

小型消消乐

—— python 开源硬件创意编程实习报告

作者：张志云、黄浩宸

摘要：通过学习和运用 python 中 micro:bit 的编程语言，本小组设计了一款娱乐性的小型消消乐游戏。这款游戏将俄罗斯方块和消消乐相结合，对玩家的策略性和反应能力提出考验，兼具趣味性与益智性，给玩家带来了乐趣和挑战。

一、选题及创意介绍

本小组的选题是基于 micro:bit 开源硬件平台的消除方块小游戏。这类游戏是一种广受欢迎且具有挑战性的休闲游戏，结合 micro:bit 的编程和硬功能，可以锻炼玩家的逻辑思维和反应能力。本小组的基本想法是将经典游戏俄罗斯方块和消消乐结合起来，设计一款简单而有趣的小型消消乐，为玩家提供一种娱乐和放松的方式。

二、设计方案和硬件连接

本小组设计的这一游戏需要使用 micro:bit 主板和 LED 显示屏，同时利用按钮 A 和按钮 B 作为玩家的输入设备。玩家通过按压按钮 A、B 以控制亮度方块的左移和右移，玩家的目标是尽可能地消除更多的方块以获得更高的得分。

本小组设计的游戏规则如下：在每一回合中，LED 显示屏顶部的随机位置会生成一个随机亮度（3、6 或 9）的方块，并以一定速度下落至底部。其间，玩家通过操作按钮控制亮度方块左右平移，以确定亮度方块最终的位置。当有两个亮度相同的方块左右相邻时，方块会向左合并成一个更大亮度的方块（若当前已是最大亮度方块，即亮度为 9 的方块时，则不会合并；若二者上下相邻，也不会合并），而当最大亮度方块铺满整行时，整行自动清除，玩家可得 500 分。当其中一列堆满方块，即新生成的方块无法下落时，游戏结束，此时清算在 LED 显示屏上未消除的亮度方块，以方块亮度大小*0.7 作为分数计入玩家得分，并显示玩家最终得分。

三、实现方案及代码分析

在实现方面，我们使用了 python 编程语言，并借助 micro:bit 的库函数来控制 LED 矩阵的显示和按钮的输入。通过随机生成不同亮度的像素点并下降，玩家需要通过按钮 A 和按钮 B 来控制亮度方块的左右移动。游戏中的消除判断和分数计算也将在代码中实现。本小组的具体代码实现如下：

```

1 from microbit import *
2 import random
3 import music
4
5 score = 0
6 end = False
7 button_a.was_pressed()
8 button_b.was_pressed()
9 while not end:
10     pixel = random.choice([3,6,9])
11     x = random.randrange(0,5)
12     y = 0
13     stop = False
14     #产生的新的随机像素点位置已经有了像素点，结束整个游戏
15     if display.get_pixel(x,y) != 0:
16         end = True
17     else:
18         display.set_pixel(x,y,pixel)
19         sleep(500)
20     while not stop:
21         #清除之前位置的亮度
22         display.set_pixel(x,y,0)
23         #循环中A被按下
24         if button_a.was_pressed():
25             left = 0
26             if x > 0:
27                 for i in range(x-1,-1,-1):
28                     if display.get_pixel(i,y) != 0:
29                         left = i+1#右边一格是左边界
30                     break
31             x = max(left,x-button_a.get_presses())#设定最小边界
32         #循环中B被按下
33         if button_b.was_pressed():
34             right = 4
35             if x < 4:
36                 for i in range(x+1,5):
37                     if display.get_pixel(i,y) != 0:
38                         right = i-1#左边一格是右边界
39                     break
40             x = min(right,x+button_b.get_presses())#设定最大边界
41         #如果y达到最大边界，或者下面已经有像素点，结束
42         if y == 4 or display.get_pixel(x,y+1) != 0:
43             display.set_pixel(x,y,pixel)
44             stop = True
45         #没有达到最大边界，下面也没有像素点
46         elif y < 4 and display.get_pixel(x,y+1) == 0:
47             y = y + 1
48             display.set_pixel(x,y,pixel)
49             sleep(500)
50         judge_stop = False
51         #判断可消除的方块
52         while not judge_stop:
53             judge_stop = True
54             for j in range(5):
55                 clear = True
56                 for i in range(4):
57                     if display.get_pixel(i,j) != 9:
58                         clear = False
59                 #当有两个相同亮度的方块左右相邻时，合并为一个更大亮度的方块
60                 if display.get_pixel(i,j) == display.get_pixel(i+1,j) and display.get_pixel(i,j) < 9:
61                     judge_stop = False
62                     light=display.get_pixel(i,j) + 3
63                     display.set_pixel(i,j,light)
64                     for k in range(j,0,-1):
65                         display.set_pixel(i+1,k,display.get_pixel(i+1,k-1))
66                     display.set_pixel(i+1,0,0)
67                 if display.get_pixel(4,j) != 9:
68                     clear = False
69             #当一行全为最大亮度的方块时，整行消除，分数加500
70

