

# Programiranje virtualnog robota za sakupljanje i organiziranje stvari

---

Frlan, Rene

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:948234>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-08-08**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Fakultet informatike i digitalnih tehnologija

Preddiplomski jednopredmetni studij informatike

Rene Frlan

# Programiranje virtualnog robota za sakupljanje i organiziranje stvari

Završni rad

Mentor: izv. prof. dr. sc. Marina Ivašić-Kos

Rijeka, 25. srpnja 2022

# Sadržaj

1. Zadak završnog rada .....	3
2. Sažetak .....	4
3. Uvod .....	5
4. Robotika .....	7
4.1 Veličina .....	7
4.1.1 Nanoboti.....	8
4.1.2 Mikroboti, miliboti i miniboti.....	8
4.1.3 Mali i srednji roboti .....	8
4.1.4Veliki roboti .....	8
4.2 Domene Primjene.....	9
4.2.1 Osobni roboti.....	9
4.2.2 Industrijski roboti .....	9
4.3 Svrha.....	10
4.3.1 Roboti za specifične zadatke.....	10
4.3.2 Roboti za opću namjenu .....	10
4.4 Broj robota .....	11
4.4.1 Pojedinačni roboti .....	11
4.4.2 Roboti u timovima .....	11
4.4.3 Robotika roja .....	12
4.5 Prednosti .....	13
4.6 Mane .....	13
4.7 Povijest .....	14
5. Izrada projekta .....	16
5.1 Cilj robota .....	16
5.2 GearsBot simulator.....	18
5.3 Senzori i aktuatori .....	19
5.3.1 Gyro senzor .....	20
5.3.3 Senzor za lasersku udaljenost.....	22
5.3.4 Senzor za ultrazvučnu udaljenost .....	23

5.3.5	Senzor za detekciju boje .....	24
5.3.6	Aktuator .....	25
5.3.7	Zaokretni pokretač .....	26
5.4	Upravljanje robotom .....	27
5.1	Funkcija za osnovno upravljanje robotom .....	28
5.4.1.1	Korištenje Gyro Senzora u osnovnim pomacima .....	28
5.4.1.1.1	Okret za n stupnjeva .....	29
5.4.1.1.2	Okret na n stupanj .....	30
5.4.1.2	Korištenje senzora za lasersku udaljenost u osnovnim pomacima .....	31
5.4.2	Funkcija za napredno upravljanje robotom .....	31
5.3.2.1	Izračun kuta pomaka .....	33
5.3.2.2	Pomak na x i y koordinatu .....	34
5.3.2.3	Izbjegavanje zapreka .....	36
5.3.2.4	Raspoznavanje zapreke i cilja .....	37
5.4.1	Mapiranje loptica .....	38
5.4.1.1	Predefinirane lokacije mapiranja .....	39
5.4.1.2	Potruga za lopticom .....	39
5.4.1.3	Navigacija do loptice .....	40
5.4.1.4	Memoriranje podataka .....	40
5.4.1.5	Povratak na lokaciju za mapiranje .....	41
5.4.2	Navigacija do loptica .....	41
5.4.3	Podizanje loptice .....	42
5.4.4	Spremanje loptice .....	43
6.	Zaključak .....	45
7.	Literatura .....	47
8.	Popis slika .....	49
9.	Prilozi .....	51

# 1. Zadatak završnog rada



Rijeka, 17. veljače 2022.

## Zadatak za završni rad

Pristupnik: **Rene Frlan**

Naziv završnog rada: **Programiranje virtualnog robota za sakupljanje i organiziranje predmeta**

Naziv završnog rada na eng. jeziku: **Programming a virtual robot to collect and organize objects**

**Sadržaj zadatka:** Proučiti osnove robotike, vrste i karakteristike robota, različita područja primjene s naglaskom na upravljanje robotom. Navesti neke od najčešćih primjena i najpoznatijih vrsta robota te planove i koncepte budućeg razvoja robota.

Istražiti alate za izradu virtualnog robota te osmisliti vlastiti dizajn robota i scenarij primjene koji demonstrira kretanje robota po nekom prostoru, pretraživanje prostora i traženje objekata, te organizaciju i spremanje objekata prema zadanom prioritetu. Napraviti virtualnog robota, mapu po kojoj se robot kreće, objekte na mapi i prepreke te opisati senzore koje robot koristi i njihovu funkciju.

Osmisliti način upravljanja robotom, korištenje motora i senzora za kretanje te aktuatora za manipulaciju s objektima. Opisati razvoj prototipa i ključne algoritme koje su razvijene za upravljanje robotom i za ostvarivanje dodatne funkcionalnosti koja uključuje prioritet objekta prilikom organizacije i slaganja.

Mentor

Izv. prof. dr. sc. Marina Ivašić-Kos

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Zadatak preuzet: 17. veljače 2022.

(potpis pristupnika)

## 2. Sažetak

U ovom radu opisana je izrada programa za upravljanje virtualnim robotom koristeći web stranicu za simulaciju robota GearsBot <sup>1</sup> koji u prostoru pronalazi stvari i slaže ih na za to predviđena mjesta. Detaljno su opisani i objašnjeni svi korišteni hardware-ski i software-ski elementi. Na početku se opisuje hardware, opisuju se svi korišteni senzori i aktuatori koji se nalaze na robotu a koji omogućuju uspješno provođenje zadataka. Što se tiče software-a svi najbitniji procesi podijeljeni su u pod procese. Pod procesi se sastoje od manjih funkcija koje su objašnjene kako i što rade te što im je potrebno za provođenje koda. Osim toga objašnjeni su osnovni termini robotike, klasifikacija robota po četiri glavne metode, prednosti i mane korištenja robota, kao i kratka povijest o robotima i robotici.

**Ključne riječi:** robotika, robot, aktuatori, senzori, upravljanje, Gears, GearsBot

---

<sup>1</sup> <https://gears.aposteriori.com.sg/>

### 3. Uvod

Roboti su se odavno počeli upotrebljavati u industriji, a danas sve više postaju prisutni i u svakidašnjim aktivnostima i privatnom životu.

U posljednja dva desetljeća zbog svoje mogućnosti da brzo, efikasno i precizno obavljaju vrlo kompleksne i ponavljajuće zadatke roboti u industriji postaju nezaobilazni u različitim industrijskim pogonima i proizvodnji kao što su tvornice za sastavljanje automobile, tvornici za punjenje boca, varenje, i ostali slični proizvodni procesi kao i procesi koji su neugodni ili opasni za ljudsko zdravlje kao što je bojanje automobila ili pregled minskih polja ili spašavanje ljudi i stvari iz srušenih zgrada ili sumnjivih prostora koji mogu biti kontaminirani nekim otrovnim plinovima, ili virusima. Roboti se koriste i u skladištima za slaganje robe na police, manipuliranje s robom u skladištima i pripremanje paketa za dostavu.

Zbog sve boljih performansi, i sve manje cijene roboti su postali sve češće prisutni i u kućanstvu. Primjene su različite, od igračaka i virtualnih pomoćnika, do uređaja za zabavu do robota usisavača, ili samohodnih vozila.

Budući da postoji mnogi primjeri korištenja robota, razumno je misliti da postoje i različiti dizajni. Kada bi htjeli vizualizirati robota, svatko on nas imao bi neku svoju ideju robota. Da li je to uređaj koji se kotrlja poput R2-D2 ili hoda uspravno kao C-3PO? Ima li ekspresionističko lice ili mehaničko? Funkcionira li kao usisivač ili bira predmete umjesto vas? Dizajn robota ovisi o njegovoj primjeni, tako da ovisno o poslovima koje će robot obavljati ovisi i kako će biti dizajniran, tj. koje senzore će koristiti, hoće li biti fiksni ili će se moći kretati, hoće li imati aktuatore za manipuliranje sa stvarima, što će svemoći raditi i kako će to obavljati, koliko će biti velik i slično.

Da bismo istražili kako se ti složeni strojevi funkcioniraju, kako se kreću, potrebno je pomno analizirati njihov arhitekturu i algoritme koji upravljaju njihovim ponašanjem. Za demonstraciju može se odabrati robot jednostavne arhitekture koji obavlja jednostavan zadatak. Zadatak ovoga rada je zato prikazati rad virtualnog robota, koji se nalazi u prostoru okružen s nepoznatim brojem loptica i koji za cilj ima skupiti i spremiti sve loptice u za to određene spremnike, unaprijed definiranim redoslijedom. Loptice koje se nalaze na mapi mogu biti u jednoj od tri podržane boje i to zelena, plava i crvena. Loptice je potrebno pokupiti prateći prioritet boja gdje zelena ima najveći prioritet a

crvena najmanji prioritet i zatim spremi u spremnik za odabranu boju. Na kraju izvođenja programa u svakom spremniku smiju biti samo one loptice koje su iste boje kao i spremnik.

Osim objašnjenja korištenih algoritama, procesa i komponenti robota, dan je kraći uvod u pojam robota i robotike općenito. Pojmovi poput klasifikacija robota po četiri glavne metode, pojavljivanje robota kroz povijest pa do ovoga stoljeća, također su neke od tema o kojima se priča u ovome radu. Osim toga navode se neke od najčešćih korištenja i najpoznatijih vrsta robota današnjice kao i neki koncepti robota budućnosti o kojima se tek razgovara i koji se još istražuju.

Zadatak je napravljen uz pomoć besplatnog on-line web stranice pod nazivom GearsBot. Stranica omogućuje konstrukciju virtualnoga robota i korištenje algoritma koji upravlja kreiranim robotom na mapi. Mapa se može prilagođavati da odgovara potrebi bilo kojega zadatka. GearsBot stranica se pokazala kao jednako odličan alat za inicijalno učenje i upoznavanje s upravljanjem i konstrukciju robota na mapi. Isto tako omogućen je razvoj kompleksnih algoritma za verzije robota s velikim brojem senzora i aktuatora, poput ovoga u projektu.



## 4. Robotika

Robotika je grana inženjerstva i znanosti koja istovremeno uključuje elektroniku, strojarstvo i računarstvo (Robotics Introduction, 2022). Robotika se također bavi dizajnom, konstrukcijom i programiranjem sustava za upravljanje robotom, te upravljanjem vrijednostima dobivenih sa senzora kao i obradom informacija koje robot prikuplja.

Roboti se dizajniraju da obavljaju vrlo specifične zadatke. Oni su u direktnom kontaktu s fizičkim svijetom i najčešće se koriste kako bi obavljali zadatke koji su monotoni, ponavljajući, opasni ili osjetljivi. U situacijama gdje roboti zamjenjuju posao ljudi, postoji mogućnost da tim strojevima direktno upravljaju ti isti ljudi koje je robot zamijenio. Isto tako moguće je da robotom upravljaju samo računala i definirani algoritmi.

Roboti se mogu klasificirati na mnogo načina, ali ovdje ćemo navesti četiri glavne metode klasificiranja (Introduction to robotics and automation, 2022):

1. Veličina robota
2. Domene primjene robota
3. Svrha robota
4. Broj robota

### 4.1 Veličina

Prema veličine roboti se mogu klasificirati u jednoj od četiri kategorije:

1. Nanoboti
2. Mikroboti, miliboti i miniboti
3. Mali i srednji roboti
4. Veliki roboti

#### **4.1.1 Nanoboti**

Uređaji se proizvode od Nano materijala i veličine su od 0,1 do 10 mikrometara. Kao usporedbu veličine jedno ljudsko crveno krvno zrnice je oko 5-10 mikrometara. Nanoboti se trenutno u ranoj fazi istraživanja. Trenutno se razrađuju mnogi koncepti o korištenju nanobotova u medicini ali za takvu primjenu potrebno je još vrlo mnogo istraživanja. Jedna od najpopularnijih vizija je da se nanobotovi ubrizgaju u tijelo pacijenta kako bi identificirali i izliječili razne bolesti (International journal of engineering research & technology, 2022).

