

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Razvoj mozga jednostavnog robota Robby

Studentski tim: Dan Ambrošić
Stjepan Dugonjić
Mihael Međan
Luka Suman
Nikola Zadavec

Nastavnik: doc. dr. sc. Marko Čupić

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Sadržaj

1. Opis razvijenog proizvoda	3
1.1 Opis problema	3
1.2 Detaljniji opis problema	3
2. Tehničke značajke	6
2.1 Algoritmi	6
2.1.1 Genetsko algoritam	6
2.1.2 Genetsko programiranje	6
2.1.3 Unaprijedna neuronska mreža	7
2.1.4 Elman-ova neuronska mreža	9
2.1.5 Podržano učenje (Q-Learning)	11
2.2 Korištene tehnologije	13
3. Upute za korištenje	14
4. Literatura	21

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Tehnička dokumentacija

1. Opis razvijenog proizvoda

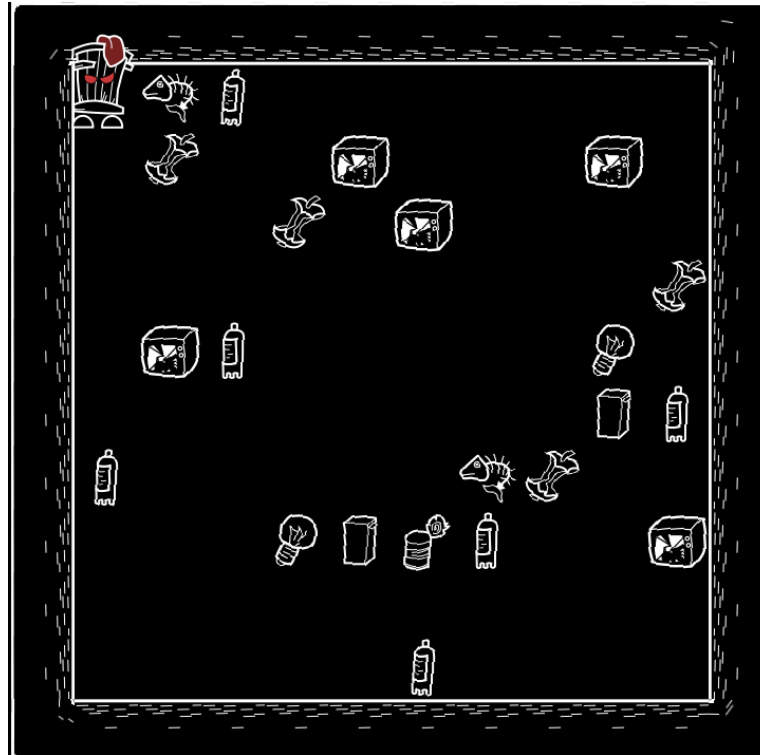
1.1 Opis problema

U knjizi Complexity: A Guided Tour, autorica Melanie Mitchell navodi i analizira niz primjera kompleksnih sustava i traži odgovor na pitanje: "kada je cjelina 'veća' od sume sastavnih dijelova?", odnosno kada sustav (biološki, tehnički, kemijski, ...) počinje iskazivati ponašanje koje nadilazi ponašanje njegovih sastavnih dijelova - postaje kompleksan. U toj knjizi prikazan je zgodan eksperiment: razvoj "mozga" jednostavnog robota nazvanog Robby. Uz razvoj "mozga" robota, implementirat će se i grafičko korisničko sučelje koje omogućava vizualizaciju postupka učenja te vizualizaciju naučenog ponašanja. Kao programski jezik za implementaciju rješenja koristi se Java.

1.2 Detaljniji opis problema

Robby je virtualni robot koji živi u 2D svijetu prožetom smećem. Posao mu je očistiti svoj svijet skupljajući smeće. Robby-jev svijet se sastoji od 100 kvadrata (polja) postavljenih u 10 x 10 mrežu.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 1. Robby skuplja smeće

