

**Elektrotehniško-računalniška strokovna šola in gimnazija Ljubljana  
Ljubljana, Vegova 4**

Seminarska naloga pri 4. predmetu na poklicni maturi

## **Simulacija evolucije umetnega življenja**

**Mentorica:** Nataša Makarovič, prof.

**Avtor:** Gašper Medved, 4. A

*Ljubljana, april 2013*

# Povzetek

V seminarski nalogi je predstavljena implementacija in ugotovitve simulacije umetnega življenja. Implementacija je bila izvedena z pomočjo programskega jezika C++ in grafične knjižnice SDL. Za simuliranje so bile uporabljene nevronske mreže. Ugotovitve so predstavljene po simuliranju evolucije.

Ključne besede: umetno življenje, nevronska mreža, evolucija, simulacija

# Summary

This term paper presents implementation and findings of a simulation of artificial life. Implementation was made using the programming language C++ and graphical library SDL. For simulating, neural networks were used. Findings are presented after simulating evolution.

Keywords: artificial life, neural network, evolution, simulation

# Kazalo

Uvod .....	5
Umetno oživljenje .....	6
Možgani organizmov .....	7
Nevronske mreže .....	8
Okolje .....	9
Organizmi .....	10
Uporaba programa .....	12
Slika možganov .....	13
Implementacija .....	14
Ugotovitve .....	16
Zaključek .....	19

# Kazalo slik

Slika 1: Človeški genom .....	6
Slika 2: Analogija možganov .....	7
Slika 3: Nevron .....	8
Slika 4: Umetni nevron .....	8
Slika 5: Okolje in hrana .....	9
Slika 6: Organizem .....	10
Slika 7: Iztrebljanje .....	10
Slika 8: Hranjenje .....	11
Slika 9: Razmnoževanje .....	11
Slika 10: Program .....	12
Slika 11: Slika možganov .....	13
Slika 12: Del kode .....	15
Slika 13: Možgani naključnih organizmov .....	17
Slika 14: Možgani razvitih organizmov .....	17
Slika 15: Primerjava naključnih in razvitih organizmov .....	18

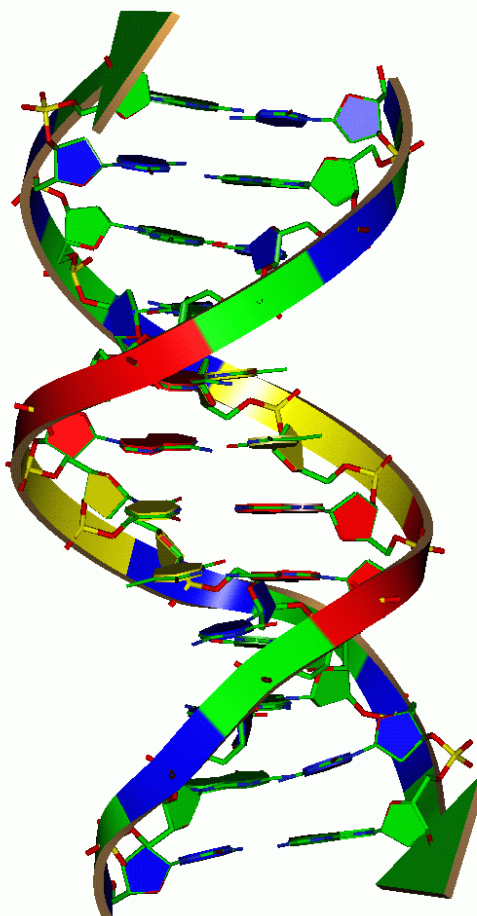
# Uvod

Vsak človek je živ in zavesten. So tudi živali zavestne? Kaj pa insekti? Taka vprašanja se ljudje sprašujejo že stoletja, znanost pa še vedno nima odgovorov. Z simuliranjem naravnih pojavov, so znanstveniki dosegli že veliko prebojev, in ravno z to metodo se bomo morda približali odgovoru na taka vprašanja. Večina modelov umetnega življenja je sestavljenih iz nekega okolja, ki vsebuje organizme. Vsak organizem ima računalniški model možganov. Ko se simulacija zažene, imajo vsi organizmi naključne možgane in gene. Nekateri so učinkoviti pri preživetju, nekateri pa takoj odmrjejo. Bolj učinkoviti organizmi ostanejo živi dlje časa, in imajo več možnosti za razmnoževanje. Pri razmnoževanju se z mutacijami spremenjeni možgani prenesejo na otroke. Nekatere mutacije so v prid organizmom, nekatere niso. Tisti organizmi, katerim je mutacija pomagala, imajo več možnosti za preživetje, in nato spet za razmnoževanje. Tako z evolucijo, po nekaj časa simuliranja, organizmi razvijejo inteligenco in boljšo sposobnost preživetja, kar bo ta seminarska naloga tudi prikazala.

# Umetno življenje

Umetnega življenja ne smemo zamešati z sintetičnim življenjem. Sintetično življenje, je fizična oblika življenja, ustvarjena s pomočjo biologije, umetno življenje, pa je simulirano življenje s pomočjo programske opreme in algoritmov.

