

Robotska rekreacija računalne igre „Pac-man“

Završni projekt

Verzija <1.0>

Autor: Viktorija Smlatić

Mentor: doc. dr. sc. Ana Sović Kržić

Sadržaj

1. Uvod	1
1.1. Kratka povijest „Pac-man“-a	1
1.2. Tok igre	2
1.3. Algoritam igre	3
1.3.1 Načini rada duhova	4
1.4 Ideja za završni projekt	5
2. Izrada projekta	7
2.1 Kreiranje mape	7
2.2 Programiranje duhova	8
2.2.1 Verzija 1 – nasumično gibanje duhova	8
2.2.2 Verzija 2 – gibanje duhova po koreografiji	11
3. Tehnička dokumentacija	18
3.1 Tehničke značajke	18
3.1.1 Thymio wireless	18
3.1.2 Robotska rekreacija „Pac-man“ igre	20
3.2 Upute za uporabu	21
3.2.1 Thymio wireless	21
3.2.2 Robotska rekreacija „Pac-man“ igre	30
4. Daljnja optimizacija	32
5. Literatura	34

Slika 1 Izgled karaktera u igri "Pac-man"	1
Slika 2 Izgled igrice.....	2
Slika 3 Ikone dodatnih bodova	2
Slika 4 Izgled duhova i njihova imena	3
Slika 5 Prikaz "Chase" načina rada	4
Slika 6 Prikaz "Scatter" načina rada	5
Slika 7 Izgled duha u "Frightened" načinu rada	5
Slika 8 Prikaz originalnog i prilagođenog FER loga	7
Slika 9 Tlocrt mape s dimenzijama	7
Slika 10 Inicijalizacija koda „verzija1_zid“	9
Slika 11 Glavni dio koda "verzija1_zid"	10
Slika 12 Glavni dio koda "verzija1_linija"	11
Slika 13 Početne pozicije karaktera.....	12
Slika 14 Prikaz gibanja crvenog duha	12
Slika 15 Prikaz gibanja plavog duha	13
Slika 16 Inicijalizacija kodova "verzija2_crveni" i "verzija2_plavi"	15
Slika 17 Sva problematična raskrižja u mapi	15
Slika 18 Dio glavnog programa namijenjen praćenju linije	16
Slika 19 Dio glavnog programa namijenjen koreografiji plavog duha.....	17
Slika 20 Prikaz Thymio wireless robota i njegovih elemenata	19
Slika 21 Ikona aplikacije Thymio Suite	19
Slika 22 Izgled završnog projekta	20
Slika 23 Paljenje/gašenje Thymio robota	21
Slika 24 Punjenje Thymia putem USB kabela	21
Slika 25. Thymiovi načini ponašanja	23
Slika 26 Postavka mijenjanja glasnoće.....	23
Slika 27 Prozor "Wireless Thymio Network Configurator"	24
Slika 28 Početni prozor aplikacije Thymio Suite	25
Slika 29 Spajanje i simuliranje robota na računalu	26
Slika 30 Izgled simulirane mape "thymio.playground"	26
Slika 31 Prikaz sučelja "Aseba Studio"	27
Slika 32 Prikaz spremanja programa u jedan ili više Thymio robota.....	28
Slika 33 Primjer kodiranja daljinskog upravljača s Thymiom	29
Slika 34 Prikaz sastavljanja zidova	30
Slika 35 Konačni izgled mape	30
Slika 36 Optimizacija zidova	32

1. Uvod

1.1. Kratka povijest „Pac-man“-a

„Pac-man“ je arkadna igra koju je razvila japanska tvrtka „Namco“ 1980. godine. Zbog svoje jednostavnosti i privlačnosti ljudima, neovisno o godinama i spolu, dosegla je visoki stupanj popularnosti. Postala je ikona arkadnih igara te je toliko utjecajna, da se često navodi kao „najbolja video igra svih vremena“.

Tvrtka „Namco“ je započela svoju karijeru s prvom razvijenom igrom „Bee Gee“. Ona nije bila uspješna, ali, uz još dva nastavka, učvrstila je temelje tvrtki za daljnji razvoj. Veliki utjecaj u njoj je imao mladi zaposlenik Toru Iwatani, koji je razvio prvu igru „Bee Gee“ i pomogao u razvijanju ostalih.

Iwatani je primijetio problem u tržištu igara tog doba; igre su bile namijenjene muškarcima, zbog sirove grafike i nasilnih tema. Htio je proširiti publiku i razbiti stigmatu vezanu za video igre. Počeo je razmišljati na koji način da privuče žene, te je došao do zaključka da se igra mora temeljiti na hrani i slatkišima. Tako je započeo razvoj igre „Pac-man“, originalno „Pakkuman“. Naziv se temelji na japanskoj onomatopeji „paku paku taberu“, koja upućuje na zvuk kontinuiranog otvaranja i zatvaranja usta.

Igra se razvijala godinu dana i pet mjeseci; tada najviše utrošeno vrijeme za razvoj igre. Razlog toliko dugom vremenu jest što se pridodavala posebna pažnja na izgled karaktera i mape. Cilj je bio da igra izgleda slatko i neopasno, tj. primjerena obiteljima, osobito ženama. Tada novo tehnološko otkriće, RGB zaslon je omogućio uljepšavanje igre raznim, svijetlim bojama, što je zasigurno pomoglo u ispunjavanju cilja. Među posljednjim stvarima su bili dodani zvučni efekti, koji su, zahvaljujući Toshiu Kaiu, postali ikonični zajedno s izgledom karaktera.



Slika 1 Izgled karaktera u igri "Pac-man"

1.2. Tok igre

„Pac-man“ je arkadna igra, koja se odvija na mapi u obliku labirinta. Igrač kontrolira Pac-mana; žutog karaktera koji otvara i zatvara svoja usta. Cilj igre je preživjeti koliko god to igrač može, prije nego što ga duhovi uhvate. Svaki put kad se susretne s duhom, igrač gubi jedan „život“ te kraj igre označava gubitak svih „života“. Za napredovanje, potrebno je „pojesti“ sve točkice. Tada igra prelazi na novi nivo; ubrzavaju se duhovi, njihov algoritam se otežava te je teže preživjeti.




Slika 2 Izgled igrice

Bodovi se skupljaju na više načina. Prvi i najbrži način je putem točkica koje „Pac-man“ jede. Oni nose najmanje bodova; 10 za svaku točkicu. Postoje 4 veće točke, na svakom kutku mape, zvane „energizers“. One prebace duhove u novi način rada, kada ih igrač može pojesti. Za svaki duh dobije $200 \times N$ bodova (N označava broj pojedenih duhova). Treći način nosi najviše bodova. Tijekom igre se periodično pojavljuju ikone, koje se nalaze ispod „kućice duhova“. Postoji 6 različitih ikona: trešnja, jagoda, naranča, jabuka, lubenica, svemirska letjelica, zvono i ključ. Svaka od njih nosi određen broj bodova; npr. trešnja daje 100, a ključ 5000 bodova.



Slika 3 Ikone dodatnih bodova

Postoje 4 duha u igri: „Blinky“ (crveni), „Pinky“ (ružičasti), „Inky“ (plavi) i „Clyde“ (narančasti). Svaki od njih ima svoju osobnost, tj. AI koji koriste za hvatanje „Pac-man“-a. „Blinky“ direktno hvata igrača; „Pinky“ ga pokušava uhvatiti u zamku postavljajući se ispred njega. „Inky“ ima poseban algoritam koji ga čini nepredvidivim, uglavnom patrolira mapom, a „Clyde“ ima nasumične pokrete, zbog kojih se čini da većinom izbjegava igrača. Različitost u ponašanju ovih duhova čine igru zanimljivom i izazovnom.



CHARACTER	NICKNAME
- SHADOW	"BLINKY"
- SPEEDY	"PINKY"
- BASHFUL	"INKY"
- POKEY	"CLYDE"

Slika 4 Izgled duhova i njihova imena

1.3. Algoritam igre

Prostor igre je podijeljen na kvadratasta polja, koja predstavljaju temeljnu jedinicu za algoritam igre. Svi aspekti igre koriste neku vrstu kalkulacije s njima.

Počet ćemo prvo s prostorom u kojem će se igra odvijati. Kreirana je mapa u obliku labirinta tako da su njegovi zidovi „zabranjena“, a slobodni prostor „dopuštena“ polja, u kojima se karakteri i igrač mogu gibati. Također, u svakom „dopuštenom“ polju je smještena točkica koju „Pac-man“ mora pojesti, a na polju ispod „kućice“ duhova, pojavljuju se dodatni bodovi.

Već znamo da svaki duh ima svoju osobnost, tj. strategiju hvatanja igrača. Njihovo ponašanje se svodi na 3 načina rada: „Chase“ („Hvataj“), „Scatter“ („Rasprši“) i „Frightened“ („Uplašen“). Svaki duh ima vlastiti AI za „Chase“ i „Scatter“, ali ne i za „Frightened“. Njihove algoritme ćemo detaljnije objasniti kasnije.

Na samom su početku igre duhovi smješteni u svojoj „kućici“, osim „Blinky“-a. Kako „Pac-man“ počinje jesti točkice, tako oni izlaze jedan po jedan. Uvijek izlaze istim redoslijedom: „Pinky“, „Inky“ pa „Clyde“. Trenutak kada će koji duh izaći određen je količinom pojedenih točkica. Kako igrač napreduje kroz nivoe, tako oni sve brže izlaze. Da bi se bolje raspršili po mapi, postoje valovi mijenjanja načina rada. Za svaki nivo oni se mijenjaju,

kako bi otežali igraču igru. Trajanje načina rada u tim valovima su predstavljeni na tablici 1 ispod.

