





"函数"在编程里被广泛使用,本关将学习"函数"的概念,并学会在游戏编程中使用它们。 为此我组建了一个由多个独立音符组成的函数乐队,你将学会指挥它们演奏自己喜欢的音乐。

199

🙆 知识点

函数

在 Arcade 里,函数是将要一起运行的代 码段进行分组的一种方式。在编写程序时经常 使用它们,以提高程序的可读性,并允许重复 使用常见操作。简单地说,就是如果你发现程 序里有一些代码段被多次重复使用,就可以考 虑用"函数"了。

Arcade的函数

Arcade 的函数,隐藏在"高级"标签下,单击"高级"标签,可以在展开的标签中,找到"函数"子标签,如右图所示。

下面将通过修改一个已有示例程序来看看,如 何创建函数以及调用函数。



▶ 任务1: 用函数重建Doe-Ray-Mi-1示例程序

本任务从示例程序 arcade-L1-Doe-Ray-Mi-1.uf2(程序见下页)开始。我放置 了3个角色在场景里,按下A键的时候,这3 个角色分别会"唱出"Doe、Ray、Mi三个音, 并伴随相应的位移动作。对于角色演奏的部分, 期望为每个音符做一个函数,通过调用对应音 符的函数来实现演奏功能。



200 │ 第06章 函数、扩展与动画

Doe 👻 设为 精灵 🎊 类型 Player 👻

Ray 👻 设为 精灵 သ 类型 Player 🗸

Mi - 设为 精灵 🛜 类型 Player -

Doe 🗸 说 "Doe" 持续 500 👻 毫秒 🔵

Ray 🗸 说 ("Ray") 持续 (500 🔹 毫秒 🔾

playDoe

设置 Doe ▼ 的位置为 x 20 y 60

设置 (Ray -)的位置为 x (40) y (60)

设置 (Mi -) 的位置为 x 60 y 60

改变 Doe ▼ 的 y ▼ 以 -10

播放 音符 中 C 延时 1/2 - 节拍

改变 Doe ▼ 的 y ▼ 以 10

改变 Ray ▼ 的 y ▼ 以 -10

当按键 Α▼ 按下▼

暂停 500 - 毫秒

动手实践

1/4 为创建函数拆分程序

进入 Arcade 编辑器,加载示例程序 arcade-L17-Doe-Ray-Mi-1.uf2,在屏 幕下方修改项目名为 Doe-Ray-Mi-2。



程序如石图所示,将"当按键 A 按下"程 序块里的程序块,按照不同角色和动作进行拆 分(如右图所示的3个区块),拆分后形态如下 图所示。期望修改的方案如下:

- 对于角色演奏的部分,为每个音符做 一个函数。
- 通过调用音符函数实现演奏功能。



201





在 Arcade 中, 通常函数名字的首字母小写, 下一个单词的首字母大写,并且单词中间没有空格。

可以看到编程区内多了一个"函数 playDoe"程序块,将上页分离出的演奏Doe 的程序块,拖曳到"函数 playDoe"的程序块内。 现在有了第一个完整的函数:playDoe,如右 图所示。

重复这个过程,分别建立playRay和 playMi函数。







有了函数程序块,需要通过调用才能让它 们发挥作用。如右图所示,单击"函数"标签, 可以看到刚创建的3个函数的调用程序块。

我们期望按下 A 按键时,演奏 Doe Ray Mi。现在只需把"调用 playDoe""调用 playRay"和"调用 playMi"三个程序块,嵌 入"当按键 A 按下"的程序块内,如下图所示。







如左图所示,现在通过函数调用实现了和 示例 **arcade-L17-Doe-Ray-Mi-1.uf2** — 样的功能。





4/4 自由演奏

有了函数的好处在于,如果要重复使用它 们,只需要通过简单的"调用"就可以。

假如想按下 B 按键时演奏以下乐段: Doe Ray Mi Mi Ray Doe Doe Ray Mi Doe Ray

现在的程序将会十分简洁和优雅,只需添加"当按键 B 按下"程序块,并按演奏顺序加入调用函数,如左图所示。

完成后,将程序下载到GameGo上运行。





扫码立即体验游戏 并获取程序





↔ 拓展玩法

在上面游戏的基础上增加一些拓展玩法,有兴趣的读者可以自己尝试。

- 在上面任务的基础上,补全其他4个音符的精灵,设置站位,以及播放函数 playFar、 playSew、playLa、playTea。
- 为A键设置使用7个音符演奏的函数调用。

