

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1795

Simulator umjetnog života u 3D prostoru

Mak Krnic

Mentor: *prof. dr. sc. Marin Golub*

Zagreb, veljača 2019.

Zagreb, 11. listopada 2018.

Predmet: **Diplomski rad**

DIPLOMSKI ZADATAK br. 1795

Pristupnik: **Mak Krnic (0036453267)**

Studij: Računarstvo

Profil: Računalno inženjerstvo

Zadatak: **Simulator umjetnog života u 3D prostoru**

Opis zadatka:

Simulacijom umjetnog života (engl. Artificial Life, Alife) moguće je proučavati sustave koji se odnose na život, njegove procese i evoluciju koristeći računalne modele. Ispitati i opisati nekoliko slobodno dostupnih programskih paketa koji omogućuju simulaciju i vizualizaciju umjetnog života. Na temelju tako stečenih iskustava osmisliti vlastitu okolinu i definirati svojstva jedinki. Koristeći slobodno dostupna razvojna okruženja i slobodno dostupne biblioteke programa programski ostvariti računalni model umjetnog života u kojem se simuliraju hranjenje, reprodukcija, umiranje i kretanje jedinki u 3D prostoru. Ispitati rad sustava prema nekoliko odabranih scenarija te različitih okolina i početnih vrijednosti parametara.

Zadatak uručen pristupniku: 12. listopada 2018.

Rok za predaju rada: 8. veljače 2019.

Mentor:



Prof. dr. sc. Marin Golub

Djelovođa:



Prof. dr. sc. Danko Basch

Predsjednik odbora za
diplomski rad profila:



Prof. dr. sc. Mario Kovač

SADRŽAJ

| | |
|--|----------|
| 1. Uvod | 1 |
| 2. Postojeći simulatori | 2 |
| 2.1. Conwayeva igra života | 2 |
| 2.2. Simulator <i>Framsticks</i> | 3 |
| 2.3. Računalna igra <i>Species ALRE</i> | 5 |
| 2.3.1. Simulacija | 5 |
| 3. Praktični rad | 9 |
| 3.1. Korištene tehnologije | 9 |
| 3.1.1. Simulator Minetest | 9 |
| 3.1.2. Biblioteka programa OpenGL | 10 |
| 3.1.3. Skriptni jezik Lua | 10 |
| 3.2. Programsko ostvarenje | 10 |
| 3.2.1. Definicija modela virtualnog života i okruženja | 10 |
| 3.2.2. Virtualni svijet | 10 |
| 3.2.3. Konfiguracija | 11 |
| 3.2.4. Biljke | 13 |
| 3.2.5. Biljojedi | 15 |
| 3.2.6. Mesojedi | 17 |
| 3.2.7. Pseudokod simulatora | 17 |
| 3.2.8. Korisničko sučelje | 19 |
| 3.3. Rezultati ispitivanja | 19 |
| 3.3.1. Biljke prevladavaju | 20 |
| 3.3.2. Svi izumiru | 22 |
| 3.3.3. Alternativne postavke | 25 |
| 3.4. Budući rad | 27 |

| | |
|---------------------|-----------|
| 4. Zaključak | 28 |
| Literatura | 29 |

1. Uvod

Čovjek je oduvijek pokušavao predviđati buduće događaje kako bi mogao što bolje iskoristiti situaciju za vlastitu dobrobit. Do prije stotinjak godina to se odvijalo isključivo ručno, dugotrajnim i mukotrpnim izračunavanjem mnogobrojnih iteracija raznih formula. No, s pojavom digitalnih računala, pokušavaju se – s većom ili manjom preciznošću i točnošću – simulirati primijećeni prirodni procesi. Od najjednostavnijih fizikalnih zakonitosti kao što je *1. Newtonov zakon*, preko izračuna i simulacija kretanja planeta do veoma složenih i teško formalno opisivih kao što je sam život.

Nastojeći simulirati život ili neke njegove dijelove, razvilo se cijelo jedno područje algoritama – *prirodom inspirirani algoritmi*. Razni genetski, evolucijski i algoritmi rojeva, koji su u svojoj biti *optimizacijski* te ne garantiraju pronalazak pravog ili točnog rješenja za zadani problem, već samo najboljeg koje nađu.

Umjetni život (*engl. Artificial life; ALife*) pojam je koji označava metode i tehnike oponašanja živih organizama pomoću računala. Istraživanje umjetnog života je započelo s ciljem stvaranja isključivo bioloških simulacija, no danas se koristi i za razmatranje svijesti i predviđanje ponašanja.

Za simulaciju se modeliraju jedinice koje su uglavnom vrlo jednostavne, no simulira ih se velik broj te mogu, ali i ne moraju međusobno komunicirati i razmjenjivati informacije (npr. optimizacija rojem čestica).

Od simulatora umjetnog života, vjerojatno je najpoznatiji *Conwayeva igra života* – takozvana "igra za 0 igrača" – koja je formalno potpuno deterministička te evolucija ovisi isključivo o početnom stanju. Više o tome u odjeljku 2.1

Osim na *Conwayjevu igru života*, u radu je dan osvrt još i na simulator *Framsticks* (2.2) te na računalnu igru-simulator *Species ALRE* (2.3).

U 3. poglavlju i njegovim potpoglavljima opisane su korištene tehnologije (*OpenGL*, *Minetest*, *Lua*), izvedbeni detalji samog simulatora, dana je definicija umjetnog života i virtualnog okruženja u okviru ovog rada te su provedena ispitivanja i analizirani rezultati.

2. Postojeći simulatori

2.1. Conwayeva igra života

Počecima umjetnog života smatra se *Conwayjeva igra života*, u kojoj je život predstavljen ćelijama u 2D matrici.

Igra je nastala kao Conwayjev pokušaj pojednostavljenja matematičkog modela stroja koji bi mogao stvarati svoje kopije Johna von Neumanna[9]. Igra ima i mogućnost univerzalnog Turingova stroja, odnosno, sve što je moguće izračunati algoritamski, moguće je prikazati i Conwayjevom igrom života.

Svaka ćelija može biti aktivna ili neaktivna, odnosno, živa ili mrtva; simulacija se odvija po koracima, a stanje svake ćelije u nekom koraku određuje se u ovisnosti o stanju okolnih ćelija u prethodnom koraku prema nekoliko jednostavnih pravila[2]:

1. **Preživljavanje:** Svaka ćelija s dva ili tri susjeda će u sljedećem koraku ostati u istom stanju.
2. **Rađanje:** Svaka neaktivna ćelija s točno tri susjeda će postati aktivna.
3. **Umiranje:** Svaka ćelija s više od tri ili manje od dva susjeda će umrijeti od prenapučenosti ili usamljenosti.

Također, Conway je postavio skup osnovnih uvjeta na samu igru:

1. Ne smije postojati početni uzorak za koji je jednostavno dokazati da populacija može neograničeno rasti.
2. Moraju postojati početni uzorci za koje *izgleda* da mogu neograničeno rasti.
3. Trebaju postojati jednostavni početni uzorci koji mogu rasti i mijenjati se dulje vrijeme završavajući jednim od tri ishoda:
4. potpuno nestajanje (zbog prenapučenosti ili nedovoljne populacije),

5. ostajanje u stabilnoj konfiguraciji koja se više ne mijenja,
6. dolazak u oscilirajuće stanje, u kojem zauvijek se ponavljaju dva ili više ciklusa.



Slika 2.1: Neki od najpoznatijih uzoraka

2.2. Simulator *Framsticks*

Framsticks je 3D simulator umjetnog života razvijen s ciljem upotrebe za više različitih primjena te, kao takav, nije specijaliziran ni za što posebno, već omogućava veoma općenito korištenje ne unoseći nikakva umjetna ograničenja na veličinu ili složenost jedinki.

Osim fizičkih svojstava jedinki, u sustavu *Framsticks* koriste se i neuronske mreže za suradnju i komunikaciju među jedinkama.

Sustav *Framsticks* sastoji se od skupa alata:

1. **Framsticks CLI** – sučelje naredbenog retka služi za napredno korištenje u slučaju dugotrajnih simulacija ili simulacija na udaljenim računalima, na kojima nije dostupno grafičko sučelje.
2. **Framsticks GUI** – grafičko sučelje. U ovom su modulu sva stvorenja, genotipi i svijet prikazani vizualno, te je moguća korisnička interakcija.
3. **Framsticks Viewer** – grafičko sučelje koje služi samo za prikaz bića određenih odabranim genotipima.
4. **Framsticks Theater** – grafičko sučelje i simulator s nekolicinom predodređenih predstava. Potrebna i moguća korisnička interakcija je minimalna.
5. **Framsticks Editor (FRED)** – grafičko sučelje koje omogućava korisnicima jednostavno dizajniranje stvorenja ne obazirući se na genetsko kodiranje.
6. **Framsticks Server i klijent** – mrežni poslužitelj i klijent. Poslužitelj nudi sučelje naredbenog retka s time da se naredbe šalje preko mreže. Klijent može biti

jednostavno grafičko sučelje prema poslužitelju ili prikaz virtualnog svijeta na poslužitelju. Osim toga, klijenti se mogu koristiti i za distribuiranu evoluciju, međusobnu interakciju i slično.

