

数据结构与算法-陈斌【C1】开源硬件创意作品报告

【C109】小组：张晟杰 陈斯熠

作品名称：重生之我是老师模拟器

摘要：重生之我是老师模拟器是一款基于掌控板的小游戏。这篇实习报告主要介绍了创意来源，游戏设计实况，python 代码分析及游戏玩法介绍，以及后续工作，实践感悟等部分。

一、 选题及创意介绍：

这次实践的初步想法是做一个小游戏，玩家意外重生成为老师，面对同学们的期末试卷，玩家开始协助调分，经过一些成绩的处理后，屏幕上显示玩家的调分结果，并给出一个评分，这个评分应当用系统内部的某个正态拟合来做。

在填写完报名表后，我们意识到 Microbit 内存很小，甚至可能连正态拟合模型都放不下，于是开始转变思路。

我们在借鉴了钢琴块之类游戏的玩法，最终决定做一款可以执行“捞捞”和“鼓励”操作的小游戏。

二、 设计方案和硬件连接

我们借鉴了钢琴块之类游戏的玩法，将对成绩的处理分为“捞捞”和“鼓励”两部分，分别设计了对应的奖惩积分机制。同时，为了增加游戏的趣味性，我们设计了一套剧情模式，尽可能地增加了互动性。最后，我们对作品进行了视觉美化，添加了灯效，添加了音效，设定了游戏难度（具体详见游戏玩法介绍）。玩家的得分将被记为“好评”，趣味性十足。

这款游戏的数据显示和操作都在掌控板上进行，无需其他外设，使用了“A”“B”“P, T, H, O, N”七个输入端，显示屏幕为 128*64，输出还有三个 RGB 彩灯以及 MUSIC，还使用了掌控板的环境亮度感知，姿态感知的功能。

三、 实现方案和代码分析

1. 游戏主体介绍

本游戏有游戏模式和剧情模式两种。

在剧情模式下，玩家需要按照指令完成调节亮度，摇晃掌控板等操作，并在剧情的引导下完成游戏（有点像新手教程）

在游戏模式下，共有六个难度以供选择，不同难度下得分会有不同，数字下落速度会变快，标准分刷新会更快，挑战更高难度以获得更多好评。

2. 实现方案（代码框架介绍）

代码封装了 snowball, bar, Game 三个类, snowball 用于开机动画演示, bar 用于创建可移动的数字对象, Game 是主体, 包括读写得分的函数（write_high_score , read_high_score ） , 游戏运行函数（game_start, game_running1, game_running2）, 页面函数（page）, 运行和控制函数（check, run）, 剧情模式（story, story2）

3. 主要代码分析

（1）

```

1. class bar():
2.     def __init__(self,mode,stack,rate,rang):
3.         self.mode=mode
4.         self.x=0
5.         self.y=16
6.         self.value=0
7.         self.stack=stack
8.         self.rate=rate
9.         self.rang=rang
10.    def set(self):
11.        if self.mode==0:
12.            self.x=randint(2,20)

```

这个类定义了数字对象的许多属性,都是为了契合 oled 显示的动态刷新,这与 turtle 作图有很大区别。

(2)

```

1.    rgb[0]=(25,25,25)
2.        rgb.write()
3.    import music
4.        music.play(music.PYTHON)

```

彩灯与音乐

(3)

```

1.    def check(self):
2.        while True:
3.            if button_a.is_pressed():
4.                oled.fill(0)
5.                oled.DispChar('下次再会', 50, 30)
6.                oled.show()
7.                time.sleep(2)
8.                oled.fill(0)
9.                image_picture = Image()
10.
11.                oled.blit(image_picture.load('face/4.pbm', 0), 32, 0)
12.                oled.show()
13.                time.sleep(1)
14.                oled.fill(0)
15.                oled.show()
16.                self.state=2
17.                break
18.            elif button_b.is_pressed():

```

```

18.                 self.state=3
19.                 break
20.             elif touchpad_t.is_pressed():
21.
22.                 self.state=4
23.                 break
24.             elif touchpad_h.is_pressed():
25.                 self.state=7
26.                 break
27.             elif touchpad_o.is_pressed():
28.                 oled.fill(0)
29.                 oled.DispChar(str(self.level), 60, 30)
30.                 oled.show()
31.                 while True:
32.                     if touchpad_p.is_pressed():
33.                         if self.level>0:
34.                             self.level-=1
35.                     if touchpad_n.is_pressed():
36.                         if self.level<5:
37.                             self.level+=1
38.                     elif button_b.is_pressed():
39.                         break
40.                     oled.fill(0)
41.                     oled.DispChar(str(self.level),
42.                                     60, 30)
43.                     oled.show()
44.                 self.state=5