#### **4.1.2 Mikroboti, miliboti i miniboti**

Ovi uređaji već postoje i učestalo se koriste. Klasificiraju se kao uređaji koji su veličine od 0.1 milimetra do 100 centimetara. Jedan od uređaja koji je u ovoj klasifikaciji i postoji je RoboBee robot koji ima raspon krila od 1,2cm i težinu od 80 miligrama. Krila kada su u pokretu mašu 120 puta u sekundi a robotom se može upravljati daljinski kako bi obavio svoj cilj formiranja letećeg roja za traženje i spašavanje ili umjetno oprašivanje (Introduction to robotics and automation, 2022).

#### **4.1.3 Mali i srednji roboti**

Ovi roboti su uređaji koji su veličine od sto centimetara do osam metara spadaju u ovu klasifikaciju. Sve više kućanskih robota, igračaka kao i društvenih robota poput humanoida i digitalnih osobnih asistenta su neki od primjere ove veličine koji se mogu pronaći u sve više kućanstva (NS Business, 2022).

#### **4.1.4 Veliki roboti**

Uređaji u ovoj klasi su veličine od osam metara pa na više. Neki se koriste u istraživanju i zabavu ali većina robota u ovoj klasi se nalazi u raznim industrijama kako bi omogućila automatizaciju u proizvodnji raznih uređaja, građevini, poljoprivredi, autonomnoj vožnji i sličnim velikim projektima.

## 4.2 Domene Primjene

Robote je moguće kategorizirati prema domeni primjene. U ovoj kategorizaciji raspoznajemo dvije glavne grupe:

1. Osobni roboti
2. Industrijski roboti

### 4.2.1 Osobni roboti

Osobni roboti su uređaji koji se koriste u svakodnevnom životu. Najčešće su napravljeni da budu korisni za osobnu upotrebu. Ovi robot omogućavaju upravljanje i osobama koji nisu upoznate sa programiranjem, dizajniranjem i tehnologijom od koje se sastoji robot. Najčešće se koriste za obavljanje ponavljajućih poslova i dosadnih zadataka kako bi nam uštedjeli vrijeme. Neki od takvih uređaja su digitalni osobni asistenta poput Alexe, Google Home i sličnih asistenata koji omogućuju upravljanje elementima kućanstava, Roomba usisivač koji održava čistoću kuće ili igračka poput Lego Mindstorm robota koji postoje kako bi nas zabavljali.

### 4.2.2 Industrijski roboti

Industrijski roboti su stvoreni kako bi obavljali vrlo specifičan zadatak na unaprijed programiran način. Takvi roboti se koriste za izvođenje zadataka u proizvodnji, građevinarstvu ili poljoprivredi. Kao primjer korištenja mogu biti montaža elemenata, zatezanje vijaka, varenje, bojanje, vizualni pregled itd. Industrijski roboti odlični su u izvođenju jednog specifičnog zadatka. Uvijek su brzi, precizni i pouzdani. Moguće ih je koristiti u industrijama poput:

- Automobilizam
- Elektronika
- Obrada metala
- Plastika i kemikalija
- Izgradnja
- Poljoprivreda
- Skladištenje i logistika
- Pakiranje
- Farmaceutski

Neke od najpopularnijih mega industrija koje se su implementirale ove uređaje u svoje lance proizvodnje su: Tesla, Apple, Boeing i Adidas (Manufacturers that use robotics, 2022).

## **4.3 Svrha**

Svaki robot ima svoju svrhu i zadatak za koji je isprogramiran da obavlja pa se prema tome . roboti mogu klasificirati. Raspoznajemo dvije vrste robota:

1. Roboti za specifične zadatke
2. Roboti za opću namjenu

### **4.3.1 Roboti za specifične zadatke**

Roboti koji su napravljeni za specifične zadatke su strojevi koji obavljaju niz specifičnih zadataka. Može biti jednostavan poput robotske ruke koja pomiče objekte od pozicije A do B, ali može biti i složen poput humanoidni društveni robota s naprednim sučeljem prirodnog jezika, raspoložjima koji se koriste umjetnom inteligencijom poput robota Nadine (AI & Machine learning, 2022). Konstrukcija i ponašanje ovih robota ne mogu se promijeniti, oni slijede unaprijed definirane programe prema svojoj izvornoj namjeni. Roboti za kućanstvo poput robotskih usisivača, čistača prozora, bazena i industrijski automatizirane ruke spadaju među takve strojeve.

### **4.3.2 Roboti za opću namjenu**

Roboti opće namjene nemaju unaprijed definiran zadatak koji obavljaju. Različite komponente robota mogu se kupiti zasebno i te se komponente mogu sastavljati na različite načine kako bi se riješile određene zadaće. Komponente mogu uključivati robotske ruke, kotače, kamere, motore, dodatne senzore i aktuatorne. Ovi roboti također mogu imati bežične veze, kao što su Wi-Fi i Bluetooth. "Mozak" robota koji je općenito malo računalo koje se može programirati za obavljanje različitih zadataka s različitim komponentama pomoću prilagođenih aplikacija napisanih u računalnim programskim jezicima. Uobičajena mala računala koja se mogu programirati, koja se nazivaju i ugrađeni sustavi su Nvidia Jetson i Jetson Nano, Raspberry Pi i Arduino. Ovi ugrađeni sustavi imaju

ulazne i izlazne veze opće namjene (GPIO<sup>2</sup>) na koje se senzori i aktuatori mogu spojiti pomoću standardnog komunikacijskog sučelja.

## **4.4 Broj robota**

Za izvođenje nekih zadataka potrebno je koristiti više robota pa postoji i mogućnost klasifikacije robota prema broju uređaja koji rade zajedno u timu. Postoje tri kategorije:

1. Samostalni roboti
2. Roboti u timovima
3. Robotika roja

### **4.4.1 Samostalni roboti**

Jedan stroj radi sam i ima zadatak koji obavlja prema unaprijed definiranom programu. Program koji upravlja robotom može biti statičan i ne uzimati u obzir okolinu ali i ne mora. Neki programi mogu uključivati naprednu tehnologiju koja mu omogućava prilagodbu okolini. Robot u ovom slučaju može biti spojen na Internet ali i dalje je samo jedan stroj koji obavlja zadatak. Ukoliko na jednoj lokaciji imamo više robota koji obavljaju zadatke sve dok oni međusobno nisu povezani i ne komuniciraju jedni sa drugima ti roboti se klasificiraju kao samostalni roboti.

### **4.4.2 Roboti u timovima**

Poput ljudi za izvođenje pojedinih zadataka potreban je rad u timu. Često je za izvođenje kompleksnijih zadataka potreban rad više robota koji komuniciraju međusobno kako bi uspješno obavili zajednički zadatak. Jedan od popularnijih primjera je sastavljanje automobila. Proces sastavljanja automobila je uvijek isti i ponavljajući tako da postoji tim robota gdje svaki robot ima i obavlja svoj specifičan zadatak koji se mora obaviti kako bi se uspješno obavio zajednički zadatak. Firma ABB je imala želju sastavljanja automobila u 46 sekundi. Kako bi to bilo moguće koristi su 18 robota. Od kojih je 8 robota bilo visokih performansi koji su izvodili procese lijepljenja i sastavljanja

---

<sup>2</sup> General purpose input/output

interijera auta. Kako bi osam visoko perforantnih robota izvodilo zadatke na točnoj poziciji koristili su se 10 robota koji su bili zaduženi za pozicioniranje i varenje karoserije automobila.

#### 4.4.3 Robotika roja

Roj uređaja je vrlo veliki broj jednostavnih robota kojima se upravlja kolaborativno. Ako iz navedenog roja izdvojimo pojedinačni robot njegova vrijednost nije toliko bitna, ali ako je taj robot okružen sa velikim brojem istih robota onda njega vrijednost postaje ključna. Neke od primjena takvih robota su spašavanje, nadzor ali ono najpopularniji i vidljivije su predstave koristeći dronove. Na takvoj atrakciji postoji vrlo veliki broj dronova gdje su oni samo opremljeni sa jednim svjetlom tako da ako izdvojim jedan robot njegova vrijednost i korisnost nestaje ali ako je on okružen sa dronovima poput njega onda je moguće raditi razne elemente u nebu (Slika 1). Iako se već počinje koristiti robotika rojeva je i dalje u ranoj fazi istraživanje.



*Slika 1 otvaranje Olimpijskih igara 2020 u Tokiju koristeći roj robota izvode predstavu.*

## 4.5 Prednosti

Korištenje robota sa sobom donosi mnogo prednosti, počevši sa mogućnosti prikupljanja informacija koje sam čovjek ne može. Ponavljajuće izvođenje zadataka visokom brzinom i efikasnošću i sve to tako dok su svi pokreti i akcije uvijek iste, bez grešaka i mogućnosti da izvedba zadatka pođe krivo. Osim toga većinu industrijskih robota upravlja računalo tj. algoritmi što znači da su oni u stanju obavljati svoje zadatak bez direktne kontrole čovjeka. Sve su ovo prednosti koje se najviše primjenjuju na robote koji se koriste za industrijske svrhe. Takve prednosti u industrijskom okruženju čine proces proizvodnje bržim, efikasnijim, jeftinijim i omogućuju brzu masovnu proizvodnju elementa koji se proizvodi (Robotics Tomorrow, 2022).

## 4.6 Mane

Osim velikih prednosti roboti sa sobom donose jednako i toliko mana. Svaki robot za svoj rad koristi električnu energiju dvadeset četiri sata dnevno. Što je veći robot to su i viši troškovi održavanja koji rastu vrlo brzo. Osim električne energije ovi uređaji često pokreću dosta kontraverzne teme zbog toga što kada se implementiraju u industrijskim lancima zamjenjuju ljudske poslove i smanjuju mogućnost zapošljavanja . Više nisu potrebni radnici da rade poslove koje sada obavlja robot, ali potrebni su ljudi koji će održavati robote kako bi oni mogli raditi neprestano bez ikakvih problema. Osim problema koje donosi *hardware*-sko održavanja i robot ima i svoje *software*-ske probleme. Iako roboti mogu pohraniti velike količine podataka nisu tako učinkoviti kod korištenja i reagiranja na te podatke kao ljudski mozak. Uvijek se prate instrukcije koje su programirane u robot, takve upute najčešće imaju dosta pretpostavki o okolini s kojom će robot morati voditi interakciju. U slučaju da nešto nije idealno ili ne odgovara postavljam pretpostavkama robot najčešće nije u stanju reagirati i prilagoditi svoj rad. Postoji mogućnost da će robot nastaviti pratiti svoje naredbe što može dovesti do štete elementa koji se proizvodi i/ili samoga robota u slučaju da nije sve idealno (The Manufacturer, 2022).

## 4.7 Povijest

Koncept robotike u povijesti prvi put spominje Leonardo da Vinci 1495 koji je po prvi puta dizajnirao stroj nalik čovjeku. Mehanički vitez bio je napravljen da mu se omoguće pokreti sjedenja i stajanja koje bi se izvodile pomoću sajli i kolotura. Povijesno se ne može dokazati dali je da Vinci izgradio robot ili ga je samo dizajnirao (Robotics Basics: Definition, Use, Terms, 2022). Moderni izraz robot potječe iz češke riječi robota („*prisilni rad*“ ili „*kmet*“) riječi i djela opisane i korištene u drami Karla Čapeka 1920 godine. Čapek u predstavi predstavlja robota nalik ljudima kojih bezdušno iskorištavaju vlasnici tvornica. Iskorištavanje robota se događalo sve dok se roboti nisu pobunili i na kraju uništili čovječanstvo. Jesu li bili biološki, poput čudovišta u Frankensteinu (1818.) ili mehanički, nije navedeno, ali mehanička alternativa nadahnula je generacije izumitelja da izgrade električne humanoide.

