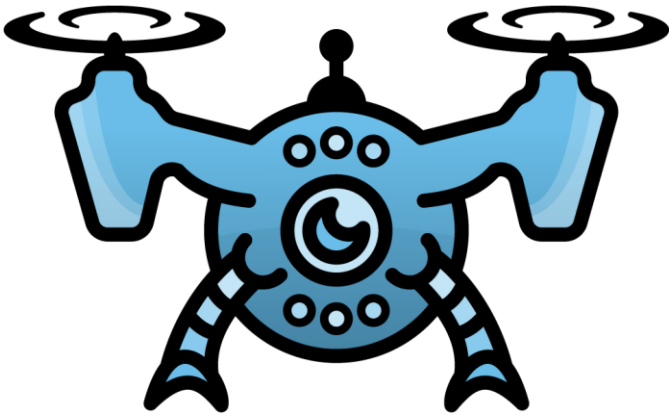
Co-funded by the
Erasmus+ Programme
of the European Union

САВРЕМЕНА ИТ ЛИТЕРАТУРА

Приручник за практичну наставу



Укратко о програму Ерасмус+

Еразмус+ је један од највећих програма ЕУ који финансира пројекте мобилности и сарадње у области образовања, обука младих и спорта. Настао је 2014. године. Општи циљ Програма је да, кроз целоживотно учење, подржи образовни, професионални и лични развој људи у области образовања, обука, омладине и спорта.

Провођење времена у другој земљи са циљем учења, односно усавршавања и рада требало би да постане стандард, као и познавање два страна језика поред матерњег.

На тај начин Програм значајно доприноси одрживом расту, квалитетнијим радним местима и социјалној кохезији, подстиче иновације и премошћавање јаза у знању, вештинама и компетенцијама у Европи.

Укратко о пројекту:

Циљ пројекта "Са роботом на ТИ" је да се кроз стратешко партнерставо развију нове кључне компетенције наставника и ученика у почетном и континуираном ИТ образовању, где ће се развити професионалне компетенције наставника стручних предмета у ИТ образовању и унапредити њихово знање из нових технологија које стижу на светска тржишта. Заједно са ученицима кроз пројекат ће се поставити три различите обуке веома важне за будућност младих и иновативност школа јер се на тај начин прате потребе тржишта и модернизује се настава у школама а све у складу са стратегијом стручног образовања. Крајњи циљ је да ИТ компаније Европе добију младе, едуковане ИТ раднике.

Co-funded by the
Erasmus+ Programme
of the European Union

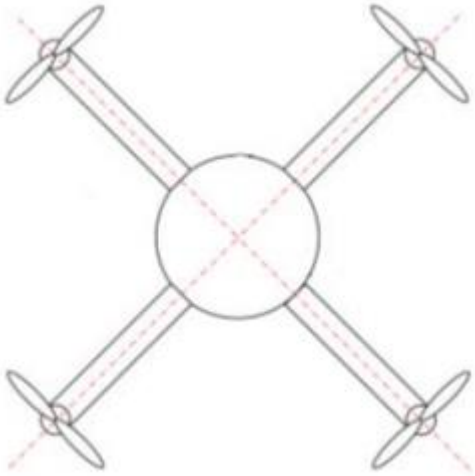
ПРВА ОБУКА – КОНТРОЛА ДРОНА

- Шта су дронови, како су настали и како раде -

Ваздухоплов којим се управља даљинским преносом сигнала или који лети аутономно по задатим иструкцијама назива се беспилотна летелица (енглески Unmanned aerial vehicle) или популарно дрон. Развој ових летелица су започели војни стручњаци и дуго су се користиле искључиво у војне сврхе. Срећом у последње време имају све више примена у цивилном сектору, тако да данас имамо експанзију примене дронова. Тој експанзији суштински доприносе ниски трошкови набавке и одржавања, али и огромне могућности примене. Користе се у експерименталне, истраживачке и забавне сврхе, али све је више и конкретних примена у индустрији, здравству, пољопривреди, итд. Опремају се камерама, помоћу којих се крећу у простору и "виде", праве снимке терена које прелећу и могу да понесу одређени терет. Поред тога могу се опремити и разним сензорима који омогућавају да дронови шаљу податке о измереним физичким величинама, али и да аутономно извршавају акције на основу програма који се извршава на микроконтролеру.

Основне карактеристике дронова, које сваки произвођач назначава, су максимална висина, брзина и трајање лета и максимални нагиб дрона. Ове величине су битне за планирање намене и управљање дроном.

Дронови се конструишу на различите начине. Најчешћа конструкција дронова се назива квадкоптер. То је четворокраки дрон са мотором на сваком краку. Елисни електромотори се налазе у истој равни и вертикално су оријентисани тако да стварају силу узгона.



Идејно конструкцијско решење квадкоптера [2]

Овакав концепт конструисања летелица потиче још од 20-тих година прошлог века, али није могао додатно да се развија. Пре свега нису могли да се реше проблеми координисаног управљања моторима па концепт није био практично примењив. Развојем микроконтролера омогућена је и ефикаснија примена квадкоптера, а развојем сензора области примене су проширене.

Делови дрона су следећи:

1. Тело дрона (frame)
2. Мотори
3. Елисе
4. Контролери мотора (ESC)
5. Разводна плоча за напајање(PDB)
6. Контролер лета (flight controller)
7. Батерије
8. Пријемник за bluetooth
9. Камера
10. Видео предајник (VTX)
11. Сензори

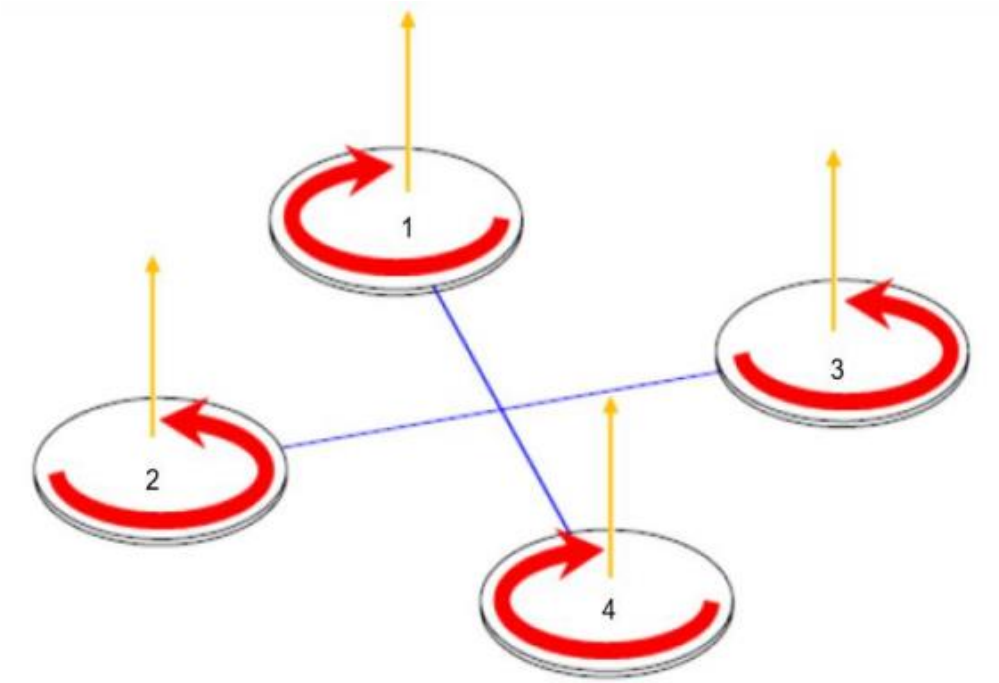
Тело дрона је основа за монтажу свих осталих делова. Потребно је да буде направљено од чврстог и лаког материјала. Већина дронова се прави од угљеничних влакана (carbon fiber).

Мотори који се користе могу бити једносмерни мотори са четкицама (BDC) или мотори без четкица. Мотори са четкицама су јефтини и једноставни, а без четкица су скупљи, мањи, издржљивији, постижу веће брзине обртања, ефикаснији су, итд. На ротор мотора су монтиране елисе (propellers) које стварају силу узгона која је сразмерна брзини обртања ротора мотора.

Као што смо већ рекли код квадкоптера погонски мотори са елисама су фиксни и вертикално оријентисани. Сваки мотор па и елиса на њему има могућност промене брзине окретања и то независно у односу на остале. Како ће се и којом брзином обртати мотори, сваки од њих појединачно, одговорни су сензори и микроконтролери. Овакво организовање елиса мотора даје могућност дрону кретања са четири степена слободе.