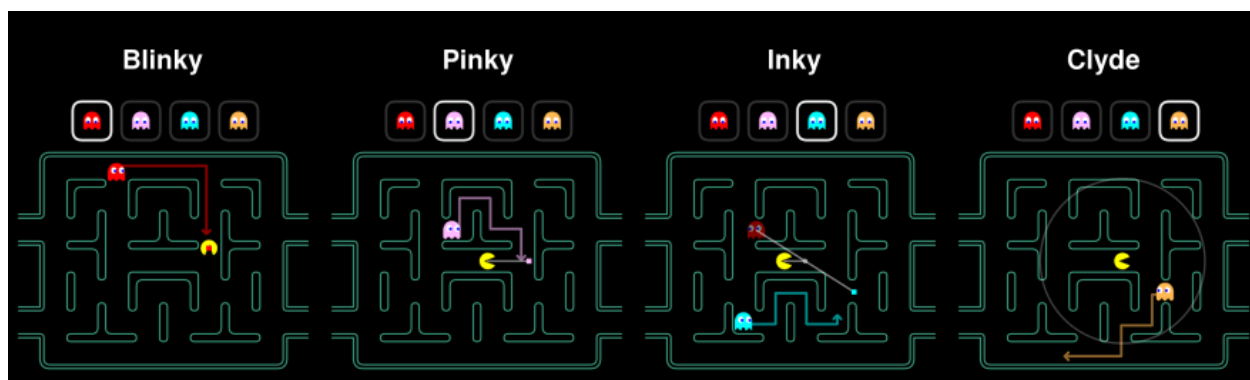
1.	„Scatter“	7 s	„Chase“	20 s
2.	„Scatter“	7 s	„Chase“	20 s
3.	„Scatter“	5 s	„Chase“	20 s
4.	„Scatter“	5 s	„Chase“	∞

Tablica 1 Valovi i trajanja načina rada u prvom nivou igre

1.3.1 Načini rada duhova

„CHASE“ („HVATAJ“) NAČIN RADA

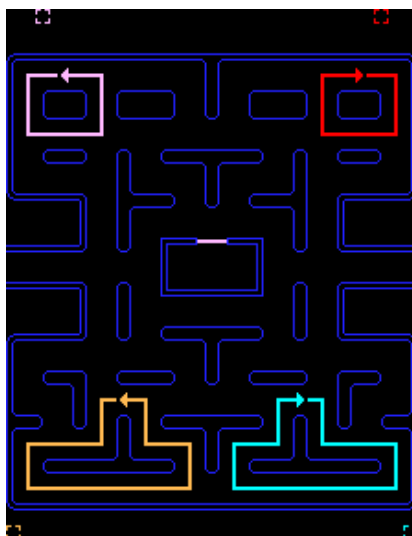
U ovom načinu rada duhovi pokušavaju uhvatiti igrača. „Blinky“ ima najjednostavniji AI; odlazi na polje u kojem se nalazi „Pac-man“. „Pinky“ će odabrati onu rutu koja ga dovodi 4 polja ispred „Pac-man“-a i tako potencijalno uhvatiti igrača u zamku. „Inky“ ima najkompleksniji AI; koristi „Blinky“-evu i „Pac-man“-ovu poziciju. Izračuna vektor koji počinje od „Blinky“-evog polja, a završava 2 polja ispred „Pac-mana“. Vektor se oda produlji u duljini 2 puta te onda njegov završetak označava polje na koje će „Inky“ otići. „Clyde“ koristi „if-else“ princip za gibanje. Ako se igrač nalazi u radijusu 8 polja od njega, prebacit će se u „Blinky“-ev „Chase“ način rada, a inače u vlastiti „Scatter“. Na slici 5 prikazan je ovaj način rada za svakog duha.



Slika 5 Prikaz "Chase" načina rada

„SCATTER“ („RASPRŠI“) NAČIN RADA

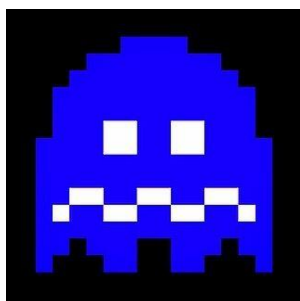
Kao što samo ime govori, cilj ovog načina rada jest raspršivanje duhova po mapi. Oni se više ne fokusiraju na hvatanje „Pac-man“-a, nego svaki od njih odlazi u dotične kornere i kruže oko njih. Na slici 6 vidljivo je koji duh kruži oko kojeg kornera.



Slika 6 Prikaz "Scatter" načina rada

„FRIGHTENED“ („UPLAŠEN“) NAČIN RADA

Duhovi se prebacuju u ovaj način rada kada „Pac-man“ pojede „energizer“. Promjene svoj izgled (slika 7) i počnu lutati mapom. Tada ih igrač može pojesti za dodatne bodove. Kada se to dogodi, vraćaju se u svoju kućicu na nekoliko sekundi te onda opet izlaze kao u početku igre. Ovaj način rada traje nekoliko sekundi te se kraj označava bljeskanjem duhova. Za svaki se idući nivo vrijeme trajanja smanjuje, sve dok „energizer“ više nema svrhe.



Slika 7 Izgled duha u "Frightened" načinu rada

1.4 Ideja za završni projekt

U ovom završnom projektu ću pokušati rekreirati igricu „Pac-man“ koristeći Thymio robote i mikrokontroler Arduino Uno. Budući da točna rekreacija nije moguća zbog financijskog i vremenskog ograničenja, pokušat ću prilagoditi igricu ovim uvjetima, a da se njezin „čar“ ne izgubi.

Na raspolaganju imamo 3 Thymio robota; jedan će predstavljati „Pac-man“-a, a ostali duhove. Igrač će kontrolirati „Pac-man“-a putem daljinskog upravljača, a duhovi će imati poseban AI, pomoću kojeg će patrolirati mapom i hvatati igrača. Zbog dodatnih problema, npr. sudaranja robota, postojat će 2 verzije tog AI-a, koje će biti prokomentirani. Zbog mogućnosti koje Thymio roboti nude, možemo otprilike rekreirati ikonični izgled karaktera i reproducirati glazbu originalne igrice.

Prostor odvijanja igrice će biti smanjen i promijenjen; i dalje će biti labirint, no u obliku FER-ovog loga. Na ovaj način je olakšano kreiranje mape, a da i dalje ostane nekakva razina izazova pri igranju.

Mikrokontroler će služiti kao registriranje te prikaz bodova i života. Na određenim dijelovima mape će biti postavljene LED svjetiljke kao prikaz točkica i ultrazvučni senzori, sve spojeni s mikrokontrolerom, koji će promatrati je li igrač prošao preko njih. Na malom ekranu, također spojeni s mikrokontrolerom, će se prikazivati stanje bodova i preostalih života.

2.2 Programiranje duhova

Ovaj dio projekta ima mnogo problema. Naime, računalo nudi niz mogućnosti, koji nisu lako izvedivi u stvarnom svijetu. Karakteri mogu ići jedno kroz drugo, teleportirati se s jednog mjesta na drugo i imati jako kompleksne algoritme za rad. Iako se ovi problemi mogu nekako popraviti, to zahtjeva jako mnogo materijala i novaca, zbog čega se sam projekt vrlo zakomplicira i postaje neprivlačan. Također, ta rješenja neće davati iste rezultate kakve očekujemo. Stoga, potrebno je prilagođavati tok igre, a da i dalje ostane njen prepoznatljiv „čar“.

Za razvoj algoritma svakog duha, postoje 2 ideje. Prva se svodi na nasumično gibanje, uz komunikaciju, a druga na unaprijed određenu koreografiju, bez komunikacije s drugim robotima. Ideje nadopunjuju jedna drugu; problemi prve su riješeni u drugoj ideji i obrnuto. Na kraju ću prokomentirati njihove performanse te objasniti odabir konačne verzije za uporabu.

2.2.1 Verzija 1 – nasumično gibanje duhova

Verzija 1 temelji se na jednostavnom algoritmu, u kojem roboti prate hodnik te na raskrižju biraju put kojem će otići. Oba duha će imati isti kod, jer je dosta teško i skupo izraditi posebno ponašanje kao u pravoj igri.

Prvi dio ovog algoritma čini praćenje hodnika. Robot se ne smije zabijati u zid, te na raskrižju moraju izabrati neki smjer. Budući da Thymio ima 7 infracrvenih senzora blizine i 2 svjetlosna senzora, prvi pokušaj realizacije se temeljio na zaobilaznje prepreka. Ovaj kod ćemo nazvati „verzija1_zid“.

Gledajući sliku 10, objasniti ćemo dijelove koda koji služe za inicijalizaciju. Na samom vrhu vidimo blok „varijable“, u kojem definiramo nove varijable za kasnije korištenje. Također, iako nije vidljivo na slici, potrebno je definirati novu konstantu „GRANICA“ i postaviti njenu vrijednost na 1500. Ovaj broj je dobiven eksperimentalno, gledajući kako Thymio reagira na širinu hodnika od 15 cm. Nadalje, dolazimo do bloka „funkcije“, u kojem definiramo funkcije gibanja (kreni, stani, okret ulijevo i okret udesno). Ovaj postupak čini kod čitljivijim, budući da će se ove funkcije često koristiti. Zatim dolazi blok „inicijalizacija“ u kojem palimo ledice za izgled robota te postavljamo brzinu robota na 0. Blok „pokretanje i zaustavljanje programa“, kako samo ime kaže, olakšava korištenje programa. Na ovaj način možemo pokrenuti program na dodir gornjeg i zaustaviti ga na dodir donjeg gumba.

```
1 #varijable-----
2 var minimum          | #|
3 var maksimum          | #| namijenjeno za varijablu svjetlosnihi senzora
4 var srednja           | #|
5
6 #funkcije-----
7 sub go                #"idi ravno"
8   motor.left.target = 250
9   motor.right.target = 250
10
11 sub stop              #"zaustavi se"
12   motor.left.target = 0
13   motor.right.target = 0
14
15 sub ulijevo           #"okret ulijevo"
16   motor.left.target = -250
17
18 sub udesno            #"okret udesno"
19   motor.right.target = -250
20
21 #inicijalizacija-----
22 call leds.top(0,0,230)          #"postavljanje boja
23 call leds.bottom.left(0,0,230)
24 call leds.bottom.right(0,0,230)
25
26 motor.left.target = 0          #"postavljanje početne brzine
27 motor.right.target = 0
28
29 #pokretanje i zaustavljanje robota (programa)-----
30 onevent button.forward          #"na pritisak gornjeg gumba
31   callsub go                    #"idi ravno
32
33 onevent button.backward          #"na pritisak donjeg gumba
34   callsub stop                  #"zaustavi se
35
```

Slika 10 Inicijalizacija koda „verzija1_zid“

Zadnji blok, vidljiv na slici 11, predstavlja glavni dio programa. Počinje s lokalnim događajem „prox“, koji se pokreće velikom frekvencijom te uzima podatke koje šalju svi senzori robota. Kada se robot kreće, on računa maksimalnu, minimalnu i srednju vrijednost prednjih senzora (označeni brojevima od 0 do 4). Kada maksimalna vrijednost bilo kojeg senzora, pređe granicu, to znači da se previše približio zidu hodnika. Tada ga je potrebno ispraviti, da se ne zabije u zid, a to činimo „if-else“ principom (od 34. do 38. linije). U konačnici, kada se uspije ispraviti, tj. kada svi senzori očitavaju vrijednost manjom ili jednakom od granične, nastavlja s radom.

```
28 #upravljanje robotom
29 onevent prox
30   if motor.right.target != 0 then
31     call math.stat( prox.horizontal[0:4], minimum, maksimum, srednja)
32
33   when maksimum > GRANICA do
34     if prox.horizontal[0] > prox.horizontal[4] then
35       callsub udesno
36     else
37       callsub ulijevo
38     end
39   end
40
41   when maksimum <= GRANICA do
42     callsub go
43   end
44 end
```

Slika 11 Glavni dio koda "verzija1_zid"

Ovaj jednostavan kod ispravno funkcionira, no ima jedan veliki problem. Naime, ne postoji senzor na Thymiu koji se nalazi na samom boku robota, stoga on nema FOV („field of view“) od 180 stupnjeva. To čini problem na već definiranoj mapi, čiji su hodnici širine 15 cm. Ako Thymio ne ide ravno u hodniku, neće uspjeti očitati bočne zidove pa će se zabiti u njih. Funkcionira tek kada povećamo širinu hodnika na 17 cm, no onda imamo komplikaciju u izgradnji mape. Pojavljuju se nejednakosti (ne može svaki hodnik biti jednake širine) i mnogo je više zidova različitih duljina potrebno nabaviti, što je dosta neefikasno. Zato je potrebno bilo promijeniti način gibanja robota.

