**Microbit实习报告：哈利波特辨音游戏**

李泽宇 2200012119 生命科学学院

概要：在本学期的开源硬件创意作业中，我应用在线模拟器，制作了一个以哈利波特故事为背景的，与音乐理论相融合的辨音游戏。下文为具体介绍。

一、创意来源

我本人是一个非常忠实的哈利波特迷，作品中构筑的奇幻魔法世界令我陶醉不已。在阅读其原著小说并且观看改编版电影时，电影版中由作曲大师John Williams先生创作的华丽精确传神的配乐强烈地吸引了我。我粗浅的研究并分析了其中各个主题所反映的情绪变化和音乐中与人物相对应的各人物主题音乐，印象极为深刻。同时，我还学习了多年的双排键电子琴。在编曲与制谱时常常遇到一些辨音难题，所以产生了一个想法，将二者结合起来：通过开源硬件小游戏的形式制作在哈利波特故事背景下，融入音乐理论中叙事思想的音乐编配，实现练习辨音能力的趣味小游戏。

二、设计方案

小游戏的设计过程主要分为四个方面：规则的显示与交代、随机音符的生成与播放、玩家的输入、系统对玩家输入结果的反馈。

1. 规则的显示与交代：通过microbit内置的speech功能，将故事背景与规则交代出来。第一版游戏中曾尝试用led显示屏来显示规则，但发现屏幕太小，阅读体验实在不佳，所以改为了通过语音提示。

2. 随机音符的生成：应用了python语言的random库，通过随机生成的0-7中的数字，创建列表对应【c,d,e,f,g,a,b】的几个音符，再通过随机生成音符的八度范围，组合生成并播放了待测的音符。

3. 玩家的输入：玩家输入采用按键式输入，用A按钮来选择答案，同时led显示屏上出现当前所选择的答案，每按7次循环出现‘’c,d,e,f,g,a,b’’，选择结束后通过B按钮来确定选项，确定后答案会闪烁两次。

4. 系统反馈：如果答案正确，led会显示对号，同时播放哈利波特的欢乐魔法主题音乐，提示答案正确；若错误，led会显示错号，同时播放伏地魔的邪恶主题音乐，提示答案错误。一次游戏共有三轮，采用积分制判断最后是否成功，每轮答对加一分，打错扣一分，初始积分为0，最终积分为正则可成功。三轮结束后，系统自动判断得分正负，相应led显示并且有语音提示反馈给玩家这一次游戏的最终结果。

三、实现方案和代码

# Imports go at the top

import random

from microbit import \*

import music

import speech

#语音提示

speech.say("Welcome to the magical world of Harry Potter! ",pitch=64,mouth=128,throat=128)

speech.say("Voldemort is tending to attack Hogwarts with his dark army",pitch=64,mouth=128,throat=128)

sleep(500)

speech.say("You can beat him by the force of music notes",pitch=64,mouth=128,throat=128)

speech.say("If you can tell which note it is for three times, you can beat the Dark Lord",pitch=64,mouth=128,throat=128)

speech.say("Now, let the game begin",pitch=64,mouth=128,throat=128)

display.on()

#开始播放哈利波特“奇异”魔法主题音乐

music.set\_tempo(bpm=86)

sleep(1000)

music.play(['b3:2', 'e4:3', 'g4:1', 'f#4:2','e4:4', 'b4:2', 'a4:6', 'f#4:6','e4:3', 'g4:1', 'f#4:2', 'd#4:4','f4:2', 'b3:10'])

sleep(100)

music.play(['b3:2', 'e4:3', 'g4:1', 'f#4:2','e4:4', 'b4:2', 'd5:4', 'c#5:2','c5:4', 'ab4:2', 'c5:3', 'b4:1','a#4:2', 'a#3:4','g4:2', 'e4:10'])

#积分初始化

score=0

speech.say("Get ready. Here come the notes",pitch=64,mouth=128,throat=128)

sleep(1000)

for round in range(0,3):

note\_reg = random.randint(0, 6) #生成随机音符

notelis = ['c', 'd', 'e', 'f', 'g', 'a', 'b']

note = notelis[note\_reg]

note\_range = random.randint(5, 6)

music.play([note+str(note\_range)+':12'])

t=-1

while True:

if button\_a.was\_pressed(): #答案选择输入过程

t+=1

display.show(notelis[t%7])

if button\_b.was\_pressed():

break

display.show(notelis[t%7])

sleep(500)

display.clear()

sleep(500)

display.show(notelis[t%7])

sleep(500)

display.clear()

sleep(500)

display.show(notelis[t%7])

sleep(500)

if t%7==note\_reg: #答案的判断与系统反馈

display.show(Image('00000:'

'00009:'

'00090:'

'90900:'

'09000'))

music.set\_tempo(bpm=120)

music.play(['c4:4', 'g4:4', 'g4:4', 'g4:4','f4:4', 'c5:4', 'e5:4', 'c5:4','e4:4', 'e4:4', 'd4:8'])

score+=1

else:

display.show(Image('90009:'

