

**《人机交互》**

**课程报告**

|  |  |
| --- | --- |
| 学生姓名 | 谭哲文 |
| 学 号 | 8202191123 |
| 专业班级 | 计科2105 |
| 指导教师 | 赵欢喜 |
| 学 院 | 计算机学院 |
| 完成时间 | 2024-1-20 |

目录

实验一：web 界面设计··················03

1. 实验目的和要求···················03

二. 预备知识与实验指导·················03

三. 实验任务······················06

四. 实验内容与步骤···················06

五. 界面设计的实验结果展示···············07

六. 给出自己设计的界面的设计原则与评估分析·······09

实验二：语音智能翻译设计················10

1. 实验教学的目的···················10
2. 实验教学的任务···················10
3. 具体实验题目名称和实验指导说明···········10
4. 实验内容和步骤···················15
5. 实验结果展示····················28

1.展示界面设计，给出演示结果··············28

2.小结实验心得体会···················30

人机交互实验一：web界面设计

**一 实验目的和要求**

1) 熟悉Web站点的信息交互模型和结构

2) 熟悉Web界面设计的基本思想和原则

3) 掌握Web界面中静态网页设计的工具和技术

4)了解人机交互系统的评估方法

5)掌握人机交互系统评估的步骤

**二 预备知识与实验指导**

Web界面设计是人机交互界面设计的一个延伸，是人与计算机交互的演变。Web界面设计与站点外观直接相关，站点的界面外观是否友好直接关系到是否能吸引人的关注。人性化的设计是Web界面设计的核心，如何根据人的心理、生理特征，运用技术手段，创造简单、友好的界面，是Web界面设计的重点。

**(1) Web信息交互模型**

用来解释Web的人机界面性质的一个模型，它提出网页是用户和知识之间的界面。对于信息提供者来说包括信息的表达，对于使用者来说则是信息的获取。信息的表达与获取分别受到两者认知结构的制约。

模型涉及到信息的三种类型

1)数据：当一条信息被反复、简单的提供时称为数据，比如机票价格。

2)复杂信息：而用来叙述事件时称为复杂信息，如多媒体信息。

3)过程性信息：在信息有明确目标，并相互作用时称为过程性信息，如在线练习、在线测试等。

模型涉及到信息的两种特性：1)动态性：信息在不断的变化，具有动态性；2)一致性：信息元素的组织方式具有一致性

**(2)Web信息设计模型**

是解释Web人机界面性质的另一个模型，是一种研究网页的信息设计模型。

设计模型中要考虑到信息的两个方面：1)第一是应该呈现或略去什么信息。 2)第二个方面指的是信息该如何被表现。

**(3)Web界面设计基本原则**

1. 了解浏览者的心理状态
2. 内容与形式的统一
3. 减少浏览层次
4. 特点明确
5. 统一整体的形象
6. Web界面设计的3C原则

**(4)Web界面概要设计**

1)Web界面框架设计：Web网站规划 、建立原型系统、详细设计、正式实施

2)Web界面的内容与风格的设计:网站内容设计的原则、Web界面的风格

3)Web界面设计的语言与文化：网站应设置多语言选择，在网站设计和建设中进行跨文化研究

**(5)Web界面设计要素**

* + Web界面布局
* Web界面的色彩
* Web界面的字体
* Web界面的动画与多媒体
* Web界面的导航

**(6)Web界面设计技术与工具**

web界面设计技术涉及服务器端（ 包括Web服务器端与应用服务器端）与客服端的开发语言与各种框架，但本实验主要讨论利用Web界面可视化的开发工具开发，一般不涉及各种各种框架与编程语言。

要设计好的Web界面，需要有良好的设计工具，随着Internet网络的发展，国际组织和许多互联网软件开发商制订了若干标准，开发了不同的Web界面设计工具。比较早一点的主要技术包括超文本标记语言HTML、客户端脚本语言JavaScript、JavaApplet、服务器端脚本语言。

早一点的页面编辑器主要包括：1)MicroSoft公司 的Frontpage；2)Macromedia公司 的Dreamweaver，Dreamweaver对于动态网页的支持特别好，可以轻而易举地做出很多眩目的互动页面特效，Dreamweaver与Flash、Firework并称为Macromedia的网页制作三剑客。

**三 实验任务**

**1、根据自己的爱好从上面选择适当的开发工具，如从https://www.xiaopiu.com/ 下载xiaopiu 界面原型设计工具，上网查资料学习xiaopiu的使用，掌握xiaopu这个开发工具。**

**2、 参照目前本校的WEB/移动网站内容，采用选用的开发出一个中南大学计算机学院的 WEB或移动界面（建议写移动界面）主页（导航页，相关资料可参看院目前网站。请写出详细设计过程并完成实验第四步与第五步。**

**3、根据界面设计理念与原则总结出你的设计思路，你的界面体现了什么原则或具有什么优点。并完成实验第六步**

**四、 实验内容与步骤**

(一) 实验内容:

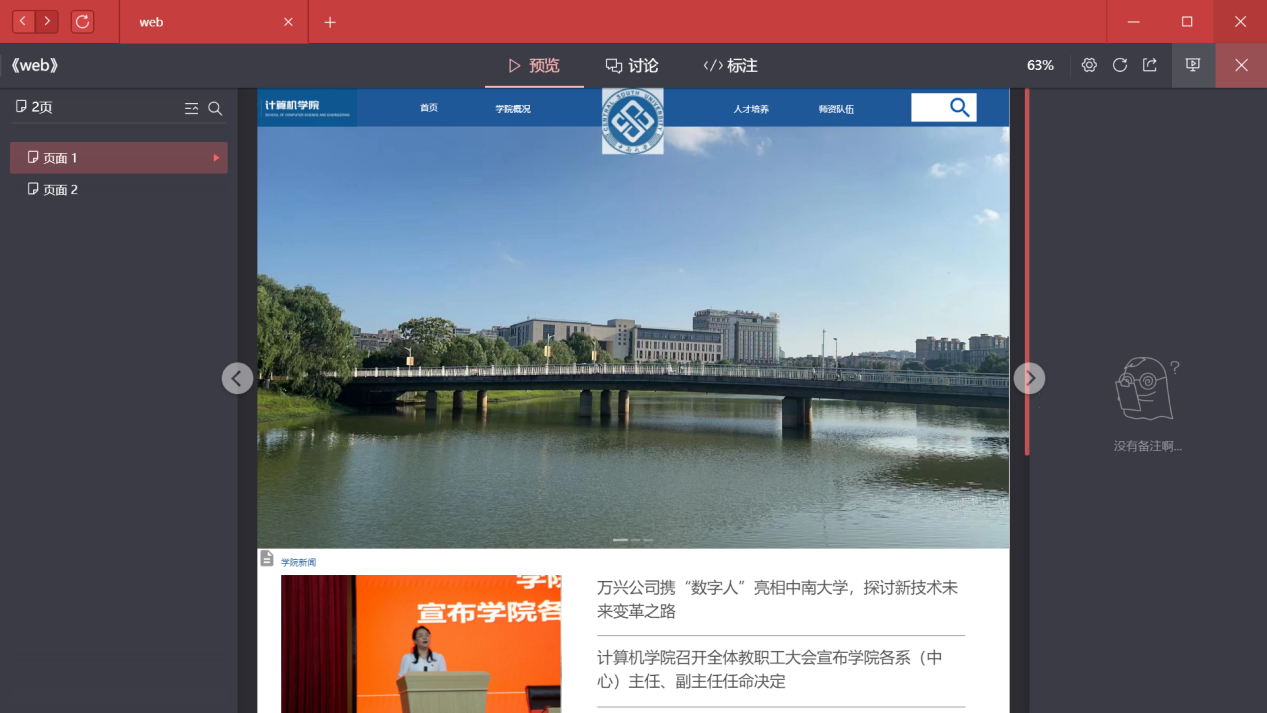
要求根据Web/移动界面设计的原则(简洁、一致性、对比度)，进行Web/移动界面规划、概要设计和设计要素的选择，利用xiaopiu设计工具完成网页设计。