Možete vidjeti Robby-ja na polju (0,0). Zamislamo da postoji zid oko rubova cijele mreže. Razna polja su popunjena sa smećem (ne više od jednog po polju). Robby nije vrlo inteligentan i njegov vid nije toliko dobar. Iz polja u kojem je, on može vidjeti sadržaj jedne susjedne stranice na sjeveru, jugu, istoku i zapadu, te sadržaj polja kojeg zauzima. Polje može biti prazno, može sadržavati smeće ili biti zid. Robby, na polju (0,0) vidi da je njegova trenutna stranica prazna (ne sadrži smeće), polja na sjeveru i zapadu su zidovi, polje na jugu je prazno, a polje na istoku sadrži smeće.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Za svaki interval čišćenja, Robby može obaviti određeni broj akcija (npr. 200). Svaka akcija se sastoji od jednog od sljedećih sedam izbora: prelazak na sjever, prelazak na jug, prelazak na istok, prelazak na zapad, odabrati slučajni smjer za prelazak, ostati na mjestu, ili pokupiti smeće na trenutnom polju. Svaka akcija može rezultirati nagradom ili kaznom. Ako Robby pokuša pokupiti smeće na polju koje sadrži smeće, dobije 10 bodova, ako pokuša pokupiti smeće na praznom polju, kažnjen je sa 1 bodom te ako se zabije u zid, kažnjen je sa 5 bodova i ostaje na polju u kojem je bio prije sudara. Očito, Robby-jeva nagrada je najveća kada pokupi što veći broj smeća, bez da se sudara u zid ili pokušava dohvatiti smeća u praznom polju.

Na prvi pogled, ovaj problem se čini jednostavnim, ali moramo uzeti u obzir ograničenja koja Robby ima. Broj različitih situacija u kojima se Robby može u bilo kojem trenutku naći je 243 (5 polja sa 3 moguća sadržaja). Za svaku od tih situacija on može odabrati jednu od 7 akcija što rezultira ogromnim brojem mogućih konfiguracija tj. strategija sakupljanja smeća (7^{243} je približno $2 \cdot 10^{205}$).

¹ $2 \cdot 10^{205}$ je jako veliki broj. Kada bi mogli u svakoj nanosekundi testirati jednu konfiguraciju, trebalo bi nam $7 \cdot 10^{179}$ milijardi godina. Za usporedbu, svemir je star "samo" 13,7 milijardi godina.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

2. Tehničke značajke

2.1 Algoritmi

2.1.1 Genetski algoritam

Najjednostavniji model robota je da svakoj mogućoj percepciji pridružimo jednu akciju. Akcije numeriramo od 0 do 6 i jedan robot u memoriji predstavlja polje brojeva (243 elemenata), gdje na indeksu i stoji akcija koju robot napravi ako vidi percepciju i . Sada je jedini problem odrediti koje akcije treba pridružiti kojim percepcijama da bi robot davao što bolje rezultate.

Genetski algoritam je pogodan za jako veliki broj problema pa ga tako možemo primijeniti i na ovaj. Korištena je obična verzija generacijskog genetskog algoritma. Početna populacija se stvori nasumično (svakoj percepciji se pridruži nasumična akcija) i pokrene se algoritam. Operator selekcije je turnirska, križanje roditelja je izvedeno jednostavnim križanjem s jednom točkom prekida i mutacija (s malom vjerojatnosti) pridruži nekoj percepciji nasumičnu akciju.

Algoritam radi dok se ne zadovolji uvjet zaustavljanja (dostignut željeni fitness ili maksimalni broj generacija) te se robot može dosta dobro istrenirati u relativno kratkom vremenu.

2.1.2 Genetsko programiranje

Genetsko programiranje je tehnika s kojom se evoluiraju računalni programi primjenom evolucijskog algoritma. Programi su predstavljeni kao stabla koja u svojim čvorovima mogu sadržavati funkcije ili terminalne vrijednosti. Prvo se generira početna populacija programa tako da se nasumično slažu čvorovi u stabla. Pritom se pazi na dubinu stabla (ili broj čvorova) te na ispravan ulaz u funkcije. Proces evolucije sastoji se od 4 operacije: križanje (eng. crossover), mutacija, reprodukcija te promjena arhitekture. Prilikom odabira roditelja za sljedeću generaciju, koristi se turnirska selekcija. Križanje odabere dva roditelja, nasumično odabere podstablo u svakome te ih zamijeni. Mutacija odabere podstablo na jednom roditelju te ga nanovo generira. Reprodukcija jednostavno