Osim u dramama riječ „*robotics*“ počinjju koristi i pisci u svojim djelima. Prvi u nizu je Isaac Asimov<sup>3</sup>. Pisac koji po prvi puta spominje naziv robotika u kratkoj priči napisanoj 1942. U toj je priči Issac predložio tri načela o tome kako voditi ove vrste robotskih strojeva. Kasnije su ova tri principa dobila naziv Issacova tri zakona robotike. Ova tri zakona navode da:

1. Roboti nikada neće nauditi ljudskim bićima.
2. Roboti će slijediti upute koje daju ljudi uz kršenje zakona.
3. Roboti će se zaštititi bez kršenja drugih pravila.

Iako nisu humanoidnog oblika, za industriju su razvijeni strojevi s fleksibilnim ponašanjem i nekoliko čovjekolikih fizičkih atributa. Prvi stacionarni industrijski robot bio je programibilni Unimate, elektronički kontrolirana hidraulička ruka za dizanje teških tereta koja je mogla ponavljati proizvoljne sekvence pokreta. Izumio ga je 1954. američki inženjer George Devol, a razvila ga je tvrtka Unimation koju je 1956. osnovao američki inženjer Joseph Engelberger. Godine 1959. prototip Unimate predstavljen je u tvornici za tlačni lijev General Motors Corporation u Trentonu. Godine 1961. Condec Corp. isporučio je GM-ovoj tvornici prvog robota za proizvodnu liniju na svijetu (Robot technology,

---

<sup>3</sup> Američki pisac znanstvene fantastike ruskog podrijetla



2022). Imao je neugodan zadatak (za ljude) uklanjanja i slaganja vrućih metalnih dijelova iz stroja za tlačno lijevanje. Kroz godine nastavlja se razvijanje robota i njihova implementacija u razne industrije. Do kasnih 1980-ih Japan je predvođen robotičkim odjelima bio svjetski lider u proizvodnji i korištenju industrijskih robota. Visoki troškovi rada u Europi na sličan su način potaknuli prihvaćanje zamjena ljudi za robote, pri čemu su instalacije industrijskih robota u Europskoj uniji premašile instalacije u Japanu prvi put 2001. godine (Robot technology, 2022).

## 5. Izrada projekta

### 5.1 Opis robota

U okviru ovog rada napravljen je robot kojem je zadatak skupljanje loptica koje se nalaze na mapi i njihovo pohranjivanje na za to predviđena mjesta. Loptice mogu biti u jednoj od tri predefinirane boje i nalaze se na mapi na slučajno odabranim koordinatama. Broj loptica je unaprijed definiran ali distribucija loptica prema boji na mapi nije fiksna niti ravnomjerna, tako da u nekim slučajevima na mapi može biti više loptica jedne boje nego drugih. Zadatak predstavlja simulaciju realnog problema pretraživanja prostora stvari i njihovo sortiranje na odgovarajuća mjesta.

Da bi zadatak bio složeniji definirani su i prioriteti pospremanja na način da nisu sve boje loptica jednako važne te da se neke loptice trebaju spremati prije drugih po unaprijed definiranom redoslijedu.

Zadatak započinje pokretanjem procesa mapiranja u kojemu se prikupljaju podaci za lopticu, tj. pozicija na kojoj se loptica nalazi (X, Y koordinate) i boja loptice. Ta tri podatka su ključne informacije koje moraju biti dostupne robotu kako bi se uspješno izvršili ostali procesi. Također, važno je predefinirati i redoslijed slaganja loptica i u ovom primjeru prioritet je definiran tako da se prvo slaže zelena, pa plava, pa crvena loptica.

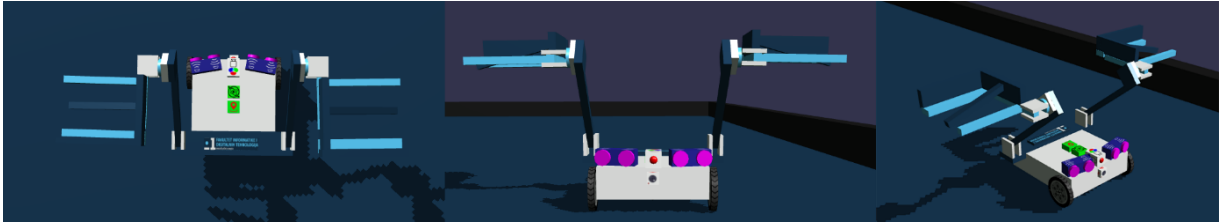
Cijeli proces izvođenja zadatka se može podijeliti na dva pod procesa:

1. Mapiranje
2. Skupljanje i spremanje loptica

Nakon što imamo ključne podatke svih loptica na mapi pokreće se procedura odabira loptice koja će se spremati. Odabir loptica se mora izvesti ispravnim i redefiniranim redoslijedom prateći zadan prioritet boja. Kao rezultat prvoga procesa generira se liste koja se sastoji od ključnih informacija svake loptice. Kada je takva lista dostupna pokreće se drugi proces a to je skupljanje i spremanje loptice.

Kada je robot došao do loptice i podignuo je pokreće se procedura spremanje loptice u odgovarajući spremnik. U ovome procesu robot odlazi od loptice skuplja ju i vraća ju u spremnik koji je iste boje kao i skupljena loptica. Kod skupljanja promatra se prioritet svake od 3 moguće boje. Najprije se skupljaju loptice koje su boje sa najvećim prioritetom (zelena) pa tako redom do boje koja je

najmanjeg prioriteta (crvena). Robot je svoj zadatak obavio kada su sve loptice spremjene u odgovarajuće spremnike. Na slici 2 prikazane je korišteni robot sa svih strana.



*Slika 2 robot koji je korišten u izradi zadatka*

Prostor po kojemu robot kreće napravljen je koristeći online alat za 3D simulaciju robota GearsBot<sup>4</sup>. Osim stvaranja prostora GearsBot je alat pomoću kojega je kreiran robot, svi senzori koji su na njemu i algoritam koji upravlja robotom.

Loptice koje se nalaze na mapi mogu biti na proizvoljnoj lokaciji ali moraju pratiti sljedeća pravila:

1. Sve loptice moraju biti istog oblika i veličine
2. Loptice se ne smiju micati
3. Loptica mora biti obojana u jednu od tri dostupne boje (crvenu, zelenu ili plavu)
4. Između dvije loptice mora biti minimalan razmak od 20 centimetara

---

<sup>4</sup> <https://gears.aposteriori.com.sg/>

## 5.2 GearsBot simulator

Gears ili GearsBot je 3D simulator robotike i podržavanje programiranje u blokovima koristeći Blockly. Osim programira Gears podržava i konverziju blokova u Python kod. Ako ne želimo vršiti konverziju moguće je i direktno programiranje u Pythonu (Ec3dev, pybricks). Generirani Python kod moguće je izvoditi na Gears simuliranom robotu koji replicira ponašanje pravoga robota s malo ili bez ikakvih promjena (GearsBot Home, 2022).

Gears je napravljen da je dovoljno jednostavan za korištenje početnika i omogućuje jednostavno i brzo učenje osnova upravljanja robota u proizvoljnoj okolini. Osim upravljanja moguće je upoznavanje s konstrukcijom robota tako da robot odgovara potrebi nekog zadatka. Neki simulatori pojednostavljaju simulaciju fizike kako bi programiranje početnicima bilo jednostavnije ali Gearsu je primarna publika srednje do napredno educirani ljudi u polju robotike. Pošto Gears pokušava postići što realniju simulaciju postoji mogućnost da robot ima probleme s proklizavanjem guma, problem u putovanju u ravnoj liniji i sličnih dosta kompleksnih problema. Takva visoka realnost simulacije može biti problem početnicima i korisnicima koji nisu upoznati s robotikom a onima iskusnijima i naprednijima omogućava razvoj sposobnosti da unaprijede algoritam tako da ukoliko se pojavi neka netočnost u izvođenju procesa da se algoritam ne prekida nego da na tu nepravilnost reagira i nastavi sa svojim izvođenjem.

## 5.3 Senzori i aktuatori

Kako bi robot bio u mogućnosti obaviti dobiveni zadatak na raspolaganju ima niz senzora i aktuatora. Robot koristi senzore kao ulazne vrijednosti pomoću kojih odlučuje kada i koji pokret ili akciju treba napraviti. Aktuatori se koriste kao izlazna vrijednosti pomoću kojih robot kroz ulazne vrijednosti zaključuje da je spreman za micanje nekih od svojih komponenti to i napravi. Svi senzori imaju zasebne postavke pomoću kojih ih se robot može još više prilagoditi kako bi bolje obavljati potrebni zadatak. Neke od tih postavaka su minimalni, maksimalni kutovi pomaka, brzina pomaka elemenata i domet senzora. Za uspješno upravljanje robota i rješavanje ovog bili su potrebni sljedeći senzori i aktuatori:

1. senzori:
  - a. Jedan Gyro senzor
  - b. Jedan GPS senzora
  - c. Jedan senzor za udaljenost (laser)
  - d. Dva senzora za udaljenost (ultrazvučna)
  - e. Jedan senzor za detekciju boje
2. Aktuatori:
  - a. Četiri aktuatora (po dva na svakoj ruci)
  - b. Dva zaokretna pokretača (po jedan na svakoj ruci)

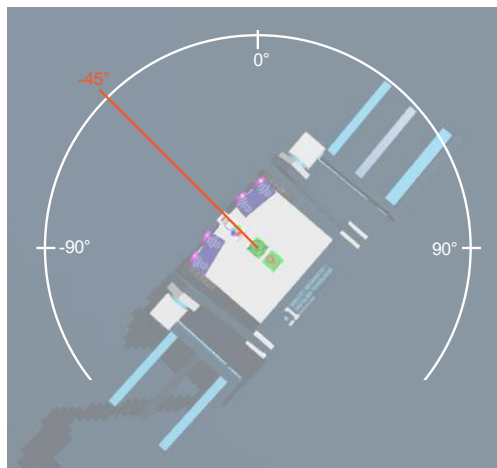
Razlike između virtualnih senzora i onih pravih jest da virtualni senzori uvijek prikazuju 100% točnu vrijednost i nemaju odstupanja. Također ne uzimaju u obzir nepravilnost i ostale utjecaje realnog svijeta koje utječu na senzore u stvarnom prostoru.

### 5.3.1 Gyro senzor



*Slika 3 gyro senzor*

Gyro senzor (Slika 3) vraća vrijednost kuta na kojega je robot okrenut. Na početku simulacije kut na koji je trenutno orijentiran robot postaje početni nulti kut. Ukoliko se robot počine okretati za vrijeme izvođenja bilo kojeg procesa svaki okret prema desno kut se povećava za pozitivnu vrijednost a za svaki pomak prema lijevo kut se smanjuje za negativnu vrijednost (Slika 4). Što znači da vrijednost (kut) koji ovaj senzora vraća može biti pozitivan ili negativan ali isto tako može biti veći od  $+360^\circ$  ili manji od  $-360^\circ$  pošto se za vrijeme izvođenja simulacije početni kut nikada ne resetira.



*Slika 4 vizualizirano korištenje gyro senzora*

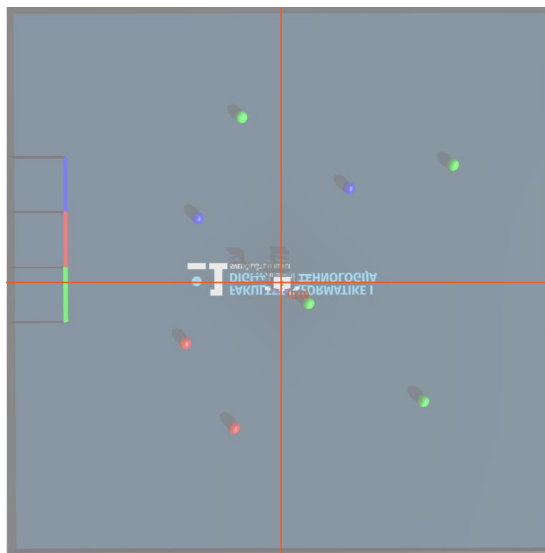
Senzor je ključan za uspješno rotiranje izvođenje procesa vezanih za rotiranje robota oko svoje osi.

