**5. óra: Mozgásérzékelő**

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. Twitter posztolás készítés:
   1. *Bevezető*:  
      Az online szociális hálózatokat böngészve megfigyelted, hogy egyes celebek feltünően sokat postolnak, akár képeket is. Mintha másból sem állna az idejük, csak az életük megörökítéséből. Elgondolkodtál azon, hogy tudnál versenyre kellni velük és olyan sokat posztolni, de anélkül, hogy igazából időt kellene rá szánnod hosszútávon.

Kis internetes tudakozódás után, kiötlötted, hogy egy fél-automata botot fogsz készíteni, ami egy gombnyomással készít egy fényképet, azt egy stílusszűrőn átereszti, hozzáfűz valami random szöveget és kiposztolja a Twitterre. Így minden egyes gomb nyomással új stílusú változatos képet és szöveget tudsz közzé tenni, akár másodpercenként. Nah ezt múlják felül a celebek!

* 1. *Előkészületek*:  
     - Elkészíteni az áramkört és a hozzá tartozó elektronikai elemeket.  
     - Végezzük el twitter API-val való kommunikáció lépéseit.  
     - Győződjünk meg, hogy létrehozunk egy *twitter* nevű mappát!
  2. *Részletekre bontás:*

- Elkészíteni az áramkört.

- Beimportálni a csomagokat amik segítik a munkánkat: cv2, time, numpy, tweepy, datetime, gpiozero, json és random.

- Létrehozni a kapcsolatot a webkamerával az opencv csomag használatával.

- Inicializálni a nyomógombot.

- Beolvasni a twitter azonosító kódokat.

- Megnyitni a kapcsolatot a Twitter API-val.

- Dictionaryben tárolni a posztolni kívánt információkat.

- Függvényt írni, ami fényképet készít, kiválaszt egy stílusszűrőt és egy szöveget.

- Függvényt írni ami kiposztolja a a Twitter posztot.

- Függvényt írni, ami meghívja az előző két függvényt és ezt hozzárendelni a gomb lenyomásához.

- Végtelen while ciklusban a q gombot lenyomva kilépni a pythonból.

- Megszakítani a kamerával a kapcsolatot kilépés esetén.

* 1. *Elektronika:*  
     - Legyen a webcam előkészítve.  
     - Kössük a nyomógomb egyik lábát a földelésre a másikat a GPIO2-es lábra.
  2. *Kód:*  
     - teszteljük le a kamera működését avval, hogy kiolvasunk képet belőle és ábrázoljuk.  
     - Ellenőrízzük le, hogy a nyomógomb is vezérelhető-e a Raspberryről.  
     - Vegyük át, hogy mi az a *json*, hogy lehet beolvasni, hogy lehet adatot kinyerni.  
       
     with open('twitter.json') as f:

dd = json.load(f)

print(dd)  
dd[‘access\_token’]  
  
- Beszéljük át, hogy mit jelent a *with* paranccsal a megynitás: nincs szükség utána külön bezárnunk a fájlt, mert elintézi a szerkezet.

- Beszéljük át a kép módosításához szükséges listának felépítését. Mutassuk be, hogy tudunk egyszerre több elemhez is hozzárendelni értékeket:  
  
b, c, d = design[0]  
  
- Vegyük át a *\*args* és *\*\*kwargs* argumentumokat.  
  
def test(\*args):

print(args)

s = 0

for x in args:

s = x + s

return s

test(1,2,3)  
  
def testkw(\*\*kwargs):

print(kwargs)

keys = list(kwargs.keys())

print(kwargs[keys[0]])

testkw(first= 'hello', second= 'raspberry', third= 'world')

* 1. *Tesztelés:*  
     - Teszteljük a megírt kód működését.
  2. *Fejlesztés:*  
     -Találjunk ki egyéb kép módosító módszereket és írjuk bele a dictionarybe, mint opciót a hozzá tartozó új szöveggel.

- A kép elkészítése után jelenítsük azt meg a képernyőn és kérdezzen rá a program, hogy szeretnénk-e azt posztolni a Twitteren.

Kód minta:

# In the Settings tab, change the Application type to Read, Write and Access direct messages

# In the Reset keys tab, press the Reset button, update the consumer key and secret in your application accordingly.

# makesquare\_test twitter account

import cv2, time

import numpy as np

import datetime as dt

import json, tweepy

from gpiozero import Button

from random import choice

# inicializald a kamerat

cap =

# inicializald a nyomogombot

button =

# ird be a .json fajl lokaciojat es nevet a ... helyere sztring formatumban

with open(...) as f:

    details = json.load(f)

auth = tweepy.OAuthHandler(details['consumer\_key'], details['consumer\_secret'])

auth.set\_access\_token(details['access\_token'], details['access\_token\_secret'])

api = tweepy.API(auth)

design = [(cv2.edgePreservingFilter, {'flags':1, 'sigma\_s':60, 'sigma\_r':0.4}, 'My eyes are hazy'),

          (cv2.detailEnhance,        {'sigma\_s':10, 'sigma\_r':0.15},           'I am so sharp'),

          (cv2.stylization,          {'sigma\_s':60, 'sigma\_r':0.07},           'I got some style :)'),

]

def snapshot():

    # rogzits egy kepet

    ret, frame =

    func, kwargs, text = choice(design)

    frame = func(frame, \*\*kwargs)

    # mentsd el a kepet a 'twitter/image.png' helyre

    cv2.imwrite(...)

    return text

def send\_tweet(text, obj):

    obj.update\_with\_media('twitter/image.png', text)

def action():

    # hivd meg a snapshot fuggvenyt hogy keszitsen kepet es kivalassza a hozzaillo textet

    text =

    # hivd meg a send\_tweet fuggvenyt hogy posztolj a twitteren

    se...

    print('Tweet has been sent!')

# rendeld hozza a megfelelo muveletet a nyomogombhoz

button.when\_pressed =

while True:

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    time.sleep(0.2)

#### Ha nincs raspberry akkor ezt a szekciot lehet csinalni

# while True:

#     answer = input('`p` - kep keszites; `q` - kilepes: ')

#     if answer == 'p':

#         action()

#     elif answer == 'q':

#         break

#     else:

#         pass

#     if cv2.waitKey(100) & 0xFF == ord('q'):

#         break

#     time.sleep(0.2)

cap.release()

# Bezarunk minden ablakot, amit a program megnyitott

cv2.destroyAllWindows()