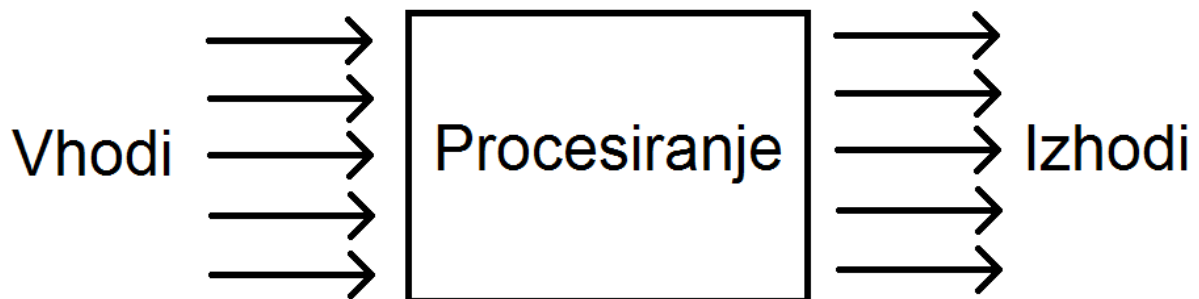
Vsak model umetnega življenja je sestavljen iz organizmov, ki živijo v nekem okolju. Okolje je definirano kot 2D ali 3D prostor, lahko pa je čisto nekaj abstraktnega. Eden izmed prvih modelov umetnega življenja sploh ni bil postavljen v prostor, ampak so organizmi tekmovali za procesorski čas in dostop do pomnilnika. Ti morajo imeti sposobnost medsebojne interakcije, in razmnoževanja v okolju. Vsak organizem ima parametre, ki določajo njegovo sestavo. Ko se organizem rodi, se njegovi parametri prenesejo iz staršev in spremenijo z mutacijo. Tem parametrom rečemo genomi. Tudi ljudje imajo genome. Ti so shranjeni v DNK, ki npr. določa barvo oči, las, kože, obliko obraza, itd. Organizmi z boljšimi genomi imajo več možnosti za preživetje in razmnoževanje. Torej, če vzamemo veliko število organizmov, bodo na koncu ostali tisti, ki zaradi svojih genomov najbolj učinkovito zagotavljajo svoje preživetje, in z tem možnost razmnoževanja. Temu procesu rečemo evolucija.



Slika 1: Človeški genom

# Možgani organizmov

Najbolj pomemben del vsakega živega bitja so njegovi možgani. Kako možgani sploh delujejo? Za kaj jih živa bitja uporabljajo? Možgane si lahko predstavljamo kakor nekakšen model odločanja. Vse odločitve, in dejanja izvirajo iz možganov. Toda na podlagi česa možgani sploh sprejemajo odločitve? Za vsako odločitvijo možgani potrebujejo nekakšne informacije o trenutnem stanju bitja, in o okolici v kateri se trenutno nahaja. Cilj možganov je zagotavljanje preživetja in razmnoževanja na osnovi teh informacij. Bolj učinkovito to možgani uresničujejo, bolj inteligentni so. Torej vse kar možgani počnejo je, da vzamejo vhodne podatke, jih procesirajo, in dajo izhodne podatke. Vhodni podatki so običajno trenutno stanje živega bitja, ter podatki o okolici, izhodni podatki pa njegove odločitve.

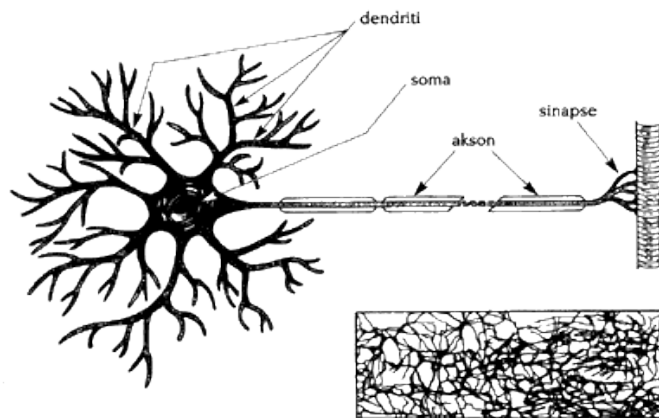


Slika 2: Analogija možganov

Nekatere simulacije umetnega življenja to implementirajo z izvršljivimi ukazi, novejša simulacije pa z nevronskimi mrežami. Struktura možganov so tudi del genoma organizma. Prenese se od staršev, malce spremeni, in z pomočjo evolucije skozi čas izboljša, kar pomeni, da organizmi skozi simulacijo postanejo bolj inteligentni.

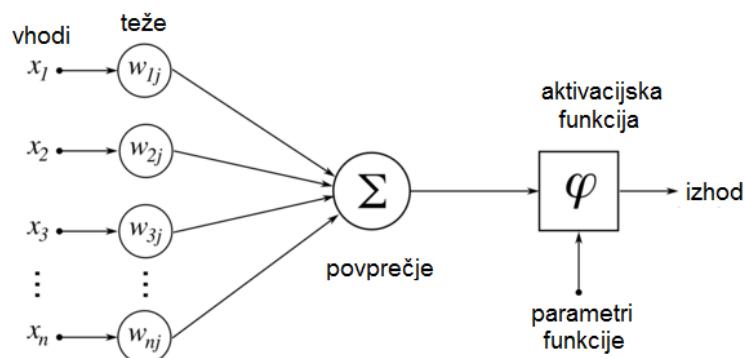
# Nevronske mreže

V našem primeru smo se za odločitveni model organizmov odločili uporabo nevronske mreže. Za razumevanje nevronske mreže na začetku pogledjmo kako so zgrajeni pravi možgani. Gradnik vseh možganov je biološka celica, ki ji pravimo nevron. V človeških možganih, ki so zelo kompleksni, jih je okoli 86 bilijonov. Vsak je povezan z tisočimi drugimi, in ta mreža nevronov omogoča kompleksno vedenje. Če njegovo delovanje poenostavimo, nevron vzame vhodne povezave, in če je njihovo povprečje večje kot neka določena vrednost, se nevron aktivira ter pošlje vrednost naprej. Nekaj podobnega imamo tudi pri nevronske mrežah.



