

让micro:bit和音乐互动起来

谢作如 浙江省温州中学
张敬云 江苏省镇江市实验高级中学

涉及学科：音乐，数学，信息技术

当下火遍整个创客教育圈的莫过于micro:bit了。如图1所示，micro:bit是一个很小的可编程的微型电脑，能够使用makecode、Mixly、Python等编程语言设计智能作品。一直期望能够用micro:bit来识别音乐节奏，可惜micro:bit的功能太弱没办法实现，只能转而希望借助于电脑的帮忙，即让电脑识别音乐节奏，再将识别结果告诉micro:bit。这种做法其实就是在设计一款互动媒体作品。

互动媒体方面，最有名的编程软件当属Processing，Processing的Minim库中有个范例程序——窗口的矩形会随着音乐的节拍而跳动，效果图如2所示。我们准备借助这个库来识别音乐，然后实现micro:bit和音乐进行互动。比如，按下micro:bit的B键，音乐播放，按下micro:bit的A键，暂停播放，当音乐中有节拍时，则发送信息给micro:bit让LED点阵屏快速闪烁心型图像。

● 互动作品的原理分析

窗口的矩形随着音乐的节拍而跳动的效果是由Processing的Minim库中的“FrequencyEnergyBeatDetection”范例程序实现的，此程序是用频率能量模式对音乐中的节拍进行监测，使用快速傅立叶变换(FFT)算法来获得频谱，然后将频谱划分为平均频带，并且独立地监测在频谱的不同部分发出的声音(如底鼓、军鼓和小擦)。简单来说，此范例程序是对音乐的频率进行监测，而Processing窗口中矩形的出现以及位置便是由实时监测到的频率决定的。

那么，只要在这个范例程序中增加功能，当特定频率出现的时候，向micro:bit发送信息，就可以达到音乐互动的目的。可见，完成这个互动作品的关键在于让micro:bit和Processing之间相互通信。

Processing支持网络通信，通过WIFI进行互动本来是很好的方式，只是micro:bit自身不带WIFI功能，外加无线设备又比较麻烦。幸好

Processing和micro:bit都支持串口通信，二者的互动只能通过串口来实现。如果需要无线互动，则可以用一块micro:bit作为中转设备，多块micro:bit作为终端同时显示效果。

为了让互动更加有意思一些，我们在micro:bit中加上了音乐控制功能。这个互动媒体作品的运行流程如图3所示。

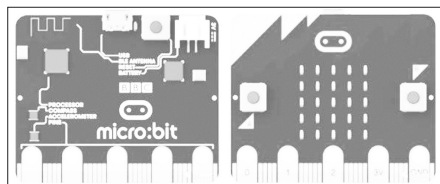


图1 micro:bit

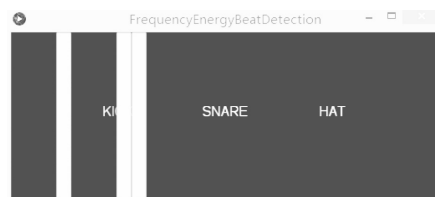


图2 范例程序效果图

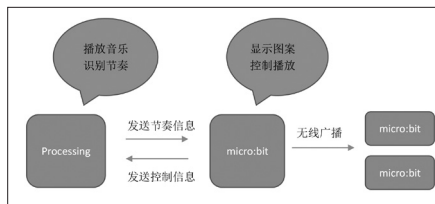


图3 系统运行流程图

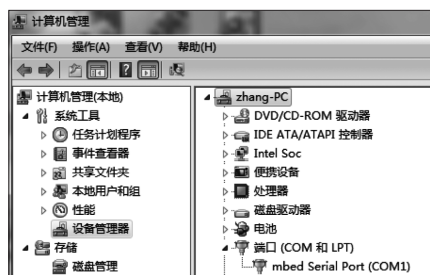


图4 设备管理器查看端口号

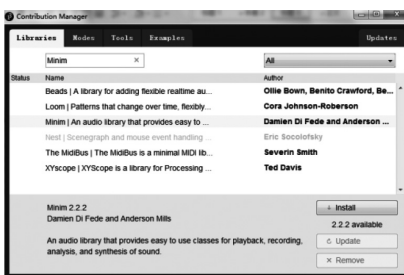


图5 安装Minim库界面

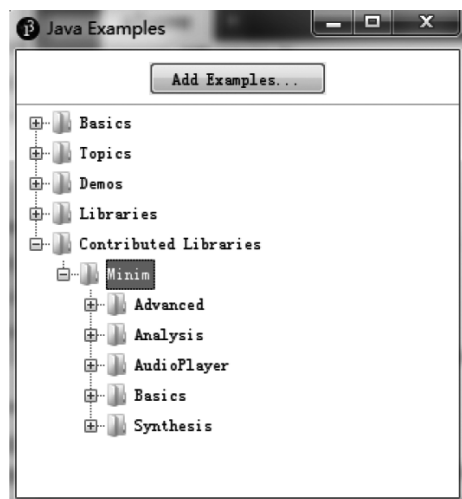
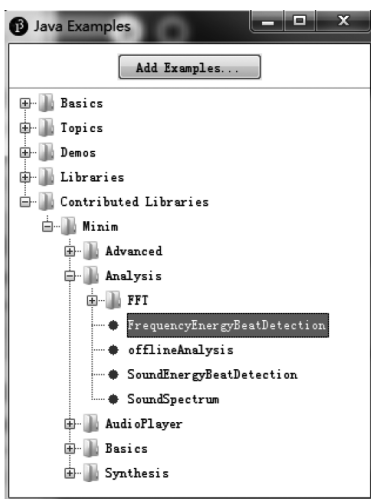


