



Универзитет у Београду -  
Електротехнички факултет



# Управљање образовном беспилотном летилицом Tello drone помоћу преносивог рачунара

**Предмет:**

Теорија електричних кола

**Професори:**

др Дејан Тошић,

редовни професор

др Милка Потребић Иваниш,

редовни професор

**Студенти:**

Лазар Гајић 2020/0365

Душан Тасић 2020/0208

Наташа Станковић 2021/0281

Београд, *Јануар 2023.* године

## РЕЗИМЕ РАДА

У овом раду проћи ћемо кроз кратак историјат беспилотне летилице, како је дошло до оваквог једног изума и промене које су на њему уследиле, чему је првобитно био намењен, а где је он сада, његову примену, а нама посебно битну, употребу дрона у едукативне сврхе. Циљ употребе дрона у едукативне сврхе није само да се научи како се њиме управља, већ да се научи како програмирати дроне и та знања касније примењивати. Биће представљено корак по корак како кодирати програм у језику Пајтон, којим ће се омогућити управљање дрона путем преносивог рачунара. За почетак ће бити потребно скинути одговарајуће развојно окружење, убацили неопходне библиотеке, а онда се прелази на сам кôд. Могућности има много, а ми смо представили неколико нама занимљивих функција. На следећем линку можете наћи снимак у коме смо показали како све оне раде <https://youtu.be/mFXvW2NqT94>.

# САДРЖАЈ

РЕЗИМЕ РАДА .....	2
САДРЖАЈ .....	3
1. УВОД.....	4
1.1. Историјат.....	4
1.2. Класификација.....	9
1.3. Примена.....	9
2. РАЗВОЈНО ОКРУЖЕЊЕ .....	11
2.1. Преузимање развојног окружења .....	11
2.2. Подешавање развојног окружења.....	12
3. ПРОЈЕКТНИ ЗАДАТАК УЗ ОБЈАШЊЕЊЕ КÔДА.....	13
3.1. Пројектни задатак .....	13
3.2. Решење .....	14
3.3. Објашњење кôда.....	16
4. ЛИТЕРАТУРА.....	22

# 1. УВОД

Дрон, познато као беспилотна летилица (енгл. *unmanned aerial vehicle, UAV*), или возило на даљинско управљање (енгл. *remotely operated vehicle, RAV*), у буквалном преводу значило би зујање, но ми смо термин прихватили из енглеског језика и свима нам је познат као „дрон”.

## 1.1. Историјат

Познато је да је човек од давнина показивао интересовање за радњу која му природно није додељена, летење. Жеља за одлазак у висине подстакла је идеју летилице. Сматра се да почеци реализације летилице почињу са кинеским змајевима, касније балоном (ваздухоплов), све до летилица каквим их данас знамо, пилотних и беспилотних.



Слика 1. ЗМАЈ, спортски



Слика 2. ЗМАЈ, пробна возња авиона из 1962.

Извор [1]

Први подаци о коришћењу беспилотне летилице потичу из августа 1849. године када је Аустрија напала Венецију беспилотним балонима напуњеним експлозивом. Напад је био неуспешан у том погледу што је само неколико балона дошло до свог циља, остатак је ветар одувао назад, неки су се утопили... али сама идеја о коришћењу летилице без пилота је остала погодна за даљи развој [2].

Сличан напад су извеле северне стране у Америчком Грађанском рату (1861 - 1865.) где су, ослањајући се на снагу ветра, пустиле балоне пуне запаљујућих бомби у покушају да изазову пожаре супротстављеној страни.

Како је човек увидео предности коришћења беспилотних летилица јасно је да је њихова примена у наредним годинама расла. Радило се на побољшању и све већој производњи.

Први светски рат изнедрио је технолошка открића међу којима је и дрон чије се управљање извршавало путем радио таласа, технике за коју је заслужан Арчибалд Лоу. Она подразумева комуникацију између предајника и пријемника, а Арчибалд се сматра оцем радиоуправљивог лета. Значајни изум је представљао и *Хјуит-Спери аутоматски авион из 1918.*, као и производ компаније *Дејтон-Рајт* познат под именом *авиоторпедо Кетеринг*, односно *Орао слободе*. Он је имао сатни механизам који би у одређеном тренутку одбацио крила и сјурио се на непријатељски положај. Иако сам дрон и нова технологија радио комуникације није примењена у Првом светском рату, послужила је за даљи развој [3-5].



