这个指令是如何计算的吧! 首先计算后面的 img_x-1,然后将计算的结果再赋值给 img_x。例如,当变量 img_x 为 0 时,变量 img_x-1 就变为 0-1,计算的结果为-1,然后将结果-1 赋值给变量 img_x,计算完成后,变量 img_x = -1。

下一次在执行这个指令时, img_x - 1, 就变成了-1 - 1, 结果为-2, 将结果-2 赋值给 img_x。每执行一次这个指令, img_x 就在上一次的结果上-1, 图片就实现了连续左移。

要实现右移,使用 img_x+1 即可,图片连上移、下移用同样的方式新建一个改变图片 Y 坐标的<mark>变量</mark> img_y 就可以了。



3. 指令学习

()+()	运算符"+",求两个操作数的和,操作数可以是常量。变量。算式等。
	运算符"-",求两个操作数的差,操作数可以是常量、变量、算式等。
\bigcirc	运算符"/",求两个操作数的商,操作数可以是常量、变量、算式等。
★★名 ● 显示二维码 内容 * https://unihiker.com* 在X 70 Y 100 边长 100	显示二维码,可以将指令中的内容转换成二维码,其中内容可以是一段文字、一个数字或者是一段网址等。

挑战自我

当我们为喜欢的名画疯狂点赞后,怎样才能让其他人看到名画的点赞量有多少呢?接下来,大家试一下使用变量来计数,将点赞的数量显示在名画页面上。 提示:点赞计数,当赞按钮按下,计数变量就加1。
第八课 智慧钢琴

你是否想象过自己成为一名钢琴家,让音符在指尖飞舞?弹钢琴的过程,如同在平静的湖面上,用指 腹敲起的一个个音符,坠落荡起的一阵涟漪,震撼着聆听者的内心。因为种种原因,或许我们没能学习过 钢琴也没能拥有一台钢琴,但这些都阻碍不了对钢琴的喜爱。接下来,利用行空板制作一台专属的智慧钢 琴吧!



任务目标

行空板下半部分显示钢琴按键,并为每个琴键附上一个音符,琴键的上方在演奏过程中自动移动琴谱 图。



<u>知识点</u>

- 1. 认识蜂鸣器
- 2. 掌握蜂鸣器的使用方法

材料清单



软件使用: Mind+编程软件 x1 下载地址: <u>https://mindplus.cc/</u>



动手实践

这个项目中,主要是在行空板屏幕上显示一个简易的钢琴琴键,然后为每个琴键都赋一个蜂鸣器的音符。按下不同的琴键,播放不同的音符。在琴键上面会有一个辅助琴谱,琴谱可以自动移动,对于初学者 来说辅助琴谱降低了演奏者的表演难度。接下来,从下面两个任务来实现智慧钢琴项目。

任务一:设计琴键布局

利用素材图片,完成行空板屏幕上的琴键布局,并为每个琴键设置一个蜂鸣器音符。

任务二:设置辅助琴谱

做一个辅助琴谱,让琴谱自动移动。

任务一:设计琴键布局

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备





3. 编写程序

开始编写程序之前,先分析一下布置琴键这个任务中要具体实现哪些功能。首先需要在行空板左侧部 分显示 1-7 的琴键,然后为每个琴键赋一个音符。



(1) 确定琴键大小

将图片素材文件中的琴键图片,加载进项目中。

種現	代码			●#止 (③代明玄) (④	文件系统		
1	Street C			文件目录	0		
政字 【	🗿 श्रक्षह 🔘	显示二维码内容					
	-			□ 项目中的文件	⊡ ≡		
(本 【	🗿 xime 🕘	· 显示线线 起点X 🤇	Puthon t RET THE	Still.png	1		
8,	×	HOURSE TA O	19 19 JUNIT	器 器(mail 2.png)			
*	S view		No. of Concession, Name	日 Still4.ong			
	🧿 xime 📀	显示地开始形 在2		I I I I I I I I I I I I I I I I I I I			
组 🦷		ENHORS IN		III 琴體的png			
		Sarry Campagner Cry		- 第1位7.png	112日 、 田片香村		
曲 (🖱 श्रक्षत 🔘	显示器的脉动电压					
# 	9 xxx 0	27800 ADD			[-	N
#	9 xxx 0	显示阈角度充积				्र इक्षा _{ong}	N Ratzpro
±	9 1986 9	显示阈值填充起用		- etelening		ー 早線1.ang W	N Fait2.prg 4
≞	<u>a</u> x x x a	STRUM MADE		e erejny	***	्र क्रिंबी prig W क्रिंडी prig	N PriqSbi译 4 安建4prig
#	9 xxx 9	经示照相关的现代		e erejny		ਤ इश्विtang ਘ ਓਗ਼ੋtang	v req5編章 4 req5編章 た 6
<u></u>	9. NAR 9			e erejny	***	ਤੇ ਉੱਛੇ ong ਘ ਵਿਛੇ Jang Vi ਵਿਛੇ Spng	N 2012 pro 4 4 50 45% 6 6
典	. N264	<u>927700000000000000000000000000000000000</u>		e erejny		내 양생1.png W 양양1.png V 양양1.png 북왕5.png V	N Statiz proj L Stati z proj Stati z proj

行空板的屏幕高为 320, 要放置 7 个相同的琴键, 每个琴键占的宽大约为 45, 每个琴键之间再留个 1 的间距, 因此将琴键的高设置为 44 比较合理。

Python主程	序开	始										(0	, 3)	((• 🔮 🖬 •		¥	
○ 对象名 d) 显	示图	片(*琴	键	1.pn	g*)	ÆΧ	0) Y	3	ĵ.		ļ			ŀ
◎ 更新对象?	d da) ff	数	刺	靫	高・	为	44		Ĵ				1			
循环执行	5	5	2	2	2	.e	ð	ie.									
1																	
															2	3V GND	

(2) 确定琴键位置

七个琴键的大小相同,但是放置的位置应该是从上到下,依次放置。并且琴键之间需要留个1的间距,因此第二个琴键的位置就是 x 坐标不变,初始坐标 3 的基础上,加琴键高(44)再加间距1,第二个琴键坐标为(0,48)。



第三个琴键的位置就应该在第二个琴键 Y 坐标(48)的基础上,加上琴键宽度(44),再加上琴键 之间的间距 1。48+44+1=93,因此第三个琴键的坐标为(0,93)。

Python主程序开始	
对象名 do 显示图片 "琴键1.png"在X 0 Y 3	-
更新对象名 do 的数字参数 高·为 44	N
对象名 re 显示图片 "琴键2.png" 在X 0 Y 48	ω
更新对象名 18 的数字参数 高•为 44	
対象名 mi 显示图片 *琴键3.png * 在X ① Y 93	
更新对象名 mi 的数字参数 高·为 44	

以此类推,下一个琴键的位置同样是 X 坐标不变, Y 坐标在现在的 Y 坐标基础上,加上琴键宽度 (44), 再加上琴键之间的间距 1。



(3) 设置音符

按下琴键,触发的对象是琴键。所以要给每个琴键对象设置一个回调函数,需要使用指令<mark>对象名 的 点击回调函数为 button_click1</mark>。

注意:对象名 的点击回调函数为 button_click1 需要和当点击回调函数 button_click1 搭配使用, 并且回调函数名要保持一致。



怎样才能在按下琴键后,发出对应的音符呢?这就需要用到板载蜂鸣器中的指令,<mark>按节拍播放音符</mark>。 **注意:**板载蜂鸣器硬件相关介绍见知识园地。

Ê,	▶板	载蜂鸣器						
列表	Õ	播放音乐	DADADADUM -	重复	播訪	k-	次・	
<mark>向</mark> 元组	0	播放音符	4 60 ft	2				
EØ	Ø	播放音符	1低C/C3 1-	拍				
字典	Ø	后台播放	音符 1 低 C/C3					
集合	Ø	停止后台	番放					
● 变量	0	蜂鸣器重	定向到引脚 PO(~A)	-				

这个指令的音符下拉框里,提供三个音区的音符,节拍下拉框里提供八种节拍。不同音区的音符与节 拍具体使用方法如下:



关于节拍的选择,一般都是根据琴谱中节拍进行选择,当然也可以根据自己的喜好来设置。以《小星

星》琴谱为例,这首歌设定是 4/4 拍,他的意思是每小节有 4 拍,琴谱中每拍里面有 4 个音符,因此每个 音符对应的节拍就该为 1/4 拍,所以在程序指令中选择 1/4 拍。



接下来就给每个琴键设置一个对应的音符吧!当按下琴键后,才会发出音符,因此需要在琴键的回调 函数中添加按节拍播放音符指令,完整程序如下:



4. 程序运行

点击运行,程序运行成功后,行空板屏幕显示效果如下图,快根据《小星星》琴谱来演奏吧!



任务二:设置辅助琴谱

1. 编写程序

任务一演奏过程中,要一边盯着琴键一边看着琴谱,稍不留神就忘了自己弹哪里了,对于初学者来说, 弹奏一首完整的曲子简直太难了。如果在我们演奏过程中,琴谱可以自己在行空板的屏幕上显示,这样就 变得更加容易了。在上一个任务基础上增加一个辅助琴谱的功能,让琴谱自己在琴键上方移动。



(1) 显示琴谱

将图片素材文件中的琴谱图片,加载进项目中。



使用显示图片指令,设置图片在行空板(160,300)的位置显示。由于琴键占的宽度大约为160,因此琴谱的宽度不能超过80,所以使用更新对象的数字参数指令,设置琴谱图片的数字参数宽为80。

	((•• ♥■•№•×&
→ X1442 VI 月一切上 [●] 契約7 npg ● 左¥ ● V 273	
	N
○ 对象名 xì 的点击回调函数为 (button_click7)	ω
○ 対象名 qinpu 显示图片 "琴谱.png" 在X 160 Y 300	4
○ 更新对象名 ginpu 的数字参数 宽 · 为 80 · · · · · · · · · · · · · · · · · ·	U
尚环执行	5
	「
	0 1 2 3V GND

(2) 琴谱移动

琴谱移动,其实就是让图片循环向上移动,大家还记得怎样让图片向上移动吗?新建一个<mark>变量 img_y</mark> 用来改变图片的 Y 坐标值。



要实现琴谱图片连续向上移动,将图片移动的程序放到循环执行里即可。



完整程序如下:



2. 程序运行

点击运行,程序运行成功后,就可以开始演奏啦!演奏过程中,有了辅助琴谱,再也不用盯着琴键的同时,再去查看琴谱了。



3. 试一试

琴谱移动到末尾时,如果我们想再弹一次,只能重新上传程序,这样就太麻烦了。能不能在屏幕上加



一个辅助琴谱按键,当我们想再弹一次时,按下辅助琴谱琴键,琴谱重新出现,并且循环向上移动。

知识园地

1. 蜂鸣器

行空板之所以能发声,是因为背部板载了蜂鸣器,简单地说,就是行空板上板载了一个可以发声的电子元件,给这个元件写不同的程序,就可以控制它发出声音。



为什么行空板上的蜂鸣器可以发出不同的音符,而有的蜂鸣器只能发出一种声音?其实蜂鸣器按驱动方式的原理分为有源蜂鸣器和无源蜂鸣器,下面一起来了解一下这两种蜂鸣器有什么区别。

有源蜂鸣器

有源蜂鸣器和无源蜂鸣器最根本的区别是输入信号的要求不一样,这里的"源"不是指电源,而是指 振荡源。有源蜂鸣器内部带振荡源,只要一通电就会有声音输出,并且输出的声音是固定的,不能改变。



无源蜂鸣器

无源蜂鸣器的内部是没有这样的振荡源,所以仅用这样的直流信号,是不能让它发出声音的,必须方 波信号去驱动振动装置,然后输出声音。



你知道什么是**方波信号**吗?

大家都有直尺吧,将直尺的一端用手固定在桌上,然后用另一只手去拨动直尺。拨动过程中你会发现 拨动的力度不一样,直尺的振动频率和发出的声音也不一样。我们可以将拨动直尺的力度,理解为无源蜂 鸣器中的方波信号输入。输入的方波信号不同,输出的声音也就不一样。



通过对有源蜂鸣器和无源蜂鸣器的学习,现在是否明白了行空板上的蜂鸣器可以发出不同音符的原因?行空板上板载的就是这种无源蜂鸣器,程序<mark>按节拍播放音符</mark>指令中,选择不同的琴键,其实就是在给 蜂鸣器设置不同的方波信号输入,所以蜂鸣器可以输出各种不同的音符。

2. 指令学习



挑战自我

在一些难一点的琴谱中,会出现了"#"和"b"这两种符号。琴谱中加了"#"和"b"的音符,代表该音符是变化音符,"#"代表升音,"b"代表降音。



而钢琴中的白色琴键用于弹奏基本音符,黑色琴键用于弹奏变化音符。变化音与黑色琴键的对应关系 如下图:



如果要弹奏有变化音符的琴谱,就需要将黑色琴键也加上去。接下来,大家一起想一想怎样才能在行 空板上添加黑色琴键,并且让蜂鸣器发出变化音符?

提示 1: 使用绘制矩形的方式, 绘制钢琴黑白琴键。

提示 2: 按节拍播放音符指令中,点击黑色琴键可以选择变化音符。



第九课 健身打卡追踪器

众所周知,经常健身可以塑形,保持身体健康,在众多健身方式中,跑步作为一种简单又有效的健身 方式被人们所喜爱。然而,健身最难的是坚持,今天我们就使用行空板制作一个健身打卡追踪器,设定一 个目标,让行空板记录你的跑步情况吧。



任务目标

跑步时晃动行空板,行空板屏幕实时显示当前跑步步数,以及达到给定目标的进度情况。



<u>知识点</u>

- 9. 掌握绘制进度条的方法
- 10. 掌握行空板加速传感器的使用方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

健身打卡追踪器的主要功能是统计步数,并根据目标步数显示达成进度。接下来我们就分两个任务来 制作吧。

任务一:获得实际步数

在此任务中,我们将学习加速度传感器的使用方法,并利用加速度传感器完成跑步步数的统计和显示。 任务二:显示目标完成进度

基于已统计的步数,我们将结合给定的目标,学习利用矩形制作进度条,实时显示运动目标的达成情况。

任务一:获得实际步数

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

你一定听说过微信运动里的计步功能,它是通过检测手机的晃动来完成计步的,这与行空板计步原理 相似,那具体如何实现呢,接下来我们就一起学习一下。

(1) 检测晃动

行空板晃动实际上是行空板由静止突然移动再静止的过程,需要使用行空板自带的加速度传感器,通 过它的数值变化判断行空板是否被晃动。获取加速度数值指令是读取加速度的值x,你可以在"行空板"的"板载传感器"分类下找到并拖出指令,然后切换"x"为"强度"。

注意:关于加速度传感器的知识,参见"知识园地"。



现在你可以把获得的加速度数值实时显示在行空板屏幕上,并使用 等待 指令让数据变化慢一点,方便观察晃动前后的数据。

Python主程序开始			ч.,	×		e .
为象名 shu 显示文字	一 读取加速度的 值	直 强度 ▼ 在	x 120 y	90 字号	20 颜	
更新对象名 shu 的基	隹点为 上 ▼	× 4 +				
循环执行			a a			
● 更新对象名 shu 的工	文本内容参数 3 1 🕥	读取加速度的值	强度◆	e e 5		
等待 0.2 秒	一减慢显示数值变化	K		 		

当行空板静止时,数据为1左右,且不会有太大的变化;当行空板晃动时,数据会出现大于1.5的情况。





晃动行空板

因此,我们可以选择大于 1.5 的数值作为行空板晃动的判断依据,也就是说,如果加速度传感器强度数值大于 1.5,就可以计步数加一。



(2) 统计步数

我们已经知道如何去判断行空板晃动了,现在就来完成步数统计和显示吧。

首先,建立 <mark>实际步数变量</mark>,并在程序开始处将 <mark>实际步数变量设为</mark>为0。然后只要在判断晃动条件 满足后,显示并增加变量即可。



为了更清楚的说明数据含义,你可以在实际步数数值上方加一行文字说明,并对显示内容如坐标、字体等进行一定的调整,完整参考代码如下:

Python主程序开始 实际步数变量开始设0 将变量 实际步数 < 赋值为 0	
○ 対象名 t 显示文字 ・已完成步数・ 在X 120 Y 90 字号 15 颜色 ●	说明文字显示
	e e e
○ 対象名 shu 显示仿数码管字体 交量 实际步数 在X 120 Y 143 字号 20 颜色	a ser a ser
● 更新对象名 shu 的基准点为 上 ▼	- 4 V
如果 🕞 读取加速度的值 强度 🔹 > 1.5 那么执行	
○ 更新对象名 shu 的文本内容参数为 变量 实际步数 · · · · · · · · · · · · · · · · · · ·	
将变量 实际步数 ▼ 赋值为 变量 实际步数 + 1	
等待 0.2 秒	

4. 程序运行

点击运行,显示"已完成步数"和实际步数数值0,晃动行空板实际步数增加。



任务二:显示目标完成进度

1. 编写程序

通过上一个任务,我们已经可以成功统计实际步数,接下来就来根据给定的运动目标,显示目标达成 进度吧。

为了更好的展示目标达成情况,项目中增加了一个进度条。进度条是由一个空心矩形外框和实心矩形 构成。



这里的矩形外框宽度表示目标步数,而进度条的宽度表示实际步数,当实际步数越来越多时,进度条 会不断变长,直到达到目标步数。所以,在设置进度条坐标的时候,要注意矩形外框和实心进度条的坐标 一致,另外,开始的时候实心进度条宽度为 0。



至于进度条矩形的宽度变化,需要根据目标步数的完成比例来处理,而完成比例=实际步数/目标步数,

进度条的当前宽度=完成比例*进度框的宽度,若设定目标步数为1000,则有**进度条的当前宽度=完成比例** *160。 现在我们建立 完成比例变量,然后通过计算并更新进度条宽度即可。



另外,在进度条宽度变化的过程中,还有一种情况就是如果实际步数超过了目标进度,进度条就会超 出矩形外框。

((• 🔮 🖬 • 🎫 🔆	
已完成步数	
L	

此时,我们可以判断目标和实际步数的关系,如果实际步数超过1000(目标步数),进度条宽度固定为160,否则遵循按比例显示宽度。

如果	变量 实际	遨 >	1000	耶么执行					
Θ	更新对象名	jindu	的数字参数	宽▼	为(160			
否则								- 22	
Θ	更新对象名	jindu	的数字参数	宽 •	为	变量 🗐	动比例)*(160

至此,我们就已经完成了案例的核心功能,完整示例代码如下:

Python 主程序开始
将变量 实际步数 ▼ 赋值为 0
将变量 完成比例 ▼ 赋值为 0
○ 対象名 t 显示文字 "已完成步数 " 在X 120 Y 90 字号 15 颜色 ●
● 更新对象名 t 的基准点为 上 ▼
○ 对象名 shu 显示仿数码管字体 变量 实际步数 在:X 120 Y 143 字号 20 颜色
● 更新对象名 shu 的基准点为 上 ▼
S 对象名 jindu 显示填充矩形 在X 40 Y 180 宽 0 高 30 填充颜色
→ 対象名 kuang 显示矩形 在X 40 Y 180 宽 160 高 30 线宽 2 边框颜色
循环执行
如果 读取加速度的值 强度 - > 1.5 那么执行
将变量 实际步数 ▼ 赋值为 变量 实际步数 + 1
将变量 完成比例 🔻 赋值为 变量 实际步数 / 1000
⑤ 更新对象名 shu 的文本内容参数为 变量 实际步数
如果 变量 实际步数 > 1000 那么执行
● 更新对象名 jindu 的数字参数 宽 • 为 160
否则
● 更新对象名 jindu 的数字参数 宽 ◆ 为 变量 完成比例 * 160
等待 0.2 秒

2. 程序运行

检查行空板连接情况,点击运行,晃动行空板,观察实际步数和进度条的变化。



3. 试一试

现在你已经可以显示实际步数和目标达成进度了,请你试一试动手丰富健身打卡追踪器的页面和功能:

(1) 你可以参考下图,或者自己设计一个健身打卡追踪器背景页并适当调整文字内容、颜色等,让它 看起来更好看吧;



(2) 你可以参考下图设计一个目标设定页,使用加减号按钮修改目标步数值,确定按钮进入计步页面 开始计步。



<u>知识园地</u>

1. 加速度传感器

加速度传感器,顾名思义是用来测量加速度的传感器,因为加速度是有大小和方向的,所以加速度传 感器的数值也会有大小和正负变化,测量加速度的方向也可以有更多的情况,如竖直方向,水平方向。 行空板上内置了一个加速度传感器,它位于行空板背面。



行空板上的加速度传感器是一个三轴加速度传感器,它可以测量 X,Y,Z 三个方向的加速度。当行空板屏幕向上水平放置时,X 轴正方向为金手指一侧的方向,Y 轴正方向为 Home 按键一侧的方向,Z 轴垂直于板子,正方向为屏幕背面一侧的方向。



我们通常使用 读取加速度的值 × 来获得加速度传感器的数值,指令下拉菜单中的"强度",表示的 是 x、y、z 三轴加速度器数值的平方和再开方的结果。



加速度传感器的具体数值变化,是以重力加速度为参考的。以 x 轴加速度为例,我们将 x 轴加速度数 值显示在行空板屏幕上,观察数值变化。当你竖着拿行空板,金手指朝下时, x 轴方向上只受到竖直向下 的重力影响,屏幕上显示数值大小接近于 1;金手指朝上时,受到的竖直向下的重力会指向光线传感器一 边,屏幕上显示数值大小接近于-1。



当你将行空板水平放在桌面上,由于 x 轴方向没有重力加速度的影响,所以行空板屏幕上显示数值大小接近于 0。



2. 指令学习

本项目主要使用了学习了行空板加速度传感器指令,接下来我们就详细学习一下。



挑战自我

请你设计一个小球移动游戏,开始小球出现在上下左右任意一个位置,通过转动行空板,控制小球上下左右移动,比如,当小球出现在最右边,你需要向左转动行空板屏幕(如下图),控制小球向左移动。 最终,小球移到屏幕中间,游戏胜利。



132 / 242

随机整数,	并修改为1到4之间。	
	U.12345b 採曲 2 位/1致	
100 A	1 列表中的数的总和 • []	
<u> </u>		
SE	1 在 1 到 10 间取随机 整数 • 4 间取随机 整数 •	
4		
同日		

然后将小球的四个位置坐标分别与数字相对应,你可以参考下图来进行实现。



将变量	位置 🔻	赋值为	1	在	1	到(4	间取	随机	整数	•	1	
如果	变量	位置 =	1	₽ ₿		5							
将变量	₹ x •	赋值为	10										
将变量	≧ y •	赋值为	160										
否则 如	#	变量 位置) = (2) 38		ΞE						
将变量	∎ x •	赋值为	120					,					
将变量	≝ y •	赋值为	10										
否则 如	R (3	变量 位置) = (3	> ĦB		ΞE						
将变量	≣ x •	赋值为	230		·	~							
将变量	∎ y •	赋值为	160					_					
否则 🤆	9												
将变量	∎ x •	赋值为	120										
将变量	≝ y ▼	赋值为	310					,					
Ð													

第十课 肺活量测量仪

肺活量测量是用来检测心肺功能的项目,测量的时候通常要求人深吸一口气然后对着呼气口吹气,通 过检测呼出气体的体积反应心肺功能。今天我们就用行空板来模拟一个肺活量测量仪,一起来看看如何制 作吧。



任务目标

按下 A 键,开始测量,通过对着行空板呼气模拟肺活量的检测过程,同时进度条实时显示你的肺活量 达标情况;多次测量肺活量后,会显示最后三个测量数据。

注意: 肺活量测量仪只是对肺活量测量过程和效果的简单模拟,并不能真的用于测量肺活量。



<u>知识点</u>

- 1. 掌握程序中的列表的基础使用
- 2. 认识程序中的条件循环语句
- 3. 认识程序中的多分支条件语句
- 4. 掌握行空板上绘制线段的方法
- 5. 掌握行空板获取声音强度的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

肺活量测量仪的主要功能是通过统计呼气时长来显示肺活量情况,自动记录测量数据。接下来我们就 分两个任务来制作吧。

任务一:获得肺活量数据

在此任务中,我们将学习利用行空板麦克风,模拟肺活量的测量,同时通过进度条显示肺活量达标情况。

任务二:自动记录测量数据

基于已统计的肺活量数据,我们将学习列表的基本操作,记录并显示多次测量数据。

任务一:获得肺活量数据

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

肺活量测试的过程中需要手持测量仪并吹气,那么用行空板如何检测吹气,并获得肺活量数据呢,接 下来我们一起看看吧。

(1) 获得肺活量数据

(a) 判断吹气

我们是通过获取声音强度来判断人是否在吹气的,获取麦克风响度数值指令为 读取麦克风声音强度, 你可以在指令区"行空板"分类下的"板载传感器"里寻找该指令。

注意:行空板正对你时,麦克风位于行空板上方左侧(如下图)。关于行空板麦克风知识,参见"知识园地"。



Python主程序	研始			麦克风强	1度数(ā							
	显示文字	Θ	读取麦克	风声音强度	e ÆX	90	10) 守	号(15	颜色		
循环执行	ca. a.	x - x	5. C	e				<i>.</i>	2	3	125	124	
◎ 更新对象?	s (t1) 的攻	之本内容	参数为 () 读取	麦克风	声音强厉	E)						
等待 1 秒	()	咸慢数(直变化速	度									

注意:你可以使用远程桌面连接行空板,方便观察数据变化。



数据方面,在没有吹气时,你看到的是当前环境声音强度;接着,你可以对着行空板上方有话筒图标的地方吹气,观察声音强度数值,基本上都在40以上。

没有吹气,环境声音强度 吹气,声音强度明显增加



因此,我们可以通过声音强度数值来判断现在是否有人吹气。

(b) 获得肺活量数据

肺活量数据是通过人持续吹气获得,于是需要使用变量来记录持续吹气的状态,它的实现过程就变成 了开始吹气后,变量持续增加,直到吹气停止。也就是说,可以通过 重复执行直到 控制 肺活量数据变量, 持续增加, 直到声音强度小于 40 停止收集数据。

注意: 重复执行直到 是条件循环语句,它位于指令区的"控制"分类下。关于该指令的介绍,详见"知识园地"。



至于按下 A 键开始测量,本节课我们要使用到的是 按钮 A 被按下 ,它在指令区"行空板"分类下的 "板载传感器"里,如果按钮 A 被按下,那么开始测量肺活量。

如果 安田 A • 被按下? 那么执行	
将变量 肺活量数据 ◆ 赋值为 0	
重复执行直到 资源发表风声音强度 < 40	
将变量 肺活量数据 • 赋值为 变量 肺活量数据 + 1	
更新对象名 shu 的文本内容参数为 变量 肺活量数据	R
و	

(2) 显示达标情况

在实际测量肺活量过程中,根据不同的年龄对应着不同的及格标准。本课项目中使用进度条来显示肺活量情况,并加入"达标线"提示测试者是否达标,下图展示了进度条的组成情况。



达标线要使用行空板 "显示线段" 指令 对象名 显示线段起点 X0Y0 终点 X240Y320 线宽 1 颜色蓝 , 在 "显示矩形" 指令上方,你也可以通过 "线段"关键字找到它。



然后,通过进度条的组成和位置计算矩形相关参数和线段起终点坐标。需要说明的是,达标线可以根据比例来设置位置,比如表示 60%的刻度,参考下图即可完成计算。



现在你就可以根据肺活量数据和进度条宽度,调整进度条的变化,当然也可以适当加入一些提示内容, 完整参考代码如下:



4. 程序运行

点击运行,显示"肺活量数据"和0;按下A键,提示音响起,开始对着行空板麦克风吹气,观察进度条变化。



5. 试一试

我们现在已知进度条外框宽度为 200, 假设对应的肺活量数据为 2000, 请你动手设置当肺活量数据超 过 2000 时, 进度条不要超出外框。



任务二: 自动记录测量数据

1. 编写程序

在任务一中,你已经可以测量肺活量数据,那如何把测量过的数据记录并显示出来呢?一起来学习吧。 (1)记录测量数据

记录测试过的数据,可以使用"列表"变量来完成,你可以把"列表"理解成程序中的"表格",相关操作指令在指令区"<mark>列表</mark>"分类下。

注意:关于"列表"知识的介绍,参见"知识园地"。



把历史数据记录进"列表",需要先建立一个空的"列表"变量,即建立<mark>数据表变量</mark>并赋值为初始 化列表[] ,记得删去指令中的内容。



然后,在肺活量数据测量完成之后,将肺活量数据加入到列表。向列表加入数据的指令为 列表[]将 0 添加到末尾,使用时,将 <mark>变量数据表</mark> 和 <mark>变量肺活量数据</mark> 分别填入即可。



(2) 显示测量数据

首先先建立好三个显示测试数据的对象。

Θ	对象名 data1 显示文字 ••• 在X 130 Y 190 字号 15 颜色	
\bigcirc	对象名 data2 显示文字 🕐 * 在X 130 Y 220 字号 15 颜色	
Θ	对象名 data3 显示文字 ** 在X 130 Y 250 字号 15 颜色	

接下来显示的数据内容,其实就是把列表里的数据取出来,要使用指令 列表[]索引 0 的值 来完成, 需要说明的是,列表的索引就是列表的第几个元素,其中第一个的索引是 0。



显示测量数据,一共有三种情况,第一次测量只显示第一个数据,索引为0;第二次测量显示前两个数据,索引分别为0和1;三次及以上的测量则显示最后三个数据。

具体实现上,需要使用 <mark>列表[]的长度</mark> 指令,帮助判断当前列表有几个数据,并对应更新列表数据内容。





对于第三种情况,显示列表的最后三个数据,这三个数据的索引是 列表长度 分别减 1,减 2,减 3,你可以使用 列表[]的长度 指令获取并更新数据显示对象。



三种情况的判断需要使用可变分支条件语句 如果那么执行 (带"+"号),点击两次加号,即可完成 多条件语句的设置。



注意:关于多分支条件语句介绍,参见"知识园地"。

最后,完善三种情况的数据显示并给对应数据加上提示信息,本节课的案例就完成啦。完整参考代码 如下:


代码接下页

● 更新对象名 data1 的文本内容参数为	引表 变量数据表	索引 🗐 列表 变量 数据表 的长度 - 3 的值
更新对象名 data2 的文本内容参数为	• 列表 变量数据表	索引 [] 列表 变量 数据表 的长度 - 2 的值
○ 更新对象名 data3 的文本内容参数为	1 列表 变量 数据表	索引 🗐 列表 变量 数据表 的长度 - 1 的值

2. 程序运行

按下A键测量肺活量,多次测量后显示最后的三个测量数据。

	肺活量数据	
	604	
194	数据1:0	
180	数据2:0	
1844	数据3: 604	

<u>知识园地</u>

1. 程序中的列表知识

列表 (List) 是用来存储多个数据的数据类型,其中每个数据都有一个"索引"来表示它在列表中的位置。你可以类比生活中的表格,对列表进行读取、添加、删除、修改等操作,但不同于表格的是,对于有元素的列表,通常第一个数据元素索引为 0。

ł	表 格	列	」表
序号	内容	索引	数据
1	0	0	0
2	Mind+	1	"Mind+"
3	12.5	2	12.5
4	行空板	3	"行空板"

在 Python 中,列表是由"[]"构成,里面什么都没有,叫做"空列表";若有两个或以上的数据元素, 元素与元素之间需要用英文","间隔起来,如[0,"Mind+",12.5,"行空板"]。

另外,列表和数字都是 Python 中基础的数据类型,关于 Python 的其它基础数据类型,如字符串、字典等,我们将在后面的课程中继续介绍。

在图形化指令中,你可以在指令区"列表"分类下找到操作列表的指令。比如,创建列表指令初始 化列表[],向后添加元素指令列表[]将0添加到末尾等等,你可以自己尝试或者上网查询这些指令的含 义和用法。



2. 程序中的条件循环语句

条件循环语句是指 <mark>重复执行直到</mark>,它是循环结构语句中的一个,它的执行过程是,先判断"直到" 后面的条件是否满足,如果满足就停止执行"重复执行"里的语句,它的代码执行流程可以被描述成下面 右边的图片。



