

Python Programming

Lecture 16 Mozart's Dice, Wordcloud, ASCII art

16.1 Mozart's Dice

MOZART MELODY DICER

MUSIKALISCHES WÜRFELSPIEL K. 516 F

COMPOSE
MUSIC
WITH DICE



COPYRIGHT, © 2013 BY CAROUSEL-MUSIC.COM

ISBN 978-0-935474-34-x

```
1 import wave # 导入 wave 模块，用于处理 WAV 音频文件
2 from random import randint # 从 random 模块导入 randint 函数，用于生成随机整数
3
4 def randomAudioFiles():
5     # 用于存储生成的音频文件名的列表
6     infiles = []
7     count = 1 # 初始化计数器
```

```
1 # 创建包含 16 个随机选择小节的列表，用于小步舞曲 (Minuet)
2 while count < 17:
3     file = "waves/M" + str(count) # 构造文件名前缀，例如 "waves/M1"
4     rollOne = randint(1, 6) # 模拟第一次掷骰子 (1 到 6)
5     rollTwo = randint(1, 6) # 模拟第二次掷骰子 (1 到 6)
6     rollTotal = rollOne + rollTwo # 两次点数之和决定小节编号
7     file += "-" + str(rollTotal) + ".wav" # 构造完整的文件名，例如 "waves/M1-7.wav"
8     infiles.append(file) # 将生成的文件名加入列表
9     count += 1 # 计数器加一，处理下一个小节
```

```
1 # 创建包含 16 个随机选择小节的列表，用于三重奏部分 (Trio)
2 while count < 33:
3     file = "waves/T" + str(count) # 构造文件名前缀，例如 "waves/T17"
4     roll = randint(1, 6) # 模拟一次掷骰子 (1 到 6)，决定三重奏小节编号
5     file += "-" + str(roll) + ".wav" # 构造完整的文件名，例如 "waves/T17-3.wav"
6     infiles.append(file) # 将生成的文件名添加到列表中
7     count += 1 # 计数器加一，直到生成 16 个文件名 (T17 到 T32)
8 return infiles # 返回包含所有随机音频文件名的列表
```

```
1 def audioGenerator():
2     # 获取随机生成的音频文件名列表，设置输出文件名
3     infiles = randomAudioFiles()
4     outfile = "dice.wav"
5
6     data = []
7     for infile in infiles:
8         w = wave.open(infile, 'rb') # 打开一个 .wav 文件进行读取
9         print(w.getparams()) # 打印音频文件的参数信息（如声道数、采样率等）
10        # 读取参数和所有音频帧，保存为列表形式
11        data.append([w.getparams(), w.readframes(w.getnframes())])
12        w.close() # 关闭文件
```

```
1 # 写入输出文件，生成合并后的 .wav 音频文件
2     output = wave.open(outfile, 'wb') # 以写入模式创建输出文件
3     output.setparams(data[0][0]) # 设置音频参数，采用第一个输入文件的参数
4     for val in data:
5         output.writeframes(val[1]) # 写入每段音频的帧数据
6     output.close() # 关闭输出文件
7     return infiles # 返回所使用的输入文件名列表
```

16.2 Wordcloud



Install WordCloud

- [Python 3.9安装问题的解决方案](#)
- Win系统安装方式: [WordCloud的安装文件下载地址](#)

- ```
1 pip install (安装文件的绝对路径)
```

- Mac系统安装方式: 先在终端输入如下命令

- ```
1 xcode-select --install
```

```
1 pip install wordcloud
```


A long string

```
1 from wordcloud import WordCloud
2
3 string = 'Importance of relative word frequencies for font-size.\
4 With relative_scaling=0, only word-ranks are considered. With \
5 relative_scaling=1, a word that is twice as frequent will have \
6 twice the size. If you want to consider the word frequencies \
7 and not only their rank, relative_scaling around .5 often looks good.'
8
9 font = r'C:\Windows\Fonts\Arial.TTF' # Windows系统
10 #font = "/System/Library/Fonts/STHeiti Medium.ttc" # Mac系统
11 wc = WordCloud(font_path=font, # Chinese
12               background_color='white',
13               width=1000,
14               height=800,
15               ).generate(string)
16 wc.to_file('s1.png') # save figure
```

```
1 from matplotlib import pyplot as plt
2 plt.imshow(wc) # show figure by plt
3 plt.axis('off')
4 plt.show()
```

importance frequent rank word
twice
frequencies
looks consider
considered
good size around
want often will
relative_scaling
font relative