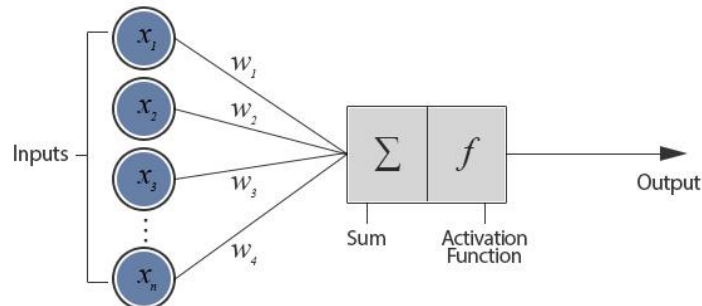
Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

kopira roditelja u sljedeću generaciju. Promjena arhitekture temeljito promijeni strukturu stabla. Od operacije se najviše koristi križanje (>90%), mutacija (1-2%), promjena arhitekture (1%) te reprodukcija (ostatak). Više je primjera u stvarnom svijetu gdje GP dovodi do rezultata koji konkuriraju ljudskim rezultatima, a u nekim slučajevima čak i bolje.

2.1.3 Unaprijedna neuronska mreža

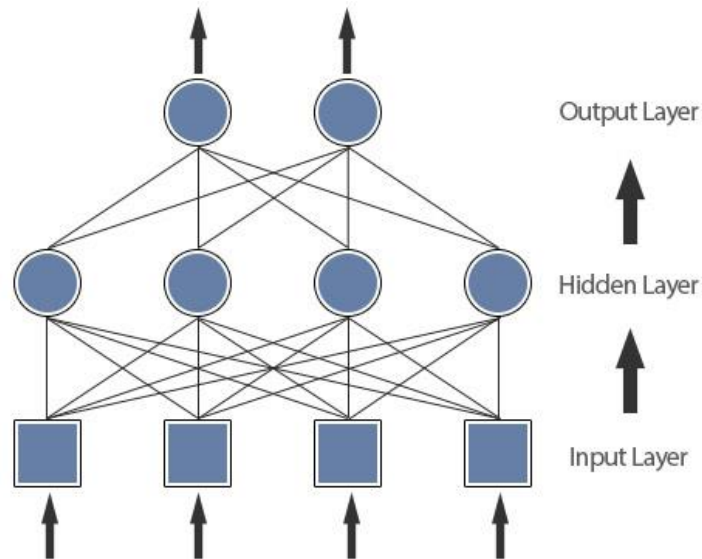
Koristili smo potpuno povezanu unaprijednu neuronsku mrežu koju smo učili pomoću generacijskog genetskog algoritma. Svaka veza između dva neurona je predstavljena nekim realnim brojem. Polje tih veza je bio genotip jedinke. Neuroni u mreži su kao aktivacijsku funkciju koristili sigmoidalnu funkciju, a kao transfer funkciju su koristili sumu svih ulazni veza pomnoženih sa pripadajućim težinama.

$$f(x) = \frac{1}{1 + e^{-net}}, net = \sum_{i=1}^n w_i x_i$$



Slika 2.1.3.2 Model neurona neuronske mreže

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 2.1.3.3 Potpuno povezana unaprijedna neuronska mreža

Korištena neuronska mreža sadrži devet neurona u ulaznom, devet neurona u skrivenom i sedam neurona u izlaznom sloju.

Ulazi u neuronsku mrežu Robot vidi neku percepciju iz svijeta oko sebe na temelju koje se treba odlučiti koju akciju sljedeću napraviti. Za svaki od susjednih polja koja robot vidi smo koristili dva neurona.

Vrijednosti neurona:

- smeće na polju – (1, 0)
- prazno polje – (0, 0)
- zid na polju – (0, 1)

Za polje na kojem se nalazi jedan (zbog ne mogućnosti pojave zida na tom polju).

Vrijednosti neurona:

- smeće na polju – (1)
- prazno polje – (0)

Izlazi iz neuronske mreže

Već je spomenuto da je 7 neurona korišteno u izlaznom sloju, zbog 7 mogućih

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

akcija koje robot može napraviti. Svaki neuron predstavlja jednu akciju, a napravi se ona koja koju predstavlja neuron s najvećom vrijednosti na izlazu.

