**6. óra: mindenttudó gép**

1. 5 perces beszámoló tetszőleges tartalommal és üzenettel valamint tetszőlegesen választott retorikai eszközökkel. Nekünk kell eldönteni, hogy mi lehetett az üzenet. Figyelni kell a beszéd felépítésére (bevezető, fő rész és konklúzió) és arra, hogy a tartalom minden eleme támogassa az üzenetet, illetve, hogy használjunk retorikai eszközöket.
2. 3 perces véleményalkotás  
   - bevezető, fő rész, konklúzió  
   - 2 pozitív és egy konstruktív javaslat  
   - konklúzióban összefoglalni mi lehetett az üzenet, volt-e retorikai eszköz használva és adni még egy pozitív megjegyzést.
3. Mindenre válaszoló gép:
   1. *Bevezető*:  
      A vidámparkok gyakori automatája szokott lenni egy olyan gép, amibe ha pénzt dobunk és felteszünk neki egy kérdést (legtöbbször a jövőnkkel kapcsolatban), akkor ad rá valami választ.   
      Arra gondoltatok, hogy megviccelitek a barátaitokat és készítetek egy ilyen gépet ami képes választ adni bármilyen kérdésre. Persze, azt nem közlitek velük, hogy ennek a válasznak nem lesz semmilyen tudományosan megalapozott háttere vagy érzelmi töltete. Lényegében teljesen véletlenszerű válaszokat fog generálni, ami kérdéstől függően akár elég vicces szituációba hozhatja a barátotokat.
   2. *Előkészületek*:  
      Az áramkor nem áll sok elemből, így könnyen el tudják a diákok készíteni. Érdemes előre letárgyalni, hogyan állapítjuk meg a szervó esetében, hogy a 3 drót közül melyik a szignál és melyek mennek a tekercshez. A barna kábelt kötjük a földelésre, a pirosat a tápra és a narancssárgát a szignálra.   
      A program esetében is érdemes számukra egy fél kész kódot adni, amit csak be kell fejezniük (erre egy példát ennek a dokumentumnak a végén látunk). A *pass* sort kell helyettesíteni a megfelelő kóddal.
   3. *Részletekre bontás:*- Első lépésként meg kell határoznunk a szervó mozgástartományát, hogy meg tudjunk állapítani 3 beállítási szöget, amikor pont a 3 válaszra mutat. Ezt mindenki letesztelheti az AngularServo objektummal.  
      - Szükség lehet az *unidecode* csomag telepítésére (ha nem lenne meg), azt megtehetjük a conda install -c anaconda unidecode paranccsal.  
      - inicializáljuk a szervó és nyomógomb objektumokat.  
      - Definiáljuk a motorhoz tartozó lehetséges szög állásokat.  
      - a netről összegyűjtjük azokat a kérdőszavakat, amikre igennel, nemmel vagy talánnal lehet válaszolni.  
      - definiálunk egy függvényt a programból való kilépésre  
      - definiálunk egy függvényt ami leellenőrzi, hogy megfelelő kérdést tettek-e fel, azaz, a kérdésben megtalalálható-e a fent kigyűjtött kérdőszavak egyike?  
      - definiálunk egy függvényt, ami válaszol a kérdésre, azaz a motort elforgatja egy random pozícióba, az előre megadott lehetőségek közül.  
      - *while* ciklusban feltesszük a kérdést, leellenőrízzük, hogy megfelel-e, ha igen, akkor váalaszolunk, ha nem akkor kiírjuk, hogy milyen kérdést szeretnénk.
   4. *Elektronika:*  
      - Kell egy szervó motor és 5V-os táp (akár power bank)  
      - Kell egy nyomógomb.  
      - Kössük be a motort és a nyomógombot a leírásnak megfelelően.
   5. *Kód:*  
      - Elkészíteni az áramkört.  
      - Inicializálni a Servo, AngularServo és nyomógomb objektumokat.  
      - Letesztelni a szervo mozgástartományát szögekben, és avval inicializálni az AngularServo-t.  
      - Definiálni az igen, nem és talán válaszokhoz tartozó szögeket és egy listában tárolni.  
      - Definiálni egy listában azokat a kérdőszavakat, amik nem felelnek nekünk meg, hogy kiszűrjük őket.  
      - Definiálni egy függvényt ami kilép a pythonból.  
      - Definiálni egy függvényt, ami kiszűri, hogy a begépelt kérdés megfelelő típusú-e.  
      - Definiálni egy függvényt ami a három válasz közül random választ egyet.  
      - Nyomógombhoz hozzárendelni a kilépés függvényt.  
      - Végtelen ciklusban beolvasni a kérdést, leellenőrízni, hogy megfelel-e, ha igen válaszolni, ha nem, akkor kiírni, hogy rossz típusú kérdés.
   6. *Tesztelés:*  
      - teszteljük a programunkat és javítsuk a hibákat ha vannak
   7. *Fejlesztés:*  
      - Állítsuk be, hogy a motor mutatója minden kérdés között vegyen fel egy semleges pozíciót, ahova minden válaszadás után visszatérhet.  
      - Egy RGB LED-del jelezzük ha helytelen kérdést tettek fel. Ha rossz a kérdés (nem lehet rá igennel, nemmel, talánnal válaszolni), akkor pirosan világítson, ha helyes a kérdés akkor pedig zölden.

Kód minta:

from gpiozero import Servo, AngularServo, Button

import unidecode  # pip or conda install might be needed

import sys, random

import time

# determine the angle range for the servo

# motor = Servo(18)

# motor.min()  # measure angle

# motor.max()  # measure angle again

angle\_min = -90

angle\_max = 90

motor = AngularServo(18, min\_angle = angle\_min, max\_angle=angle\_max)

button\_quit = Button(22)

motor.angle = None

yes\_angle = -45

no\_angle = 45

maybe\_angle = 0

angle\_list = [yes\_angle, no\_angle, maybe\_angle]

question\_words = ['hogy', 'hogyan', 'miként', 'hol', 'kinél', 'kitől', 'hol', 'honnan',

                    'honnét', 'hová', 'hova', 'Meddig', 'Merre', 'Mettől', 'Minél',

                    'Mitől', 'Ki', 'Kié', 'Kiért', 'Kiig', 'Kihez', 'Kiként', 'Kinek',

                    'Kitől', 'Kivel', 'Mennyi', 'Hány', 'Mi', 'Mié', 'Miért', 'Miig',

                    'Mihez', 'Miként', 'Minek', 'Mitől', 'Mivel', 'Mikor', 'Hánykor',

                    'Hányig', 'Hánytól', 'Meddig', 'Mettől', 'Mikor', 'Mikortól',

                    'Milyen']

def exit\_program():

    pass

# check if the question contains any of the above words

def check\_for\_words(txt, word\_list = question\_words):

    txt = txt.lower()

    txt = unidecode.unidecode(txt)

    txt = txt.replace('?','')

    txt\_list = txt.split(' ')

    for word in word\_list:

        wl = unidecode.unidecode(word.lower())

        if wl in txt\_list:

            return True

    return False

def answer\_the\_question(motor, angle\_list):

    pass

button\_quit.when\_pressed = exit\_program

while True:

    pass