Nova ideja, koja popravljajući navedeni problem, temelji se na korištenjem 2 svjetlosna senzora ispod Thymia. On će tada, umjesto praćenja zidova, pratiti crne linije po sredini svakog hodnika. Na ovaj način ostavljamo izgled mape i olakšavamo programiranje, jer koristimo samo 2 senzora. Za razlikovanje od prethodnog koda, ovaj ćemo nazvati „verzija1_linija“.

Jedina razlika u inicijalizaciji koda jest da umjesto konstante „GRANICA“ uvedimo konstantu „CRNA_LINIJA“ čiju vrijednost postavljamo na 300. Također, vidimo da je ostala ista struktura koda; lokalni događaj, računanje minimalne, maksimalne i srednje vrijednosti, uočavanje prekoračenja, popraćeno „if-else“ principom za ispravljanje gibanja te nastavak gibanja robota. Razlika jest što robot ne prati zidove, nego crnu liniju pomoću 2 senzora.

```
28 #upravljanje robotom-----
29 onevent prox                                     #provjeri blizine
30 if motor.right.target != 0 then                 #ako se robot giba
31     call math.stat( prox.ground.delta[0:1], minimum, maksimum, srednja) #izračunaj min, max i srednju vrijednost svjetlosnih senzora
32
33
34 when maksimum > CRNA_LINDA do                   #ako je uočio svjetlost (makao se s linije)
35
36 if prox.ground.delta[0] > CRNA_LINDA and prox.ground.delta[1] <= CRNA_LINDA then #ako s lijeve strane svjetlo
37     callsub udesno                               #idi udesno
38 else                                             #inace
39     callsub ulijevo                               #idi ulijevo
40 end
41 end
42
43 when maksimum <= CRNA_LINDA do                 #kad se okrene
44     callsub go                                     #idi ravno
45 end
46 end
```

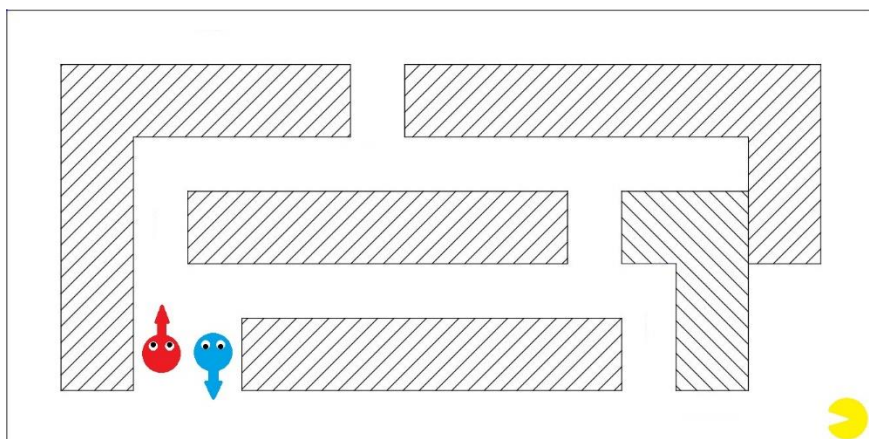
Slika 12 Glavni dio koda "verzija1_linija"

Ovakva izvedba gibanja robota je kvalitetnija od prethodne. Nema nikakvog zabijanja u zid, bira raskrižja na temelju nesavršenosti postavljenih linija te puno brže reagira. Jednostavan je kod za razumijevanje i on će činiti Verziju 1. Kada uvedemo drugog robota u igru, potrebno je postaviti nekakvu komunikaciju. Tada se uspostavlja mreža robota, pomoću koje oni mogu međusobno komunicirati. To nam ide u korist, jer tako duhovi mogu znati na koga su naišli: drugog duha ili igrača te time označiti mogući kraj igre. Naravno, nije efikasno imati dodatno računalo kako bi igra funkcionirala, jer čini igru teže prenosivom.

2.2.2 Verzija 2 – gibanje duhova po koreografiji

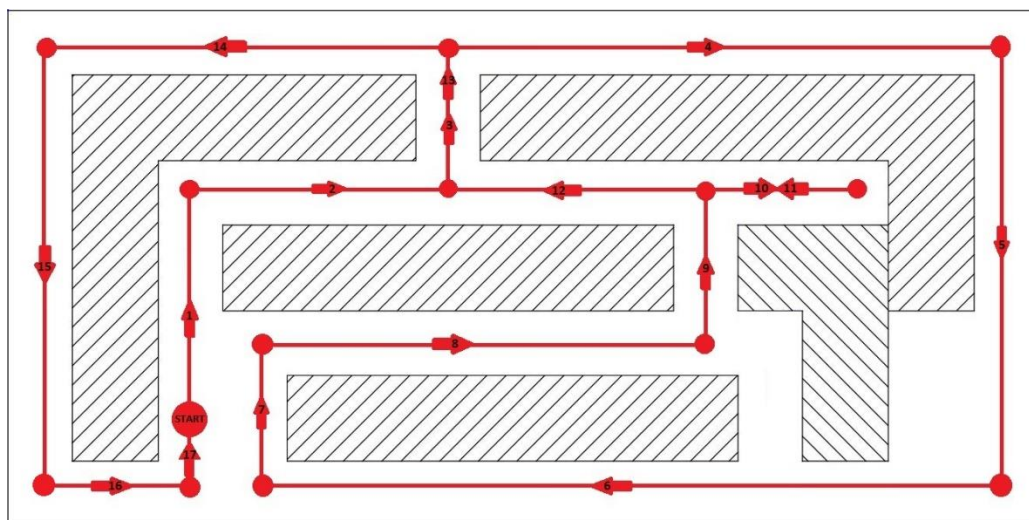
Verzija 2 radi kao nekakav ples. Svaki duh ima definiran način gibanja. Obilaze mapu, tako da se nikada ne susretnu i da cijela mapa bude prijeđena. Igrač, bar u početku, to neće znati, jer se roboti gibaju naizgled nasumično.

Za početak, potrebno je odrediti kako će se koji duh gibati. Početne pozicije svih robota će biti isti neovisno o kojoj se verziji radi; „Pac-man“ s donje desne strane, a duhovi jedan do drugog, na lijevoj strani mape. Na slici 13, osim početnih pozicija, možemo vidjeti i početnu orijentaciju robota.

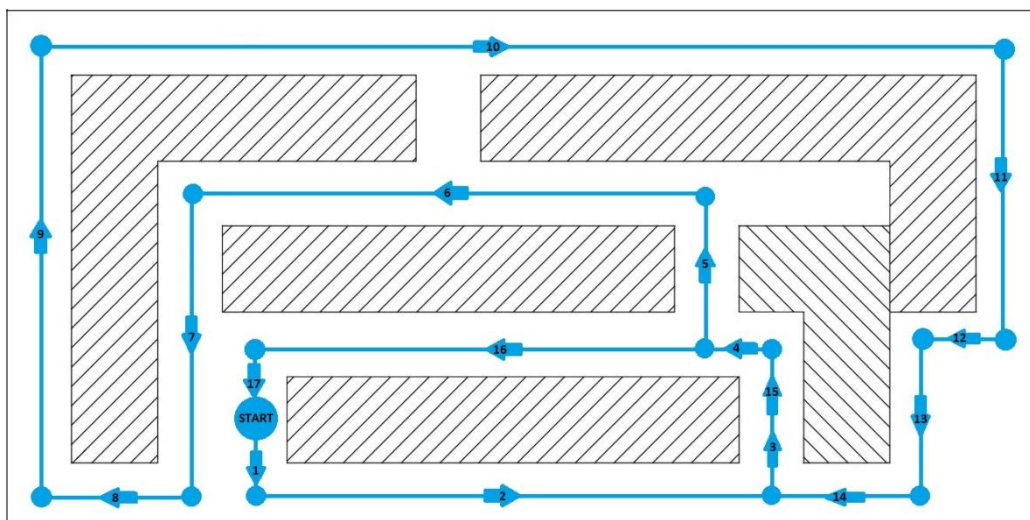


Slika 13 Početne pozicije karaktera

Sada je potrebno kreirati mapu gibanja duhova. Za svakog duha je isti proces. Počnemo s nasumičnim crtanjem. Pri tome imamo na umu da moraju biti, koliko je moguće, u svakom trenutku što dalje jedan od drugog, da u obidu mapu u cijelosti te da trajanje jednog kruga bude otprilike jednako. Ispravljamo crteže, dok ti uvjeti nisu približno ispunjeni. Na slikama 14 i 15 prikazani su konačni modeli gibanja duhova.



Slika 14 Prikaz gibanja crvenog duha



Slika 15 Prikaz gibanja plavog duha

Zatim za svakog duha računamo koliko mu vremena treba za neki hodnik. Za te kalkulacije smo postavili brzine motora na „200“, što predstavlja 8 cm/s. Također, uzeli smo obzir trajanje rotacije robota, što pri istim brzinama motora, iznosi 0.9229 rad/s. Na ovaj način dobivamo točniji dojam njihovog gibanja te, ako je potrebno, možemo još ispraviti modele kako bi uvjeti bili ispunjeni. Na tablicama vidimo izračunate vrijednosti svakog modela. Oznake se poklapaju sa slikama 14 i 15. Na podnožju tablica vidimo vrijednosti ukupnog prijeđenog puta s_{uk} , vrijeme trajanja prelaska puta t_{suk} , broj rotacija n_{rot} , vrijeme trajanje svih rotacija t_{rotuk} i ukupno trajanje gibanja robota t_{uk} .