(二)实验步骤：

利用了xiaopiu轮播功能设计了几张图片，设计了中英双字，搜索功能并未设计，暂时留存

1. **界面设计的实验结果展示**

**界面首页：**



轮播功能：

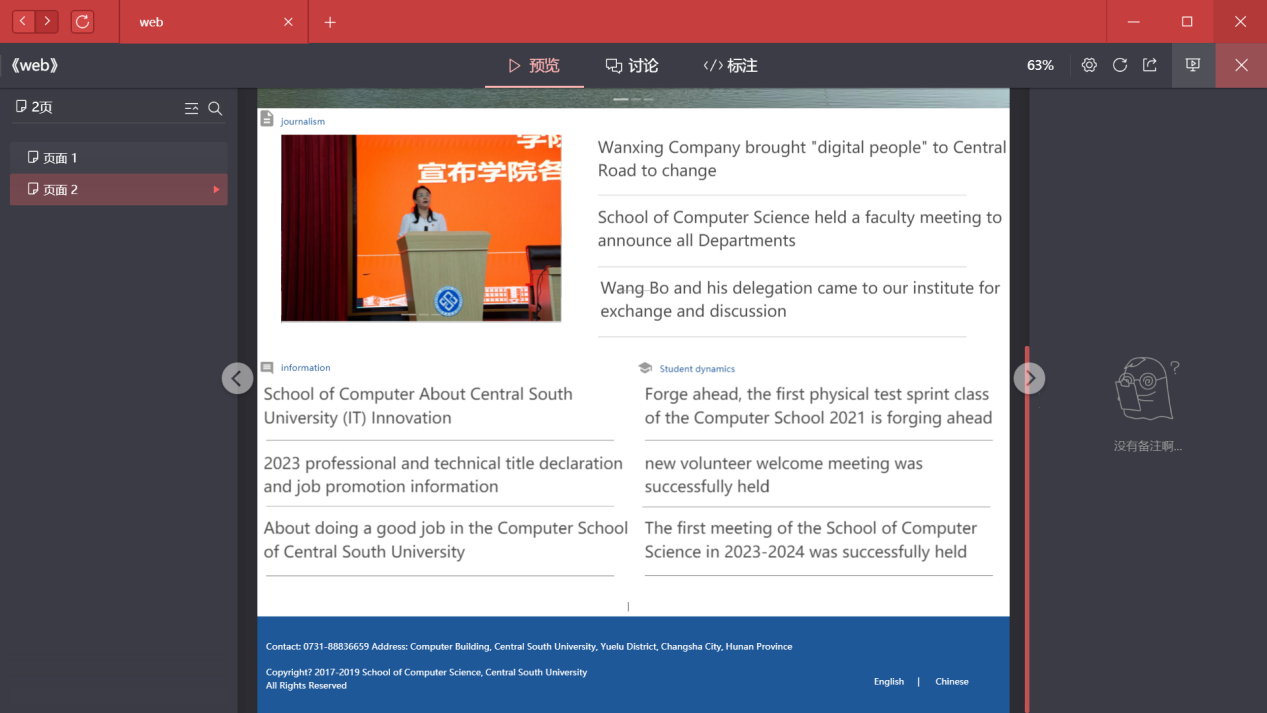


**校内新闻：**





中英双字转换：



**六、 给出自己设计的界面的设计原则与评估分析**

界面设计：主要以简洁明了来设计页面，界面设计简单却分门别类，便于查找，为母语为中文和英语的不同类群设计了语言跳转功能，方便学生阅读，直观性很不错，也比较灵活界面设计是直观的，用户无需过多的思考或培训即可理解如何使用系统。界面设计支持用户以最小的认知和操作负担完成任务，提高用户的工作效率。

人机交互实验二：语音智能翻译设计

一．实验教学的目的

通过本实验课程的实践，使学生了解《人机交互》与计算机图形、 程序设计、认知心理学以及计算机硬件的发展等领域密切相关，并加深学生对人机交互知识的理解，增强学生的实际运用能力和开发高可用性的具有自然交互特性界面的能力。

二．实验教学的任务

通过实验学习，让学生了解不同的人机交互模型设计类型，以及成功与失败案例所带

来的启示。通过原型设计使学生了解原型的作用，并了解用户需求对设计一个良好人机交

互界面的重要性。通过原型和界面评估，使学生掌握针对交互系统的评估方法。

1. 具体实验题目名称和实验指导说明

（一）、实验目的

1． 借助一个应用场景，通过开发一个简单式“能听“或”会说“、”懂你”的智能人机交互系统，使同学了解多通道交互系统的优点和交互模型的构建方法。

2．理解语音合成技术的基本原理。

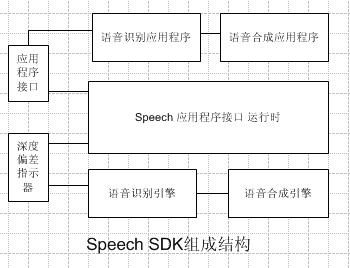
3．了解和掌握阿里智能语音交互平台//百度智能语音开发平台/讯飞智能语音开发平台/Microsoft speech sdk的使用方法。