Сваки од четири ротора окреће се у односу на два суседна у супротним смеровима. Ротори који се налазе један насупрот другог крећу се у истим смеровима. Приказ смера окретања ротора приказан је на слици[2].

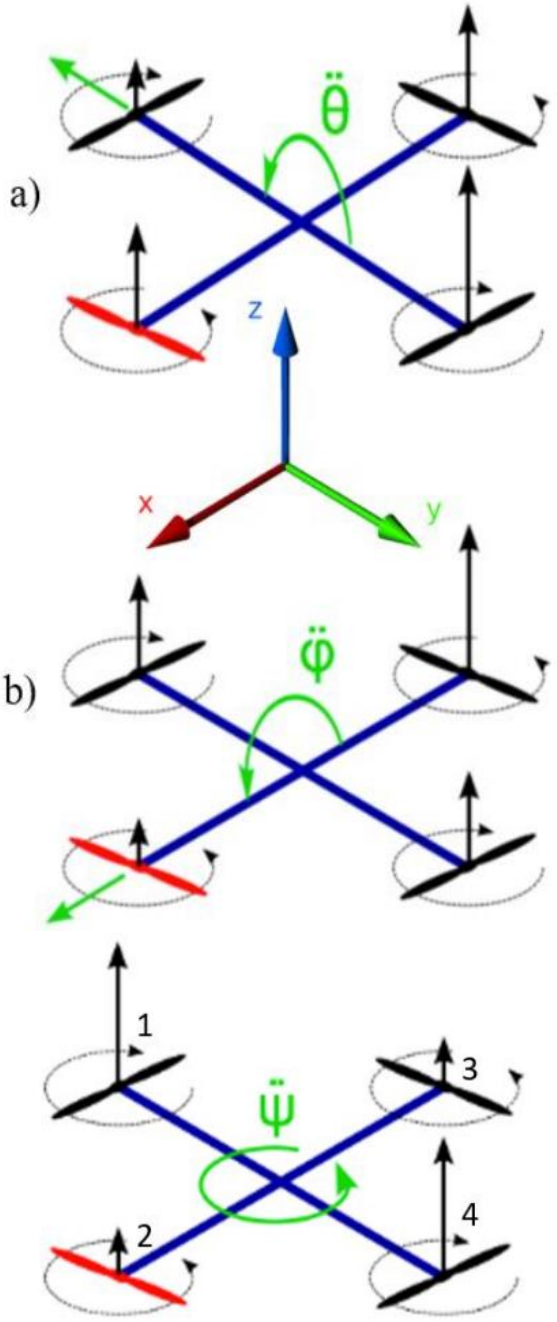
Conceptual design of quadcopter [(2)]



Смер обртања ротора квадкоптера[2]

Када дрон лети на сталној позицији изнад одређене тачке онда је дрон у режиму лебдења. У овом режиму летења погонска сила узгона једнака је тежини дрона. Погонска сила узгона настаје као последица рада сва четири мотора. Уколико је сила узгона већа од тежине дрона он ће се кретати верикално на горе. Ако је сила узгона мања од тежине дрона он ће се кретати вертикално на доле. Колика је сила узгона зависи од брзине обртања ротора мотора. Ако је брзина мања и сила узгона је мања и обрнуто.

За обезбеђење бочног кретања, у равни тежишта дрона, без промене висине летења, потребно је да се ротори три мотора окрећу истом брзином, а ротор четвртог мотора да се окреће другачијом брзином. Кретање у једном од четири смера може се обезбедити и повећањем брзине обртања ротора два суседна мотора. Приликом оваквог кретања дрон се нагиње на страну па је потребно пазити на максималну величину угла нагиба. За ротационо кретање у равни тежишта дрона потребно је да се смањи брзина обратања ротора наспрамих мотора. Следеће слике демострирају покретање дрона[2].



Објашњење кретања дрона у односу на x, y и z осу [2]

Управљање моторима се врши употребом контролера мотора (Electric Motor Controller, ESC). Сваки мотор може да се управља са по једним контролером али могу да се користе и контролери који управљају свим моторима.

Разводна плоча за напајање(Power Distribution Board, PDB) обезбеђује напајање свим активним компонентама дрона: моторима и контролеру. Напајање мотора се остварује разним типовима батерија које морају бити лагане, ефикасне и у могућности да обезбеде максимално време лета.

Контролер лета (flight controller) је централна управљачка јединица дрона. Ради се о микроконтролерској плочици. Микроконтролер на основу сигнала управљања које добија преко Bluetooth пријемника, сигнала са сензора и програма у себи, преко контролера мотора управља дроном.

Bluetooth пријемник омогућава комуникацију дрона са управљачким уређајем на земљи. Упаривањем уређаја корисник добија могућност да шаље управљачке наредбе употребом џојстика или тастатуре управљачког уређаја.

Стандардни део опреме дрона је камера која може снимљени видео да смешта у меморијски модул на дрону или да га шаље путем бежичне мреже (wireless). Трансфер видеа се обавља употребом видео предајника (Video Transmitter, VTX).

Сензори су уређаји који мере физичку величину и обезбеђују њен електрични еквивалент (аналогни или дигитални) који се преко улаза микроконтролера користе за дефинисање понашања дрона. Они су често подразумевани делови дронова, али се дронови могу опремити и додатним сензорима. Најчешће су дронови опремљени сензорима притиска, који служе за одређивање висине на којој дрон лети, GPS сензор за одређивање позиције дрона и сензори за одређивање убрзања и нагиба дрона.

Основне конструкцијски захтеви су следећи:

1. Раван ротора мотора мора бити изнад равни у којој се налази тежиште квадкоптера. Ако би раван у којој се налазе ротори била и раван у којој се налази тежиште постоји опасност од превртања квадкоптера приликом маневара око хоризонталне равни и то услед момената тежине око тежишта. Превртање значи губитак сила узгона и пад. Ово имплицира да тежиште дрона треба да буде што ниже и да се сваки додатни терет или опрема постављају испод основе дрона. Ово је услов за већу покретљивост дрона, јер се приликом нагињања дрона референтна раван елисе додатно „спушта“.
2. Вертикална оса свих мотора мора лежати у истој кружници са центром који се налази на средишњој оси тежишта конструкције.
3. Симетричност конструкције ради лакше регулације и боље стабилности
4. Осигуран простор за батерије и осталу опрему
5. Што мање масе али да се задовоље захтеви за чврстоћом.
6. Простор изнад и испод ротора са што мање конструкцијских делова и по могућности постојање елемената који би при евентуалном паду преузимали већину енергије удара, а да притом буду лако заменљиви.

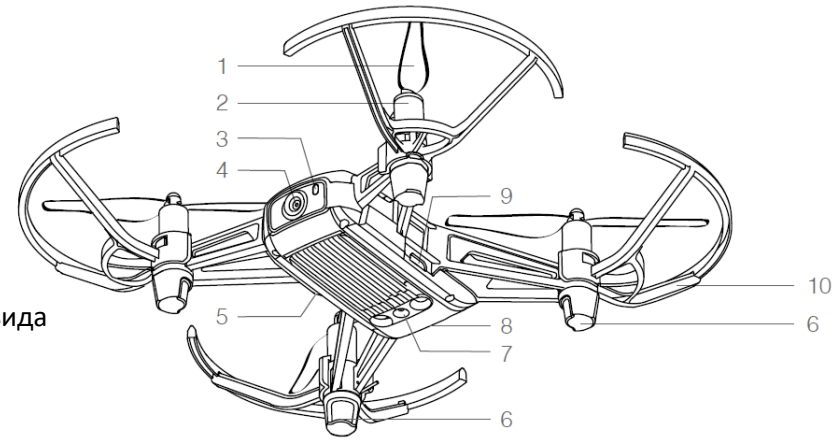
Co-funded by the
Erasmus+ Programme
of the European Union

TELLO DRONE

- главне карактеристике-

Опис компоненти беспилотне летелице Tello Edu:

1. Пропелери
2. Мотори
3. Индикатор статуса дрона
4. Камера
5. Дугме за укључивање
6. Антене
7. Систем за позиционирање вида
8. Батерија
9. Микро УСБ порт
10. Штитници елисе



Опис ваздухоплова

Његове димензије и карактеристике га чине врло управљивим.