Jednostavna reprezentacija (podatkovna struktura) neuronske mreže kao jedinke nam je omogućila lako korištenje već postojećih operatora križanja i mutacije.

Najbolje rezultate smo dobili korištenjem operacija:

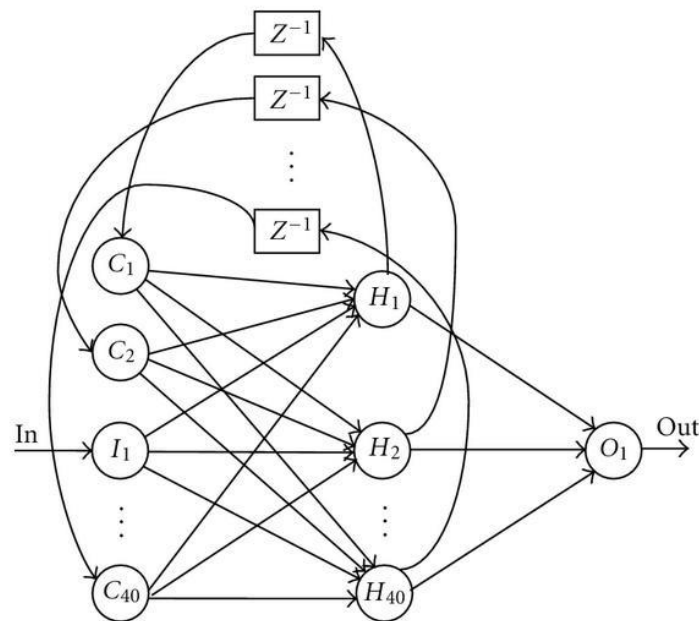
- turnirska selekcija
- diskretno križanje (komponente su ulazne težine pojedinog neurona i stvara dva djeteta)
- mutacija svakog kromosoma jedinke s fiksnom vjerojatnošću dodavanjem slučajne vrijednosti dobivene iz Gauss-ove distribucije

Važno je napomenuti da ova pristup razboju „mozga“ robota daje jednako dobre rezultate kao i uporaba genetskog algoritma uz izravno kodiranje akcija, a vrijeme treninga traje duže, ovisi o više parametara te je stoga bilo teže naći najprikladnije.

2.1.4 Elman-ova neuronska mreža

Koristili smo Elman-ovu neuronsku mrežu koju smo učili pomoću generacijskog genetskog algoritma. Veze između dva neurona su jednako predstavljane i aktivacijska i transfer funkcija su iste kao i kod unaprijedne neuronske mreže.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 2.1.4.1 Elman-ova neuronska mreža

Glavna razlika između unaprijednih i rekurzivnih neuronskih mreža je ta da izlaz ne ovisi samo o trenutno predodčenom ulaznom uzorku već i o stanju u kojem se mreža nalazi. Kod Elman-ove mreže to je stanje eksplicitno pohranjeno u kontekstnom sloju. Taj sloj je proširenje ulaznog sloja, a u njemu se nalazi upravo onoliko neurona koliko ih postoji i u prvom skrivenom sloju i oni pamte izlaze neurona tog sloja kakvi su bili nakon predodčavanja prethodnog ulaznog uzorka.

Korištena neuronska mreža sadrži devet neurona u ulaznom, devet neurona u skrivenom i sedam neurona u izlaznom sloju (na prvi pogled čini se isto kao i kod unaprijedne mreže no nemojmo zaboraviti na kontekstni sloj, iako postoji, ovdje nije bio ubrojen, jer je pretpostavljen).

Ulazi i izlazi u neuronsku mrežu su jednaki kao i kod unaprijedne mreže, a tako i reprezentacija mreže.

Najbolje rezultate smo dobili korištenjem operacija:

- turnirska selekcija
- BLX-alfa križanje
- mutacija svakog kromosoma jedinice s fiksnom vjerojatnošću dodavanjem slučajne vrijednosti dobivene iz Gauss-ove distribucije

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Kako ovaj tip mreže ima „mogućnost pamćenja“ (pohranjivanja stanja) Robby više nije state-less robot što znači da za određenu percepciju on u različitim stanjima može raditi različite akcije. Zbog toga se ovaj pristup razlike od svih ostalih i daje najbolje rezultate bez obzira na trajanje treniranja.