	[cm]		[s]
s_1	55	t_1	6.875
s_2	60	t_2	7.5
s_3	35	t_3	4.375
s_4	130	t_4	16.25
s_5	105	t_5	13.125
s_6	175	t_6	21.875
s_7	35	t_7	4.375
s_8	105	t_8	13.125
s_9	35	t_9	4.375
s_{10}	35	t_{10}	4.375
s_{11}	35	t_{11}	4.375
s_{12}	60	t_{12}	7.5
s_{13}	35	t_{13}	4.375
s_{14}	95	t_{14}	11.875
s_{15}	105	t_{15}	13.125
s_{16}	35	t_{16}	4.375
s_{17}	15	t_{17}	1.875
s_{uk}	1150	t_{suk}	143.75
n_{rot}	17	t_{rotuk}	15.6895
		t_{uk}	159.4395

Table 1 Tablica vrijednosti gibanja crvenog duha

	[cm]		[s]
s ₁	15	t ₁	1.875
s ₂	120	t ₂	15
s ₃	35	t ₃	4.375
s ₄	15	t ₄	1.875
s ₅	35	t ₅	4.375
s ₆	120	t ₆	15
s ₇	70	t ₇	8.75
s ₈	35	t ₈	4.375
s ₉	105	t ₉	13.125
s ₁₀	225	t ₁₀	28.125
s ₁₁	70	t ₁₁	8.75
s ₁₂	20	t ₁₂	2.5
s ₁₃	35	t ₁₃	4.375
s ₁₄	35	t ₁₄	4.375
s ₁₅	35	t ₁₅	4.375
s ₁₆	120	t ₁₆	15
s ₁₇	20	t ₁₇	2.5
s _{uk}	1110	t _{suk}	138.75
n _{rot}	16	t _{rotuk}	14.76659
		t _{uk}	153.51659

Table 2 Tablica vrijednosti gibanja plavog duha

Uočavamo da su konačni modeli ispunili uvjete koliko je to moguće; duhovi se će se susresti nakon dugo vremena, crveni duh kasni za plavim za otprilike 6 sekundi (približno jednako trajanje kruga) te obilaze mapu u cijelosti. Naravno, moramo uzeti u obzir da su ovi podaci izračunati u idealnim uvjetima. Robot se nikada ne giba 8 cm/s, ovisno o podlozi te ove kalkulacije imaju pretpostavku da ćemo ih svaki put točno postaviti u početnoj poziciji. Modeli su dovoljno dobri, da se roboti neće susresti nakon nekog vremena, više nego što će vjerojatno igrač potrošiti na igru. Za drugi problem je potrebno uvesti kalibraciju robota; u principu Verziju 1.

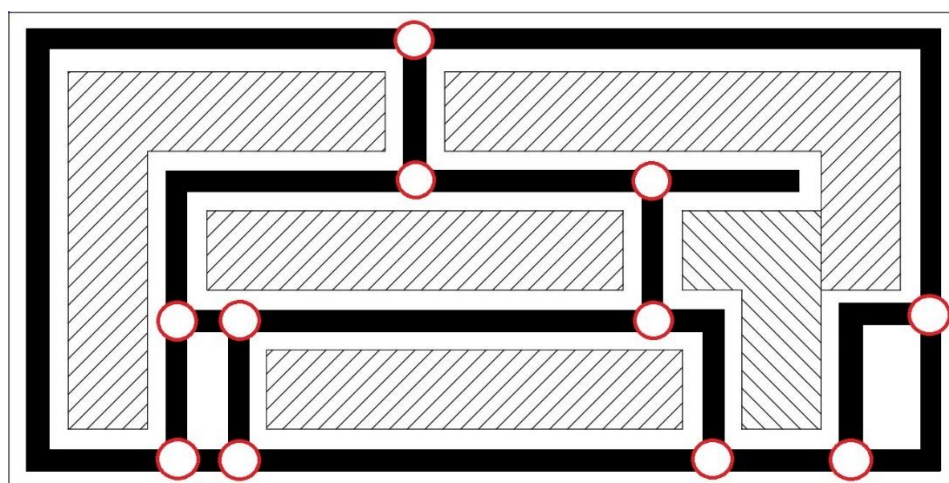
Sada je potrebno prevesti koreografije u kod. Kao kod Verzije 1, početak ćemo s inicijalizacijom koda. Kodovi „verzija2_crveni“ i „verzija2_plavi“ imaju skoro identičnu inicijalizaciju, prikazanu na slici 16. U bloku „varijable“, osim već poznatih minimuma, maksimuma i srednjih varijabli, uočavamo novu zvanu „raskrižja“. Ona predstavlja sva raskrižja u kojima moramo robotu zadati smjer kretanja (slika 17), što će činiti njegovu koreografiju. Definiraju se 2 konstante: CRNA_LINIJA (vrijednost „300“) i GRANICA (vrijednost „1500“). Ostatak koda je identičan onom u Verziji 1.

```

1 #varijable-----
2 var minimum                                     #|
3 var maksimum                                     #| namijenjeno za "okreti" varijablu
4 var srednja                                     #|
5
6 var minimum1                                     #|
7 var maksimum1                                   #| namijenjeno za varijablu svjetlosnih senzora
8 var srednja1                                   #|
9
10 var minimum2                                    #|
11 var maksimum2                                   #| namijenjeno za varijablu senzora za daljinu
12 var srednja2                                   #|
13
14 var raskrizja[14]                               #varijabla svih raskrizja u koreografiji
15 call math.fill(raskrizja,0)                   #punjenje varijable "raskrizja" nulama
16
17 #inicijalizacija-----
18 call leds.top(230,0,0)                         #postavljanje boja
19 call leds.bottom.left(230,0,0)
20 call leds.bottom.right(230,0,0)
21
22 motor.left.target = 0                          #postavljanje početne brzine
23 motor.right.target = 0
24
25 #funkcije-----
26 sub go                                           #"idi ravno"
27   motor.left.target = 200
28   motor.right.target = 200
29
30 sub stop                                         #"zaustavi se"
31   motor.left.target = 0
32   motor.right.target = 0
33
34 sub ulijevo                                     #"okret ulijevo"
35   motor.left.target = -200
36
37 sub udesno                                     #"okret u desno"
38   motor.right.target = -200
39
40 #pokretanje i zaustavljanje robota (programa)-----
41 onevent button.forward                         #na pritisak gornjeg gumba
42   callsub go                                   #idi ravno
43
44 onevent button.backward                       #na pritisak donjeg gumba
45   callsub stop                                #zaustavi se
46

```

Slika 16 Inicijalizacija kodova "verzija2_crveni" i "verzija2_plavi"



Slika 17 Sva problematična raskrižja u mapi

Glavni dio programa se sastoji od 2 dijela: praćenje linije i koreografija. Prvi dio, prikazan na slici 18, je preuzet iz Verzije 1 i identičan je za oba duha. Jedini dodatak vidimo u liniji 51, koji je namijenjen drugom dijelu koda.

```
47 #upravljanje robotom-----
48 onevent prox                                #provjeri blizine
49 if motor.right.target != 0 then              #ako se robot giba
50     call math.stat( prox.ground.delta[0:1], minimum1, maksimum1, srednja1) #izračunaj min, max i srednju vrijednost svjetlosnih senzora
51     call math.stat( prox.ground.delta[0:1], minimum2, maksimum2, srednja2) #izračunaj min, max i srednju vrijednost senzora za daljinu
52
53 #dio programa za praćenje linije-----
54 when maksimum1 > CRNA_LINDA do                #ako je uočio svjetlost (makao se s linije)
55
56     if prox.ground.delta[0] > CRNA_LINDA and prox.ground.delta[1] <= CRNA_LINDA then #ako s lijeve strane svjetlo
57         callsub udesno                                #idi udesno
58     else
59         callsub ulijevo                                #inace
60     end
61 end
62
63 when maksimum1 <= CRNA_LINDA do                #kad se okrene
64     callsub go                                         #idi ravno
65 end
66 end
67
```

Slika 18 Dio glavnog programa namijenjen praćenju linije

Zadnji dio koda sadrži cilj ove verzije. Započinje računanjem minimalne, maksimalne i srednje vrijednosti varijable „raskrižja“. Njenu minimalnu vrijednost odmah koristimo za provjeru je li krug ispunjen. Ako je ispunjen, tj. sva polja su ispunjena s „1“ onda treba resetirati varijablu, kako bi robot započeo novi krug. Ovaj sljedeći veliki blok je namijenjen samoj koreografiji. Uvjet na liniji 74 služi za provjeru nalazi li se robot na raskrižju. Koristimo sve senzore za što kvalitetniji rezultat. Nadalje, robot kreće s ispunom varijable „raskrižja“. Ona su označena redom kako robot na njih nailazi, počevši od 0. Kada naiđe na raskrižje, provjerava na koje je naišao, okrene se i onda popuni to polje s „1“. Neprođena raskrižja su popunjena s „0“. Na slici 19 možemo vidjeti primjer koda za plavog duha.

```
68 #dio programa za koreografiju robota-----
69 call math.stat(raskrizja, minimum, maksimum, srednja)#izračunaj min, max i srednju vrijedost varijable "okreti"
70
71 if minimum == 1 then                                #ako je najmanji broj u nizu "raskrizja" 1
72   call math.fill(raskrizja,0)                       #znaci da je robot obavio koreografiju pa resetiraj varijablu
73 else                                                 #inace
74   when (prox.horizontal[0]<GRANICA or prox.horizontal[4]<GRANICA) and maksimum2 < GRANICA and maksimum1<= CRNA_LINDA do #kad uoci raskrizje
75     if (raskrizja[0] == 0)then                       #|
76       callsub ulijevo                                #|
77       raskrizja[0] = 1                                #|
78     elseif (raskrizja[1] == 0)then                   #|
79       callsub ulijevo                                #|
80       raskrizja[1] = 1                                #|
81     elseif (raskrizja[2] == 0)then                   #|
82       callsub udesno                                 #|
83       raskrizja[2] = 1                                #|
84     elseif (raskrizja[3] == 0)then                   #|
85       callsub go                                     #| okreći se po koreografiji
86       raskrizja[3] = 1                                #|
87     elseif (raskrizja[4] == 0)then                   #|
88       callsub go                                     #|
89       raskrizja[4] = 1                                #|
90     elseif (raskrizja[5] == 0)then                   #|
91       callsub udesno                                 #|
92       raskrizja[5] = 1                                #|
93     elseif (raskrizja[6] == 0)then                   #|
94       callsub go                                     #|
95       raskrizja[6] = 1                                #|
96     elseif (raskrizja[7] == 0)then                   #|
97       callsub udesno                                 #|
98       raskrizja[7] = 1                                #|
99     elseif (raskrizja[8] == 0)then                   #|
100      callsub udesno                                 #|
101      raskrizja[8] = 1                                #|
102     elseif (raskrizja[9] == 0)then                   #|
103      callsub udesno                                 #|
104      raskrizja[9] = 1                                #|
105     elseif (raskrizja[10] == 0)then                  #|
106      callsub go                                     #|
107      raskrizja[10] = 1                               #|
108     elseif (raskrizja[11] == 0)then                  #|
109      callsub ulijevo                                #|
110      raskrizja[11] = 1                               #|
111   end
112 end
113 end
```