Слика 3. Арчибалд Лоу

Извор [4]



Слика 4. аутоматски авион

Извор[5]



*Слика 5. авиоторпедо Кетеринг*

*Извор [6]*

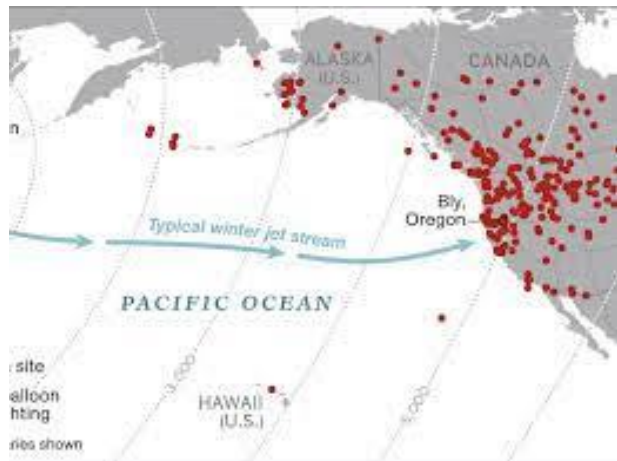
Током касних 1930-их, америчка морнарица је развила беспилотну летилицу под називом *Кертис Н2Ц-2* којом је могло да се контролише путем даљинског из друге летилице, што је дизајн учинило револуционарним. Ваздухопловство америчке војске је такође усвојило овај концепт и почело да га унапређује. Међутим, како се Америка спремала за Други светски рат, експерименти БПЛ-а су били преусмерени за борбену употребу [3].

У Другом светском рату Јапанци су пуштали балоне у Пацифик знајући да ће их струја однети до обала Северне Америке. „Сваки балон је био наоружан противпешадијском бомбом од 15 килограма и четири запаљиве бомбе од 4,5 килограма, као и флеш бомбом за уништавање доказа о уређајима” [7, 8].



Слика 6. Јапанска балон бомба  
фотографисана у граду Њујорк 2. јула  
1945.

Извор [7]



Слика 7. Балон бомбе лансиране из  
Јапана биле су намењене Сједињеним  
Државама- многе су погодиле свој циљ

Извор [8]

Што се тиче иновација током Другог светског рата, дронови којима се управљало путем радио комуникација ушли су коначно у употребу. Развој радијске технологије нагло је порастао.

Компанија *Радиоплејн* је потписала уговор са Сједињеним Америчким Државама и за њихову војску произвела око 15 000 дрона пре и током Другог светског рата. Њихов најпознатији модел звао се *Радиоплејн ОQ-2*. (Занимљива је прича човека који стоји иза ове компаније. Наиме, Дени Рециналд је 1919. године отишао из Енглеске у Сједињене Државе намеравајући да постане глумац у Холивуду. Био је талентован, али је поред глумачке каријере следио још један сан. Заједно са својим партнерима је отворио *Рационалд Дени Индустрију* и продавницу специјализовану за моделе авиона. Посао је еволуирао у компанију *Радиоплејн*.) [4, 10]

Русија се трудила да не заостаје за Америком и последње деценије рата је конструисала *једрилицу посебне намене* модела *ПСН-1* и *ПСН-2*. Навођење је радило на бази инфрацрвене светлости. Експериментисали су на 9 различитих пројеката, али су 1940. године одустали услед високих трошкова. Остао је само један пројекат – авион *Златни орао*, навођен радио-таласима [3, 11].





Слика 8. Радиоплејн OQ-2

Извор [9]

Након рата, развој дронова није стао. Изглед какав данас има појавио се за време Хладног рата када је средином 50-их година 20. века компанија *Рајан Аеронати* акал произвела дрон под називом *Рајан фајерби* који је имао млазни мотор. Овај модел је био опремљен модерним ракетама које су имале могућност праћења циљане мете [12].



Слика 9. Рајан фајерби

Извор [12]



Каснијих година радило се на побољшању многих перформанси, разне компаније су покушале да уђу у овај бизнис, неким је кренуло за руком, неким не. Данас на тржишту имамо широку понуду, а државе које тренутно предњаче у овој области су свакако Турска и Иран.