3. 程序中的多分支条件语句

多分支条件语句是 如果 那么执行 否则如果 那么执行 否则 语句,是条件判断语句中的一种。它通 常被用来做两个以上可能性的判断,你可以点击"+"号增加判断情况,也可以点击"-"号减少判断情况。 以三种情况的判断为例,下图左边的执行过程是:判断条件1是否成立,成立就执行代码语句1,否 则判断条件2是否成立,成立就执行代码语句2,否则执行代码语句3,它的代码执行流程可以被描述成 下面右边的图片。



4. 行空板板载麦克风

当行空板正对你的时候,行空板板载麦克风位于行空板上方左侧。它是电容硅麦克风,小巧且灵敏度 较高,可以用来获取环境中的声音。



行空板板载麦克风和加速度传感器、光线强度传感器一样,都属于板载传感器,所以你可以在"行空板""板载传感器"分类中找到读取麦克风声音强度指令,获取环境音量数值。



麦克风强度数值是一个百分比值,范围为 0-100。这个数值和电脑麦克风音量值一样,根据麦克风灵 敏度,录入的音频响度会有一个范围,最小的可录入音量对应 0,最大对应 100,如果超过最大音量依然 会按最大音量录入。需要说明的是,行空板板载麦克风,只可以通过百分比定性的分析音量的大小。



5. 指令学习

本项目主要使用了学习了行空板麦克风声音强度、列表操作等指令,接下来我们就详细学习一下。

运 读取麦克风声音强度	该指令用于获取板载麦克风接 收的声音强度数值。
按钮 A ▼ 被按下? ✓ A B	该指令用于判断行空板 A/B 键 是否被按下。
○	该指令用于在行空板屏幕显示 线段对象,在指令中可以设置 对象名、线段起终点坐标、线 宽以及颜色。
崑]初始化列表 ["Mind+", 0, 0]	该指令用于建立并初始化一个 列表。
月 列表 [] 的长度	该指令用于获取列表的长度。
圓 列表 [] 将 0 添加到末尾	该指令用于在列表的尾部添加 元素数据。
圓 列表 [] 索引 0 的值	该指令用于获取列表第"索引" 个数据。使用时需要写明列表 和索引内容,其中索引0为第 一个数据,索引不能超过列表 长度。
如果 那么执行 四回 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	该指令用于设置条件语句,点 击"+"号,可增加判断的情况; 出现多种情况后,点击"-"号, 可减少判断情况。

挑战自我

在本项目中你已经完成了肺活量测量和历史数据显示,但在班级场景下的肺活量测量通常需要记录每 个同学的测量数据,现在请你自己动手丰富你的肺活量测量仪,完成以下任务:

(1)为你的肺活量测量仪设计一个开始页面,此页面包括选择学号部分,点击"+""-"按钮选择学 号,以及操作文字说明,你可以参考下图设计;

地择学 4	- -
1	+
主行空 建后,大	反上部 声吹 ^在
-}	
ي ا	3
	选择学等

(2) 多次测量任务,按下A键,开始测量,进入测量页面,包含进度条和学号及对应历史数据;按下B键,屏幕返回学号选择页面,重新开始选定测量学生;

按下 A 提示音响起,开始测量 测量页面显示进度条和历史数据

	学号	为
	2	
按下日	键确认	上传数排
	学是1	1387
	学号2	327

按下 B 上传数据, 返回学号选择页



第十一课 loT 数据助手

你还记得学校体测时,体育老师是怎样记录体测数据的吗?老师一般有一个体测项目的名册表,每次 体测时,老师会按照名单点名分组进行测试,测试完成后将学生体测数据填写到表格中。这种传统记录数 据的方式,非常不利于老师开展工作。

物联网是一个非常强大的数据存储服务器,如果测试时,能够使用物联网来记录体测数据,体测完成 后,老师只需要登录物联网平台就可以查看数据,这样就可以更好地帮助老师记录数据。接下来,我们就 制作一款这样的 IoT 数据助手,来帮助老师吧!



任务目标

将检测到的肺活量数据,按照学号加肺活量数据的形式发送到 SloT 物联网平台。



<u>知识点</u>

- 1. 了解物联网的概念
- 2. 认识 SloT 物联网平台
- 3. 了解 MQTT 协议
- 4. 了解字符串

5. 掌握行空板物联网发送数据与接收数据的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://mindplus.cc/



<u>动手实践</u>

上一节课程中,学习了如何将检测到的肺活量数据存放在列表中。虽然列表里可以存储数据,但是行 空板一旦掉电,列表里的数据也不会被保存。如何才能有效地存储数据呢?接下来,从以下三个任务来实 现如何将数据存储到 SloT 物联网平台。

任务一: 行空板向 SloT 平台发送数据

学习使用行空板 MQTT 指令,在行空板上开启 SloT 服务器,并向 SloT 平台发送消息,登录 SloT 平台后,学会查看对应主题中的数据。

任务二:行空板接收 SloT 平台的数据

行空板接收 SIoT 平台的数据,并将接收到的数据在 Mind+终端中打印出来。

任务三: 行空板向 SloT 平台发送肺活量数据

将检测到的实时肺活量数据,发送到 SloT 物联网平台。

任务一: 行空板向 SloT 平台发送数据

1. 硬件连接

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

这个任务,主要是学习行空板如何向 SloT 物联网平台发送一条 "hello" 的文本消息,在发送消息之前需要做哪些准备。接下来,就一起来学习一下行空板开启和使用物联网需要哪些步骤吧!

注意:关于"物联网及物联网 SloT 平台"的相关介绍见知识园地。

(1) 检查 SIoT 是否开启 长按行空板的"Home"键进入菜单页,点击"应用开关",检查 SIoT 服务是否开启,如果显示"已

- ((• ♀■• 📑 ≍ 🄆 ((• ♀■•₩•= ♦ ((• 및■•₩ • = 🔆 退出 退出 HOME 键 Jupyter: 已启用 1-查看使用教程 2-切换运行程序 行空板 文件共享:已启用 3-应用开关 UNIHIKER **畢業共享**: 已启用 4-查看网络信息 长按HOME俚进入菜单 开机自启:已禁用 5-开关无线热点模式 Δ 0 1 2 3V GND 0 1 2 3V GND 0 1 2 3V GND
- 禁用",点击 SloT 选项,切换 SloT 状态为"已启用"。具体操作如图所示:

(2) 添加 MQTT-py 库

要向 SIoT 平台发送数据,要有发送相关指令才能实现。但是现在行空板的指令区中没有这样的指令, 该怎么办呢?点击"拓展",在官方库中找到"MQTT-py"并点击,完成添加。



(3) MQTT 初始化

指令添加完成后,要使用 MQTT 构建物联网,首先需要使用指令 初始化 MQTT,点击"设置"图标,将"SIoT 服务器"的内容修改为行空板连接电脑的默认 IP 地址"10.1.2.3"。



初始化设置完成后,MQTT 发起连接,使用指令 MQTT 发起连接; 连接成功后,需要保持连接,使

用指令 MQTT 保持连接永久.

要向 SIoT 平台发送数据,就需要先订阅要发送到 SIoT 平台的主题,使用指令 MQTT 订阅。需要特别说明的是,MQTT 订阅指令中填写的主题,并且固定的格式为"**项目 ID/设备号**"。



(4) 行空板向 SloT 物联网平台发送数据

完成了"MQTT 的初始化——发起连接——保持连接——订阅"这四个步骤后,接下来,就可以向 SIoT 平台发送数据了,使用 MQTT 发布指令。



每隔 5 秒,向 SIoT 平台发送一条 "hello" 的文本数据。

÷	Python主程序)	开始							
MOTT	初始化MQTT	0							
MOTT	MQTT发起连	接							
MQTT	MQTT保持连	接永久	а.	-					
MOTT	мотт订阅	"肺活量/	/1班				/		发送到的主题需要和订
循环	执行								<u> </u>
MQT	MQTT发布	hello	") 7		肺泪	量/	1班	•	
斜	寺 5 秒					发送	数据	t "hell	o″

4. 程序运行

运行程序,程序运行成功后,Mind+终端界面上打印"连接结果:连接成功"。

代日 ● 八田 ● 八田 ● 八田 ● 〇 八田 ● ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○	▲Kind+ 项目 • 教理 • 10.1.2.3 •	🕐 🗰 Ulizini sonari Livari (rymoneri 🌒 🍊
XA A	機块 代码	一 你止
MQTT-py PFR PR	文本 🔘 私執新設設	BW:1:W
Image: Imag	MQTT-py	14 while True: 15 siot.publish(topics"师问题/1批", data="hello") 16 time.slsep(5)
Pape MQTTACHABABANANIE Python主版形形的 Python主版形的 Python主版形的 Python主版形的 Python主版的 Python主版 Python主版的 Python主版的 Python主版的 Python主版 Python主版 Python主版的 Python主版的 Python主版 Python主版的 Python主版的 Python主版的 Python主版 Python主版的 Python主版 Python主版 Python主版 Python主版 Python主版 Python主版 Python	DI nerr 元组 III III III MQTU发展进程	17
・ ・ ・		 //> //> //> //> // // // // // // // //
MQTTALE (Topic) MQTTALE (Topic) MQTTALE (The Set Constraints) MQTTALE (The Set Constraint		>>> 已成功连接到10.1.2.3 Last login: Fri Seo 23 15:38:19 2022 from 10.1.2.101
Python Pyt		Prot\$unihiker:+ root\$unihiker:+ root\$unihiker:+ root\$unihiker:+ root\$unihiker:+ #目前建建目,2022,10,27,21,4,31此件目他
・ ・		中, 请稽后 項目上传成前, 开始运行 access control disabled, clients can connect from any bo
		st root#unihiker:-# od */root/mindplus/cache/新建项目-2022- 10-27-21-4-31"
(三) 连续结果上准接成功		root@uninikeri31# python /root/mindplus/cache/新建項目-2022-10-27-21-4 -31/:cache-file.py
		三 连接结果: 這種成功

程序设置每隔 5 秒给 SIoT 平台发送一条 "hello" 的文本数据。但是当数据发送成功后, 行空板屏幕 上没有任何反馈, 数据又应该怎么查看呢?

查看数据需要使用网页进入行空板服务页面,双击打开浏览器在地址栏输入"10.1.2.3"。

	日 新	「建标签页	Ę	× +
\leftarrow	С	ଜ	() 10.1.2.3	
1	1%	TIT	ALL N	MIX IN IVE I

进入行空板服务页面后,点击"应用开关",找到 SloT 并点击下方的"打开页面",进入 SloT 服务页。

◎ 行空板 - 网页菜单	× +		v = D
← → C (ⓒ 10.1.2.3			
			刷新
	SIoT		
	SloT是一个针对	学校场景的开源免费的M	IQTT服务器软件,可一键创建本地物联网服务
器 应用开关	器,摆脱联网困期	π.	
☆ 网络设置	运行状态 正在注		
L 文件上传	停止服	Slot	
		ture	
输入对应的账号和额	密码,点击登录。		
账号: siot			
密码: dfrobot			
SloT 项目	列表 设备列表	发送消息	
帐号(默认:siot)			
siot			
密码(默认:dfrobot)			
dfrobot			
登陆			
登录成功后,点击	"设备列表",找到	订阅的王题。	
SloT 项目列表	设备列表发	送消息	
△ 述:小友			
			400次 大次
	口	61	
项目ID	名称	奋 注	f架l'F
肺活量	1班		查看消息清空消息删除设备添加备注

点击"查看消息",就可以看到对应的数据了。

SIoT 项目列表 设	备列表 发送消息						
当前主题: 肺活量/1班							
发送消息 消息内容	发送						
(为消息加上->前缀代表此消息为	394指令消息,不会被存入数据库	。 例如"->off")					
开始时间	结束时间	100条 ~	查询	导出查询结果	隐藏/显示图表	自动刷新消息	
Торіс	消息		时间			、「毎回「私	左 夕 <u>新</u> 6336
肺活量/1班	helio		2023	2-09-23 15:56:50		母隔 5 秒	,有一余新的消息。
肺活量/1班	helio		2023	2-09-23 15:56:45			
肺活量/1班	helio		2022	2-09-23 15:56:40	/		
脉活漏/1班	hello		2022	2-09-23 15:56:35			
肺活量/1班	hello		2022	2-09-23 15:56:30			
肺活量/1班	hello		2022	2-09-23 15:56:25			
肺活量/1班	hello		2022	2-09-23 15:56:20			

5. 试一试

要实现按下 A 按键后,向物联网 SIoT 平台发送一条内容为 "hello" 的消息,程序应该怎么修改呢? 提示:使用<mark>当按键 A 被按下</mark>指令。



任务二: 行空板接收 SloT 平台的数据

1. 编写程序

上一个任务学习了如何向 SloT 平台发送数据,接下来,一起学习行空板如何接收 SloT 平台数据的方法,并且将该数据在 Mind+终端中打印出来。

(1) SloT 平台向行空板发送消息

SloT 平台查看消息界面上,有一个发送消息的消息内容框,在消息内容框里输入"你好,行空板!", 按下发送按钮,完成消息发送。消息发送成功后,Topic 里面会多一条"你好,行空板!"的消息。

SIOT 项目列表 輸主题:時活量/1班 送消息 (财子,行空板) 为消息加上->前缀代表比	设备列表 发送当息 消息内容 发送 肖息为纯指令消息,不会被存入数	^{容框中输入} "你好 ^{赌库。例如"->off"} 发	F, 行空	≊板!"。 1		
开始时间	结束时间	100条 🗸	查询	导出查询结果	見 障敵显示图表	自动刷新消息□
Торіс	消息				时间	
柿活量/1班	你好,行空都	7 !			2019-02-14 18:28:02	
闹舌量/1班	hello				2019-02-14 18:27:57	
ll 活量/1班	hello	消	息发送	成功	2019-02-14 18:27:52	
前舌量/1班	hello				2019-02-14 18:27:47	
応舌量/1 班	hello				2019-02-14 18:27:42	

(2) 行空板接收 SloT 平台发送的数据

行空板要接收来自 SloT 平台发送的消息数据, 需要使用指令当 MQTT 从 MQTT 主题接收到 MQTT



接收数据,首先需要判断数据是否是来自"肺活量/1 班"这个主题。需要注意的是,MQTT 消息的数据类型为字符串型,在进行 MQTT 消息判断时,需要将判断数据"肺活量/1 班"放入英文状态下的引号内。



MQTT 主题判断成功后,将接收到的 MQTT 消息在 Mind+终端中打印出来。



完整程序如下:



2. 程序运行

点击运行,程序运行成功后,Mind+终端界面上打印"连接结果:连接成功"的消息。然后在 SloT 平台的消息内容框中输入"你好,行空板!",点击发送按钮,Mind+中终端打印接收到的消息数据"你好,行空板!"。