```

```

71         if clear:
72             music.play(['c4:2','c4:2','d4:2','a4:2','b4:2'])
73             judge_stop = False
74             for l in range(5):
75                 display.set_pixel(l,j,0)
76                 for m in range(j,0,-1):
77                     display.set_pixel(l,m,display.get_pixel(l,m-1))
78                 display.set_pixel(l,0,0)
79             score += 500
80         sleep(500)
81
82     #游戏结束时，清算剩余方块
83     for j in range(5):
84         for i in range(5):
85             score += display.get_pixel(i,j)*0.7
86
87     #显示得分
88     music.play(['c4:4','g4:4','f4:3','d4:4','g4:2','b4:4','f4:3','g4:2','d4:1'])
89     display.scroll('score:'+str(score))

```

如图所示，在外层循环中，生成随机方块的亮度（3、6 或 9），并随机选择方块的 x 坐标，y 坐标初始化为 0。通过判断方块的位置是否被占用来确定是否结束游戏。在内层循环中，根据按钮 A 和按钮 B 的按下事件，判断方块是否可以向左或向右移动，并更新方块的 x 坐标。如果方块达到最底部或下方有方块存在，则停止移动；否则，方块每 0.5 秒向下移动一格。

当下落停止时，进入判断可消除方块的状态，即在循环中，遍历每一列，并判断当前方块与右侧方块的亮度是否相同且不为 0，如果满足条件，则进行合并操作，若合并后空缺处上方有方块，则按上方方块受“重力”下落。

与此同时，代码还设定了判断是否可以整行消除的部分，即在循环中先将清除标志“clear”初始化为 True，然后在遍历每一行的方块时，如果有方块的亮度不等于 9，则说明该行不是全亮度方块，将“clear”设置为 False。由于规则的设定，方块均为向左合并，故在遍历时只需遍历前四列，然而，在判断整行消除时，必须遍历全部的 5 列。因此，代码中还附加判断最右侧方块的亮度是否为 9，如果不为 9，则说明该行不是全亮度方块，仍需将“clear”设置为 False。

之后进入整行消除操作，如果“clear”为 True，则表示该行所有方块都是最大亮度的方块，进行整行消除操作，播放一段简单的音乐，并将当前行上方的所有方块向下移动一格，同时分数加 500。

当游戏结束时，计算剩余方块的分数，播放一段简单的音乐，并使用`display.scroll()`在屏幕上显示得分信息。

四、后续工作展望

通过这次的项目设计，我们不仅深入了解了 micro:bit 开源硬件平台的功能和应用，还锻炼了团队合作和解决问题的能力。我们成功地设计并实现了一款有趣的消除方块游戏，并在此过程中积累了丰富的编程经验。通过不断改进和优化，我们相信这个小游戏在未来还可以进一步发展和完善。例如，进一步完善游戏的功能和界面，引入不同的下落速度，作为不同难度的设定，以及引入不同形状的方块等。本小组将在后续工作中持续探索，以便让这款游戏增加更多的挑战性与趣味性。

五、小组分工合作

最后，此次项目的完成，离不开小组中两人的分工合作。通过团队合作，我们成功地完成了基于 micro:bit 的小型消消乐游戏的设计和实现，并取得了良好的成果。两人在具体讨论了游戏规则制定细节后，各自完成下落部分与消除部分的初步设计，编写代码进行测试。在项目中，我们更深入地体会到了编程和创造的乐趣。从提出设想到确认可行性，到努力克服编程过程中的种种困难，再到最后的模拟测试，亲自体验设计好的游戏，我们切实感受到了通过自己努力动手实现产品功能的成就感和趣味性！