   （二）、实验器材

1．配有语音卡、耳麦、扬声器等多媒体设备的PC机

2. Windows操作系统

3. 阿里智能语音交互平台//百度智能语音开发平台/讯飞智能语音开发平台/Microsoft speech sdk



作为应用程序开发者，只需调用speech sdk中的函数，即可实现语音合成与语音识别功能。

3．下载安装

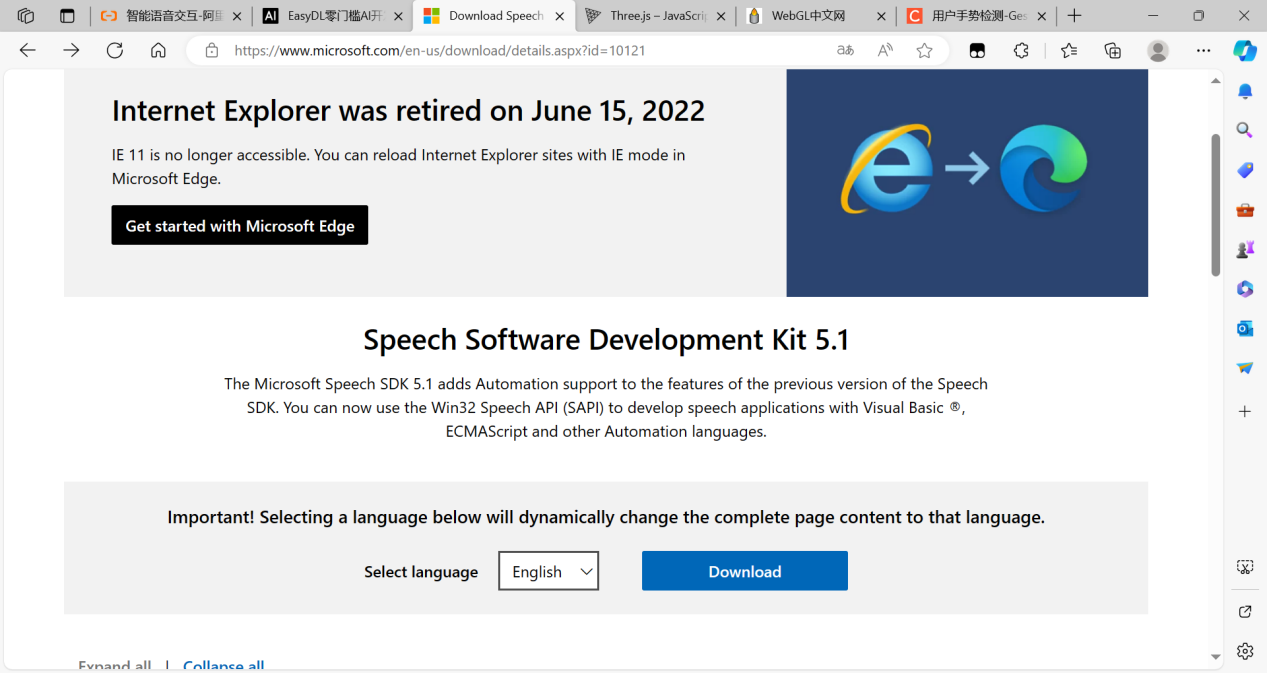
从Microsoft官方下载站点搜索Speech SDK 5.1并下载安装，

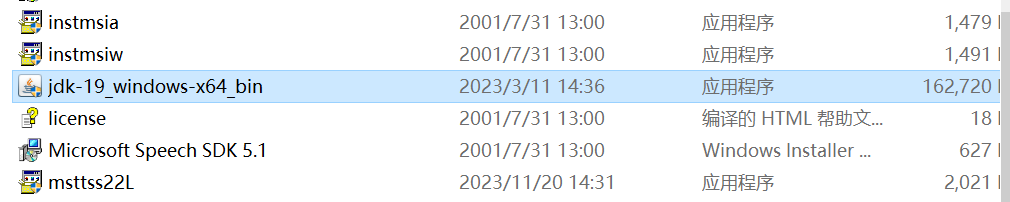
官网地址：http://www.microsoft.com/en-us/download/details.aspx?id=10121

下载三个关键的程序：SpeechSDK51.exe 、 msttss22L.exe、SpeechSDK51LangPack.exe

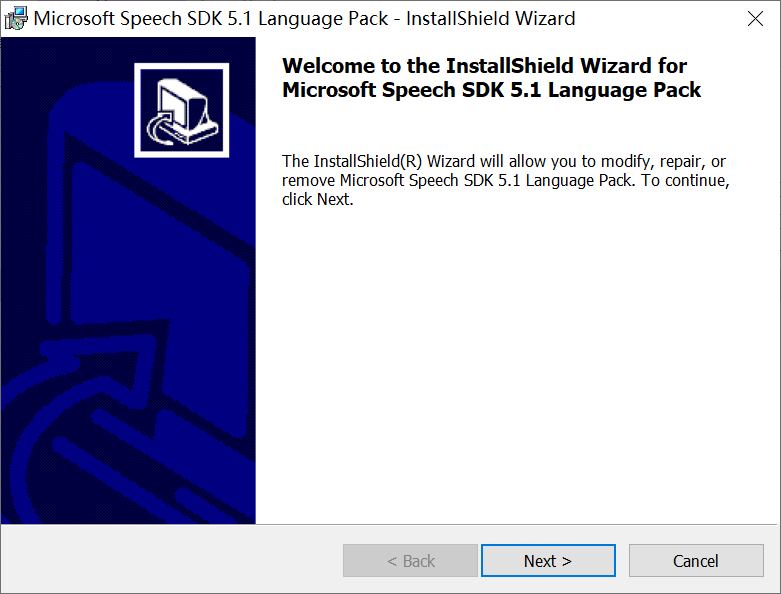
中文语言包可选择安装。安装好Speech SDK后，语音控制程序将被添加到 “控制面板”->“语音”选项中。利用下图对话框可以设置语音识别和文字-语音转换的各项属性，包括语言、语调、语速和输入设备等。

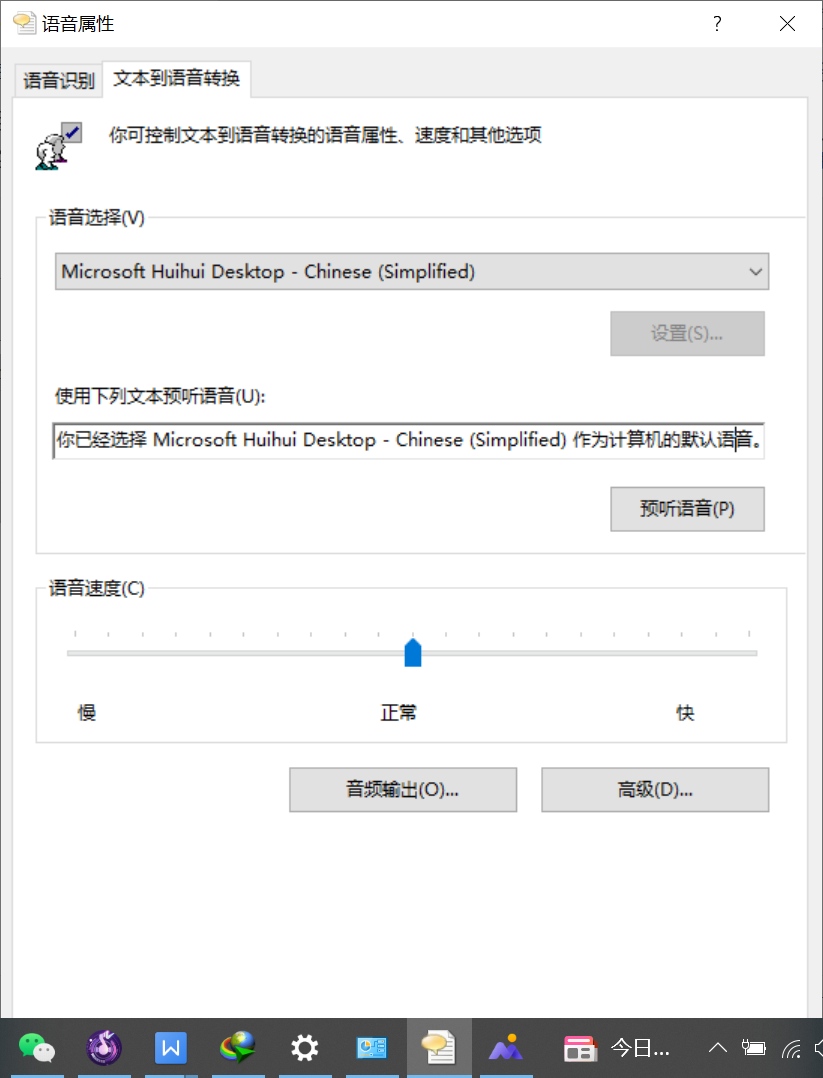
在特定网页下载了下载三个关键的程序：SpeechSDK51.exe 、 msttss22L.exe、SpeechSDK51LangPack.exe





