**6. óra: Kamerás távirányítású autó**

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. Távirányítású autó készítés:
   1. *Bevezető*:  
      Ha látni akarjuk, hogy a távirányítású autónk merre közlekedik és sikeresen ki tudjuk kerülni az akadályokat, még akkor is, ha szabad szemmel nem látható terepen jár, akkor kamerát kell szerelnünk az autóra. Ez nem csak a játékautóknál hasznos, hanem már az igazi autóknál is sok rendelkezik tolatókamerával, hogy jobban lássuk mi van mögöttünk, valamint ahogy haladunk az önvezető autók felé, egyre fontosabb szerephez jut a kamerával rögzített kép és annak elemzése.
   2. *Előkészületek*:  
      - Elkészíteni az áramkört és a hozzá tartozó elektronikai elemeket.  
      - Készítsük ki az autót amit vezérlünk.  
      - Érdemes lehet power banket is előkészíteni!  
      - Szükség lesz elemekre is, hogy a motorokat meghajtsuk.
   3. *Részletekre bontás:*

- Elkészíteni az áramkört.

- Inicializálni a Motor objektumokat.

- Inicializálni a kamera objektumot.

- Definiálni egy-egy függvényt a jármű előre, hátra, jobbra és balra mozgatására.

- Figyelni a billentyűzetet és a megfelelő billentyűk lenyomásakor a megfelelő mozgást végrehajtani.

- Megjeleníteni a kamera képét a képernyőn.

* 1. *Elektronika:*  
     - Legyen a webcam előkészítve.  
     - Kössük be a motor kontrollert L293D mind a motorra, mind a raspberryre és az áramforrásra.
  2. *Kód:*  
     - teszteljük le a kamera működését avval, hogy kiolvasunk képet belőle és ábrázoljuk.  
     - Ellenőrízzük le, hogy a motorok vezérlése is működik.  
     - Prezentáljuk, hogy miképp működik a pygame esemény érzékelése:  
       
     pygame.init( )

size = 500, 500  
screen = pygame.display.set\_mode(size)

while 1:

for event in pygame.event.get():

print(event)

time.sleep(0.1)  
  
- Vegyük át, hogy lehet elfordítani a képet és a színét megváltoztatni:  
  
frame = np.rot90(frame)

frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

- Prezentáljuk, hogy lehet a kamera képet megjeleníteni a pygame modulban:  
  
while 1:

ret, frame = cap.read()

frame = np.rot90(frame)

frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

frame = pygame.surfarray.make\_surface(frame)

screen.blit(frame,(0,0))

pygame.display.flip()

* 1. *Tesztelés:*  
     - Teszteljük a megírt kód működését.
  2. *Fejlesztés:*  
     - Írjuk át a kódot és tervezzük át az áramkört úgy, hogy ne a billentyűzetről, hanem gombokról kapja az irányítást.

- Csak a kép piros (RED) színét jelenítsük meg a képernyőn a színes kép helyett.

Kód minta:

# importald a Motor es Button objektumokat a gpiozerobol

# importald a numpy-t es nevezd el np-nek

import sys, pygame, pygame.freetype

# importald a time es a cv2 csomagokatimport time

motor\_left = # inicializald a bal oldali motort, elore 23as pin, hatra 24es pin, enable 25os pin

motor\_right = # inicializald a bal oldali motort, elore 27as pin, hatra 22es pin, enable 17es pin

cap = # inicializald a kamerat, bemeno parameternek a '/dev/video0' parametert add meg

pygame.init( )

#Set canvas parameters

size = 500, 500

#Display size

screen = pygame.display.set\_mode(size)

def move\_forward(m1, m2, speed=0.8):

    # mozgasd az m1 motort elore a speed sebesseggel

    # mozgasd az m2 motort elore a speed sebesseggel

def move\_backward(m1, m2, speed=0.8):

    # mozgasd az m1 motort hatra a speed sebesseggel

    # mozgasd az m2 motort hatra a speed sebesseggel

def move\_left(mleft, mright, speed=0.8):

    # mozgasd az mleft motort hatra a speed sebesseggel

def move\_right(mleft, mright, speed=0.8):

    # mozgasd az mright motort hatra a speed sebesseggel

def stop\_motors(m1, m2):

    # megallitod mindket motort egyenkent

while 1:

    ret, frame = # keszits egy kepet

    frame = # forgasd el a kepet 90 fokkal az np.rot90 fuggvennyel

    frame = # alakitsd at a kep szinskalajat a cv2.cvtColor fuggvennyel a cv2.COLOR\_BGR2RGB szinskalaba

    frame=pygame.surfarray.make\_surface(frame)

    screen.blit(frame,(0,0))

    pygame.display.flip()

    for event in pygame.event.get():

        # print(event)

        if event.type == pygame.QUIT:

            # zarjuk le a python programot a sys csomag exit fuggvenyevel

        if event.type == pygame.KEYDOWN:

            if event.key == pygame.K\_RIGHT:

                print('Move right...')

                # mozgasd az autot jobbra a move\_right fuggvennyel es a megfelelo bemeno parameterekkel

            if event.key == pygame.K\_LEFT:

                print('Move left...')

                # mozgasd az autot balra a move\_left fuggvennyel es a megfelelo bemeno parameterekkel

            if event.key == pygame.K\_DOWN:

                print('Move backward...')

                # mozgasd az autot jobbra a move\_backward fuggvennyel es a megfelelo bemeno parameterekkel

            if event.key == pygame.K\_UP:

                print('Move forward...')

                # mozgasd az autot jobbra a move\_forward fuggvennyel es a megfelelo bemeno parameterekkel

        if event.type == pygame.KEYUP:

            print('Stop motors...')

            # allitsd meg az autot a stop\_motors fuggvennyel es a megfelelo bemeno parameterekkel

    time.sleep(0.1)#Wait for 100ms before next button press