## 1.2. Класификација

Постоје различите класификације беспилотних летилица, једна од њих би била:

- По начину управљања:
  - ✓ неуправљиве
  - ✓ аутоматски управљане
  - ✓ даљински управљане с разних места и платформи.
- По маси, трајању и висини лета:
  - Нано (до 20 грама, 20 до 25 минута аутономије)
  - Микро (до десет килограма, сат времена лета и радијусом до једног километра)
  - Мини (до 50 килограма, неколико сати лета и радијусом до пет километара)
  - Миди (до 1 000 килограма, 10 до 12 сати лета и висине до 10 000 метара)
  - Тешке (вишестонске, на висинама до 20 000 метара и аутономијом до 24 сата)

Намене су им такође различите, па могу бити: мета и мамац, борбене, логистичке, експерименталне, грађанско-комерцијалне и привредне [3].

## 1.3. Примена

Првобитна улога дронова била је у војне сврхе, но они су последњих година нашли широку примену у свакодневном животу човека и тиме знатно унапредили и олакшали извођења одређених радњи. Користе се за проучавање природних појава и друга научна

истраживања, у пољопривреди, филмској индустрији, медицини, за доставу хране, забаву...

Предности коришћења дрона је много, свуда где је човеку неприступачно дрон може да приђе. Неки примери би били испорука медицинских средстава у Руанди, надгледање лешинара у Монголији, мировне мисије у Демократској Републици Конго...[13] Током ерупције вулкана Јасур 2014. године, подацима добијеним са камера на ободу вулкана утврђено је да постоје три активна отвора која емитују лаву, док је анализом снимка и фотографије прикупљене са беспилотне летелице која је летела директно унутар гасног облака вулкана утврђено да постоји шест активних вентилационих отвора [14].

Употреба дрона у образовању отвара нове трендове у настави и пракси учења на иновативан и привлачан начин. Постоје различите врсте дрона по величини и карактеристикама које их сачињавају, што пружа више могућности за сваког да нешто научи. Циљ није само научити како се управља дроном, већ научити како програмирати дроне и та знања касније примењивати. Кинеска компанија *Pajz Роботикс* изумела је један дрон погодан за такве циљеве. У питању је *Tello drone*, квадрокоптер који је представљен у пролеће 2018. године у сарадњи са *Ди-Џеј Технологи* компанијом.

*Tello drone* користи Вај-Фај од 2.4 GHz за комуникацију (телеметрију, контролне команде и видео пренос). Графичку обраду подржава *Intel Movidius Myriad* процесор, који се такође користи за препознавање објеката. Трајање лета је врло кратко (око 13 минута). Уз званичну апликацију, дрон се може контролисати у просторији и могу се користити различити режими летења. Произвођач такође омогућава приступ роботици за подучавање STEM-а кроз развојно окружење Скреч, Пајтон и Свифт. Обезбеђивањем комплета за развој софтвера и других бесплатних софтверских библиотека, дрон се користи у истраживању и развоју апликација у компјутерском свету.



Слика 10. *Tello drone*

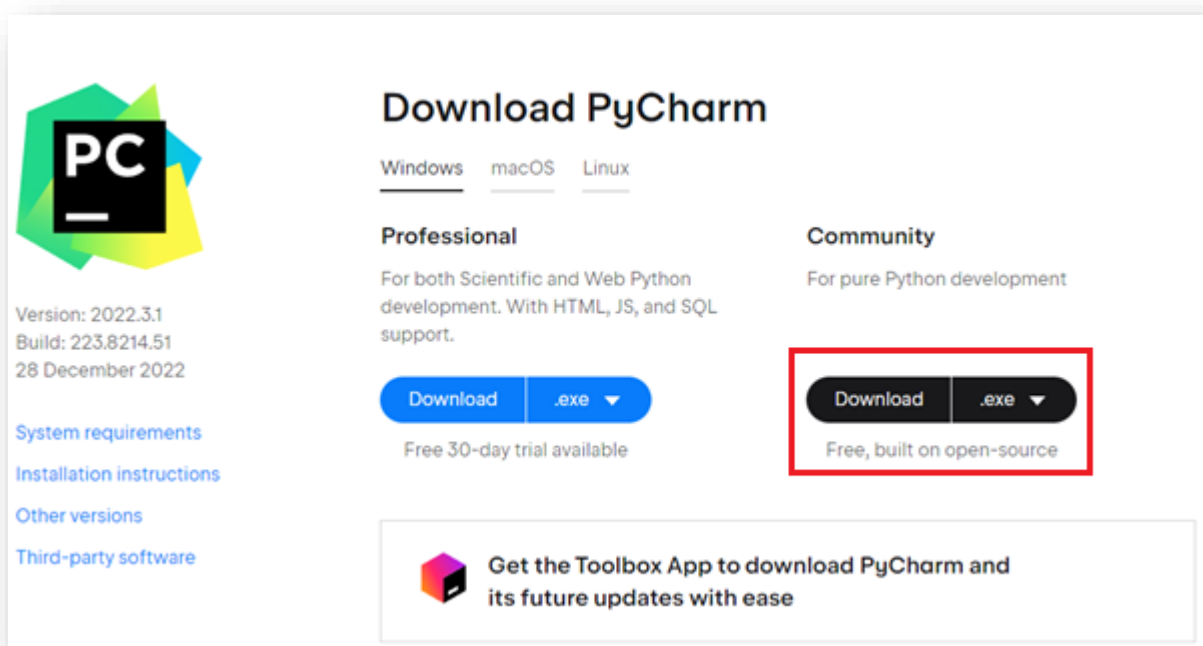
Извор [15]