Tabela 1. Main charasteristics

Weight	87 g
Dimensions	98×92.5×41 mm
Propeller	3 inches
Integrated Functions	Telemetric sensor
	Barometer
	LED
	Vision System
	Wi-Fi 2.4 GHz 802.11n
	Real-time streaming 720p
Port	USB battery charging port
Operating temperature range	from 0° to 40°
Operating frequency range	from 2.4 to 2.4835 GHz
Transmitter (EIRP)	20 dBm (FCC)
	19 dBm (CE)
	19 dBm (SRRC)

Детаљи функционисања

Будући да је беспилотна летелица за унутрашњу употребу, његове карактеристике функционисања су прилично ограничене.

Tabela 2. Flight charasteristics

Maximum distance of flight	100 m
Maximum speed	8 m/s
Maximum flight time	13 min
Maximum flight height	30 m

Детаљи о батерији

Има врло једноставно промењиву батерију; може се пунити директно помоћу УСБ кабла са батеријом унутар дрона или помоћу станица за пуњење.

Tabela 3. Battery details

Removable	Yes
Capacity	1100 mAh
Voltage	3.8 V
Type	LiPo
Energy	4.18 Wh
Net Weight	25 ± 2 g
Temperature range when charging	from 5° to 45°
Maximum Load Power	10 W

Детаљи о камери

Камера нам омогућава да добијемо видео запис доброг квалитета (HD), након обраде слике могуће је развити различите апликације.

Tabela 4. Camera details

Photo	5 MP (2592x1936)
Field of view	82.6°
Video	HD 720p 30 fps
Format	JPG (Photo)
	MP4 (Video)
Electronic stabilization	Yes

Систем за позиционирање вида: Састоји се од камере и инфрацрвеног 3Д модула. Овај систем може радити у распону од 0,3 м до 30 м висине, али су његови оптимални радни услови високи од 0,3 м до 6 м.

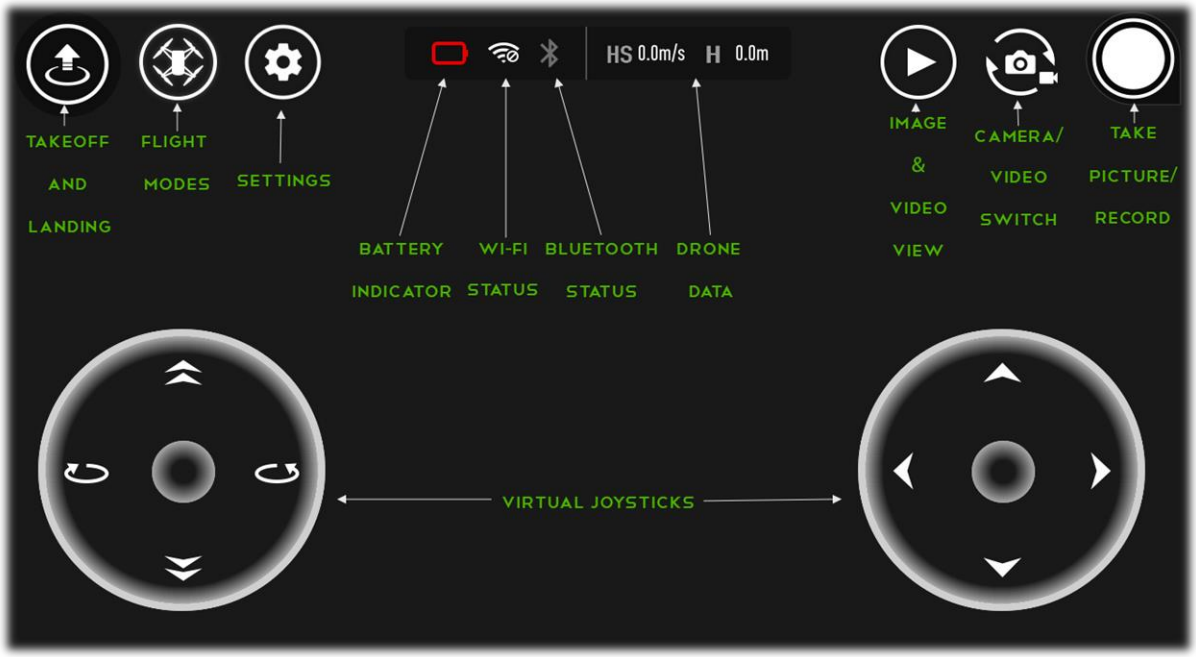
Индикатор статуса беспилотне летелице: То је ЛЕД лампица која има дрон и показује у каквом је стању беспилотна летелица у сваком тренутку. (Укључује се, без примања порука ...)

Видно поље: То је отворена видљива област коју камера дрона може видети.

Електронска стабилизација: То је техника побољшања слике помоћу електронске обраде.

Како управљати дроном

Преузмите и инсталирајте апликацију Tello. Након тога морате се повезати на Tello Wi-Fi.



DRONEBLOCKS

- шта је DRONEBLOCKS, примери -

Рачунарско програмирање или кодирање брзо привлачи пажњу у образовним круговима. А оно што се веровало да је област софтверских инжењера и информатичара сада је доступно скоро свима. Све због појаве програмског интерфејса Drag and Drop (превуци и пусти).

Како се појављују нове технологије, познавање кодирања или барем разумевање начина на који рачунари раде ће нас боље припремити за интеракцију са технологијом око нас и њено потпуно коришћење.

Шта је програмски језик превлачења и пуштања?

Као што име сугерише, програмски језик превлачења и пуштања је визуелни интерфејс у којем без знања о синтакси (синтакса је попут правописа и граматике рачунарског програмског језика) и даље можете програмирати рачунар тако што ћете слагати делове загонетке и склапати их у једно. Када правилно поставите блокове, ваш програм ће радити.

Лепота ове врсте програмирања је у томе што је могу користити сви, чак и деца, а без ометања интерпункцијских знакова можете се усредсредити на праву логику иза алгоритама (алгоритми су логички кораци на које делимо задатак да буде разумљив компјутеру).

Једна од таквих апликација је DroneBlocks, која се користи за писање програма за управљање дроновима. DroneBlocks је програмско окружење блокова превлачења које подржава многе водеће беспилотне летелице, као што су: Phantom 3, Phantom 4, Mavic Pro, Mavic Air, Spark, и Tello. Ово је бесплатна апликација која је доступна на различитим платформама: iOS App Store, Google Play Store and Chrome App Store. Једна од највећих предности је та што се мобилни телефон може користити за прављење програма и управљање дроном.

Мисија 1 - "Здраво свете"

Традиција је да се приликом учења новог програмског језика започиње једноставним задатком „Здраво свете!“. Ово се генерално састоји од разумевања језичке синтаксе, а затим и доживљаја "Здраво свете!" одштампан на екрану.

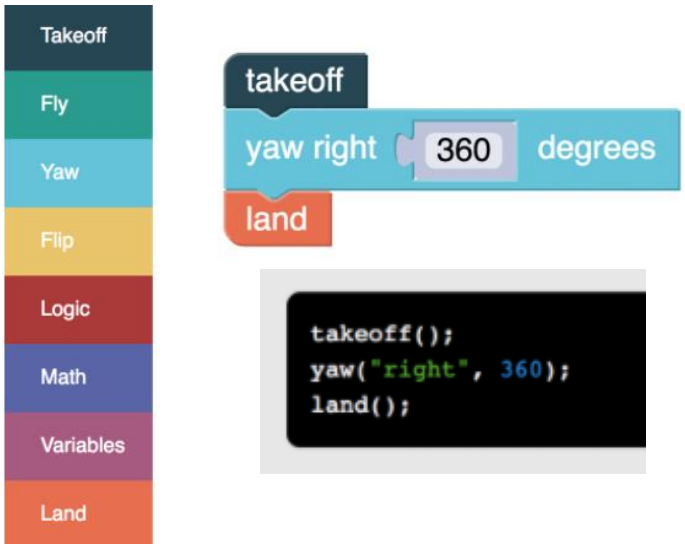
Пошто нећемо ништа штампати на екрану, бавићемо се нечим још узбудљивијим. Ми ћемо програмирати основну мисију где дрон полети, ротира се за 360 степени, а затим слети. Осим тога, ово ће бити одличан начин да се упознате са неколико програмских блокова у DroneBlocks.

DroneBlocks пружа велики број блокова који могу контролисати понашање дронова. Сваки блок може се извршити појединачно или у групи познатој као програм или мисија. Први програм који ћемо креирати састојаће се од три блока:

- полетање
- окретање око своје осе („завијање десно“ или „завијање лево“)
- слетање

Приликом издавања команде за полетање, дрон Tello ће се попети на висину од око 1,2-1,5 метара. Треба узети у обзир да овај дрон, за разлику од неких напреднијих, нема уграђен ГПС, па тачност овог дрона није константно прецизна. Tello користи сензоре и логику контроле лета за одређивање своје висине и удаљености.