SIOT 项目列表 设	备列表 发送消息					
当前主题:肺活量/1班 发送消息 你好,行空板! (为消息加上->前缀代表此消息;	发送为纯指令消息,不会被存入数据	库。 <mark>例如</mark> "->off")				
开始时间	结束时间	100条 🗸	查询	导出查询结果	隐藏/显示图表	自动刷新消息
Торіс	消息			E	时间	
肺活量/1班	你好,行空板!			2	022-10-11 20:09:34	£
肺活量/1班	hello			2	2022-10-11 20:09:30)
肺活量/1班	hello		息发送店	戎功 2	022-10-11 20:09:30)
肺活量/1班	hello			2	022-10-11 20:09:29)
文本 ・ Python主税序开始 デ ・ 並入論出 ・ 並入論出 ・ 並入論出 ・ 並入論出 ・ ジス時功能 ・ ジス時功能 ・ ジス時功能 ・ ジス時功能 ・ ジス時功能 ・ ジス時功能 ・ ジス時功能 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・「日日の」 ・ ジューロ・ ・ ジューロ・ ・ ジョー ・ ジョー ・ ジョー <t< th=""><th>Please ・ ・ ・</th><th>2/1班・ 12章 18校2部 MQ17行5 19前6章/1班・ 「考久代4</th><th></th><th>1 # -*- co 2 # MindPlu 4 # Pythen 5 inport 51 6 # FHOW 8 def on me 9 ff (m 10 11 12 13 siot.init 14 siot.com 15 siot.loop 16 siot.loop 16 siot.loop 16 siot.loop 16 siot.loop 17 siot.set, 17 siot.set, 17 siot.set, 18 siot.com 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.set, 17 siot.set, 17 siot.set, 17 siot.set, 18 siot.loop 18 siot.set, 17 siot.set, 18 siot.set, 19 siot.set, 19 siot.set, 19 siot.set, 10 siot</th><th><pre>ding: UTF-8 .*- is ot igg issage_callback(client, sg.topic == "Hold_/13 wint(msg.payload.decoor (client_id-"",server-" ect() (client_id-"",server-" ect() (client_id-",server-" hold_bus/cache/If t=/mindplus/cache/If t=/mindplus/cache/If</pre></th><th>userdata, msg): [*]; le(]) 18.1.2.3",port-1883,use liback) //獅~) ⑧ 這除輸出 防電量数据统计一任 防電量数据统计一任 腳活量数据统计一一</th></t<>	Please ・ ・ ・	2/1班・ 12章 18校2部 MQ17行5 19前6章/1班・ 「考久代4		1 # -*- co 2 # MindPlu 4 # Pythen 5 inport 51 6 # FHOW 8 def on me 9 ff (m 10 11 12 13 siot.init 14 siot.com 15 siot.loop 16 siot.loop 16 siot.loop 16 siot.loop 16 siot.loop 17 siot.set, 17 siot.set, 17 siot.set, 18 siot.com 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.loop 18 siot.set, 17 siot.set, 17 siot.set, 17 siot.set, 18 siot.loop 18 siot.set, 17 siot.set, 18 siot.set, 19 siot.set, 19 siot.set, 19 siot.set, 10 siot	<pre>ding: UTF-8 .*- is ot igg issage_callback(client, sg.topic == "Hold_/13 wint(msg.payload.decoor (client_id-"",server-" ect() (client_id-"",server-" ect() (client_id-",server-" hold_bus/cache/If t=/mindplus/cache/If t=/mindplus/cache/If</pre>	userdata, msg): [*]; le(]) 18.1.2.3",port-1883,use liback) //獅~) ⑧ 這除輸出 防電量数据统计一任 防電量数据统计一任 腳活量数据统计一一
			(=)	连接结果: 连接 你好,行空极! -	り消息	接收成功

3. 试一试

行空板能够接收到 SloT 平台发送的数据,并且可以将接收到的数据在 Mind+终端打印出来。接下来, 大家试一下将接收到的数据显示在行空板屏幕上。行空板屏幕显示效果如下图:

SIoT	项目列表	设备列表	发送消息			((•
当前主题:肺活 发送消息 (你好 (为消息加上->前	量/1班 F, 行空板! 边缀代表此消]	- 【发送】 ¹ 退,不会被存入数据]]]]]]]]]]]]]]]]]]]]		你好,行空板!
开始时间		结束	时间	100条	~	
隐藏/显示图表	自动属	新消息口				
Торіс		消息		时间		0 1 2 3V 3V GND
肺活量/1班		你好,行空	:板!	2022-09-	23 16:14	1:44

任务三: 行空板向 SloT 平台发送肺活量数据

1. 编写程序

第十一课的任务一中,完成了对肺活量数据的检测。接下来,就在该程序的基础上,添加一个选择学号功能和发送数据功能,通过"+/-"按钮来进行学号选择,按下A键开始检测肺活量,进度条反应肺活量的大小。按下B按键,将数据发送到SloT平台的"肺活量/1班"的主题中。行空板界面效果图如下:

选择	<u>; </u>
	+
肺活量 (量数据)
	2 3V GND

(1) 选择学号功能

选择学号功能,当按下"+"按钮,学号加一;按下"-"按钮,学号减一。这个功能中用到的<mark>文字对</mark> 象、<mark>按钮对象</mark>以及<mark>变量</mark>,都是学习过的知识,这里就不做详细地讲解了。

NOTT MQIT发起连接
MQTT保持连接永久
() MQTT订阅 "肺活量/1班"
将变量 肺活量数据 • 赋值为 ①
将变量 学号・ 賦值为 ①
○ 対象名 【1 显示文字 "选择学号" 在X 120 Y 30 字号 15 颜色 ●
○ 更新对象名 11 的基准点为 中心・
○ 対象名 xuehao 显示文字 ● AX 120 Y 60 字号 15 颜色 将交量 学号・ 賦值为 交量 学号・ 1
● 更新对象名 11 的基准点为中心・
○ 对象名 b1 增加按钮 **** 在X 40 Y 60 宽 40 高 30 点击回调函数 button_click1
○ 対象名 b2 増加按钮 "+" 在X 160 Y 60 宽 40 高 30 点击回调函数 button_click2
○ 对象名 【1】显示文字 ("肺活量数据") 在X (120) Y (120) 字号 (15) 颜色
○ 更新对象名 t1 的基准点为 中心・
◎ 更新对象名 shu 的基准点为 中心・
内象名 (jindu) 显示填充矩形 在X 20 Y 180 宽 变量 肺活量数据 高 30 填充颜色
🕞 对象名 (kuang) 显示矩形 在X (20) Y (180) 宽 (200) 高 (30) 线宽 (3) 边框颜色 🛑
🕞 对象名 11 显示线段 起点X 140 Y 180 终点X 140 Y 210 线宽 2 颜色 🛑
循环执行
如果 🔄 按钮 A・ 被按下? 那么执行
摄放音符 1 低 C/C3 1 → 拍
将变量 肺活量数据• 赋值为 0
重复执行直到 🕞 读取麦克风声音强度 < 40
将变量 肺活量数据 • 赋值为 变量 肺活量数据 + 2
○ 更新对象名 shu 的文本内容参数为 变量 肺活量数据
如果
○ 更新对象名 (jindu) 的数字参数 宽 - 为 (200)
● 更新对影名 Jindu 的数字参数 宽 · 为 变量 肺活量数据 / 10

(2) MQTT 准备工作

实现行空板与 SIoT 物联网平台联网的第一步,先完成 MQTT 的准备工作。



(3) 发送肺活量数据

按下 B 按键,将检测到的肺活量数据发送到 SloT 物联网平台的"肺活量/1 班"主题中。判断 B 按键 是否按下,使用指令<mark>当按键 B 被按下</mark>;向 SloT 平台发送数据,使用 MQTT 发布指令。

当按键 B → 被按下	发送肺活量数据。
MQTT发布 变量 肺活量数	数据 到 肺活量/1班 "
	发送到的 SloT 主题。

增加一个数据发送成功的反馈,数据发送成功后,蜂鸣器会响起,并清除检测到的数据和进度条。



(4) 发送带学号的肺活量数据

上面程序中使用 MQTT 发布指令,发布的数据是肺活量数据,没有学号,即使 SloT 平台接收到了数据,也不知道该数据是哪位同学的。所以在发送数据时,最好带上学号标识,将学号和肺活量数据合并后 一起发送到 SloT 平台。合并字符使用文本指令区的<mark>合并字符</mark>指令。

注意:关于"字符串"的相关介绍见知识园地。



修改 MQTT 发布指令中的内容为合并后的学号和肺活量数据。

\bigcirc	当按键 B	• 被按	T	8 - 31										
MQTT	MQTT发布	i 😫 f	}并(变量 学	号	变量	肺活	量数	据	到("肺	活量	/1刊	
$\overline{\Theta}$	播放音符(1 低 C/	C3	1 • ∦	1			4			4		ŝ	
	1秒	14 - 14	-	 (a) 										
0	更新对象名	shu	的文	本内容	参数ソ	0") נ	"							
6	更新对象名	(jindu	的數	如字参数	女 宽	• *	0							

完成程序如下:



2. 程序运行

点击运行,程序运行成功后,通过行空板屏幕上的"+/-"按钮,选择对应的学号。然后按下A按键, 开始测量肺活量,测量结束后,按下B按键,将带学号的肺活量数据发送到SloT平台。



3. 试一试

为了避免重复多次的测量,当一个班的学生都测试完成后,行空板就不能再向 SloT 平台发送肺活量数据了。接下来我们以一个班十个学生为例,当这十个学生都测试完成后,就断开 MQTT 连接, SloT 平台不再接收行空板发送的消息。

提示: 十个学生测试完成后, 使用指令 MQTT 断开连接



知识园地

1. 什么是物联网

物联网 (Internet of Things,简称 IoT) 是借助互联网、传统电信网等,让具有一定功能的设备实现 互联互通的网络。比如在智能家居场景下,通过房间里的无线网,将灯、风扇、摄像头等设备与手机相连, 这就形成了万物互联。



物联网的原理是什么呢?

一个完整的物联网系统包含三个部分,分别是服务器、智能终端、移动终端。以智能家居场景为例, 我们可以通过手机控制和查看灯、风扇、摄像头这些设备的状态,手机就是物联网系统中的**移动终端**;而 那些可以直接连接网络的灯、摄像头等智能家电就是物联网系统中的**智能终端;**另外,在物联网系统中, 移动终端和智能终端不能直接传输数据,它们之间还需要一个枢纽,用来存储和收发数据,这就是**服务器。**



智能终端

行空板也可以构建这种有智能终端、移动终端、服务器的智能家居系统,如果我们手上只有一块行空板时,可以让这块行空板做智能终端,连接灯、风扇、摄像头这些设备,使用行空板自带的 SloT 平台做物联网服务器端,我们的手机或者电脑做移动终端。



需要说明的是,本节课项目中的行空板,即是智能终端又是服务器,而电脑则对应移动终端。其中电

脑是使用 USB 线这种有线连接的方式去查看服务器数据的。



其实,移动终端访问服务器数据还可以使用无线连接方式,这种方式是通过行空板自带的WIFI模块功能实现,详细操作,你可以进入行空板官网"物联网项目"进行查(<u>https://wiki.unihiker.com/iot_project</u>)。

2. 物联网 SloT 平台

物联网中的服务器是通过物联网平台(又称 IoT 平台)实现的,常用的 IoT 平台有 Easy IoT、SIoT。 要实现物联网中的设备通信,通常采用 HTTP 协议或者 MQTT 协议。这节课使用到的是物联网 SIoT 平台, 行空板与 SIoT 平台之间的通信采用 MQTT 协议,接下来,一起来了解一下什么是 SIoT 平台,什么是 MQTT 协议吧!

SIoT 是行空板自带的物联网服务器平台,你可以通过有线或无线的形式快速创建本地物联网服务器。 SIoT 重点关注物联网数据的收集和导出,是采集科学数据的最好选择之一。



3. MQTT 协议

什么是 MQTT 协议?

MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输协议) 是一个基于客户端 - 服务器的消息发布/订阅传输协议。

MQTT 协议是轻量、简单、开放和易于实现的,这些特点使它适用范围非常广泛。



MQTT协议在行空板与 SloT 平台中起到一个什么作用呢?

我们可以将 MQTT 协议理解为行空板与 SloT 平台通信的桥梁,在行空板和物联网 SloT 平台通信过程中,MQTT 协议有三种身份:发布者(Publish)、代理(Broker)(服务器)、订阅者(Subscribe)。其中,消息的发布者和订阅者都是设备端(行空板),消息代理是服务器,消息发布者可以同时是订阅者。

换种简单的理解方式,我们可以将这个消息代理理解为一个群,如果你想在这个群里发消息,首先你 得进入这个群,才能发送消息,而进群的方式就是 MQTT 协议。进群后,发消息这个人就是发布者,群就 是服务器,其他群成员就是订阅者,你发的消息自己也能看到,因此发布者也可以是订阅者。

4. 什么是字符串



什么是**字符串**?字符串是一种数据类型, Python 中有六种标准的数据类型, 分别是字符串(文本)、 数字、列表、元组、集合、字典。

大家注意一下,上面的字符串指令里都有一对引号。我们可以将字符串理解为放在引号里面的一系列 字符,通俗的理解,就是由标点符号、字母、数字、文字组成的单词或句子就是字符串。其中组成字符串 的标点符号、字母、数字、文字就是字符,字符串就是字符的集合。 现在你知道为什么任务二中,判断 MQTT 主题时,需要将"肺活量/1班"放入引号内了吗?



因为 MQTT 主题的数据类型是字符串,如果不将"肺活量/1班"放入引号内,两个数据就不能作比较,因为数据类型不一样。如果觉得不好理解,简单地举个例子,现在有两组数据,一组是23,另一组是 "23"。第一组数据的数据类型是数字型,第二组数据类型是字符型,两组数据的类型不一样,不能做比较。只有当两组数据类型一样时才可以作比较,所以"肺活量/1班"需要放入引号内。



为什么要用<mark>合并字符</mark>指令呢?举个例,现在有"你"、"好"","、"行"、"空"、"板"、"!" 这七个字符。如果不组合,那就是单个的字符,而<mark>合并字符</mark>指令的作用就是将这些字符组成一个完整的句 子"你好,行空板!"。

	你	ţ	好	,	行	空][板	!	
--	---	---	---	---	---	---	----	---	---	--

5. 指令学习



мотта мотта • topic •	该指令用于订阅 MQTT 服务器的设备主题。 主题名 (topic) 的格式为 "项目 ID/设备号"。
MQTT发布 • hello • 到 • topic •	该指令用于向 MQTT 服务器的对应主题发送 数据。
	该指令用于接收 SloT 物联网平台的数据,指
ACTT 当MQTT从 MQTT主题 接收到 MQTT信息	令中包含 MQTT 主题和 MQTT 消息。
🗐 合并 ("apple") ("banana")	该指令用于合并两个及以上的字符。

挑战自我

学号和肺活量数据合并后,学号确实发送成功了。如果学号两位数,学号和肺活量数据合并后,很难 分清数据。如果在学号与肺活量数据中间添加冒号(:)将学号和肺活量数据隔开,发送的数据形式为:学 号:肺活量数据,这样就可以分清数据了。接下来,大家一起来试一下吧! 提示:添加冒号使用合并字符指令,在合并字符指令中嵌套合并字符指令。

oT 接收到的消息如下	图:	
opic	消息	时间
标量/1班	13: 288	2022-09-23 15:59:36
話量/1班	12: 376	2022-09-23 15:59:28
活量/1班	3: 480	2022-09-23 15:58:16
活量/1班	2: 474	2022-09-23 15:58:05
活量/1班	1: 542	2022-09-23 15:57:57

第十二课 肺活量数据可视化报告

经过上节课的学习,我们已经可以把测量好的肺活量数据通过 SloT 存储下来了,今天我们就来对这些数据进行进一步的分析。本项目在 DF 球星寻找了 10 个小伙伴使用我们的肺活量检测仪并从 SloT 服务器 下载并处理好所有数据,10 个小伙伴测出的肺活量数据如下表:

编号	1	2	3	4	5	6	7	8	9	10
性别	男	女	男	女	男	女	男	女	男	女
肺活量	1675	1028	1205	1165	1985	1305	1897	1195	1498	995

通过观察上面的数据表,你能获取哪些信息呢?你能看出谁的肺活量最高吗?想一想,性别与肺活量 有关系吗?让我们带着这些问题,一起来探索一下吧!

