Proyecto Final DAM
Aplicación que habla y recibe comandos voz
Python 3.6

Alumno: Óscar Úbeda Halcón
Professor: Gonzalo Blanca Bonilla

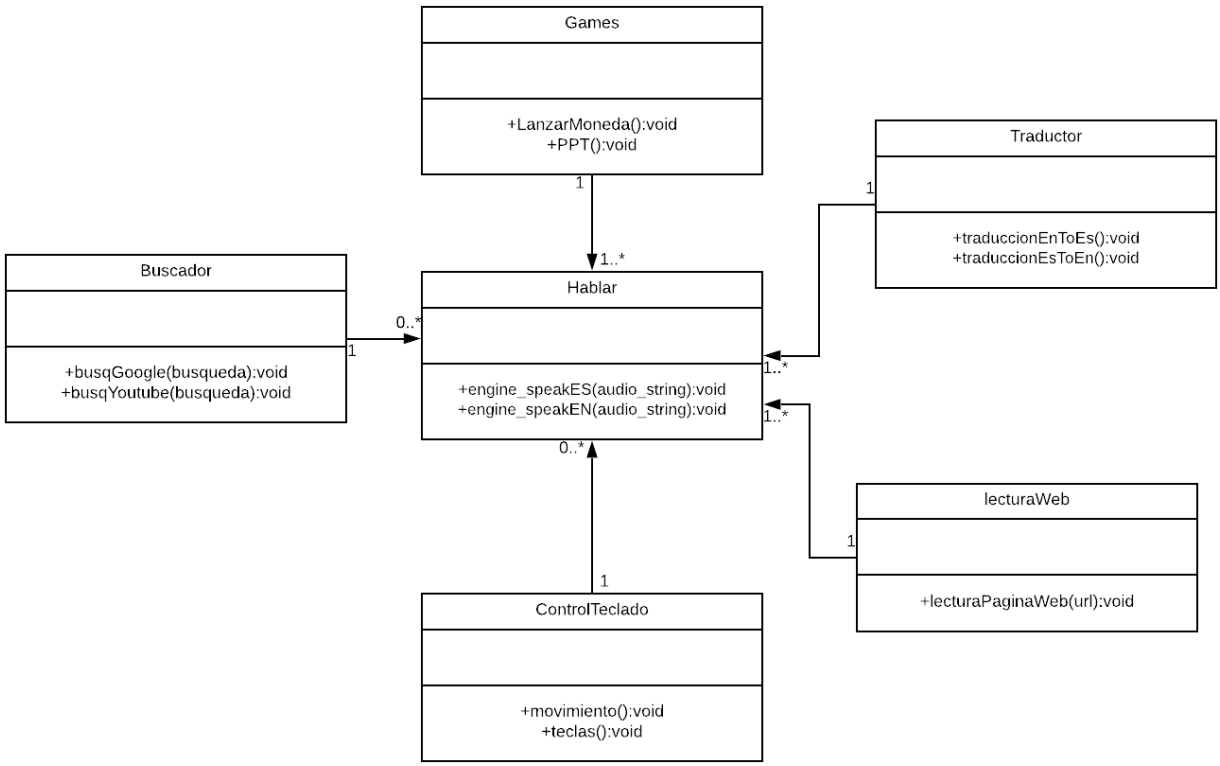
Indice

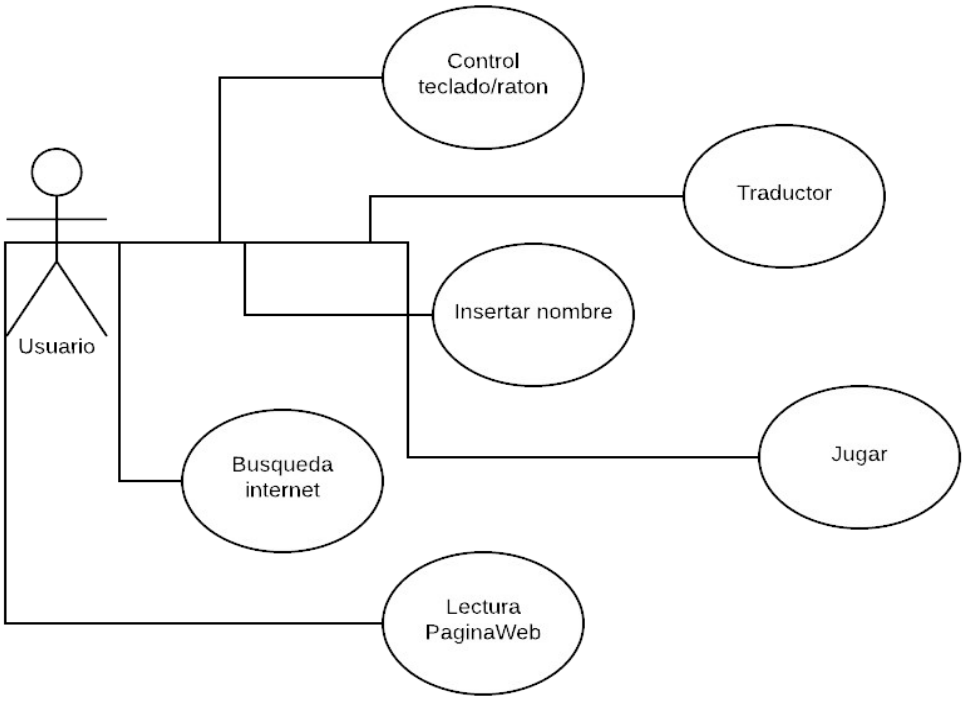
Resumen proyecto

El proyecto que yo propuse era una aplicacion en la que tu le podias dar ordenes de voz de las cuales el tenia predefinidas y tambien que el programa te pueda hablar unas frases. Todo esto programado en python y con ayuda de librerias de reconocimiento de voz en python que comentare mas adelante. Despues de esta base inicial he implementado unas clases:

1. Clase para traducir entre ingles y español que tu le dices y la traduccion te la dice por voz.
2. Clase para hacer busquedas en google y youtube con uso de palabra sobre para separar la busqueda entre la frase dicha por voz.
3. Clase de leector pagina web la que le pasas una pagina web de parametro y ella te la lee en voz alta.
4. Clase con un par de juegos de piedra papel y tijera, cara o cruz.
5. Clase de control de teclado y raton por voz.