Slika 3: Nevron

Vsaka nevronska mreža je sestavljena iz umetnih nevronov ali vozlišč, in povezav med njimi. Nevroni so lahko vhodni, skriti ali izhodni. Na vhodne nevrone običajno podajamo podatke, na izhodnih pa jih beremo. Skriti nevrone so pa zadolženi za procesiranje teh podatkov. Vsaka povezava med vozlišči ima svojo težo, ki pove koliko vpliva na vozlišče v katerega vodi. Vozlišče torej naredi dve stvari. Na začetku sešteje vse vrednosti vhodov, da dobi njihovo povprečje. Pri tem upošteva teže povezav. Ko dobi povprečje, ga pošlje funkciji za aktivacijo, ki na izhod vozlišča vrne določeno vrednost. To vrednost pa dobi spet nova povezava.

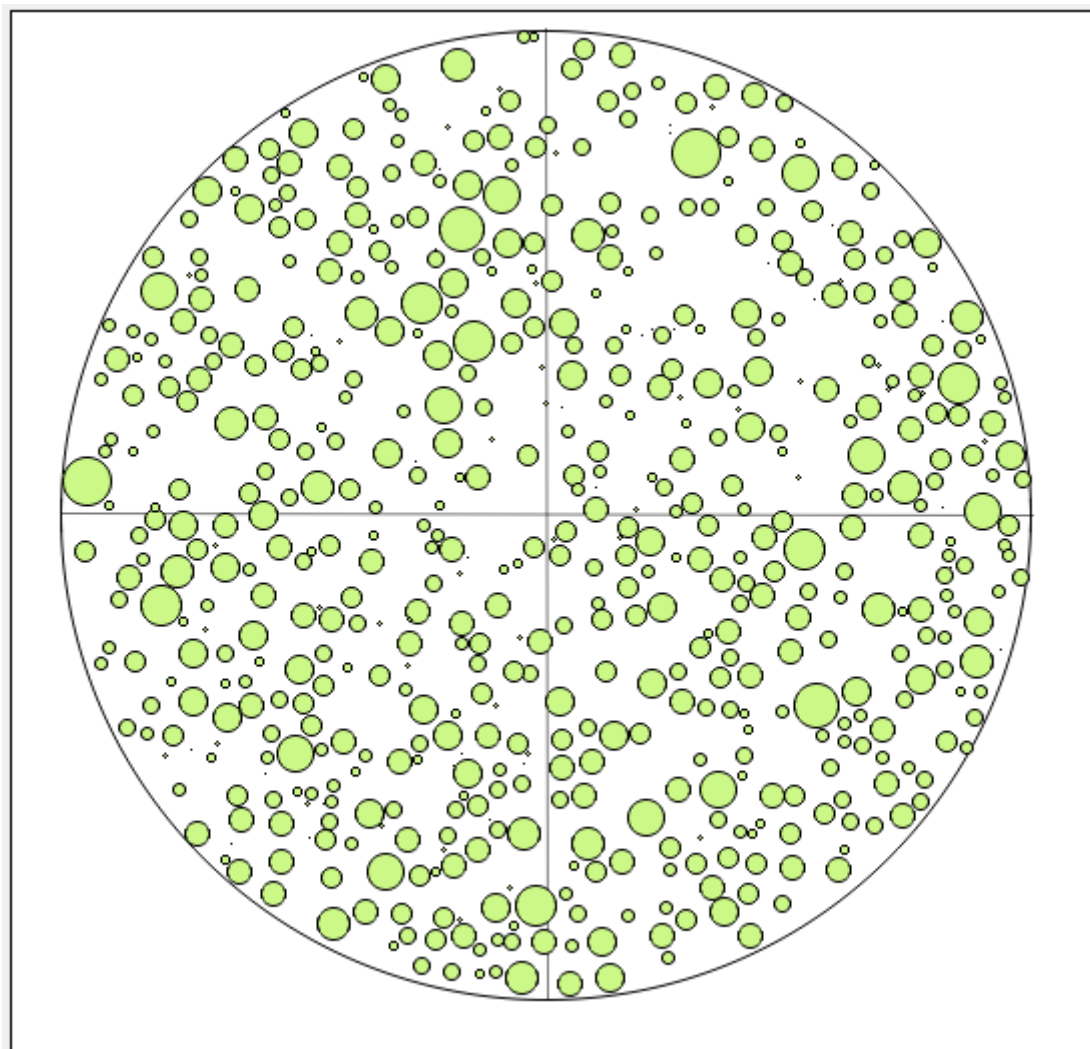


Slika 4: Umetni nevron



# Okolje

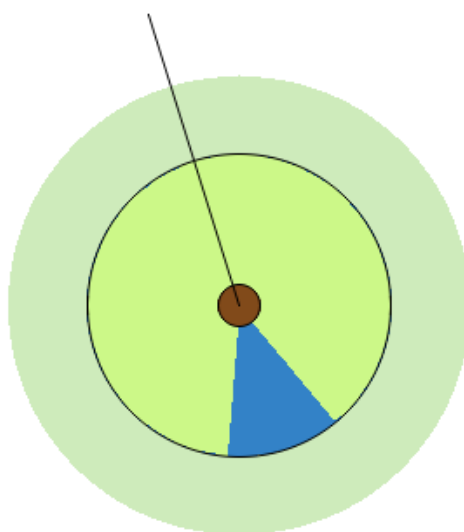
Ko hočemo zasnovati umetno življenje, si moramo na začetku zamisliti neko okolje. V našem primeru smo uporabili dvodimenzionalni okrogel omejen prostor. Velikost prostora lahko kadarkoli z parametri simulacije spremenimo. V tem prostoru se giblje trdna okrogla hrana, ki se ob trku odbije ena od druge. Trk je implementiran z formulo neprožnega trka, čigar prožnost se tudi lahko nastavi z parametri simulacije. Energija, ki jo vsaka hrana vsebuje je definirana kot površina, ki jo pokriva. V sistemu nikoli ne smemo dodajati ali vzemati energije, saj z pomočjo evolucije organizmi skoraj vedno najdejo način kako to izkoristiti v svoj prid. Če naredimo napako pri programiranju sistema in prenašanja energije, bodo organizmi to napako po nekaj časa simuliranja z pomočjo evolucije odkrili, in jo izkoristili, kar se je tudi v zgodnjih fazah razvoja programa tudi dogajalo.