然后安装了语音包





安装好Speech SDK后，语音控制程序将被添加到 “控制面板”->“语音”选项中。利用下图对话框可以设置语音识别和文字-语音转换的各项属性，包括语言、语调、语速和输入设备等。

（四）、实验内容和步骤

 在visual c++或.net环境下，用微软的Speech SDK实现一个具有语音提示功能的界面，要求设计交互模型，并根据实际应用情况来确定是否要用语音。

调用Speech sdk的方法简述如下：

首先在visual c++主窗口的菜单栏中选择“工程”－> “设置” 。在弹出对话框中选择“c/c++”选项卡，在分类中选择“preprocessor”,在附加包含路径中，输入speech sdk安装路径下的include目录，如：

C:\Program Files\Microsoft Speech SDK 5.1\Include

然后选择“link”选项卡，在分类中选择“Input”,在附加库路径中，输入speech sdk安装路径下的lib目录，如：

C:\Program Files\Microsoft Speech SDK 5.1\Lib\i386

程序中的关键代码如下：

#include <sapi.h>

#include <sphelper.h> //引入头文件，以调用SDK中的API

UpdateData();

ISpVoice \* pVoice = NULL;

if (FAILED(CoInitialize(NULL)))

{

AfxMessageBox("Error to intiliaze COM");

return;

}//初始化COM

HRESULT hr = CoCreateInstance(CLSID\_SpVoice, NULL, CLSCTX\_ALL, IID\_ISpVoice, (void \*\*)&pVoice);

if( SUCCEEDED( hr ) )

{

hr = pVoice->SetVolume(80);//音量控制0～100

hr = pVoice->SetRate(0);//语速控制-5～5

hr = pVoice->Speak(L“hello world”, SPF\_DEFAULT, NULL);//语音的内容

ISpObjectToken \* spToken = NULL;

 if (SUCCEEDED(SpFindBestToken(SPCAT\_VOICES, L"language=804", NULL, &SpToken)))

{   pVoice->SetVoice(SpToken);//切换至中文语音模式，language=804代表中文，409代表英文

  pVoice->Speak(L"世界你好", SPF\_DEFAULT, NULL);

   SpToken->Release();

}

if (SUCCEEDED(SpFindBestToken(SPCAT\_VOICES, L"language=409", L"Name=Microsoft Mike", &SpToken)))//说话人控制

{

  pVoice->SetVoice(SpToken);

pVoice->Speak(L"This is Mike speaking", SPF\_DEFAULT, NULL);

  SpToken->Release();

}

 pVoice->Speak(L"<LANG LANGID='804'>你好</LANG> ", SPF\_IS\_XML, NULL);//也可直接写成xml标签的格式

pVoice->Speak(L"This sounds normal <pitch middle = '-10'/> but the pitch drops half way through", SPF\_IS\_XML, NULL ); //声调控制，-代表低音，+代表高音

pVoice->Release();

pVoice = NULL;

}

CoUninitialize();

实验步骤：

1．熟悉Text-To-Speech（TTS）基本函数的调用方法，编程实现最简单的语音合成原型系统。

2．体会交互过程和实现方法。

3．设计一个完整的人机交互界面，在适当的环节嵌入语音合成模块，实现语音提示功能。

（五）、实验结果展示

1.展示界面设计，给出演示结果。

源代码：

import json

import sys

import tkinter as tk

from tkinter import ttk

from urllib.request import urlopen, Request

from urllib.error import URLError

from urllib.parse import urlencode, quote\_plus

import pygame

from io import BytesIO

from translate import Translator

from PIL import Image, ImageTk

# 百度云 API 相关信息

API\_KEY = 'bCV1HZdfiOeUIUFhfqaNhfGL'

SECRET\_KEY = 'P2XulgdcP605UPqnW5K1EjgvxMFTudX2'

TOKEN\_URL = 'http://aip.baidubce.com/oauth/2.0/token'

SCOPE = 'audio\_tts\_post'

FORMATS = {3: "mp3", 4: "pcm", 5: "pcm", 6: "wav"}

TTS\_URL = 'http://tsn.baidu.com/text2audio'

# 是否为 Python 3

IS\_PY3 = sys.version\_info.major == 3

# Definition of DemoError class

class DemoError(Exception):

    pass

# Initialize pygame mixer for audio playback

pygame.mixer.init()

class TranslationApp:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("智能翻译")

        # 设置窗口背景颜色

        self.root.configure(bg="#ADD870")

        self.create\_widgets()

    def create\_widgets(self):