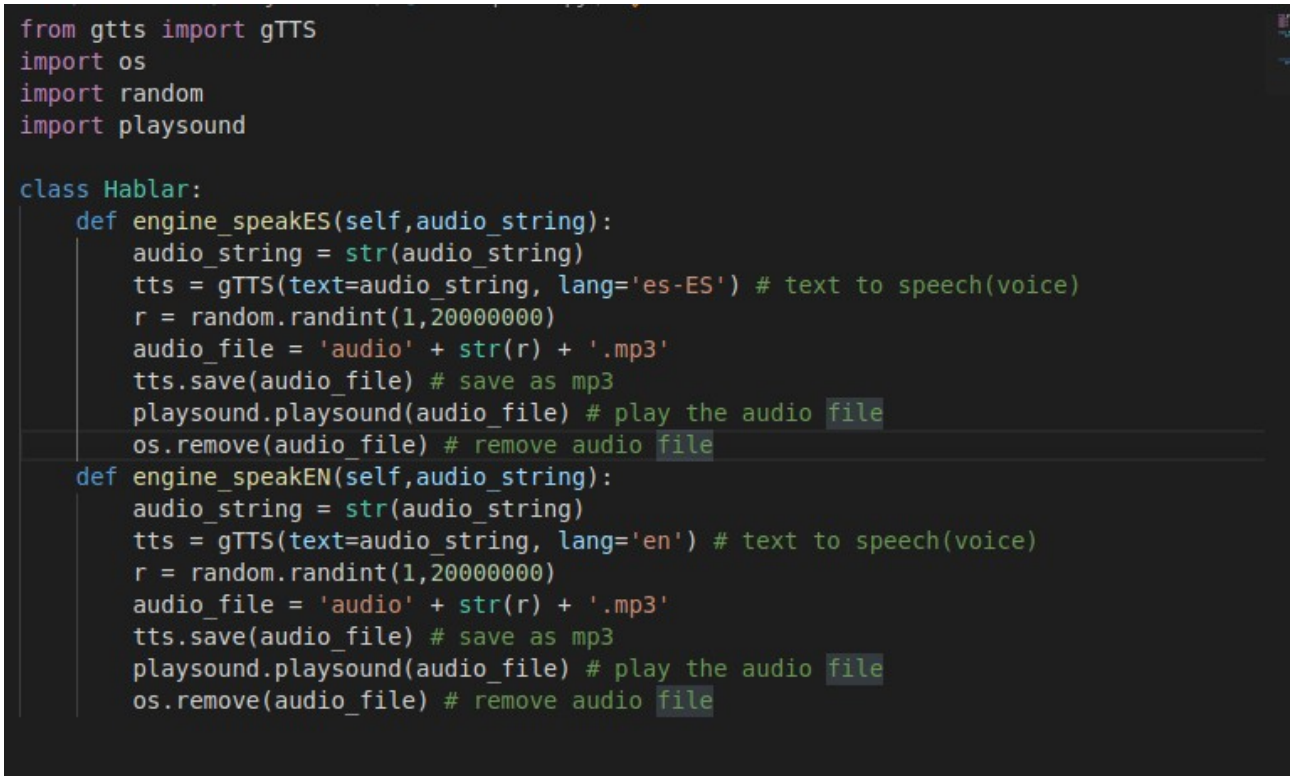
UML Estatico y Dinamico




Control
teclado/raton
Traductor
Usuario
Insertar nombre
Busqueda
internet
Jugar
Lectura
PaginaWeb

Explicacion de classes y librerias

Clase Hablar

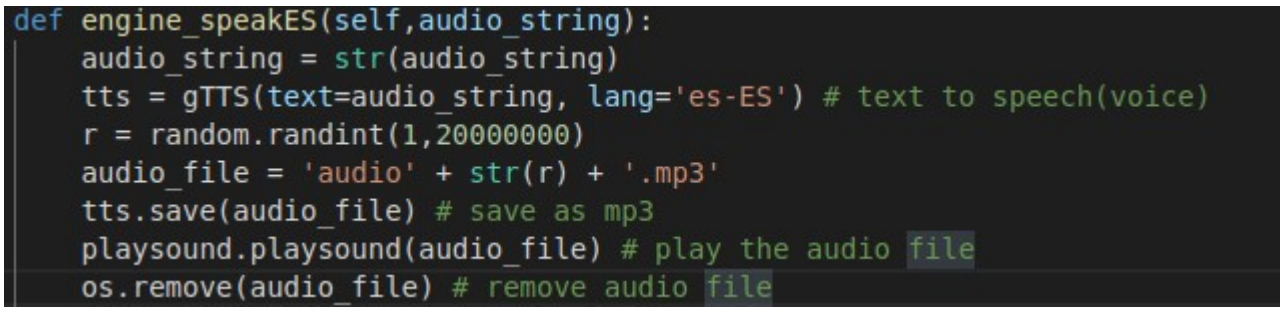


```python
from gtts import gTTS
import os
import random
import playsound

class Hablar:
    def engine_speakES(self,audio_string):
        audio_string = str(audio_string)
        tts = gTTS(text=audio_string, lang='es-ES') # text to speech(voice)
        r = random.randint(1,20000000)
        audio_file = 'audio' + str(r) + '.mp3'
        tts.save(audio_file) # save as mp3
        playsound.playsound(audio_file) # play the audio file
        os.remove(audio_file) # remove audio file
    def engine_speakEN(self,audio_string):
        audio_string = str(audio_string)
        tts = gTTS(text=audio_string, lang='en') # text to speech(voice)
        r = random.randint(1,20000000)
        audio_file = 'audio' + str(r) + '.mp3'
        tts.save(audio_file) # save as mp3
        playsound.playsound(audio_file) # play the audio file
        os.remove(audio_file) # remove audio file
```