## 2. РАЗВОЈНО ОКРУЖЕЊЕ

Како би пројекат био прегледнији и лакши за разумевање потребно је изабрати адекватно развојно окружење. У овом случају најадекватније развојно окружење изабрано од стране чланова тима је Пај-Чарм. У наставку текста биће описан начин преузимања и подешавања развојног окружења.

### 2.1. Преузимање развојног окружења

Развојно окружење Пај-Чарм може се преузети са линка <https://www.jetbrains.com/pycharm/download/#section=windows>. Линк нас доводи на званичну страницу развојног окружења (Слика 11.), где је потребно изабрати једно од две понуђене верзије. За израду овог пројекта коришћена је community верзију.



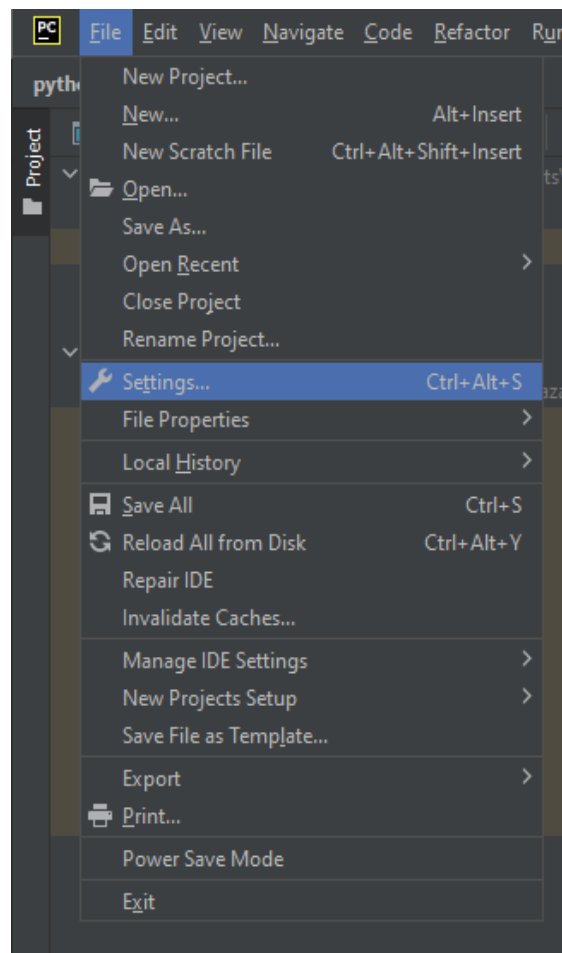
Слика 11. Изглед званичне странице

## 2.2. Подешавање развојног окружења

Након преузимања и инсталације развојног окружења потребно је направити нови пројекат и додати следеће библиотеке у пројекат:

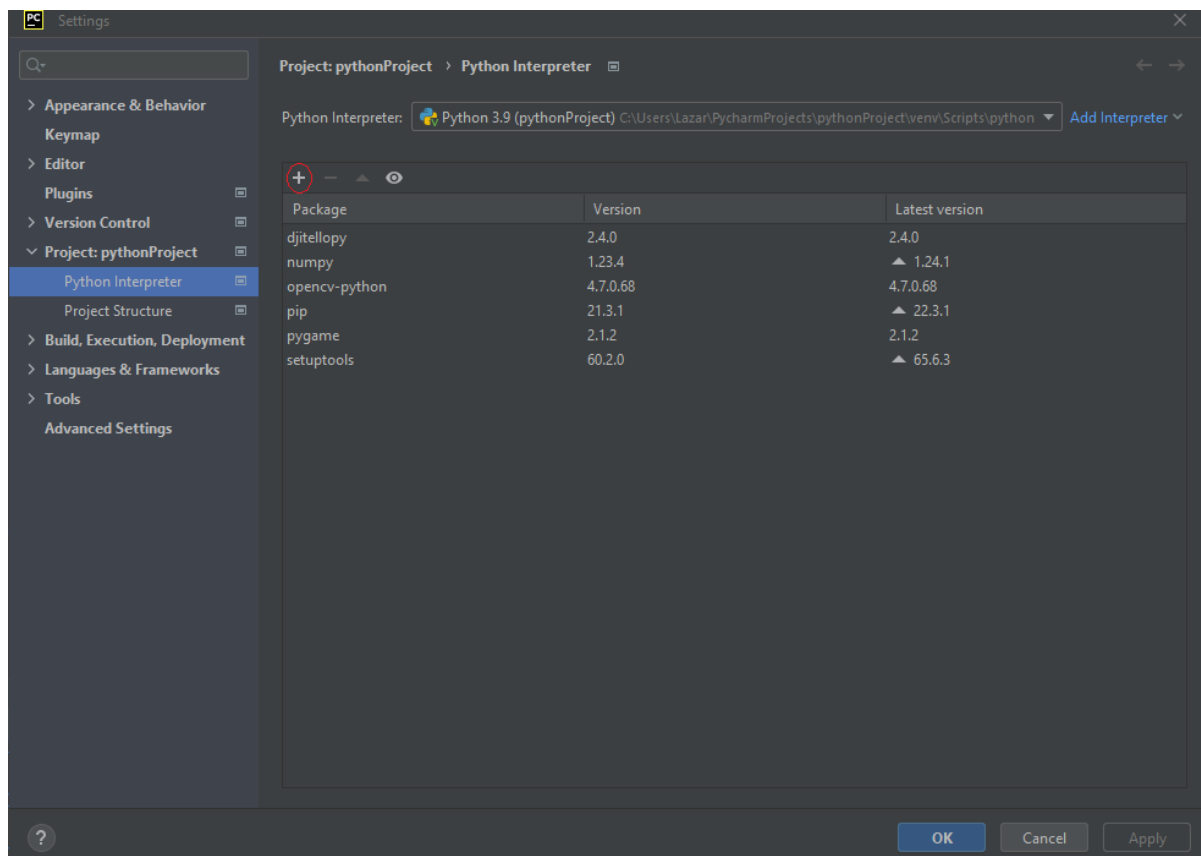
- Djitellory
- Opencv-python
- Pygame

Библиотеке се додају кликом на File->Settings које отвара нови прозор (Слика 12.) или притиском тастера Ctrl+Alt+S.



Слика 12. Отварање прозора Сеттингс

У новоотвореном прозору потребно је кликнути на опцију Project: -> Python Interpreter и кликом на плус одабрати додавање нове библиотеке (слика 13.).



Слика 13. Додавање нове библиотеке

## 3. ПРОЈЕКТНИ ЗАДАТАК УЗ ОБЈАШЊЕЊЕ КОДА

### 3.1. Пројектни задатак

Пројектни задатак у оквиру предмета Теорија електричних кола који је додељен нашем тиму везан је за реализацију једноставне апликације, контролисане из развојног окружења, за покретање дрона и управљање над њим. Притиском на тастер „Т“ дрон полеће и укључује камеру која све време приказује слику на новоотвореном прозору. Притиском на тастер „L“ дрон се спушта на земљу и обуставља свој лет. Притиском на тастер „ESC“ дрон искључује камеру. Притиском на тастер „Р“ дрон прави нову слику коју чува у новом фолдеру под називом „pictures“. За кретање лево, десно, напред и назад користе се тастери „А“, „D“, „W“ и „S“ респективно. За ротацију лево, десно и кретање нагоре и надоле користе се стрелице. Притиском тастера „CTRL“ и неког од тастера „А“, „D“, „W“ или „S“ могуће је урадити окрет са дроном у одговарајућу страну (енгл. Flip).