Slika 19 Dio glavnog programa namijenjen koreografiji plavog duha

Verzija 2 je puno kompleksnija od Verzija 1, što usporava robota. Iako nema dodatnog računala za komunikaciju, ova izvedba je lošija od prve, jer nije primjenjiv za drugačije mape. Također, igrač može shvatiti algoritam i time olakšati igru, a u slučaju greške, kompleksnost koda puno otežava njeno popravljjanje. I dalje duh ne zna je li uočio „Pac-man“-a ili zid, što dovodi do problema. Uvedemo li komunikaciju s drugim robotima, ova verzija nema smisla, jer smo dodali nešto što smo pokušali izbjeći. Možemo ga riješiti tako da duh osluškuje je li bilo šoka, tj. nekakvog zabijanja. No, to dovodi do kompliciranja koda, zbog čega ovoj verziji vrijednost još više pada. Naposljetku, koristi Verziju 1 kao potporu, stoga lako zaključujemo da je ona puno kvalitetnija od Verzije 2.

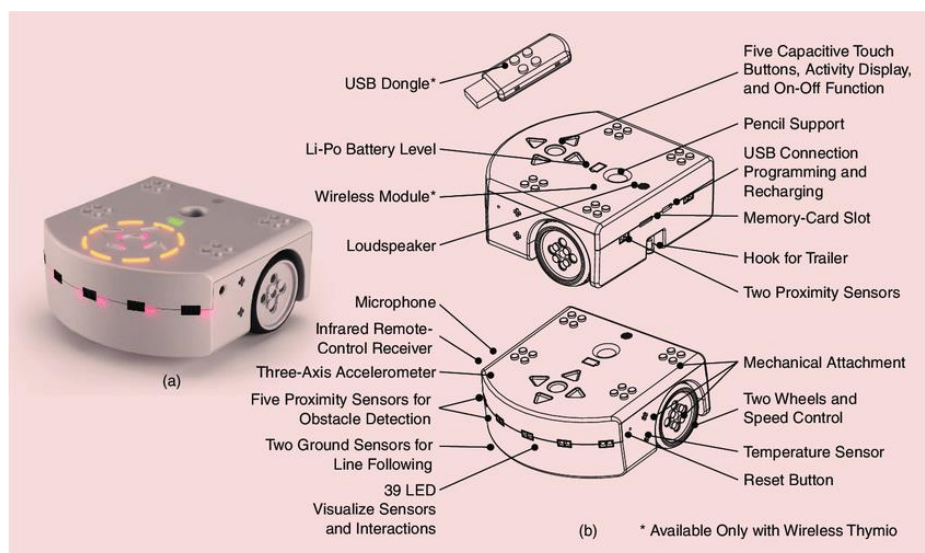
3. Tehnička dokumentacija

3.1 Tehničke značajke

3.1.1 Thymio wireless

DIMENZIJE	11 cm x 11.2 cm x 5.3 cm
TEŽINA	270 g
DIMENZIJE PAKETA	13.5 cm x 15 cm x 7.5 cm
SADRŽAJ PAKETA	<ul style="list-style-type: none">• Bežični Thymio robot• USB kabel za punjenje i programiranje• USB hardverski ključ („dongle“) za daljinsko programiranje• Korisnički priručnik
PROTOKOL BEŽIČNOG MODULA	2.4 GHz, protokol 802.15.4
BATERIJA	Li-Po, 3.7 V, 1500 mAh, punjiv putem USB-a
ŽIVOT BATERIJE	3 do 5 sati
VRIJEME PUNJENJA	1 do 2 sata
SENZORI	<ul style="list-style-type: none">• 9 infracrvenih (IR) senzora za blizinu (doseg oko 10 cm)• 5 kapacitivnih gumbi na dodir• 1 akcelometar u 3 smjera• 1 termometar• 1 mikrofoni (za snimanje ili detekciju zvuka)• 1 IR prijamnik (za daljinski upravljač)• 1 bežični modul• 39 LED svjetiljki
AKTUATORI	<ul style="list-style-type: none">• 2 DC motora• 1 zvučnik
MAKSIMALNA BRZINA	14 cm/s

Tablica 2 Tehničke značajke robota Thymio wireless



Slika 20 Prikaz Thymio wireless robota i njegovih elemenata

DODATNE MOGUĆNOSTI:

- Sadrži rupu između kotača za držanje olovke
- Može pratiti crnu liniju koju smo nacrtali ili isprintali
- Kuka iza robota može biti korisna za povlačenje prikolice
- Može se puniti korištenjem USB-micro punjača (za pametne telefone) ili spajajući se na računalo
- Sadrži utor za MicroSD karticu (za glazbu, kod ili druge datoteke)
- Jako otporan na šok
- Sadrži ispupčenja za dodatke
- Sadržava 6 načina ponašanja

THYMIO SUITE

Thymio suite je aplikacija za programiranje Thymio robota. Temelji se na Asebi jeziku, koja olakšava programiranje mobilnih robota. Aplikacija nudi 4 načina programiranja: „Visual programming“, „Blockly“, „Aseba Studio“ i „Scratch“. Također, nudi 5 jezika za programiranje u Asebi: engleski, njemački, francuski, talijanski i španjolski. Ova aplikacija se može instalirati na Linuxu, Mac OS –u i Windowsu.

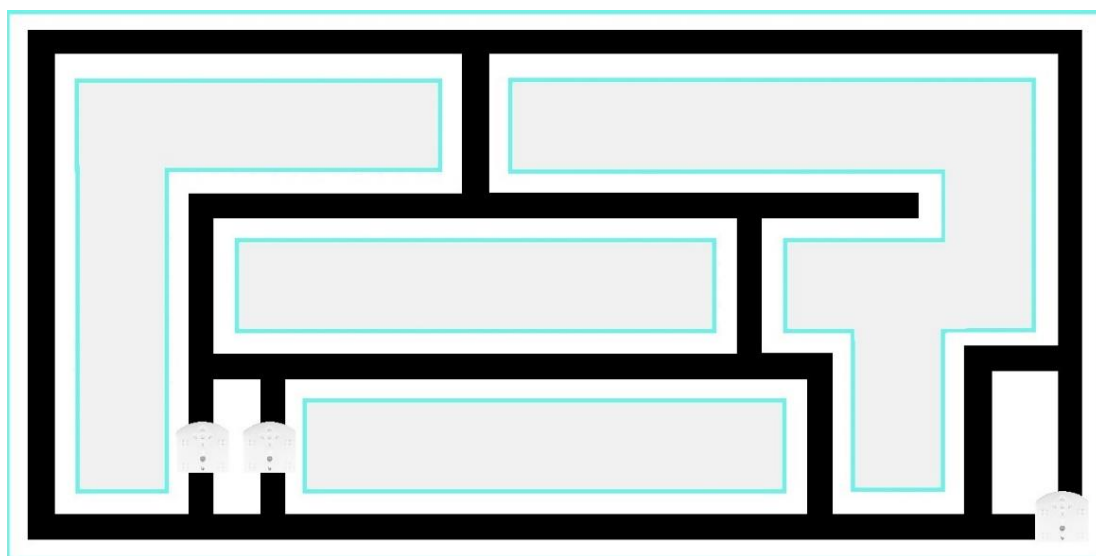


Slika 21 Ikona aplikacije Thymio Suite

3.1.2 Robotska rekreacija „Pac-man“ igre

DIMENZIJE	120 cm x 240 cm x 10 cm
TEŽINA	1 kg
SADRŽAJ	<ul style="list-style-type: none">• Zidovi od stirodura• Podloga za robote• 3x role crne izolir-trake (10 m x 1.5 cm)• 3x Thymio wireless• Micro SD kartica• Philips daljinski upravljač• Prijenosno računalo• Thymio Suite aplikacija• Programske datoteke
DIMENZIJE I KOLIČINA ZIDOVA	<ul style="list-style-type: none">• 10x (80 cm x 2 cm x 10 cm)• 23x (35 cm x 2 cm x 10 cm)• 22x (20 cm x 2 cm x 10 cm)
DIMENZIJA TRAKE	1185 cm x 3 cm ili 1185 x 5 cm

Tablica 3 Tehničke značajke završnog projekta



Slika 22 Izgled završnog projekta

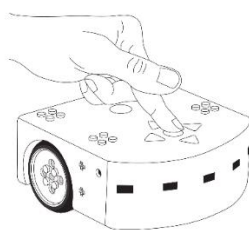
3.2 Upute za uporabu

3.2.1 Thymio wireless

PALJENJE I GAŠENJE ROBOTA

Za upaliti robota potrebno dodirnuti okrugli gumb, smješten u središtu 4 gumba u obliku strelica, te držati dok robot ne pozeleni i oglasi se određenim zvukom. Također će se osvijetliti mala ikona ispod gumba donje strelice, koja označava stanje baterije.

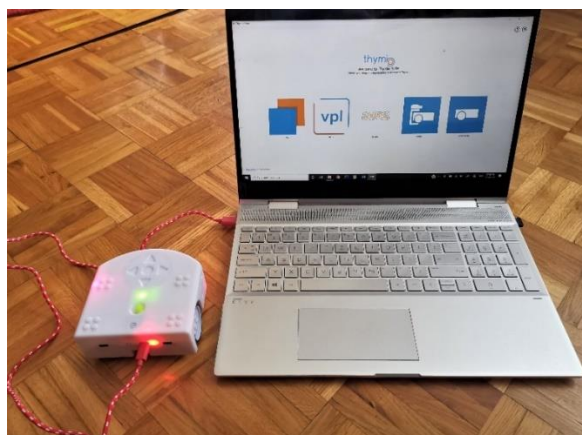
Gašenje robota se odvija istim procesom; dodirivanjem središnjeg, okruglog gumba dok se robot ne oglasi zvukom i više ne svijetli.



Slika 23 Paljenje/gašenje Thymio robota

PUNJENJE ROBOTA

Na poledini robota se nalazi USB-micro priključak, koji Thymio koristi za punjenje. Može se puniti spajanjem na računalo, ili pak korištenjem zidnog punjača (kao što se puni vlastiti telefon).