Empezaremos por la clase mas importante despues del main y esa es clase Hablar en esta clase usamos la libreria gtts una libreria para guardar strings en una variable con la voz del texto y en el idioma especificado y en el sistema con la ayuuda de random . Tambien usamos la libreria os para poder eliminar esos audios. Por ultimo la libreria playsound para poder oir el audio de voz de la frase.

Explicacion metodo



```python
def engine_speakES(self,audio_string):
    audio_string = str(audio_string)
    tts = gTTS(text=audio_string, lang='es-ES') # text to speech(voice)
    r = random.randint(1,20000000)
    audio_file = 'audio' + str(r) + '.mp3'
    tts.save(audio_file) # save as mp3
    playsound.playsound(audio_file) # play the audio file
    os.remove(audio_file) # remove audio file
```

Por empezara en el metodo tenemos un audio_string que es la frase que queremos que diga la maquina y esta se le “pasa” a string(este apartado se usa mas como verificador de string para que no pete el programa) despues de tener ese string usamos el gtts para tener la variable de audio al haber

pasado el texto a voz lo siguiente es guardar ese audio para eso uso un numero random y con este creo un nombre para guardar el audio. Al tener el nombre usamos el metodo save del gtts y con esto lo tenemos guardado en el sistema lo siguiente es usar el playsound para reproducir el audio y finalmente usar el os para eliminarlo de esta forma no se nos llena el sistema de audios.