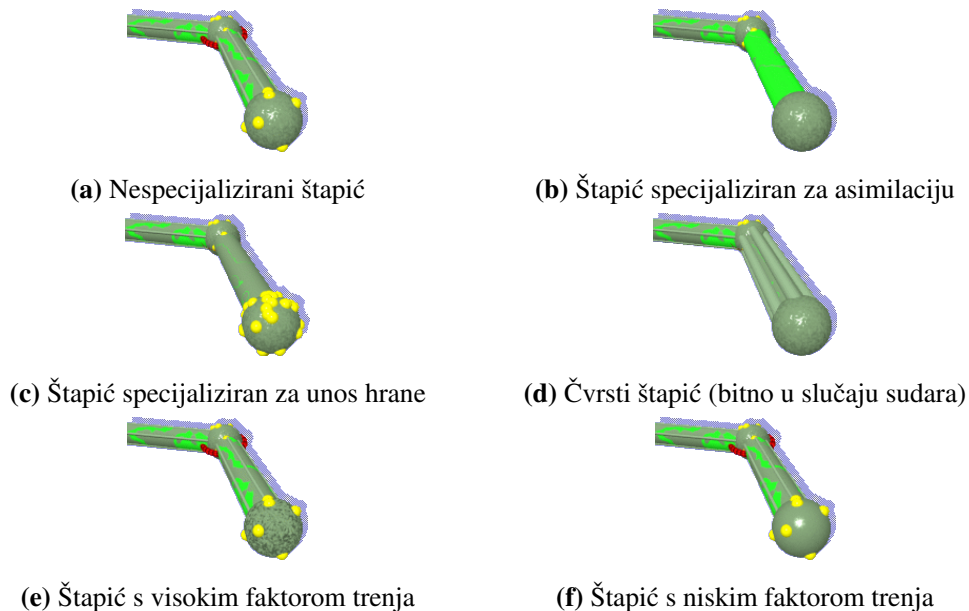
7. Ostali pomoćni programi

U simulaciji u obzir se uzimaju interakcije među objektima – trenje, prigušenje, gubici energije, gravitacija i uzgon.

Kao i pri svakoj simulaciji, potrebno je postaviti parametre ili za precizniju ili za bržu izvedbu. U evolucijskom računanju naglasak se često stavlja na brzinu zato što je obično potreban vrlo velik broj koraka da bi se došlo do željenog rezultata. Za ubrzanje računanja u *Framsticks* simulatoru neki su izračuni zanemareni, kao što je detekcija kolizije između dijelova istog organizma.

Framsticks jedinka sastoji se od dva dijela: tijela i mozga. Temeljni su element dva štapića spojena zglibom, a dijelovi i spojevi imaju vlastite značajke kao što su pozicija, orijentacija, masa, detekcija blizine, čvrstoća, trenje i sl. (Slika 2.2, preuzeto sa [1]).

Mozak je neuronska mreža. Neuron može obrađivati podatke, ali može i interagirati s tijelom kao senzor ili kao aktuator.



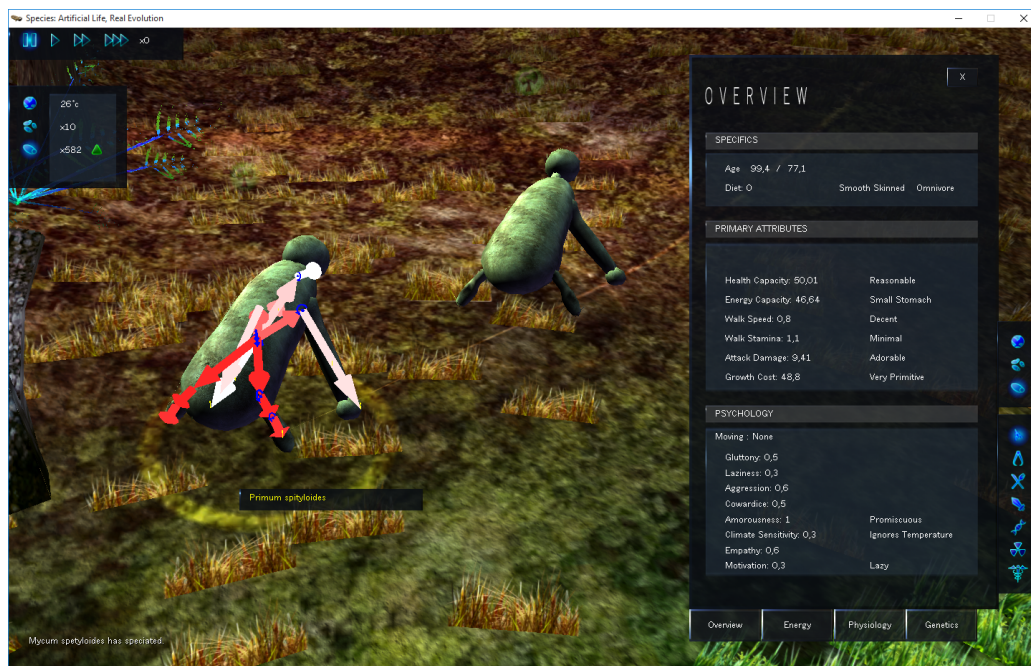
Slika 2.2: Neki tipovi štapića

2.3. Računalna igra *Species ALRE*

Računalna igra *Species ALRE* (Artificial life, real evolution), trenutno u *alfa* fazi razvoja, iako formalno nije namijenjena proučavanju evolucije s akademskog stajališta, nego prvenstveno kao računalna igra pokazala se kao dobar izbor za proučavanje zahvaljujući svojoj relativno dobroj točnosti i preciznosti, kao i veoma jednostavnom i intuitivnom korisničkom sučelju te dobrim mogućnostima prikaza povijesti i genetskog stabla.

Prema vlastitim navodima *Species* je "znanstveno precizna računalna igra simulacije evolucije, a bazira se na temeljnim evolucijskim principima: varijaciji, mutaciji i prirodnoj selekciji".

Za svaku vrstu (i jedinku) postoje svojstva koja određuju oblik, veličinu, način prehrane, način reprodukcije, brzinu kretanja i ostale bitne parametre (Slika 2.3). Sva se svojstva mogu mijenjati evolucijom te se na taj način odvija diverzifikacija vrsta.



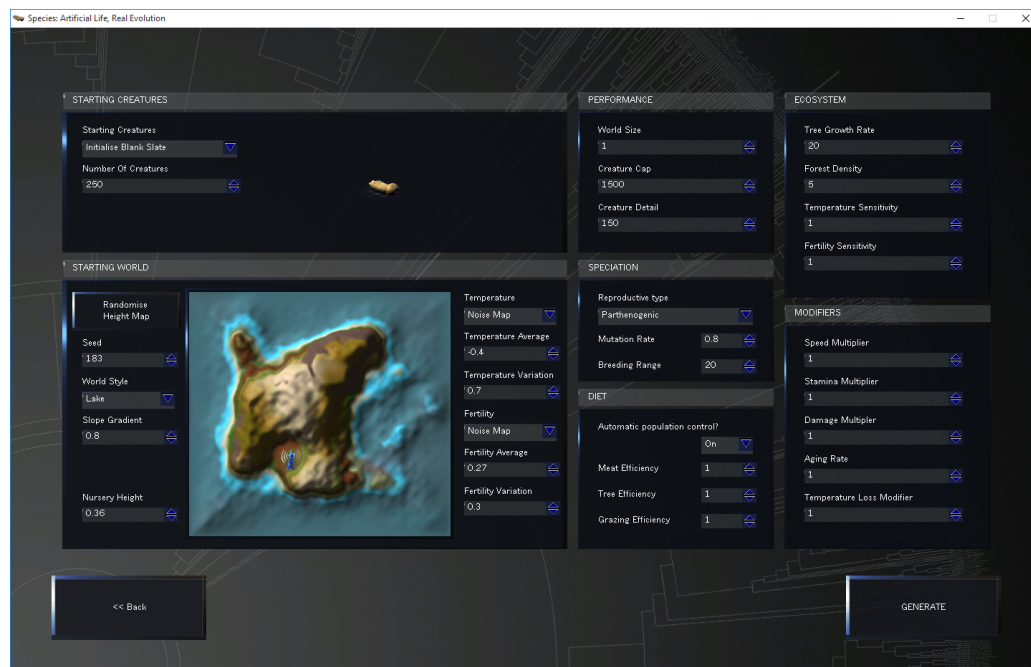
Slika 2.3: Svojstva vrste

2.3.1. Simulacija

Simulacija se pokreće postavljanjem inicijalnih postavki svijeta u kojem će se odvijati (Slika 2.4). Na tom se ekranu mogu podešavati postavke simulacije kao što su

veličina generiranog svijeta, broj i oblik početnih jedinki, način reprodukcije, faktor iskoristivosti hrane itd.

Simulacija se pokreće s postavkama definiranim u prethodnom koraku te započinje *samostalni* razvoj. U našem primjeru kreće se s populacijom od 250 jedinki osnovne preddefinirane vrste. Budući da su jedinke raspoređene po cijeloj površini bez obzira na pogodnost okruženja, u početku populacija znatno pada: s 250 na oko 50 jedinki. Te se jedinke zatim formiraju u prividne grupe i tada započinje predmet našeg promatranja – evolucija.



