

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4373

**PROGRAMSKO SUČELJE ZA SIMULACIJU
UMJETNOG ŽIVOTA**

Toni Vlaić

Zagreb, lipanj 2016.

Sadržaj

1. Uvod	1
1.1 Povijest umjetnog života	2
1.2 Vrste umjetnog života	2
1.3 Umjetne neuronske mreže	3
2. Simulatori umjetnog života	5
2.1 3D simulator umjetnog života Critterding	6
2.1.1 Ekosustav	7
2.1.2 Jedinke.....	8
2.1.3 Statistika.....	10
2.2 Simulator 3D Virtual Creature Evolution (3DVCE).....	11
2.2.1 Ekosustav	11
2.2.2 Jedinke.....	15
3. Praktičan rad	17
3.1 Tehnologije	17
3.1.1 Java 8	17
3.1.2 JavaFX	17
3.2 Program za simulaciju	18
3.2.1 Početni prozor	18
3.2.2 Prozor za definiciju parametara	19
3.2.3 Simulacijski prozor.....	20
3.2.4 Demo simulacija	23
5. Budući rad.....	26
6. Zaključak	27
7. Literatura	28

1. Uvod

Umjetni život (engl. *Artificial Life - ALife*) je područje istraživanja koje promatra sustave vezane uz život, životne procese i njegovu evoluciju korištenjem simulacija uz pomoć računalnih modela, robotike i biokemije. Umjetni život nastoji imitirati znanje tradicionalne biologije i tako rekreirati određene biološke fenomene.

Danas postoje dva filozofska pogleda na umjetni život, slab i jaki umjetni život. Zagovaratelji slabog umjetnog života gledaju na inteligentne sustave kojima se dodaju aspekti ljudskog ponašanja kao pokušaj da se nauči više o biološkom životu, no te modele ne smatraju živima. S druge strane zagovaratelji jakog umjetnog života smatraju kako bi stvorenii sustavi mogli replicirati sva svojstva biološkog ponašanja i biti smatrani živima poput bioloških organizama.

Umjetnu inteligenciju simuliranih organizama moguće je ostvariti pomoću nekog od evolucijskih algoritama ili neuronском mrežom (više o neuronским mrežama u poglavljiju 1.3). Evolucijski algoritmi su populacijski algoritmi inspirirani biološkom evolucijom, pa tako implementiraju metode križanja, mutacije i selekcije pri stvaranju novih jedinki u populaciji.

Najčešći algoritam iz skupine evolucijskih za simulaciju inteligencije u umjetnom životu je genetsko programiranje. Zadatak genetskog programiranja je stvoriti skup gena, često u memoriji reprezentiran kao stablo, koji dobro rješava zadani zadatak. Početna, nasumično stvorena populacija rješenja se korištenjem selekcije, križanja, mutacije i evaluacije pokušava evoluirati kako bi se dobilo rješenje koje uspješno rješava zadani zadatak. Svaki čvor u stablu predstavlja neku funkciju, dok su listovi stabla operandi. Moguće je modelirati čvorove kao uvjete koji se provjeravaju te se ovisno o njima bira njihova određena grana, a listove stabla kao akcije jedinke. Jedinka, čije je ponašanje oblikovano takvim stablom, ostvaruje se neprestanim izvršavanjem opisanog stabla koje se može promatrati i kao njegov "DNA".

1.1 Povijest umjetnog života

Sredinom 19. stoljeća Darwin je pokušao objasniti evoluciju organizama u knjizi "Podrijetlo vrsta" koja je izdana 1859.g. u kojoj iznosi premisu kako sav živi svijet ima zajedničkog pretka te se nove vrste razvijaju procesom prirodne selekcije.

Pod utjecajem Darwina, Samuel Butler se prvi počeo zalagati za mogućnost razvoja svijesti u strojevima korištenjem prirodne selekcije u svojoj knjizi "*Erewhon*". Kasnije, sredinom 20. stoljeća von Neumann pod utjecajem prijatelja matematičara Stanisława Ulama i univerzalnog Turingovog stroja razvija prvi samorazmnožavajući stanični automat čime je postavio temelje za računalno usmjereni pristup umjetnom životu [1]. Umjetna inteligencija, kao znanstveno područje, nastaje nekoliko godina kasnije (1955.g.), a John McCarthy, ujedno jedan od osnivača područja, opisao je umjetnu inteligenciju kao znanost o izradi intelligentnih strojeva. Dvadeset godina kasnije Chris Langton upotrijebio je pojam umjetnog života za opis svojih eksperimenata kako se računala mogu iskoristiti za modeliranje bioloških sustava. Konačno na konferenciji održanoj na MIT-u 1994., umjetni život je osnovan kao novo područje istraživanja od strane Langtona, Raya i ostalih.

1.2 Vrste umjetnog života

Umjetni život može biti svrstan u "soft" (engl. *Software*), "hard" (engl. *Hardware*) ili "wet" (engl. *Wetware*) umjetni život, a određen je vrstom medija koju znanstvenici koriste prilikom izrade umjetnog života [2].

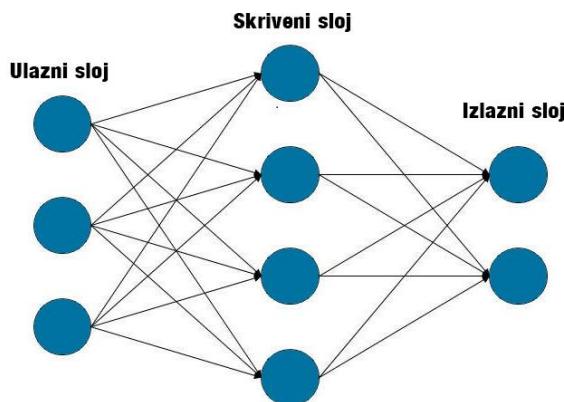
- *Soft* umjetni život ukorijenjen je u računarsku znanost i predstavlja ideju da je život karakteriziran svojom organizacijom, a ne materijom. Dakle, moguće je izraditi virtualni ekosustav u kojem male komponente programa predstavljaju organizme ekosustava koji su u međusobnoj interakciji.

- *Hard* umjetni život se uglavnom sastoji od automatski upravljenih strojeva koji mogu samostalno odrađivati poslove. Ovaj pristup navodi na to da je za izgraditi inteligentan sustav nužno imati njegovu reprezentaciju u fizičkom svijetu.
- *Wet* umjetni život je najbliži stvarnoj biologiji. Zasniva se na znanstvenom pristupu koji provodi eksperimente korištenjem populacije organskih makromolekula u tekućem mediju, kako bi se promatrala samoorganizirajuća svojstva molekula.

1.3 Umjetne neuronske mreže

Umjetna neuronska mreža (engl. *Artificial Neural Network – ANN*) je skup umjetnih neurona raspodijeljenih po slojevima, a uređena je po uzoru na ljudski mozak. ANN se sastoji od jednog ulaznog sloja i jednog izlaznog sloja i proizvoljnog broja skrivenih slojeva. Svaki od slojeva načinjen je od jednog ili više neurona. Slojevi su međusobno povezani vezama kroz koje razmjenjuju signale.

Glavna razlika između vrsta mreža jest u njihovoј arhitekturi. Mreže se dijele na one s vezama isključivo prema idućem sloju (engl. *Feedforward Artificial Neural Network – FFNN*, Slika 1.1) i povratne mreže sa zatvorenim ciklusom koji vraća rezultat analize na ulaz čime se proces uvježbavanja ponavlja (engl. *Recurrent Neural Networks*).



Slika 1.1 Feedforward Neural Network

Umjetne neuronske mreže se odlikuju svojstvom učenja. Podaci ulaze u neuronsku mrežu preko ulaznog sloja, prolaze kroz neurone skrivenih slojeva i na kraju dolaze do izlaznog sloja s nekom vrijednošću (Slika 1.1). Svaki od poveznica između neurona ima svoju "težinu" koja određuje kako se mijenjaju podaci koji prolaze kroz neuronsku mrežu.

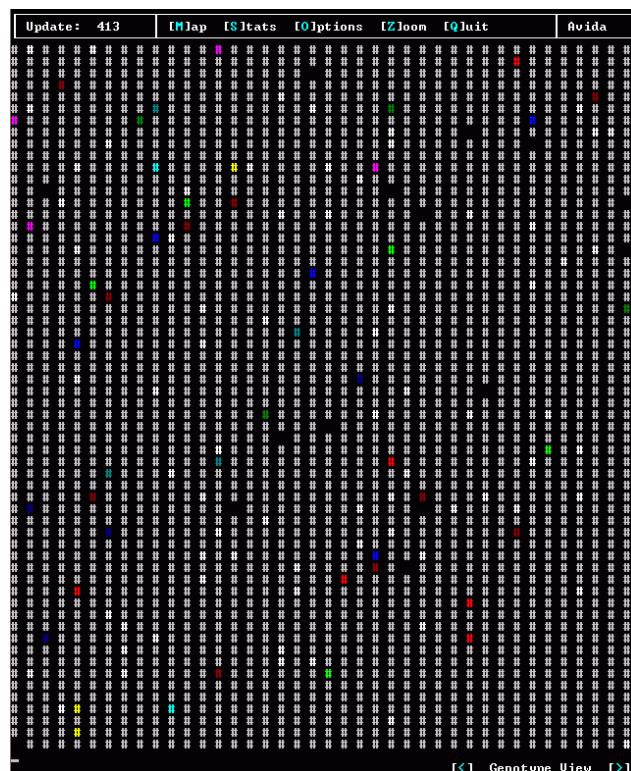
Učenje neuronskih mreža svodi se na manipulaciju težina između neurona dok se ne dobije željeni rezultat na izlazu. Više je strategija učenja kojim se neuronske mreže mogu istrenirati, pa tako postoji nadgledano, nenadgledano i potporno učenje. Nadgledano učenje neuronsku mrežu trenira s postojećim skupom podataka s točnim odgovorima, a neuronska mreža ažurira svoje težine između neurona kako bi se njeni izlazi podudarali s očekivanim izlazima. Nenadgledano učenje je pristup u kojem postoji skup podataka, no oni ne sadrže točne odgovore. Primjer nenadgledanog učenja bilo bi traženje skrivenog uzorka u skupu podataka na osnovu kojeg će se nad podacima izvršiti grupiranje (engl. *Clustering*). Potporno učenje svoje podatke dobiva interaktivno iz okoline te se neuronska mreža nakon donesene odluke "kažnjava", odnosno "nagrađuje" i tako se neuronska mreža usmjerava prema točnom rješenju.