任务目标

在行空板上以折线统计图的形式,显示肺活量数据,并且使用柱状统计图统计男女生肺活量的平均值。



知识点

- 1. 了解函数概念
- 2. 了解 Python 中的循环语句
- 3. 了解程序中的映射语句
- 4. 学习利用线和形状绘制统计图表实现数据可视化

材料清单

硬件清单:



<mark>软件使用:</mark> Mind+编程软件 x1 下载地址: <u>https://mindplus.cc/</u>



动手实践

这课主要学习如何使用折线统计图与柱状统计图实现肺活量数据可视化, 接下来就从下面两个任务来 学习如何实现数据可视化。

任务一: 绘制折线统计图

对肺活量数据进行处理后,在行空板上的坐标系图中描点并利用线段连接各个数据点,最后绘制成折 线统计图。

任务二:绘制柱状统计图

在任务一的基础上,新增计算男女生肺活量平均数的功能,并且将肺活量平均数分别进行绘制,最后 形成柱状统计图。

任务一: 绘制折线统计图

1. 硬件连接

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+,按照下面图示完成软件准备过程。



3. 编写程序

开始编写程序之前,我们先来分析一下,要将肺活量数据绘制成折线图,首先需要一个横坐标为编号, 纵坐标为肺活量数据的**直角坐标系。**然后将肺活量数据进行处理后在坐标系上进行**描点**,最后将各个数据 点进行**连线。**



(1) 直角坐标系

直角坐标系是提前做的图,因此只需要将这个坐标图从文件种加载到行空板的文件系统中。



使用显示图片指令,设置坐标轴图片在行空板 X0Y0 处开始显示。

	國名	aixs	副示	图片	坐标	1.pr	ng ")在	x 🕧) Y	0
循环执	行				 						

(2) 描点

首先,需要将检测到的肺活量数据表中的数据,放入肺活量列表中。新建一个名为<mark>肺活量数据</mark>的变量

175 / 242

			/ weren							
编号	1	2	3	4	5	6	7	8	9	10
性别	男	女	男	女	男	女	男	女	男	女
肺活量	1675	1028	1205	1165	1985	1305	1897	1195	1498	995

用来存放列表,并初始化列表,初始化列表的数据为下面表格的肺活量数据。

Pyt	thon±	程序)	T婝	ì																					
	\$ 2 (aixs	显	示图》	1 (" 坐板	<u>⊼</u> 1.j	ong	•) ā	x (0	Y 🌔	0	÷	Q	÷	ŝ,	2	÷	3	2	2	9	÷	
将变量	肺活量	数据	•	赋值	沩	ġ,	Ø	始化	例表	(167	'5,1()28,	120	5,11	65,	1985	,13(05,1	897	,119	95,1	498,	995]
盾环执诉	7	1	11			1			11	17	111		11		1.5		1	11				12	1.5		
		1.1																							

数据被存入列表后,怎样才能将列表的数据一个一个取出来并在行空板的坐标系图上进行描点呢?使用 for 循环指令,取出列表中的肺活量数据,并新建一个<mark>肺活量值</mark>变量用来存放从列表中取出来的肺活量数据。





肺活量数据被取出来之后,就要在行空板上的坐标图中进行描点。使用<mark>填充圆形</mark>指令,在行空板上进行描点。如何才能确定填充圆形的 XY 坐标呢?以第一个数据 1375 为例,结合下面坐标轴的图来分析一下。



首先来确定填充圆形指令的 X 坐标, 获取的第一个肺活量数据, 描点的 X 坐标在横坐标第一个刻度的 位置, 因为每个刻度之间的距离为 20, 所以描点的 X 坐标为 19+20, 新建一个<mark>变量 X</mark>, 并设置初始值为 19。

对应的Y坐标表示肺活量数据,但我们可以直接使用肺活量数据作为Y坐标吗?显然不可以,因为肺活量数值远超过屏幕坐标范围,数值大小和Y坐标大小也呈相反的关系。为了解决这些问题,我们可以使用数字类型中的<mark>映射</mark>指令,将肺活量值直接映射到行空板的Y坐标轴上。现在你可以新建一个<mark>变量Y</mark>,用来存放映射后的肺活量数据,填充圆形指令的Y坐标,就是映射后的变量Y。

注意:关于"映射"的相关介绍见知识园地。

要映射的肺活量数据范围 9. 2100 到 150,10] 9. 10 时间 150,10] 9. 10 时间 150,10] 9. 10 时间 150,10]	0 肺活量数据 2100 150 10 行空板 Y 坐标范围
🕹 Python主程序开始	
○ 対象名 aixs 显示图片 *坐标1.png 在X 0 い	Y 🕕
将变量 肺活量数据 • 赋值为 🔋,初始化列表 [1675	5,1028,1205,1165,1985,1305,1897,1195,1498,995])
将变量 X → 赋值为 19 将变量 Y → 赋值为 0	
使用 my variable • 从范围 0 到 9 每隔 1	
将变量 肺活量值 • 赋值为 📴 列表 变量 肺活量数	数据 索引 变量 my variable 的值
将变量 X ▼ 赋值为 变量 X + 20	
将变量 Y → 赋值为 🕕 映射 变量 肺活量值 从[0,2100]到[150,10]]
○ 对象名 circle 显示填充圆形在X 变量 X Y 变	量 ¥ ¥径 2 填充颜色 ●
循环执行	

(3) 连线

上面的程序完成了将肺活量数据进行描点,接下来,就是用显示线段指令,将这些数据点用线段连接 起来。首先需要确认线段的起点 XY 坐标,以及终点的 XY 坐标。当行空板上显示第一个点时,不需要连线。 当行空板上显示第二点时,就需要将第一个点和第二点用线段连接起来。当变量 X>39 时,新建一个 变量 Y1,用来获取第一个点的变量 Y 值。因此,起点的 XY 坐标为:(变量 X-20,变量 Y1);终点的 XY 坐标为:(变量 X,变量 Y)。

Python主程序开始	Á												
🕞 対象名 aixs 显	示图片("坐标1	.png ") 在>	(() Y (0									
将变量 肺活量数据•	赋值为 🔋 1	叨始化列表([1675,1	028,120)5,1165	5,1985,1	305,1	897,11	95,14	98,9	95]		
将变星 X • 赋值为	19	4.4.8	191-8	x = x	a a	e e s	- 6	R 8	- 21	5	a a	G.	
将变量 Y • 赋值为	0												
将变量 Y1 • 赋值为	0												
使用 my variable •	从范围 🕕 到	9 毎隔 (1	5.5		8.1		1.1					
将变量 肺活量值•	赋值为 🔋 列	表 变星	标舌星数据	日 索引	安里	my varia	able	的值					
将变星 X · 赋值为	· (支量 X) + (2	0											
将变星 Y • 赋值为	1 映射 变	副 肺活量值	从(0	, 210)]到[150	10	1)					
内 対象名 circle	显示填充圆形在	X 安全 X	Y (变重	Y +7	2 I	直充颜色							
如果	39 那么执行		1000										
分 对象名 line	显示线段起点)	(_{变星} X	- 20 Y	变量)	11 総	ax 👳	₽X	Y 委員	Y	進売 (1	颜色	
		10 10 10											
将变重 Y1• 瓢值5	为 受里 Y												
等待一秒	- 10 - 10 - 16 - 16 	18 18 18 1		- 31 - 21	3.3	18 18	~	8.8		8	8 8	2	2
循环执行	- xe - xe - xe	- 01 - 01 - 01	10.0	- 10 - 20	(A) (A)	. a. a		00 R	10	×.	ai ai		

(4) 创建折线图函数

程序中,绘制折线图描点和连线的功能,都是在 Python 主程序开始下面实现的。现在 Python 主程 序开始下程序太多了,看起来也很乱。如何才能做到让 Python 主程序开始下面只保留一些初始化的程序, 还要让绘制折线图的功能不变呢?

点击模块中的"函数",然后点击"自定义模块",修改函数名为:绘制折线图。点击完成,完成<mark>绘</mark> 制折线图的函数<mark>创建。</mark>

注意:关于"函数"的相关介绍见知识园地。

Minde 项目 + 故屋 + 10.1.2.3 +		主动模式 上传模式 Python版式 🕘 🕚
模块 代码		
		自动生成
自定义模块 Python 教技 使 Python主招序开始	#M—个自己X.帮助 •	1 # -*- coding: UTF-& -*- 2 # MindPlus 4 # Python 5
7.祖 第2. 第2. 第2. 第2. 第2. 第2. 第2. 第2.	和自己的	
 注合 > 文件功能 ● 以 (*UTF6*) 拾式 该现・株 ● 以 (*UTF6*) 拾式 该现・株 	anne ya	
	4.00 A.00	② 終端(行空板) ● 清除输出
Python	(2)	务一正在上传中, 诸相后 项目上传成功, 并航运行 root@unihikerr+# access control di sabled, clients can connect from a
 ● ####示 ● 対象名 ● 協示文字 ●打증 	(Q) (E)	ny host rootBunhiker:-# cd "/root/mindplu #/cache/給制折截周-任务 -* rootBunhiker:-/mindplus/cache/給 制折截周-任务 * # pythen /root/mind churche/納制折載用 低多 - (root
	书包	e-file.py


函数创建完成后,脚本区会出现一个新的定义绘制折线图函数指令。

然后将 Python 主程序开始下实现描点、连线的程序,用<mark>绘制折线图函数名</mark>来代替,这样 Python 主 程序开始下面的程序指令就很简洁了。描点和连线的程序都放到<mark>绘制折线图函数</mark>下。但是,需要注意的是 绘制折线图函数中使用到的变量,都需要在函数下定义为<mark>全局变量</mark>。完整程序如下:

Python主程序开始
○ 対象名 aixs 显示图片 ●坐标1.png ● 在X 0 Y 0
将变星 肺活星数据• 赋值为 🛐 初始化列表 [1675,1028,1205,1165,1985,1305,1897,1195,1498,995])
将变量 X → 賦值为 19
将变量 Y→ 赋值为 ①
将変量 Y1 ・ 赋值为 00
绘制折线图 调用绘制折线图函数
循环执行
• Provide the second s second second seco
定义 绘制折线图 工工工工工工工工工工工工工工工工工工工工工工工工工工工工工
全局global 变星 X
全局global 变量 Y
全局global 变量 Y1 And
全局global 变量 肺活量数据
使用my variable · 从范围 ① 到 9 每隔 1
将变星 肺活量值• 赋值为 💼 列表 变星 肺活星数据 索引 变星 my variable 的值
将变星 X → 赋值为 变星 X + 20
将变量 Y • 赋值为 📙 映射 变量 肺活量值 从[0, 2100]到[150, 10]
○ 対象名 circle 显示填充圆形在X 变量 X Y 变量 Y 半径 2 填充颜色
如果
将变量 Y1 ・ 贓值为 变量 Y

4. 程序运行

点击运行,程序运行成功后,行空板界面上显示绘制的肺活量折线图。



任务二: 绘制柱状统计图

1. 编写程序

大家知道为什么要使用柱状统计图,来统计男女生肺活量的平均数吗?因为柱状统计图可以一眼看出数据大小,还能对比各个数据之间的比例差异。接下来,使用柱状统计图来看看性别与肺活量数据之间是 否有关系?

开始编写程序之前,还是来分析一下,要绘制一个男女生肺活量平均数的柱状图,需要实现哪些功能。 首先,需要统计男女生肺活量的平均数,然后将平均数映射到坐标图上,最后在柱状图上标出对应的男女 生平均数值。



(1)新建<mark>绘制柱状图函数</mark>,并在 Python 主程序开始</mark>下调用这个函数名,接下来所有绘制柱状图的程序 都编写在<mark>绘制柱状图函数</mark>下。

や Python主程序开始	1																			
○ 対象名 aixs 显	示图片 🥊	坐标2	2.png ") 在)) Y	0													
将变量 肺活量数据•	赋值为		初始化》	刘 表([1	675	,102	8,12	205,	116	5,19	85,	1305	,18	97,1	195	,149	8,99	95]	
绘制折线图																				
绘制柱状图	定义	绘制柱	状图																	
循环执行																				

(2) 计算女生肺活量的平均数

编号	1	2	3	4	5	6	7	8	9	10
性别	男	女	男	女	男	女	男	女	男	女
肺活量	1675	1028	1205	1165	1985	1305	1897	1195	1498	995

从上面的数据表中,可以很容易看出,编号为2、4、6、8、10的为女生肺活量,剩下的为男生肺活

量。当肺活量数据都存放在列表中后,如何才能将男生肺活量和女生肺活量数据分开呢? 新建一个<mark>女孩肺活量列表</mark>,并<mark>在 Python 主程序开始</mark>下,初始化女生肺活量列表。

Python主程序开始
将变量 肺活量数据 • 赋值为 🗐,初始化列表 [1675,1028,1205,1165,1985,1305,1897,1195,1498,995])
将变量 女生肺活量 - 赋值为 💼 初始化列表 [1]
将变量 X - 赋值为 19
将安星 Y · 赋值为 0
将变量 Y1 · 赋值为 0
绘制折线图 定义 绘制柱状图
绘制性状图 全局global 变量 女生肺活量
循环执行 一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一
使用 <mark>for 循环</mark> 指令,从肺活量数据列表中按索引 1、3、5、7、9 取出数据存放在女生肺活量列表中。
定义 绘制柱状图 变量 my variable 的范围为
全局global 变量 女生肺活量 间隔 2,从1开始,my variable 每次增加 2, 这样就可以获取列表中编号为 1、3的数 使用 my variable • 从范围 1 到 9 產 區 2
副,列表 变量 女生肺活量 将 📓,列表 变量 肺活量数据 索引 变量 my variable 的值 添加到末尾

女生的肺活量数据索引出来后,如何才能统计女生肺活量的平均数呢?使用数字类型中的<mark>列表中的平</mark> 均数<mark>指令,并将女生肺活量列表放入求列表平均数据的指令中,计算女生肺活量列表中的肺活量平均数。</mark>



(3) 绘制柱状图

在绘制柱状图时,使用显示线段指令,设置线段的宽为 20。现在就要先确定线段起点的 XY 坐标以及 终点的 XY 坐标。线段的 X 坐标使用来指定柱状图的显示位置,因此线段的起点和终点 X 坐标,可以直接 设定为:110。



线段的终点 Y 坐标也可以确定,因为柱状图的横轴对应在行空板 Y 的位置为 300,所以线段的 Y 坐标也为 300。使用<mark>映射</mark>指令,将女生平均数映射后的值赋值给<mark>变量 Y2</mark>,线段起点的 Y 坐标就为变量 Y2。



男生肺活量平均值的柱状图绘制用同样的方法即可,需要注意的是,男生使用 for 循环取肺活量数值的范围为: 0-9。



(4) 显示平均值

柱状图绘制成功后,使用<mark>显示文字</mark>指令,在柱状图的上方显示对应的男女生平均数值。



完整程序如下:



2. 程序运行

点击运行,程序运行成功后,行空板上先绘制肺活量数据的折线统计图,然后再绘制男女生肺活量平 均数的柱状统计图。显示效果如下图:



3. 试一试

接下来,大家试一下用任务二中的方法,将总的肺活量平均数也绘制在柱状图中吧! 绘制好柱状图之后,再完善一下柱状统计图中的细节。参考图如下:



知识园地

1. 什么是数据可视化

数据可视化的意义是帮助人从大量数据中高效、快速获取有效信息,信息的质量很大程度上依赖于其 表达方式。数据可视化的意义,就是借助图形化的手段,让数据变得更加直观。



图表是"数据可视化"的常用手段,其中折线统计图与柱状统计图最为常用。折线统计图不仅能够看 出数量多少,还能看出数量增减的变化趋势。柱状统计图可以一眼看出数据大小,还能对比各个数据之间 的比例差异。当然还存在很多不同形式的图表用来展示数据的其它特性,有兴趣你可以自己查询研究一下。



2. 认识函数

什么是**函数**?函数,就是将一段经常使用的程序封装起来,在需要使用这个程序的位置调用这个函数即可,并且这个函数可以多次调用,这样做可以减少重复代码。在一个较大的程序中,一般会将程序分为多个程序块,每个程序块实现指定的功能,我们可以将这些实现指定功能的程序块定义为函数。

比如,在程序中写好了一段程序,这段程序用于实现一个特定的功能。问题来了,如果下次需要实现 同样的功能,难道要把前面程序再重新编写一次?如果这样做效率实在太低了,这意味着每次当程序需要 实现同样功能时,都要将前面的程序复制一次。正确的做法是,将实现特定功能的程序定义成一个函数, 每次当程序需要实现该功能时,只要执行(调用)该函数即可。



其实,函数的本质就是一段有特定功能、可以重复使用的程序,这段程序已经被提前编写好了,并且 为其起一个"好听"的名字。在后续编写程序过程中,如果需要同样的功能,直接通过名字就可以调用这 段程序。

3. 认识 for 循环

在项目程序中,我们使用了 for 循环指令,如下图。为什么这条指令叫 for 循环呢?



因为该指令生成的 Python 代码中,有关键词 for,因此称为 for 循环。

这条循环指令的实现原理是什么呢? 该指令调用的是 Python 内置的 range() 函数创建一个整数序 列。指令中,需要设置三个参数,开始值、停止值、步长。这个指令可以实现循环从开始值计数到达停止 值,一旦到达停止值就立刻退出循环。也就是说,这个循环指令是按照变量 my variable 的变化控制循环 次数。



什么是 **range () 函数**? 我们将这条指令结合 Python 代码来分析, 其中 my variable 是变量, range () 是函数, range 表示在指定范围内, 等间隔排列的整数集合对象。例如, 现在设定指定开始值(1), 停止值(9), 以及步长(2), range() 函数就是将 1-9 的数字进行等间隔排列 {1、2、3、4、5、6、7、8、9}, 步长就是控制变量 my variable 从 1 开始每次递增 2, 从而实现将肺活量列表中编号为 1、3、5、7、9 这五个数据取出来放入女生肺活量列表中。



细心的同学肯定发现了,在指令中我们设置的范围是 1-9,步长为 2,但是生成的 Python 代码的是 range (1,10,2),范围变为了 1-10,这是生成的 Python 代码出错了吗?