Slika 5: Okolje in hrana

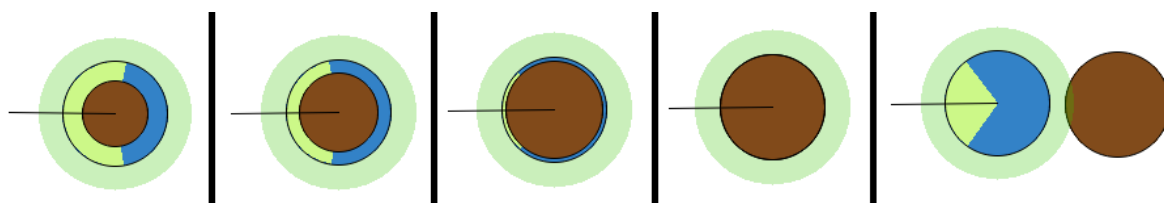
# Organizmi

Ko imamo okolje definirano, se moramo lotiti organizmov. V našem primeru so organizmi tudi okrogli. Črta iz njihove sredine prikazuje smer v katero so obrnjeni, njihovo telo pa je modro-zelene barve. Količina zelene barve prikazuje koliko energije imajo. Če je celo telo zelene barve, ne potrebujejo nič energije, če je skoraj celo telo modro, pa pomeni da nimajo energije, in če je ne dobijo kmalu, bodo umrli. Vsak organizem pa ima tudi ovojnico, ki pa ni trda. Ovojnica prikazuje njegovo razpoloženje. Rdeča pomeni napadalnost, zelena pa pripravljenost na razmnoževanje.



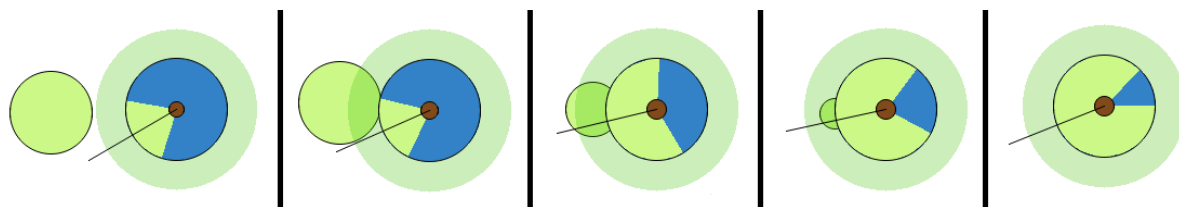
Slika 6: Organizem

Rjav krog v sredini je iztrebek. Vsa porabljena energija gre v večanje tega iztrebka. Ko doseže velikost organizma, ga izvrže. Po določenem času se iztrebek v okolju spremeni nazaj v hrano.



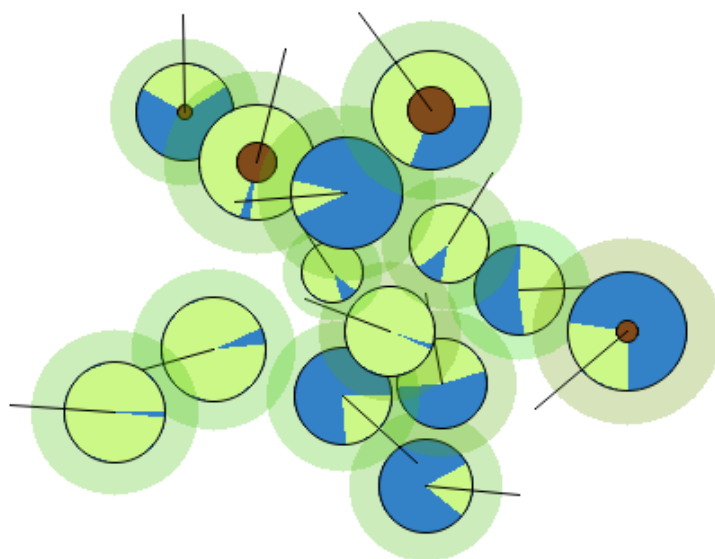
Slika 7: Iztrebljanje

Vse kar organizem počne vzame energijo. Ko energijo izgubi, umre. Zato je njihov cilj dobiti čimveč hrane in z tem energije. Hranjenje poteka tako, da se hrani približa in se je dotakne. Ko njihovo telo prekriva del hrane, se ta del vsrka v organizem. Ko se njihovo telo napolni, tj. če je celo telo zeleno, hrane ne morejo vsrkavati več.



Slika 8: Hranjenje

Razmnoževanje poteka tako, da se dva organizma, ki imata zeleno ovojnico in dovolj energije približata in dotakneta. Pri tem se ustvari eden ali več otrok, ki podedujejo genome iz obeh staršev. Ta proces seveda tudi vzame energijo, saj jo otroci potrebujejo za začetek svojega življenja. Energije nikoli ne damo ali vzamemo iz sistema.