Neuronske mreže također je moguće upotrijebiti za simulaciju umjetnog života, odnosno modeliranje umjetne inteligencije agenata neuronskom mrežom koja bi utjecala na njihovu interakciju s ekosustavom u kojem se nalaze. Jedinke bi svojim razmnožavanjem prenosile karakteristike koje su stekle na svoje potomstvo (Lamarckovo nasljeđivanje) uz malu šansu mutacije koja bi dovela do stvaranja nove vrste. Sposobnije jedinke bi se brže razmnožavale i njihova populacija bi dominirala ekosustavom dok bi manje sposobne jedinke izumrle.

2. Simulatori umjetnog života

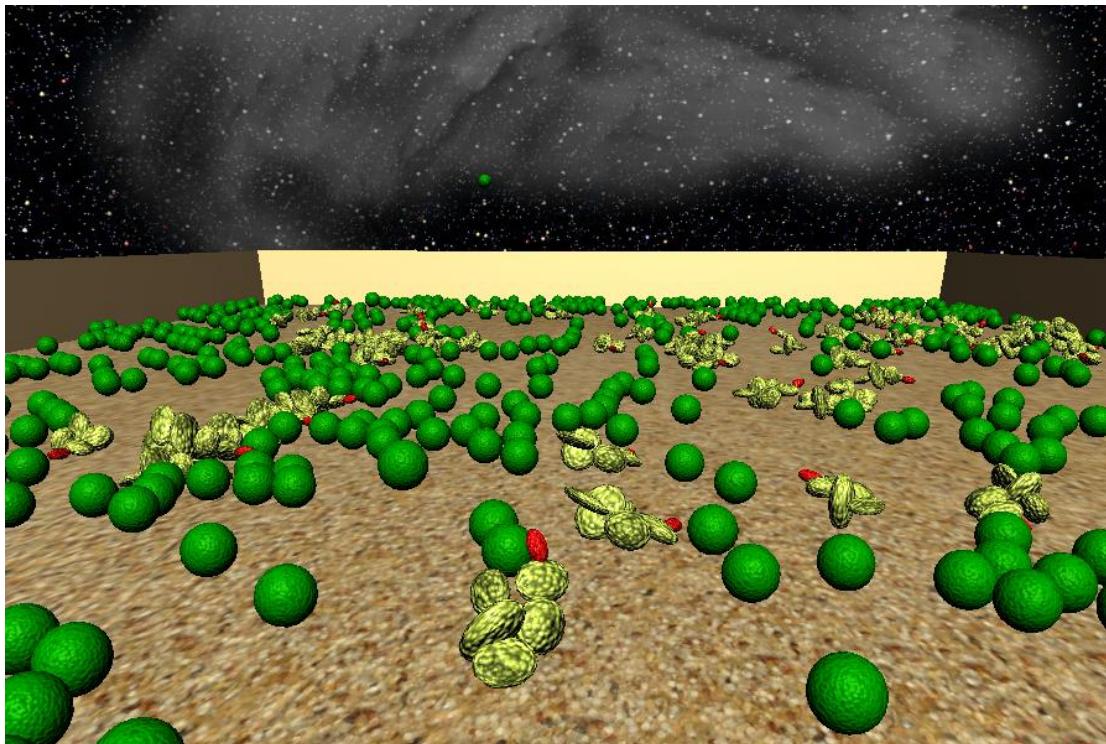
Simulatori umjetnog života stvaraju ekosustav, odnosno virtualno okruženje u kojem žive umjetni organizmi. Oni nam omogućavaju postavljanje parametara kojima ćemo modelirati ekosustav, koji uvelike utječe na razvoj jedinki koji njime obitavaju. Modeliranjem okoline uvjetujemo koja su optimalna svojstva jedinke u takvom okruženju. Primjerice, ako oblikujemo ekosustav s neravnim terenom jedinke koje njime obitavaju će biti uspješnije ako evoluiraju kako bi se efikasnije kretale po zahtjevnom području.

Među prvim simulatorima umjetnog života nastao je *Avida* 1993. godine, razvili su ga Charles Ofria, C. Titus Brown i Chris Adami te se još uvijek unaprjeđuje. *Avida* je inspirirana *Tierra* simulatorom kojeg je napravio Thomas S. Ray u ranim 90-ima. *Tierra* je simulirala računalne programe koji su se natjecali za pristup memoriji i procesorsko vrijeme te su imali mogućnost mutacije, rekombiniranja i samorazmnožavanja. *Avida*, za razliku od *Tierra*, svakoj jedinki pridružuje njen privatni blok memorije i izvršava ju na zasebnom virtualnom procesoru. Druga karakteristika *Avide* je što organizmi mogu izvršavati svoj kod različitim brzinama koji je određen parametrima i zadacima koje organizam izvršava.



Slika 2.1: Avida pogled mape [5]

Noviji simulatori omogućuju veći opseg parametara kojim se oblikuju organizmi, ekosustav i veći broj jedinki s kompleksnijom umjetnom inteligencijom. Primjer takvog simulatora je *Critterding* koji uz 3D vizualizaciju jedinki i ekosustava omogućuje direktnu interakciju korisnika s jedinkama.



Slika 2.2:Pogled ekosustava Critterding

2.1 3D simulator umjetnog života Critterding

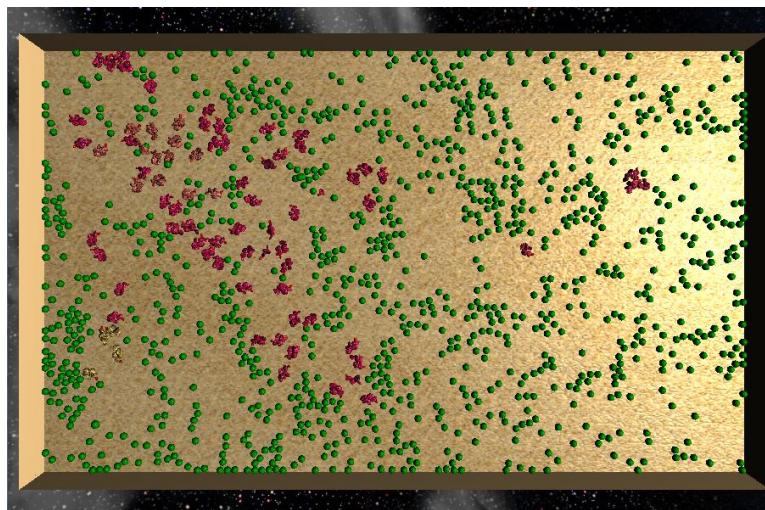
Critterding je 3D simulator umjetnog života s implementiranom fizikom kojeg je napravio Bob Winckelman 2005. godine, a njegov razvoj i dalje traje. Jedinke se nazivaju "critteri", a čine ih dijelovi tijela koji su povezani zglobovima. Svaka jedinka ima glavu prikazanu crvenom bojom, koja omogućava hranjenje i percepciju okoline.

Pri pokretanju simulatora nasumično se stvaraju jedinke i cijeli sustav se maksimalno napuni hrannom. Simulator omogućava postavljanje minimalnog broja jedinki u populaciji, kao i maksimalnog broja jedinki u populaciji. Ukoliko se broj jedinki u populaciji smanji ispod minimalnog broja, generira se nova nasumična jedinka, a ukoliko broj jedinki u populaciji dosegne maksimalan broj jedinki, tada simulator ubije polovicu jedinki i smanji dostupnu količinu hrane u sustavu.

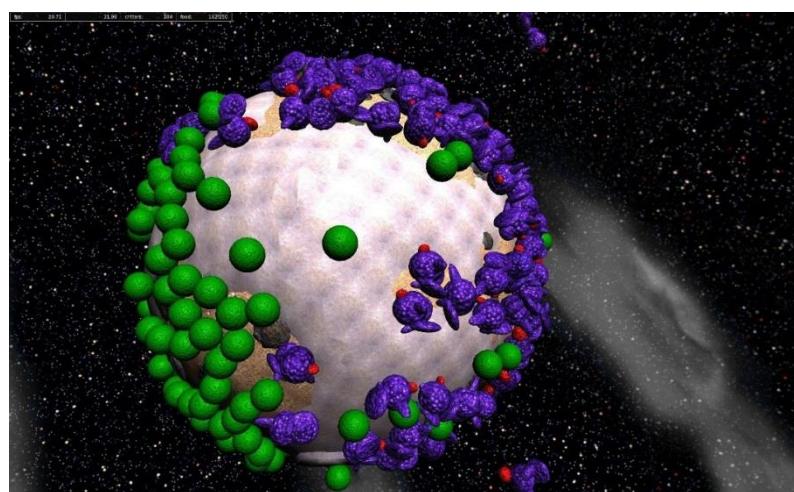
Jedinke umiru ukoliko dosegnu određenu starost ili ako ostanu bez energije, odnosno od gladi. Nasumično stvaranje jedinki kako bi se njihov broj zadržao u vijek iznad minimalnog omogućava simulatoru da će se nakon nekog vremena pojaviti jedinka koja se uspješno reproducira prije smrti.

2.1.1 Ekosustav

Critterding omogućava dvije vrste ekosustava. Prvi i zadani tip ekosustava je ravna ploha sa zidovima na rubovima kako jedinke i hrana ne bi mogle pasti van sustava (Slika 2.3) te drugi okrugli ekosustav sa svojom gravitacijom koji zadržava jedinke na svojoj površini (Slika 2.4). Od verzije beta 14 *Critterding* omogućava definiranje mapa pomoću wavefront obj datoteka kako bi se omogućilo još slobodnije oblikovanje svijeta jedinki.