## 3.2. Решење

```
from djitellopy import tello
import os
import cv2
import time
import shutil
import threading
import pygame

"""Pravljenje puta dir gde ce se cuvati slike"""
file = os.getcwd() + "\pictures"

"""Ako put vec postoji, obrisati prethodne rezultate i opet napraviti
prazan dir"""
if (os.path.exists(file)):
    shutil.rmtree(file)
os.mkdir(file)

drone = tello.Tello()

drone.connect()
print(drone.get_battery())

"""funkcija koja ce se izvorsavati u posebnoj niti"""

def camera():
    while pause:
        try:
            frm = drone.get_frame_read().frame
            cv2.imshow("Image", frm)
            cv2.waitKey(1)
        except Exception:
            print("Nije uspeo da uhvati prvi frame")

"""Pokretanje uzivo strima"""
drone.streamon()
```

```

pause = True

mov = threading.Thread(target=camera)
mov.start()

"""Inicijalizacija prozora iz pygame biblioteke"""
pygame.init()

pygame.display.set_mode((300, 300))

"""Pokretanje main programa"""
while True:
    """Inicijalizacija konstanti parametara koji ce se koristiti za
    kretanje i postavljanje u stanje mirovanja"""
    speedLeft = 0
    speedRight = 0
    speedForward = 0
    speedBackward = 0
    speedUp = 0
    speedDown = 0
    yawLeft = 0
    yawRight = 0
    shift = 1
    pygame.event.pump()
    keys = pygame.key.get_pressed()
    if keys[pygame.K_t]:
        if not drone.is_flying:
            drone.takeoff()
            time.sleep(2)
        continue
    if keys[pygame.K_l]:
        if drone.is_flying:
            drone.land()
            time.sleep(2)
        continue
    if keys[pygame.K_ESCAPE]:
        drone.streamoff()
        pause = False
        drone.end()
        break
    if keys[pygame.K_p]:
        img = drone.get_frame_read().frame
        cv2.imwrite(file + f"/{time.time()}.jpg", img)
        time.sleep(0.5)
    if drone.is_flying:
        if keys[pygame.K_LCTRL]:
            if keys[pygame.K_a]:
                drone.flip_left()
                time.sleep(1)
            if keys[pygame.K_d]:
                drone.flip_right()
                time.sleep(1)
            if keys[pygame.K_w]:
                drone.flip_forward()
                time.sleep(1)
            if keys[pygame.K_s]:
                drone.flip_back()
                time.sleep(1)

```



```

else:
    if keys[pygame.K_a]:
        speedLeft = 50
    if keys[pygame.K_d]:
        speedRight = 50
    if keys[pygame.K_w]:
        speedForward = 50
    if keys[pygame.K_s]:
        speedBackward = 50
    if keys[pygame.K_UP]:
        speedUp = 50
    if keys[pygame.K_DOWN]:
        speedDown = 50
    if keys[pygame.K_LEFT]:
        yawLeft = 50
    if keys[pygame.K_RIGHT]:
        yawRight = 50
    if keys[pygame.K_LSHIFT]:
        shift = 1.5
    drone.send_rc_control(int(shift*(speedRight - speedLeft)),
int(shift*(speedForward - speedBackward)),
                        int(shift*(speedUp - speedDown)),
int(shift*(yawRight - yawLeft)))
    pygame.display.update()

mov.join()

```

### 3.3. Објашњење кода

Пре почетка израде решења потребно је увести одговарајуће библиотеке које ће се користити:

```

from djitellopy import tello
import os
import cv2
import time
import shutil
import threading
import pygame

```

- Djitelory - библиотека у којој се налазе све коришћене функције за управљање дроном
- Os - библиотека за рад са датотекама и фајловима
- cv2 - библиотека која се користи за рад са сликама
- time - библиотека која служи за мерење протеклог времена током извршавања, паузирања кода на одређено време и слично

- shutil - библиотека која служи за копирање фајлова и премештања фајлова на основу задате путање
- Threading - библиотека за додавање нових токова контроле у програм (нити)
- Pygame – библиотека за препознавање унетих карактера са тастатуре

Директоријум у којем ће се чувати слике које дрон направи се креира помоћу библиотека os и cv2. Најпре се специјализира путања до директоријума помоћу функције os.getcwd() на коју се надовезује назив директоријума.

```
file = os.getcwd() + "\\pictures"
```