Након полетања, DroneBlocks ће издати следећу команду и послати је контролеру дрона. Ово се сматра секвенцијалном логиком, где се сваки блок извршава одређеним редоследом. За усмеравање дрона ка ротацији од 360 степени користи се ротациони блок око своје осе.



Слика 2. Решење мисије 1 и приказ у JavaScript-у

Коначно, користи се блок за слетање, који дрону даје команду да слети на своју тренутну локацију. Уверите се да је ваше подручје чисто за слетање како дрон не би ударио у препреке. Ваш командни блок би требало да изгледа као на слици 2.

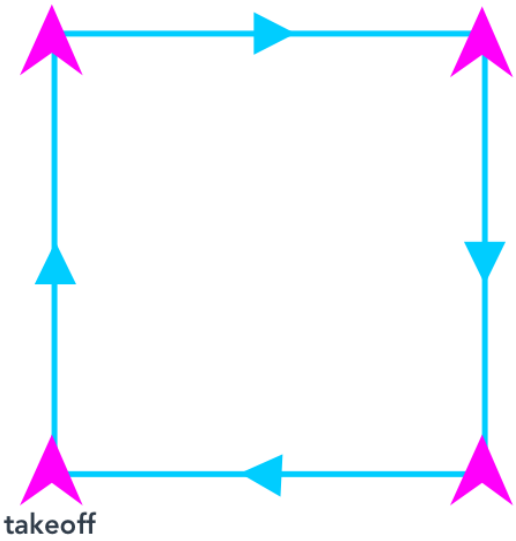
Одлична карактеристика DroneBlocks је што можете да видите JavaScript у својој блок мисији. Кликните на дугме са стране (hamburger - три линије) и изаберите „Прикажи шифру мисије“. Слика 2. је приказ JavaScript кода наше прве мисије.

Мисија 2 - квадрат

У овој лекцији ћемо објаснити како се користе DroneBlocks и Tello за кретање по оквиру квадрата. Овој мисији ћемо приступити користећи две различите технике: једну мисију која држи нос дрона усмереном напред и другу која користи блокове „завоја“ да усмери нос дрона у правцу лета. Ови концепти лета корисни су за разумевање приликом учења програмирања сложенијих мисија.

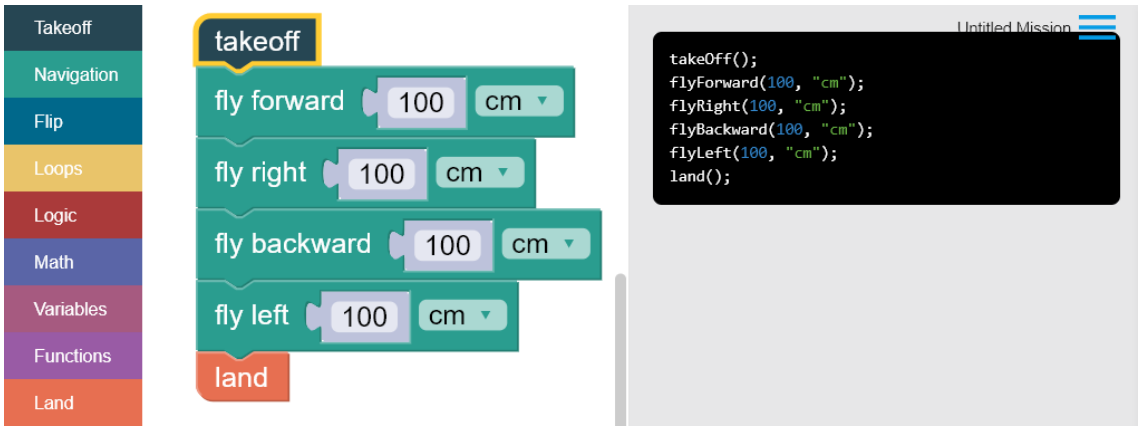
Направите програм који ће контролисати кретање дрона према шеми са слике 4.

Полетање је место где ће наш дрон полетети. Са слике се види да се дрон не сме окретати, односно нос (предњи део) увек треба да буде окренут напред. Мисија се завршава када се дрон врати на почетну позицију.



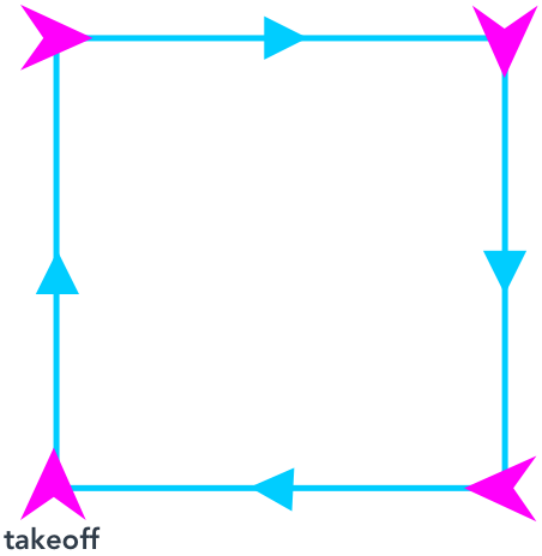
Слика 4. Шема мисије 2

Слика 5. представља решење ове мисије. Прво смо поставили блок за полетање, затим смо поставили блок тако да се дрон помера напред. Овај блок се налази на картици „Навигација“ (као и сви други који се односе на кретање дронова). Вредност за коју би беспилотна летелица требало да се помера изражена је у центиметрима и износи 100 (могуће ју је подесити у инчима). Након тога, дрон се окреће удесно, затим, уназад и на крају улево. На крају, наравно, поставили смо „копнени“ блок, којим завршавамо сваки програм / мисију.



Слика 5.Решење мисије 2 и приказ у JavaScript-у

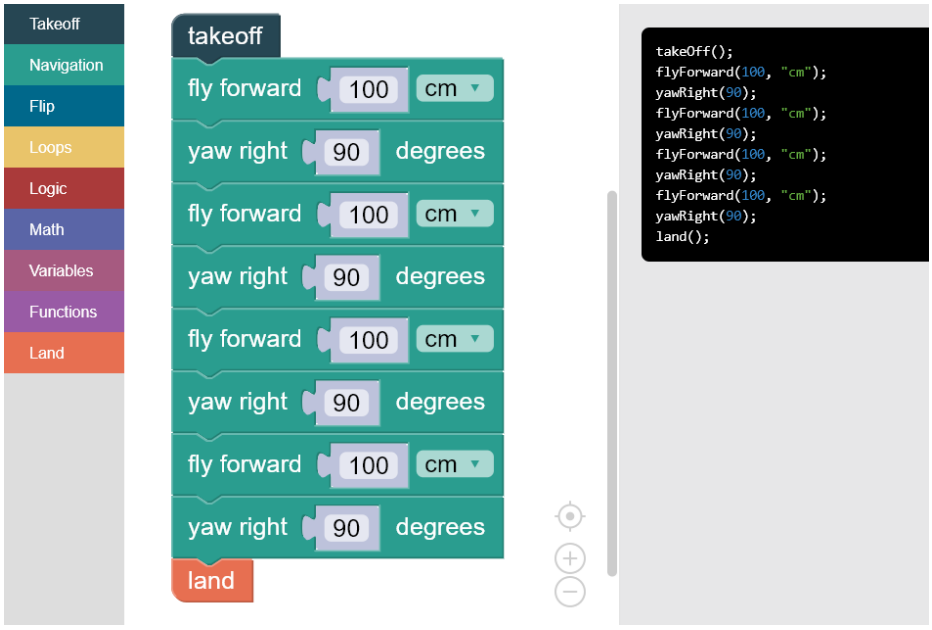
Направите програм који ће контролисати кретање дрона према шеми са слике 6.



Слика 6. Шема за кретање дрона

Ова мисија ће започети слично претходној, само ће се нос дрона морати окренути у правцу дрона. Приликом писања кода користићемо блок под називом „закрени десно“.