Slika 2.3: Ravnji Critterding ekosustav



Slika 2.4: Okrugli Critterding ekosustav [6] [6] [6]

Tijekom simulacije *Critterding* omogućava kretanje kamere uz pomoć strelica i rotaciju kamere oko svojih osi korištenjem numeričke tipkovnice. Uz kretanje ekosustavom postoji i panel u kojem se može podešavati izgled ekosustava, odnosno osvjetljenje i njegove boje koje ne utječu na samu simulaciju, već na njenu vizualnu prezentaciju korisniku.

2.1.2 Jedinke

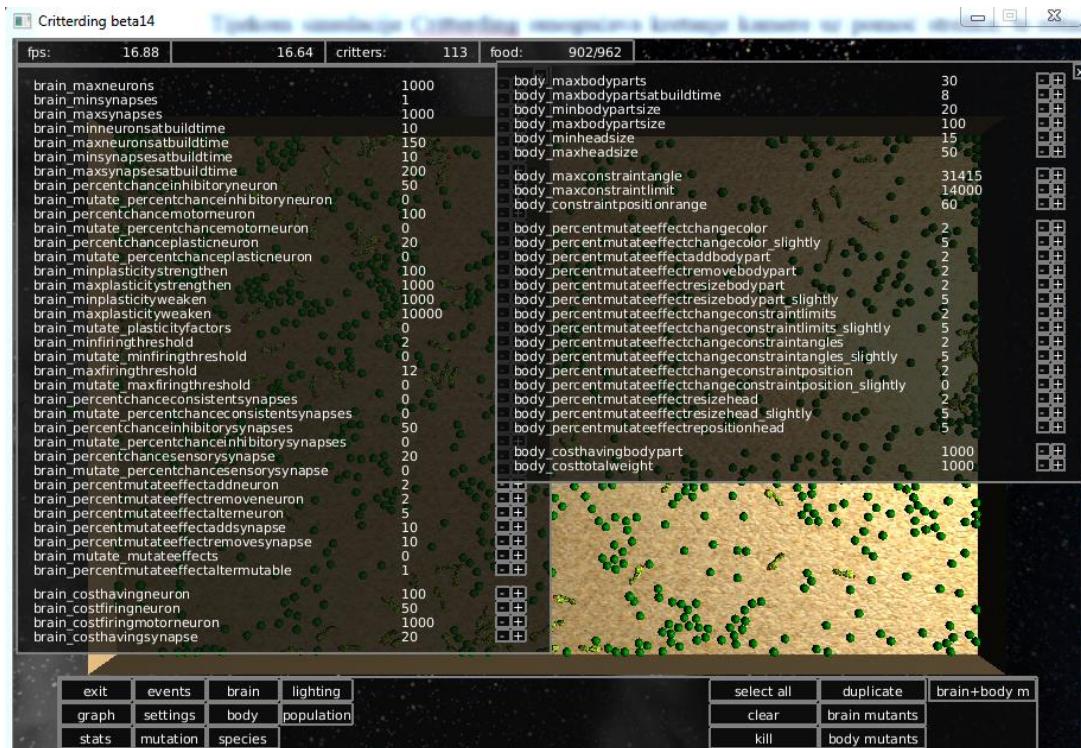
Akcije jedinki u ekosustavu simulatora *Critterding* određene su njihovim neuronskim mrežama. Akcije neuronske mreže vezane za kretanje su savijanje zgloba u jednu i njegovo savijanje u suprotnu stranu. Osim kretanja ona također određuje kada će se jedinka hraniti i razmnožavati.

Svaka jedinka koristi svoja "osjetila" za dobivanje informacija o svijetu i donošenje odluka neuronskom mrežom. Osjetilima dobivaju informacije da li glavom diraju hrana ili drugu jedinku i vide svijet koji ih okružuje. Svaka jedinka zna svoju dob, trenutno energetsko stanje, položaj tijela i mogućnost razmnožavanja.

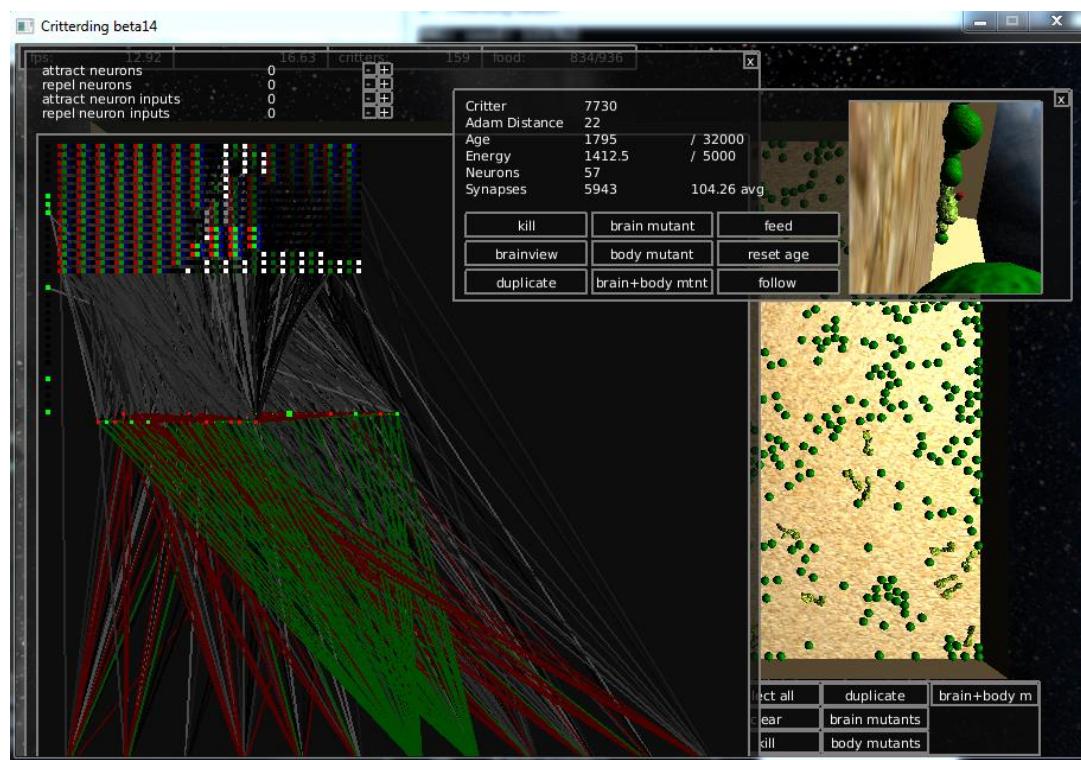
Simulator omogućava korisniku definiranje ograničenja za jedinku koja utječu na njene motoričke sposobnosti, razmnožavanje i sposobnost mutacije koja određuje mogućnost stvaranja novih vrsta. Jedinkama je moguće postaviti ograničenja maksimalne dobi, energije, početne energije i brzine kojom stare, odnosno troše energiju. Osim osnovnih parametara simulator također omogućava definiranje ograničenja za "mozak" jedinke odnosno njenu neuronsku mrežu te za pokretljivost i izgled tijela. Primjerice, jedinku možemo limitirati na konkretni broj neurona i veza između neurona ili maksimalnu težinu tijela i maksimalan broj udova od kojih je načinjeno tijelo jedinke (Slika 2.5).

Odabirom jedinke otvara se novi prozor s podacima o jedinki, prikazom njenog pogleda na ekosustav i mogućnostima interakcije s pojedinom jedinkom te praćenje kretanja jedinke u trenutnom ekosustavu (Slika 2.7). Prozor korisniku dopušta stvaranje duplikata jedinke, duplikata jedinke s mutiranim tijelom, mozgom ili mutiranim mozgom i tijelom, čime direktno utječe na ekosustav stvaranjem novih vrsta. Postojećim jedinkama simulator omogućava resetiranje dobi ili "hranjenje", tj. povećavanje iznosa trenutne energije i tako spašava jedinku

od izumiranja. Uz spašavanje jedinke od izumiranja, korisnik također može pogledati izgled neuronske mreže jedinke, odnosno "mozga" i utjecati na njezino ponašanje mijenjanjem njene mreže.



Slika 2.5: Parametri za tijelo i mozak jedinke

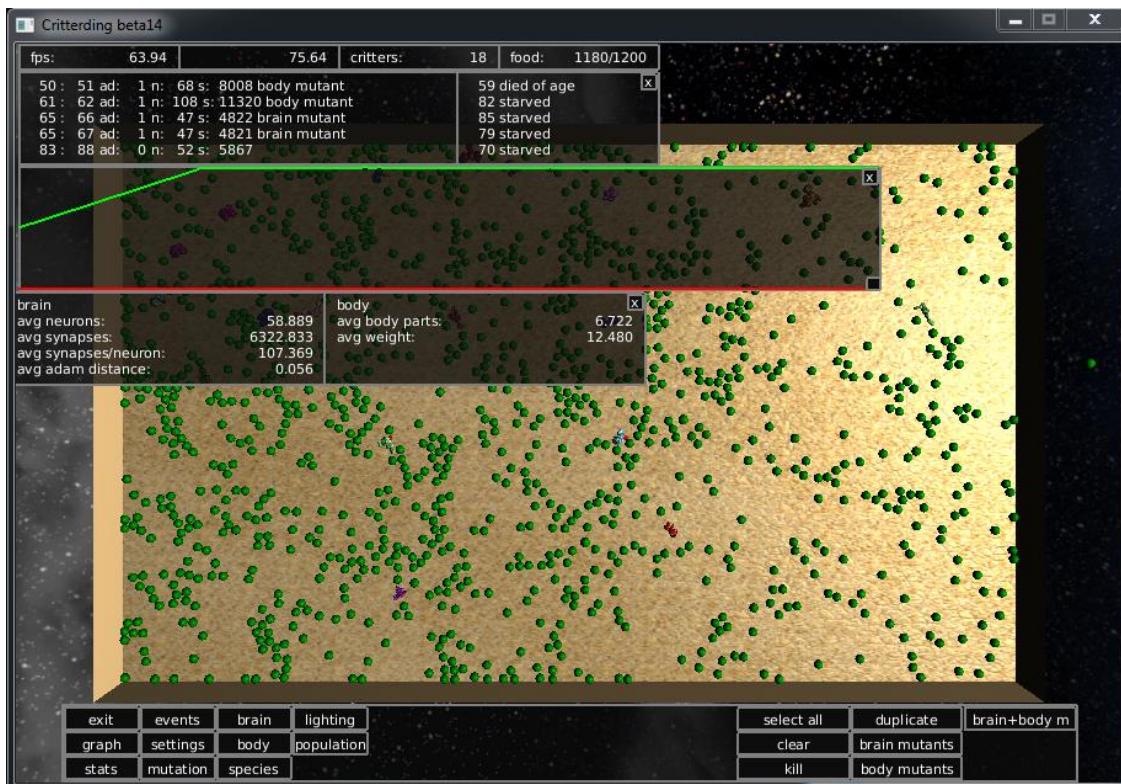


Slika 2.6: Prozor interakcije s jedinkom

2.1.3 Statistika

Tijekom simulacije također se vodi statistika o mozgu jedinki u obliku prosječnog broja neurona, sinapsi i prosječnog broja sinapsi po neuronu. Simulator također vodi evidenciju od koliko dijelova tijela je sačinjena prosječna jedinka, njenoj prosječnoj težini i računa "adam distance", koji određuje udaljenost prosječne populacije od početne populacije.

