

项目说明文档

用户交互技术 ——语音识别交互设计

姓 名： 范 思 琪

学 号： 1752095

同济大学

Tongji University

目录

一	设计	3
	1. 需求分析	3
	2. 对资料的整理分析	3
二	实现	3
	1. 功能实现	3
	2. 界面实现	4
三	改进	6
	1. 语音识别的准确性	6
	2. 提高.....	6

一 设计

1. 需求分析

本次项目的目标是在用户界面下实现语音交互功能。即为，运行程序，程序可以根据用户的语音指令执行一定的操作。

为了完善用户界面，需要使用 PyQt5 中的函数。为了实现语音识别，需要使用 speech_recognition 的功能。根据识别得到的用户指令，实现相应操作，则需要将语音识别得到的文本结果存入变量中，再判断其中的关键字符合那种操作的要求，并访问相应路径运行对应程序或文件。

2. 对资料的整理分析

根据示范中的 guessTheWord.py，我们可以得到使用 speech_recognition 识别并输出用户语音的方法。将其功能总结为一个函数加入到 asr.py 中，从而实现项目的识别语音功能。

更改示范中的 asrInterface.py 文件，从而改变 asr.py 中调用的界面设置函数，实现调整界面的功能。

另外还需要增加的功能是根据用户的语音指令实现对应操作。由于在识别语音部分已经得到了用户的语言指令，现在只需要对其进行关键词对照。如果具有对应关键词，即可直接调用对应执行程序。

二 实现

1. 功能实现

根据 guessTheWord.py，可以概括出 recognize_speech_from_mic 函数如下：

```
def recognize_speech_from_mic(recognizer, microphone):
    if not isinstance(recognizer, sr.Recognizer):
        raise TypeError("`recognizer` must be `Recognizer` instance")

    if not isinstance(microphone, sr.Microphone):
        raise TypeError("`microphone` must be `Microphone` instance")

    with microphone as source:
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)
        command=recognizer.recognize_sphinx(audio)

    return command
```

其中，将识别语音敏感度调整到此时适合环境噪音的状态，提高识别的敏感度。之后监听用户输入的语音，并利用 recognize_sphinx 将其识别并赋值给 command。并且将 command 作为函数返回值，便于之后 main 函数的调用。

实现了语音识别功能之后，编写 main 函数如下：

```

if __name__ == "__main__":
    app = QtWidgets.QApplication([])
    application = myWindow()
    application.show()

    print("Begin speaking")
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()
    phrase = myWindow.recognize_speech_from_mic(recognizer, microphone)

    print(phrase)

    if "play music" in phrase:
        win32api.ShellExecute(0, 'open', 'E:\\song.wma', '', '', 1)
    elif "playing music" in phrase:
        win32api.ShellExecute(0, 'open', 'filecapae.wav', '', '', 1)
    elif "open note pad" in phrase:
        win32api.ShellExecute(0, 'open', 'notepad.exe', '', '', 1)

    sys.exit(app.exec())

```

其中，在用户界面出现后，给出“Begin speaking”的提示，并调用语音识别函数将用户指令赋值给变量 phrase。检查 phrase 中是否有相应关键字，如果存在，则启动相应程序。

经过实践我们发现，控件对于用户语音“play music”的识别经常出错，识别为“playing music”，所以增加了词组“playing music”作为播放音乐的触发短语。而对于“notepad”的识别，返回的值是两个词的短语形式，所以“note pad”也作为打开记事本的触发短语。