🙆 知识点

带参数的函数

上面创建的函数只是一段程序,如果在调用函数的时候,能够为函数段内的程序指定一些数值、 文字甚至精灵的话,可以让函数的使用变得更加灵活和强大。

如下图所示,在上个任务创建函数的时候,在弹出的"编辑函数"对话框,可以看到"添加参数" 一栏有 4 个按钮: "文本" "布尔值" "数字" "Sprite"。

CARL REAL PLANE.	man should		TT BACK				
添加参数	工文本	24 布尔值	圖 以子	A Sprite			
				函数 pl	layDoe		

下面将通过任务2来了解,如何为函数添加参数,并在函数中使用这些参数。

▶ 任务2:使用带参数的函数进一步优化 Doe-Ray-Mi-3示例程序

仔细观察任务1创建的3个函数,它们的基本结构都一样,只是每个函数调用了不同的精灵, 每个精灵说了不同的话,播放了不同的音乐。可 以通过向函数添加参数,来进一步优化程序,把3 个类似的函数合并为1个。



动手实践



为函数添加参数

进入 Arcade 编辑器,继续使用上个任务的程序,或加载示例程序 arcade-L17-Doe-Ray-Mi-3.uf2。在屏幕下方修改项目 名为 Doe-Ray-Mi-4。

选中 playDoe 函数程序块,右击,弹出快 捷菜单,选择"编辑函数",如右图所示。

函数 playDoe

如右图所示,现在再次弹出了"编辑 函数"对话框。



期望添加的参数有:

- 精灵(为函数指定是哪个精灵进行表 演)。
- 精灵说的话(文本)。
- 演奏的音符(音符通常是数字)。

分别单击添加参数的"Sprite""文本"和"数 字"按钮,现在窗口内的函数框变成了右图所 示的样子。再单击"完成"按钮。





数字

mySprite 说 文本 持续 500 ▼ 毫秒 ○

将函数参数里红色椭圆的"文本"程序块 拖曳到 "mySprite 说 'Doe' 持续 500 毫秒" 程序块内, 替换"Doe", 如右图所示。

现在参数"文本"取代了函数里精灵说的话。

接下来将函数参数里红色椭圆的"数字" 程序块拖曳到"播放音符中C延时1/2节拍"

现在函数参数"数字"取代了播放的音符。

程序块里, 替换"中C", 如右图所示。

函数 play mySprite 文本 数字 改变 mySprite 的 y - 以 -10 mySprite 说 文本 持续 500 - 毫秒 👄 播放 音符 数字 到 1/2 ▼ 节拍 暂停 500 - 毫秒 ••••• 改变 mySprite 的 y v 以 10

函数 playDoe mySprite 文本

改变 mySprite 的 y ▼ 以 -10

播放 音符 中 c 延时 1/2 · 节拍 暂停 500 - 毫秒 改变 mySprite 的 y - 以 10

最后,单击函数名的位置,修改函数的名 称为"play",如右图所示,一个通用的函数



就搭建好了。





- 😣

回到"当按键A按下"程序块,原来的 playDoe 函数已经变成了 play 函数,单击 "mySprite"旁边的下三角按钮,选择"Doe", 为函数指定第1个精灵参数为Doe,如右图 所示。

如右图所示,在"文本"参数处,输入精 灵说的内容"Doe"。



如左下图所示,单击"音乐"标签,找到"中C"音符程序块,拖曳到函数的"数字"参数处, 如右下图所示。现在,一个完整的带参数的函数调用就实现了。



接下来,大胆删除原来的playRay、 playMi函数,然后重新为A键定义其他2个 调用。

如右图所示,复制"调用 play Doe'Doe' 中C"程序块,修改精灵为 Ray,说的文本修 改为"Ray"。单击"中C",会弹出一个音 阶选择图,可以指定这个函数播放音符"中D"。