Slika 2.4: Postavke simulacije

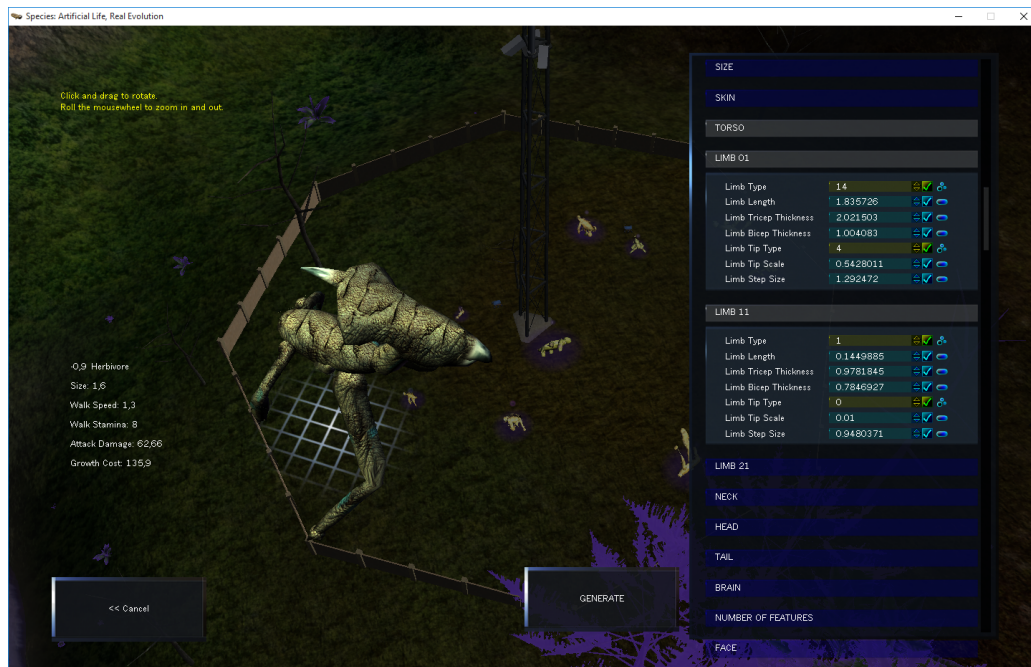
S obzirom na znatnu fizičku udaljenost između grupa, kao i na činjenicu da vrste evoluiraju prilagođavajući se lokalnim uvjetima, nakon nekog vremena može se primijetiti znatna raznolikost vrsta (Slika 2.5).

Nakon što je simulacija započela korisnik može utjecati na virtualni svijet na više načina. Indirektno promjenom temperature, plodnosti ili razinom mora, postavljanjem ograda za odvajanje grupa populacije, te direktno, hranjenjem jedinki, zračenjem (povećava vjerojatnost mutacije) ili ciljano, prepuštajući simulatoru da umjetnom selekcijom potiče razvoj odabranih svojstava.



Slika 2.5: Kladogram

Još je jedan zanimljiv način pritiska selekcije *uzgajalište* (eng. *nursery*) (Slika 2.6). To je izolirani dio svijeta u koji se postavi jedno ili više bića te se u dijaloškom okviru odaberu razne postavke za željeni ishod, nakon čega simulator sâm malo po malo potiče *dobre*, odnosno, najbolje dostupne jedinice, a uklanja one *lošijih* svojstava. Specificirati se može svaki dio bića posebno – udovi, tijelo, glava, mozak, odnosno, osobnost (agresivnost, lijenost, osjetljivost na klimatske promjene...).



(a) Odabir svojstava



(b) Napredak

Slika 2.6: Uzgajalište

3. Praktični rad

3.1. Korištene tehnologije

U ovom se radu obrađuju dva, svaki za sebe vrlo složena koncepta: *umjetni život i simulacija u 3D prostoru*. Budući da je u fokusu rada umjetni život, za simulaciju 3D okruženja korišten je postojeći simulator otvorenog koda *Minetest*[5].

3.1.1. Simulator Minetest



Simulator Minetest je istovremeno proširiva pogonska jezgra igre (eng. *game engine*) i sama igra *minetest_game*, otvorenog koda i slobodno dostupna za preinake. Simulirano okruženje u *Minestestu* je izgrađeno od *voxela* (3D piksela), te su programski ostvarene fizičke zakonitosti kao što su gravitacija i detekcija sudara. Simulacija agenata te njihovo ponašanje ostvareni su kao *mod* (modifikacija ili dodatak originalnom programu) u programskom jeziku LUA, pomoću postojećeg programskog sučelja predviđenog za izravnu interakciju s programom putem drugih programa koji koriste utvrđeni protokol za komunikaciju (eng. *Application programming interface - API*).

Sam simulator *Minetest* napisan je najvećim dijelom u jeziku C++, a za prikaz virtualnog okruženja koristi se *Irrlicht Engine*[3], koji, u ovom slučaju, kao grafički sustav koristi *OpenGL*[7].

Minetest je organiziran na principu klijent-server arhitekture, pri čemu poslužitelj vodi računa o programskoj logici, a klijent se brine isključivo za iscertavanje svijeta te procesiranje korisničkih unosa.

3.1.2. Biblioteka programa OpenGL



OpenGL je skup standarada koji definiraju sučelja pomoću kojih računalni programi mogu iskoristiti puni potencijal suvremenih grafičkih kartica, te istovremeno i skup biblioteka koje ostvaruju sučelja definirana OpenGL standardom. OpenGL biblioteke podrazumijevaju rad s 2D i 3D prikazima te su predviđene za jednostavnu prenosivost između računalnih platformi i sustava. U sklopu ovog rada OpenGL se koristi kao pozadinska tehnologija okosnice *Irrlicht* za prikaz 3D svijeta.

3.1.3. Skriptni jezik Lua



Lua je dinamički, slabo tipizirani, ugrađiv skriptni jezik koji podržava proceduralno, objektno orijentirano, funkcijsko te programiranje vođeno podacima. Zbog navedenih svojstava, Lua je jezik koji se često koristi upravo kao ugrađeni jezik za prilagođavanje pojedinih dijelova programa – prilagođavanje korisničkog sučelja, skriptiranje akcija u računalnim igrama i sl.

3.2. Programsko ostvarenje

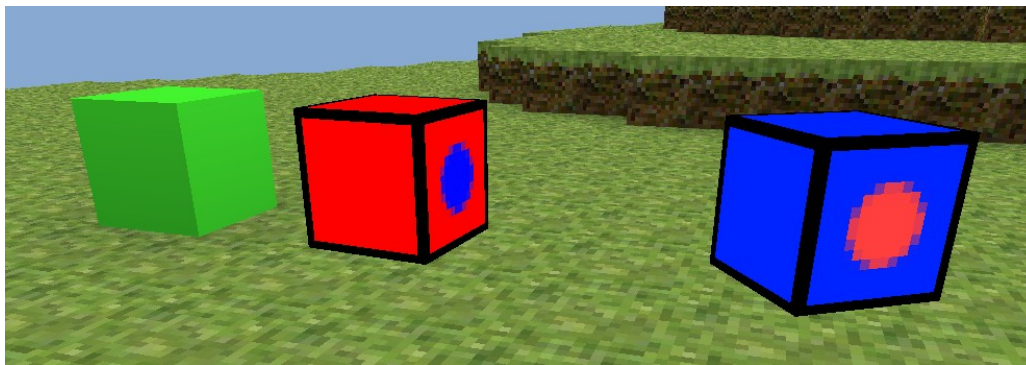
3.2.1. Definicija modela virtualnog života i okruženja

Kao što je već navedeno, ovaj simulator izveden je kao proširenje za virtualno okruženje *Minetest*. Uslijed toga, koristi se već postojeći model stvarnosti, određen *voxelima*.

Unatoč velikoj raznolikosti biljnog i životinjskog svijeta, prema najjednostavnijoj kategorizaciji živa se bića mogu stvrstati u tri glavne kategorije: *biljke*, *biljojed* i *mesojedi*, te su te kategorije kao takve ostvarene i u simulatoru.

3.2.2. Virtualni svijet

U okviru ovog rada virtualni svijet ograničen je na veličinu $80*80*80$ *voxela*, kako bi se simulacija mogla odvijati u razumnom vremenu. Generiranje svijeta prepušteno je okruženju *Minetest*, uz podešavanje parametara generiranja konfiguracijskom datotekom. Kamera je smještena na poziciju "igrača", te se može slobodno kretati kroz 3D prostor tipkama *w*, *a*, *s*, *d*, *shift* i *razmak*.



Slika 3.1: Postojeća bića: biljka, mesojed i biljojed

Biljke su ostvarene kao dio okoline (Node), a biljojedi i mesojedi kao `LuaEntitySAO`. Na taj se način za sve entitete jednostavno primjenjuju fizikalna pravila, odnosno u prostoru se ponašaju dosljedno i očekivano.

Za razliku od stvarnog svijeta, u modelu korištenom u simulaciji ne postoji koncept starosti, već su jedina dva načina umiranja umiranje od nedostatka energije (gladi) te umiranje kao žrtva lova predatora (biljke biljojeda, a biljojedi mesojeda).

Simulacija se odvija u diskretnim koracima trajanja 1/20 sekunde, a prikaz se osvježava brzinom iscrtavanja podržanom od strane grafičke kartice.

3.2.3. Konfiguracija

Konfiguracija simulatora podijeljena je na dva dijela. Općeniti parametri koji se tiču generiranja 3D svijeta (grubost mape, generiranje planina i špilja, razina vode i sl.) podešavaju se u konfiguracijskoj datoteci `minetest.conf` u korijenskom direktoriju, odakle se simulator i pokreće. Parametri bića u simulaciji podešavaju se promjenom vrijednosti varijabli pri vrhu datoteke `games/alsim/mods/alsim/init.lua`. Prilagodljivi parametri navedeni su u tablici 3.1.