Simulator, uz statistiku, korisniku omogućava lakše praćenje zadnjih događaja u ekosustavu kroz prozor u kojem se ispisuje vrijeme i opis događaja te se tako lakše prati izumiranje i stvaranje novih vrsta (Slika 2.8).



Slika 2.7: Critterding statistika

2.2 Simulator 3D Virtual Creature Evolution (3DVCE)

3DVCE je simulator umjetnog života kojeg je 2008. godine razvio Lee Graham u jeziku C++ za operacijski sustav Windows, no uz program *Wine* moguće ga je pokrenuti i na Linuxima i OSX-u. Od 2012. godine Lee Graham je prestao razvijati simulator i ugasio službenu stranicu, no korisnici su simulator održali na životu i moguće je preuzeti njegovu arhiviranu verziju.

Simulacija umjetnog života i kriterij preživljavanja u 3DVCE-u određuje funkcija dobrote, koja svakoj jedinki daje određenu vrijednost ovisno koliko je bila uspješna u obavljanju zadanih ciljeva, za razliku od simulatora *Critterding* u kojem je jedinka sama odgovorna za preživljavanje svoje vrste ako se nauči izboriti za hranu i reproducirati. Osim razlike u stvaranju novih jedinki simulator *Critterding* i 3DVCE se razlikuju i u implementaciji umjetne inteligencije. *3D Virtual Creature Evolution* simulator koristi se genetskim programiranjem dok *Critterding* za "mozak" jedinke koristi neuronsku mrežu.

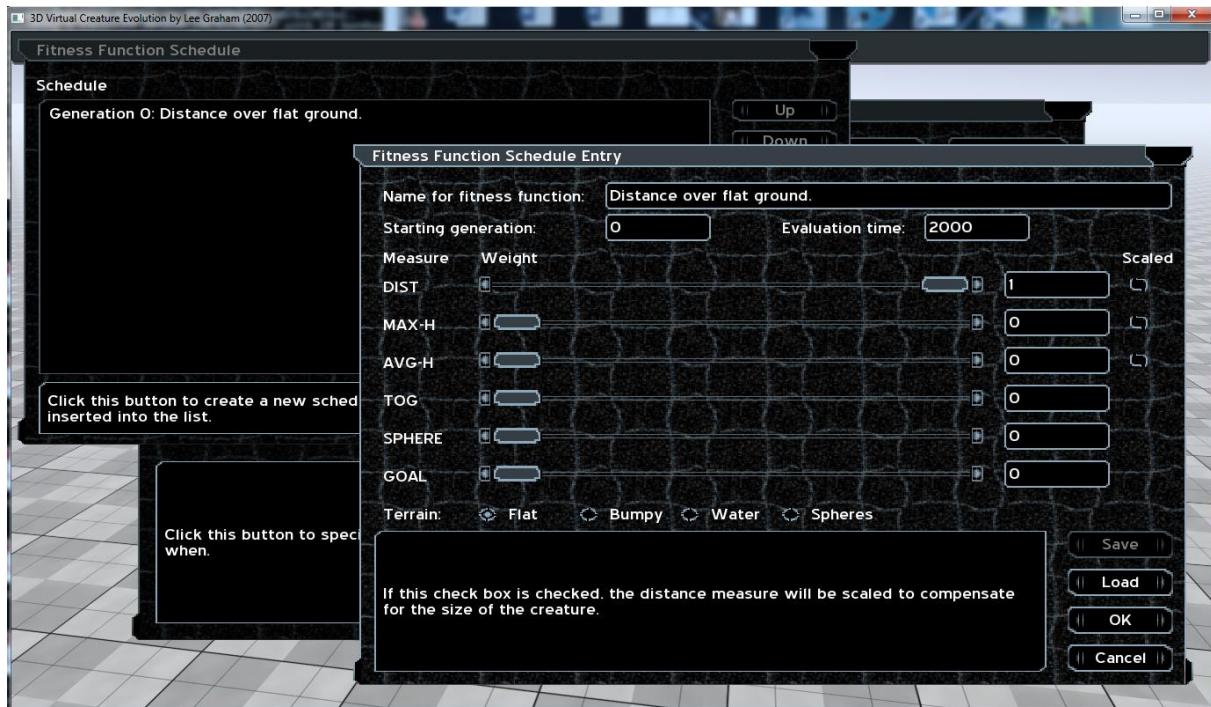
Projekt je nastao s ciljem vizualizacije strategija jedinki pri svladavanju zadanih problema te promatranja smjera evolucije tijela i ponašanja koji im omogućava primjenu takve strategije u zadanom ekosustavu. Primjerice, ako se jedinku vrednuje po maksimalno pređenoj distanci i prosječnoj visini tijela prilikom kretanja jedinke se često razviju u crvoliku strukturu koja se nauči prelaziti udaljenosti skupljanjem i propinjanjem glave u zrak poput opruge, čime se zadovoljava kriterij prosječne visine uz maksimiziranje pređene udaljenosti.

2.2.1 Ekosustav

Simulator dopušta nekoliko opcija korisniku pri oblikovanju ekosustava u kojem se jedinka simulira, čime se postavljaju osnovna ograničenja ili ciljevi za jedinku. Postoje četiri predviđene opcije, od kojih su tri uspješno implementirane: simulacija na ravnom terenu, brdovitom terenu čija visina raste kako se jedinka kreće od početne pozicije i ravna ploha uz padanje sfera na organizam koje doprinose ukupnoj dobroti jedinke, ako ih jedinka uspije zadržati na sebi.

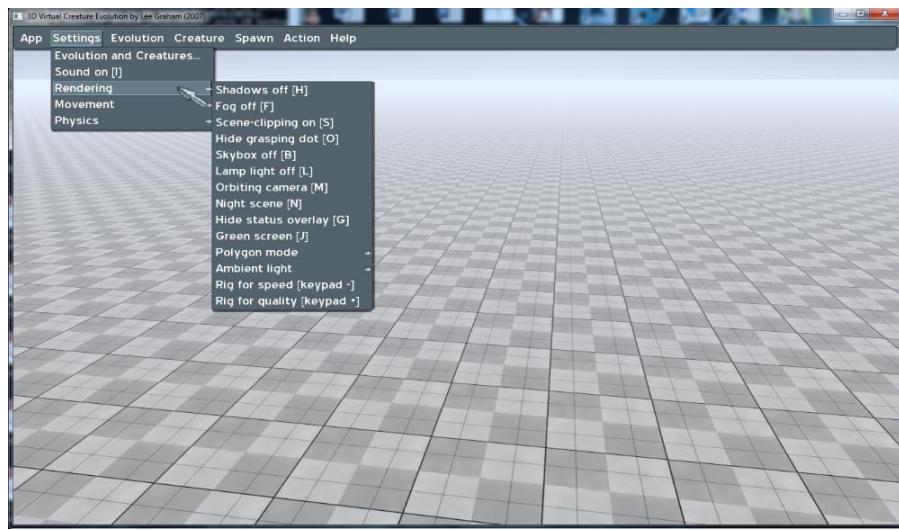
Prije pokretanja simulacije moguće je postaviti funkciju dobrote koja će određivati kriterij preživljavanja pojedine jedinke. Prepostavljena funkcija je pređena udaljenost, no ona može biti i funkcija udaljenosti, maksimalne ili

prosječne visine jedinke, vremena provedenog u kontaktu sa tlom ili kombinacija parametara (Slika 2.9). Osim definiranja funkcije dobrote moguće je i definirati raspored s više funkcija te kada će se koja koristiti tijekom simulacije i tako simulirati dinamične promjene u ekosustavu na koje se jedinke moraju privikavati kako bi preživjele.