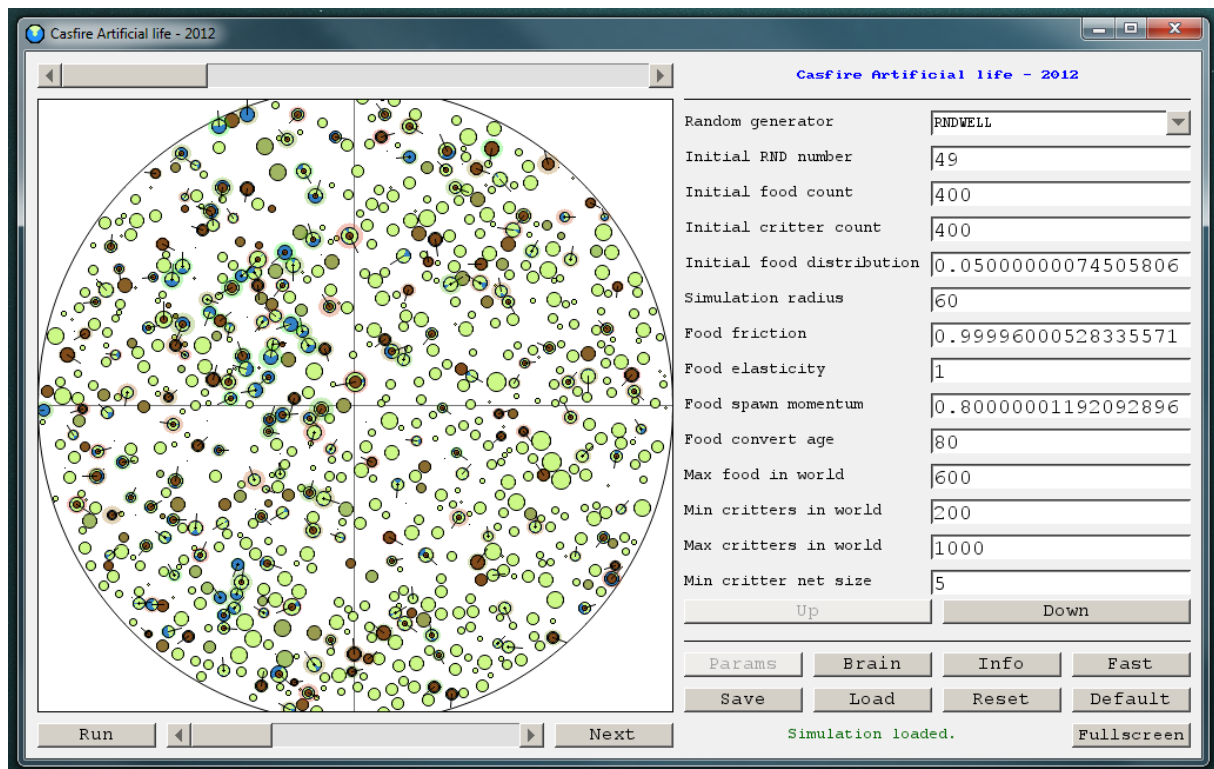


Slika 9: Razmnoževanje

Ko se rodijo, so neskončno majhni, nato pa zrastejo, in se povečajo v naravno velikost. Naši organizmi imajo 4 genome. Hitrost, ki pove kako hitro se lahko premikajo po okolju, radij, ki pove do koliko lahko zrastejo, hitrost rasti, ki pove kako hitro zrastejo, in seveda sama nevronska mreža. Večja hitrost pomeni, da mu premikanje vzame več energije, zato da genoma hitrosti ne morejo izkoriščati. Podobno je z njihovo velikostjo. Večji so, več energije jim vzame premikanje.

# Uporaba programa

Program mora imeti v isti mapi nekaj stvari, da se lahko zažene. V Windows okolju je odvisen od datotek *libfreetype-6.dll*, *SDL.dll*, *SDL\_ttf.dll* in *zlib1.dll*. Te datoteke omogočajo inicializacijo SDL knjižnic. Poleg tega mora program imeti datoteko *font.ttf* v kateri je definirana pisava za uporabniški vmesnik, in mapo *save* v katero se shranjuje trenutno stanje simulacije.



Slika 10: Program

Ko se program naloži, pogleda v mapo save, in če tam ni shranjene simulacije, ustvari novo privzeto. Na levi strani je grafični prikaz simulacije. Drsnik zgoraj nadzira rotacijo, drsnik spodaj pa njeno hitrost. Gumb Run požene in ustavi simuliranje, gumb Next pa jo premakne za eno sličico naprej. Na desni strani pa je vmesnik za nadzor in informacije o simulaciji.

*Fullscreen* - Prikaže samo grafični prikaz simulacije.

*Save* - Shrani simulacijo na disk.

*Load* - Naloži simulacijo iz diska.

*Reset* - Resetira simulacijo na začetek.

*Default* - Parametri simulacije na privzete vrednosti.

*Params* - Odpre parametne simulacije.

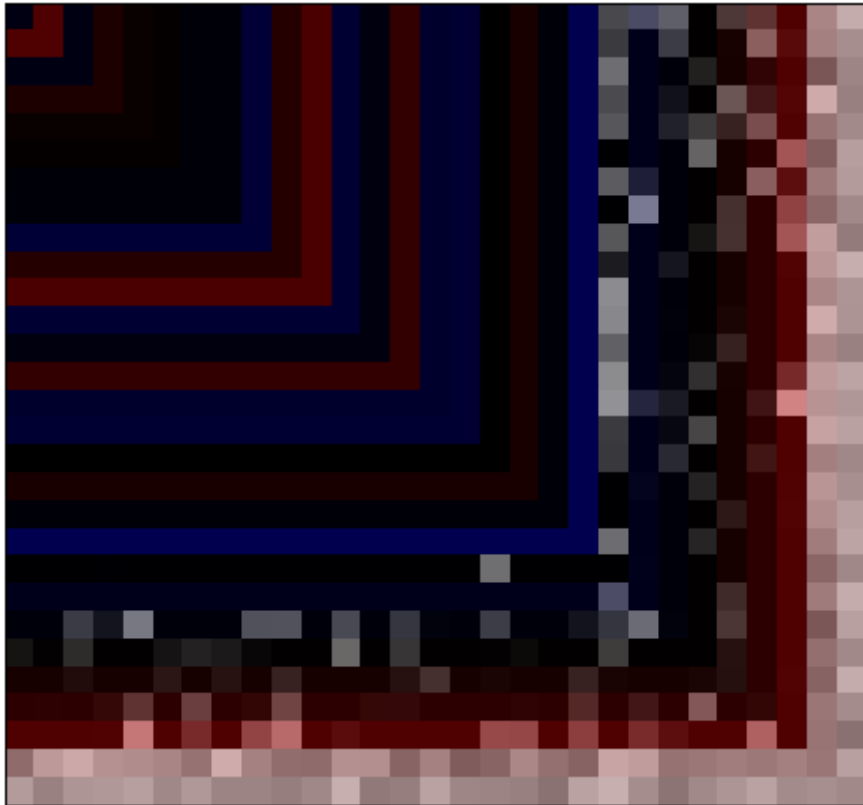
*Brain* - Prikaže možgane in informacije o izbranem organizmu.