Slika 24 Punjenje Thymia putem USB kabela

Kada se postavi Thymia na punjenje, kraj priključka će se upaliti LED svjetiljka određenom bojom. Ako je baterija puna, pojavit će se plava boja, inače crvena.

UPRAVLJANJE ROBOTOM

Thymio ima već ugrađena 6 načina ponašanja. Svaki od njih je označen paljenjem LED svjetiljki određenom bojom. Možemo odabrati jedan od njih čim se Thymio upali. Posloženi su određenim redoslijedom, a biramo ih koristeći gumb u obliku strelica (gornji i desni za jedan smjer, a ostali za drugi). Za pokretanje i zaustavljanje načina ponašanja, potrebno je dodirnuti središnji gumb.

Svaki od ovih programa ima ime koje opisuje što Thymio radi. Koristeći desni ili gornji gumb u obliku strelica, Thymio će nuditi načine ponašanja redom:

- „Friendly“ („Priateljski“)

Thymio će pratiti objekt ispred sebe te će se zaustaviti ako dođe do ruba podloge. Ovaj način rada je prikazan zelenom bojom, čim se robot upali.

- „Obedient“ („Poslušan“)

Ovaj način ponašanja je označen paljenjem ljubičastih LED svjetla. Dodirivanjem lijeve ili desne strelice Thymio će se okretati u tom smjeru, a gornja ili donja strelica označavaju gibanje robota naprijed ili nazad. Da bi ubrzali njegovo gibanje, potrebno je dodirnuti više puta onu strelicu koja označava smjer u kojem se Thymio kreće, a za usporavanje suprotnu strelicu. Također, robot se neće se zaustavljati u slučaju ako dođe do ruba podloge.

- „Investigator“ („Istraživač“)

Prikazan svijetlom plavom bojom, ovaj način rada služi za praćenje crne linije. Linija mora biti široka najmanje 3 cm. Treba paziti na rub podloge, jer će to Thymio shvatiti kao crno polje na koje treba ići.

- „Attentive“ („Pažljiv“)

Thymio pali LED svjetiljke ljubičastom bojom i počne slušati. Pljeskom rukama određujemo kako će se robot gibati na sljedeći način:

- 1 pljesak: okreće se ili ide ravno
- 2 pljeska: zaustavlja/pokreće se
- 3 pljeska: rotira se oko vlastite osi

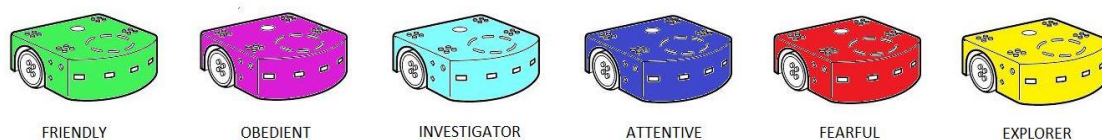
Treba jako glasno i brzo pljeskati, da bi Thymio to mogao čuti. Kada ih uspije registrirati, pali svijetla u krug (koji okružuju gumbe u obliku strelica), ovisno o broju pljesaka.

- „Fearful“ („Bojažljiv“)

Označen crvenom bojom, ovaj način ponašanja je suprotan „Prijateljskom“ načinu. Robot će bježati od objekta u blizini pazeći na rub podloge.

- „Explorer“ („Avanturist“)

Osvijetljen žutim LED svjetiljkama, Thymio se kreće nasumično po podlozi. Izbjegava stvari oko sebe te se zaustavlja ako dođe do ruba podloge, ili ako je polje crne boje.



Slika 25. Thymiovi načini ponašanja

Moguće je mijenjati postavke robota. Držeći lijevi i desni gumb 3 sekunde, upalit će se novi način rada. U njemu je moguće mijenjati 3 postavke. Svaki od njih, kao kod načina ponašanja, prikazan je određenom bojom za lakše raspoznavanje. Narančasta predstavlja glasnoću, zelena brzinu kotača i ljubičasta bežično uparivanje modula. Dodirivanjem središnjeg gumba možemo početi mijenjati određenu opciju.

Kod promjene glasnoće, gornji gumb predstavlja povećanje, a donji smanjenje glasnoće. Trenutno stanje je vidljivo na LED svjetiljkama koje okružuju gumbe u obliku strelice. Za spremanje promjene, treba dodirnuti središnji gumb 3 sekunde.



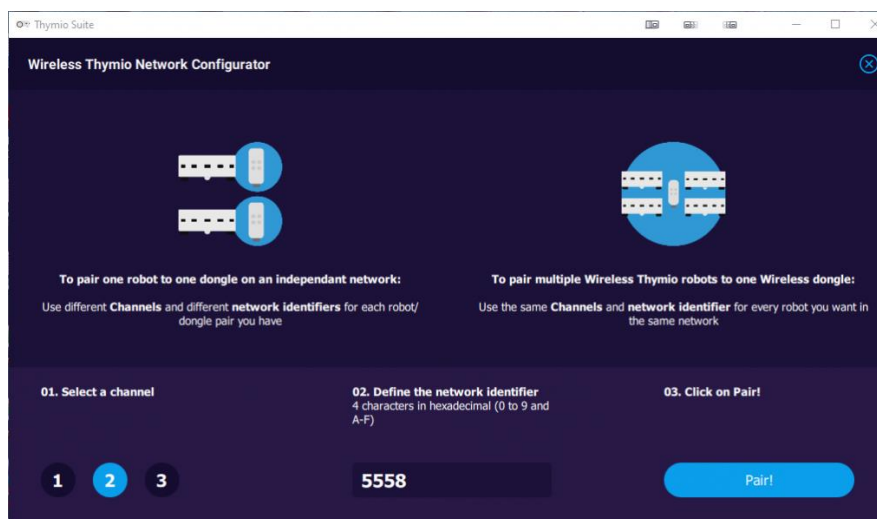
Slika 26 Postavka mijenjanja glasnoće

Brzinu kotača treba pažljivo mijenjati, jer mijenjamo brzinu individualnog kotača. Nakon odabira ove opcije, treba namjestiti Thymia da radi u „Obedient“ načinu ponašanja. Gornji i donji gumb služe za pokretanje Thymia naprijed ili nazad. Kada je robot u pokretu, lijevi i desni gumb mijenjaju brzine kotača. Recimo, ako Thymio skreće udesno, treba pritisnuti lijevi gumb da bi išao ravno. Dodirivanjem središnjeg gumba spremi se promjena. Zatim, da bi mogli koristiti Thymia, potrebno ga je resetirati.

Thymio može komunicirati s računalom putem kabela ili USB „dongle“-a, ali i drugim Thymio robotima. Da bi to bilo moguće, potrebno je zadovoljiti 3 uvjeta:

- Svi moraju biti namješteni na isti radio kanal
- Svi moraju imati isti „network identifier“
- Svaki mora sadržavati jedinstven „node identifier“

Nakon pokretanja Thymio Suite aplikacije, u gornjem desnom kutku se nalazi gumb koji otvara dodatne opcije. Pritiskom na opciju „Pair a Wireless Thymio to a Wireless dongle“ i potvrđivanjem upita, prikazat će se prozor „Wireless Thymio Network Configurator“. Nakon priključivanja željenog Thymio robota putem USB kabela i „dongle“ drugog robota te pritiskom na gumb „Pair!“, prikazat će se nove opcije.



Slika 27 Prozor "Wireless Thymio Network Configurator"

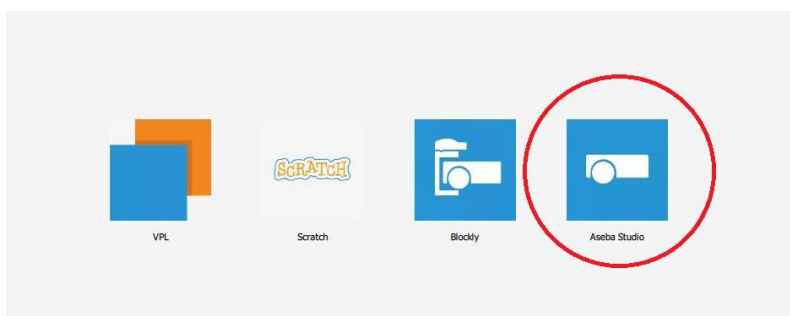
Potrebno je postaviti kanal na koji će se spojiti Thymio roboti i odabrati neki „network identifier“ kao na slici 27. Na kraju samo treba kliknuti „Pair!“. Zatim, uključite robota kojeg ste spojili putem USB kabela i uđite u postavku bežičnog uparivanja modula. „Dongle“ i Thymio moraju bljeskati istom frekvencijom u znak uparivanja (približite robota ako je

potrebno). Kada se to dogodi, treba isključiti robota kako bi se promjene spremile. Konačno, na računalu treba kliknuti „Flash into dongle“ kako bi završili postupak.

TEKSTUALNO PROGRAMIRANJE ROBOTA

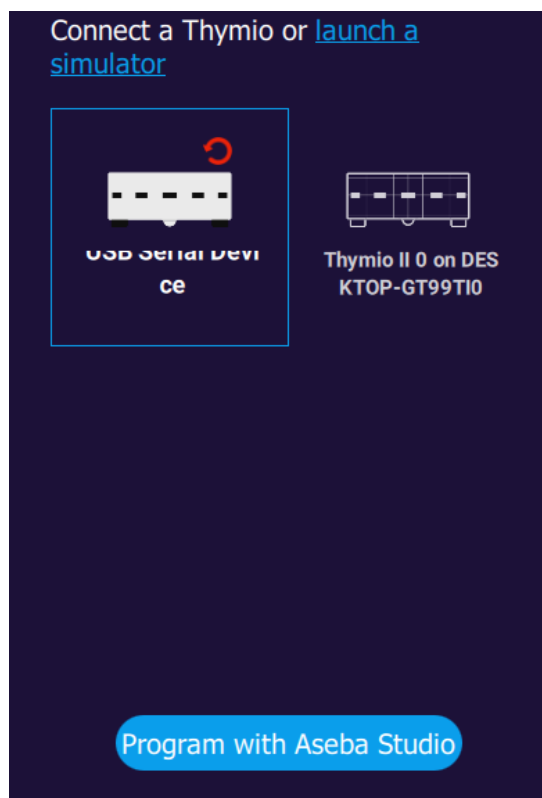
Postoje 4 načina programiranja Thymio robota: „Visual programming“, „Blockly“, „Aseba Studio“ i „Scratch“. U ovom radu objasnit ću „Aseba Studio“ (tekstualni način programiranja), koji je bio korišten za izradu samog projekta.