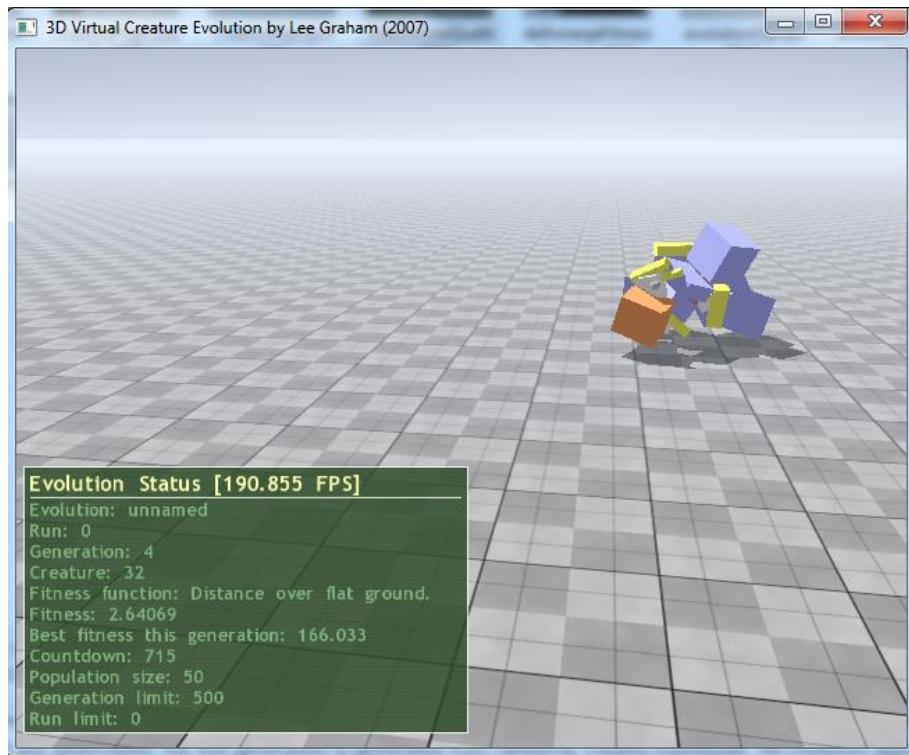


Slika 2.8:3DVCE fitness funkcija

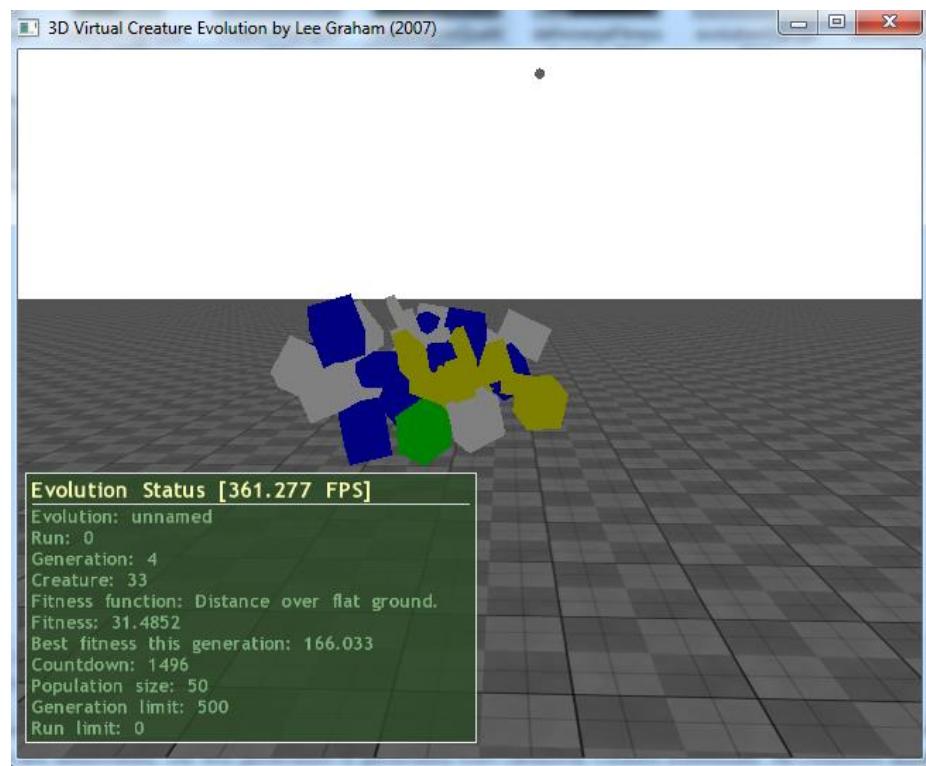
Nakon postavljenih parametara i odabira oblika ekosustava simulator kreće sa stvaranjem nasumičnih jedinki te im dodjeljuje zadano vrijeme tijekom kojeg se jedinke evaluiraju. Sustav ne limitira broj slika koje prikazuje korisniku u sekundi, odnosno nastoji izvršiti simulaciju što prije može. Moguće je ubrzati simulaciju tako da se isključe osvjetljenja, detaljne teksture i svjetlost u ekosustavu kako bi se što manje vremena trošilo na iscrtavanje ekosustava i jedinke koja se trenutno evaluira (Slika 2.10, Slika 2.10: Simulacija veće kvalitete Slika 2.12).



Slika 2.9: Opcije prikaza simulacije



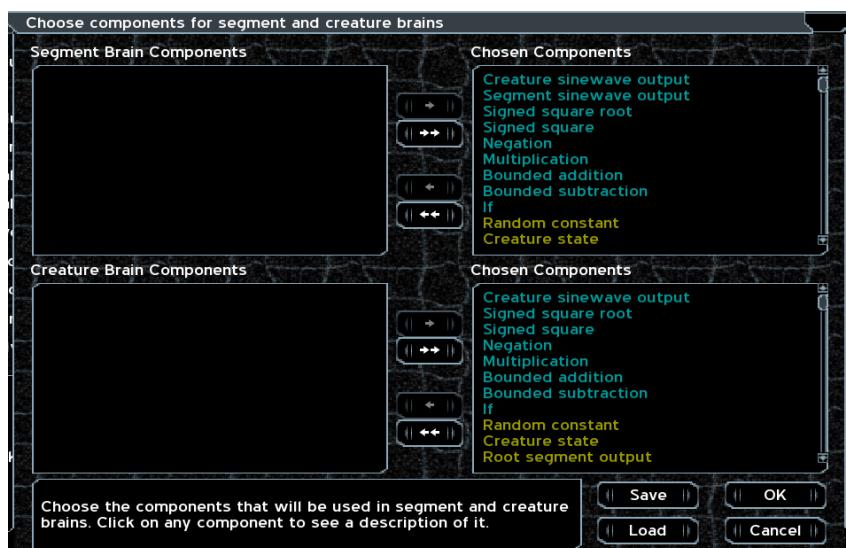
Slika 2.10: Simulacija veće kvalitete



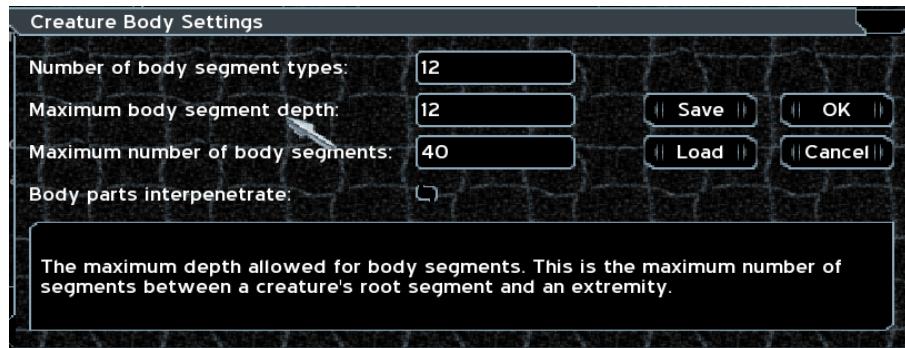
Slika 2.11: Simulacija manje kvalitete

2.2.2 Jedinke

Početna generacija jedinki kreira se nasumično uz ograničenja broja komponenata i vrste komponenata koje su sustavu dozvoljene pri stvaranju jedinki (Slika 2.12, Slika 2.13). Svaka jedinka početne populacije stvori se u ekosustavu i dodjeljuje joj se predefinirano vrijeme života, nakon čega se evaluira trenutno aktivnom funkcijom dobrote.



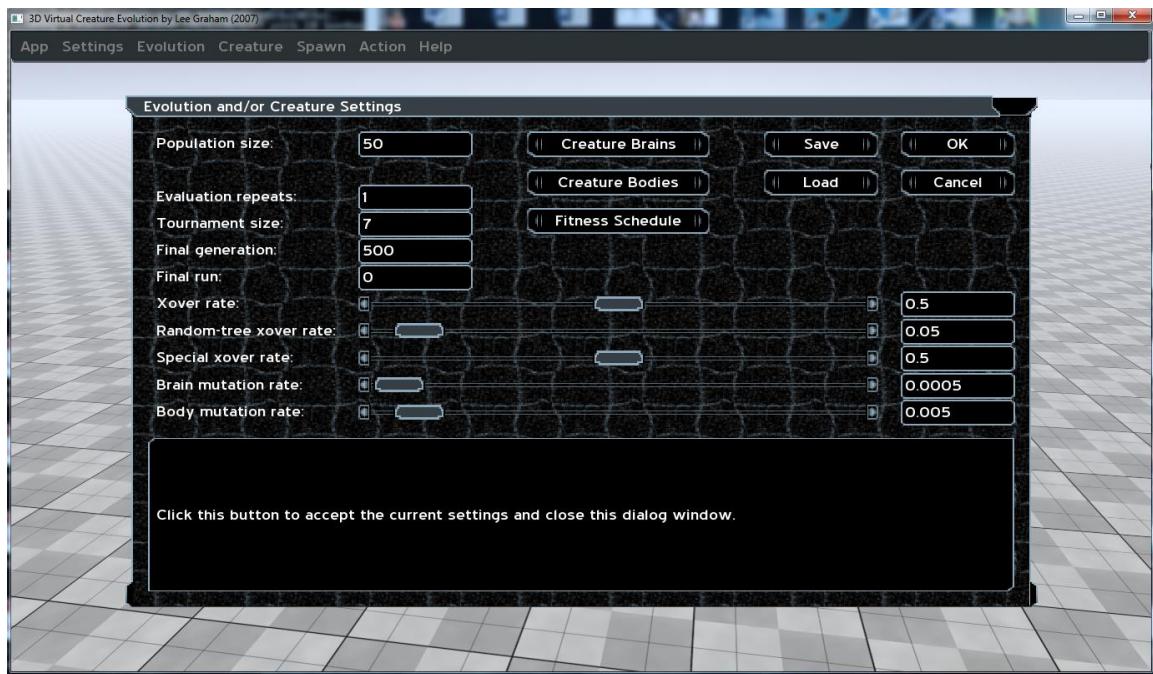
Slika 2.12: Postavke parametara mozga



Slika 2.13: Postavke parametara tijela

Za stvaranje sljedeće generacije jedinki koristi se ona prijašnja čija je evaluacija završena. Nova jedinka stvara se križanjem svojstava dvaju roditelja koja se biraju iz prošle generacije korištenjem turnirske selekcije, a broj natjecatelja u selekciji roditelja odabire korisnik u postavkama.