*Info* - Prikaže informacije o simulaciji.

*Fast* - Zmanjša okno in generira brez risanja.

# Slika možganov

Ker so vozlišča v naši implementaciji nevronske mreže povezana vsak z vsakim, lahko teže povezav prikažemo kot mrežo kvadratkov, čigar svetlost prikazuje težo posamezne povezave. Parametri aktivacijske funkcije niso prikazani. Na sliki aktivnost možganov vidimo z čez mrežo kvadratkov narisanimi modrimi in rdečimi črtami, ki prikazujejo trenutno vrednost vozlišč. Rdeča je negativna vrednost, modra pa pozitivna.



Slika 11: Slika možganov

Taka slika nam omogoča opazovanje razvoja možganov skozi čas. Na začetku simulacije so možgani naključni, brez vzorcev, ko pa simuliramo nekaj časa, se začnejo pojavljati vzorci.

# Implementacija

Izvorna koda je napisana v jeziku C++. Za grafični prikaz organizmov je bila izbrana knjižnjica SDL, saj ima podporo na veliko platformah. Sama knjižnjica pa ni dovolj, saj ne podpira risanja enostavnih oblik, kot so krogi ali črte. V ta namen je bil uporabljen malce spremenjen dodatek iz interneta `SDL_gfx`, ki je vključen v izvorno kodo. Celotna izvorna koda je modularna, kar pomeni, da je razdeljena na več delov, ki se lahko uporabijo v drugih projektih.

`SDLHelper.hpp`   `SDLHelper.cpp`

Prvi napisan modul je pripomoček za SDL, ki pomaga pri programiranju v tej knjižnjici. Definira funkcije za inicializacijo, ustvarjanje okna, novih površin za risanje, pretvorbo tipov spremenljivk, itd. Vsebuje pa tudi razreda za sledenje SDL dogodkom in časovnik.

`SDLGui.hpp`   `SDLGui.cpp`   `SDLGuiStyle.cpp`

Ta modul skrbi za risanje uporabniškega vmesnika, kot npr. gumbi, vnosne vrstice, tekst, itd. Uporablja `SDL_ttf` za nalaganje ttf datotek. Ravna z SDL dogodki, in skbi za notranje stanje vseh elementov uporabniškega vmesnika. Je najbolj obširni napisan modul.

`RandomGenerator.hpp`   `RandomGenerator.cpp`

Ta modul je bil napisan za generiranje naključnih števil, ki so ključna pri simulaciji, saj hočemo, da so mutacije naključne. Na različnih platformah se stdlib `rand()` funkcija različno obnaša, zato je bilo treba uporabiti druge algoritme. Implementirani algoritmi so `STDLIB`, `LCG`, `MWC`, `WELL512` in `MT19937`.

`MovableView.hpp`   `MovableView.cpp`

V namen, da lahko simulacijo premikamo, rotiramo, približujemo in oddaljujemo, je bil napisan modul ravno za to. Preko tega modula se dogaja vso risanje simulacije.

`NeuralNet.hpp`   `NeuralNet.cpp`

Implementacija nevronske mreže je najbolj pomemben modul. Ima svoj razred za vozlišče in razred za mrežo, ki vsebuje vozlišča. Za nize je bil uporabljen `std::vector`. Ima metode za nastavljanje in dobivanje vrednosti posameznih vozlišč, tež povezav, in parametre aktivacijskih funkcij vozlišč.

`critter.hpp`   `critter.cpp`

Implementacija razreda za organizme, njihovo obnašanje v okolju, interaktiranje, in vse ostalo povezano z organizmi.

`food.hpp`   `food.cpp`

Implementacija razreda za hrano, formula za trčenje, prenos energije.

params.hpp params.cpp

Implementacija uporabniškega vmesnika za parametre simulacije.

simulation.hpp simulation.cpp simulationParams.hpp

Implementacija razreda za simulacijo. Razred vsebuje nize hrane in organizmov, metode za shranjevanje in nalaganje iz diska, in ostalo povezano z simulacijo.

main.cpp

Implementacija uporabniškega vmesnika za simulacijo, in main funkcija.

```
class Critter{
//genomes
uint32_t gen_netSize;
NeuralNet net;
float gen_radius;
float gen_growth;
float gen_speed;

//outputs
float out_move;
float out_spin;
float out_mood;
float out_expose;

//inputs
float in_random;
float in_size;
float in_energy;
float in_waste;
float in_age;
float in_foodNearAng;
float in_foodNearDis;
float in_foodAng;
float in_critNearAng;
float in_critNearDis;
float in_critAng;
float in_sensorU, in_sensorUR, in_sensorR, in_sensorDR, in_sensorD, in_sensorDL, in_sensorL, in_sensorUL;

//physical
float x, y, rot, rad, mrad, mradPow;
float energy, waste;
float maxEnergy, maxGenEnergy;
float processedEnergy;

//temp
float tmp_foodX, tmp_foodY, tmp_foodClosestDistance; Food *tmp_foodClosest;
float tmp_critX, tmp_critY, tmp_critClosestDistance; Critter *tmp_critClosest;
float tmp_exp;

//sensor positions
float tmp_sensorUx, tmp_sensorUy, tmp_sensorURx, tmp_sensorURy;
float tmp_sensorRx, tmp_sensorRy, tmp_sensorDRx, tmp_sensorDRy;
float tmp_sensorDx, tmp_sensorDy, tmp_sensorDLx, tmp_sensorDLy;
float tmp_sensorLx, tmp_sensorLy, tmp_sensorULx, tmp_sensorULy;
```