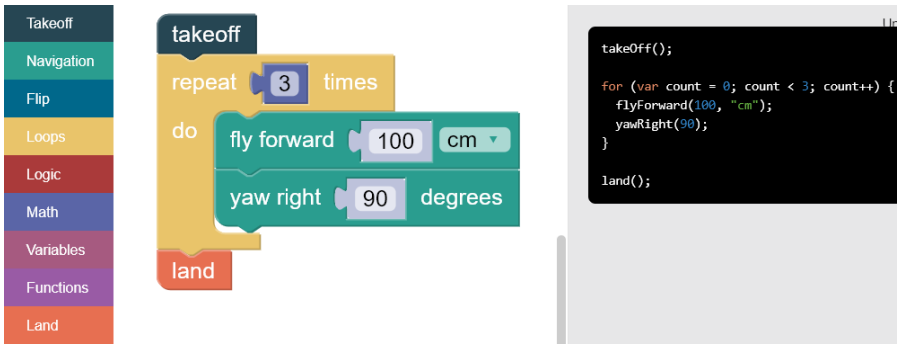
Прво ћемо поставити блок „полетање“, а затим пустити дрон да се креће напред, као у претходном задатку. Затим, морамо да окренемо нос дрона у правцу следећег покрета, односно нос увек треба да буде напред. То чинимо уз помоћ блока „скретање десно“, а угао под којим окрећемо дрон је 90 степени. Након тога понављамо блок команде „лети напред“ и „завијај десно“, тако да имамо укупно 4 понављања. Завршавамо са земљишним блоком. Слика 7. приказује решење ове мисије. Пошто се у овој мисији понавља много блокова, тако да их не позивате стално са картице, можете десним тастером миша кликнути на блок да бисте добили нови мени у којем имате опцију да дуплирате изабрани блок.



Слика 7.

Мисија 3 – Квадрат из картице Петљи

Проблем, чије је решење приказано на слици 9, може се решити са далеко мање блокова помоћу петљи. У том случају користимо блок са картице „Петља“ у који смештамо део из претходног задатка који се понавља четири пута („лети напред“ и „скрени десно“). Слика 8 приказује графички приказ решења ове мисије.



Слика 9. Шема за кретање дрона уз помоћ петље

Мисија 4 – употреба варијабли

У овој лекцији ћемо погледати варијабле. Упамтите, можете унети вредности удаљености и степена у статичке блокове кода, као што су блокови за навигацију. Ове вредности су статичне и могу се променити само када креирате своју мисију. У многим случајевима желите да одређени улази буду динамички или променљиви током програмирања. Променљиве вам омогућавају да промените унос током извршавања кода.

Одличан начин да се демонстрира употреба варијабли је програмирање вожње дроном.

Прво ћемо показати зашто су варијабле важне. У нашој вожњи авионом, дрон ће летети 50 цм напред, а затим ће се вратити у почетну позицију. Током другог дела мисије, дрон ће се окренути и одлетети 100 цм напред, а затим се вратити. Ови кораци ће се понављати у корацима од 50 цм, са максималном удаљеношћу од 250 цм, укупно пет понављања. Без употребе

Слика 10. Решење мисије 4 без варијабли

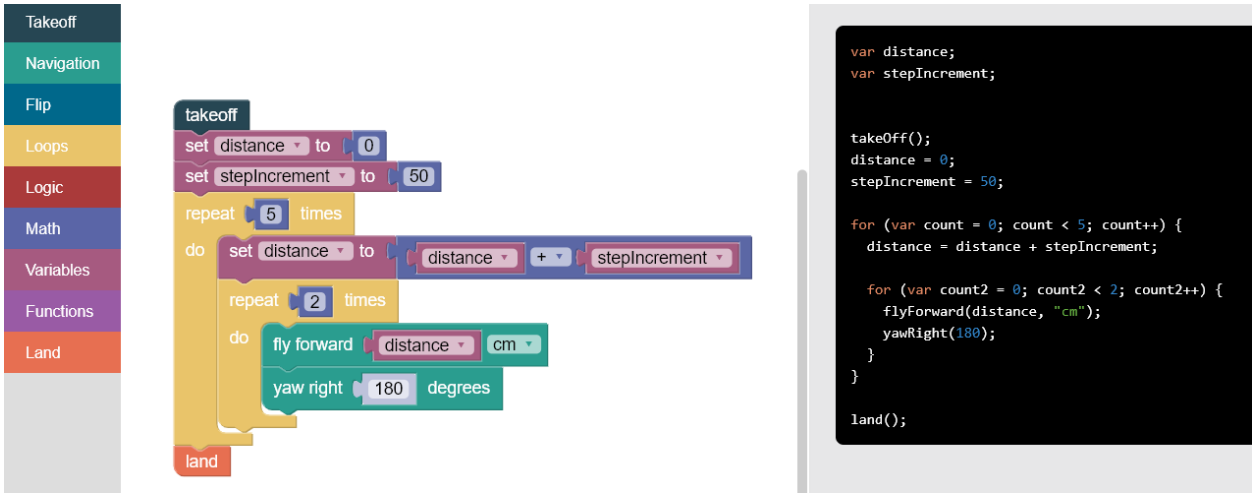
петљи и променљивих, наш код мисије изгледа као на слици 9.

Јасно је да овај програм укључује велики број сувишних блок кодова! Зато је ово сјајна прилика за измену нашег кода. Међутим, постоји проблем, с обзиром на то да су удаљености унутар сваког од блокова „летења напред“ фиксне. Некако мора бити динамичан. То ћемо учинити користећи варијабле. Варијабле нам омогућавају да повећавамо удаљености са сваком петљом и олакшавају управљање кодом. На пример, ако желите да промените код са слике 9 тако да лети у корацима од 100 цм уместо 50, за обављање овог задатка бисте користили варијабле. Ако не користите варијабле, а желите да промените удаљеност, морали бисте да промените у 10 различитих блокова.

Да бисмо уопште могли да користимо варијабле, морамо их прво креирати. Израђују се на картици "Варијабле" и уноси се назив нове променљиве. У овом случају ћемо ставити "удаљености". Након тога правимо другу променљиву која се зове " stepIncrement ". Када направимо ове две променљиве, постављамо их и дајемо им вредности 0 и 50.

Настављамо позивањем једне петље која ће се поновити пет пута. Ту морамо поставити да се вредност променљиве „раздаљина“ математички мења, односно да њена вредност представља збир променљивих „дистанце“ и „stepIncrement“. Збирни блок (и друге математичке операције) могу се пронаћи на картици "Математика". Пошто се беспилотна летелица мора вратити на место поласка, морамо да убацимо још једну петљу која ће се поновити два пута и у којој ће бити тачна упутства о томе шта би беспилотна летелица требало да уради - „лети напред“ и „скрени десно“. За димензију блока "лети напред" треба поставити променљиву "раздаљина". Решење ове мисије дато је на слици 10.

Као што видите, променљиве дозвољавају софтверу да буде флексибилан и омогућавају вам да креирате моћне мисије.

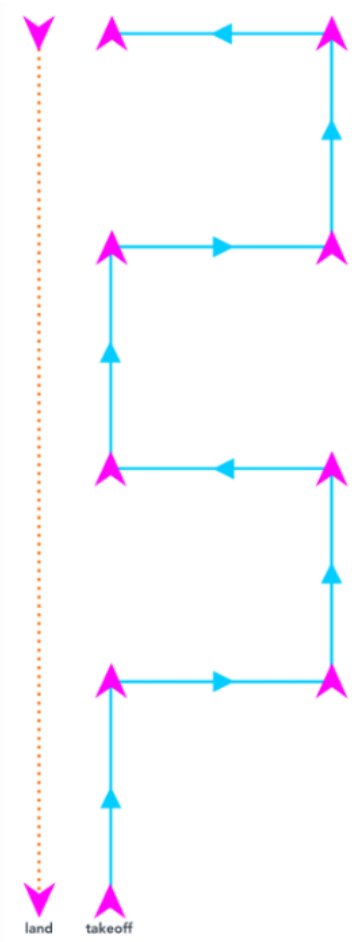


Слика 10. Решење мисије 4

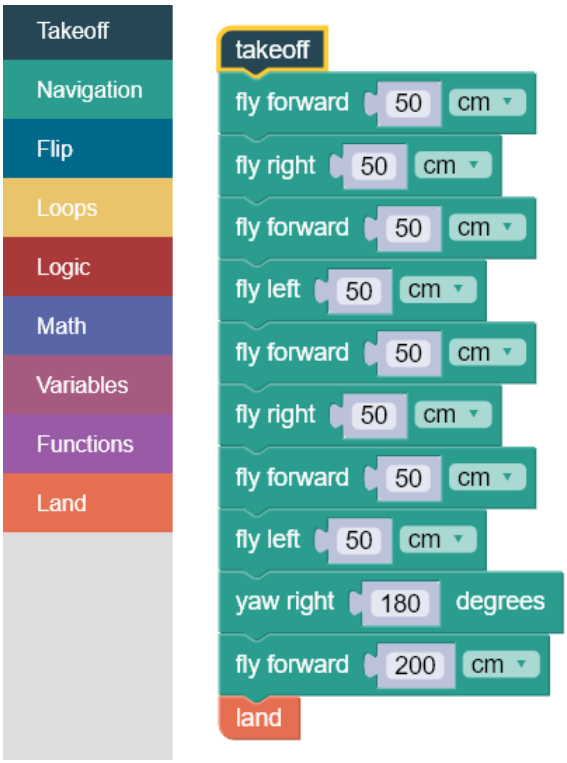
Мисија 5 – IF / ELSE

Ако желимо да програмирамо дрон да следи образац са слике 11. најбоље решење би било коришћење **IF / ELSE** услова. Слика 12 приказује како бисмо то могли учинити без услова и колико недостатака постоји током таквог рада и касније смањене аутономије при промени одређених параметара. Слика 12 има редундантни код који, ако желите да летите дужим обрасцем змије, чини код веома неуредним и тешко управљивим. Ово се може исправити додавањем логике - IF / ELSE.