        # 在 Canvas 上添加其他组件

        self.text\_var = tk.StringVar()

        self.text\_entry = tk.Entry(self.root, textvariable=self.text\_var, width=50)

        self.text\_entry.grid(row=0, column=0, padx=10, pady=50)

        self.translate\_button = tk.Button(self.root, text="翻译", command=self.translate\_text,bg="#1E90FF", fg="white")

        self.translate\_button.grid(row=0, column=1, padx=10, pady=10)

        # 图片按钮 - 输入框旁边

        self.image\_path\_input = "img/laba.jpg"  # 替换为你的图片路径

        self.laba\_img\_input = Image.open(self.image\_path\_input)

        self.laba\_img\_input = ImageTk.PhotoImage(self.laba\_img\_input.resize((20, 20), Image.LANCZOS))

        self.laba\_button\_input = tk.Button(self.root, image=self.laba\_img\_input, text="", compound=tk.LEFT, command=self.speak\_input\_text,bg="#FFFACD", fg="white")

        self.laba\_button\_input.grid(row=0, column=2, padx=20, pady=20)

        self.from\_lang\_label = tk.Label(self.root, text="源语言:")

        self.from\_lang\_label.grid(row=1, column=0, padx=20, pady=20)

        self.from\_lang\_combobox = ttk.Combobox(self.root, values=["auto", "zh", "en"], state="readonly")

        self.from\_lang\_combobox.set("auto")

        self.from\_lang\_combobox.grid(row=1, column=1, padx=20, pady=20)

        self.to\_lang\_label = tk.Label(self.root, text="目标语言:")

        self.to\_lang\_label.grid(row=2, column=0, padx=20, pady=20)

        self.to\_lang\_combobox = ttk.Combobox(self.root, values=["zh", "en"], state="readonly")

        self.to\_lang\_combobox.set("en")

        self.to\_lang\_combobox.grid(row=2, column=1, padx=20, pady=20)

        self.translation\_label = tk.Label(self.root, text="翻译结果:")

        self.translation\_label.grid(row=3, column=0, padx=20, pady=20)

        self.translation\_text = tk.Text(self.root, height=5, width=40)

        self.translation\_text.grid(row=3, column=1, padx=20, pady=20)

        # 图片按钮 - 输出框旁边

        self.image\_path\_output = "img/laba.jpg"  # 替换为你的图片路径

        self.laba\_img\_output = Image.open(self.image\_path\_output)

        self.laba\_img\_output = ImageTk.PhotoImage(self.laba\_img\_output.resize((20, 20), Image.LANCZOS))

        self.laba\_button\_output = tk.Button(self.root, image=self.laba\_img\_output, text="", compound=tk.LEFT, command=self.speak\_output\_text,bg="#FFFACD", fg="white")

        self.laba\_button\_output.grid(row=3, column=2, padx=5, pady=10)

    def translate\_text(self):

        text\_to\_translate = self.text\_var.get()

        if not text\_to\_translate:

            self.translation\_text.delete(1.0, tk.END)

            self.translation\_text.insert(tk.END, "请输入要翻译的文本")

            return

        # 获取源语言

        from\_lang = self.from\_lang\_combobox.get()

        # 获取目标语言

        target\_lang = self.to\_lang\_combobox.get()

        # 进行翻译

        translated\_text = self.translate\_text\_with\_translate(text\_to\_translate, from\_lang, target\_lang)

        if translated\_text:

            self.translation\_text.delete(1.0, tk.END)

            self.translation\_text.insert(tk.END, translated\_text)

        else:

            self.translation\_text.delete(1.0, tk.END)

            self.translation\_text.insert(tk.END, "翻译失败")

    def translate\_text\_with\_translate(self, text, source\_language, target\_language):

        translator = Translator(to\_lang=target\_language, from\_lang=source\_language)

        try:

            translation = translator.translate(text)

            return translation

        except Exception as e:

            print("翻译出现错误:", e)

            return None

    def speak\_input\_text(self):

        text\_to\_speak = self.text\_var.get()

        self.text\_to\_speech(text\_to\_speak)

    def speak\_output\_text(self):

        text\_to\_speak = self.translation\_text.get(1.0, tk.END)

        self.text\_to\_speech(text\_to\_speak)

    def text\_to\_speech(self, text):

        # 调用百度云文字转语音 API

        token = self.fetch\_token()

        tex = quote\_plus(text)

        params = {'tok': token, 'tex': tex, 'per': 1, 'spd': 5, 'pit': 5, 'vol': 5, 'aue': 3, 'cuid': "123456PYTHON", 'lan': 'zh', 'ctp': 1}

        data = urlencode(params)

        req = Request(TTS\_URL, data.encode('utf-8'))

        has\_error = False

        try:

            f = urlopen(req)

            result\_str = f.read()

            headers = dict((name.lower(), value) for name, value in f.headers.items())

            has\_error = ('content-type' not in headers.keys() or headers['content-type'].find('audio/') < 0)

        except URLError as err:

            print('tts http response http code : ' + str(err.code))

            result\_str = err.read()

            has\_error = True

        if has\_error:

            print("tts api  error")

        else:

            self.play\_audio(result\_str)

    def fetch\_token(self):

        print("获取token开始")

        params = {'grant\_type': 'client\_credentials',

                  'client\_id': API\_KEY,

                  'client\_secret': SECRET\_KEY}

        post\_data = urlencode(params)