Slika 12: Del kode

# Ugotovitve

Ko imamo program napisan, je čas da simulacijo zaženemo, pustimo nekaj časa, in analiziramo vedenje organizmov.

Na začetku, so vsi organizmi, in njihovi genomi naključni, kar pomeni, da je tudi njihovo obnašanje nepredvidljivo in naključno. Večina se jih vrti v krogih, premikajo se nazaj, naprej, so veliki in majhni, ovojnice vidimo rdeče in zelene. Pri preživetju niso učinkoviti, zato je na začetku hrane dovolj, da jih preživi vsaj malo. Čeprav je hrane dovolj, jih večina umre in število organizmov pade do minimuma.

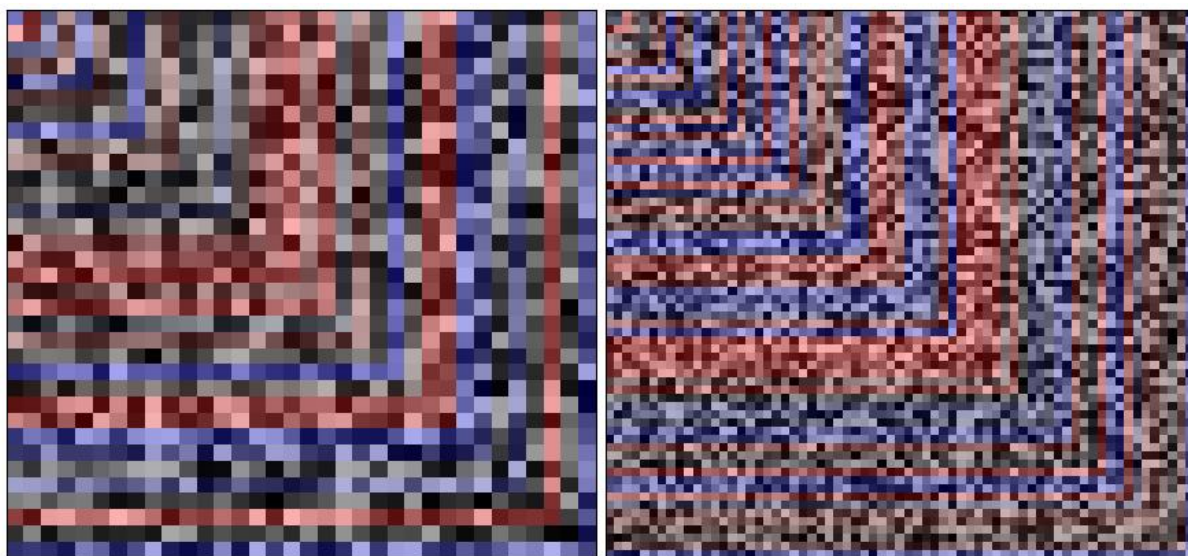
Zanimivo je, ko pustimo simulacijo teči nekaj časa, vsi napadalni organizmi odmrjejo, in ostanejo samo tisti z zeleno ovojnico. Čeprav napadalni dobijo več hrane in zato preživijo dlje časa, se taki ne razmnožujejo, in zato odmrjejo.

Če jih pustimo še dlje, se začnejo opazovati prvi znaki inteligence. Čim se to zgodi, se njihova populacija dvigne do maksimuma. Bolj so inteligentni, več jih bo v simulaciji. Zaradi tega, moramo pogoje življenja postopoma zviševati. V našem primeru smo to storili z povečanjem prostora. Z večjim prostorom se hrana bolj razporedi, in jo je težje najti.

S časoma se organizmi razvijejo do te točke, da ko potrebujejo hrano, se obrnejo do najbližje in jo pojejo. Ko pa hrane ne potrebujejo, pa iščejo partnerja. Vsi organizmi se razvijejo majhni, saj tako porabljajo manj hrane. Ustavijo pa se nikoli. Vozliščev v možganih tudi izgubijo precej, saj več vozlišč pomeni večja poraba energije.