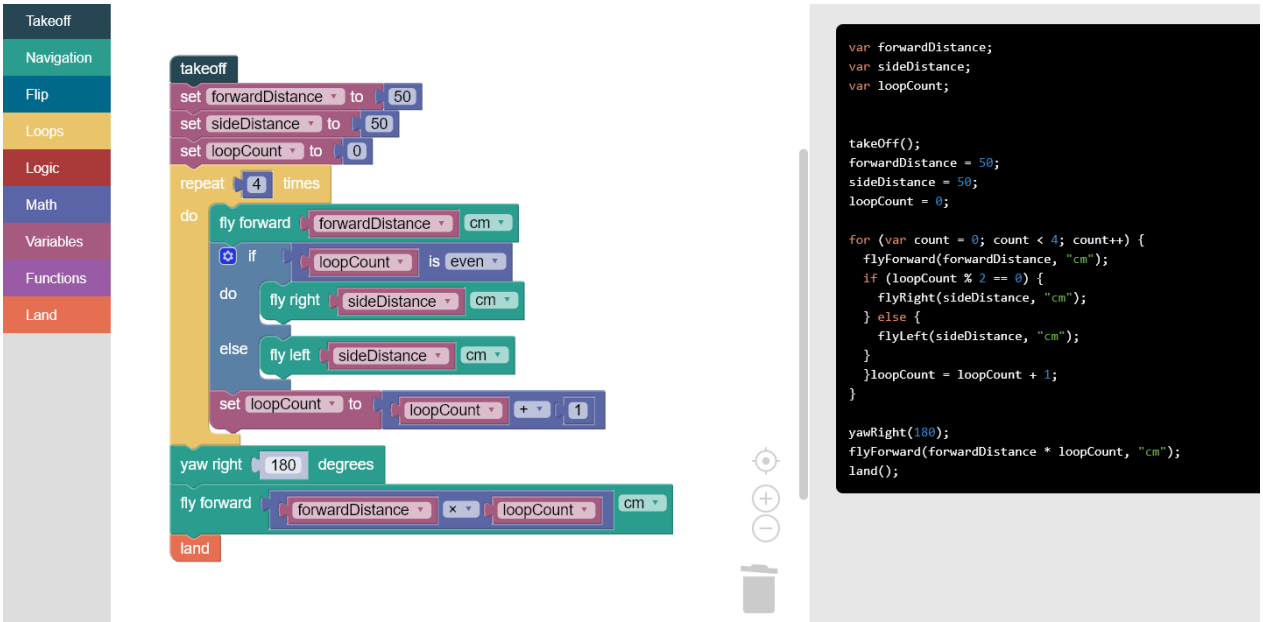
Прво желимо да направимо три променљиве: "forwardDistance", "sideDistance" and "loopCount" (да проверимо да ли је вредност парна или непарна да бисмо утврдили да ли дрон треба да иде лево или десно). Са слике 11 видимо да постоје 4 понављања где дрон иде напред, па у страну, па ћемо направити петљу са четири понављања. У ову петљу ћемо ставити и логички тест (IF DO ELSE), помоћу кога ћемо утврдити да ли дрон треба да иде на леву или десну страну. Пошто смо креирали променљиву "LoopCount" и дали јој вредност 0, проверавамо да ли је парна или непарна. Ако се ради о парном дрону, требало би да иде десно, ако није, дрон ће летети лево. Након тога правимо математички блок који ће вредност променљиве „LoopCount“ повећати за 1. Коначно, потребно је да уз помоћ блока „скретање удесно“ за 180 ° окренемо дрон и вратимо дрон на старт положај. Решење ове мисије можете видети на слици 13.



Слика 11. Шема за мисију 5



Слика 12. Решење мисије 5



Слика 13. Решење мисије 5 преко IF ELSE

SCRATCH

- како инсталирати, пример-

Да бисмо могли да користимо програм Scratch, морамо да следимо следећа упутства:

• Преузмите верзију 2.0

• преузмите и инсталирајте node.js from nodejs.org/en

• преузмите download Tello.js and Tello.s2e from dl-cdn.ryzerobotics.com/downloads/tello/20180222/Scratch.zip

• Отворите Scratch и инсталирајте претходно преузете датотеке притиском на тастер Shift и кликом на File. Затим изаберите Увези експерименталну ХТТП екстензију и изаберите Tello.s2e

Сада морамо да повежемо рачунар са дроном:

• У ЦМД -у отворите директоријум у који сте инсталирали Scratch и када сте у њему откуцајте следећу команду: node Tello.js

• Укључите дрон

• Отворите Ви-Фи везе на рачунару и изаберите Tello мрежа

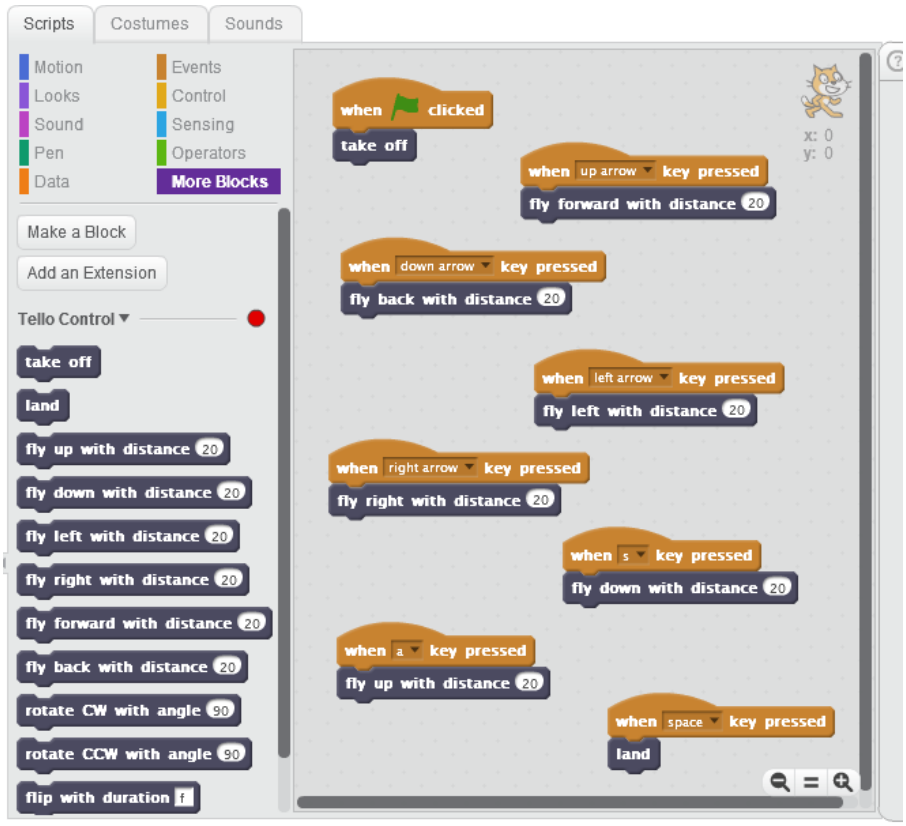
Задатак 1.

Направите програм који ће контролисати кретање дрона:

- Када притиснете тастер са стрелицом нагоре за напред
- Када се притисне тастер са стрелицом надоле за повратак
- Када притиснете тастер са стрелицом налево да бисте отишли улево
- Када притиснете тастер са стрелицом надесно за кретање удесно
- Када притиснете дугме а да бисте прешли нагоре
- Када притиснете тастер с за доле
- Када притиснете тастер ентер за покретање програма
- Када се притисне тастер за размак, дрон слети.