### 5.3.2 GPS senzor



Slika 5 GPS senzor

Robot se uvijek nalazi na mapi čiji centar poprima XY koordinatu (0,0) čime se mapa uz pomoć GPS senzora (Slika 5) pretvara u pravokutni koordinatni sustav (Slika 6). Senzor prati x i y koordinatu robota dok se miče po podlozi tj. dok se miče po pravokutnom koordinatnom sustavu. Preko ovoga senzora se uvijek može generirati točka  $T(x_1, y_1)$  koja reprezentira trenutnu lokaciju robota u koordinatnom sustavu tj. na mapi. Na bilo koji pomak robota GPS senzor automatski ažurira sve podatke.



Slika 6 vizualizirano korištenje GPS senzora

Senzor je ključan kod kalkulacija pomaka robota i u procesu mapiranja loptica pomoću njega se izračunavaju koordinate svih loptica. Ovaj senzor u kombinaciji sa gyro senzorom omogućuje primjenu osnovnih pravila trigonometrije. Trigonometrija je temelj na kojemu se osniva izračun podataka koji su nam potrebni za pomak robota. Jedna od vrijednosti koja je ključna i računa se koristeći pravila trigonometrije je udaljenosti koju robot mora prijeći do cilja i lokaciju loptice prilikom mapiranja.

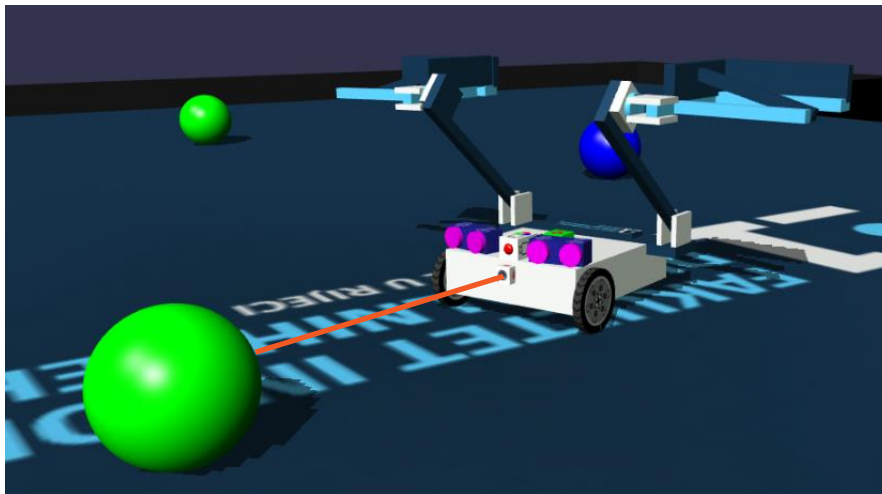
### 5.3.3 Senzor za lasersku udaljenost



*Slika 7 senzor za lasersku udaljenost*

Senzor (Slika 7) mjeri udaljenost između sebe i prvoga objekta od kojega se odbija laserska zraka. Moguće je odrediti domet senzora nakon kojega senzor više nije u mogućnosti detektirati elemente koja je u ovome primjeru postavljena na sto centimetara. Senzor je u mogućnosti pružiti veću preciznost kod izračuna udaljenost zbog toga što koristi samo jednu zraku pomoću koje izračunava udaljenost između sebe i elementa od kojega se ta zraka odbija (slika 8). Senzor ne uzima u obzir kut pod kojim detektira objekt nego odmah vraća udaljenost.

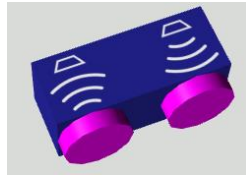
Senzor se koristi u više ključnih procesa izvođenja zadatka. Kod mapiranja koriste se za detekciju optica i precizno navođenje robota prema detektiranoj optici. Osim mapiranja koristi se i kod procesa pomicanja robota i procesa skupljanja optica. U tom slučaju promatra udaljenost robota od cilja kako bi se uspješno provela provjera dali je robot dovoljno blizu za provedbu sljedećeg proces poput skupljanja ili ispuštanja optice.



*Slika 8 vizualizirano korištenje senzora za mjerenje laserske udaljenosti*



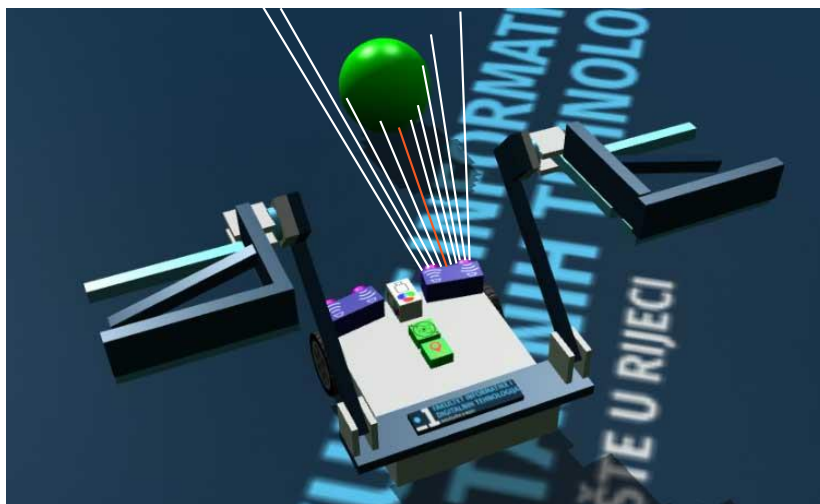
### 5.3.4 Senzor za ultrazvučnu udaljenost



*Slika 9 senzor za ultrazvučnu udaljenost*

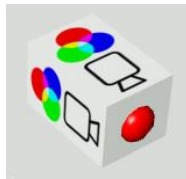
Senzor (Slika 9) radi na principu slanja dvadeset i jedne zrake i to svaku pod različitim kutem. Nakon što pošalje zrake promatra se da li i koja od tih zraka ima najmanju vrijednost. Ukoliko se detektira zraka koja ima manju vrijednost od ostalih, a ta vrijednost je manja od dometa senzora, onda senzor vraća tu vrijednost kao udaljenost koju je detektirao (Slika 10). Prednost ovoga senzora je to što ima mogućnost provjeravanja veće površine za mogućim pronalaskom elementima pošto ne koristi samo jednu zraku za provjeru već njih dvadeset i jednu.

Ovaj senzor ima samo jedan ključni zadatak a to je za detekciju prepreka. Koristi ga algoritam za detekciju prepreka koji promatra vrijednosti udaljenosti koje senzor vraća. Ovisno o dobivenim vrijednostima robot može detektirati dali postoji neka prepreka.



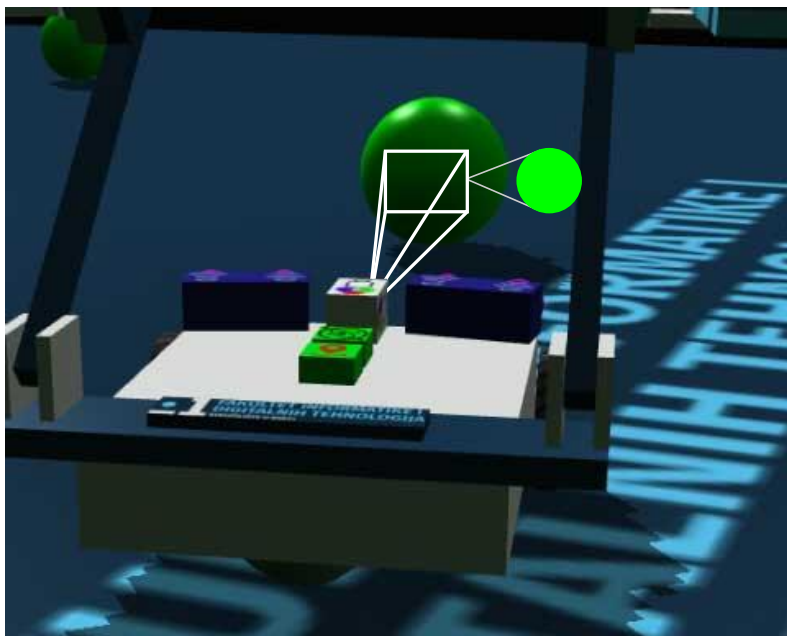
*Slika 10 vizualizirano korištenje senzora za ultrazvučnu udaljenost*

### 5.3.5 Senzor za detekciju boje



*Slika 11 senzor za detekciju boje*

Senzor za detekciju boje (Slika 11) funkcioniра tako da simulira kameru koja napravi sliku prostora ispred sebe. Slika se zatim procesira i određuje se prosječna RGB svih boja koje se nalaze na simuliranoj slici (Slika 12). Postavke senzora omogućuju prilagođavanje dimenziju slike koja uzima kao i domet senzora. Ako se dogodi da na se na slici detektira previše boja i ne može se odrediti jedna konkretna boja koja većinski zauzima simuliranu sliku, tada senzor prikazuje da je detektirana crna boja.



*Slika 12 vizualizirano korištenje senzora za detekciju boje*

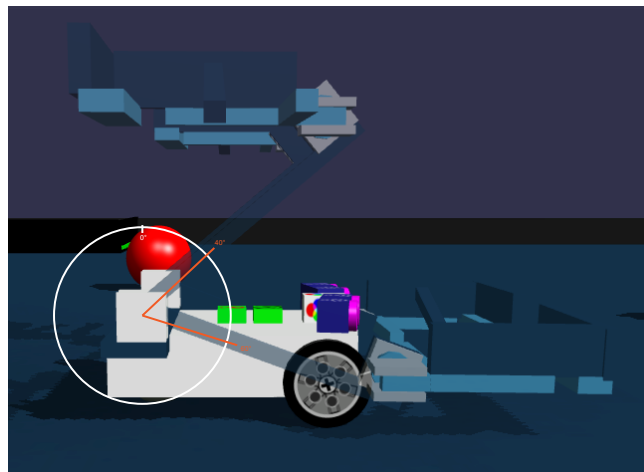
### 5.3.6 Aktuator



*Slika 13 aktuator*

Aktuator (Slika 13) simulira mehanizam koji se pomiče po jednoj osi. Aktuator se može postaviti bilo gdje na simulirani robot a na kraj aktuatora moguće je dodavati proizvoljne elemente ili senzore tako da se gradnja robota može nastaviti. Pomak aktuatora se mjeri u stupnjevima te je u postavkama moguće odrediti početni, završni, minimalni i maksimalni kut na koji se aktuator može pomaknuti.

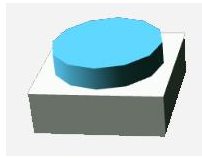
Osim toga moguće je i kontrolirati brzinu kojom se aktuator pomiče na željeni stupanj (Slika 14).