U početnom se prozoru aplikacije Thymio Suite nalazi ikona „Aseba Studio“, koja označava tekstualni način programiranja, kao što je prikazano na slici 28.



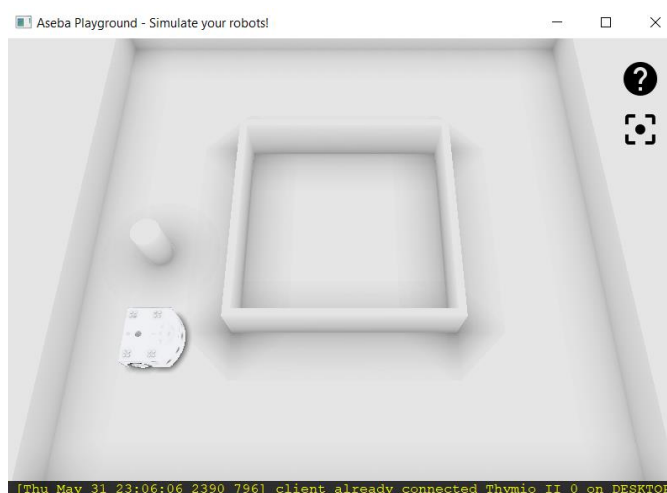
Slika 28 Početni prozor aplikacije Thymio Suite

Jednim klikom na ikonu sa slike 28, prikazat će se novi prozor u kojem određujemo hoćemo li programirati na neki spojeni Thymio ili putem simulacije. Za svaki način programiranja se prikazuje ovaj prozor, koji također sadrži kratak opis tog načina programiranja. Ako želimo programirati na pravom, upaljenom Thymiu, potrebno ga je spojiti ili putem USB „dongle“-a ili USB kabela. Tada se pojavljuje njegova ikona te kada ga označimo, možemo krenuti s programiranjem pritiskom na gumb „Program with Aseba Studio“. Na slici 29 vidimo ikonu spojenog Thymia (lijevo) i simuliranog Thymia (desno).

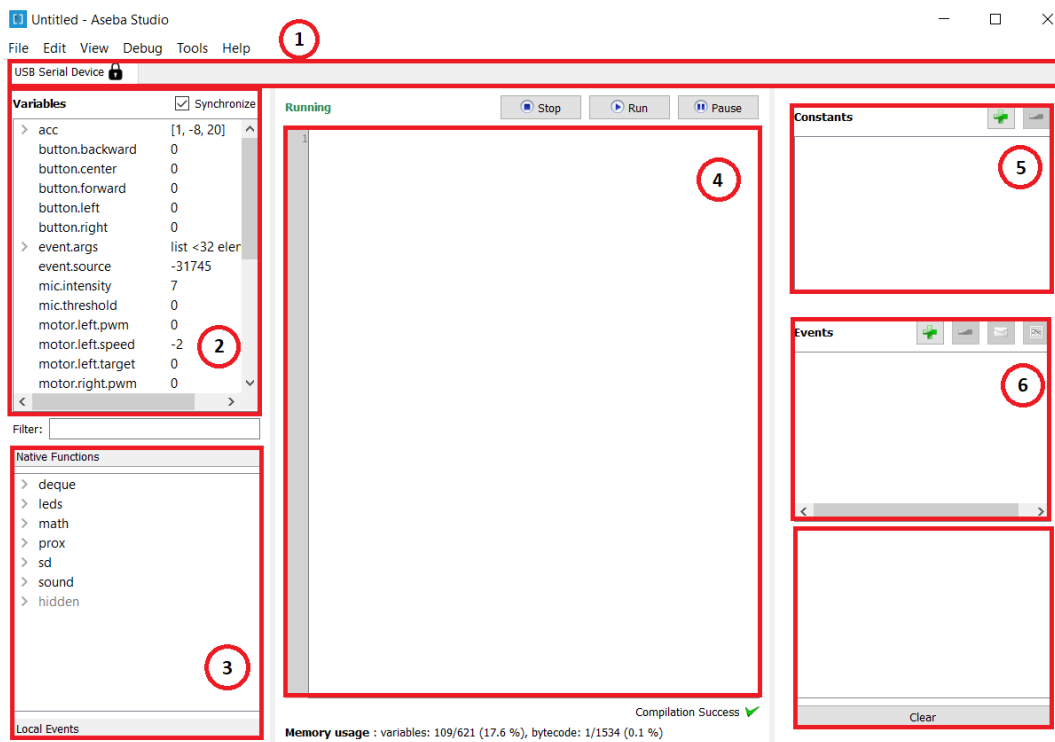


Slika 29 Spajanje i simuliranje robota na računalu

Ukoliko želimo programirati putem simulacije, potrebno je kliknuti na „launch a simulator“ te odabrati izgled mape. Na slici 30 možemo vidjeti izgled mape „thymio.playground“. Zatim, pojavit će se slična ikona, koju je isto tako potrebno označiti da bi mogli početi s programiranjem.



Slika 30 Izgled simulirane mape "thymio.playground"



Slika 31 Prikaz sučelja "Aseba Studio"

Izgled sučelja može se podijeliti na 6 osnovnih dijelova:

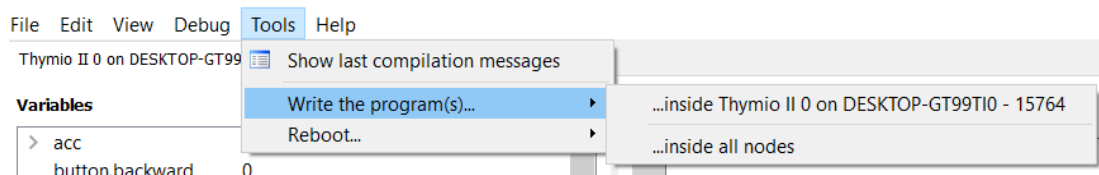
1. TRAKA ZA ODABIR THYMIA – ova traka daje mogućnost istodobnog programiranja više Thymia (simuliranih ili pravih). Svaki od njih ima vlastito sučelje i program što olakšava kontroliranje cjelokupne mreže.
2. PROZOR S PRIKAZOM VARIJABLI – u ovom prozoru možemo vidjeti sve varijable koje Thymio sadržava. Također, kada je polje „Synchronize“ označeno, one se kontinuirano ažuriraju velikom frekvencijom. Te varijable kontroliraju ponašanje robota. Recimo; brzine individualnog motora, gumbi, glasnoća mikrofona, senzori, temperatura, brojač...itd.
3. PROZOR IZVORNIH FUNKCIJA I LOKALNIH DOGAĐAJA – ovdje se nalazi popis funkcija i definiranih događaja koji olakšavaju programiranje. Recimo paljenje/gašenje LED svjetiljki, matematičkih funkcija, snimanje/reproduciranje zvuka...itd.
4. PROSTOR ZA PROGRAMIRANJE – ovo veliko bijelo polje je namijenjeno samom kodu programa. Koristi se jezik Aseba, čija je sintaksa jednostavna za naučiti zbog sličnosti drugim programskim jezicima, kao npr. Matlabu. Njezinu dokumentaciju i kratki vodič možete naći na stranici: <http://wiki.thymio.org/en:asebausermanual>. Poviše se nalaze gumbi za pokretanje („Run“), zaustavljanje („Stop“) i pauziranje („Pause“) programa. Također, Aseba će vas

upozoravati u slučaju sintaksnih ili kompilacijskih grešaka, ispod prostora za programiranje. Kada pokrenemo program, njegovo izvršenje možemo gledati na tom Thymiu.

5. PROZOR POSTOJEĆIH KONSTANTI – ovaj prozor služi za prikaz i definiranje konstanti koji će se dalje koristiti u programu. Nove se definiraju pritiskom gumba „+“, te brišu gumbom „-“,.

6. PROZOR DOGAĐAJA NA MREŽI – u ovom prozoru se ispisuju nedavni događaji zajedno s njihovim nazivima, vremenskim oznakama i parametrima. Mogu se dodavati i brisati kao i konstante, ali i poslati drugom Thymiu te skicirati graf. Ovaj prozor olakšava praćenje ponašanja na cijeloj mreži.

U samog Thymia može se spremati samo jedan program. On se prilikom gašenja robota, ili pokretanja drugog programa, neće izbrisati. Bit će nadodan kao 7. način ponašanja, koji neće biti prikazan bojom. Moguće ga je spremati na više Thymia, ukoliko su spojeni na istu mrežu. Proces se odvija na slijedeći način: pritiskom na gumb „Tools“, koji se nalazi na gornjoj traci, prikazat će se 3 opcije. Potrebno je pritisnuti srednju „Write the program(s)...“ te onda odabrati na kojeg Thymia želite prebaciti program. Prikaz postupka je vidljiv na slici 32. Za brisanje programa, potrebno je kliknuti treću opciju „Reboot...“ i odabrati na kojem Thymiu želite to odraditi.



Slika 32 Prikaz spremanja programa u jedan ili više Thymio robota

Thymio već ima nekoliko zvučnih efekata u svojoj memoriji. Njih smo čuli pri paljenju načina ponašanja, pritisku gumbova...itd. Možemo dodati i vlastite zvukove, no za to je potrebna Micro SD kartica. Na nju stavljamo zvukove, .wav formata, te ju uključimo u Thymio. Snimanje zvukova se obavlja na razne načine. Na stranicama <http://aseba.wikidot.com/en:thymiosoundrecorder> i <https://www.techykids.com/creating-and-playing-a-sound-file/> nalaze se detaljne upute snimanja zvuka pomoću koda kojeg nude ili aplikacijom Audacity. Nakon dodavanja zvuka, bit će spremljeni u Thymiu pod nekim brojem. Da bi Thymio reproducirao zvuk, potrebno je samo dodati „call sound.play(x)“ (x označava broj kojim odabiremo zvuk).

Robot se može kontrolirati putem daljinskog upravljača. Postoji već napravljen upravljač za njega, no može se kontrolirati i drugim daljinskim upravljačima. Potrebno je da taj upravljač ima rc5 enkodiranje, koje Thymio prepoznaje. Zasadu imamo sigurnost u Philipsove upravljače, te univerzalne koji imaju mogućnost namještanja enkodiranja. Da bi testirali upravljač, potrebno je uperiti u Thymia, dok je on u ljubičastom načinu ponašanja, te pritiskati gumb. Ako Thymio ne reagira i ne bljeska u znaku uspješnog komuniciranja, upravljač nije kompatibilan s njim. U programu možemo namjestiti da gumbi reagiraju po vašoj želji pod lokalnim događajem rc5. Varijabla „rc5.command“ pokazuje što Thymio očita kad se pritisne određen gumb. Na slici 33 možemo vidjeti primjer koda s opisom koraka s desne strane.