图6 Minim库范例程序

图7 范例
FrequencyEnergyBeatDetection

● 开发环境准备

1. 安装串口驱动

micro:bit在windows中需要另外安装Mbed串口驱动。而在Linux和Mac OS上无需安装驱动。下面以Win7为例介绍Mbed串口驱动的安装。首先连接micro:bit到计算机,然后双击Mbed串口驱动进行安装,安装完成后,便可通过设备管理器查看micro:bit的端口号了,如图4所示的COM1是笔者使用的计算机中micro:bit的端口号。端口号与计算机连接过的设备有关,不同的设备端口号不同。

2. 安装Minim库

本案例中,Processing要用到Minim库。3.3.6版本已内置了

此范例程序的基础上完成的。将代码另存后开始修改范例程序,主要增加了定义串口语句、接收信息语句以及发送信息语句(加下画线部分),以下是修改过的Processing代码(限于篇幅,仅提供部分代码,如果有需要请发邮件索取)。

Processing代码说明:运行程序,开始播放音乐,并且Processing窗口中的矩形会随着节拍的变化而变化,窗口上的文字KICK、SNARE、HAT也会随着各自的鼓点大小发生变化。当检测到Kick时,则向micro:bit发送信息K;当接收到信息为48(即数字0)时,则停止播放音乐;当接收到信息为49(即数字1)时,则继续播放音乐(如下页表1)。

2. 编写micro:bit代码

当micro:bit接收到信息K时,则LED显示图像;当按下A键时,则向Processing发送信息0;当按下B键时,则向Processing发送信息1。下页表2为micro:bit代码。

● 互动效果测试

运行程序,其测试效果如下页图8所示。随着音乐的律动,电脑的画面和micro:bit的LED屏幕也跟着闪烁,感觉很酷。因为手头的micro:bit不够,笔者没有测试多块micro:bit一起无线互动的效果,但理论上是没有问题的。实现无线也很简单,只要在代码中加一句发送,其他的板子接收并且判断这一信息即可。

● 代码编写

1. 修改Processign范例程序

范例中用Minim库中的BeatDetect实时读取Buffer中的数据并采集音频频率识别Kick(底鼓)、Snare(军鼓)和Hi-Hat(小擦)三种鼓点。本案例的音乐互动作品便是在

表1

```

.....
float kickSize, snareSize, hatSize;
import processing.serial.*;
Serial myPort;
class BeatListener implements AudioListener
{
.....
}
void setup()
{
.....
  textFont(createFont("Helvetica", 16));
  textAlign(CENTER);
  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 115200);
}
void draw()
{
//判断来自micro:bit的控制信息
  if ( myPort.available() > 0) { // If data is available,
    int val = myPort.
    read(); // read it and store it in va
    if (val==48){
      song.pause();
    }
    if (val==49){
      song.play();
    }
  }
  background(0);
  float rectW = width / beat.detectSize();
  for (int i = 0; i < beat.detectSize(); ++i)
  {
    if ( beat.isOnset(i) )
    {
      fill(0, 200, 0);
      rect( i*rectW, 0, rectW, height);
    }
  }
  int lowBand = 5;
  int highBand = 15;
  int numberOfOnsetsThreshold = 4;
  if ( beat.isRange(lowBand, highBand,
numberOfOnsetsThreshold) )
  {
    fill(232, 179, 2, 200);
    rect(rectW*lowBand, 0, (highBand-lowBand)*rectW, height);
  }
  if ( beat.isKick() ) {kickSize = 32;myPort.write('k'); }
  if ( beat.isSnare() ) snareSize = 32;
  if ( beat.isHat() ) hatSize = 32;
  .....
}

```

表2

```

from microbit import *
uart.init(baudrate=115200, bits=8, parity=None, stop=1)
d="09090:99999:99999:09990:00900"
i=9
while(1):
  if uart.any():
    incoming = str(uart.readall(), "UTF-8")
    incoming=incoming.strip(' ')
    if incoming=="k":
      i=9
  if (i>0):
    i=i-1
    temp=d.replace("9",str(i))
    display.show(Image(temp))
    sleep(20)
  if button_a.was_pressed():
    uart.write("0")
  if button_b.was_pressed():
    uart.write("1")

```

● 结语

和音乐互动的方式很多,这个互动媒体作品还有很多地方可以改良:①micro:bit的A、B键控制背景切换;②micro:bit的LED阵列屏朝上控制音乐播放,LED阵列屏朝下暂停播放;③用micro:bit的A键控制下一首音乐的播放,用micro:bit的B键控制上一首音乐的播放等。Processing是

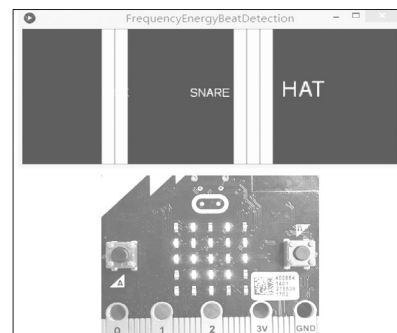


图8 程序运行效果图

一款很不错的互动编程软件,很多看起来很难的问题,如音乐识别、视频识别、语音识别等,都可以借助各种扩展库来完成。让学生们玩玩有趣的新技术从而爱上技术,这正是创客教育的核心工作之一。e

如果对相关内容感兴趣,请关注主持人博客。