Nakon izabralih roditelja radi se križanje, ovisno o postavkama koje su odabrane. Primjerice, može se zadati da svojstva prvog roditelja više doprinose pri oblikovanju nove jedinke ili da se prvi roditelj križa sa nasumično stvorenom jedinkom (Slika 2.14). Kako bi se prerana konvergencija izbjegla, simulator



Slika 2.14: Parametri križanja i mutacije

omogućava postavljanje vjerojatnosti mutacije "mozga" odnosno tijela jedinke, čime se osigurava mogućnost stvaranja nove jedinke koja će pristupiti problemu na drugačiji način, odnosno ostvariti bolju dobrotu u simulaciji.

3. Praktičan rad

Simulator umjetnog života ima nekoliko zadataka. Osnovni zadaci sučelja su modeliranje organizama i njihovog ponašanja, kao i ekosustava u kojem žive i njegovih ograničenja. Osim toga simulator je zadužen za prikaz ekosustava i organizama koji u njemu obitavaju kako bi korisnik mogao pratiti izvođenje simulacije.

3.1 Tehnologije

3.1.1 Java 8



Slika 3.1: Java logo

Java je objektno orijentirani računalni programski jezik, dizajniran kako bi imao što manje implementacijskih ovisnosti, što znači da se prevedeni kod može pokrenuti na svim platformama koje podržavaju Javu, bez potrebe za ponovnim prevođenjem.

3.1.2 JavaFX



Slika 3.2: JavaFX logo

Za grafički sustav odabran je JavaFX, koji je od svoje verzije 2.2 uključen u standardno izdanje Jave 7, nadogradnje 6. JavaFX je programska platforma za stvaranje desktop aplikacija.

JavaFX je dizajniran za pružanje jednostavnog i sklopoški ubrzanog Javiniog korisničkog sučelja. Zahvaljujući transformacijama i CSS podrškom, JavaFX omogućava stvaranje jedinstvenog dizajna sučelja te efikasan i brz prikaz ekosustava.

3.2 Program za simulaciju

Izrađeni simulator ostvaruje svoje osnovne funkcionalnosti kroz nekoliko prozora. Početni prozor služi za definiciju osnovnih parametara, poput veličine ekosustava i vrsta jedinki u simulaciji. Nakon odabranih vrsta organizama i veličine ekosustava korisniku se otvara prostor za postavljanje parametara svake vrste te na kraju simulacijski prozor u kojem se umjetni život vizualizira.

3.2.1 Početni prozor

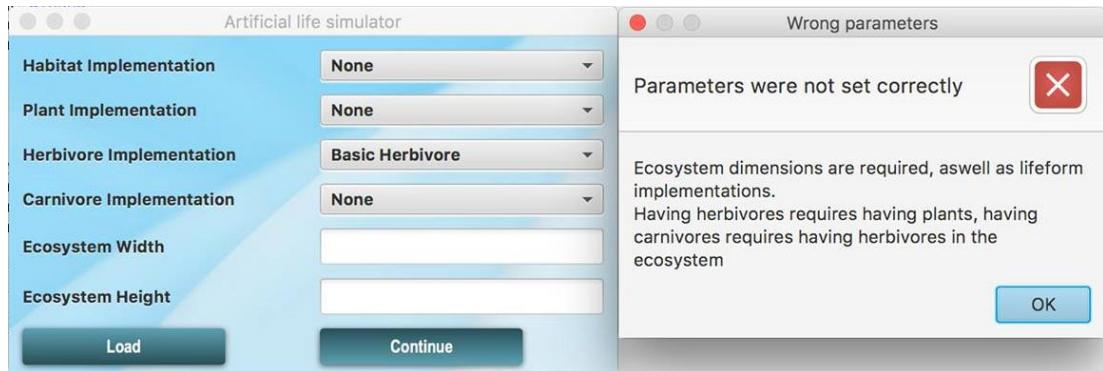
Početni prozor je prva stvar koju korisnik vidi prilikom pokretanja programa. Njegova svrha je definicija osnovnih parametara, poput vrsta koje se žele simulirati, kao i veličine ekosustava u kojem se jedinke simuliraju, čija je maksimalna vrijednost 2000x2000 (Slika 3.3).



Slika 3.3: Početni prozor

Početni prozor također vrši provjeru korisnikovih odabira. Primjerice, ne može se dogoditi da korisnik odabere simulaciju u kojoj su jedina vrsta biljojedi jer takva simulacija nema smisla (Slika 3.4). Legalne kombinacije za simuliranje su biljke,

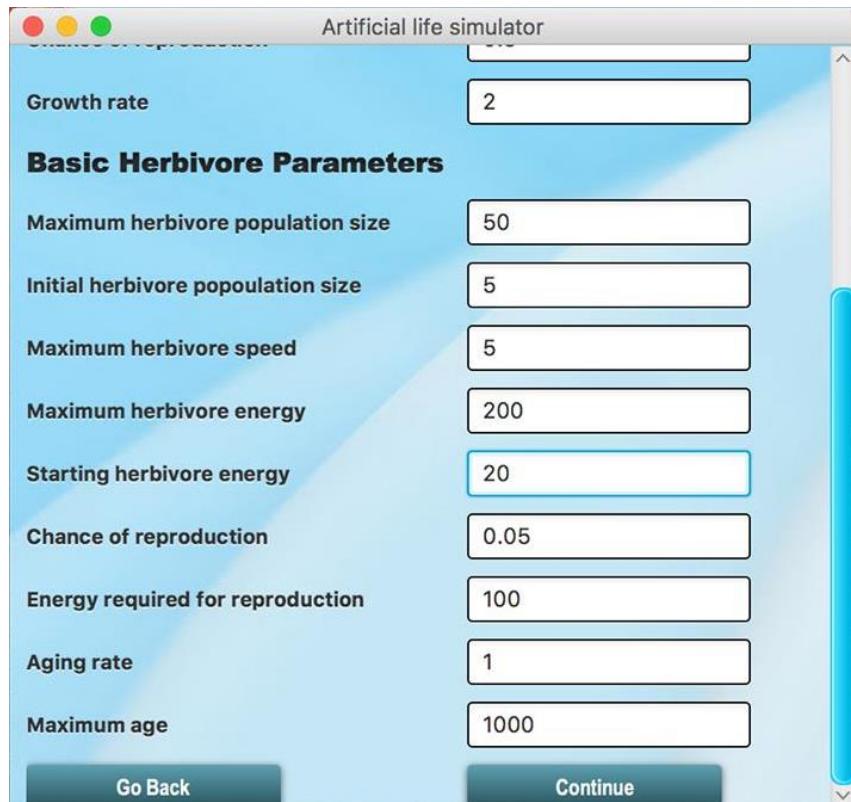
biljke i biljojedi i biljke, biljojedi i mesojedi, odnosno simulator zahtjeva da postoji hranidbeni lanac među jedinkama kako bi simulacija imala smisla.



Slika 3.4: Pogrešni parametri

3.2.2 Prozor za definiciju parametara

Nakon odabranih vrsta i veličine ekosustava, u sljedećem prozoru korisnik postavlja parametre svakoj od odabranih vrsta koje će se simulirati u ekosustavu (Slika 3.5).



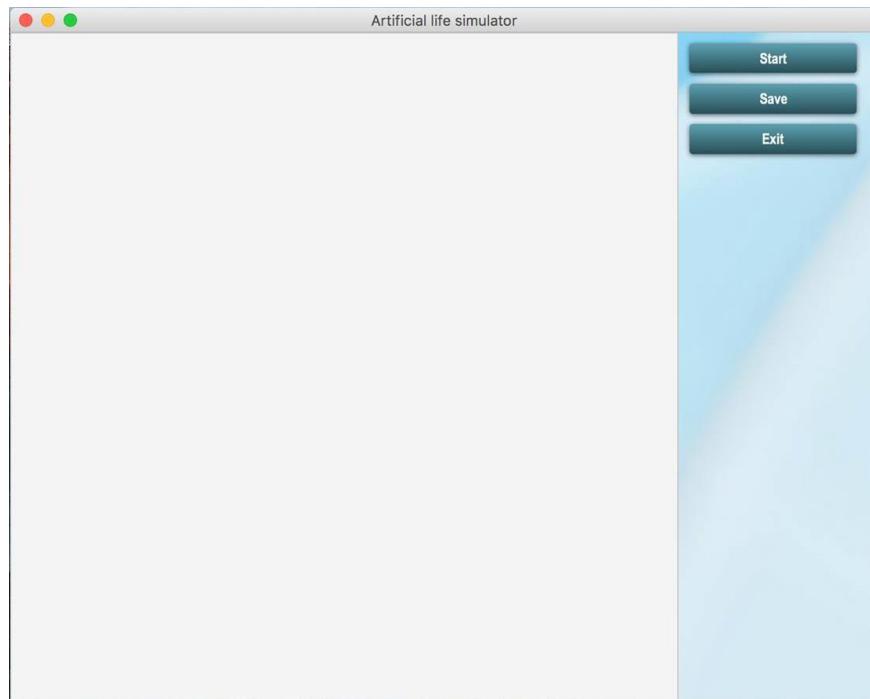
Slika 3.5: Prozor za definiciju parametara vrste

Polja za postavljanje parametara jedinkama se dohvaćaju tijekom izvođenja programa. Ukoliko se implementira nova vrsta i njena populacija, sve što je potrebno da se ona integrira u postojeći simulator je napraviti novu klasu koja nasljeđuje *AbstractParameterScene*, implementirati njene metode i dodati ime vrste u *ImplementedLifeforms*. Dinamično dohvaćanje polja za definiranje parametara znači da svaka vrsta može imati drugačiju implementaciju i potrebe za različitim parametrima te se ne ograničava mogućnost implementacije novih vrsta.

Kako bi se smanjila potreba za provjerom korisničkog unosa, implementirana su tekstualna polja koja ograničavaju korisnika da njegov unos bude cijeli broj ili realni broj u intervalu [0, 1], ako polje definira vjerojatnost nekog događaja.

3.2.3 Simulacijski prozor

Zadnji prozor je simulacijski prozor do kojeg korisnik dolazi nakon što uspješno definira sve parametre simulacije ili ako je korisnik učitao datoteku u kojoj je spremljena postojeća simulacija koju želi nastaviti simulirati (Slika 3.6).