*Slika 14 vizualizirano korištenje aktuatora*

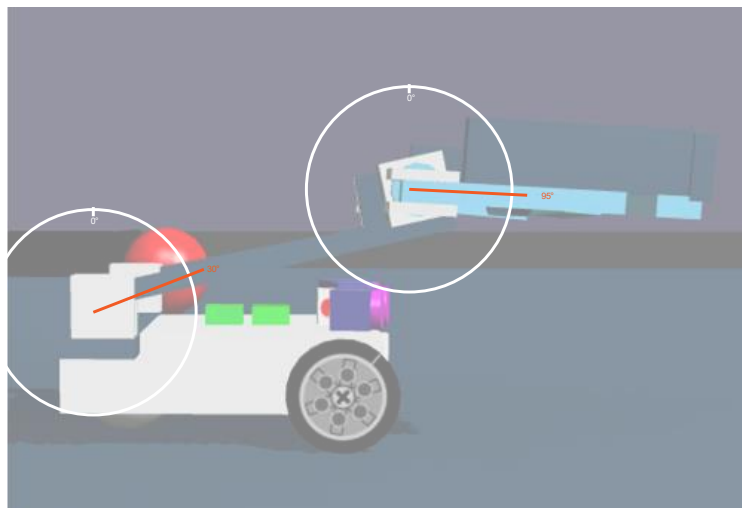
Aktuatori u ovome zadatku omogućuju micanje dviju ruka koje se nalaze na robotu. Micanje je potrebno kako bi se omogućili procesi hvatanja, podizanja i ispuštanja loptica.

### 5.3.7 Zaokretni pokretač



*Slika 15 zaokretni pokretač*

Zaokretni pokretač (Slika 15) simulira okretnu platformu. Bijela baza se može staviti na proizvoljnu lokaciju na robotu i ona ostaje na mjestu dok je plavi dio onaj koji se rotira. Na plavi dio moguće je nastaviti graditi robota te sve komponente koje su spojene na plavi dio rotirati će se kao i pokretač. Prilikom postavljanja određuje se početni kut. Kod korištenja postavlja se kut na koji želimo da se pokretač zaokrene te brzina kojom želimo da dođe do željenog kuta (Slika 16).



*Slika 16 vizualizirano korištenje zaokretnog pokretača*

Zaokretni pokretač se nalazi na rukama robota. Koristi se kako bi održavao vilice koje podižu loptice horizontalno ravne. Tako osiguravamo da su vilice za skupljanje uvijek horizontalno ravne što nam je bitno pri podizanju i pomaku kada je lopta u vilicama.

## 5.4 Upravljanje robotom

Kako bismo mogli izvršiti bilo kakav zadatak potrebni su nam osnovni procesi kojima se može pomicati robot u bilo kojemu smjeru i na bilo koju koordinatu. Za bilo koji pomak moramo imat dvije vrste funkcija kako bi upravljanje robota bilo uspješno. Prvi niz funkcija ima zadatak pomicanja robota i upravljanja robotom koristeći minimalni broj nužnih senzora koji su mu potrebni da obavi pokret i ne obazire se na moguće probleme koje bi taj pokret mogao napraviti u interakciji sa simuliranim svijetom. Drugi niz funkcija ima zadatak pratiti simulirani svijet i mogućnost utjecanja simuliranog svijeta na provođenje pokreta koji prvi niz funkcija radi i ako je potrebno napraviti odgovarajuće korekcije, kako bi se pomak neometano dovršio. Upravljanje robotom se izvršava kroz niz procesa i funkcija. Sve funkcije koje su zadužene za micanje robota po mapi mogu se grupirati u dvije glavne cjeline a to su:

1. Osnovni pomaci

Osnovno pomicanje je niz funkcija koje su odgovorne za jednostavne i temeljne pomake robota. Ove funkcije ne uzimaju u obzir nikakve ulazne senzore i moguće probleme već samo izvršavaju pomak za vrijednost koju dobije na ulazu.

2. Napredni pomaci

Funkcije za napredno upravljanje služe da nadziru sve ulazne vrijednosti sa potrebnih senzora pa onda ovisno o ulazu sa tih senzora kalkulira i ažurira vrijednosti za koje će se robot pomaknuti koristeći funkcije dostupne iz cjeline osnovnih pomaka.

## 5.1 Funkcija za osnovno upravljanje robotom

Funkcije osnovnog pomaku provode procese odgovorne za izvođenje okretanja robota i pomaka robota unaprijed. Za svoje izvođenje ovaj niz funkcija oslanja se na rad dva senzora a to su gyro senzor i senzor za lasersku udaljenost. Pomaci se izvršavaju ovisno o vrijednostima koje dolaze sa navedenih senzora. Pošto su ovo osnovne funkcije one jedino prate da pomaknu robot sa trenutne lokacije na vrijednost koju dobiju na ulazu u funkciju. Vrijednost koja dolazi na ulazu u funkciju i određuje za koliko se robot miče izračunava naprednija funkcija koja uzima u obzir okruženje robota i bilo kakve probleme koji bi se mogli pojaviti kao uzrok pomaka koja ova funkcija provodi.

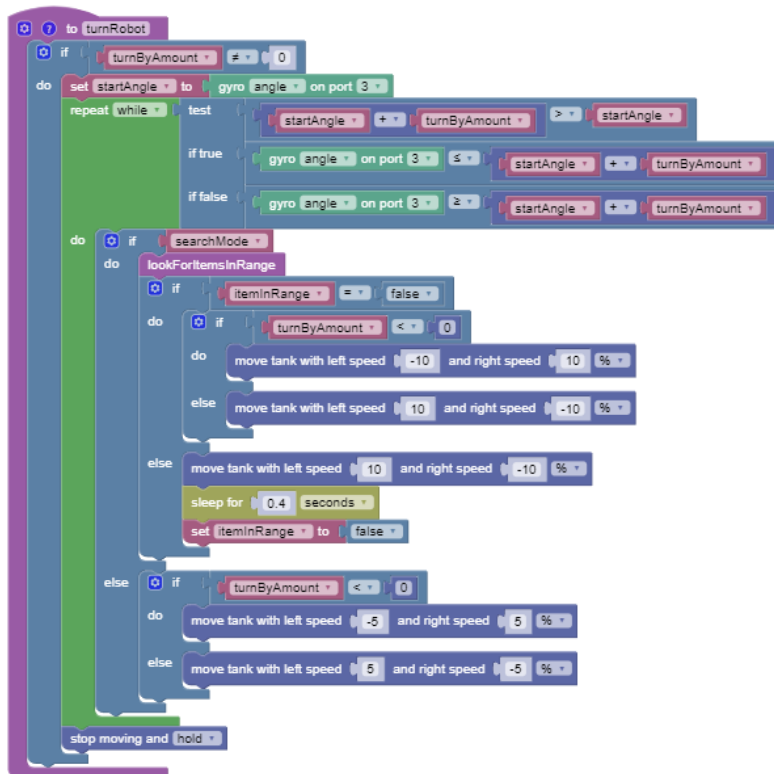
### 5.4.1.1 Korištenje Gyro Senzora u osnovnim pomacima

Kada promatramo Gyro senzor koji se nalazi na robotu raspoznajemo da je njegov jedini zadatak da prati kut i rotaciju robota. Definiramo dva osnovna procesa kojima naprednije funkcije imaju mogućnost okretanja robota:

1. Okret **za** n stupnjeva
2. Okret **na** n stupanj

### 5.4.1.1.1 Okret za $n$ stupnjeva

Ova funkcija (Slika 17) se poziva kada je potrebno robot pomaknuti sa trenutnog stupnja  $\alpha$  za  $n$  stupnjeva u pozitivnom ili negativnom smjeru na njegovu novi kut  $\alpha + n$ .



Slika 17 funkcija za okret robota za  $n$  stupnjeva

Za klasičnu rotaciju robota za  $n$  stupnjeva su nam potrebne sljedeće informacije:

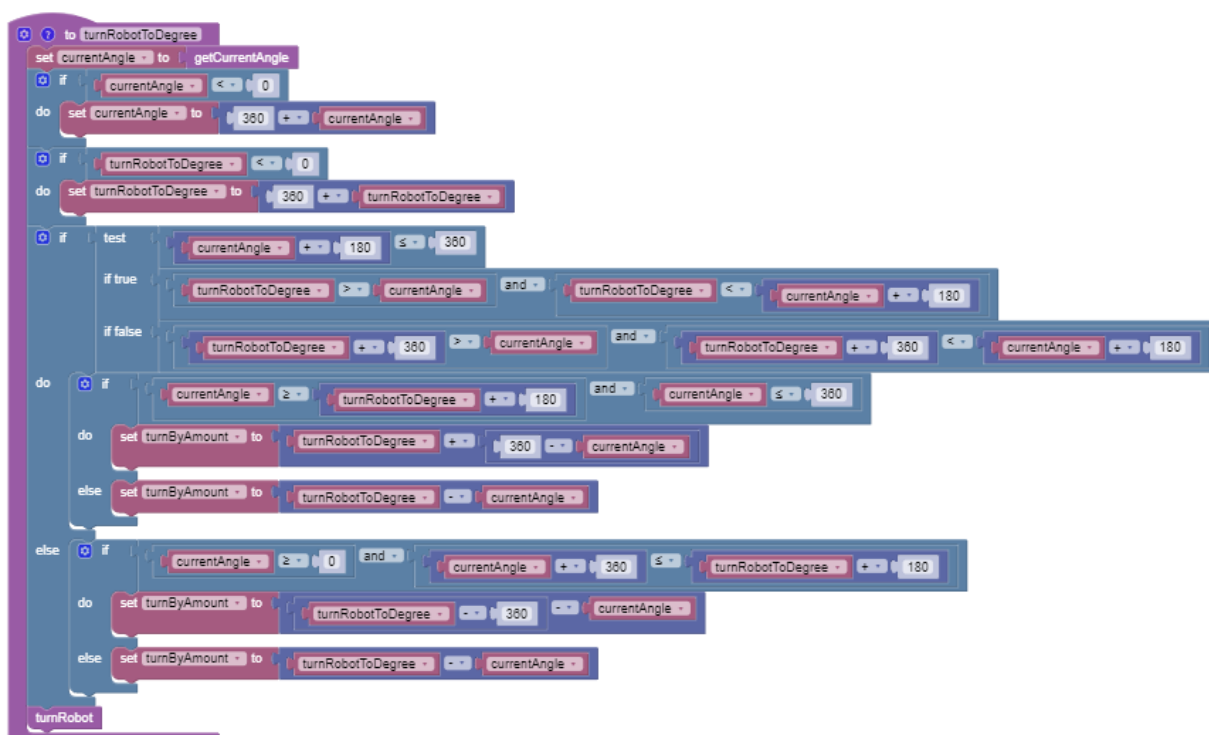
1. Početni kut robota (varijabla: *startAngle*)
2. Kut za koji se robot rotira (varijabla: *turnByAmount*)

Ako funkcija ima te dvije varijable onda je moguće izvršiti zaokretanje robota. Početni kut se dobiva sa gyro senzora i ovisno o tome dali je *turnByAmount* veći ili manji od nule robot se počinje okretati. Robot će provodit proces okretanja sve dok je kut  $\alpha$  sa gyro senzora manji ili jednak od onog kuta na koji se mora okrenuti  $\alpha \leq (\alpha + n)$ . Osim okretanja robota funkcija omogućava i potragu za elementima dok se robot rotira. Funkcija detekcije elementa se omogućava kroz varijablu *searchMode* i koristi se kod mapiranja mape.

### 5.4.1.1.2 Okret na $n$ stupanj

Funkciju (Slika 18) se poziva kada želimo robot okrenuti sa trenutnog stupnja  $\alpha$  na željeni  $\beta$  stupanj gdje je  $\beta$  kut vrijednosti  $0^\circ \geq \beta \leq 360^\circ$ . Kako bi rotacija robota na određeni  $\beta$  stupanj bila uspješna potrebna je sljedeća informacija:

1. Kut na koji se robot mora okrenuti (varijabla *turnRobotToDegree*)



Slika 18 funkcija za izračun rotacije na  $n$  stupnjeva

Funkcija promatra trenutni kut orijentacije  $\alpha$  koristeći gyro senzor i kut  $\beta$  na koji robot mora biti okrenut kada je funkcija okretanja završena. Promatrajući te dvije vrijednosti prvo je potrebno zaključiti u kojem smjeru bi okretanje bilo brže, ono u pozitivnom ili negativnom smjeru. Ovisno o prošloj odluci funkcija kreira pozitivnu ili negativnu varijablu *turnByAmount* i poziva funkciju za osnovno pomicanje robota *turnRobot()* koja je zapravo zadužena za okretanje robota za vrijednost koju je ova funkcija izračunala.