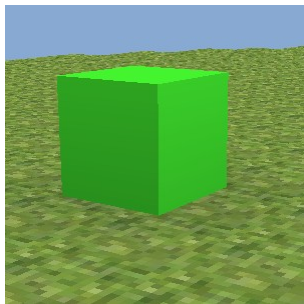
| Naziv parametra | Opis |
|---------------------------------------|---|
| <code>plants_spread_interval</code> | Vrijeme u sekundama između razmnožavanja biljaka |
| <code>plants_spread_chance</code> | Vjerojatnost da će biljka stvoriti potomstvo. Računa se prema formuli $1/\text{plants_spread_chance}$ |
| <code>herbivore_initial_energy</code> | Razina energije pri stvaranju novog biljojeda |

| | |
|--------------------------------------|---|
| herbivore_passive_energy_consumption | Pasivna potrošnja energije (bez kretanja) za biljojede |
| herbivore_walking_energy_consumption | Potrošnja energije za biljojede pri normalnom hodanju |
| herbivore_jumping_energy_consumption | Potrošnja energije za biljojede pri skakanju |
| herbivore_speed | Brzina kretanja biljojeda (u <i>voxelima</i>) |
| herbivore_view_distance | Radijus vidnog polja biljojeda (u <i>voxelima</i>) |
| herbivore_force_hunt_threshold | Donja granična razina energije ispod koje biljojed bezuvjetno kreće u "lov" |
| herbivore_force_mate_threshold | Gornja granična razina iznad koje biljojed traži partnera za razmnožavanje |
| energy_in_plant | Količina energije koju biljojed dobije pojevši biljku |
| carnivore_initial_energy | Razina energije pri stvaranju novog mesojeda |
| carnivore_passive_energy_consumption | Pasivna potrošnja energije (bez kretanja) za mesojede |
| carnivore_walking_energy_consumption | Potrošnja energije za mesojede pri normalnom hodanju |
| carnivore_jumping_energy_consumption | Potrošnja energije za mesojede pri skakanju |
| carnivore_speed | Brzina kretanja mesojeda (u <i>voxelima</i>) |
| carnivore_view_distance | Radijus vidnog polja mesojeda (u <i>voxelima</i>) |
| carnivore_force_hunt_threshold | Donja granična razina energije ispod koje mesojed bezuvjetno kreće u "lov" |

| | |
|--------------------------------|---|
| carnivore_force_mate_threshold | Gornja granična razina iznad koje mesojed traži partnera za razmnožavanje |
| energy_in_herbivore | Količina energije koju mesojed dobije pojevši biljojeda |
| plants_count_cap | Maksimalni broj biljaka u svijetu |
| initial_plants_count | Broj biljaka koje se generiraju neposredno nakon stvaranja svijeta |
| initial_herbivores_count | Broj biljojeda koji se generiraju neposredno nakon stvaranja svijeta |
| initial_carnivores_count | Broj mesojeda koje se generiraju neposredno nakon stvaranja svijeta |

Tablica 3.1: Popis prilagodljivih parametara

3.2.4. Biljke



Slika 3.2: Model biljke je kocka zelene boje

Zbog svoje jednostavnosti i nemogućnosti kretanja biljke su ostvarene kao poseban tip bloka u okolini. Biljke se razmnožavaju samostalno, *kloniranjem*, odnosno bez potrebe za partnerom. Iz tog je razloga moguće pokrenuti simulaciju samo s jednom biljkom, koja će se postupno proširiti na cijelo dostupno područje.

Kao model biljke koristi se kocka zelene boje bez ikakvih razlika između stranica (slika 3.2).

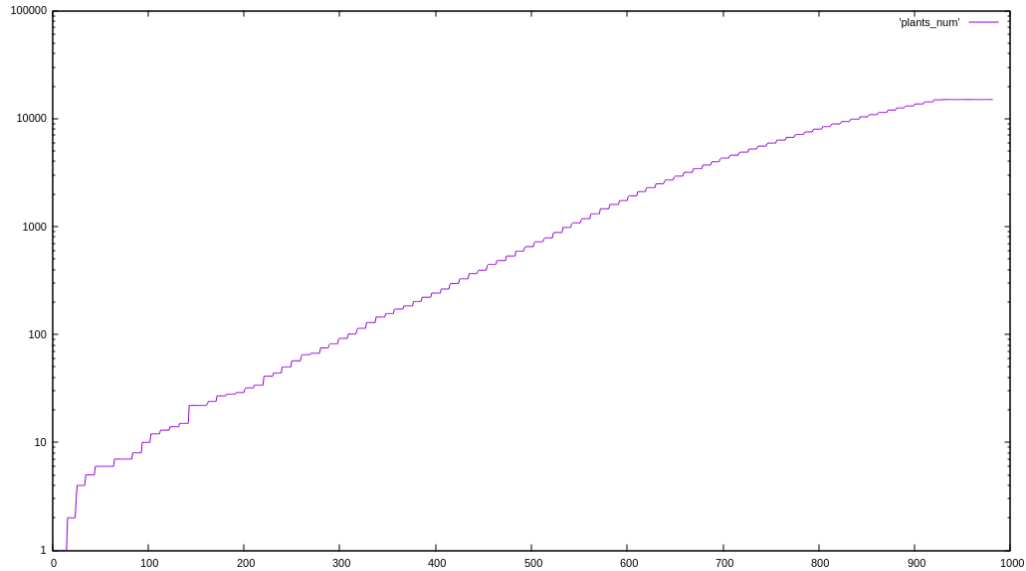
Pretpostavljeni je izvor energije Sunce, iz kojeg biljke, kao početak hranidbenog lanca, dobivaju svoju energiju.

Upravo zato kod biljaka ne postoji mogućnost kako skladištenja tako ni potrošnje energije, već biljke služe samo kao prilagodnik sunčeve energije za konzumaciju biljojedima.

Brzina razmnožavanja biljaka određena je parametrima `plants_spread_interval` i `plants_spread_chance`. Za razmnožavanje biljaka koristi se *Active block modifier* API pozivom `minetest.register_abm`. Kao parametre taj poziv prima polje naziva (tipova) blokova za koje se izvršava, frek-

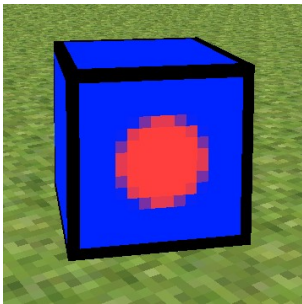
venciju pokretanja akcije (`plants_spread_interval`), inverznu vjerojatnost pokretanja akcije (`plants_spread_chance`) za svaku biljku i funkciju kojom se definira akcija koja se izvršava za odabrane jedinke. Primjer rasta broja biljaka prikazan je na slici 3.3 (broj biljaka ograničen je na 15000). Pseudokod akcije prikazan je u nastavku:

```
ponavljaj {  
  ako je broj biljaka veći od dozvoljenog {  
    odustani  
  }  
  
  odaberi jedinku za razmnožavanje  
  odaberi slučajan voxel unutar zadanog radijusa  
  dok je odabrani voxel zauzet {  
    povećaj y koordinatu odabira za 1  
    ako je dosegnut maksimalni broj ponavljanja {  
      odustani  
    }  
  }  
  
  postavi biljku u odabrani voxel  
  pribroji novu biljku u statistiku  
}
```



Slika 3.3: Razmnožavanje biljaka – na x -osi je vrijeme u sekundama od početka simulacije, a na y -osi broj biljaka prikazan u logaritamskoj skali

3.2.5. Biljojedi



Slika 3.4: Model biljojeda je kocka plave boje, s prednjom stranom označenom crvenim krugom

Biljojedi su u hranidbenom lancu jednu stepenicu iznad biljaka, te se biljkama hrane. Za prikaz biljojeda koristi se kocka plave boje s crnim bridovima, a prednja je strana označena crvenim krugom (slika 3.4).

Kako su biljojedi pokretni entiteti, za izvedbu se koristi ugrađeni tip `minetest API`-ja `LuaEntitySAO` te API poziv za dodavanje entiteta `minetest.register_entity` [6].

Za razliku od biljaka, kod biljojeda je ostvaren koncept energije, te se energija iz zalihe troši različitim brzinama u ovisnosti o tome što biljojed trenutno radi. Osnovna potrošnja definirana je iznosom varijable `herbivore_passive_energy_consumption` i primjenjuje se uvijek, bez obzira na trenutno stanje i ponašanje biljojeda.

U tijeku simulacije, biljojed mijenja sljedeća stanja: *wander*, *hunt*, *eat*, *find_mate*, *mate*. Svako od stanja detaljnije je opisano u nastavku:

1. *wander* – Biljojed nasumično luta svijetom. Primjenjuje se kada je biljojed sit i

ne pronalazi partnera za razmnožavanje.

2. *hunt* – Biljojed traži i "lovi" biljku. Primjenjuje se kada razina energije biljojeda padne ispod granice određene s `herbivore_force_hunt_threshold` ili kada ne može pronaći partnera za razmnožavanje. Ako unutar radijusa vidljivosti ne postoji niti jedna biljka, privremeno prelazi u stanje *wander*.
3. *eat* – Biljojed se hrani "ulovljenom" biljkom. Ovo stanje traje najduže jedan korak simulacije, nakon čega biljojed prelazi u jedno od preostalih stanja.
4. *find_mate* – Biljojed traži partnera za razmnožavanje. U ovo stanje prelazi ako je razina energije iznad vrijednosti određene s `herbivore_force_mate_threshold`. Ako ne može pronaći partnera za razmnožavanje, prelazi u stanje lova (*hunt*).
5. *mate* – Biljojed se razmnožava. U ovo stanje prelazi isključivo sinkrono u slučaju dodira s drugim biljojedom. U ovom stanju nastaje biljojed-dijete, procesom opisanim niže. Kao i u slučaju stanja *eat*, i u ovom stanju biljojed se može nalaziti najdulje jedan korak simulacije, nakon čega prelazi u jedno od preostalih stanja.