最终的程序如下图所示,实现了只用一个 函数,通过指定函数的参数,让3个不同的角 色演奏不同的音符。











🕀 拓展玩法

在上面游戏的基础上增加一些拓展玩法,有兴趣的读者可以自己尝试。

■ 在任务2的基础上,补全其他4个音符的精灵,设置站位。

■ 为A键设置使用7个音符演奏的函数调用。





18 关 使用 "飞镖" 扩展制作 "愤怒的小鸟"





在编程时,软件开发人员通常希望避免花费时间重新创建已经写过的 代码。17 关学习了单个开发人员如何使用"函数"达到这个目的,但没 有解决开发人员之间如何共享代码的问题。本关将学习如何使用 Arcade 提供的"扩展"标签来简化游戏开发。另外还会学习如何自己创造扩展, 以及给小伙伴们分享自创的扩展。



🙆 知识点



了一个直观的弹射抛物线有一定关系。

▶ 任务1:使用 "darts" (飞镖)扩展发射小鸟

我玩过《愤怒的小鸟》的游戏后,觉得游戏开发者的弹射体验设计真的很赞,如果要自己来 实现弹射小鸟的功能,初学者会觉得有些无从下手。幸运的是,借助 Arcade 的"darts"(飞镖) 扩展,初学者也能较为轻松地做出类似发射小鸟的功能。



动手实践

1/3 添加 "darts" 扩展

在浏览器中打开 Microsoft Arcade 平台 (具体操作见 00 关)。

在 Arcade 主页新建项目,命名项目名称 为**发射小鸟**。

展开"高级"标签,单击"扩展"子标签, 在弹出的"扩展"列表里,单击"darts"扩展。 回到编程界面后,如左图所示^①,发现标签 栏的最上方出现了"飞镖"标签,单击它可以 看到有关飞镖的程序块,扩展添加成功。

2/3 发射小鸟

把"将 myDart 设为飞镖种类 Player"程 序块,拖曳到"当开机时"程序块里,如左图所示。

 注意,左图中"飞镖"栏的第一个程序块中有一处翻 译错误,"非标"应该是"飞镖"。后面出现该程序块时 也存在这个问题,不再重复说明。

213



214

 \mathbb{R}^{2}

从图库里选择黄色的小鸟作为精灵的形象 (如下图所示),现在从左图所示的模拟器画面 中可以看到小鸟出现在屏幕的左下角。



单击程序块右侧的"+"按钮,可以看到 这里可以指定小鸟的起始位置。为了方便观看 效果,把起始坐标修改为(x=10,y=60), 如下图所示。现在小鸟站在了屏幕左侧的中间 位置,如左图所示。









操作 Q

搜索....

0

9

场景

如左图所示,模拟器画面上已经可以看到, 小鸟真的被"扔"了出去,只是这个动作只会 自动运行1次,玩家并不能控制。

215

3/3 添加发射控制

现在小鸟已经能够被发射,不过我们期望 的操作控制是这样的: 玩家可以通过方向键调 整发射角度和距离,并在认为合适的时机,按 **下A键发射**。

如左图所示,再次单击"飞镖"标签, 看看还有哪些程序块可以用,先试试"跟 踪 myDart 路径估计",把它拖曳到编程 区。为了避免小鸟一开始就被扔掉,把"扔 myDart"的程序块先移出,如下图所示。





如下图所示,单击"跟踪 myDart 路径估计" 程序块右边的"+"按钮,设置为ture。现在 可以在左图模拟器里看到多了一条代表路径的 虚线,感觉和《愤怒的小鸟》有点接近了。