### 5.4.1.2 Korištenje senzora za lasersku udaljenost u osnovnim pomacima

Funkcija (Slika19) ima zadatak micanja robota unaprijed sve dok je udaljenost na senzoru za lasersku udaljenost veća od 4 centimetara. Ova funkcija se koristi u slučaju da x koordinata i y koordinata cilja nije poznata. Jedan takav slučaj je dok se provodi proces mapiranja loptica i robot putuje do svake loptice koja se nalazi na nepoznatoj lokacije sve dok nije mapirana. Osim pomaka dok je funkcija u *searchMode*-u i pronađen je element do kojega se putuje tj. varijabla *itemInRange* je istinita funkcija ima mogućnost provođenja minimalnih korekcija i skretanja tako da osigura da je element kojemu prilazi čim bliže sredini robota.



Slika 19 funkcija za micanje robota u ravnoj liniji

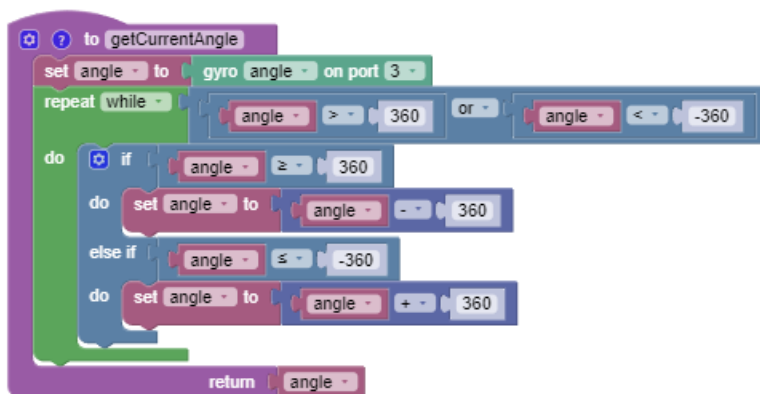
### 5.4.2 Funkcija za napredno upravljanje robotom

Funkcije za napredno pomicanje robota su potrebne za izvršavanje zadataka kompleksnog računanja i stvaranja podataka koji su potrebni za izvođenje funkcija za osnovno pomicanje. Dok funkcije za osnovno pomicanje jednostavno miču robota, funkcije za napredno micanje osiguravaju da su izračunate vrijednosti koje su potrebne za provođenje kompleksnih pokreta uvijek ispravne i

ažurne. I dok se izvodi bilo kakav pomak napredne funkcije provode provjeru dali postoji potreba za prekidom i ili ponovnom kalkulacijom podataka ovisno o tome što se događa u simuliranoj okolini.

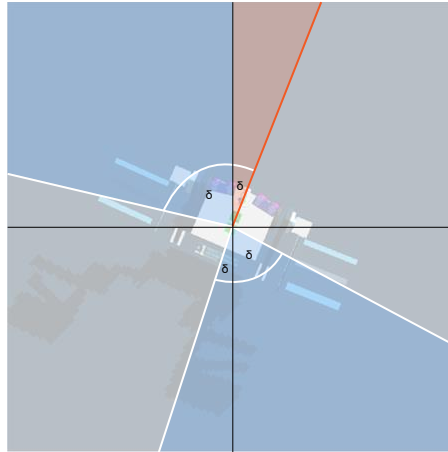
Kako bi vrijednosti senzora bile uniformne kroz sve funkcije postavljena su glavna dva pravila:

1. Za sve kutove  $\delta$  koji se koriste u trigonometriji i ostalim izračunima vrijedi  $-360^\circ \geq \delta \leq 360^\circ$ . Moguće je da gyro senzor vrati vrijednost izvan tog područja. Taj problem rješavamo tako da bilo koja funkcija kojoj je potreban trenutni kut robota ne koristi direktan output gyro senzora nego poziva funkcije  $f$  (Slika20) koja vraća kut  $\delta$  koji je uvijek  $-360^\circ \geq \delta \leq 360^\circ$  :



Slika 20 funkcija za pretvorbu kuta u prihvatljivi oblik

2. Kod kompleksnih izračuna gdje se računa koristeći trigonometrijska pravila sinus i kosinus algoritmi uvijek koriste unutarnji kut  $\delta$  prikazani na slici 21.



Slika 21 vizualizacija unutarnjega kuta

Najkompleksniji proces koji se izvršava je pomaka robota sa trenutne lokacije  $L = (x_1, y_1)$  na neku drugu lokaciju  $T = (x_2, y_2)$ . Takav zadatak se može podijeliti na pet manjih pod procesa:

1. Kalkuliranje kuta na koji se robot mora okrenuti kako bi bio poravnat sa lokacijom  $T$
2. Izračunati udaljenost koju robot mora prijeći da sa lokacije  $L$  stigne do  $T$
3. Pokrenuti micanje robota
4. Dok se robot miče potrebno je konstantno provjeravati za bilo kakve moguće prepreke. Ukoliko se pojavi prepreka potrebno je adekvatno reagirati.
5. U slučaju da je neki element na lokaciji  $T$  njega je potrebno raspoznati kao cilj a ne kao prepreku

### 5.3.2.1 Izračun kuta pomaka

Cilj ove funkcije (Slika 22) je izračunati vrijednost kuta  $\alpha$  ili u funkciji prikazanoj na slici 21 to bi bila vrijednost varijable *angleToTarget* koja reprezentira kut na koji robot mora biti orijentiran tako da se cilj nalazi direktno ispred robota.

```

to calcTargetAngle
  set angleToTarget to atan (getBside / getAside)
  set isObstructed to false
  if (targetXcord < gps x on port Auto and targetYcord > gps y on port Auto)
    do set angleToTarget to 270 + 90 - angleToTarget
  else if (targetXcord > gps x on port Auto and targetYcord < gps y on port Auto)
    do set angleToTarget to 90 + 90 - angleToTarget
  else if (targetXcord < gps x on port Auto and targetYcord < gps y on port Auto)
    do set angleToTarget to 180 + angleToTarget
  set angleToTarget to round angleToTarget
  return angleToTarget

```

Slika 22 funkcija za izračun kuta pomaka

Kako bi funkcija izračunala potrebni kut  $\alpha$  prvo se izračunava unutarnji kut  $\delta$  koristeći

trigonometrijsko pravilo:  $\delta = \arctan\left(\frac{b}{a}\right)$ .

Izračunati kut  $\delta$  je unutarnji kut kojega je potrebno transformirati u potrebni kut  $\alpha$ . Kalkulacije transformacija kuta se izvode ovisno o tome u kojemu kvadrantu je cilj. Rezultat funkcije je kut  $0^\circ \leq \alpha \leq 360^\circ$  na koji se onda robot mora zaokrenuti koristeći osnovnu funkciju *turnRobotToDegree()*

### 5.3.2.2 Pomak na x i y koordinatu

Kada je potrebno robot pomaknuti sa trenutnih koordinata na neku drugu x i y koordinatu poziva se ovaj proces. Kako se ovaj proces poziva nakon funkcije *turnRobotToDegree()* koje je zadužena za postavljanje pravile orijentacije robota kako bi se pomak ispravno izvršio možemo pretpostaviti da je robot ispravno orijentiran i spreman za pomak prema cilju.



Slika 23 prikaz senzora i generiranog trokuta pri pomaku na x i y koordinatu

Kako bi pomak robota na željenu lokaciju bio moguć potrebne su sljedeće informacije:

1. Koordinate cilja

Prvo se izračunavaju odgovarajuće stranice,  $a$  i  $b$  pa zatim koristeći pravilo pravokutnog trokuta gdje vrijedi  $c = \sqrt{a^2 + b^2}$  dobivamo i hipotenuzu (Označeno na slici 23). U stvorenom trokutu stranica  $c$  nam reprezentira udaljenost robota od ciljne lokacije.

Dok se robot miče konstantno se ažuriraju sve vrijednosti stranica kao i zelena zona vidljiva na slici 23. Zelena zona je područje između robota i cilja i proces pomaka na  $x$  i  $y$  koordinatu se smije samo izvoditi u toj zoni. Zelena zona nam koristi kao osiguranje u slučaju da robot ne ide u pravom da se zaustavi i proces pauzira. U slučaju da se proces pauzira tj. robot izađe iz zelen zone kalkulacija svih vrijednosti se ponovno izvodi i robot ponovno pokreće micanje.

Robot će se moći micati samo ako su svi sljedeći uvjeti istiniti:

1. Laserska udaljenost je veća od 3 centimetra
2. Robot se nalazi u zelenoj zoni (zelena podloga na slici 23)

3. Zona za detekciju prepreka je prazna (Svijetlo zeleni rub na slici 23)

Ako je bio koji od ova 3 uvjeta istinit robot staje i poziva adekvatne funkcije koje mogu nastaviti tok programa ili napraviti poziv na funkcije poput zaobilaznja ili ponovne kalkulacije vrijednosti.

### 5.3.2.3 Izbjegavanje zapreka

U isto vrijeme izvođenja procesa micanja robota prema x i y koordinatama cilja provodi se i provjera za izbjegavanje zapreka. Ukoliko je udaljenost robota do cilja veća od 35 centimetara automatski je aktivirana provjera tj. funkcija za izbjegavanje zapreka.

Za detekciju zapreka koriste se dva senzora za ultrazvučnu udaljenost gdje je svaki postavljen na jednu stranu robota i promatra lijevo odnosno desnu stranu za bilo kakve zapreke kojima robot prilazi. Za vrijeme micanja vrijednosti senzora se konstantno provjeravaju i uvijek se uspoređuje koji od dva senzora vraća manju vrijednost. Senzori inicijalno detektiraju objekte na udaljenosti od sto centimetara tako da se mogućnost zapreke detektira relativno rano. Kada se detektira mogućnost neke prepreke tj. jedan od senzora ima manju vrijednost od drugog automatski se ažurira varijabla *obstacleOnLeft*. Navedena varijabla nam je potrebna zbog toga što se proces zaobilaznja prepreke mora izvoditi na suprotnoj strani od one na kojoj je detektirana prepreka. Ako se prepreka nalazi sa lijeve strane robota varijabla *obstacleOnLeft* će imati vrijednost true i robot će tu prepreku zaobilaziti sa desne strane. Varijable *obstacleOnLeft* se ažurira konstantno ali sami proces zaobilaznja se aktivira kada jedan od dva ultrazvučna senzora vraća vrijednost manju od 17 centimetara. 17 centimetara je određeno kao minimalan prostor koji je potreban robotu da uspješno provede proces zaobilaznja oko zapreke.

Proces zaobilaznja se može izvoditi na dva moguća smjera prikazana na slici 24:

1. Zaobilazak u desnom smjeru. Robot prvo skreće za  $+45^\circ$  pa putuje 3 sekunde u tome smjeru nakon čega se ponovno okreće za  $-45^\circ$
2. Zaobilazak u lijevom smjeru. Robot prvo skreće za  $-45^\circ$  pa putuje 3 sekunde u tome smjeru nakon čega se ponovno okreće za  $+45^\circ$

Kada robot uspješno zaobiđe zapreko ponovno se poziva funkcija za micanje robota sa trenutne lokacije na isti cilj na koji je robot inicijalno putovao.