То ћемо учинити на следећи начин:

- Прво ћемо изабрати и превући блокове „када је тастер притиснут“ из менија Догађаји на радну површину и означити жељено дугме
- Затим у мени Више блокова који смо добили пратећи упутства додајемо блокове помоћу којих можемо управљати дроном



Слика 6. Шема за кретање дрона

PYTHON

-О Python, како инсталирати, пример-

PYTHON- OCHOBE

Python је један од најчешће коришћених програмских језика међу програмерима, дизајнерима, инжењерима, наставницима и ученицима. Протеклих година (најмање 15) Python се налази на врху листе програмских језика.

Крајем 80 -их, Гвидо Ван Росум, холандски програмер постао је творац програмског језика Python. Изјавио је да је то био „хоби пројекат“ који га је заокупљао током божићних празника, децембра 1989.

Портрет Гвида Ван Россума у седишту Dropbox 2014. Снимио фотограф Ден Строуд из Сан Франциска, 3. октобра 2016.

Иако његов лого представља две змије, име Пајтон изабрано је због Летећег циркуса Монти Пајтона, чији је обожавалац Пајтонов оснивач.

Python logo, taken from the https://www.python.org/

Претходник Python био је програмски језик АБЦ, који је такође развио Гвидо ван Росум.

Које су главне карактеристике програма Python? Године 1999. Ван Росум је дао предлог који је поднео DARPA -и, назвавши га "Рачунарско програмирање за свакога". У том предлогу су Python-ове највредније карактеристике или „циљеви за Python (као што је речено на Википедији):

1. Лак и интуитиван језик, пођеднако моћан као и велики конкуренти

2. Отворени извор, тако да свако може допринети његовом развоју

3. Код који је разумљив као и обичан енглески

4. Погодност за свакодневне задатке, омогућавајући кратко време развоја

Осим ове четири, Python има и следеће предности:

5. То је интерпретерски језик - свака наредба (линија кода) се извршава једна по једна. Отклањање грешака је лакше него у језицима који се компајлирају.

6. Познато је да Python представља значајну библиотеку и садржи код за различите намене.

7. Код написан једном у Python може се покренути било где, за разлику од неких језика (C, C++).То се зове *Write Once Run Anywhere (WORA).*

8. Python се може проширити на друге језике (C, C++).

9. Python има своје апликације у различитим доменима као што су:

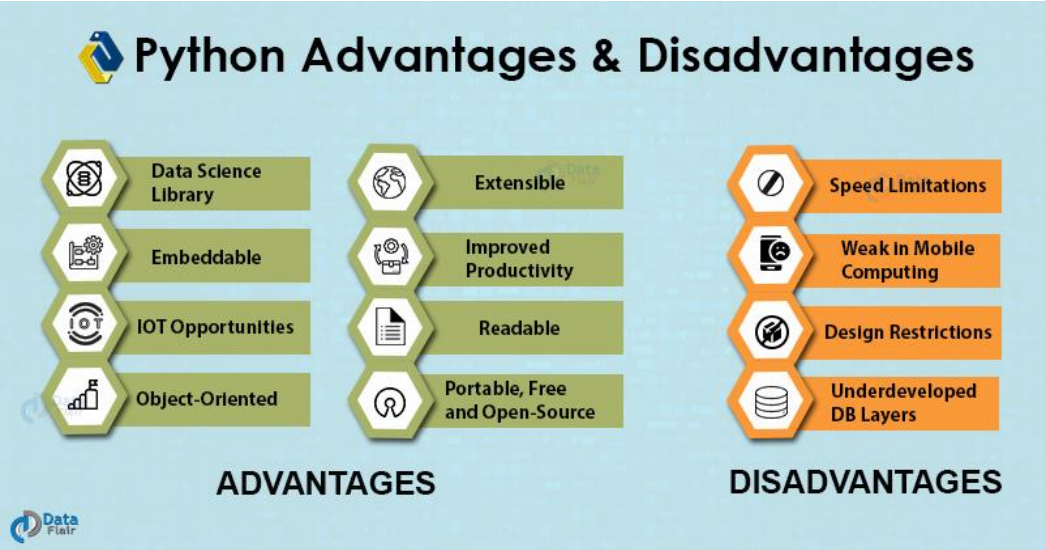
- Веб и развој Интернета - Django, Pyramid, Flask
- Развој софтвера
- Приступ бази података
- Развој игара - PyGame, PyKyra
- Апликације за рачунаре - Kivy, pyqt, pyside
- Образовање
- Наука и нумерика - SciPy, Pandas, IPython, NumPy…

Постоји неколико недостатака које треба напоменути:

1. Пошто се извршава ред по ред, Python често резултира спорим извршавањем.

2. Ретко се користи за имплементацију апликација заснованих на паметним телефонима.

3. Једноставност Python-а понекад може довести до грешака током извођења (ограничења дизајна).



Taken from https://data-flair.training/blogs/advantages-and-disadvantages-of-python/

У неколико речи, Python се може описати као објектно оријентисан, интуитиван програмски језик који је динамички написан. То значи да нема потребе да се приликом писања кода декларише тип варијабле. Укуцајте само један ред у IDLE (Integrated DeveLopment Environment):

```
>>>print ("Hello World")
```

Појавиће се резултат

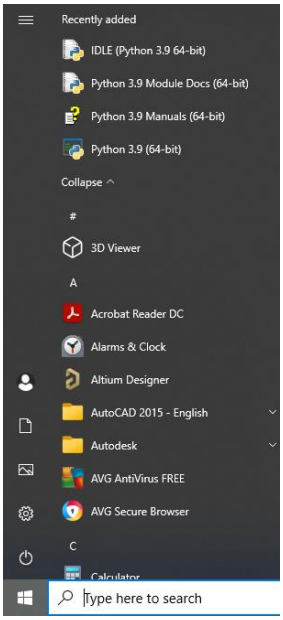
```
>>> Hello world!
```

Python извршава оно што му је речено

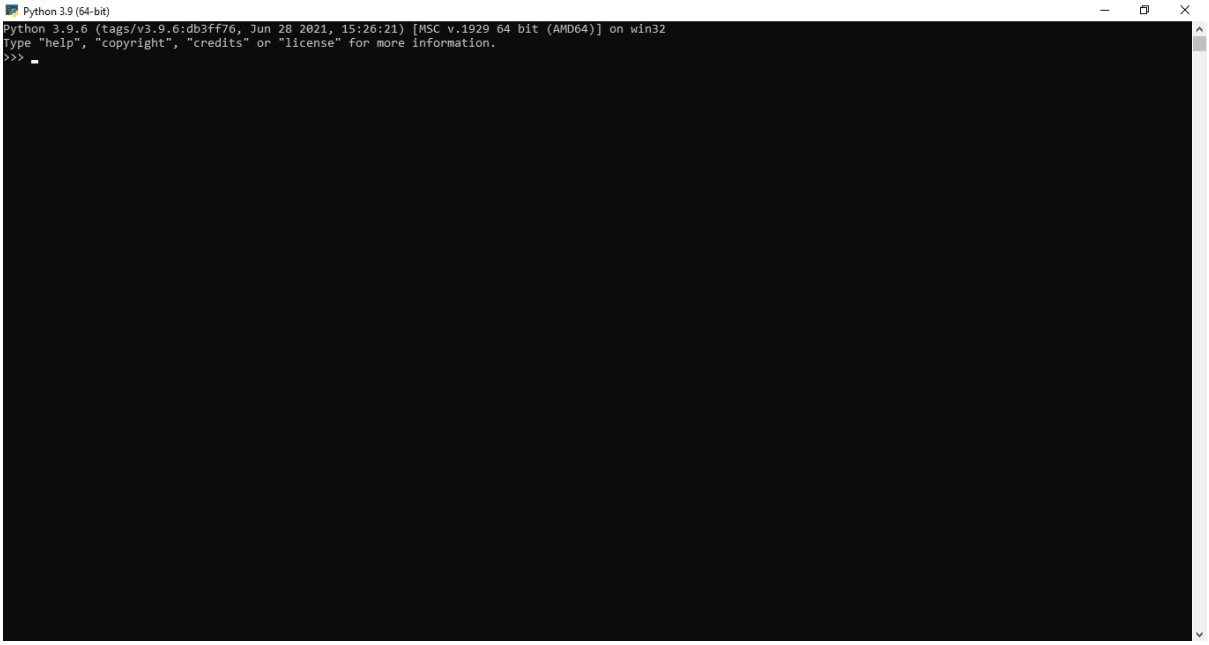
```
>>>2+2
4
```

ПРЕУЗИМАЊЕ И ИНСТАЛАЦИЈА

Званична веб локација на којој се може пронаћи најновије издање Python (верзија) је:
https://www.python.org
Постоје наведена издања за Windows, MacOS…



Најновије издање Python 3 је Python 3.9.6, велико око 25 МБ. Да бисте започели инсталацију, морате кликнути на преузету датотеку и потврдити неколико пута (током инсталације). Инсталирана апликација је наведена у менију Старт.
Покретањем Python 3.9 (64-bit), добијамо апликацију-IDLE (Integrated DeveLopment Environment for Python):



Python за дронове

Програм за дронове који ћемо користити је програмски језик Python. Python је програмски језик који вам омогућава да брзо радите и ефикасније интегришете системе. Да бисмо то урадили, инсталираћемо Python и користити ИДЕ које је интегрисано програмско окружење за писање кода. Прво идемо на python.org у одељку за преузимање где можемо да преузмемо најновију верзију. Затим можемо отићи до одељка за преузимање PyCharm и преузети ИДЕ. Када су Python and PyCharm инсталирани, спремни смо за почетак.