Kretanjem biljojedi također troše energiju i to hodanjem – `herbivore_walking_energy_consumption`, a u slučaju da naiđu na prepreku i skakanjem – `herbivore_jumping_energy_consumption`. Energija potrebna za skakanje je, očekivano, redovito višestruko veća od energije potrebne za hodanje.

Kod biljojeda postoji radijus vidljivosti, definiran varijablom `herbivore_view_distance`, unutar kojega su biljojedi "svjesni" svoje okoline. Sve izvan tog radijusa njima je nepoznato.

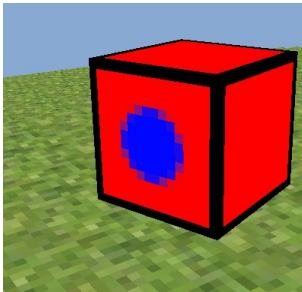
Kako bi potrošenu energiju nadomjestili, biljojedi se hrane biljkama. Ukoliko je biljojed gladan i unutar radijusa vidljivosti postoji biljka, biljojed se počinje kretati prema toj biljci te će je pojesti, za što mu je dovoljno približiti se biljci tako da je dodiruje. Jedenje se odvija trenutačno – biljka nestaje iz svijeta, a biljojedova se energija povećava za iznos definiran varijablom `energy_in_plant`.

Kada je biljojed sit, unutar radijusa vidljivosti traži partnera za razmnožavanje. Ako partnera ne uspije pronaći, ili nastavlja s hranjenjem stvarajući tako zalihu energije ili nasumično luta svijetom. Ako pronađe partnera i taj se partner odazove, tada se oba biljojeda počinju kretati jedan prema drugome, te kada se dodirnu nastaje novi

biljojed-dijete. Prijenos energije se odvija prema pravilu: biljojed-dijete će dobiti trećinu ukupne energije oba roditelja, a svaki od roditelja će ostati samo s trećinom svoje početne energije (ukupno, po jedna trećina energije svakog od roditelja gubi se pri razmnožavanju).

Ako biljojed ne traži hranu niti partnera za razmnožavanje nasumično luta svijetom.

3.2.6. Mesojedi



Slika 3.5: Model mesojeda je kocka crvene boje, s prednjom stranom označenom plavim krugom

Mesojedi se hrane isključivo biljojedima i na vrhu su hranidbenog lanca. U simulatoru su prikazani kockom crvene boje s crnim bridovima, a prednja je strana označena plavim krugom (slika 3.5).

Kao i u slučaju biljojeda, za izvedbu se koriste ugrađeni tip `minetest` API-ja `LuaEntitySAO` te API poziv za dodavanje entiteta `minetest.register_entity` [6].

Logika i programski kod mesojeda analogni su onome biljojeda, osim u stanju lova, gdje se meta (biljojed) može kretati, za razliku od biljaka te se za nadopunu energije pojevši biljojeda koristi vrijednost `energy_in_herbivore`. Za uspješan lov, mesojedu je dovoljno približiti se biljojedu i dodirnuti ga. Nakon toga, biljojed nestaje iz virtualnog okruženja, a mesojedova energija se nadopunjava.

3.2.7. Pseudokod simulatora

Svaka jedinka donosi odluke samostalno te je sama za sebe u potpunosti nezavisna o ostalim jedinkama u svijetu. Jedina interakcija s drugim jedinkama omogućena je vanjskim pozivom funkcije `receive_mate_call`, pri čemu se predmetna jedinka može, ali i ne mora odazvati, ovisno o svojem trenutnom stanju.

Kako je svaka jedinka nezavisna, interna logika, kao i osvježavanje pozicije u svijetu odvijaju se paralelno.

Budući da je gotovo sva funkcionalnost dijeljena između biljojeda i mesojeda, dovoljno je prikazati samo jedan, zajednički pseudokod. Navedeni pseudokod, zajedno sa pseudokod za razmnožavanje biljaka, predstavlja potpuni opis simulatora. Sljedeći odsječak pokreće se za svaku jedinku mijenjajući tako svojstva iste:

```

ponavljaj {

    pomakni se prema trenutnom cilju
    smanji energiju za odgovarajući iznos

    ako je razina energije ispod minimalne dozvoljene {
        prijeđi u stanje 'hunt'
    }
    ako je stanje 'wander' {
        slučajno odaberi ciljnu poziciju
    }
    ako je stanje 'hunt' i lovina se dodiruje {
        prijeđi u stanje 'eat'
    }
    ako je stanje 'hunt' {
        pronađi lovinu (biljku ili biljojeda)
        ako lovina nije pronađena {
            prijeđi u stanje 'wander'
        }
    }
    ako je stanje 'eat' {
        pojedite lovinu
        prijeđi u stanje 'wander'
    }
    ako je razina energije iznad minimalne potrebne {
        prijeđi u stanje 'find_mate'
    }
    ako je stanje 'find_mate' {
        pronađi partnera
        ako partner nije pronađen {
            prijeđi u stanje 'hunt'
        }
    }
    ako je stanje 'find_mate' i partner se dodiruje {
        prijeđi u stanje 'mate'
    }
}

```

```

}
ako je stanje 'mate' {
    razmnoži se
    prijeđi u stanje 'wander'
}

ako je energija <= 0 {
    ukloni
}
}

```

3.2.8. Korisničko sučelje

Nakon prevođenja programa i pokretanja s `bin/minetest`, prikazuje se ekran kao na slici 3.6. Na tom je ekranu potrebno odabrati "New", kako bi se stvorilo novo okruženje u kojemu će se simulacija odvijati. Na sljedećem ekranu (slika 3.7) dovoljno je samo pritisnuti tipku *enter* i simulacija započinje. Nakon što je generiranje svijeta završilo, prikazuje se slika slična slici 3.8.

Odmah nakon pokretanja u direktoriju odakle se simulator pokreće stvara se datoteka dinamički generirana imena `<timestamp>_stats`. U tu se datoteku svake sekunde zapisuju statistike o broju biljaka, biljojeda i mesojeda i to u formatu:

```
vremenska_oznaka biljke biljojedi mesojedi
```

Iz te se datoteke mogu jednostavno iscrtati grafovi koristeći program *gnuplot*. naredba za generiranje grafova koja se koristi u radu jest: `plot 'stats' using 1:2 with lines title 'biljke' lt rgb '#00FF00', " using 1:3 with lines title 'biljojedi' lt rgb '#0000FF', " using 1:4 with lines lt rgb "#FF0000" title 'mesojedi'`

3.3. Rezultati ispitivanja

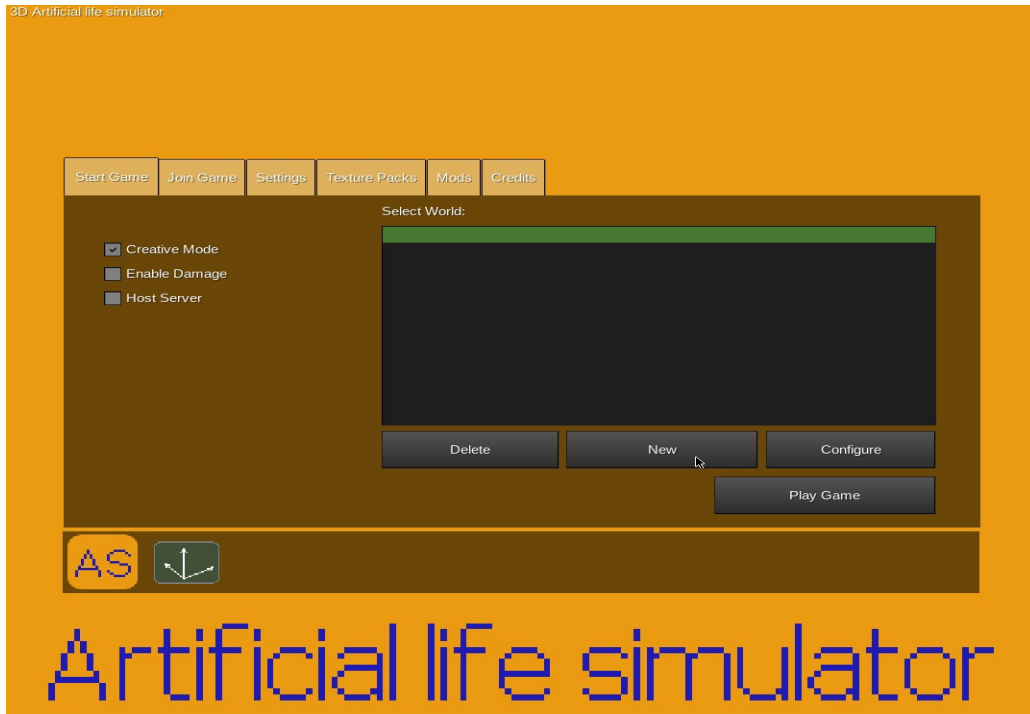
Ishod simulacije ovisi ponajprije o broju pojedinih jedinki na početku simulacije. Na slikama 3.9 i 3.10 mogu se vidjeti dva tipična scenarija: *Biljke prevladavaju* i *Svi izumiru*.