Python主程序开始	4 # Python 5	
使用 my variable - 从范围 11 到 9 5 5 6 2	<pre>6 7 for my_variable in range 1, 10, 2): 8 pass 9</pre>	

注意:这是因为 Python 中 range()函数实现的是从开始值计数到达停止值-1,就退出循环。我们循环指令要实现从开始值计数到达停止值,就退出循环,所以生成的 Python 代码是在指令停止值的基础上+1。

4. Python 中的循环语句

Python 中,除了 for 循环还有 while 循环。Mind+控制类型指令中,有以下五种不同的循环指令,接下来,将通过循环指令结合它们生成的 Python 代码来了解这几种循环指令的用法。



通过循环指令生成的 Python 代码,发现前三条循环指令的 Python 代码中都有关键词 for,最后两条循环指令的 Python 代码中都有 while。也就是说,在 Python 代码中,循环结构可以分为 for 循环和 while 循环两大类。指令说明可以参考下列表格:

循环语句	图形化指令	Python 代码	说明
	使用 my variable + 从序列 [0,1,2]	for my_variable in [0,1,2]:	该指令可以让列表、元组、字符串
		pass	中的每个元素都执行一次, 然后在
			指令中对每个元素都进行访问和
for 循环			操作。
		for index in range(5):	该指令的作用是指定指令下的程
	重复执行 5 次	pass	序执行多少次。
		while not (False):	该指令执行过程是, 先判断 "直到"
while 循	重复执行直到	pass	后面的条件是否满足,如果满足就
环			停止执行"重复执行"里的语句。
			表达式一直为假时,程序不会被停

		止执行,这种循环就称为死循环。
	while True:	该指令的作用是为了让 Python 主
循环执行	pass	程序开始后,程序一直保持运行状
		态,属于一种特殊的死循环。

5. 认识映射

映射,是用来把一个范围的数据向另一个范围等比例转换的指令,你可以把它理解为一个比例尺转化 工具。比如,下面映射指令的意思,就是将 0-100 之间的数字 20 变成 0-2 之间的数字 2。



任务一中的映射程序,将具体的肺活量值映射为行空板的Y坐标值。这样,就可以根据映射后的值在 行空板的Y坐标上进行描点。



6. 指令学习





挑战自我

这课中,我们学习了如何使用线段和图形绘制肺活量数据的折线统计图与柱状统计图。接下来,大家 试一下在上面程序的基础上增加两个按钮。例如:按下"折线图"按钮,行空板上开始绘制肺活量数据的 折线图;按下"柱状图"按钮,行空板上绘制男女生肺活量平均数的柱状统计图。



第十三课 IoT 课堂互动答题系统

随堂小测作为非常好的检测学生知识掌握情况的方法,被老师们广泛使用。今天我们就利用行空板物 联网服务,制作一个课堂互动答题系统,实现老师随时都可以下发题目,获得学生对知识点的掌握情况。



任务目标

- 教师端:负责发送题目和收集学生回答的答案。
- 学生端: 当接收到教师发送过来的题目, 学生作答。



学生端

<u>知识点</u>

- 6. 了解程序中的线程知识
- 7. 了解行空板无线网络服务知识
- 8. 了解两块行空板构建简单物联网系统的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



<u>动手实践</u>

课堂互动答题系统由两部分组成:学生端和教师端。学生端可以在有限时间内答题,教师端能够远程 查看学生答题情况,接下来我们就分三个任务来看看如何完成吧。

任务一: 倒计时答题

在此任务中,我们将学习行空板开启线程方法,同时利用行空板文字和按钮对象完成学生端限时答题 的功能。

任务二: 教师端和学生端互发消息

在此任务中,我们将学习如何使用多块行空板搭建简单的物联网系统,完成项目的核心功能,实现教师端和学生端互相通信。

任务三: 互动答题

在完成了教师端和学生端互发消息之后,我们将结合任务一程序,进一步完善答题系统,实现教师端和学生端的互动答题。

任务一: 倒计时答题

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

学生端的行空板是一个答题器,主要功能是在倒计时时间内,完成答题,接下来我们来一起动手实现 吧。

(1) 显示题目和选项

为了更好地完成题目和选项的显示,先来分析一下学生端界面的组成。整个界面由倒计时、题目、选项内容、选项按钮以及选择答案构成。



现在你可以根据分析,在屏幕上依次添加显示文字和按钮的对象。

注意:为了让比较长的文字完整的显示在行空板屏幕上,可以更新题目文字对象的数字参数宽,实





点击选项按钮,记录并显示学生对应答案,别忘了记录时要使用全局变量。以按钮 A 的操作为例,参考如下实现方法:



(2) 倒计时答题

现在,我们已经可以选择答案了,接下来需要加入倒计时,限制答题时间。

(a) 显示倒计时

倒计时的显示我们可以利用 <mark>变量</mark> 和 <mark>重复执行几次</mark> 来实现, 即 <mark>变量</mark> 每隔一秒减少 1, <mark>重复执行</mark> 开 始设定的时间次, 并实时更新倒计时的显示。

将变量	倒计时 🔹	赋值为 5		
重复执行	〒 5 次	K. K. K	5 d	e v s
	更新对象名	5 🕕 的文本内	 容参数为	变量 倒计时
等待	1 砂	, , , ,	• ×	
将变量	星 倒计时 •	「赋值为 受望	量 倒计时	- 1
	٦			
	更新对象名(t 的文本内容	释参数为 🥊	• O • •

(b) 倒计时答题

在使用答题器的过程中,"选择答案"显示的同时不断计时,我们可以使用"线程"指令来完成。在 "行空板""多线程"分类下,找到 线程对象 thread1 启动和 当线程对象 thread1 启动后执行 两指 令,搭配使用。

注意: "线程"常被用来实现某段时间多个功能同时运行,关于"线程"的介绍,参见"知识园地"。









4. 程序运行

点击运行,倒计时为0之前可以点击按钮选择答案;倒计时结束后,按钮变灰,不可再答题。



5. 试一试

现在请为你的答题器设计简单的提示语,并在倒计时后判别答案的正确与否(正确答案为 "B"),你可以参考下面的界面设计。



任务二: 教师端和学生端互发消息

1. 任务分析

课堂互动答题系统中,需要至少两块行空板实现教师端和学生端对于答题相关信息的传输,我们可以 利用行空板搭建物联网系统完成。

在这个物联网系统中,负责存储、收发答题数据的教师端是服务器,同时利用行空板屏幕在教师端实 现控制发题和显示接收答案,也就是说,此时教师端也具备了移动终端的功能。而学生用来答题产生选择 答案的学生端答题器是智能终端。



分析完两个行空板在物联网系统中的作用之后,需要做好行空板配置网络、开启 SloT 服务等准备工作, 才能开始编程,实现学生端和教师端间数据传输。

2. 准备工作

要实现教师端和学生端的数据传输,首先需要教师端和学生端行空板接入同一网络下,另外,为了可 以使用电脑访问服务器连接两块行空板,电脑也需要接入同一网络下。接下来,我们就来分别配置电脑、 教师端行空板和学生端行空板的网络,然后在教师端行空板上开启 SloT 服务。

(1) 配置网络

首先,先将电脑连入路由器或手机的热点 (需使用 2.4G Wi-Fi);

\leftarrow wlan	
交 zq123 已连接,安全	()
	断开连接

然后,取一块行空板用 USB 线连上电脑,并打开浏览器在地址栏输入"10.1.2.3",进入行空板服务 页面。接下来,点击"网络设置",寻找"连接 WiFi"部分,你可以点击"扫描"寻找刚才电脑连接的无 线网,点选 WiFi 名称并输入密码,点击"连接",等待连接成功。

注意: 有关行空板无线网络知识, 参见"知识园地"。

	W/iEittr本
ā	WIFIAABS
	行空板板载WiFi网卡,可连接WiFi热点.
1 11 天	<u>米市市市东 ac123</u>
的逻辑	三前则注按。20123
上传	
	连接WiFi
	连接WiFi 注意仅支持2.4G WiFi热点.
	连接WiFi 注意仅支持2.4G WiFi热点. 名称: zq123 Y 扫描

完成后,拔下 USB 线,用同样的方式给另一块行空板配置网络。

最后,需要给两块行空板同时上电,使它们都保持开机状态。

注意:给行空板上电的方式有三种:USB或 type-c 口输出的电源适配器、USB或 type-c 口输出的充电宝或电脑 USB 端口。

接下来,我们需要分别进入两块行空板的 Home 菜单,查看它们的网络信息中的 IP 地址,检查是否连接成功。



如果此时两块行空板的 IP 地址,前三段数字相同,证明它们是在同一局域网下。现在我们可以选择其中一块行空板作为服务器,去开启 SloT 服务。在本课项目中,选择了 IP 地址为 192.168.43.222 的行空板开启 SloT 服务,作为服务器;而另一块 IP 地址为 192.168.43.218 的行空板作为智能终端。



(2) 开启 SloT 服务

此时你可以像 11 课一样,在作为服务器的那块行空板的"Home 菜单"中打开 SloT 服务,也可以直接在浏览器中输入它的 IP 地址对 SloT 服务进行设置或查看。

← → C △ ▲ 不安全 19	2.168.43.222, pc/application-switch
○行空板 UNIHIKER	
企 主页	SIoT
盟 应用开关	SIoT是一个针对学校场景的开源免费的MQTT服务器软件,可一键创建本
☆ 网络设置	运行状态: ● 正在运行 SIoT
上 文件上传	停止服务
	打开页面

(3) 远程连接 Mind+

由于接下来我们要给两块行空板编程,所以需要打开两个 Mind+文件,分别起名为"教师端"和"学生端",并采用远程连接的方式,使用行空板 IP 地址分别将行空板和对应的 Mind+文件远程连接起来。远程连接方式如下:

.实际的教师端IP地址	输入实	SSH登录	接远程终端 ▼	"教师端" ⁄lind+文件
	.168.43.222	* 地址: 19	10.1.2.3	
		L	192.168.123.1	
	t	*用户名: ro	手动输入	- 1
			打开网络中心	
æ		*密码:	帮助文档	<u>.</u>
₩ 家际学生詳IP###	取消 确定 論入应	SSH容录		"学生端"
》 文际学生端IP地址	取消 确定 输入实	SSH登录	<u>新元程终端</u> ▼ 10.1.2.3	"学生端" lind+文件
实际学生端IP地址	取消 确定 输入实 168.43.218	SSH登录 *地址: 192	鼓元程终端 • 10.1.2.3 192.168.43.222	"学生端" /ind+文件
。 文际学生端IP地址	取消 确定 输入实 168.43.218	SSH登录 *地址: 192 * 用户名: roc	鼓元捏终端 ◆ 10.1.2.3 192.168.43.222 手动输入	"学生端" Mind+文件
。 文际学生端IP地址	取消 确定 输入实 168.43.218	SSH登录 *地址: 192 *用户名: roc	安元程终端 ▼ 10.1.2.3 192.168.43.222 手动輸入 打开网络中心	"学生端" Mind+文件

3. 编写程序

准备工作已完成,接下来就可以编程实现教师端和学生端互相收发答题消息的任务了。

(1) 教师端

教师端需要实现发送开始答题命令和接收学生答案两个功能,我们可以设置发送按钮和接收文字显示 对象,方便查看。

$\overline{\bigcirc}$	对象名	text	显示文字	(*教	切币端")在X(80 Y	80	字号	20	颜色		提示	文字	
\bigcirc	对象名	tip	显示文字	u u) 在X(80 Y	120	字号	20	颜色	•	一接收	文字	发送按钮	
Θ	对象名	butt	on 增加的	ġil	" 发送 ") 在X	80	200	宽	80	高 30	点击回	调函数	button_click1	

我们可以使用 MQTT-py 模块, 设置、连接、订阅 两个主题:"开始答题/答案",发出正确答案给 学生端,控制开始答题;"学生答案/1",用于接收学生端发出的答案。



接下来,可以参考 11 课收发消息的方法,按下按钮发送正确答案作为开始答题消息,然后接收学生 消息并显示内容。完整教师端程序如下:



(2) 学生端

学生端可以接收教师端的开始做答消息,也可以发送自己的答案给服务器,你可以参考复制"教师端" 的代码到"学生端",适当修改提示文字和收发主题即可完成。

这里跨文件复制大段程序,可以利用 Mind+下方的"书包"完成。

Mind+	V1.7.2 RC3.0 教师端 .sb3			-23		×
Min	@1-0→ 项目 ▼ 教程 ▼	连接远程约	端 ▼ ⑦ ^意 反	ル 演	无制机	
模块	代码	▶湖	(④代码区)	(● 文件)	系统)	
Q	控制	Ŧ	*a			
0	等待 1 秒		- 市場答照/答案*			
控制	等待直到		学生容素/1・	P (8)	V 8	
运算符			20007 (2000) 20007 (11) 11	x 40 v 12	20) 7 8 2	0
1	重复执行 10 次		100) 75.68591 (* 28 3	≝•) ∉X (80	v 🗖	
数字					O)
文本	重复执行直到		butten attaint	1940	- Co	
Ê,	Mind+最	下方的书籍				
*		1	10 89X-4448-850	- C. B.	E	J
_	tom matter	书包				

点击打开"书包",然后将要复制的代码整段拖入"书包"区域,然后保存当前文件。

重复执行 10 次										如果
7										
重复执行直到			鼠标	左键	按	۴,	拖	进书	包	
	书包		1							
		(土回调e	Ažx b	utton_c	lick1) 私支援的	战			
		口发布	•в•) 🗊 💽	开始领	客题/零	客案 "			

现在,打开要粘贴的 Mind+文件,点击"书包",即可找到复制的代码,此时直接拖出要使用代码到 "脚本区"就完成粘贴操作了。



现在,你可以复制"教师端"文件中的代码到"学生端",并适当的修改收发内容和对应主题,完成

Python主程序开始	
	ξ) IP
▲ MQIT发起连接 MQIT牢合: SIOT	D
SloT服务書: 192.168.43.22	
MQTT MQTT保持连接,水久 SloT 用户名: Siot	51.0
MQTT订阅 "开始答题/答案" 学生端提示信息 SloT 密码: dfrobot	
MQTT订阅 *学生答案/1 *	
○ 7 象名 text 显示文字 ● 学生端● 在X 80 Y 80 字号 20 颜色 ● 20 000 00000000000000000000000000000	
○ 対象名 tip 显示文字 ● 在X 80 Y 110 字号 20 颜色 ●	
A 象名 button 増加按钮 「确认答案」 在X 80 Y 200 宽 80 高 30 点击回调函数 button_clic	k1
学生端接收开始答题消息。如果是学生端接收开始答题消息。	
つつつつつつつつつつつつつつつつつつつつつつつつつつつつつつつつつつつ	
Month Single Si	
如果 MQTT主题 = "开始答题/答案" 那么执行 MQTT发布 "B" 到 "学生答案/1"	
○ 更新对象名 tip 的文本内容参数为 "开始答题" ○ 更新对象名 tip 的文本内容参数为 "已发	ا ۳

学生端代码,参考代码如下:

4. 程序运行

检查学生端和教师端的远程连接情况,先运行教师端,后运行学生端,模拟整个答题过程,观察运行 情况。

注意:如果遇到"网络受限"导致教师端或学生端无法接收,可尝试重新运行程序或者更换无线网络。



任务三: 互动答题

1. 编写程序

现在,我们已经可以实现教师端和学生端互发消息,但是这还不是一个完整的答题系统,接下来我们 将任务一和任务二结合起来,完善教师端和学生端界面和功能,完成互动答题任务。

(1) 设计教师端和学生端界面

在互动课堂答题系统中,教师端界面包含题目以及控制"开始答题"按钮,我们可以直接利用"书包" 复制任务一的题目和选项代码,适当修改并添加学生答案文字对象即可完成。



学生端需要在接收教师端控制消息以后,开始显示答题界面,也就是说学生端开始只显示等待提示, 当接收到教师端消息以后再显示完整的答题器界面。当然,你也可以直接复制任务一的代码并修改。