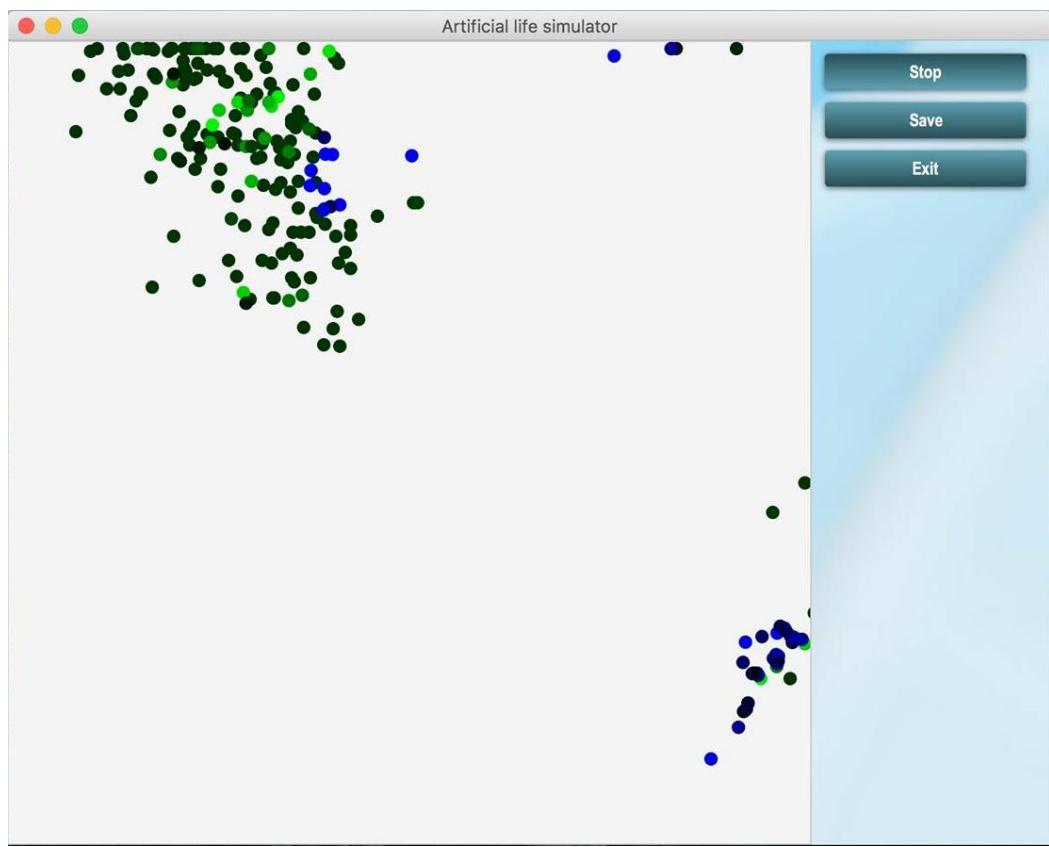


Slika 3.6: Simulacijski prozor

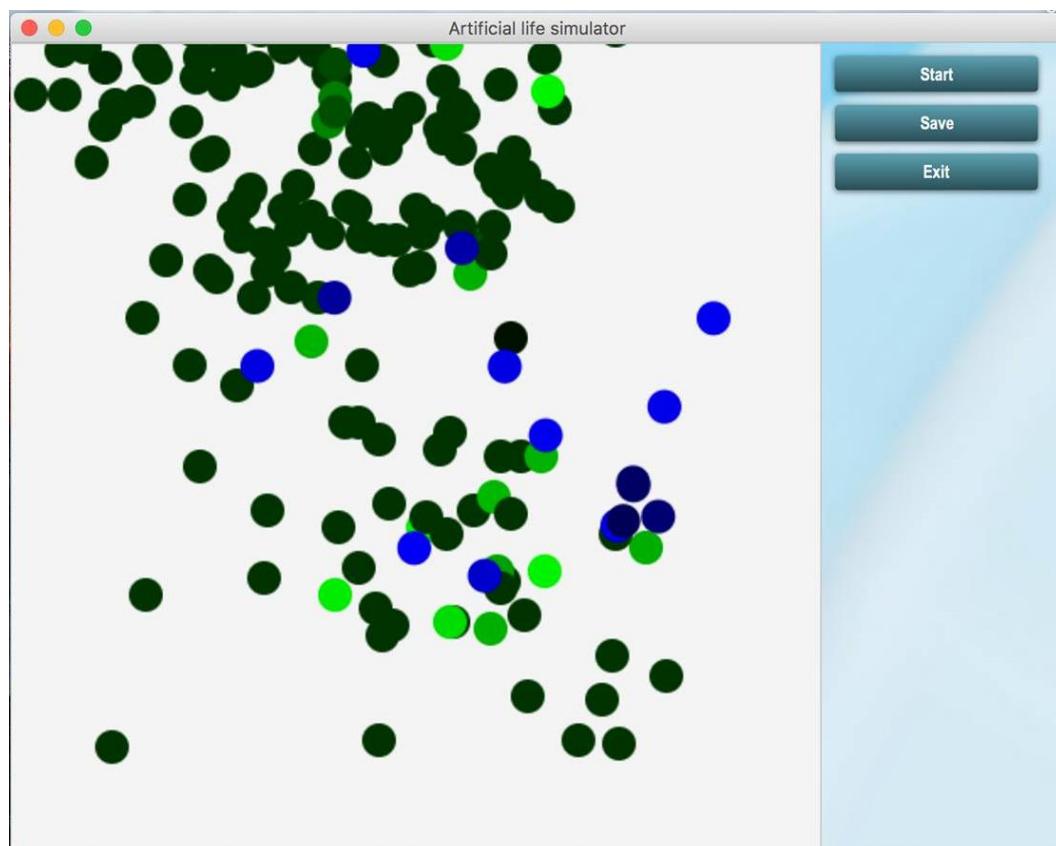
Iscrtavanje je podijeljeno u tri sloja. Vršni sloj je zaslužan za iscrtavanje biljojeda i mesojeda, odnosno jedinki koje se mogu kretati po ekosustavu. Srednji sloj je zadužen za crtanje biljaka koje su statični elementi u ekosustavu. Zadnji sloj iscrtava sam ekosustav i njegove karakteristike. Ovakva podjela omogućava vizualizaciju ekosustava u kojem su pokretni elementi uvijek vidljivi i neće biti zaklonjeni biljkama ako se nalaze na istom mjestu.

Prikaz simulacije nije ograničen na određeni broj sličica po sekundi te se simulacija nastoji izvršavati što je brže moguće kako se učenje složenijih jedinki ne bi limitiralo prikazom. Iscrtavanje jedinki se odvija u svojoj dretvi, dok se izračunavanje iteracija jedinki odvija u drugoj. Dretve komuniciraju preko blokirajućeg reda limitiranog na 5 elemenata, kako ne bi došlo do zagušenja radne memorije ako je simulacijska dretva brža od dretve zadužene za iscrtavanje. Način crtanja vrsta definiran je u samoj implementaciji vrste i neovisan je o implementaciji grafičkog sučelja.

Korisniku je omogućeno upravljanje tijekom simulacijom u obliku pauziranja i ponovnog pokretanja simulacije te spremanje trenutnog ekosustava s njegovim jedinkama ako kasnije želi nastaviti simulaciju. Prilikom simulacije korisnik može zumirati pritiskom miša na određeni prostor u ekosustavu, nakon čega se ekosustav prikazuje s mjestom pritiska u centru prostora za iscrtavanje (Slika 3.7 i Slika 3.8).



Slika 3.7: Prije zumiranja



Slika 3.8: Poslje zumiranja

3.2.4 Demo simulacija

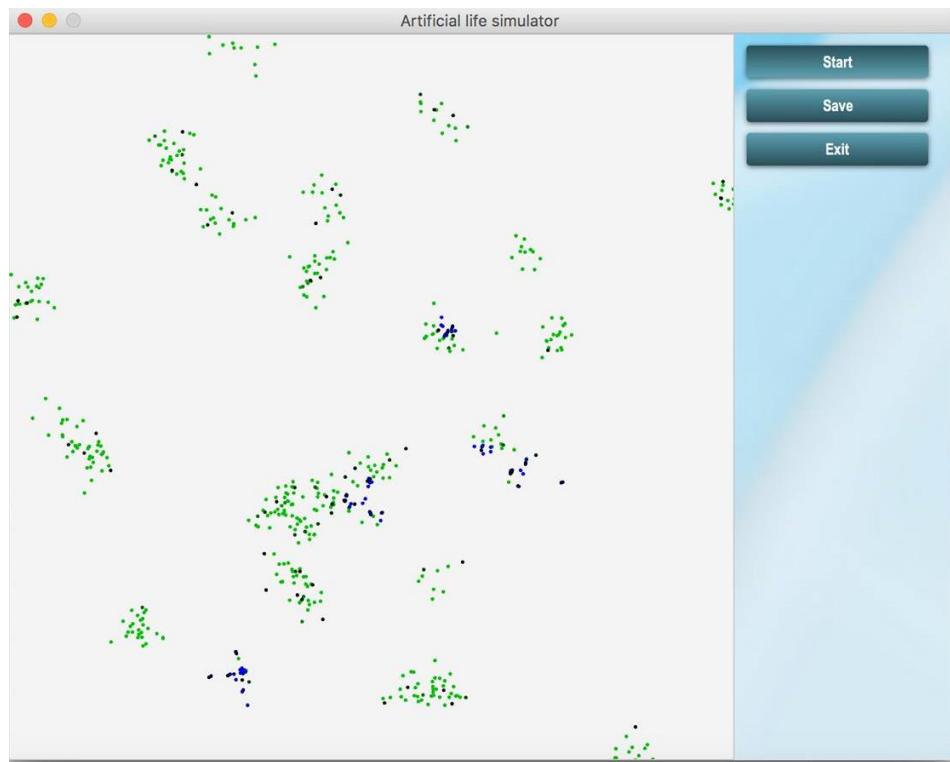
Demo simulacija napravljena je korištenjem implementacije osnovnih biljojeda i biljaka (*Basic Plants, Basic Herbivores*).