        if IS\_PY3:

            post\_data = post\_data.encode('utf-8')

        req = Request(TOKEN\_URL, post\_data)

        try:

            f = urlopen(req, timeout=5)

            result\_str = f.read()

        except URLError as err:

            print('token http response http code : ' + str(err.code))

            result\_str = err.read()

        if IS\_PY3:

            result\_str = result\_str.decode()

        result = json.loads(result\_str)

        if 'access\_token' in result.keys() and 'scope' in result.keys():

            if not SCOPE in result['scope'].split(' '):

                raise DemoError('scope不正确')

            print('成功获取token: %s ; 过期时间（秒）: %s' % (result['access\_token'], result['expires\_in']))

            return result['access\_token']

        else:

            raise DemoError('API\_KEY或SECRET\_KEY可能不正确: token响应中找不到access\_token或scope')

    def play\_audio(self, audio\_data):

        audio\_stream = BytesIO(audio\_data)

        pygame.mixer.music.load(audio\_stream)

        pygame.mixer.music.play()

        while pygame.mixer.music.get\_busy():

            continue

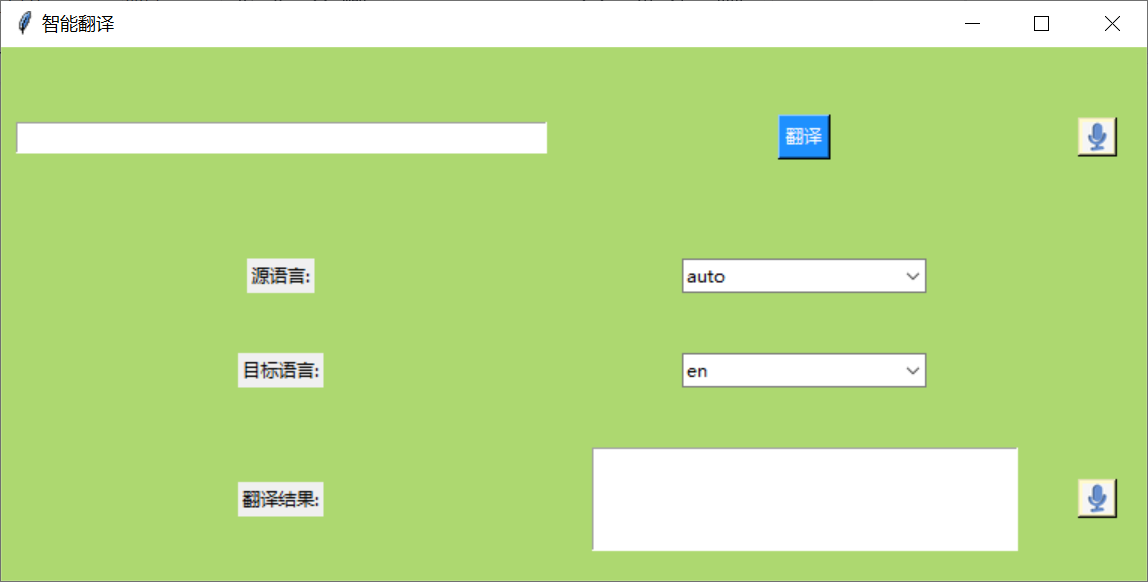
if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

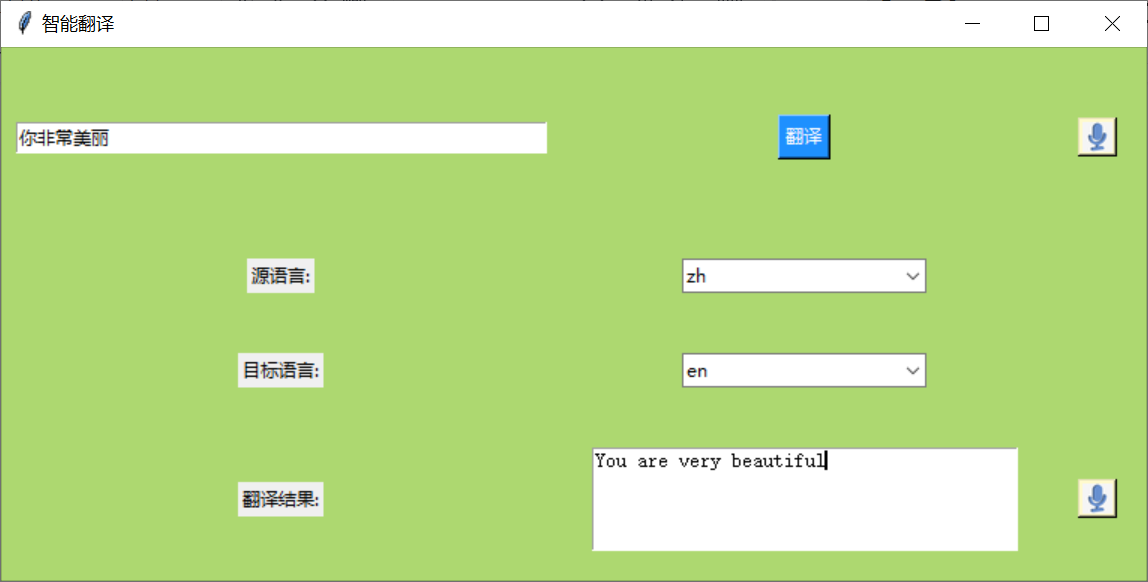
    app = TranslationApp(root)

    root.mainloop()