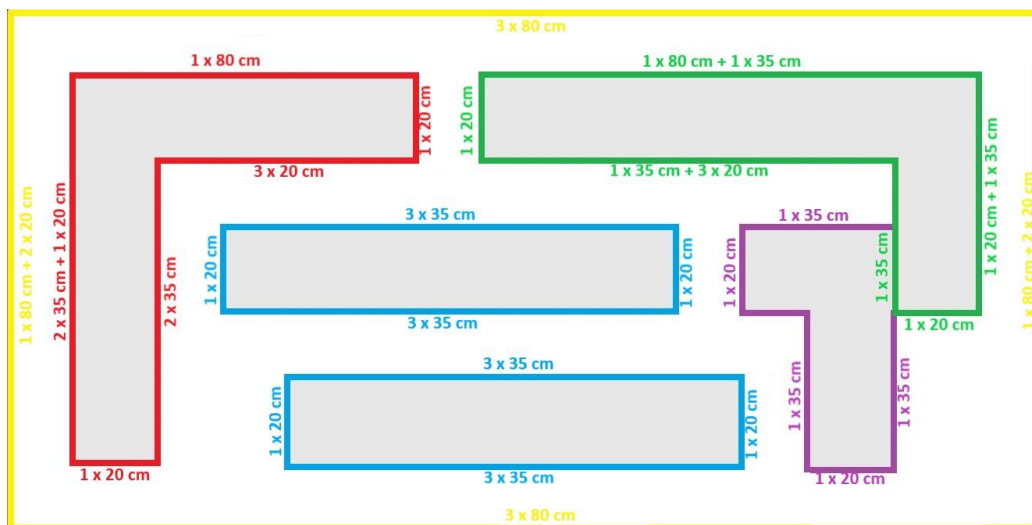
```
1 onevent rc5                                     #kada daljinski pravljac pošalje signal
2   if rc5.command == 1 then                       #kada se stisne gumb 1
3     motor.left.target = 200                     #neka Thymio ide ravno
4     motor.right.target = 200
5   end
```

Slika 33 Primjer kodiranja daljinskog upravljača s Thymiom

3.2.2 Robotska rekreacija „Pac-man“ igre

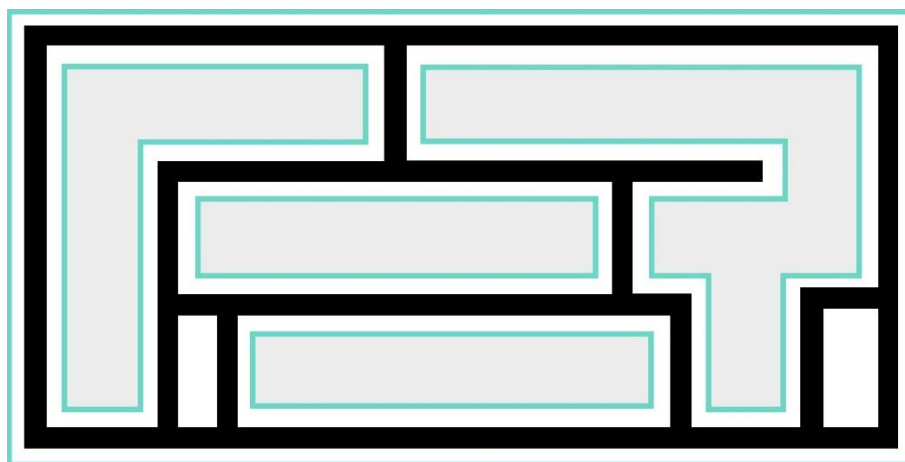
PRIPREMA MAPE

Da bi mogli igrati igru, prvo je potrebno pripremiti prostor. Za to su potrebni: podloga za robote, 55 zidova od stirodura i 3 role crne izolir-trake. Zidove treba postaviti u obliku FER-ovog loga, na podlozi za robote, tako da širina hodnika bude ili 15 cm ili 30 cm, kao na slici 34. Također, na njoj vidimo poviše svakog zida opis od koliko i kojih je elemenata sastavljen. Recimo, duljina vanjskog okvira mape je sastavljena od 3 zida, duljine 80 cm.



Slika 34 Prikaz sastavljanja zidova

Izolir-traku postavljamo na sredini hodnika. Moraju biti širine od minimalno 3 cm, da bi roboti mogli efikasno funkcionirati. Na slici 35, možemo vidjeti kako u konačnici mapa treba izgledati.



Slika 35 Konačni izgled mape

PRIPREMA ROBOTA

Od 3 Thymio robota, jedan predstavlja „Pac-man“-a, a ostala dva duhove. Sva tri robota je potrebno spojiti na istu mrežu, putem USB „dongle“-a. Na taj način su svi roboti povezani, mogu međusobno komunicirati te ih je lakše pokrenuti istovremeno. Preporučuje se, za daljnje namještanje, uključiti Aseba Studio.

„Pac-man“-a predstavlja igrač, koji ga kontrolira putem daljinskog upravljača. Prvo se preporuča testirati upravljač, kao što je objašnjeno u uputama za upravljanje Thymio wireless-a. Konačno, potrebno je namjestiti Thymia u ljubičasti način rada.

Crveni i plavi duh imaju isti način pripreme. Budući da ih igrač ne kontrolira, nije potrebno ništa drugo, osim pripremanja programa. Na oba Thymia treba učitati program "Duh.aesl".

Konačno, treba postaviti robote na njihove početne pozicije. Po slici 22, „Pac-man“ se postavlja u donjem desnom kutku, a duhovi jedan kraj drugog, na lijevoj strani mape. Za pokretanje igre, samo je potrebno kliknuti gumb „Run“ na Thymio Suite aplikaciji.

TOK IGRE

Igrač upravlja „Pac-man“-ovo gibanje gumbima u obliku strelica na daljinskom upravljaču. Cilj jest preživjeti što više, prije nego li duhovi uhvate igrača. Oni se gibaju nasumično po mapi te se ubrzavaju što igra traje dulje. Lako je raspoznati robote na sceni; svaki je osvijetljen određenom bojom kao u originalnoj igri.

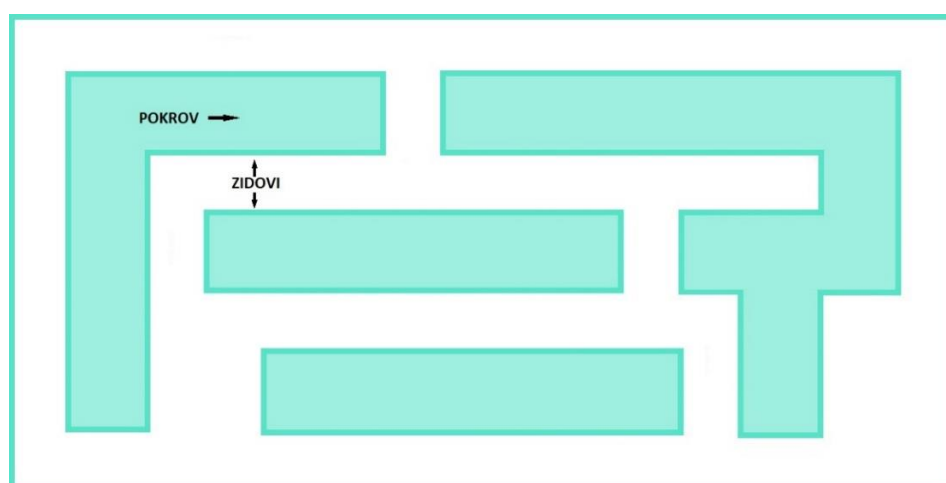
4. Daljnja optimizacija

Budući da projekt nije izrealiziran kako je bilo početno zamišljeno, postoji mnogo mogućnosti optimizacije. Očito je da fali način bodovanja te brojanja „života“. To je moguće izvesti pomoću Arduino Uno mikrokontrolera i ultrazvučnih senzora, postavljenih na određenim dijelovima mape. Za igrača, prikaz bodova (točkica) se ostvaruje LED svijetlima. Sve u konačnici mora biti povezano s mikroračunalom, koji će gledati je li robot „skupio“ bod te onda ugasiti određeno LED svijetlo. Kada duh susretne igrača, gubi se jedan „život“. Igrač ga može i dobiti, tako da preživi na neko određeno vrijeme.

Nakon uspostavljenog bodovnog sustava, možemo i dodati „težine“ u igri. Ili kao početni odabir igrača, koji se ne mijenja tijekom cijele igre, ili kao napredovanje igrača, tako da svaki nivo otežava igru.

Nadalje, možemo popraviti način upravljanja „Pac-man“-a. Daljinski upravljač je jako neudoban te ima previše nepotrebnih gumbi koji samo zbunjuju igrača. Najbolja alternativa jest joystick, koji je stvoren za igre. No, za tu realizaciju, potrebne su dodatne komponente, kao npr. USB IR Toy, koje će uspostaviti komunikaciju između joystick-a i robota.

Tanki zidovi od stirodura, iako su lako prenosivi i jeftini, dosta su neefikasni u igri. Ako se robot zabije u njih, odmah padnu, lako se slome te vizualno nisu atraktivni. Bolja alternativa bi bila da se naprave od čvršćeg (onda i skupljeg) materijala te da se izradi ploča koja će zidove učiniti punijima kao na slici 36. Time se više stabiliziraju zidovi i logo upada više u oči.



Slika 36 Optimizacija zidova

Najčešća izolir-traka na tržištu širine je 1.9 cm. S obzirom da nije dovoljne širine da ju Thymio prepozna, mora se stavljati više puta na istom hodniku, što dovodi do nesavršenosti linije, velikog troška vremena i novaca. Isplativiji način jest nabavljanje izolir-trake širine 5 cm, jer će se time uštedjeti na vremenu i novcu.

5. Literatura

- „Pac-man“

<https://en.wikipedia.org/wiki/Pac-Man>

<https://dev.to/code2bits/pac-man-patterns--ghost-movement-strategy-pattern-1k1>

<https://gameinternals.com/understanding-pac-man-ghost-behavior>

<https://pacman.live/play.html>

- Thymio

<http://wiki.thymio.org/en:thymiospecifications>

<https://www.thymio.org/>

<https://hallab.cs.dal.ca/images/e/e2/Tutorials1.pdf>

<http://wiki.thymio.org/en:thymiostarting>

<https://www.thymio.org/help/>

<http://wiki.thymio.org/en:start>

<http://wiki.thymio.org/en:asebausermanual>

<http://aseba.wikidot.com/en:thymiosoundrecorder>