| Naziv parametra | Vrijednost |
|--------------------------------------|-------------------|
| plants_spread_interval | 3 |
| plants_spread_chance | 100 |
| herbivore_initial_energy | 500 |
| herbivore_passive_energy_consumption | 0.001 |
| herbivore_walking_energy_consumption | 0.4 |
| herbivore_jumping_energy_consumption | 3 |
| herbivore_speed | 5 |
| herbivore_view_distance | 15 |
| herbivore_force_hunt_threshold | 400 |
| herbivore_force_mate_threshold | 700 |
| energy_in_plant | 500 |
| carnivore_initial_energy | 600 |
| carnivore_passive_energy_consumption | 0.01 |
| carnivore_walking_energy_consumption | 0.5 |
| carnivore_jumping_energy_consumption | 3 |
| carnivore_speed | 5 |
| carnivore_view_distance | 10 |
| carnivore_force_hunt_threshold | 600 |
| carnivore_force_mate_threshold | 1000 |
| energy_in_herbivore | 800 |
| plants_count_cap | 10000 |
| initial_plants_count | 200 |
| initial_herbivores_count | 100 |
| initial_carnivores_count | 10 |

Tablica 3.2: Pretpostavljene vrijednosti parametara

3.3.1. Biljke prevladavaju

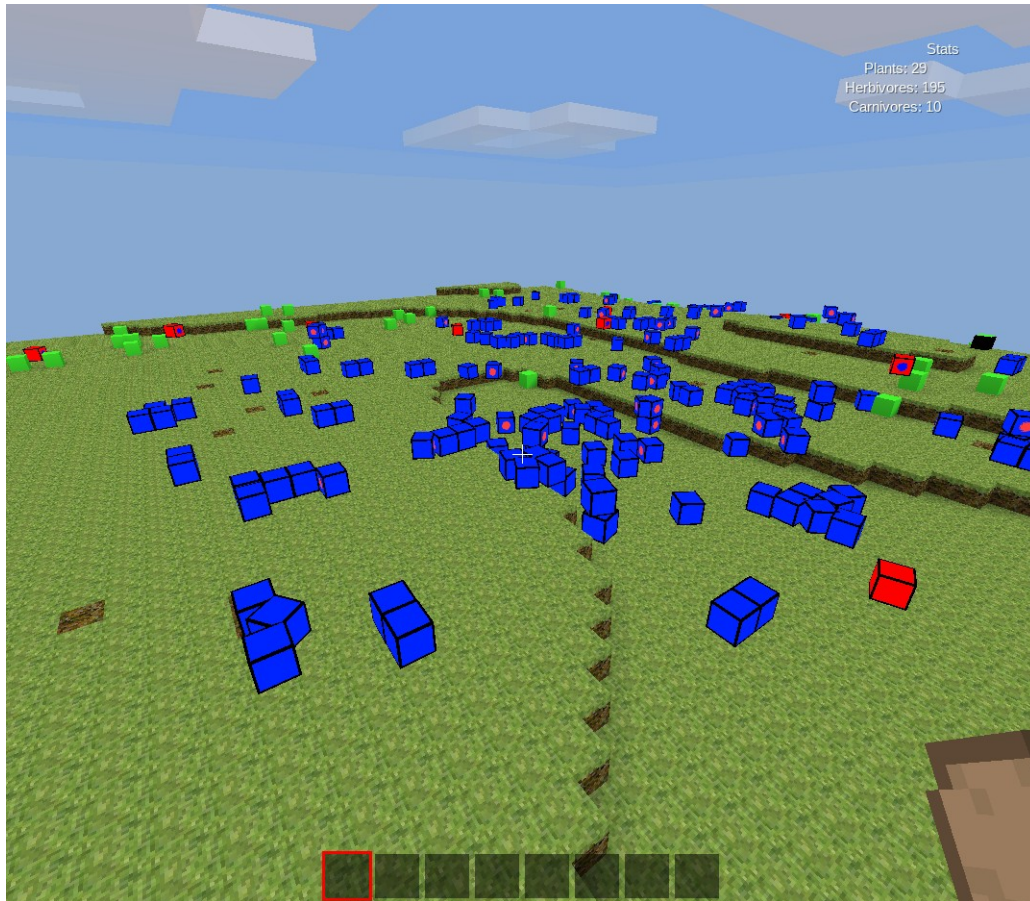
Najčešći ishod simulacije je upravo scenarij *Biljke prevladavaju*, prikazan na slici 3.9. Iako simulacije počinju s različitim brojem jedinki, završno stanje i put do završnog stanja su slični.



Slika 3.6: Glavni izbornik



Slika 3.7: Postavke generiranja svijeta



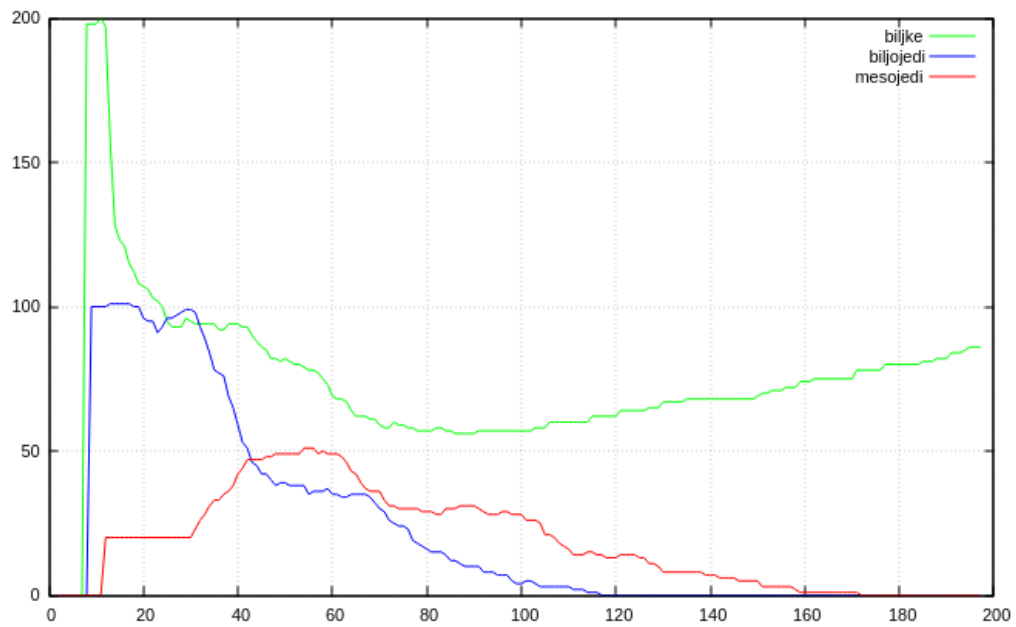
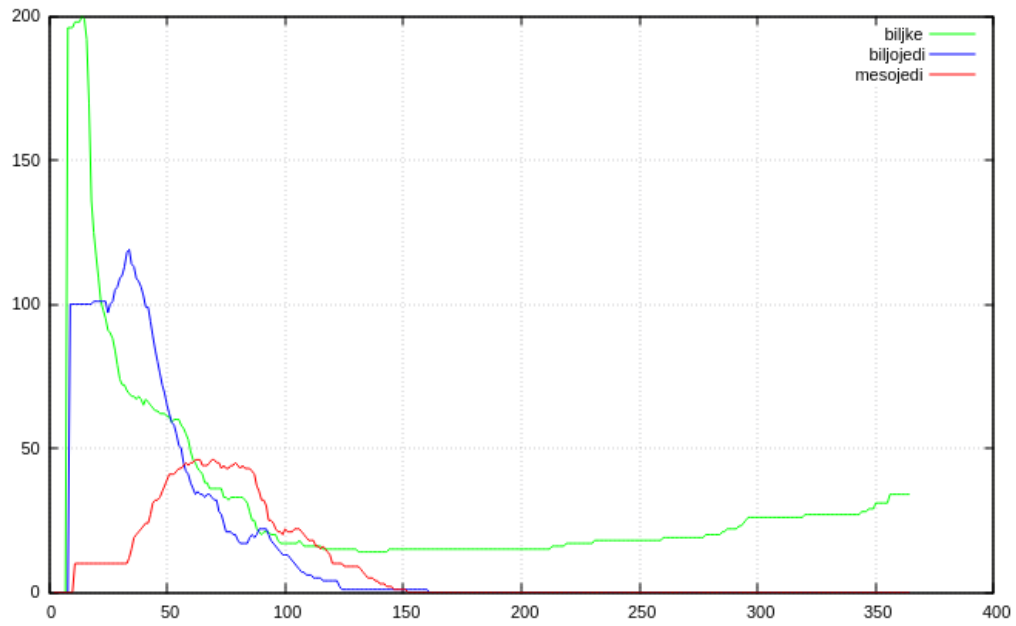
Slika 3.8: Stvoreni svijet

U početku hrane ima u izobilju. Zbog toga se biljojedi razmnožavaju što uzrokuje još veću potražnju hrane. Zbog većeg broja biljojeda, drastično se smanjuje broj biljaka. Istovremeno, veći broj biljojeda predstavlja više hrane za mesojede, koji se sada počinju vrlo brzo razmnožavati. Kao i u slučaju biljojeda i biljaka, i u slučaju mesojeda i biljojeda dolazi do prevelike potražnje za hranom (biljojedima), a premale ponude – biljojeda je sve manje i manje djelomično zbog utjecaja mesojeda, a djelomično zbog nedostatka hrane nakon prve faze razmnožavanja. Sada se broj biljojeda smanjuje do izumiranja što direktno dovodi do izumiranja mesojeda. U nedostatku prirodnih neprijatelja, broj biljaka se stabilno povećava.