Након тога је потребно направити директоријум у случају да он не постоји (ако директоријум већ постоји биће обрисан, направиће се нови директоријум и биће му додељен исти назив). То се постиже употребом следећих наредби:

```
if (os.path.exists(file)) :
    shutil.rmtree(file)
os.mkdir(file)
```

Помоћу конструктора класе tello.Tello() из библиотеке djetellory се приступа дрону, додељује му се назив „drone“ и надаље се то име користи за упућивање различитих команди дрону.

```
drone = tello.Tello()
```

Конекцију (повезивање) са дроном вршимо помоћу методе drone.connect(). Потребно је приметити да се све методе за управљање и конекцију са дроном врше над називом drone (инстанца класе Tello). Пре самог покретања програма потребно је прво укључити дрон и повезати се са дроном преко његове приступне тачке.

```
drone.connect()
```

За укључивање камере користи се метода drone.streamon().

```
drone.streamon()
```

Пошто камера захтева посебан ток контроле (енгл. Thread), потребно је успоставити нови ток контроле. Нови ток контроле ће извршавати функцију camera() чија је имплементација дата у наставку.

```
def camera():
    while pause:
        try:
            frm = drone.get_frame_read().frame
            cv2.imshow("Image", frm)
            cv2.waitKey(1)
        except Exception:
            print("Niје uspeo da uhvati prvi frame")
```

За покретање новог тока контроле потребно је направити нови објекат класе Thread и затим позвати методу start(). То се изводи позивањем конструктора и прослеђивањем назива функције коју ће та нит извршавати. Нови ток контроле називамо mov.

```
mov = threading.Thread(target=camera)
mov.start()
```

Препознавање притиснутих тастера се обавља коришћењем библиотеке pygame. За прикупљање притиснутих тастера потребно је отворити нови прозор. То се изводи коришћењем функције pygame.init(). За подешавање величине прозора се користи функција pygame.display.set\_mode((300, 300)), која поставља величину прозора на 300x300 пиксела.

```
pygame.init()
pygame.display.set_mode((300, 300))
```

За започињање лета дрона користи се метода drone.takeoff(). Функција као повратну вредност враћа енгл. Boolean тип податка (True-полетање успешно, False-полетање неуспешно).

```
drone.takeoff()
```

За обуставу лета дрона користи се метода drone.land().

```
drone.land()
```

Помоћу метода drone.flip\_forward(), drone.flip\_right(), drone.flip\_back(), drone.flip\_left() могуће је изводити окрете дрона у различите стране.

```
drone.flip_forward()
drone.flip_back()
drone.flip_right()
drone.flip_left()
```

Регулација кретања у одговарајућу страну се омогућава коришћењем методе `drone.send_rc_control()`. Метода захтева 4 аргумента, `int left_right_velocity`, `int forward_backward_velocity`, `int up_down_velocity`, `int yaw_velocity`. Сви аргументи захтевају целобројне бројеве у размаку од -100 до 100. Први аргумент (`left_right_velocity`) одређује да ли ће се дрон кретати улево или удесно. Ако је аргумент мањи од нуле дрон ће се кретати улево, уколико је аргумент већи од нуле дрон ће се кретати удесно, ако је пак аргумент једнак нули дрон се неће кретати ни у једну од наведених страна. Други аргумент на исти начин као и први аргумент одређује да ли ће се дрон кретати унапред или уназад. Трећи и четврти аргумент на исти начин одређују да ли ће се дрон кретати нагоре или надоле, односно да ли ће се дрон ротирати улево или удесно.

```
drone.send_rc_control(int left_right_velocity, int  
forward_backward_velocity, int up_down_velocity, int yaw_velocity)
```