◎ 飞镖 路径估计 🕀 ▲ 精灵 myDart • ∞ 控制器 游戏 停止 myDart -音乐

控制 🛛 myDart 🔹 以箭头键 💮





如左图所示,找到并添加"控制 myDart 以箭头键"程序块到编程区。



如左图所示,现在已可以通过方向箭头, 调整路径虚线的角度和远近了。

如下图所示,单击"控制器"标签,添加"当 按键 A 按下"程序块,并将"扔 myDart"程 序块放入该程序块内。

为了方便发射小鸟,单击"飞镖"标签, 拖曳添加"改变 myDart 的角度 以 0"程序块, 并修改角度为 45 度,让小鸟直接处于待发射 的最佳角度位置。



现在只要按A键,就可以"扔"小鸟了。



将程序下载到 GameGo 上进行体验。

arcade-L18-发射小鸟 .uf2



扫码立即体验游戏 并获取程序

C 任务2: 分享"发射小鸟"项目,并在其他项目里调用

通过任务 1 获得了一只可以发射的小鸟。Arcade 还提供了分享功能,让用户能够通过链接 或二维码把自己制作的游戏分享出去,并可以在其他项目中调用。在任务 1 中为小鸟添加了按键 控制,移动方向键,可以调整发射小鸟的角度和距离。如果在新项目里也添加按键的控制程序, 会有什么影响? 来试试看。

动手实践

1/3 分享"发射小鸟"项目

加载 arcade-L18-发射小鸟.uf2 程序,如下图所示。





如右图所示,弹出"分享项目" 窗口,单击右下方的"发布项目"按钮。 项目发布后,会弹出一个新窗口, 给出一个分享项目的链接。除了获得 链接,还能把项目直接发布到社交媒 体或 MakeCode 的论坛。

分享项目窗口还提供了一个二维 码,如果用手机扫二维码,可以直接 在手机中打开分享项目的模拟器并试 玩。本书中所有示例项目的二维码,都 是通过这种方式获得的,如右图所示。

接下来,记得复制链接,并打开 一个记事本,将项目的共享链接粘贴 进去并保存。



2/3 建立新项目,导入"发射小鸟"项目

单击"主页"按钮,回到 Arcade 的首页。单击"新建项目" 按钮,输入新项目名称**发射小鸟共享**。

在新项目中,单击"高级"标签,然后单击"扩展"子标签, 打开扩展窗口。在扩展窗口的搜索框内,粘贴上一步创建的"发 射小鸟"项目共享链接的地址,单击搜索按钮,如下图所示。









3/3 检验导入的共享项目与现有项目的冲突

任务 1 为小鸟添加了按键控制,移动方向键,可以调整发射小鸟的角度和距离。如果在新项 目里也添加按键的控制程序,会有什么影响?让我们来试试看。

为新项目添加一个修改背景色的程序块。

添加一个小英雄的精灵到场景中。单击"控制器"标签,拖曳并添加"使用按键移动 mySprite"程序块。

219



完成的程序如右图所示。 如下图所示,在模拟器中使用方向按键, 可以看到抛物线和小英雄都在变化。



↔ 拓展玩法

第月期間 设置将京颜色为 将 mySprite・ 设为 府灵 (会) た型 Player・ 使用按键移动 mySprite・ (金)
arcade-L18-发射小鸟共 享.uf2
日码立即体验游戏 并获取程序

在"发射小鸟"程序的基础上,改造出一个真正可玩的游戏,下面是一些建议。

■ 在屏幕右侧添加随机出现的射击目标,如果小鸟命中目标,增加得分。

- 添加倒计时,玩家可以比较在倒计时范围内的得分高低。
- 共享自己改造的游戏,让其他小伙伴或父母体验你的游戏。







对大多数游戏而言,"动画"可能是制作游戏的基础功能。对于 Arcade 来说,默认的功能 并不包含"动画"功能,需要通过"animation"扩展,获得动画能力。



🕢 注 意

🕗 知识点

快速翻本书,看这个位置,便能体验"逐帧动 画"效果。

费纳奇镜与逐帧动画

早在1832年,比利时人约瑟夫·普拉陶 (Joseph Plateau)和奥 地利人西蒙·冯施坦普费尔 (Simon von Stampfer)发明了费纳奇镜 (Phenakistoscope),可播放连续动画,是早期无声电影的雏型。