```

check 函数，通过输入按键来返回一个页面或者状态

(4)

```

1. def run(self):
2.     while True:
3.         if self.state==0:
4.             self.game_start()
5.         elif self.state==1:
6.             self.check()
7.         elif self.state==2:
8.             while True:
9.                 if button_b.is_pressed():
10.                     self.game_start()
11.                 break
12.         elif self.state==3:
13.             self.game_running1()

```

```

14.         elif self.state==4:
15.             oled.fill(0)
16.             oled.DispChar('最高得分为
'+str(self.maxscore), 32, 15)
17.             oled.DispChar('按下 B 再来一次',20,30)
18.             oled.show()
19.             time.sleep(2)
20.             self.state=5
21.         elif self.state==5:
22.             self.page()
23.         elif self.state==6:
24.             self.game_running2()
25.         elif self.state==7:
26.             self.story()
27.         elif self.state==8:
28.             self.story2()

```

run 函数通过不间断的捕获状态来控制游戏运行

(5)

```

1.  def game_running1(self):
2.         rgb.fill((0,64,64))
3.         rgb.write()
4.
5.         nn1=[68, 63, 73, 78, 74, 75, 62, 66, 56, 67,
57, 76, 61, 69, 72, 65, 59, 79, 58, 64, 70, 71, 55, 77,
60]
6.         nn1+=[76, 72, 58, 62, 77, 67, 66, 64, 69, 59,
56, 57, 74, 61, 78, 79, 73, 63, 71, 70, 55, 75, 60, 68,
65]
7.         lx=16
8.         ly=62
9.         stack=[100,100,100,100]
10.        dl=133
11.        rate=4
12.        curnum=59
13.
14.        mode=1
15.        timetool=randint(0,30)
16.        cnt=0
17.        tmp=0
18.        nownum=nn1[timetool]
19.        judge=0
20.        obj=bar(randint(0,3),stack,randint(3,5),1)

```

```

21.         obj.set()
22.         stop=0
23.         dydown=0
24.         timeonly=0
25.
26.         stack=obj.stack
27.         error=0
28.         rightornot=1
29.         stop=0
30.         busy=1
31.         while True:
32.             if self.level==0:
33.                 timetool=cnt//25
34.             else:
35.                 timetool=cnt//(30-3*self.level)
36.                 timetool=timetool%(len(nnl))
37.                 if timetool!=tmp:
38.                     nownum=nnl[timetool]
39.                     tmp=timetool
40.                     oled.fill(0)
41.                     oled.vline(0,0,64,1)
42.                     oled.vline(32,16,64,1)
43.                     oled.vline(64,16,64,1)
44.                     oled.vline(96,16,64,1)
45.                     oled.vline(127,0,64,1)
46.                     oled.hline(0,16,128,1)
47.                     oled.pixel(lx, ly, 1)
48.                     oled.pixel(lx,ly+1,1)
49.                     oled.pixel(lx-1, ly+1, 1)
50.                     oled.pixel(lx+1, ly+1, 1)
51.
52.                     oled.DispChar(str(nownum)+str('捞捞')+' '+'
好评: '+str(self.score), 1, 0, 1)
53.                     if timeonly==64//(obj.rate):
54.                         busy=0
55.                         if judge==obj.mode and
stack[judge]<nownum:#没捞到还选中要扣分
56.                             self.score-=1
57.                             error+=1
58.                         elif judge!=obj.mode and
stack[judge]>nownum:#积极为学生服务,加分
59.                             self.score+=1
60.                     if not busy:
61.                         if self.level==0:

```

```

62.         obj=bar(randint(0,3),stack,randint(3,5),1)
63.             else:
64.         obj=bar(randint(0,3),stack,randint(self.level,5),1)
65.             obj.set()
66.             dydown=0
67.             stack=obj.stack
68.             busy=1
69.             stop=0
70.             rightornot=1
71.             timeonly=0
72.         if not stop:
73.             oled.DispChar(str(obj.value),obj.x,obj.y+dydown,1)
74.             else:
75.                 pass
76.             #oled.DispChar(str(nownum)+'
77.             '+str(stack[judge]), 2, 50)
78.
79.             oled.show()
80.
81.         if button_a.value()==0 :
82.
83.             if stack[judge]<nownum and
84.                 judge==obj.mode:
85.                 stack[judge]=100
86.                 stop=1
87.                 if self.level==0:
88.                     self.score+=obj.rate
89.                 else:
90.                     self.score+=self.level
91.                 rightornot=0
92.                 busy=0
93.             elif stack[judge]>=nownum and
94.                 judge==obj.mode:
95.                 stack[judge]=0
96.                 stop=1
97.                 self.score-=2
98.                 error+=1
99.                 rightornot=1
100.                 busy=0

```

```

100.
101.             if touchpad_p.is_pressed():
102.                 lx = 16
103.                 judge=0
104.             elif touchpad_t.is_pressed():
105.                 lx = 48
106.                 judge=1
107.             elif touchpad_h.is_pressed():
108.                 lx = 80
109.                 judge=2
110.             elif touchpad_n.is_pressed():
111.                 lx = 112
112.                 judge=3
113.
114.
115.             if dydown<=63:
116.                 dydown+=obj.rate
117.
118.             if rightornot==1:
119.                 if error==3:
120.                     for er in range(error) :
121.                         rgb[er]=(96,0,0)
122.                         rgb.write()
123.                     oled.fill(0)
124.                     image_picture = Image()
125.                     oled.blit(image_picture.load('face/11.pbm', 0), 32, 0)
126.                     oled.show()
127.                     time.sleep(1)
128.                     oled.fill(0)
129.                     oled.DispChar('第一个游戏结束',
130.                                     32, 20)
131.                     oled.show()
132.                     if self.score>=self.maxscore:
133.                         self.maxscore=self.score
134.                         self.write_high_score(self.maxscore)
135.                         oled.fill(0)
136.                         oled.DispChar('干得漂
                            亮!', 35, 15)
137.                         oled.DispChar('你获得了
                            年度最佳好评!', 0, 30)

```



```

137.         oled.DispChar(str(self.score)+'这个好评将被载入史册!', 0, 45)
138.                                     oled.show()
139.                                     import music
140.
141.         music.play(music.POWER_UP)
142.         while True:
143.             if
144.                 button_b.is_pressed():
145.                     break
146.
147.                 time.sleep(1)
148.                 self.state=8
149.                 break
150.             else:
151.                 for er in range(error) :
152.                     rgb[er]=(96,0,0)
153.                     rgb.write()
154.
155.                 cnt+=1
156.                 timeonly+=1

```

游戏函数，定义了初始随机列表，绘制像素，计时工具，busy 判断能否初入下一个 bar，每个创建的 obj 都有自己的私有计时器 tiomeonly, stack 存储当前光标选中框里的数据，judge 判断选中框，rightornot 判断按键是否正确，error 统计错误次数，level 决定的 dydown 的改变速度以及 score 的变化等

四、 小组分工

张晟杰负责代码实现，陈思熠提出修改意见，共同完善。

五、后续工作展望

(1) framebuffer

我们浏览了 framebuffer 的有关内容，感觉可以添加进去像素图案，使画面更生动。

(2) wifi/network

希望能加入联机模式

(3) 操作手感，误触仍需优化

*六、个人实践感悟（张晟杰）

这次 C1 编程中用了大概 11 天时间，5 月 2 日开始研究 mpython 文档，5 月 12 日代码完成，5 月 13 日完成提交，我的感悟和收获颇多。最开始学习经典的弹球，flyingbird 游戏实现，我感悟到了 oled 刷新的与众不同，这让我联想到了许多小时候玩的像素游戏，我对帧的理解更加深刻，也深深体会到了一个小小的游戏竟然需要这么长时间的思考与 debug。（曾经一个 up 主为了充实一款 flash 小游戏竟然占用了超过 200G 素材内存，花费了 6 个多月时间）同时，我也体会到了 python 代码封装的优越性，各部门（各个类）之间协同工作，接口扮演着相当重要的角色，各个类内部的修改与优化以及接口的稳定性是提高产品质量的前提和合作的保证。优化可以通过合理分配各个类、函数的工作，可以通过优化接口，可以通过优化类、函数内部结构，成功离不开每一个部门的通力合作。这个过程很累，作出的成果也不是方方面面都尽如人意，但作品成功运行的那一刻，是我最快乐、最满足的时刻。