2. 界面实现

考虑到可能存在语音识别不准确，用户需要手动输入或修改命令的情况，我在原有的界面之上增加了文本框，便于下一步功能的改进和提高。

文本框上方的提示语设置如下：

#文本框上方提示

```

self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(90, 250, 450, 65))
font = QtGui.QFont()
font.setFamily("Calibri")
font.setPointSize(14)
self.label_5.setFont(font)
self.label_5.setStyleSheet("color: rgb(0, 117, 210);")
self.label_5.setWordWrap(True)
self.label_5.setObjectName("label_5")

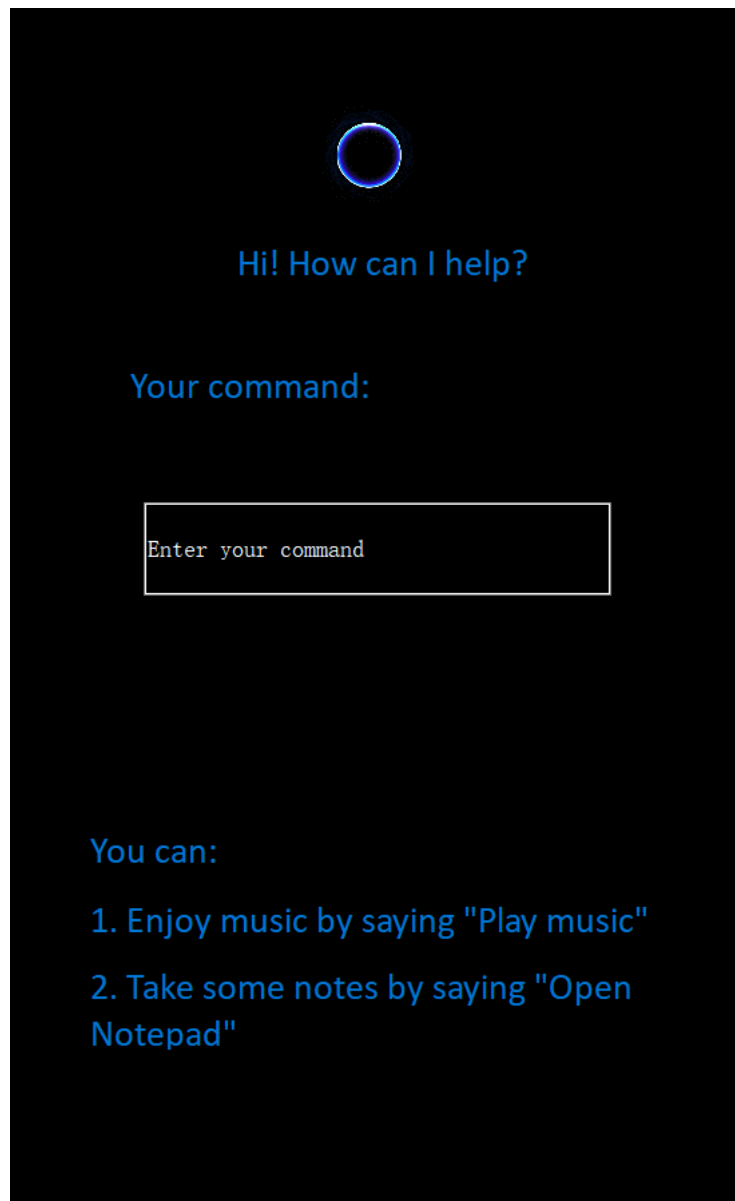
self.label_5.setText(_translate("MainWindow", "Your command:"))

```

文本框设置如下：

```
# 输出文本框
self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)
font = QtGui.QFont()
self.lineEdit.setText("Enter your command")
font.setFamily("Calibri")
font.setPointSize(14)
font.setWeight(75)
self.lineEdit.setGeometry(QtCore.QRect(100, 370, 350, 70))
self.lineEdit.setObjectName("lineEdit")
self.lineEdit.setStyleSheet("color:white")
```

改变后的界面如下：



三 改进

1. 语音识别的准确性

经过多次实验可以发现，当用户说出的指令为含有关键字的长句时，语音识别更加准确。而当用户只说出关键字时，识别的准确性较差。

2. 提高方法

- (1) 在实验中可以发现，有一些字极易被误识别成类似读音的字：如“play”经常会被识别成“playing”。所以，我们可以在 class 中建立一个命令集，一方面便于添加其他关键短语和命令的对应关系，另一方面也便于为用一个操作添加更多相似的关键字集。
- (2) 可以把识别出的用户语言内容重定向输出到界面的文本框中，用户可以在此基础上进行修改，然后再调用命令。
- (3) 可以增加一个判定函数，如果用户识别后的语音指令未包含关键字集中的内容，提示用户继续输入语音指令，直到包含关键字内容。
- (4) 经过资料查找可以发现，speech 包更为常用，可以考虑更换更准确的语音识别控件。