2.1.5 Podržano učenje (Q-Learning)

Algoritam se sastoji od agenta (u našem slučaju robota koji se trenira), skupa stanja u kojem se agent može naći (u našem problemu to je skup svih mogućih percepcija na koje robot može naići) te skupa akcija koje agent može poduzeti u određenom stanju.

Izvršavanjem akcija agent se kreće između stanja te kao povratnu informaciju dobiva nagradu ili kaznu (predstavljenu kao pozitivan, odnosno negativan broj).

Svaki par (*stanje*, *akcija*) dodatno ima pridruženu numeričku „Q“ vrijednost koja označava njegovu dobrotu.

Treniranje agenta vrši se kroz određeni broj epizoda, gdje svaka epizoda uključuje izvođenje određenog broja akcija (npr. 200) na određenoj mapi (svijetu). Agent akcije bira po „*epsilon-greedy*“ načelu, odnosno s vjerojatnošću „*epsilon*“ akciju odabire nasumično, a sa suprotnom vjerojatnošću odabire akciju takvu da je $Q(\text{trenutno_stanje}, \text{akcija})$ maksimalno.

Cilj algoritma je kroz niz epizoda ažurirati Q vrijednosti svakog para (*stanje*, *akcija*) kako bi agent dobio mehanizam odabira optimalne akcije u svakom stanju.

Q vrijednosti svih parova (*stanje*, *akcija*) na početku su inicijalizirane na 0, a agent vrši ažuriranje po sljedećoj formuli:

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t}_{\text{learning rate}} \cdot \left(\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

- „*learning rate*“ je broj iz $[0, 1]$ koji govori u kojoj mjeri novostečene informacije utječu na promjenu stare vrijednosti. U našoj implementaciji korišten je statički faktor od 0.01.
- „*discount factor*“ je broj iz $[0, 1]$ koji iskazuje važnost *budućih* nagrada (donekle sprječava greedy ponašanje agenta). U našoj implementaciji korišten je statički faktor od 0.8.
- nagrada „*r*“ je unaprijed određena, a mi smo koristili sljedeće vrijednosti:
 - +100 za ispravno skupljanje smeća
 - -50 za zabijanje u zid
 - -10 za skupljanje smeća na praznom polju
 - -5 za „*do nothing*“ akciju
 - -1 za bilo koju akciju kretanja

Nakon završetka procesa treniranja, agent se postavlja na početnu poziciju mape (0, 0) te jednostavno odabire akcije sa najvećim Q-vrijednostima. Takav redoslijed izvođenja akcija predstavlja rješenje našeg problema.

Korisniku se kao mogućnost daje proizvoljan odabir broja epizoda treniranja uz zadanu traženu dobrotu (ukoliko se tražena dobrota postigne prije zadanog broja epizoda, treniranje se završava) te mogućnost mijenjanja „*života*“ skupa mapa nad kojim se robot evaluira.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

2.2 Korištene tehnologije

1. Programski jezik Java 8
2. libGDX
3. Git (BitBucket)

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

3. Upute za korištenje



Slika 4. Početni izbornik

Klikom na gumb EXIT program se gasi.

Pod opcijama (gumb OPTIONS) mijenjamo parametre simulatora:

- Trash – određuje vjerojatnost da pojedino polje sadrži smeće
- Actions – određuje broj akcija za svaki interval čišćenja tj. pokretanje simulacije
- Reward – nagrada (dodavanje bodova) koje robot dobije za svako pokupljeno smeće
- Fine Can – kazna (oduzimanje bodova) ako robot pokuša pokupiti smeće na praznom polju
- Fine Wall - kazna (oduzimanje bodova) ako robot udari (pokuša se pomaknuti) u zid

Klikom na gumb BACK aplikacija se vraća natrag na početni meni.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 5. Izbornik postavki