Loading text file

```
1 from wordcloud import WordCloud
2
3 filename = 'Harry Potter.txt'
4 with open(filename, encoding="utf-8") as f_obj:
5     contents = f_obj.read()
6
7 font = r'C:\Windows\Fonts\Arial.TTF' # Windows系统
8 #font = "/System/Library/Fonts/STHeiti Medium.ttc" # Mac系统
9 wc = WordCloud(font_path=font,
10                background_color='white',
11                width=1000,
12                height=800,
13                ).generate(contents)
14 wc.to_file('s2.png')
```

see people Hagrid looked really anything
find day will head hand Aunt Petunia heard
inside outside try Percy reached cloak told place window nothing remember long point onto wizard letter said Hagrid Weasley
tell left put name front good something look now bit keep Quidditch even
two around room ter gave looking enough fire mirror took first Potter everyone next saw large
W Hermione said Malfoy Muggle
saying leg course almost seen want must across side everyone better
new lot quickly door gone school pulled yer
year came house think dark Yes an' turned mean Ron voice arm stop
feet second Quirrell full seemed let say table three never us done last always Filch found corridor wanted
black Hogwarts eye floor Oh said Ron kept Wood wand dragon thing owl anyone knew might thought asked another boy already moment time well Harry Dumbledore
great right open go started face though said Ron Dursley yeh night toward Snape bed book Neville without yet ground come
sat behind make Professor McGonagall still magic made half Dudley Neville yet ground come

Loading Chinese text file

```
1 from wordcloud import WordCloud
2
3 filename = 'sanguo.txt'
4 with open(filename, encoding="utf-8") as f_obj:
5     contents = f_obj.read()
6
7 font = r'C:\Windows\Fonts\simhei.ttf' # Windows系统
8 #font = "/System/Library/Fonts/STHeiti Medium.ttc" # Mac系统
9 wc = WordCloud(font_path=font,
10                background_color='white',
11                width=1000,
12                height=800,
13                ).generate(contents)
14 wc.to_file('s3.png')
```


Using jieba

```
1 from wordcloud import WordCloud
2 import jieba
3
4 filename = 'sanguo.txt'
5 with open(filename, encoding="utf-8") as f_obj:
6     contents = f_obj.read()
7
8 s = jieba.lcut(contents)
9 txt = " ".join(s)
10
11 font = r'C:\Windows\Fonts\simhei.ttf' # Windows系统
12 #font = "/System/Library/Fonts/STHeiti Medium.ttc" # Mac系统
13 wc = WordCloud(font_path=font,
14                 background_color='white',
15                 width=1000,
16                 height=800,
17                 ).generate(txt)
18 wc.to_file('s4.png')
```

[illegible]

Stopwords

中文常用停用词表

```
1 from wordcloud import WordCloud
2 import jieba
3
4 filename = 'sanguo.txt'
5 with open(filename, encoding="utf-8") as f_obj:
6     contents = f_obj.read()
7
8 def sw(filename):
9     with open(filename, encoding="utf-8") as f_obj:
10         x = f_obj.readlines()
11         y = [word.strip() for word in x]
12         return y
13
14 name_list = ['baidu_stopwords.txt', 'cn_stopwords.txt',
15             'hit_stopwords.txt', 'scu_stopwords.txt']
16
17 stop_word = []
18 for x in name_list:
19     stop_word.extend(sw(x))
20 stop_word = list(set(stop_word))
21
22 s = jieba.lcut(contents)
23 result = [word for word in s if word not in stop_word]
24 s = [word for word in result if len(word)>1]
25 txt = " ".join(s)
```