Slika 24 vizualizirano izbjegavanje zapreke

#### 5.3.2.4 Raspoznavanje zapreke i cilja

Kada robot putuje prema odabranom cilju postoji mogućnost da se neki element nalazi na tome cilju i da je on zapravo njegov cilj. U ovome zadatku raspoznajemo dva takva slučaja

1. Odlazak prema odabranoj loptici kako bi se robot pozicionirao za skupljanje odabrane loptice
2. Povratak robota u spremište kako bi pustili lopticu u za to odgovarajuće spremište

Ukoliko je robot bliže od 35 centimetara od svoga cilja funkcije za izbjegavanje zapreka se isključuju čime se zanemaruju ulazi sa ultrazvučnih senzora. Pošto su ultrazvučni senzori onemogućeni robot nastavlja putovati prema svome cilju ali od sada do kraja procesa promatra senzora na lasersku udaljenost. Pošto je robot ispravno orijentiran i udaljenost do cilja mu je manja od 35 centimetara i pratimo pretpostavku da dva elementa ne smiju biti jedan do drugoga senzor za lasersku udaljenost je u mogućnosti procijeniti da li je na cilju neki element. U slučaju da se na cilju nalazi element robot će se nastaviti micati u tom smjeru sve dok je vrijednost sa senzora za lasersku udaljenost veća od 3 centimetara. Kada je vrijednost postane manja od 3 centimetara robot se prestaje micati. U slučaju da na cilju nema nikakvog elementa robot će pratiti svoj GPS senzor i zaustavit će se kada dođe do koordinata cilja.

### 5.4.1 Mapiranje loptica

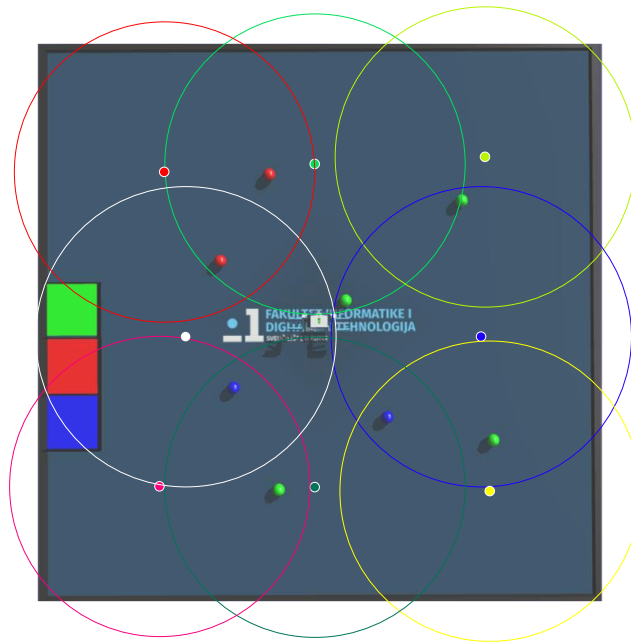
Prvi i osnovni proces koji se poziva nakon pokretanja simulacije je mapiranje loptica (Slika 25). U ovome procesu robot ima zadatak micanja na osam predefiniрани lokacija. Kada robot stigne na jednu od tih lokacija pokreće se kruženje za  $360^\circ$  u kojemu se traže bilo kakvi elementi odnosno loptice koje se nalaze u dometu skeniranja. Provjera postojanja loptice se izvršava uz pomoć senzora na lasersku udaljenost i u koliko on vrati vrijednost koja je manja od dometa raspoznaje se postojanje loptice nakon čega se pokreću pod procesi kako bi se dovršio proces mapiranja loptica. Za izvođenje svih procesa koji dolaze nakon mapiranja loptica, potrebna je lista koja je rezultat ove funkcija. Lista se sastoji od ključnih informacija koje se prikupljaju za vrijeme izvođenja procesa mapiranja a to su:

1. X koordinata svake loptice
2. Y koordinata svake loptice
3. Boja svake loptice
4. Ukupni broj loptica na mapi

Kako bi algoritam generirao sve ključne informacije koje su iznad navedene, poziva se funkcija *mapItems()*. Navedena funkcija odgovorna je za obavljanje procesa mapiranja. Cjelokupni proces mapiranja se može podijeliti na pet pod procesa a to su:

1. Pomak robota na predefiniране lokacije za mapiranje
2. Potraga za lopticom na svakoj lokaciji
3. Putanja do loptica
4. Memoriranje potrebnih informacija
5. Povratak na lokaciju mapiranja i nastavak mapiranja





*Slika 25 vizualizirano memoriranje lokacija loptica*

#### **5.4.1.1 Predefinirane lokacije mapiranja**

Cilj predefinirani lokacija je da se nakon posjećivanja svih lokacija možemo osigurati da je cijela mapa bila unutar dometa od barem jedne lokacije. Da bismo to osigurali na mapi postoji osam predefinirani lokacija na koje robot odlazi i provodi proces mapiranja što osigurava da cijela mapa bude uspješno mapirana. U slučaju da se pronađe loptica rotiranje se pauzira i pokreću se sljedeći procesi. Kada se robot vrati u početnu lokaciju mapiranje se nastavlja sa pauziranog kuta do krajnjeg kuta.

#### **5.4.1.2 Potraga za lopticom**

Kada robot dođe na određenu lokaciju za mapiranje započinje okretanje za 360° dok istovremeno koristeći senzor za lasersku udaljenost i senzor za detekciju boja provjerava da li postoji neki element koji se nalazi unutar zone koju mapira i ako da da li je njegova boja različita od crne. Provjera boje se provodi zbog toga što su zidovi mape crni i njih se ne smije mapirati. Ukoliko postoji

takav element koji je unutar zone detekcija i nije crne boje okretanje se privremeno pauzira i pomoću varijable *itemInRange* se označava da postoji element koji se nalazi u zoni provjere.

#### 5.4.1.3 Navigacija do loptice

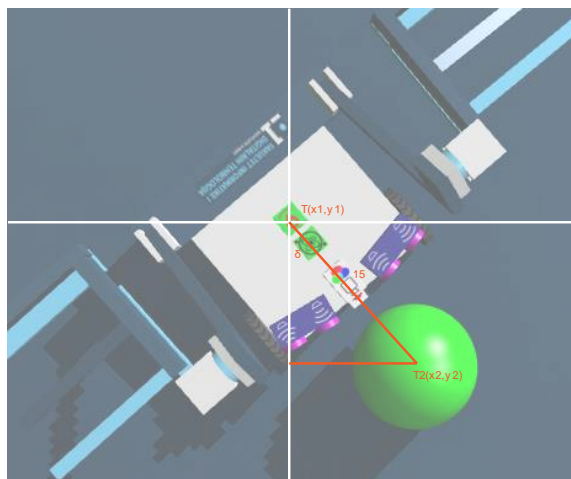
Koristeći osnovne funkcije za pomak robot započinje micanje prema elementu kojega je detektirao da se nalazi unutar zone detekcije. Robot će se kretati u tom smjeru sve dok senzor za mjerenje laserske udaljenosti vraća vrijednost veću od šest centimetara. Kada robot stigne na tu udaljenost robot je spreman i na poziciji za pokretanje sljedećeg procesa provjere i memoriranje podataka loptice.

#### 5.4.1.4 Memoriranje podataka

Kada je robot dovoljno blizu elementa kojega su detektirali prethodni procesi, pokreće se funkcija *logItem()*. Prvo se provode dvije ključne provjere na svakom elementu kojemu se priđe a to su:

1. Da li postoji neki element na lokaciji na kojoj je robot
2. Da li se taj element već nalazi u listi već poznatih tj. memoriranih loptica

Jedino ako su oba dvije provjere lažne pokreće se memoriranje podataka. Prvo se bilježi boja loptice koristeći senzor za detekciju boje. Nakon boje izračunava se točka loptice  $T_2(x_2, y_2)$  lokacija koja reprezentiraju sredinu loptice koji se mapira (Slika 26). Pošto se loptica nalazi ispred robota i potrebne su nam koordinate loptice a ne samoga robota potrebno je izračunati x i y koordinate. Izračun se provodi koristeći osnove trigonometrije koristeći izračunati unutarnji kut  $\delta$  i konstantu hipotenuze od 15 centimetara.



*Slika 26 vizualizirano memoriranje lokacije loptice*

#### **5.4.1.5 Povratak na lokaciju za mapiranje**

Vrijeme micanja robota prema loptici se mjeri u sekundama. Kada se memoriraju sve potrebne informacije i prethodni procesi su gotovi. Robot će se micati prema natrag istom brzinom za isti broj sekundi za koji se micao prema naprijed. Kada se robot vrati na početnu lokaciju proces rotacije se nastavlja sa pozicije na kojoj je bio stao kada je započeo završeni proces mapiranja loptice.

#### **5.4.2 Navigacija do loptica**

Funkcije koje provode navigaciju robota do loptice imaju tri važna zadatka koja obavljaju. Prvi od njih je iz liste svih dostupnih loptica odabrati onu koja ima boju najvišeg prioriteta. Prioriteti boja su unaprijed definirani na način da zelena boja ima najveći prioritet a nakon nje slijedi plava boja pa na kraju crvena boja sa najmanjim prioritetom.

Nakon odabira loptice koja se skuplja kontroliraju se pozivi osnovnim funkcijama za micanje robota koje su odgovorne za dovođenje robota do lokacije loptice prateći vrijednosti koje dobiju na ulazu. Kada je robot na lokaciji odabrane loptice pokreće se izvođenje posljednjeg pod procesa ovoga procesa a to je provedba preciznog poravnavanja robota. Cilj preciznog poravnavanja je da se loptica koju robot planira podignuti mora nalazi direktno ispred u sredini robota. U slučaju da robot prilazi loptici sa nekog idealnog kuta ili kada se loptica mapirala robot nije bio savršeno poravnat sa

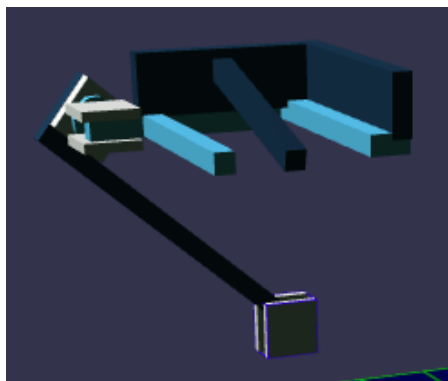
lopticom može se dogoditi da robot nije savršeno poravnat sa njom ali se nalazi blizu loptice. Ukoliko se pokrene proces podizanja loptice a robot nije ispravno poravnat sa lopticom postoji mogućnost da se ruke koje se spuštaju u procesu za podizanje loptice udare i odgurnu lopticu, što je neprihvatljivo.

### 5.4.3 Podizanje loptice

Prethodni procesi su doveli robota na lokaciju na kojoj je savršeno poravnat sa lopticom koju podiže na udaljenosti od dva centimetra od nje. Kada se robot nalazi u toj poziciji moguće je pokrenuti proces podizanja loptice bez da dođe do nekog problema. Za podizanje loptice koriste se dvije ruke koje se nalaze sa svake strane robota.

Svaka ruka (Slika 27) sastoji se od dva aktuatora i jednog zaokretnog pokretača. Proces podizanja loptice se može podijeliti na 4 koraka koji se moraju izvršiti:

1. Spuštanje ruku prema tlu koristeći prvi i veći aktuator. Početni kut ruku je  $+40^\circ$  a nakon završetka zadatka one se spuštaju na  $-60^\circ$
2. Zatvaranje ruku koristeći manji aktuator koji se nalazi na kraju prvoga. Ovaj ima zadatak da lopticu koja se nalazi ispred robota podigne i pozicionira u sredinu „košare“ na rukama.
3. Podizanje ruku na početnih  $+40^\circ$
4. Za vrijeme provođenja 1. i 3. koraka kružni aktuator radi na tome da „košara“ koja se spušta prema zemlji uvijek ostane horizontalno ravna. Ovo je potrebno zbog toga što kada košara stigne do tla ona mora biti ravna kako ne bi došlo do sudaranja sa tlom i prilikom podizanja loptice ona mora ostati ravna u košari kako ne bi ispala van.