За програмирање Tello дрона потребне су нам библиотеке. Зато их морамо и инсталирати. Потребна нам је библиотека *djitellopy* и библиотека *opencv-python* . Након што увеземо ове библиотеке, можемо почети са кодирањем.

```
from djitellopy import tello
```

Мораћемо да увеземо библиотеку времена. Ово ће нам омогућити да убацимо кашњења између сваке команде.

```
from time import sleep
```

Морамо направити Tello објекат

```
me = tello.Tello()
```

Сада можемо једноставно повезати овај објекат и он ће се побринути за све ИП адресе и све комуникацијске делове уместо вас.

```
me.connect()
```

Направљени објекат нам омогућава да издајемо команде дрону и читамо неку вредност из дрона, на пример ако желимо да знамо какав је статус батерије користимо методу *get_battery ()*

```
print(me.get_battery())
```

Листа метода је доступна ако кликнете на Ctrl, а затим притиснете лево дугме миша. Сада можемо видети све функције које можемо да користимо.

Пре него што започнемо код, морамо да проверимо да ли је наш дрон повезан са WiFi мрежом. Морамо да укључимо дрон. Након тога идите на подешавања wifi и повежите дрон. Када то учинимо, можемо започети наш код.

Сада морамо да видимо како можемо да контролишемо наш дрон. Можемо једноставно написати - само напред

```
me.move_forward(30)
```

Али нећемо моћи да контролишемо брзину, као што смо разговарали у представљању дрона. Може се превести у дрво па желимо то да контролишемо. Написаћемо:

```
me.send_rc_control(left_right_velocity,forward_backward_velocity,up_down_velocity,yaw_velocity)
```

Ова функција има четири броја, прва је лево десно брзина, друга је напред назад брзина, трећа је горе доле брзина и четврта је брзина скретања. Све ове брзине се користе за управљање дроновима и све ове се могу користити од минус 100 до 100. Комбиновањем ових брзина постижемо жељено кретање дрона.

Након издавања наредбе, треба одредити трајање радње коју команда даје, а то радимо позивањем функције спавања са параметром датим у секундама.

```
sleep(2)
```

Пре него што употребимо команду за померање дрона, требало би да наредимо дрону да полети и на крају да издамо команду за слетање. Ове команде су:

```
me.takeoff()
```

and

```
me.land()
```

Дакле, ово су основни покрети које можемо контролисати. Можемо да контролишемо горе и доле, лево и десно, напред и назад и можемо да контролишемо скретање.

Сада ћемо научити како да снимимо слику са нашег дрона. Оно што треба да урадимо је да укључимо stream. Овај ток нам даје све оквире један по један и можемо га обрадити.

```
me.stream_on()
while True:
    img=me.get_frame_read().frame
    img=cv2.resize(img,(360,240))
    cv2.imshow("Image",img)
    cv2.waitKey(1)
```

Задатак 1

Помоћу команди за кретање беспилотних летелица направите програм који демонстрира кретање беспилотних летелица. Нека дрон иде 50 цм напред, па 50 цм лево, па 50 цм назад, па 50 цм десно. Пре слетања, наредите да се иде 10 цм горе, а затим 10 цм доле.

```
from djitellopy import tello
from time import sleep

me = tello.Tello()

me.connect()

me.takeoff()

me.move_forward(50)
sleep(2)
me.move_left(50)
sleep(2)
me.move_back(50)
sleep(2)
me.move_right(50)
sleep(2)
me.move_up(50)
sleep(2)
me.move_down(50)
sleep(2)


me.land()
```

Задатак 2

Измените програм тако да се кретање дрона понавља 5 пута пре слетања.

```
from djitellopy import tello
from time import sleep

me = tello.Tello()

me.connect()

me.takeoff()


def movements():
    me.move_forward(50)
    sleep(2)
    me.move_left(50)
    sleep(2)
    me.move_back(50)
    sleep(2)
    me.move_right(50)
    sleep(2)
    me.move_up(50)
    sleep(2)
    me.move_down(50)
    sleep(2)


for i in range(5):
    movements()

me.land()
```

Задатак 3

Помоћу команде rc креирајте програм тако да се дрон креће у различитим правцима (напред, назад, лево, десно, горе, доле и ротира се). Тестирајте рад ових функција кроз неколико примера.

```
from djitellopy import tello
from time import sleep

me = tello.Tello()

me.connect()

me.takeoff()

me.send_rc_control(50, 0, 0, 0)
sleep(2)
me.send_rc_control(0, 50, 0, 0)
sleep(2)


me.land()
```

Задатак 4

Успоставите везу са камером на дрону и покажите шта она „види“. Оставите дрон да се подигне и окрене 5 пута док шаљете видео.

```
from djitellopy import tello
import cv2

me = tello.Tello()

me.connect()

me.takeoff()

me.streamon()

i = 0
while i < 5:
    me.send_rc_control(0, 0, 0, 50)
    img = me.get_frame_read().frame
    img = cv2.resize(img, (360, 240))
    cv2.imshow("Image", img)
    cv2.waitKey(1)
    i = i + 1

me.land()
```

Задатак 5

Направите програм тако да се дроном управља тастатуром.

```
import KeyPress as kp
from djitellopy import tello
from time import sleep

kp.init()
me = tello.Tello()
me.connect()
me.takeoff()


def getKeyboardInput():
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 50
    if kp.getkey("LEFT"):
        lr = speed
    elif kp.getkey("RIGHT"):
        lr = -speed
    if kp.getkey("UP"):
        fb = speed
    elif kp.getkey("DOWN"):
        fb = -speed
    if kp.getkey("w"):
        ud = speed
    elif kp.getkey("s"):
        ud = -speed
    if kp.getkey("a"):
        yv = speed
    elif kp.getkey("d"):
        yv = -speed
    if kp.getkey("q"): me.land()
    if kp.getkey("i"): me.takeoff()
    return [lr, fb, ud, yv]


while True:
    vals = getKeyboardInput()
    me.send_rc_control(vals[0], vals[1], vals[2], vals[3])
    sleep(0.05)
```

Задатак 6

Измените претходни програм додавањем дугмета које ће сачувати тренутну слику камере у датотеци.

```
import KeyPress as kp
from djitellopy import tello
import time
import cv2

kp.init()
me = tello.Tello()
me.connect()
global img
me.streamon()

me.takeoff()

def getKeyboardInput():
    lr, fb, ud, yv = 0, 0, 0, 0
    speed = 50
    if kp.getkey("LEFT"):
        lr = speed
    elif kp.getkey("RIGHT"):
        lr = -speed
    if kp.getkey("UP"):
        fb = speed
    elif kp.getkey("DOWN"):
        fb = -speed
    if kp.getkey("w"):
        ud = speed
    elif kp.getkey("s"):
        ud = -speed
    if kp.getkey("a"):
        yv = speed
    elif kp.getkey("d"):
        yv = -speed
    if kp.getkey("q"): me.land(); time.sleep(3)
    if kp.getkey("i"): me.takeoff()
    if kp.getkey("z"):
        cv2.imwrite(f'Resources/Images/{time.time()}.jpg', img)
        time.sleep(0.3)
    return [lr, fb, ud, yv]


while True:
    vals = getKeyboardInput()
    me.send_rc_control(vals[0], vals[1], vals[2], vals[3])
    img = me.get_frame_read().frame
    img = cv2.resize(img, (360, 240))
    cv2.imshow("Image", img)
    cv2.waitKey(1)
```