Slika 6. Konfiguracija treninga

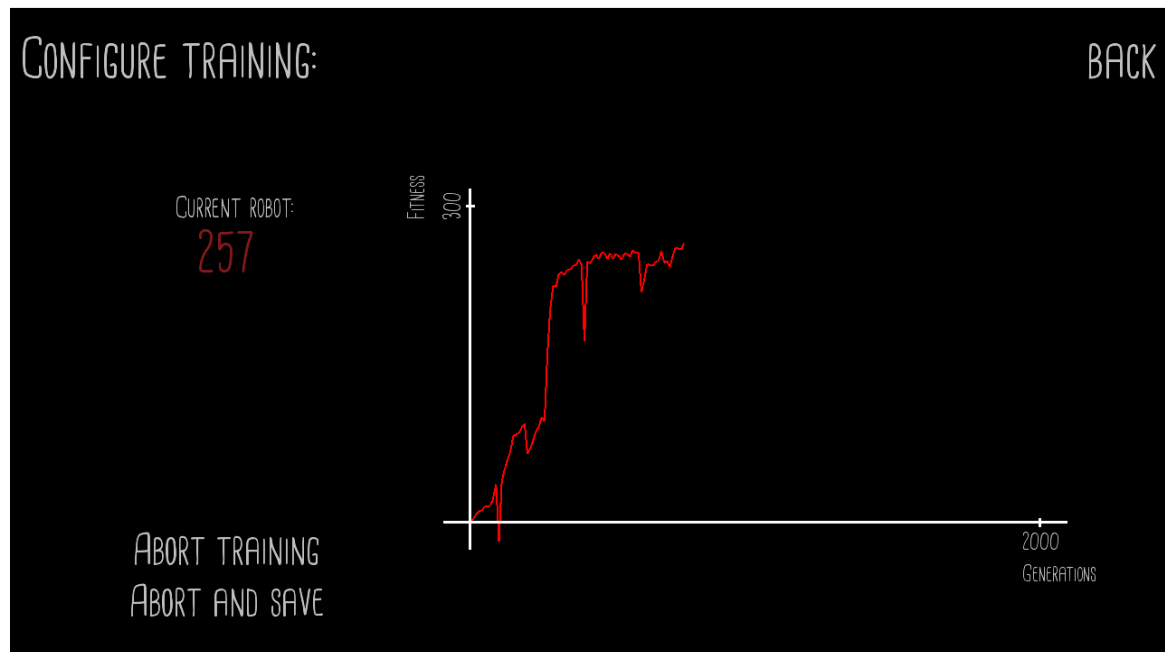
Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Klikom na gumb TRAIN ROBOT otvara se prozor u kojem se može izabrati način učenja (algoritam) Robby-jeva “mozga” (strategije skupljanja smeća). Detaljnije o svakom algoritmu pročitajte u nastavku.

U popisu parametara za treniranje moguće je podesiti sljedeće parametre:

- Generations – broj generacija
- Population Size – veličina populacije u svakoj generaciji
- Number Of Maps – broj mapa na kojima se računa prosječni fitness svake jedinke iz populacije
- Map Set Life – broj generacija nakon kojih se generira novi skup mapa
- Target Fitness – cilj treniranja, ovisi o postotku smeća na mapi postavljenog u opcijama

Treniranje počinje klikom na gumb START TRAINING. Parametre više nije moguće mijenjati te se umjesto njih prikazuje fitness trenutno najboljeg robota nakon svake generacije.



Slika 7 Tijek treniranja

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

Graf prikazuje fitnes (y-koordinata) najboljeg robota u svakoj generaciji (x-koordinata).

Treniranje se zaustavlja kada prođu sve generacije ili ako se dosegne Target Fitness kojeg je korisnik prethodno definirao kao parametar.