*Slika 27 prikaz grupe aktuatora i kružni zaokretala koji podižu lopticu*

#### 5.4.4 Spremanje loptice

Kada je robot u stanju da su mu ruke podignute i u njima se nalazi loptica onda može započeti proces vraćanja loptice u za to prikladno spremište. Na mapi se nalaze tri spremišta za svaku boju po jedno. Lokacije spremišta su definirane kao tri točke  $T = (x_1, y_1)$ . Navede točke su unaprijed definirane na početku izvođenja cijelog programa i nikada se ne mijenjaju (na slici 28 su lokacije označene točkicama ispred spremišta)

Iz liste svih memoriranih loptica prijašnjim funkcija bile su potrebne informacije o lokaciji loptice dok je ovom procesu jedino bitno koje je boje loptica. Pošto već imamo odabranu lopticu koja se nalazi u rukama tako već znamo i koje je ona boje. Svaka loptica se mora spremiti u spremnik te boje i taj proces se može podijeliti na pet pod procesa:

1. Orijentacija i putovanje robota prema spremištu
  2. Orijentacija robota na  $-90^\circ$  kako bi bio poravnat sa spremištem
  3. Posljednji pomak i približavanje spremištu
  4. Puštanje loptice
  5. Odmicanje od spremišta
1. i 2. korak izvode osnovne funkcije za upravljanje robotom gdje je funkcija za spremanje robota jedino zadužena za ažuriranje ciljnih koordinata spremišta. Koordinate spremišta moraju odgovarati boji loptice koja se nalazi u rukama. Kada se robot nalazi neposredno ispred spremišta pokreće se 3. korak koji ponovno koristi osnovnu funkciju za pomak i pomiče robota sve dok je udaljenost između robota i zida spremišta veća od dva centimetara. Ako je robot na udaljenosti od dva centimetara ili manje započinje sljedeći proces a to je ispuštanje loptice. Loptica se ispušta tako da se košara koja je držala lopticu samo otvori čime se i ruke resetiraju na početnu poziciju i spremne su za ponavljanje procesa podizanja loptice. Nakon što je loptica spuštena u spremnik robot se udaljava od spremišta za jednu sekundu kako bi bilo dovoljno prostora između robota i spremnika da robot može neometano provesti okretanje prema sljedećoj loptici i ponoviti proces .



*Slika 28 vizualiziran povratak robota u spremište*

## 6. Zaključak

Mogućnosti korištenja robota u privatnoj i u industrijskoj domeni raste sve više s napretkom softvera koji se koristi za upravljanje robotom, te padom cijena hardverskih komponenti i senzora kao i njihova lakša dostupnost i mogućnost izvođenja ponavljajućih zadataka umjesto nas. Kada se promatra industrijska upotreba sve češće se spominje implementacija robota i automatiziranje procesa kako bi se olakšao i ubrzao proces proizvodnje i izrade proizvoda ili logistički servisi i manipulacija robom u skladištima. Također i kod privatne upotrebe, roboti uspješno izvršavaju jednostavne i ponavljajuće zadatke, sve su dostupniji pa su postali i dio svakodnevne upotrebe. Uređaji poput robotskih usisivača, virtualnih pomoćnika, pametnih uređaja i igračka pojavljuju se u sve više kućanstva.

Robotika je privukla moju pozornost zbog toga što se često spominje kao nova tehnologija i buduće rješenje mnogih problema. Tehničko područje robotike je još relativno novo i ima mnogo prostora za razvoj pa se uvijek može pronaći neka nova tema i korist koja sa sobom nosi nove mogućnosti ali i nedostatke.

Sve je to razlog zbog kojega sam odabrao ovako interesantan zadatak za moj završni projekt. Za vrijeme izrade ovoga projekta suočio sam se sa mnogim pitanjima i problemima kako riješiti neke kompleksnije dijelove zadataka. Ali koristeći znanje iz objektno orijentirano programiranje svaki veći zadatak sam uspješno podijelio na manje pod procese i zasebne funkcije. Svaka od tih funkcija izvodi svoj specifičan zadatak i ne uzima u obzir ono što ostale funkcije rade za to vrijeme. Kod izrade algoritma najkompleksniji procesi za programiranje i uspješno izvođenje su bili proces mapiranje svih optika i proces micanje robota sa jedne koordinate na drugi. Svaki od navedenih procesa imao je tri do četiri iteracije i svakom novom iteracijom proces je postao bolji, brži i autonomniji ali istovremeno i kompleksniji. Za uspješno rješavanje zadatka trebao sam mnogo toga naučiti i o sensorima koji su dostupni za korištenje. Svaki senzor ima svoje mogućnosti ograničenja i postavke koje su morale biti ispravno postavljene kako bi sve radilo.

Svi senzori su vrlo dobro dokumentirani i objašnjeni tako da je odabir senzora i samim time konstrukcija robota bila relativno jednostavna. Najkompleksniji dio konstrukcije robota bio je određivanje i otkrivanje minimalnog broja potrebnih senzora tako da algoritam na ulaznim podacima sa senzora ne prima duplicirane vrijednosti.

Iako robot ispravno radi i dalje postoje zadaci za koje se upravljanje robotom može prilagoditi i unaprijediti. Zbog takvih izazova volio bi nastaviti raditi na algoritmu za upravljanje robota i još ga više unaprijediti. Jedan od takvih mogućih problema je slučaj ako se dvije loptice nalaze jedna pored druge. U takvom slučaju bi mapiranje lokacije loptice trebalo biti puno preciznije i kod skupljanja loptice kut prilaza odabranoj loptici bi mora biti kontroliran kako bi se spriječilo udaranje ruku u lopticu koja se ne skuplja.

Informacije koje sam prikupio i znanje koje sam savladao za vrijeme izrade ovoga projekta o robotici i specifičnim robotima kao i algoritmima koji upravljaju robotima je veliko i svakako mi je otvorilo novu perspektivu sa koje sada gledam na kompleksnost robotike i automatiziranih sistema.



## 7. Literatura

*AI & Machine learning.* (23. Srpnja 2022). Dohvaćeno iz SimpliLearn:

<https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/humanoid-robots>

*GearsBot Home.* (20. Srpnja 2022). Dohvaćeno iz Github: <https://github.com/QuirkyCort/gears/wiki>

*International journal of engineering research & technology.* (20. Srpnja 2022). Dohvaćeno iz Nanobots in Medical Field: <https://www.ijert.org/nanobots-in-medical-field-a-critical-overview#:~:text=In%20association%20with%20medicine%2C%20nanobots,the%20development%20of%20medical%20studies.>

*Introduction to robotics and automation.* (20. Srpnja 2022). Dohvaćeno iz

[https://courses.minnalearn.com/en/courses/emerging-technologies/robotics-and-automation/introduction-to-robotics-and-automation/:](https://courses.minnalearn.com/en/courses/emerging-technologies/robotics-and-automation/introduction-to-robotics-and-automation/)

<https://courses.minnalearn.com/en/courses/emerging-technologies/robotics-and-automation/introduction-to-robotics-and-automation/>

*Manufacturers that use robotics.* (23. Srpnja 2022). Dohvaćeno iz Manufacturing tomorrow:

<https://www.manufacturingtomorrow.com/article/2017/01/manufacturers-that-use-robots/9050/>

Mr.sc. Jasmin Velagić, d. (20. Srpnja 2022). *Uvod u Robotiku.* Dohvaćeno iz Fakultet elektrotehnike

Sarajevo: [https://people.etf.unsa.ba/~jvelagic/laras/dok/Robotika\\_uvod.pdf](https://people.etf.unsa.ba/~jvelagic/laras/dok/Robotika_uvod.pdf)

*NS Business.* (20. Srpnja 2022). Dohvaćeno iz [https://www.ns-](https://www.ns-businesshub.com/technology/household-robots/)

[businesshub.com/technology/household-robots/](https://www.ns-businesshub.com/technology/household-robots/)

*Robot tehnology.* (20. Srpnja 2022). Dohvaćeno iz Britannica:

<https://www.britannica.com/technology/robot-technology>

*Robotics | Introduction.* (20. Srpnja 2022). Dohvaćeno iz Geeks for Geeks:

[https://www.geeksforgeeks.org/robotics-](https://www.geeksforgeeks.org/robotics-introduction/#:~:text=Robotics%20is%20a%20branch%20of,sensory%20feedback%20and%20information%20processing.)

[introduction/#:~:text=Robotics%20is%20a%20branch%20of,sensory%20feedback%20and%20information%20processing.](https://www.geeksforgeeks.org/robotics-introduction/#:~:text=Robotics%20is%20a%20branch%20of,sensory%20feedback%20and%20information%20processing.)

*Robotics Basics: Definition, Use, Terms.* (23. Srpnja 2022). Dohvaćeno iz Infineon:

<https://www.infineon.com/cms/en/discoveries/fundamentals-robotics/>

*Robotics Introduction.* (23. Srpnja 2022). Dohvaćeno iz GeeksForGeeks:

<https://www.geeksforgeeks.org/robotics-introduction/>

*Robotics Tomorrow.* (20. Srpnja 2022). Dohvaćeno iz The Benefits of Automating Production Lines

with Robots: <https://www.roboticstomorrow.com/article/2018/02/the-benefits-of->

automating-production-lines-with-robots/11394/#:~:text=Compared%20to%20humans%2C%20robots%20move,requiring%20both%20speed%20and%20accuracy.

*The Manufacturer*. (20. Srpnja 2022). Dohvaćeno iz The limitations of industrial robots:  
<https://www.themanufacturer.com/articles/the-limitations-of-industrial-robots/>

*Virtual Robots, Brick Emulators, and Simulated Environments*. (10. Siječnja 2022). Dohvaćeno iz LEGO Engineering: <http://www.legoengineering.com/virtual-robotics/>

## 8. Popis slika

Slika 1 otvaranje Olimpijskih igara 2020 u Tokiju koristeći roj robota izvode predstavu.....	12
Slika 2 robot koji je korišten u izradi zadatka .....	17
Slika 3 gyro senzor.....	20
Slika 4 vizualizirano korištenje gyro senzora .....	20
Slika 5 GPS senzor .....	21
Slika 6 vizualizirano korištenje GPS senzora .....	21
Slika 7 senzor za lasersku udaljenost.....	22
Slika 8 vizualizirano korištenje senzora za mjerenje laserske udaljenosti .....	22
Slika 9 senzor za ultrazvučnu udaljenost .....	23
Slika 10 vizualizirano korištenje senzora za ultrazvučnu udaljenost.....	23
Slika 11 senzor za detekciju boje .....	24
Slika 12 vizualizirano korištenje senzora za detekciju boje .....	24
Slika 13 aktuator.....	25
Slika 14 vizualizirano korištenje aktuatora .....	25
Slika 15 zaokretni pokretač .....	26
Slika 16 vizualizirano korištenje zaokretnog pokretača .....	26
Slika 17 funkcija za okret robota za n stupnjeva.....	29
Slika 18 funkcija za izračun rotacije na n stupnjeva .....	30
Slika 19 funkcija za micanje robota u ravnoj liniji.....	31
Slika 20 funkcija za pretvorbu kuta u prihvatljivi oblik .....	32
Slika 21 vizualizacija unutarnjega kuta .....	33

Slika 22 prikaz senzora i generiranog trokuta pri pomaku na x i y koordinatu .....	35
Slika 23 vizualizirano izbjegavanje zapreke .....	37
Slika 24 vizualizirano memoriranje lokacija loptica .....	39
Slika 25 vizualizirano memoriranje lokacije loptice .....	41
Slika 26 prikaz grupe aktuatora i kružni zaokretala koji podižu lopticu .....	42
Slika 27 vizualiziran povratak robota u spremište .....	44

## **9. Prilozi**

- Projekt u digitalnom obliku

- Video materijali prikaza autonomnog kretanja robota i procedure skupljanja loptica prema zadanom prioritetu