Clase Traductor

```python
from claseSpeaker import Hablar
from translate import Translator
import speech_recognition as sr

class Traductor:
    def traduccionEnToEs(self):
        mic = sr.Microphone()
        r = sr.Recognizer()
        habla=Hablar()
        translator= Translator(to_lang="es-ES",from_lang="en")
        while True:
            with mic as source:
                r.adjust_for_ambient_noise(source)
                audio = r.listen(source)
            a = r.recognize_google(audio, language='en')
            print(a)
            habla.engine_speakES(translator.translate(a))
            break

    def traduccionEsToEn(self):
        mic = sr.Microphone()
        r = sr.Recognizer()
        habla=Hablar()
        translator= Translator(to_lang="en",from_lang="es-ES")
        while True:
            with mic as source:
                r.adjust_for_ambient_noise(source)
                audio = r.listen(source)
            a = r.recognize_google(audio, language='es-ES')
            print(a)
            habla.engine_speakEN(translator.translate(a))
            break
```

Seguimos con la clase traductor que usa libreria speech_recognition que permite usar el microfono para poder reconocer la voz. Tambien usa la libreria translate para poder traduccir lo que nos reconoce y por ultimo la clase Hablar para que nos diga la traduccion.

Explicacion metodo

```
def traduccionEnToEs(self):
    mic = sr.Microphone()
    r = sr.Recognizer()
    habla=Hablar()
    translator= Translator(to_lang="es-ES",from_lang="en")
    while True:
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        a = r.recognize_google(audio, language='en')
        print(a)
        habla.engine_speakES(translator.translate(a))
        break
```

Para empezar este metodo no tiene parametros de entrada. Al llamar al metodo instanciamos una variable de microfono, una de reconocedor de voz, un objeto hablar y uno de traductor en el cual le especificamos que idioma es el que lee y traduce en este caso viene del ingles y traducir al castellano. Despues hacemos un bucle infinito para que escuche todo el rato mas adelante tenemos el mic que inicia el identificador de voz con el primer paso un regulador para que no detecte sonido ambiente como el viento y el segundo es el asignador de la voz cuando este no oye ningun sonido en unos segundos deja de oir y pasa al siguiente paso en este usamos un metodo para pasar la voz a texto con el metodo recognize_google que viene del speech_recognition en este hay que especificarle que audio tiene que convertir y idioma en el que esta. Luego tenemos un print para poder visualizar que todo ha ido correcto y finalmente llamamos al metodo de hablar y le pasamos de parametro la traduccion del texto.

Clase LecturaWeb

```python
import requests
from bs4 import BeautifulSoup
from claseSpeaker import Hablar

class lecturaWeb:
    def lecturaPaginaWeb(self,url):

        #open with GET method
        resp=requests.get(url)

        #http_respone 200 means OK status
        if resp.status_code==200:
            print("Successfully opened the web page")

            # we need a parser,Python built-in HTML parser is enough .
            soup=BeautifulSoup(resp.text,'html.parser')

            # l is the list which contains all the text i.e news
            l=soup.find("div",{"class":"ue-l-article__body ue-c-article__body"})

            p=0
            habla= Hablar()
            for i in l.findAll("p"):
                print(i.text)
                habla.engine_speakEN(i.text)
                p=1+p
                if p == 3:
                    break
        else:
            print("Error")
```

Ahora la clase LecturaWeb en esta clase usamos la libreria requests para poder hacer una conexion a la pagina web que queremos leer, luego tenemos la libreria bs4 para usar BeautifulSoup que este nos da el acceso de cambiar un html en un texto y por ultimo la clase hablar para que nos lea la informacion.

Explicacion metodo

```
def lecturaPaginaWeb(self,url):

    #open with GET method
    resp=requests.get(url)

    #http_respone 200 means OK status
    if resp.status_code==200:
        print("Successfully opened the web page")

        # we need a parser,Python built-in HTML parser is enough .
        soup=BeautifulSoup(resp.text,'html.parser')

        # l is the list which contains all the text i.e news
        l=soup.find("div",{"class":"ue-l-article__body ue-c-article__body"})

        p=0
        habla= Hablar()
        for i in l.findAll("p"):
            print(i.text)
            habla.engine_speakEN(i.text)
            p=1+p
            if p == 3:
                break
    else:
        print("Error")
```

En este metodo se le pasa una url de la pagina que queremos leer. Luego hacemos una conexion con requests a la paginna web si todo ha funcionado se continua en el caso que no se termina el metodo con un print error. Al conseguir conexion a la pagina web instanciamos el soup con cambio del html en un texto del propio objeto al tener este cambio podemos guardar en una variable el indice del apartado div con el atributo “class: xxxx”. Al tener guardado este podemos ya recorrer los paragrafos en los que contiene el texto que queremos leer con un for. **AVISO** yo he usado una variable int para que solo me diga los 3 primeros paragrafos para que no recorrer todo esto solo hay que quitar el if y recorreria todos los paragrafos. Ademas en el apartado de conseguir el indice del apartado div el nombre del atributo clase difiere en cada pagina web quiero decir el nombre de una pagina web no es el mismo que el de las otras.