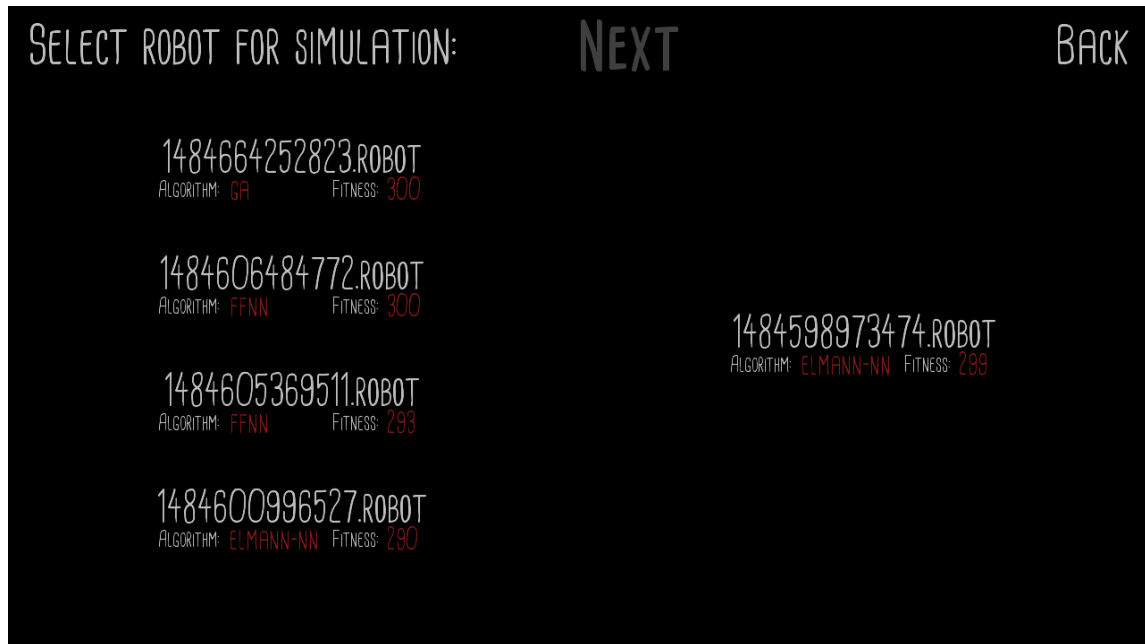
Korisnik u bilo kojem trenutku može zaustaviti treniranje jednim od sljedeće dva načina:

1. Klikom na gumb ABORT AND SAVE treniranje se zaustavlja te se najbolji robot sprema.
2. Klikom na gumb ABORT TRAINING zaustavlja se treniranje, a najbolji robot se odbacuje tj. ne sprema se.

Napomena: algoritam Q-Learning ima dodatni parametar "Number Of Episodes" umjesto parametara "Generations" i nema "Population Size".

Spremanje robota se vrši u direktorij „robots“ u radnom direktoriju.

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 8. Odabir robota

Simulacija se pokreće iz glavnog izbornika klikom na gumb RUN SIMULATION. Korisnik zatim bira jednog od ponuđenih robota koji su prethodno istrenirani (učitani iz direktorija „robots“ u radnom direktoriju). Nakon toga se mora kliknuti na gumb NEXT pri čemu se otvara prozor za odabir mape (LOAD WORLD).

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 9. Odabir mape

Na lijevoj strani se nalazi izbornik, a na desnoj strani je grafički prikaz trenutne mape.

- CANS prikazuje broj komada smeća na trenutno prikazanoj mapi.
- SAVED MAPS prikazuje spremljene mape u direktoriju „maps“ koje se mogu učitati.
- Postoje dva gumba prikazana slikama:
 - igrača kocka – stvori mapu sa nasumičnim rasporedom smeća (sadržaj svakog polja ovisi o parametru Trash zadan u opcijama)
 - disketa – sprema trenutno prikazanu mapu u direktorij „maps“
- START SIMULATION pokreće simulaciju
- CANCEL odbacuje promjene i vraća se na prethodni prozor

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.



Slika 10. Simulacija

Mogućnosti tijekom simulacije:

- SIMULATION SPEED – podešava brzinu kretanja robota
- CANDY MODE – korisnici se ohrabruju da sami isprobaju ovu opciju
- ROBO VISION – prikazuju se sadržaji samo onih polja koje robot vidi oko sebe
- CANS REMAINING – broj preostalog smeća
- CANS COLLECTED – broj prikupljenog smeća
- MOVES LEFT – broj preostalih akcija
- BACK TO MAPS – zaustavlja se simulacija i vraća se na prozor za odabir mape

Razvoj mozga jednostavnog robota Robby	Verzija: 1.0
Dokumentacija	Datum: 17.01.2017.

4. Literatura

1. Čupić - Prirodom inspirirani optimizacijski algoritmi. Metaheuristike (2003)
2. Čupić, Bašić, Šnajder - Umjetne neuronske mreže (2008)
3. Čupić, Dalebo Bašić, Golub – Neizrazito, evolucijsko i neuroračunarstvo (2013)
4. Mitchell – Complexity: A Guided Tour (2011)