Osnovna implementacija biljaka napravljena je tako da se razmnožava u radijusu od 50 jedinica oko sebe, ako joj je trenutna energetska vrijednost iznad zadane razine, pri čemu joj se energetska razina ne smanjuje kako biljke ne bi bile istrebljene prebrzo.

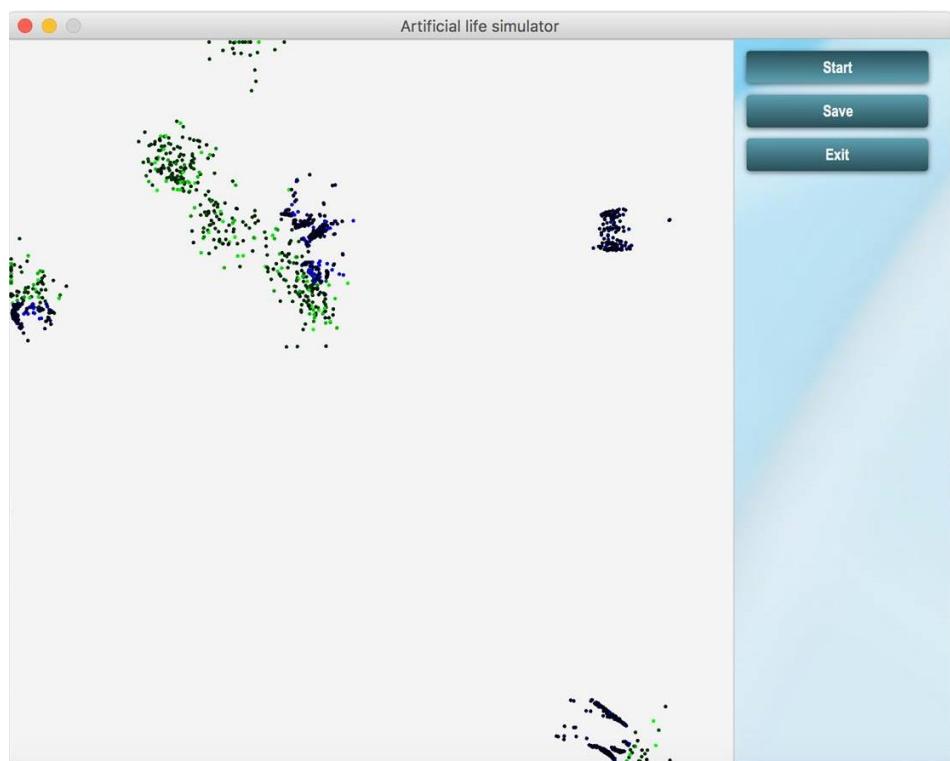
Osnovna implementacija biljojeda dobiva informaciju o najbližoj biljki i nastoji svoju razinu energije zadržati na što višoj razini. Ukoliko je vrijednost energije veća od zadane energije potrebne za razmnožavanje, tada se događa dioba i stvara se novi biljojed pri čemu roditelj gubi energiju.

Biljke su prikazane zelenom bojom i njihova boja indicira količinu energije. Tamnija boja znači da je biljka "zrelja", odnosno sadrži više energije. Biljojedi su prikazani plavom bojom, uz pravilo da tamnija boja određuje starost biljojeda.

Početni izgled ekosustava sadrži manje skupine biljaka koje biljojedi kasnije istrijebe, zbog čega kasnije migriraju u prostore koji sadrže biljke za ishranu (Slika 3.9, Slika 3.10). Ekosustav je veličine 2000x2000, a za parametre jedinke ostavljeni su svi prepostavljeni parametri osim maksimalne veličine populacije biljojeda koja je postavljena na 5000. Prepostavljeni parametri biljkama omogućuju brže razmnožavanje i rast, nego što su sposobnosti biljojeda da ih istrijebe, zbog čega biljojedi stalno migriraju u područja gdje rastu nove biljke i simulacija traje vječno (Slika 3.11).



Slika 3.9: Početni izgled ekosustava



Slika 3.10: Nakon istrebljenih nekoliko skupova biljaka



Slika 3.11: Migracije u prostoru s biljkama

5. Budući rad

Postojeći sustav sadrži dvije jednostavne implementacije jedinki koje nemaju implementiranu umjetnu inteligenciju već im je ponašanje jako primitivno. S obzirom na to da sustav omogućava jednostavnu implementaciju novih vrsta, potrebno je implementirati nove vrste koje bi imale umjetnu inteligenciju ostvarenu neuronskom mrežom ili nekim od evolucijskih algoritama, primjerice genetskim programiranjem.

Kako bi se tijek evolucije lakše pratio, potrebno je također implementirati i prozor koji će omogućavati vizualizaciju stvaranja novih populacija koje nastaju križanjem i mutacijom postojećih jedinki u ekosustavu.

Osim poboljšanja ponašanja jedinki, sustavu je potrebno implementirati i oblik staništa koji je dinamički promjenjiv, a svojim stanjem bi imao utjecaj na jedinke. Stanište bi moglo modelirati promjene u vremenu, npr. simulirati kišno ili sunčano vrijeme, koje bi utjecalo na rast biljaka ili vidljivost u sustavu te bi moglo imati dinamične promjene reljefa koje bi uzrokovale promjene u kretanju jedinki poput stvaranja planina ili rijeka u ekosustavu.

Sustav je predviđen da se jedinke ne evaluiraju i da se nad njima ne radi selekcija kako bi se dobivale nove bolje jedinke. To je prepusteno jedinkama, što znači da proces evolucije do prve jedinke koja se naučila preživjeti u ekosustavu može trajati duže vrijeme. Kako bi to inicijalno vrijeme smanjili, potrebno je nadograditi sustav da korisniku omogući ubrzavanje evolucije uz pomoć selekcije i evaluacije, čiji kriterij završetka bi odabrao korisnik te bi time dobio "istreniranu" populaciju do odabranog kriterija, primjerice naučene ishrane, koju dalje može promatrati u ekosustavu.

6. Zaključak

Život i sposobnost živih bića da se prilagode na uvjete svoje okoline oduvijek su bile teme koje su fascinirale ljudi te su ih nastojali reproducirati. Uz novije algoritme i veću procesorsku moć računala moguće je simulirati sve složenije oblike umjetne inteligencije, koje upravljaju jedinkama u simulacijama umjetnog života.

Simulator je napravljen u jeziku Java, a grafičko sučelje u JavaFX-u koji je uključen u standardno izdanje Jave od Jave 7 verzije 6.

Grafičko sučelje simulatora sastoji se od početnog prozora, prostora za definiciju parametara i simulacijskog prozora. U početnom prozoru korisnik bira implementacije organizama i definira parametre ekosustava. Odabranim organizmima korisnik zatim postavlja željene parametre, nakon čega se otvara simulacijski prozor u kojem korisnik može pauzirati ili pokrenuti, snimiti simulaciju ili zumerati simulaciju ukoliko je definirao veći ekosustav te na kraju izaći iz simulacije.

Ostvareni simulator je predviđen za simulacije biljaka, biljojeda i mesojeda u svom ekosustavu koji je ograničene veličine. Simulator je lako nadogradiv, a dodavanje novih vrsta jedinki se svodi na implementaciju jedinke, njene populacije i prozora za definiciju parametara koji se koriste za inicijalizaciju. Lakom implementacijom novih vrsta omogućava se simulacija npr. biljojeda kojima upravlja neuronska mreža te mesojeda čije akcije su rezultat genetskog programiranja, čime se korisniku omogućuje usporedba performansi različito upravljenih jedinki.

Proces evolucije i nastanka novih vrsta je složen i nepredvidiv proces te se u istim uvjetima mogu razviti različiti oblici života koji jednako uspješno obitavaju ekosustavom.

7. Literatura

- [1] Christoph Adami. Introduction to Artificial Life. Springer-Verlag, Berlin, 1998.
- [2] Emmeche, Claus, Artificial Life, dostupno na adresi:
http://www.encyclopedia.com/topic/Artificial_life.aspx, pristupano: 17.5.2016
- [3] Winckelmans, Bob, Critterding, dostupno na adresi:
<http://critterding.sourceforge.net/>, pristupano 22.5.2016
- [4] Artificial Life, dostupno na adresi: https://en.wikipedia.org/wiki/Artificial_life,
pristupano 22.5.2016
- [5] Avida, dostupno na adresi: <https://en.wikipedia.org/wiki/Avida>, pristupano
22.5.2016
- [6] Winckelmans, Bob, Critterding, dostupno na adresi:
<https://sourceforge.net/projects/critterding/>, pristupano 22.5.2016
- [7] 3DVCE, dostupno na adresi: http://3dvce.wikia.com/wiki/Main_Page

Programsko sučelje za simulaciju umjetnog života

Sažetak

U ovom radu navedeni su neki od algoritama korištenih za simulaciju umjetne inteligencije u umjetnom životu. Opisana su dva besplatno dostupna simulatora umjetnog života koji su ostvarili umjetnu inteligenciju jedinki korištenjem različitih algoritama.

U praktičnom dijelu ostvareno je sučelje koje simulira tri vrste jedinki u 2D okruženju, biljke, biljojede i mesojede i omogućava nadogradnju dodavanjem novih implementacija. Na kraju je prikazana simulacija primitivnih jedinki i dana ideja za implementaciju intelligentnijih vrsta.

Ključne riječi

umjetni život, simulator, simulacija, biljke, biljojedi, mesojedi, umjetne neuronske mreže

Programming interface for the simulation of artificial life

Sažetak

This paper describes algorithms used for simulation of artificial intelligence in artificial life. Two simulators that are freely available were described, as well as their way of achieving artificial intelligence of simulated organisms.

In the practical part an interface for simulation of organisms in a 2D environment was created. The supported organisms are plants, herbivores and carnivores, with an included option of adding new implementations of organisms. Paper also describes a simple simulation with implemented basic plants and herbivores, with an explained idea of ways to implement more intelligent organisms.

Ključne riječi

artificial life, simulator, simulation, plants, herbivores, carnivores, artificial neural networks