Clase ControlTeclado

```
import pyautogui
import speech_recognition as sr
class ControlTeclado:
    def movimiento(self):
        r = sr.Recognizer()
        mic = sr.Microphone()
        while True:
            with mic as source:
                r.adjust_for_ambient_noise(source)
                audio = r.listen(source)
            a = r.recognize_google(audio, language='es-ES')
            print(a)
            if "diagonal" in a:
                if "derecha" in a:
                    if "arriba" in a:
                        if "poquito" in a:
                            x,y=pyautogui.position()
                            pyautogui.moveTo(x+100, y-10)
                        else:
                            x,y=pyautogui.position()
                            pyautogui.moveTo(x+100, y-100)
                    if "abajo" in a:
                        if "poquito" in a:
                            x,y=pyautogui.position()
                            pyautogui.moveTo(x+100, y+100)
                        else:
                            x,y=pyautogui.position()
                            pyautogui.moveTo(x+1000, y+1000)
```

```
if "izquierda" in a:
    if "arriba" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-10, y-10)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-100, y-100)
    if "abajo" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-10, y+10)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-100, y+100)
```

```
else:
    if "arriba" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x, y-10)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x, y-100)
    if "abajo" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x, y+10)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x, y+100)
    if "derecha" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x+10, y)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x+100, y)
    if "izquierda" in a:
        if "poquito" in a:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-10, y)
        else:
            x,y=pyautogui.position()
            pyautogui.moveTo(x-100, y)
if "termina" in a:
    break
```

```
def teclas(self):
    r = sr.Recognizer()
    mic = sr.Microphone()
    while True:
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        a = r.recognize_google(audio, language='es-ES')
        print(a)
        if "manten" in a:
            if "adelante" in a:
                for i in range (0,5):
                    pyautogui.press("w")
            if "atras" in a:
                for i in range (0,5):
                    pyautogui.press("s")
            if "derecha" in a:
                for i in range (0,5):
                    pyautogui.press("d")
            if "izquierda" in a:
                for i in range (0,5):
                    pyautogui.press("a")
```

```
            else:
                if "adelante" in a:
                    pyautogui.press("w")
                if "atras" in a:
                    pyautogui.press("s")
                if "derecha" in a:
                    pyautogui.press("d")
                if "izquierda" in a:
                    pyautogui.press("a")
                if "termina" in a:
                    break

pyautogui.PAUSE = 1
pyautogui.FAILSAFE = True
```

En esta clase empleamos la libreria speech_recognition y pyautogui. Esta libreria lo que hace es simular movimientos del cursor y tambien teclas del teclado. **AVISO** al usar esta libreria se recomienda colocar las dos ultimas linias. La primera es para que haya una pausa de 1 segundo entre cada accion de autogui y la segunda es un modo de seguridad que consiste en posicionar el

cursor a la esquina arriba izquierda de la pantalla esto es por el caso de que si hay un problema por ejemplo de bucle poder tomar control del cursor y teclado asi no se satura el ordenador en el caso de demasiados clicks.

Explicacion metodo movimiento

```
def movimiento(self):
    r = sr.Recognizer()
    mic = sr.Microphone()
    while True:
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        a = r.recognize_google(audio, language='es-ES')
        print(a)
        if "diagonal" in a:
            if "derecha" in a:
                if "arriba" in a:
                    if "poquito" in a:
                        x,y=pyautogui.position()
                        pyautogui.moveTo(x+100, y-10)
                    else:
                        x,y=pyautogui.position()
                        pyautogui.moveTo(x+100, y-100)
                if "abajo" in a:
                    if "poquito" in a:
                        x,y=pyautogui.position()
                        pyautogui.moveTo(x+100, y+100)
                    else:
                        x,y=pyautogui.position()
                        pyautogui.moveTo(x+1000, y+1000)
```