(2) 学生端倒计时答题

任务一中,我们使用线程实现了程序开始运行就倒计时的效果,但是在答题系统中,应该是学生端收 到教师的答案消息以后才开始倒计时,此时我们可以巧妙的以教师端发来的"正确答案"为标志开启倒计 时。也就是说 正确答案变量 开始赋值为"",当接收到答案时,正确答案变量 被赋值为 传送过来的消 息,此时,正确答案变量 不为"",开始倒计时。当然,为了持续检测 正确答案 的值,别忘了 循环执 行。

开始赋值	收到教师端答案,更新 正确答案	
Python主程序开始	ANT MATTA MATTER 接收到 MATTER	S 当线程对象 thread1 启动后执行
将变量 正确答案 • 赋值为 ***	全局global 变量 正确答案	全局global 变量 倒计时 实时检测是否收到消息
👼 аланимотт 🔕	如果 MQIT主题 开始答题/答案 琳么执行	全局global 变量 正确答案
MQTT发起主接	将变量 正确答案 · 赋值为 MQTT信息	箱林块门 如果 非 查是正确答案 — 1 1 4 地方
	● 更新对象名 text 的数字参数 × → 为 70	将变量倒计时 • 赋值为 5

最后,只需要在学生端设置结果判断并发送学生选择答案,帮助老师获得学生答题情况即可。完整的 教师端和学生端参考代码如下:







2. 程序运行

检查 Mind+远程连接,分别运行教师端和学生端程序。教师端部分,点击开始答题发送正确答案,等 待学生答完题目,收到学生回答的答案;学生端部分,当接收到教师开始答题消息,学生读题并作答。



学生端

<u>知识园地</u>

1. 程序中的线程

线程,它是一个程序在运行过程中正在进行的任务,需要说明的是这里的任务是指一系列有先后执行顺序的代码,你可以把它类比成"线"。当一个程序只有一个任务时,通常我们说这个程序是单线程的, 比如上节课学习的绘制可视化图表;但是当一个程序中需要多个任务同时执行时,这个程序就需要使用多 个线程,来同时完成多个子任务,我们称从开始到最后一直执行的线程为主线程,其它子任务为子线程。



你可以类比在家做饭帮助理解单线程和多线程,当只有一个灶头时,你需要先完成蒸鸡蛋,然后再炒菜,两个菜只能按顺序制作,效率比较低,但当有两个灶头时,你可以在蒸鸡蛋的同时完成炒菜任务,这 样效率就提起来了。

行空板与线程操作相关的指令有三个,位于行空板"多线程"分类下,你可以用 <mark>线程对象 thread1</mark> 启动 指令启动线程,<mark>线程对象 thread1 停止</mark> 结束指定线程,<mark>当线程对象 thread1 启动后执行</mark>实现子线 程任务。



另外,由于线程启动和运行会分掉主线程时间,线程使用越多,程序执行就越慢,甚至导致程序卡死, 所以建议谨慎使用。

2. 行空板无线网络模块

行空板自带 Wi-Fi&蓝牙模块 RTL8723DS 2.4G&4.0,对应天线位于行空板上方左侧。



当有无线网络 (或手机网络热点) 时,你可以通过访问"行空板服务页面设置"对 WiFi 热点服务进行 连接设置,利用它完成访问网络或物联网搭建任务。

另外,在没有无线网络的情况下,行空板也可以开启网络热点服务,虽然不能上网,但是其它行空板 或可无线连接的设备可以通过连接此热点,完成物联网的搭建。行空板无线热点模式的开启,需要进入"行 空板 Home 菜单",按下行空板 B键,找到并进入"开关无线热点模式",检查并开启"无线热点",启 用后就会显示无线网络账号(SSID)和密码(PASS)。



3. 指令学习

本项目主要使用了学习了线程以及按钮启用/禁止指令,接下来我们就详细学习一下。





挑战自我

请参照本节课内容,加入多个学生端进行答题,并为教师端设计多个学生选择答案的显示界面以及答 案正确与否的判断。



第十四课 语音翻译机

你遇到过和外国人沟通交流时不知道如何表达的难题吗?今天我们就利用行空板制作一个语音翻译 机,自动帮你翻译你想说的话,让你从此和外国人沟通无忧。



任务目标

按下行空板 A 键,对着麦克风录入语音,显示录入文字;按下行空板 B 键,显示英文译文。



<u>知识点</u>

9. 了解语音识别基本操作方法

10. 了解机器翻译基本操作方法

11. 学习行空板录音操作指令

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



<u>动手实践</u>

语音翻译机的主要功能是让行空板识别并翻译我们所说的话。具体如何实现呢? 接下来我们就分两个 任务来完成吧。

任务一: 语音转文字

在此任务中,我们将学习行空板录音功能以及语音识别的操作方法,完成行空板对于语音的识别和识 别文本的显示。

任务二:显示翻译结果

基于话语的识别结果,我们将进一步学习机器翻译操作方法,将中文翻译成英文并显示。

任务一: 语音转文字

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。


2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

你一定发过微信语音,我们可以对着麦克风发送语音消息,也可以将语音转化为文字。如果将整个过 程分为两个步骤,就是录音和识别。行空板也类似,具体如何实现呢,接下来我们看一看吧。

(3) 录制声音

行空板自带录音功能,你可以在"行空板"里的"音频录放"分类下找到 录音 10 秒直到结束,文件 名"record.wav" 完成声音的录制。



使用时,只要设置好录音时长和确定录音文件名就可以了。



注意:录音文件 "record.wav" 被存放在行空板缓存文件夹中,你可以打开 "文件系统" 的 "行空板 中的文件"→"mindplus"→"cache",点击项目名称文件夹寻找录音文件。

● 停止 ●代码区 ●	• 文件系统
文件目录	C
项目中的文件	⊕≡
目 箭头png 电脑中的文件	€ ≡
○ 行空板中的文件	⊕ ≡
 0-演示程序 00_安装依赖 01_人工智能预则出行 	
 mindplus cache 第十四课语音翻译机任务— record.wav 	

(4) 识别录音

录制好语音以后,就可以开始准备识别录音了。识别录音本项目将使用线上人工智能开放平台——"讯 飞开放平台"的语音识别技术。只要行空板连接好无线网络,然后将录制语音输入申请好的平台服务引擎 内,就可以很快获得语音对应的文本内容了。 注意:关于语音识别技术的介绍,参见"知识园地"。



(a) 行空板连接无线网络

双击打开浏览器访问 http://10.1.2.3/, 进入行空板服务页面连接无线网络 (若没有无线网支持手机热

点)。

当前连接: 未连接 P地址: 未连接
WiFi
主意仅支持2.4G WiFi热点.
3称: 日田 · · 日描
密码: <u>连接</u> ◎ 连接
ViFi 连接完成 ViFi 连接成功

然后, 注册并登录"讯飞开放平台"账号;

→ C ☆ â xfyun.cn ② 中交給 v 新手指面 文(音中心 SDK下菱							875	<i>년</i> 8中心 乐宴会云	介 [注]	* *		
	务 解决方案	行业专题	服务市场	AI大学型	1024	AI大赛	生态合作	服务与支持	最新活动				
登陆成功后,点	武击右上角 控制台	的"控制	创台",	进入应	用界面	ā,点ə Frònm	告"创建 ឃ་་	副》 ^{1日又}	,开始填	真写应	用信	恴	;
持 最新活动						©]	<u></u> 我的应	如用	创建非	版用			

③ 我的应用 > 创建应用



填写完成后,点击提交,就可以在"我的应用"中看到刚才创建的应用;

应用名称	Ŧ	APPID	分类	创建时间	状态	编辑
语	语音翻译机-行空板	988a1c33	应用·教育学习·学习	2022-11-01 13:11:16	o 上次操作的应用	Q

点击进入应用查看应用的服务、账号、密钥等信息。



(c) 获得识别结果

获得账号、密钥等信息后,我们就可以使用指令完成语音的识别任务了。"讯飞语音识别"操作指令 需要在"扩展","用户库"中输入网址(https://gitee.com/liliang9693/ext-xunfeiyuyin)检索获得。 注意:首次使用用户库时,必须手动输入网址添加,若直接打开已添加好库的程序,不用网址搜索添加,运行时会报错。

and bud and the second		
	https://gitee.com/liliang9693/ext-xunfeiyuyin	
	注意: 用户库中的模块由Mind+爱好者制作, 点击这里百看开发教	程以及用户库列表
	4 30 胡玉开放平台	
	讯飞语音	
	Python模式運用訊飞语言服务 油石市系的000	





使用的过程中,首先拖出 <mark>讯飞语音 设置账户 APPID:" " APISecret:" " APIKey:" "</mark>, 填入刚才 讯飞应用中的对应信息,放在程序最开始部分。

(1) 讯飞语音	设置账户APPID: (****) APISecret: (****) APIKey: (*****
	5705
APPID	min /
APISecret	conversion of the first first section in the last
APIKey	State of the second
*SDK调用方式只	需APPID。APIKey或APISecret适用于WebAPI调用方式。

然后将录音文件写入 讯飞语音 识别录音文件 ""<mark>指令,就可以获得对应录音的文本内容了,当然为</mark> 了方便使用,你可以把识别的结果存放在变量里。



最后,项目中是通过按下行空板 A 键,开始录制显示话语文本,但是若在有限时间内未录入声音时, <mark>变量识别内容</mark> 为空,我们可以添加一个文本提示"数据错误"。



现在,你可以添加文字对象显示话语的文本。完整参考代码如下:

Python主程序开始										
👝 讯飞语音 设茜和户APPID: 🚺 APISecret: 😷 🚛 👘	10170	10.00		APIKey:	(the	151P	alia n	el 5	1.53	
🕞 对象者 Chinese 显示文字 🎌 在X 30 Y 60 字号 20 颜色 🔵		8	8		0.0	× 1	e . e			
○ 更新对象名 Chinese 的数字参数 意•为 180 ← 设置文字宽度,	自动换行									
循环执行										
如果 🕞 技细 A • 被按下? 那么执行										
□ 录音 3 秒直到结束,文件名 record.wav*										
将变星 识别内容 • 赋值为 (iii) 武飞语音 识别录音文件 • record.wav •										0
如果 (交量 识别内容 = () 那么执行										0
将变量 识别内容 ▼ 賦值为 "数据错误"										~
and the second										4
○ 更新对象名 Chinese 的文本内容参数为 变量 录别内容										\odot
										Q

4. 程序运行

点击运行,按下A键,对着行空板麦克风说话,等待一会显示说话内容。

注意:项目运行时可能会出现报错,可以尝试重新运行或断开重新连接行空板。当出现如下错误,请 对应操作修改。

。 1993年1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日	修改方式					
error: Handshake status 401 Unauthorized	当前行空板没有正确访问开放平台,此时 需要更换无线网络并等待一会,再重试。					
error: [Errno -3] Temporary failure in name	无线网络连接有问题,需要检查或更换无					
resolution	线网络。					



任务二:显示翻译结果

1. 编写程序

基于已经获得文本内容,本项目中的翻译使用的是"百度翻译开放平台"功能,获得翻译内容的方法 类似,依然需要先获取"百度翻译开放平台"账户密钥,然后在使用对应指令实现翻译功能。 (1)获取"百度翻译开放平台"账户密钥

- 首先,使用浏览器访问"百度翻译开放平台"网站(https://fanyi-api.baidu.com/);
- 然后,注册并登录"百度翻译开放平台"账号;



登录成功后,点击"产品服务",进入并开通免费的翻译资源(申请过程可能需要实名认证)。

Bai (3) 百度 翻译开放平台	首页 产品服务 文档与支持 AI同传 管理控制台	开通通用翻译API
2000		● 选择开通的服务版本 今人开发者、企业开发者
通用翻译 垂直领域 文档翻译 语种识别	免费体验通用翻译API	·⑧ 标准版
智慧多模 國片翻译 语音翻译	完成身份认证,最高每月享200万免费字符量 完成身份认证,最高每月享200万免费字符量 点击开始建立 立即使用	 基础文本範译服务 每月5万字符免费调用量 超出部分技 49元百万字符计费 QPS (每秒访问量)=1 翻译数据量份至百度网盘
		● 稳定服务保证 选择开通标准版 开通标准版

之后会进入应用信息填写部分,填写完毕后,只需要点击"提交申请",即可完成免费翻译应用的申 请和创建。

● 小人开支書、企业开業:		GIMAL RA证明关资料,等符案按照过	
您的应用名称是什么? (名称不振	过40个字符)		
百度翻译测试		0	
有无应用相关的介绍链接(不超过40	0个字符)		
間編入相关介紹問題			
清简单介绍下您的应用(不相过200	7)		
总带壁实现什么样的功能? 机应用	新用在什么编辑?		
服务器地址	:(使用服务器调用API的用户,) 中间请用"隔开,如"113.173.40	青墳写服务器IP地址, APP等) 2551124.12.122.1*: 墙写IP	客户端不需要填写; 后,只有填写的P才
服务器地址 多个服务器 可被调用,	:(使用服务器调用API的用户,) 中间请用1隔开,如113.173.4(请谨慎填号,不超过100个字符,	青填写服务器IP地址,APP等).255(124.12.122.1*; 靖写IP)	客户端不需要填写; 后,只有填写的IP才
服务器地址 参个服务器 可被调用, 店输入题	:(使用服务器调用API的用户,) 中间请用竹隔开,如"113.173.40 清谨值填写,不超过100个字符, 务器地址	青填写服务器IP地址,APP等 0.255[124.12.122.1*; 填写IP)	客户端不需要填写; 后,只有填写的IP才
服务器地址 多个服务器 可被调用, 请输入版	:(使用服务器调用API的用户,) 中间请用1隔开,如113.173.4(清谨慎填写,不超过100个字符, 务器地址	青填写服务器IP地址,APP等 0.255[124.12.122.1*; 填写IP])	客户端不需要填写; 后,只有填写的IP才
服务器地址 参个服务器 可被调用, 情输入题	:(使用服务器调用API的用户,) 中间请用"T器开,如"113.173.40 清谨值填写,不超过100个字符; 务器地址	倉填写服务器IP地址,APP等 0.255[124.12.122.1*; 填写IP])	客户端不需要填写; 后,只有填写的IP才

创建成功后会显示目前翻译资源使用的情况。

昨日使用	字符数	本月使用字符数	总使用字符数
0		0	0
2022-1 昨日消费金	2-07 馥:0元	2022-12 本月消费金额: 0元	息消费金额: 0元
郓时间 2022-114 11条 字符数总计	8 门 - 2022-12-8 0十百费金额总计0元)		
择时间 2022-114 [31条 字符数总计 序号	5 〇 - 2022-12-8 0 防費金額会计0元) 时间	 二 読述 清空 字符数 	消费金额 (元)
择时间: 2022-114 431条 宇符数总计 序号 1	8 〇 - 2022-12-8 0 時費金額2计0元) 時间 2022-12-08	 一 端近 清空 デ符数 0 	消费金额 (元) 次日枯算
採町町 2022-114 は1年 宇符数には 序号 1 2	8 日 - 2022-12-8 0 時費金額の計の元) 利用 2022-12-08 2022-12-07	 一 論述 清空 デ符数 0 0 	消费金额 (元) 次日結算 0
择时间 2022-114 31条 字符数是计 序号 1 2 3	8 [] - 2022-12-8 0 時費金額の計の元) 8対何 2022-12-08 2022-12-07 2022-12-06	 一 端选 清空 - 字符数 - 0 	消費金額(元) 次日結算 0 0
择时间 2022-114 431条 字符数是计 序号 1 2 3 4	3 日 - 2022-12-8 0 時費金額の計の元) 月1日 2022-12-06 2022-12-06 2022-12-06 2022-12-05	 一 第誌 清空 デ符数 〇 0 〇 0 〇 0 〇 0 〇 0 〇 0 	消費金額(元) 次日結算 0 0 0

账号和密钥信息,需要点击"开发者信息"里的"申请信息"查看。

a	开发者中心	
	申请信息	
88 总览	APP ID:	
Q 开发者信息		

(2) 获取翻译结果

添加"百度翻译模块"和"讯飞语音"的方法相似,在"扩展","用户库"中输入网址 (https://gitee.com/liliang9693/ext-baidufanyi)检索获得。