Након разумевања рада датих метода може се прећи на изградњу самог програма. Програм се реализује помоћу бесконачне петље (`while True`) у којој се помоћу метода `pygame.event.pump()` и `pygame.key.get_pressed` дохвата притиснути тастер на тастатури. На основу притиснутог тастера се извршава нека од горенаведених команди. Након једне завршене итерације `while` петље позива се метода `pygame.display.update()` која освежава прозор и омогућава унос новог карактера са тастатуре.

```

while True:
    """Inicijalizacija konstanti parametara koji ce se koristiti za
    kretanje i postavljanje u stanje mirovanja"""
    speedLeft = 0
    speedRight = 0
    speedForward = 0
    speedBackward = 0
    speedUp = 0
    speedDown = 0
    yawLeft = 0
    yawRight = 0
    shift = 1
    pygame.event.pump()
    keys = pygame.key.get_pressed()
    if keys[pygame.K_t]:
        if not drone.is_flying:
            drone.takeoff()
            time.sleep(2)
        continue
    if keys[pygame.K_l]:
        if drone.is_flying:
            drone.land()
            time.sleep(2)
        continue
    if keys[pygame.K_ESCAPE]:
        drone.streamoff()
        pause = False
        drone.end()
        break
    if keys[pygame.K_p]:
        img = drone.get_frame_read().frame
        cv2.imwrite(file + f"/{time.time()}.jpg", img)
        time.sleep(0.5)
    if drone.is_flying:
        if keys[pygame.K_LCTRL]:
            if keys[pygame.K_a]:
                drone.flip_left()
                time.sleep(1)
            if keys[pygame.K_d]:
                drone.flip_right()
                time.sleep(1)
            if keys[pygame.K_w]:
                drone.flip_forward()
                time.sleep(1)
            if keys[pygame.K_s]:
                drone.flip_back()
                time.sleep(1)
        else:
            if keys[pygame.K_a]:
                speedLeft = 50
            if keys[pygame.K_d]:
                speedRight = 50
            if keys[pygame.K_w]:
                speedForward = 50
            if keys[pygame.K_s]:
                speedBackward = 50
            if keys[pygame.K_UP]:
                speedUp = 50

```

```
        if keys[pygame.K_DOWN]:
            speedDown = 50
        if keys[pygame.K_LEFT]:
            yawLeft = 50
        if keys[pygame.K_RIGHT]:
            yawRight = 50
        if keys[pygame.K_LSHIFT]:
            shift = 1.5
    drone.send_rc_control(int(shift*(speedRight - speedLeft)),
int(shift*(speedForward - speedBackward)),
int(shift*(speedUp - speedDown)), int(shift*(yawRight - yawLeft)))
    pygame.display.update()
```

## 4. ЛИТЕРАТУРА

- [1] URL: <https://www.enciklopedija.hr/natuknica.aspx?id=67328>
- [2] URL: <https://www.findingdulcinea.com/news/on-this-day/July-August-08/On-this-Day--Austria-Rains-Balloon-Bombs-on-Venice.html>
- [3] URL: [https://www.warhistoryonline.com/military-vehicle-news/short-history-drones-part-1-x.html?chrome=1&Exc\\_D\\_LessThanPoint002\\_p1=1](https://www.warhistoryonline.com/military-vehicle-news/short-history-drones-part-1-x.html?chrome=1&Exc_D_LessThanPoint002_p1=1)
- [4] URL: [https://oruzjeonline.com/2022/08/16/leti-leti-dron/#Prvi\\_koraci](https://oruzjeonline.com/2022/08/16/leti-leti-dron/#Prvi_koraci)
- [5] URL: <https://www.youtube.com/watch?v=sDsSL2HHbEY>
- [6] URL: <https://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/198095/kettering-aerial-torpedo-bug/>
- [7] URL: <https://www.findingdulcinea.com/news/on-this-day/May-June-08/On-this-Day--Japanese-WWII--Balloon-Bomb--Kills-Six-in-Oregon.html>
- [8] URL: <https://www.nationalgeographic.com/science/article/130527-map-video-balloon-bomb-wwii-japanese-air-current-jet-stream>
- [9] URL: <https://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/196292/radioplane-oq-2a/>
- [10] URL: [https://www.militaryfactory.com/aircraft/detail.php?aircraft\\_id=331](https://www.militaryfactory.com/aircraft/detail.php?aircraft_id=331)
- [11] URL: <http://all-aero.com/index.php/60-gliders/19970-nikitin-psn-1>
- [12] URL: <https://www.scalespot.com/reviews/kits/firebee-icm/review.htm>
- [13] URL: <https://nationalgeographic.rs/istorija-i-kultura/arheologija/a21729/dronovi-visokoletacka-pomoc.html?page=2>
- [14] Jordan, B. R. (2015). A bird's-eye view of geology: The use of micro drones/UAVs in geologic fieldwork and education. *GSA today*, 25(7), 50-52.
- [15] URL: <https://www.ryzerobotics.com/tello>