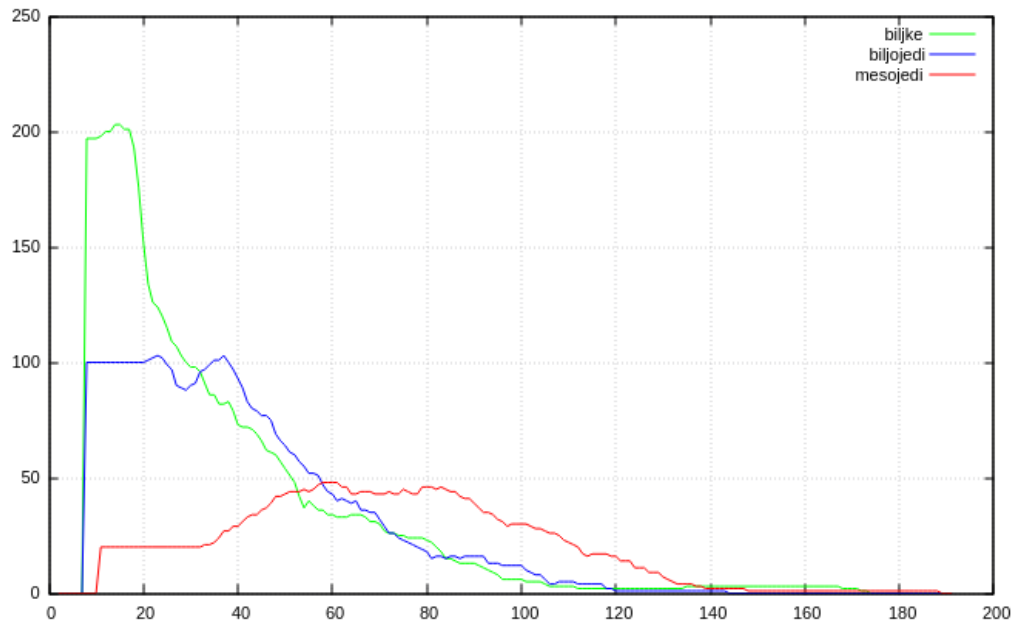
3.3.2. Svi izumiru

Alternativni razvoj događaja prikazan je grafovima na slici 3.10. U ovom scenariju primjećujemo dva značajno različita puta do istog krajnjeg stanja.

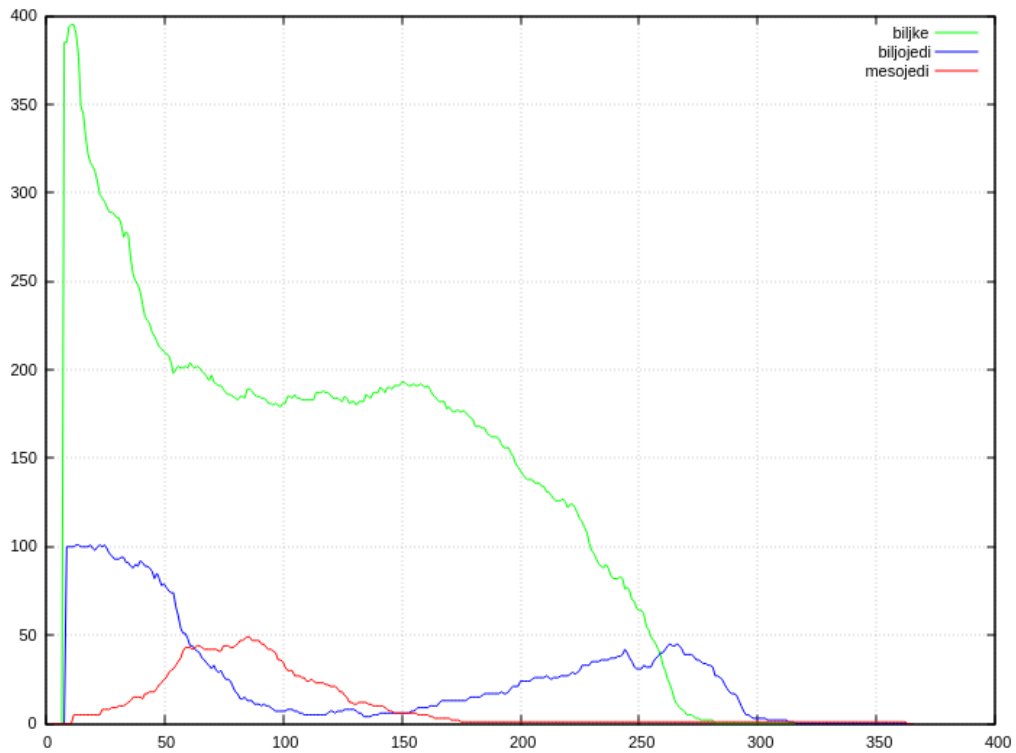
Na slici 3.10a (*Biljojedi izumiru prije*) primjećujemo početni rast broja biljojeda,



Slika 3.9: Grafovi scenarija *Biljke prevladavaju* – nakon što mesojedi dovedu biljojede do izumiranja, pa i sami ostanu bez hrane, izumiru. Za to vrijeme broj biljaka kontinuirano raste budući da više nema biljojeda.



(a) Biljojedi izumiru prije



(b) Mesojedi izumiru prije

Slika 3.10: Grafovi scenarija *Svi izumiru* – zbog prevelikog broja biljojeda, biljke izumiru. Nakon biljaka izumiru biljojedi, ostavši bez izvora energije, te naposljetku i mesojedi.

koji, kao i u scenariju *Biljke prevladavaju*, uzrokuje nestašicu biljaka. Broj biljojeda se smanjuje, čemu značajno doprinosi povećani broj mesojeda, no broj biljaka je već premali da bi uzdržavao čak i ovu smanjujuću populaciju biljojeda. Zato biljojedi i biljke izumiru otprilike istovremeno, a mesojedi nešto kasnije, kada potroše svoje zalihe energije.

Na drugom grafu (slika 3.10b – *Mesojedi izumiru prije*) vidimo da su mesojedi uspješni smanjiti broj biljojeda, te su se biljke počele oporavljati, no zbog međusobne fizičke udaljenosti ili razdvojenosti mesojeda od biljojeda mesojedi izumiru prvi, dok se biljojedi hrane preostalim biljkama i nastavljaju razmnožavati što opet dovodi do izumiranja – prvo biljaka pa zatim i biljojeda.

U oba slučaja primjećujemo gomilanje resursa (energije) i iskorištavanje zaliha čak i nakon što je taj resurs iz okoline nestao.

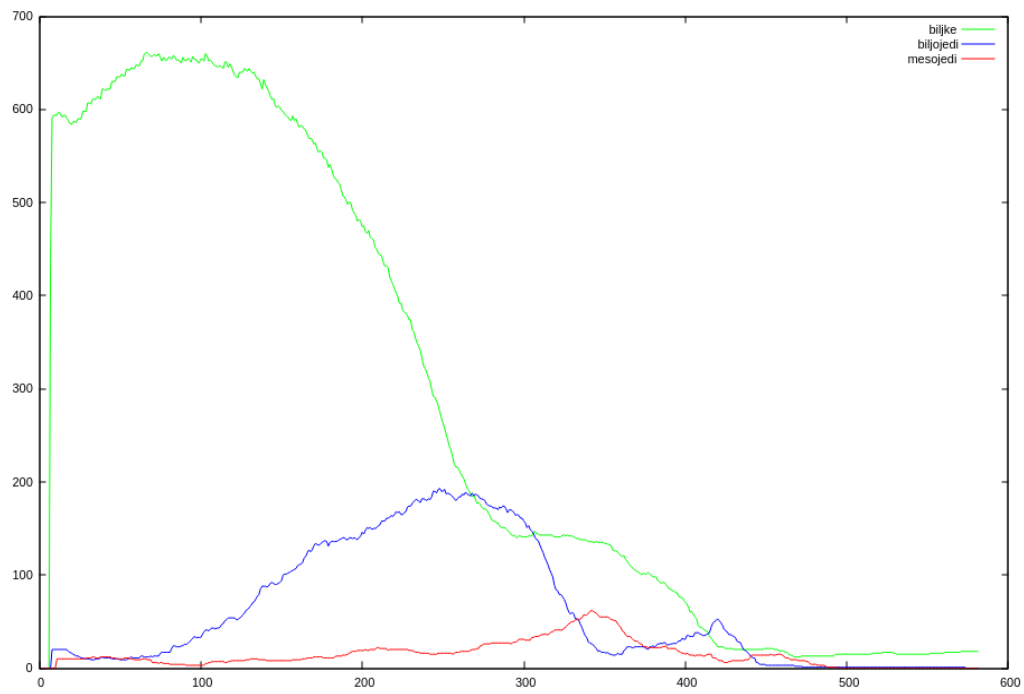
3.3.3. Alternativne postavke

Za iscrpnije rezultate testiranja simulator je pokrenut više puta i sa različitim vrijednostima parametara bića. U ovom primjeru postoji pozitivna diskriminacija prema biljojedima, što rezultira značajnim početnim rastom biljojeda.