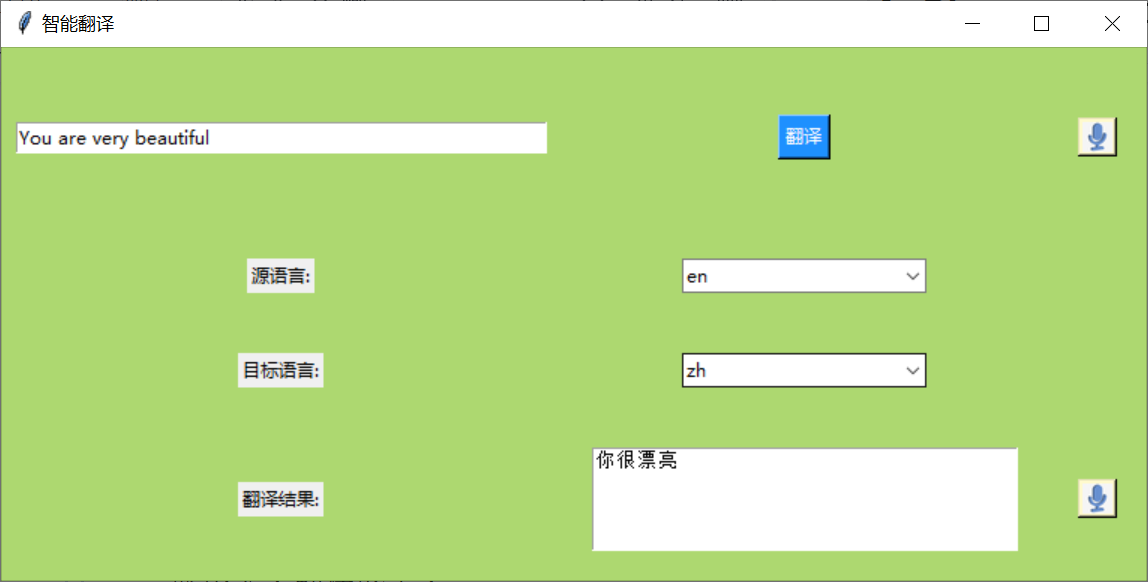
界面设计如下：



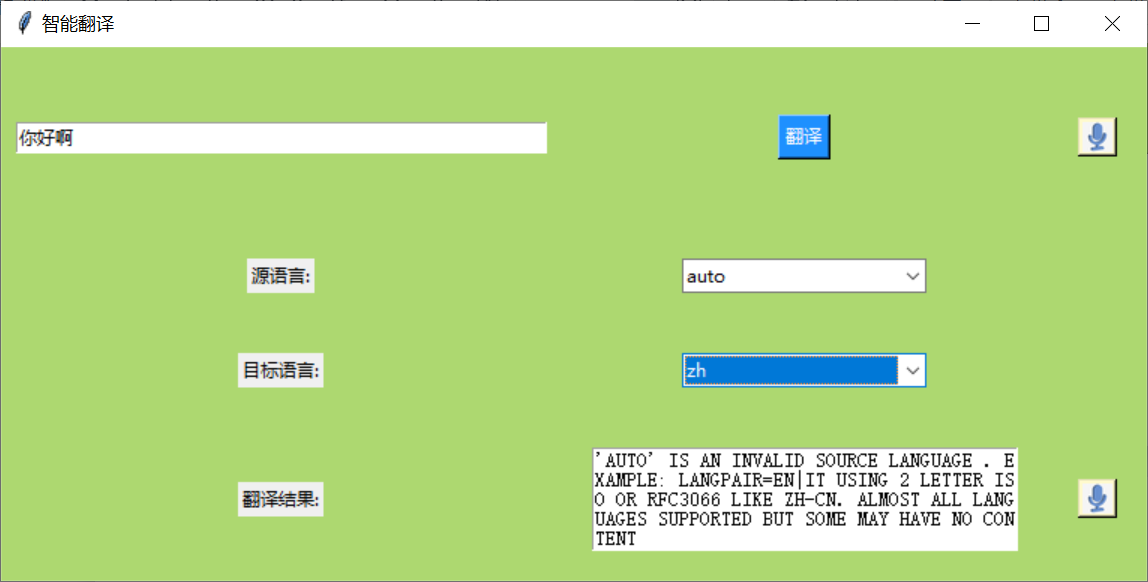
展示中文翻译成英文：



展示英文翻译成中文：



展示翻译出错：



小结实验心得体会：

利用了百度云翻译打造的翻译小程序，虽然在一定程度上能够大差不差地翻译一些单词词汇，但是我发现当涉及到更复杂的句子和文本时，百度云翻译的准确性有时会有所缺陷。虽然它能够处理一些复杂的语法结构，但在涉及到上下文和语义的理解方面，它可能会出现一些困惑。这导致翻译结果可能有时会不够准确或不通顺。另一个我观察到的问题是百度翻译在涉及专业术语和文化背景的翻译时的不足。当翻译特定行业的术语或包含文化特色的内容时，百度翻译往往无法提供准确的翻译。这可能与其缺乏相关语料库和专业领域知识有关。我认为此翻译小程序在用户交互方面还有一些改进的空间。此外，它也可以提供更多的选项和自定义功能，以满足用户的个性化需求。