En este metodo primero cojemos el reconocedor y el microfono despues de eso hacemos que escuche nuestros comandos para poder diferenciar el movimiento de raton que queremos. Despues de eso nos vamos a los if estos se dividen entre diagonal o else(caso que no se diga diagonal) dentro de diagonal esta el caso de derecha o izquierda y despues se elije la diagonal de derecha arriba o la de abajo en la parte de la izquierda igual. En el apartado else solo tiene apartados arriba, abajo, derecha y izquierda. A partir de aqui todos ellos tienen el apartado poquito es algo implementado para que el cursor no se pase por alto algun lugar que queramos tenerlo. Dentro de estos condicionales tenemos primero la asignacion de la posicion actual de nuestro cursor al obtenerla usamos el metodo moveTo que nos mueve a una posicion de la pantalla en este caso le ponemos la posicion del cursor con las modificaciones necesarias para hacer el movimiento. Por ultimo tenemos un condicional de si dices pulsar se llama un metodo de click para hacer un click de raton. **AVISO** mas que un aviso es una aclaracion las y empiezan desde la esquina izquierda arriba de la pantalla no existen y negativas tampoco con las x.

Explicacion metodo teclas

```
def teclas(self):
    r = sr.Recognizer()
    mic = sr.Microphone()
    while True:
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        a = r.recognize_google(audio, language='es-ES')
        print(a)
        if "manten" in a:
            if "adelante" in a:
                for i in range (0,5):
                    pyautogui.press("w")
            if "atras" in a:
                for i in range (0,5):
                    pyautogui.press("s")
            if "derecha" in a:
                for i in range (0,5):
                    pyautogui.press("d")
            if "izquierda" in a:
                for i in range (0,5):
                    pyautogui.press("a")
```

En este metodo se simula un uso del tipico movimiento en videojuegos de las teclas wasd. En este caso tenemos apartado manten o else en el de manten se usa un for para repetir el pulsar la tecla y asi dar la simulacion de estar manteniendo dicha tecla para simular una tecla se usa el metodo press y le pasamos por parametro la tecla que queremos las teclas disponibles estan definidas en el metodo. En el caso de else la unica diferencia es no usar un bucle.

Clase Games

```
from claseSpeaker import Hablar
import random
import speech_recognition as sr
class Games:
    def LanzarMoneda(self):
        habla=Hablar()
        moves=["cara", "cruz"]
        cmove=random.choice(moves)
        habla.engine_speakES("Ha salido " + cmove)
    def PPT(self):
        r = sr.Recognizer()
        mic = sr.Microphone()
        habla=Hablar()
        habla.engine_speakES("Escoje entre piedra, papel o tijeras")
        while True:
            with mic as source:
                r.adjust_for_ambient_noise(source)
                audio = r.listen(source)
            q = r.recognize_google(audio, language='es-ES')
            pmove=q
            break
        moves=["piedra", "papel", "tijeras"]

        cmove=random.choice(moves)
```

```
habla.engine_speakES("El ordenador ha sacado " + cmove)
habla.engine_speakES("Tu has sacado " + pmove)
if pmove==cmove:
    habla.engine_speakES("Empate")
elif pmove== "piedra" and cmove== "tijeras":
    habla.engine_speakES("Jugador gana")
elif pmove== "piedra" and cmove== "papel":
    habla.engine_speakES("Ordenador gana")
elif pmove== "papel" and cmove== "piedra":
    habla.engine_speakES("Jugador gana")
elif pmove== "papel" and cmove== "tijeras":
    habla.engine_speakES("Ordenador gana")
elif pmove== "tijeras" and cmove== "papel":
    habla.engine_speakES("Jugador gana")
elif pmove== "tijeras" and cmove== "piedra":
    habla.engine_speakES("Ordenador gana")
```

En esta clase se usa los commandos de voz para elejir opciones y el programa luego nos dice por voz los resultados.

Clase Buscador

```
import webbrowser
class Buscador:
    def busqGoogle(self,busqueda):
        url = "https://google.com/search?q=" + busqueda
        webbrowser.get().open(url)
    def busqYoutube(self,busqueda):
        url = "https://www.youtube.com/results?search_query=" + busqueda
        webbrowser.get().open(url)
```

Aqui usamos la libreria webbrowser para poder abrir el navegador predeterminado con una url pasada como parametro. Esta clase contiene dos metodos que la unica diferencia es la url. En ambos metodos entra como parametro busqueda que queremos hacer puedes ser tanto palabra como frase.