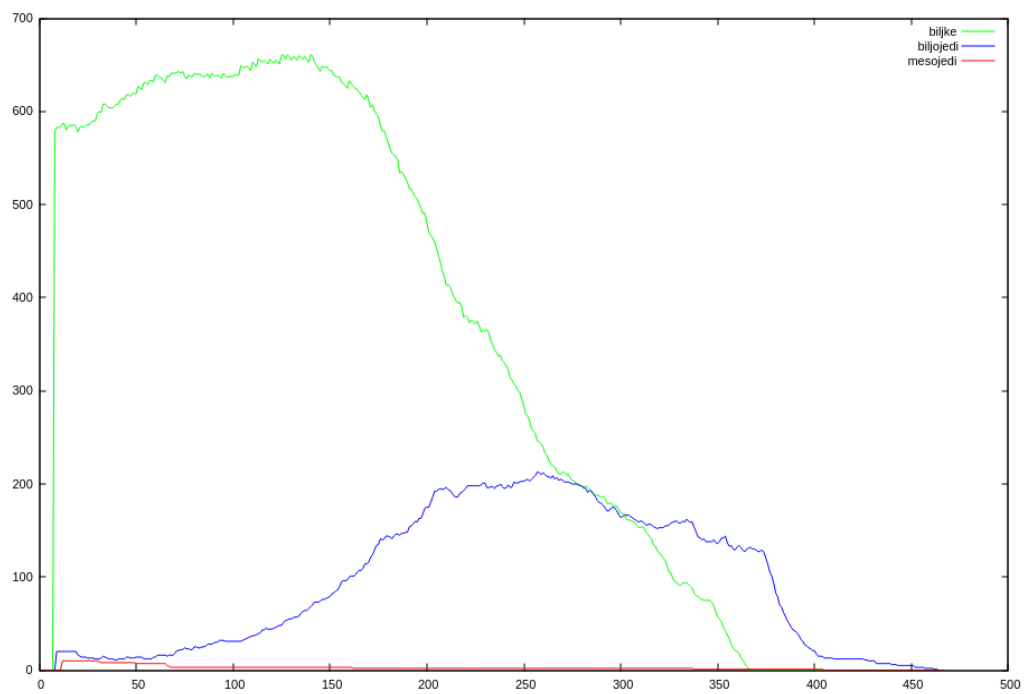
Kako se može vidjeti iz slike 3.11, iako je početak simulacije različit, odnosno biljojedi su u velikoj većini, krajnji je rezultat sličan već opisanim scenarijima: *Biljke prevladavaju* i *Svi izumiru - mesojedi izumiru prvi*. U tablici 3.3 prikazane su vrijednosti parametara različite od pretpostavljenih (prikazanih u tablici 3.3).

| Naziv parametra | Vrijednost |
|--------------------------------------|------------|
| herbivore_walking_energy_consumption | 0.3 |
| herbivore_jumping_energy_consumption | 2 |
| herbivore_speed | 6 |
| energy_in_plant | 800 |
| carnivore_speed | 4.5 |
| energy_in_herbivore | 500 |
| initial_plants_count | 600 |
| initial_herbivores_count | 20 |
| initial_carnivores_count | 10 |

Tablica 3.3: Modificirane vrijednosti parametara



(a) Biljke prevladavaju



(b) Mesojedi izumiru prije

Slika 3.11: Grafovi rezultata ispitivanja s alternativnim postavkama

3.4. Budući rad

Život je sam po sebi izrazito složen sustav. Svaki pokušaj aproksimacije života nekim umjetnim sustavom uvijek će imati potencijala za daljnje nadogradnje. Trenutna verzija simulatora nudi velik broj prilagodljivih parametara, no korištenje bi se moglo znatno pojednostavniti dodavanjem grafičkog sučelja za postavljanje vrijednosti tih parametara.

Za sada su sva bića pojedinog tipa (biljka, biljojed, mesojed) međusobno u potpunosti jednaka. U slučaju lova i jedenja lovine biće-lovac uvijek dobije jednaku količinu energije. Za realističniju simulaciju biće-lovac bi moglo biti nagrađeno s više ili manje energije, ovisno o tipu i stanju bića koje je ulovilo, potičući istovremeno lov bića koja sadrže više energije.

Statistika koja se bilježi sastoji se od tri podatka kroz vrijeme: broja biljaka, broja biljojeda i broja mesojeda. Za detaljnije analize korisnom bi se mogli pokazati još neki statistički podaci, kao što su: prosječna razina energije bića, najviša dosegnuta razina energije bića, starost jedinki, prosječna starost i sl.

U sadašnjoj verziji bića nemaju nikakvu inteligenciju, već se vode isključivo nagonima. Kao mogućnost za unapređenje ostavlja se dodavanje inteligencije pojedinim bićima, iz čega bi izravno sljedilo dodavanje genetske strukture "mozga", koja bi se mogla naslijeđivati – križati i mutirati – iz roditelja na djecu.

U slučaju postojanja genetskog zapisa, taj bi se zapis mogao izvoziti iz simulatora (npr. za daljnje proučavanje), uvoziti ili čak slučajno generirati na početku svake simulacije. Također, to bi omogućilo dijeljenje primjeraka jedinki između različitih instanci simulacija.

Genetski zapis omogućio bi i značajniju diverzifikaciju bića: osim što bi bića mogla *preferirati* jedan tip prehrane, ali pribjegavati i drugima po potrebi (svejadi), i unutar samih razreda bića mogla bi se razviti raznolikost.

Trenutni je sustav izveden u programskom jeziku *Lua*. Budući da je *Lua* skriptni jezik, zbog svoje je jednostavnosti relativno spor za kompleksnije izračune. Zbog toga bi bilo dobro izvesti cijeli sustav ili barem njegove kritične dijelove u programskom jeziku *C++* (u kojem je izvedeno i samo okruženje *Minetest*), kao integralni dio simulatora.

Postojeće je ograničenje okruženja *Minetest* fiksna brzina izvođenja simulacije (*step*). Kako bi se omogućila promjena brzine izvođenja (najčešće nauštrb preciznosti u izvođenju), potrebno je prilagoditi samu jezgru okruženja.

4. Zaključak

Umjetni život je puno napredovao od svoga začetka u 1970-ima te se danas koristi za znatno šire primjene od teoretskog proučavanja, kao što je bila *Conwayjeva igra života*.

Iako se razvija već više od 40 godina, umjetni život je još uvijek novo i nedovoljno istraženo područje tako da i dalje postoje velike mogućnosti daljnjeg istraživanja i napretka, osobito u 3D simulacijama. Iskorištavajući sve moćnija i jeftinija računala tek je u posljednjih nekoliko godina razvoj umjetnog života uzeo maha.

U okviru rada postavljen je i opisan korišteni model umjetnog života koji se sastoji od biljaka, biljojeda i mesojeda te je programski izveden simulator umjetnog života u 3D prostoru koristeći slobodno dostupno okruženje otvorenog koda *Minetest* kao okosnicu. Sva su bića jednostavna, vodeći se isključivo nagonima i ne postoji nikakav oblik inteligencije niti pamćenja.

Simulator je izveden u programskom jeziku Lua kao proširenje za okruženje *Minetest*. Okruženje pruža već ostvarenu primjenu fizičkih zakonitosti, upravljanje korisničkim unosima kao i 3D grafički prikaz svijeta.

Statistika se bilježi u tabelarnom formatu prilagođenom prikazivanju pomoću programa *gnuplot*.

Ispitivanjem rada simulatora uočeni su obrasci koji se često pojavljuju: *Biljke prevladavaju* i *Biljke izumiru* što vodi do scenarija *Biljojedi izumiru prije* ili *Mesojedi izumiru prije*. U okviru obrasca *Biljke izumiru* primjećena je tendencija gdje se u tom scenariju veća količina energije akumulira paralelno s izumiranjem biljaka, a neposredno prije izumiranja bilo biljojeda, bilo mesojeda.

Uz identične početne postavke, moguće je očekivati u potpunosti različite ishode. Očitovanje je to složenosti života samog, kao i cijelog ekosustava koji ga okružuje. Baš kao u stvarnome svijetu i u simuliranome se primjećuje važnost utjecaja slučajnih odabirana na životnu (ne)predvidivost.

LITERATURA

- [1] Framsticks. URL <http://www.framsticks.com>.
- [2] Mathematical Games - The fantastic combinations of John Conway's new solitaire game "life" - M. Gardner - 1970. URL http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm.
- [3] Irrlicht engine. URL <http://irrlicht.sourceforge.net/>.
- [4] Lua. URL <https://www.lua.org/about.html>.
- [5] Minetest, . URL <https://www.minetest.net>.
- [6] Minetest api documentation, . URL <https://dev.minetest.net/Category:Methods>.
- [7] Opengl. URL <https://www.opengl.org/>.
- [8] Species: Artificial Life, Real Evolution. URL <http://speciesgame.com/>.
- [9] Conway's game of life - wikipedia, the free encyclopedia. URL https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life.
- [10] Z. Fras. Simulacija umjetnog života. Diplomski rad, Fakultet Elektrotehnike i Računarstva Sveučilišta u Zagrebu, 2014.

Simulator umjetnog života u 3D prostoru

Sažetak

U ovom je radu opisan koncept umjetnog života, neke od dosadašnjih izvedbi te je definiran novi model uveden za potrebe rada. Opisani su neki od simulatora umjetnog života kroz povijest (*Conwayeva igra života*, *Framsticks*, *Species ALRE*) i njihove mogućnosti.

Praktični rad sastoji se od izvedbe simulatora umjetnog života u 3D prostoru temeljenog na upisanom modelu te njegovog ispitivanja. Analizirani su rezultati ispitivanja i obrasci koji se najčešće pojavljuju. Prikazana je nepredvidivost sustava temeljenih na simulaciji života.

Ključne riječi: umjetni život, simulator, 3d prostor, simulacija, virtualno okruženje

3D Space Artificial Life Simulator

Abstract

This paper describes the virtual life concept, some of the existing implementations, and introduces the new model created for the paper itself. Usages of the historically used simulators (*Conway's game of life*, *Framsticks*, *Species ALRE*) and their features are described.

The practical part includes 3D space artificial life simulator based on the described model, and testing of the implementation. Testing results are then analyzed and the most common patterns are noted. The life simulating systems' unpredictability is demonstrated.

Keywords: artificial life, simulator, 3d space, simulation, virtual environment