16.3 ASCII art

```
sansforensics@siftworkstation: /mnt/hgfs/Shared/crowdstrike-ctf/catapult-spider/much-sad
$ cat much_sad.txt
```

[illegible]

MUCH SAD?
1337 DOGE = 1337 DOGE
DKaHBkFEJKEf6r3L1SmouZZcxgkDPPgAoE
SUCH EMAIL shibegoodboi@protonmail.com

sansforensics@siftworkstation: /mnt/hgfs/Shared/crowdstrike-ctf/catapult-spider/much-sad

```
1 from moviepy import *
2
3 clip = VideoFileClip("ultraman.flv")
4 clip = clip.subclipped(0, 6) #剪切前6秒
5 clip.write_gif("video.gif") #保存为gif
```

```
1 import os
2 exist = os.path.exists('images') # 判断是否存在文件夹
3 if not exist:
4     os.mkdir('images')
5 else:
6     pass
```

```
1 from PIL import Image
2 #将gif的每一帧保存为png图片到images文件夹
3
4 def get_imgs():
5     gif = Image.open('video.gif')
6     try:
7         gif.save(f"images/{gif.tell()}.png")
8         while True:
9             gif.seek(gif.tell()+1)
10            gif.save(f"images/{gif.tell()}.png")
11    except Exception as e:
12        print("finished")
13
14 get_imgs()
```

图片转ascii图

- RGB色彩模式是通过对红(R)、绿(G)、蓝(B) 三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色 通常情况下, RGB各有256级亮度 其区间为[0,255] 也就是256个阶梯
- 灰度图是指只含亮度信息, 不含色彩信息的图象, 就象我们平时看到的黑白照片: 亮度由暗到明, 变化是连续的。
- rgb向灰度值转换的公式是: $\text{gray} = 0.299r + 0.587g + 0.114b$
- 所以灰度值的区间是 $[0, (0.299+0.587+0.114)*255] \sim [0, 255]$

```
1 # 设置要用的ascii码
2 ASCII_CHAR = "$@B%8&WM#*oahkbdpqwmZO0QLaCJUYXzczjhdsdavunxr \
3 jft/\|()1{}[]?~<>i!lI;:,\\"^`'."
```

```
1 def rgb_to_ascii(r, g, b, alpha=256):
2     # 通过灰度值的映射
3     # 将每一个rgb值对应成一个ascii符
4     # 也就实现了rgb -> ascii
5
6     # 当像素透明时，直接返回一个空白字符串
7     if alpha == 0:
8         return ' '
9
10    length = len(ASCII_CHAR)
11    gray = int(0.299 * r + 0.587 * g + 0.114 * b)
12    # 灰度值和字符串的对应关系
13    # 每个字符串对应灰度值的区间是
14    unit = (256.0 + 1)/length
15    # 找到灰度值所对应字符串的下标
16    index = int(gray/unit)
17    return ASCII_CHAR[index]
```

```
1 def image_to_ascii_chart(image):
2
3     width, height = image.size
4     text = ''
5     for y in range(0,height,10):
6         line = ''
7         for x in range(0,width,5):
8             # 找到对应位置的像素点
9             dot = image.getpixel((x, y))
10            line += rgb_to_ascii(*dot)
11        text += line
12        text += '\n'
13    return text
```

```
1 # 每个图转化为字符串
2 files = os.listdir('images')
3 xxx = []
4 for i in range(0,len(files)):
5     pic = Image.open(f'images/{i}.png').convert('RGB')
6     xxx.append(image_to_ascii_chart(pic))
```



```
1 def Start():
2     os.system("clear")
3     print('Press any key...')
4     input()
5     os.system("clear")
6     i=0
7     print(xxx[i])
8     c = input()
9     while c != 'q':
10         os.system("clear")
11         print(xxx[i])
12         c = input()
13         i=i+1
14     print('End')
15
16 Start()
```

Summary

- Mozart's Dice, Wordcloud, ASCII art