“Clase Main”

```
import speech_recognition as sr
from claseSpeaker import Hablar
from claseTraductor import Traductor
from claseControlTecl import ControlTeclado
from claseLecWeb import lecturaWeb
from claseBuscador import Buscador
from claseGames import Games

r = sr.Recognizer()

mic = sr.Microphone()
habla=Hablar()
traducir=Traductor()
controlador=ControlTeclado()
leectorWeb=lecturaWeb()
busqueda=Buscador()
juegos=Games()
```

```
while True:
    print("Puedes Hablar")
    while True:
        habla.engine_speakES("Hola, como quieres que me llame?")
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        a = r.recognize_google(audio, language='es-ES')
        nombre=a
        habla.engine_speakES("Entendido me llamo "+nombre)
        print(a)
        break
    with mic as source:
        r.adjust_for_ambient_noise(source)
        audio = r.listen(source)
    c = r.recognize_google(audio, language='es-ES')
```

```
if nombre in c:
    while True:
        print("En que puedo ayudarte? ")
        habla.engine_speakES("En que puedo ayudarte? ")
        with mic as source:
            r.adjust_for_ambient_noise(source)
            audio = r.listen(source)
        b = r.recognize_google(audio)
        print(b)
        if "cómo" in b and "me llamo" in b:
            habla.engine_speakES("No lo se. Como te llamas?")
            while True:
                with mic as source:
                    r.adjust_for_ambient_noise(source)
                    audio = r.listen(source)
                q = r.recognize_google(audio, language='es-ES')
                nombreCre=q
            habla.engine_speakES("Te llamas "+nombreCre)
```

```
    if "búscame" in b:
        if "youtube" in b:
            #usar sobre al preguntar para poder saber que parte escojer
            search_term = b.split("sobre")[-1]
            busqueda.busqYoutube(search_term)
            habla.engine_speakES("Esto es lo que he podido encontrar")
        else:
            search_term = b.split("sobre")[-1]
            busqueda.busqGoogle(search_term)
            habla.engine_speakES("Esto es lo que he podido encontrar")
    if "léeme" in b and "web" in b:
        leectorWeb.lecturaPaginaWeb("https://www.elmundo.es/economia/2020/05/15/
    if "Toma" in b and "control" in b:
        controlador.movimiento()
    if "traduce" in b and "español" in b:
        traducir.traduccionEsToEn()
    if "traduce" in b and "ingles" in b:
        traducir.traduccionEnToEs()
```

```
            if "Juguemos" in b:
                while True:
                    habla.engine_speakES("Que quieres jugar?")
                    with mic as source:
                        r.adjust_for_ambient_noise(source)
                        audio = r.listen(source)
                    o = r.recognize_google(audio, language='es-ES')
                    if "moneda" in o:
                        juegos.LanzarMoneda()
                    if "piedra" in o or "papel" in o or "tijera" in o:
                        juegos.PPT()
                    if "nada" in o:
                        break
            if "apágate" in b:
                print("Finalizar reconocimiento")
                habla.engine_speakES("Finalizar reconocimiento")
                break
        break
```

En el main empezamos por importar todas las clases y llamandolas. Lo siguiente es entrar en un bucle y que nos pregunte por el nombre que queremos referirlo para mas tarde al decir su nombre podremos empezar a usar las opciones como con el alexa de amazon. Mas adelante al decir su nombre tenemos las opciones de las clases creadas y una opcion de guardar tu nombre. Lo que se puede destacar en esta parte es la de buscador antes de llamarlo al tener que darle el termino de busqueda lo que hacemos es cojer lo que dice y hacer una separacion a partir de la palabra sobre asi todo lo que este detras de sobre lo pasara al metodo de buscador.

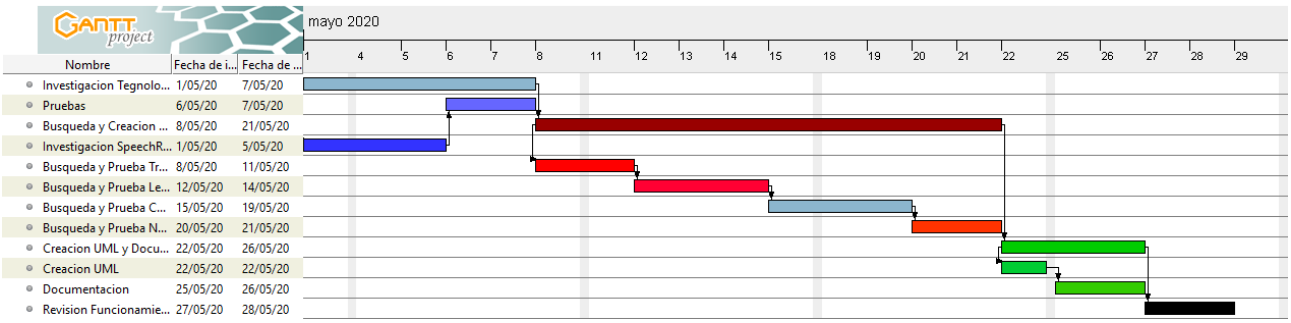
Dessarollo de proyecto

El dessarollo del proyecto en este mes se podria categorizar en 3 apartados en los que cada uno de ellos serian de una semana de duracion menos el apartado 2.