注意: 首次使用用户库时,必须手动输入网址添加,若直接打开已添加好库的程序,直接运行不点击添加,会报错。

官方库 pinpong库 用户	庫
	https://gitee.com/liliang9693/ext-baidufanyi
	注意:用户库中的模块由Mind+爱好者制作, <u>点击这里</u> 查看开发教程以及用户库
已加载:	
讯飞语音 Python模式使用讯飞语音服务 Pyth	百度翻译 on模式使用百度翻译服务 进行文字翻译

相关指令,在指令区"用户库"分类下,找到百度翻译相关指令。

Python ● 指iliang:百度關译 役置百度翻译账号 ID 役置百度翻译账号 ID ● 中 日 日 日 中 日 日 日 日 日 一 一 一 一 一 一 一 一 一 一 一 一 一	r: •••• 密報 法》 英语 •		
使用的过程中,还是先设置百度翻译账号 ID 密钥,容翻译。	然后使用	将"hello'	['] 翻译为英语 <mark>,直接完成内</mark>
开发者中心 申请信息 ・ * * · · · · · · · · · · · · · · · · ·		翻译内容 将 [*] hello	翻译语言 * 翻译为英语
APP ID: 对应填写 密钥: 对应填写			✓ 英语 中文 繁体中文
			 粤语 文言文
			新语 案语 葡萄牙语
			希腊语

现在,回到我们的任务——翻译识别出的内容,也就是在完成账号设置以后,将 变量 识别内容, 放进翻译指令的 "hello" 部分即可。现在你可以建立新的文字对象,实现加入按键 B,显示翻译后的文本。 完整参考代码如下:

Python主程序开始		
(h) 讯飞语音 设置账户APPID: APISecret:	APIKey:	
・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・		
○ 対象名 Chinese 显示文字 ● ● 在X 30 Y 60 字号 20 颜色 ●		
○ 更新对象名 Chinese 的数字参数 宽 ▼ 为 180		
○ 対象名 English 显示文字 ● ● 在X 30 Y 150 字号 20 颜色		
◯ 更新对象名 English 的数字参数 宽 ▪ 为 200		
循环执行		
如果 🚫 按钮 A • 被按下? 那么执行		
将变量 识别内容 • 赋值为 👍 讯飞语音 识别录音文件 "record.wav"		
如果 变量 识别内容 = ••• 那么执行		
将变量 识别内容 • 赋值为 '数据错误'		
○ 更新对象名 Chinese 的文本内容参数为 变量 识别内容		
如果 🚫 按钮 B • 被按下? 那么执行		
🕞 更新对象名 English 的文本内容参数为 🔛 将 变量 识别内容 翻译为 英语•		
	-	

2. 程序运行

检查行空板连接情况,点击运行,按下A键录入语音,等待一会,屏幕显示录入语言文本;按下B键, 屏幕下方出现英文翻译。



3. 试一试

你的现在已经可以讲中文语音翻译成英文了,机器翻译指令可以翻译成韩语、粤语、文言文等其它种 类的语言,自己动手切换试一试吧。

<u>知识园地</u>

1. 语音识别技术

语音识别技术,也被称为自动语音识别 (Automatic Speech Recognition,简称 ASR)是让机器通过 识别把语音信号转变为相应的文本或命令的人工智能技术,也就是让机器听懂人类的语音的技术。它的基 本工作流程是,将录入的声音进行数字化和预处理之后,通过特征提取获得能够表征语言特点的特征向量, 然后通过加入模型库进行模式匹配,获得概率最高的文本,输出最终结果。



语音识别技术作为人工智能技术非常重要的组成部分之一,市面上有不少的人工智能开放平台如百度、 讯飞、腾讯等,都提供了相关的技术引擎,供大家使用。

2. 机器翻译技术

机器翻译(machine translation,简称 MT)是将一种自然语言翻译成另一种自然语言的技术。它的 基本工作流程是,对原文句子进行分词、删除、整理等规范化预处理得到一系列短语序列,然后将他们输 入翻译模型转化成译文序列,最后在进行拼接、特殊符号处理等处理,得到符合人类阅读习惯的文本译文。



在机器翻译技术的工作流程中,最核心的部分翻译模型又有很多建立方法,现如今最为常用的方法是 基于神经网络的机器翻译方法,如果你感兴趣的话,不妨自己查询研究一下吧。

3. 指令学习

本项目主要使用了学习了录制音频、讯飞语音识别、百度机器翻译相关指令,接下来我们就详细学习 一下。





挑战自我

在本项目中你已经完成语音翻译机的主要功能,现在请你进一步完成下面任务,完善项目功能。 (1)为你的语音翻译机设计界面和提示语,你可以参考下面的图片进行设计(图标图片在素材文件夹中);

	中文	>英:	文
按门	FA键开	始录入	
译文	、(按下	B鍵开	始翻译
ī.			

(2) 设置一个可以点击切换转换语言的按钮或图片(如"中文"后的图标),让你的翻译机可以翻译更多类型的语言吧。



第十五课 天气助手

准备外出旅游的时候,查询天气,选择合适的衣物是非常重要的准备工作,今天我们就利用行空板来做一个"天气助手",来帮你查询计划出行的城市天气吧。



任务目标

行空板屏幕显示当前城市、天气、日期及温度。



<u>知识点</u>

11. 了解程序中的接口知识

12. 掌握程序中字典的基本使用方法

13. 掌握获取天气信息的方法

材料清单

硬件清单:



软件使用: Mind+编程软件 x1 下载地址: https://www.mindplus.cc



动手实践

天气助手的主要功能是通过网络获取和显示城市的天气和气温。具体如何实现呢? 接下来我们就分两 个任务来完成吧。

任务一:获得天气数据

在此任务中,我们将学习利用"天气 API"开放平台获得并显示天气数据。

任务二:显示指定天气信息

基于已获得天气数据,我们将进一步学习字典知识,完成数据筛选,显示所需天气信息。

任务一:获得天气数据

1. 硬件搭建

使用 USB 连接线将行空板连接到计算机。



2. 软件准备

打开 Mind+, 按照下面图示完成软件准备过程。



3. 编写程序

手机中显示的天气数据都是通过网络获取的,行空板也类似。接下来我们就来学习获取天气数据的方法。 天气数据需要通过"接口"获得,本节课我们使用"天气 API"获取。

注意:关于"接口"知识,参见"知识园地"。

首先,访问天气 API 网站 (https://tianqiapi.com/),完成注册和登录;

🐯 天气АРІ	网站	皆页 API文档 网页播件 更	新日志 关于我们 黑田戸中心
	更友好的全球化天	气预报API接口	andes
	curl "https://v0.yiketiangi.com/api?unescape=18version=v918vappid=- 登录控制台	43656176&appsecret=I42og6Lm&ext=&cityid=8	xity=" 欢迎使用
账号登录 		账号注册 登录邮箱 登录密码	
登录 您还没有账号 点击注册 找回密码		重复登录密码 注册 已有账号? 立即登录	
然后,点击"返回天气 A つ 易客云API ダ 返	PI 官网"→选择"API 文档"查询的	免费天气接口;	

功能荣单		网站首页 API文档 网页插件 更新日志 关于我们
□ 控制台	● (宋米縣定员上微信吧, 可以收到	
	账户信息	更友好的全球化天气预报API接口
V MULTIMATING	账户余额:¥0	
个人中心	测试豆: 1998个	

免费天气信息在 API 文档左边栏的"免费基础实况天气"中,另外,在右边的"请求示例"下方就是要使用的接口网址。

MUSCA WAPT	tæsses∠. nutpe	認識者2. Https://yiketiandi.com/Tree/day									
 空气污染指数 	请求示例	请求示例 https://www.yiketianqi.com/free/day?appid=&appsecret=&unescape=1									
▶ 31项生活指数接口	https://www.										
▶ 全国空气质量城市排行	请求参数说明]:									
▶ 全球空气质量国家排行	参数名	必选	类型	说明	备注 (示例)						
▶ 乡镇实况天气											
▶ 乡镇7日天气	appid	是	string	用尸appid	汪册升友账号						
▶ 乡镇实况+7日天气	appsecret	是	string	用户							
▶ 四季天气				appsecret							
▶ 全国所有城市实况api	cityid	否	string	城市ID	请参考 城市ID列表						
▶景区天气api					如果您想查询多个城市(最多30个),律 cityid=101120101,101120201,10105						
▶ 免费基础实况天气	city	否	string	城市名称	不要带市和区,支持市区县,不支持约						
▶ 男费基础工口入气	1000	12:21	100000	- 446-45-452.452							

你也可以复制访问该网址,查看当前所在城市的天气数据,你能看懂数据中数值的含义吗。



{"nums":0,"cityid":"101020100","city":"上海" [date":"2022-11-04],"week":"星期 五","update_time":"09:23",[wea":"多云" "wea_img":"yun",[tem":"16],"tem_day":"17","tem_night" 风","win_speed":"1级","win_meter":"2km\/h","air":"39","pressure":"1026","humidity":"62%"}

另外,如果你想更换城市的天气数据,只需要在接口网址末尾加上"&city=某城市",比如,杭州天 气的接口网址为:

"https://www.yiketianqi.com/free/day?appid=xxxx&appsecret=xxxx&unescape=1&city=杭州"

天气数据已经获得,如何放进程序呢?这需要通过"Requests 模块"对天气 API 发出请求才能获取数据。

点击"扩展"→"官方库"中,找到"Requests"并添加。



接下来,就可以在"Requests"分类下找到对应指令,本节课我们要使用的是发起一个 GET 请求"" 指令向天气 API 发出请求,以及解析请求内容为文本格式将获取数据转化成刚才网页中看到的样子。



注意: 解析 请求内容为 文本格式 指令解析数据时有三种格式可选,文本格式是字符串形式;二进制格式是指二进制类型串(bytes 类型),你可以把它理解成由二进制数组成的字符串;JSON 格式是打包好的数据格式,可以用关键字找到对应的数据。为了方便待会处理数据,这里选择JSON 格式。

为了更好理解代码, 实现时可以建立 <mark>url、response 以及 data</mark> 三个变量来分别储存接口网址、请求 响应、天气数据。



接下来,你可以直接将 data 变量 显示在行空板上,为了更完整的看到天气数据,可以设定显示的字体大小和文字宽度。完整参考代码如下:

Python主程序开始	
将变量 url ▼ 赋值为 🗐 合并 (* https://www.yiketianqi.com/free/day?appid=5 📲 ■ 配&appsecret=3 📲 📲	&unescape=1&city= " (" 上海 ")
将变量 response ▼ 赋值为 余 发起一个GET请求 变量 url	
将变量 data ▼ 赋值为 😭 解析 变量 response 请求内容为 JSON格式 ▼	
○ 対象名 text 显示文字 变量 data 在X 20 Y 0 字号 12 颜色 ●	
● 更新对象名 text 的数字参数 宽 ◆ 为 200 设置文字宽度,多文字自动换行	
循环执行 · · · · · · · · · · · · · · · · · · ·	

4. 程序运行

运行代码前,你需要使用上节课的方法,让行空板连上无线网络; 点击运行,等待数据获取成功后,屏幕居中显示获得的完整天气数据。



任务二:显示指定天气信息

1. 编写程序

在任务一中,我们已经成功获得天气数据,但显示的内容过于复杂,其实只需要"城市"、"日期"、 "天气"以及"气温"即可。也就是说,我们需要从 data 变量 中筛选出我们所需的信息,如何完成呢? 其实,现在你看到的数据形式是 Python 中的"字典",它是通过":"将前后两个内容绑定的数据类型,可以通过":"前的"键"获得冒号后的"值"。

注意:关于程序中的"字典"知识,参见"知识园地"。



"字典"相关的操作指令,在"指令区"的"字典"分类下,要获取"值",需要使用字典{)键"" 的值 指令并填好"字典"和"键"的内容,比如要获得上图中的日期,可以使用字典 data 变量键"date" 的值。



然后,你就可以将获取的日期信息显示在行空板屏幕上了。

○ 対象名 tt 显示文字 [10] 字典 变量 data 鍵 * date * 的值 在X 70 Y 110 字号 15 颜色 ●

同样的,所在城市、天气和温度信息可以用 "city"、"wea"和 "tem" 三个键获得。当然,显示的 时候可以加入一些文字提示。完整参考代码如下:

Python主程序开始
將变量 url ▼ 赋值为 🗐 合并 (* https://www.yiketianqi.com/free/day?appid=51797918&appsecret=7jGbjgwc&unescape=1&city= *) (* 上海 *)
將变量 response ▼ 「「類值为 」 定
将交量 data ▼ 疑値为 🕋 解析 交量 response 请求内容为 JSON格式 ▼
○ 対象名 tc 显示文: 20 字典 交量 data 鍵 'city' 的值 在X 100 Y 70 字号 20 颜色 ●
🕞 X 揉名 (t) 显示文字 🚺 字典 支量 data 键 (*date*)的值 在X (70 Y (110 字号 (15)颜色 🌍)
○ 対象名 tw 显示文 = 日 合并 •天气 • 10 字典 变量 data 键 •wea • 的值 在X 80 Y 150 字 = 15 颜色 ●
○ X缘名 tem 显示 字 辑 合并 *温度 * 辑 合并 通 字典 支量 data 键 *tem*的值 **C* 在X 75 Y 190 字号 15 颜色
箱环执行

2. 程序运行

检查行空板连接情况,点击运行,等待一会,显示代码填写城市天气、日期等信息。



知识园地

1. 程序中的接口

应用程序接口,又叫接口(简称 API)它是用来提供一些功能或数据的程序。你可以把它理解成一个 别人定义的功能函数,使用时你不需要访问源码,也不需要知道实现原理,只要给出参数并调用函数,就 会得出你想要的数据。就像本节课项目中的"天气 API",网址对应着参数,输入网址并通过访问网站调 用对应的功能函数,就可以获取天气数据,不需要知道这些数据是通过什么手段得到的。



接口的出现是为了让写代码的人可以更方便的实现更多的应用,也正因如此,接口的输入参数和编写 需要更加规范。其实,你可以在很多场景中使用接口,在程序内使用系统定义的函数和指令,直接下载安 装别的公司或开发者写好的库,甚至你可以直接通过网络平台,远程调用更强大的功能,就像本节课通过 "天气 API"获得天气数据,上节课的"讯飞开放平台"获得语音识别功能。

2. 程序中的字典

字典是 Python 程序中基础数据类型之一,它常被用来表示数据之间——对应的关系,比如:姓名和学号、游戏中的技能和伤害点。字典是由"{}"构成,当大括号里什么都没有时,表示空字典;有数据时,数据以"键值对"存在,每个"键值对"形式为"键:值",多个"键值对"用逗号隔开。

```
注意:所有符号都是英文符号。
```



在图形化指令中,在指令区"字典"分类下找到 操作字典的指令。对于上图中的例子,我们可以使用初始化字典{:}和 number变量 来建立字典,同时使用 字典 number 键 "小明"的值 读取上图 number 变量中 "小明"的学号。

将变量 number	动权	财始化字典({ "小明":2	20220101, "/]	02, "/]	硴":2	0220	103,	¤и.		
将变量 小明学号 ▼ 赋值为	50 7	字典 变量	number	键 🕛 小明 "	的值	6	n	x	e.	e.	-

3. 指令学习

本项目主要使用了学习了网络请求、字典相关指令, 接下来我们就详细学习一下。



挑战自我

在本项目中你已经完成了不同城市的天气消息获取,现在请你设计界面布局,完成下面任务:

(1) 添加切换城市按钮,实现开始显示当前所在城市天气,当切换不同的城市后,按下行空板 A 键 更新对应城市天气;

(2)使用素材文件夹中的图片素材,给不同的天气配上不同的图片,当然你也可以显示不同的信息在你的行空板上,界面设计可参考下图。



提示:城市名称以及天气图片的显示,都可以使用字典变量分别将数字和城市名对应、天气和图片名 对应。你可以参考下图完成两变量的初始设置:

>	*		× 4	4 X 4	数字	和城市名对应	Ž				
	将变量	city ▼ 赋值为	[1] 初始)化字典 <mark>{ 1</mark> :	"上海",2:"杭州"	,3:"北京" }	• •	•	5 ×	-	
	将变量	天气图片 • 赋	值为 🔝	初始化字典({ "晴": "晴.pn	g","霾":"霾.png'	',"阴":"阴	天.png	',"多云":'	多云,p	ong" }
						天气和图	十名对应	Z ·			



扫码关注 蘑菇云创造 了解更多信息