费纳奇镜变体之一是在一个手柄上垂直安装的盘片(如右图所示)。盘片上围绕中心绘制了一系列图片,是动画对应的帧,图 片的周围是一系列狭缝。使用者旋转盘片,通过移动的狭缝看 盘片在镜子里反射的图像。使用者看到图片接连出现,由于视 觉暂留,便会得到连续播放效果。另一种变体有两个盘片, 一张有狭缝,另一张上面是图片,这样不需要镜子的反射就 可以看到动画。

如下页右图,一个女人透过圆盘的缝隙观看镜子中的 动画 (图片源自 E.Schule 在费纳奇镜圆盘盒标签上的插图 细节,大约绘于 1833 年)。

Arcade 图库中的动画素材

Arcade 使用"逐帧动画"的方式来实现动画效果。其 原理是在"连续的关键帧"中分解动画动作,也就是在时间轴 的每帧上逐帧绘制不同的内容,使其连续播放而形成动画。

通过添加精灵的程序块,打开熟悉的 Arcade 图库资源,如 下图,可以看到这里为许多角色都提供了逐帧动画的素材,下面将 学习如何让它们在游戏里动起来。





▶ 任务1: 让小丑鱼游起来

如下图所示,在 Arcade 的动画库里,可以找到一组 小丑鱼尼莫的图像,这是尼莫游动动画的 4 帧图像。下面 看看如何利用这 4 张图像,让尼莫在游戏里游起来。





动手实践

共享"发射小鸟"项目 1/2

在浏览器中打开 Microsoft Arcade 平台 (具体操作见 00 关)。 进入 Arcade 编辑器,新建项目,修改项目名称为尼莫动画。



如左图所示,在编程界面 中展开"高级"菜单,单击"扩展" 中标签,在弹出的"扩展"窗 口的列表里, 单击 "animation" 扩展。

如左下图所示,回到编程 界面后,发现标签栏新增了"动 画"标签,单击它可以看到有 关动画的程序块,扩展添加 成功。

让尼莫动起来

在"动画"标签栏里,有个"动画 mySprite"的程序块,先将它拖曳到编程区试

> 试。如左图所示,出 现了警告,原因是 还没有添加精灵到 场景中。

补上设置精灵的程序块后,警告消失了, 如左图所示。

接下来,单击 frames 右边有动画图标的 灰色方块。

如下图所示,出现了熟悉的编辑器,只是 右侧多了一个和动画有关的区域,这里是添加 动画帧的地方(红色方框区域)。





如下图所示,现在看到右侧有了第一张图。单击第一张小丑鱼图片下方的"+"按钮,重复 刚才的步骤,直到4张不同小丑鱼的图片全部被添加进来。





小丑鱼的帧序列上万,有个播放按钮,和 数字"500"(代表每帧播放时间为500毫秒), 单击播放按钮,感觉小丑鱼的动作有点迟钝, 将500毫秒修改为200毫秒,发现动作流畅 了许多,单击右下方的"完成"按钮。回到编 程区域,如左图所示,可以在模拟器中看到尼 莫动了,但只动了一下。

227

如右图所示,单击程序块 loop 右侧的开关 按钮,设置为"开"状态,现在尼莫动个不停了。 对于在海里的鱼儿,如果尼莫可以随着波 浪做一定幅度的上下摆动,就会更真实。

如下图所示,单击"动画"标签,有一个"动 画 mySprite with 飞到中心 for(ms)2000 loop 关"的程序块,尝试将它拖曳到编程区。









▶ 我的海底世界

根据上个任务的动画知识,我利用素材库搭建出了一个海底世界,读者可以直接查看海底世界的示例。

进入 Arcade 编 辑 器,加 载 示 例 程 序 arcade-L19-我的海底世界.uf2,然后将程 序下载到 GameGo 运行。

加载示例程序后,在模拟器中按A键,可 以释放一条珊瑚鱼游过屏幕,按B键可以释放 一个寄居蟹爬过屏幕。请仔细查看程序,了解 以下动画的实现方法。

- 珍珠贝的气泡动画是如何实现的?
- 按键 A 或 B 释放珊瑚鱼或寄居蟹的动 画是如何实现的?

💮 拓展玩法

使用素材库的动画素材,搭建自己的动画 世界。