Fase1. Primera Semana: Al empezar buscar informacion sobre como recojer tu voz y ejemplos de programas. En este apartado me centre en 2 programas que usaban la libreria el cual escoji los 2 fue porque uno era en ingles mientras el otro español asi podia contrastar los dos para poder hacer uno en español pero tambien con apartado del programa en ingles. Al tener estos programas empeze por probar si podia recojer mi voz y actuar acorde con lo que le decia. Al conseguir que funcione luego fue el paso de que la maquina nos hable en este me mire distintas formas de hacerlo y tambien compare con el de los programas anteriores, despues de escojerme por uno fue probarlo. Terminando con las pruebas ya solo quedaba viernes en este dia lo que hice es pensar en ideas de usos que se puede dar al programa que se implementa en la suigiente fase.

Fase2. Segunda y Tercera Semana: En esta semana las he juntado porque son muy parecidas ambas estas semanas encompasan el tiempo de estudio de cada metodo implementado al programa como el de LecturaPaginaWeb. En este recorrido se ha seguido el paso de busqueda de informacion sobre la idea, luego programas de ejemplo para poder entender con mas claridad, despues es intentar añadir ese ejemplo al programa con los usos del habla y por ultimo el cambio completo del ejemplo a mis necesidades. Despues de hacer estos pasos con todos los apartados lo siguiente de la lista fue convertirlo en clases ya que todo lo anterior estaba organizado en funciones. Al tenerlo en clases lo ultimo que quedaba fue otra prueba de que todo funcionase correctamente.

Fase3. Ultima Semana: Para la ultima semana estaba el lunes hacer los dos UML de como han quedado las clases y el caso de uso del usuario con el programa. Martes estaria el hacer la documentacion sobre el proyecto y tambien la presentacion. Miercoles revision con el tutor de proyecto sobre documentacion y presentacion. Jueves y viernes seria repaso de la aplicacion para que no haya problemas.



Matizes a destacar

En este apartado he puesto algunas cosas que resaltar con nuestro programa.

1. El programa se usa con python 3.6 ya que el 2.7 me ha dado problemas con la codificacion utf-8 y el 3.7 no tiene soporte de SpeechRecognition.

2. Al ejecutar el programa se pueden ver unas linas de problemas sobre ALSA libreria esto son solo avisos de uso del microfono para poder quitarlos hay que hacer unos cambios en archivos de configuracion.

3. Hay que tener en cuenta que se ha de vocabulizar para que el programa pueda entenderte y el programa entiende los accentos cuando se hace una busqueda de palabra hay que ponerla con accentos para que la pueda encontrar(en esta parte hablo sobre los if dentro del programa).

4. En el caso que le hables en otro idioma el programa intentara pasarlo al idioma que deberia recojerlo en este caso castellano.

Posibilidades de mejora

Como se puede ver este programa no esta del todo desarollado quiero decir no llega al potencial que podria tener y en este apartado pondre algunos usos que se pueden añadir como classes al programa.

1. Se podria hacer una conexion por WebSockets para poder hacer un chat pero en voz el usuario habla y esto a su vez se envia por los WebSockets a un servidor y los mensajes que recibe del servidor puede luego el programa reproducirlo con la clase Habla.

2. El programa puede estar connectado algun dispositivo como una rassphery pi y con el poder hacer un control de luces o dispositivos. En este seria implementar una clase con un metodo que contenga una conexion a un programa arduino o al dispositivo en si.

3. Tambien se puede crear una classe con una conexion al calendario para asi poder poder hacer recordatorios o eventos y se guarden en el calendario. Tambien si es el dia de evento que el ordenador te lo diga por voz.

Bibliografia

Programa español con SpeechRecognition:https://www.youtube.com/watch?v=x8xjj6cR9Nc

Programa ingles con SpeechRecognition:https://www.youtube.com/watch?v=8QD6HqL9Qc0

Libreria BeautifulSoup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Libreria pyautogui: https://pypi.org/project/PyAutoGUI/ , https://pyautogui.readthedocs.io/en/latest/

Libreria Translate: https://pypi.org/project/translate/