'09090:'

'00900:'

'09090:'

'90009'))

music.set\_tempo(bpm=120)

music.play(['a3:8', 'c4:8', 'g#3:8'])

score-=1

sleep(1500)

if score>0: #三轮结束后的结算与反馈

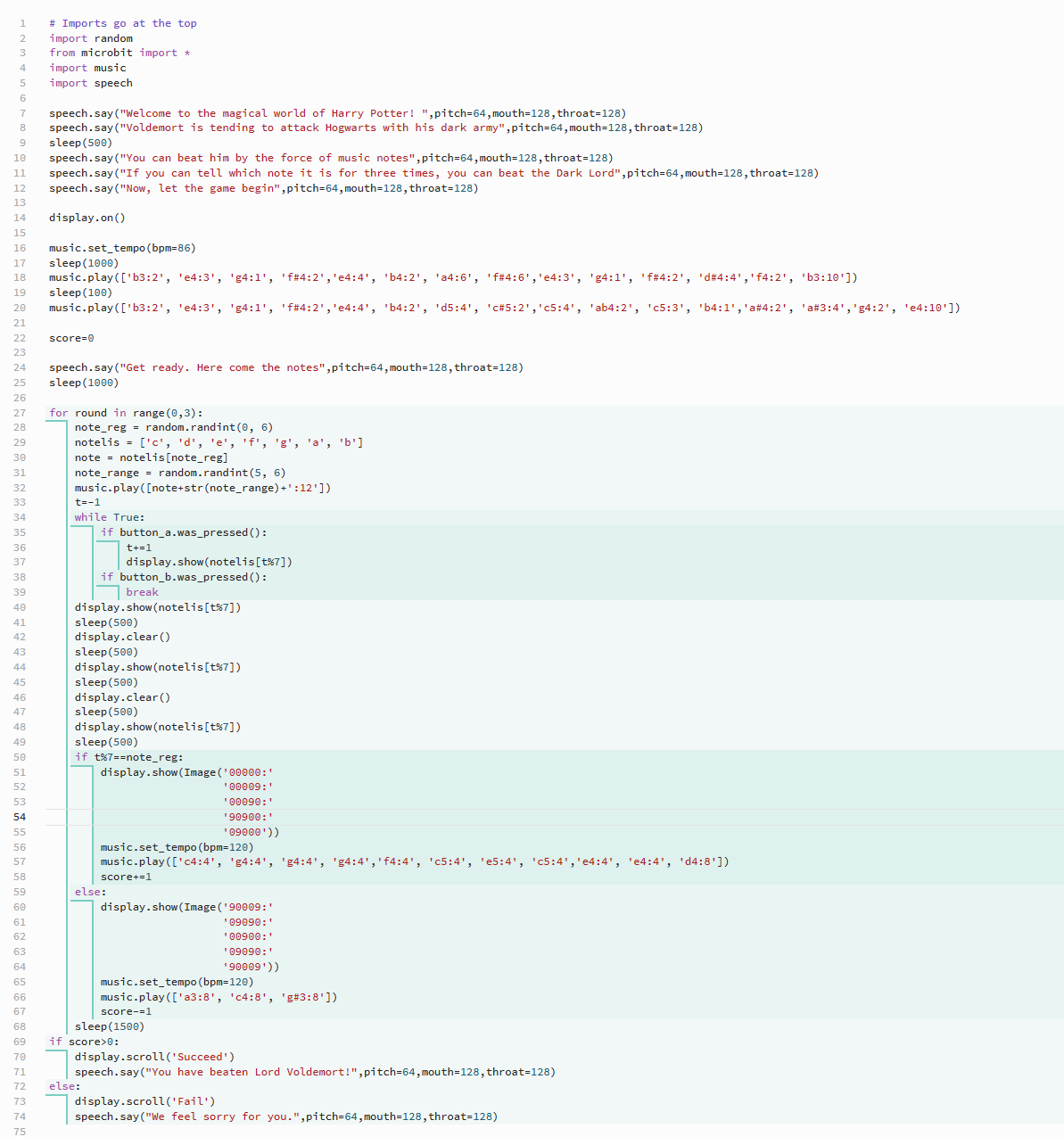
display.scroll('Succeed')

speech.say("You have beaten Lord Voldemort!",pitch=64,mouth=128,throat=128)

else:

display.scroll('Fail')

speech.say("We feel sorry for you.",pitch=64,mouth=128,throat=128)



四、后续工作展望

后续进行这项作品，希望可以延续这个思路扩展一些不同难度级别的模式，如低、中、高三模式挑战，逐渐进阶以提高其音符识别能力水平。中高模式可以采取减小音符播放时长或者采用乐段旋律形式进行播放，增加音符数量以提高玩家的灵敏度与反应力。

五、小组分工

这个小作品并未与他人合作，由一人独立完成。