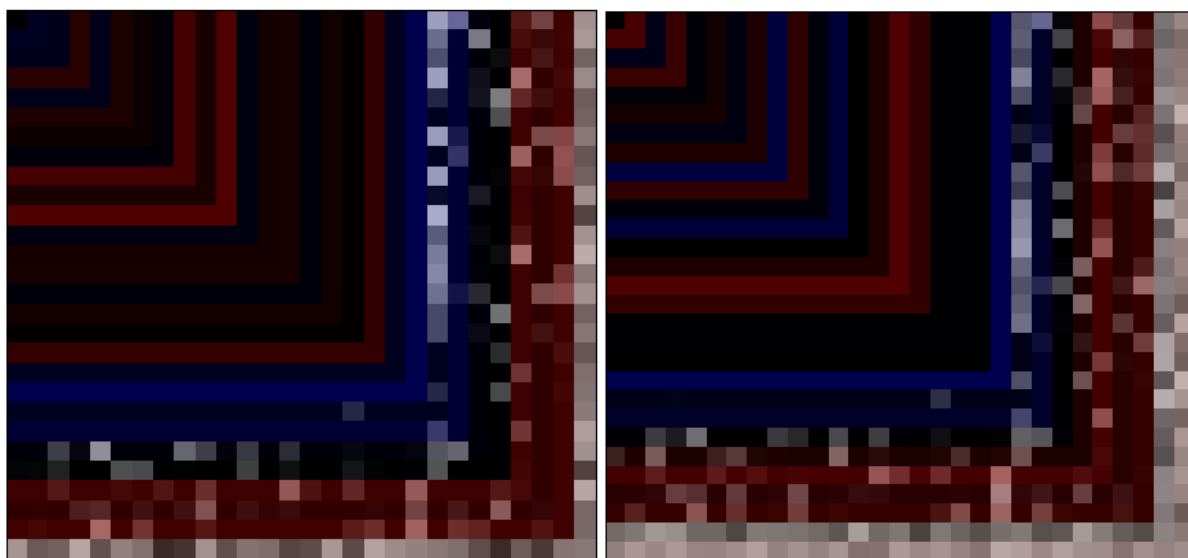


Primer možganov naključnih organizmov.



Slika 13: Možgani naključnih organizmov

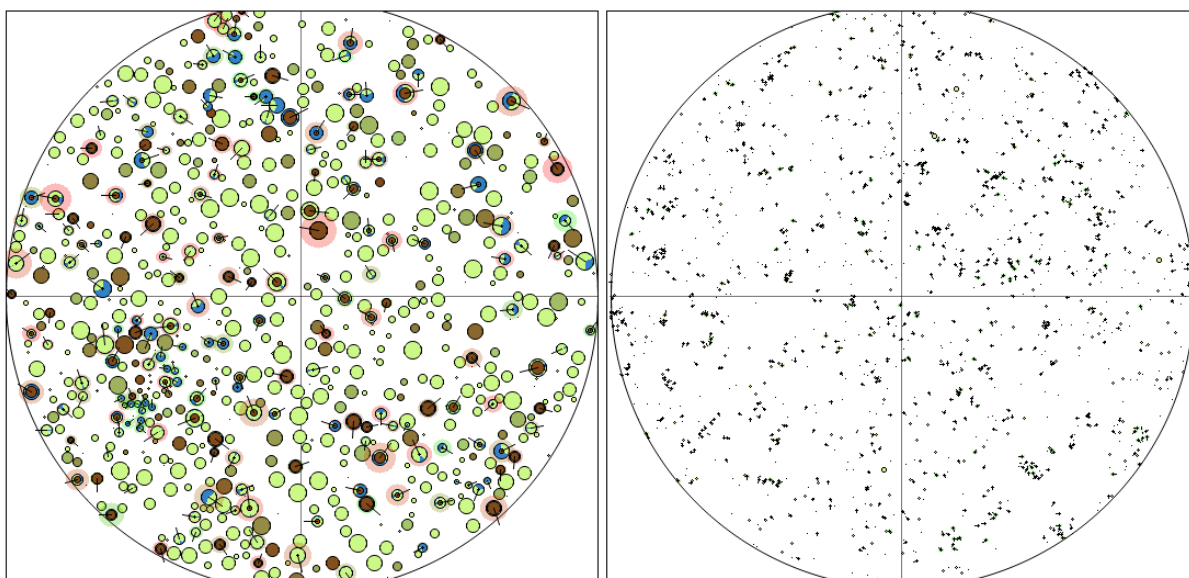
Primer možganov razvitih organizmov.



Slika 14: Možgani razvitih organizmov

Iz teh slik vidimo, da se tudi struktura možganov razvije. V naključnih možganih ni vzorcev, aktivnost je nepredvidljiva, v razvitih pa vidimo vzorce in celo podobno aktivnost med različnimi organizmi.

V naslednji sliki vidimo na levi strani naključne organizme, na desni pa razvite.



Slika 15: Primerjava naključnih in razvitih organizmov

Na desni strani so organizmi manjši, saj so bolj sposobni iskanja hrane in partnerjev za razmnoževanje.

Iz tega sledi, da lahko z simulacijo evolucije umetnega življenja neposredno opazujemo razvoj organizmov. Z bolj kompleksnim okoljem, modelom možganov, večceličnimi organizmi, bomo v prihodnosti z takšnimi simulacijami mogoče lahko opazovali tudi kompleksne socialne interakcije, več vrst organizmov, in višje oblike inteligence.

Možnosti je neskončno.

# Zaključek

Ali so umetni organizmi lahko živi?

Življenje lahko definiramo na več različnih načinov. Najbolj sprejeta definicija je sestavljena iz sedmih točk. Če določen organizem zadostuje vsem naslednjim sedmim točkam, lahko rečemo da je živ.

- Homeostaza (regulacija notranjosti za ohranjanje konstantnega stanja)
- Organizacija (biti strukturno sestavljen iz ene ali več celic, ki so osnovni gradniki življenja)
- Metabolizem (preoblikovanje energije v celične komponente)
- Rast (večanje organizma v velikosti in vseh svojih delih)
- Prilagajanje (sposobnost, da se v določenem času spremenijo kot odziv na okolje)
- Odziv na dražljaje
- Reprodukcijska (sposobnost ustvarjanja novih organizmov)

Zanimivo pa je, da nobena od teh točk ne zahteva, da mora živo bitje nujno obstajati v realnem svetu. Torej, če umetni organizem zadostuje vsem navedenim točkam, lahko trdimo da je živ. Večini ljudi ta ideja ni tako privlačna, saj si živa bitja predstavljajo kot nekaj oprejemljivega in stvarnega. Zamislimo si človeka, ki je sestavljen iz celic. Za človeka lahko trdimo da je živ, in se zaveda samega sebe. Vsaka celica v človeku je sestavljena iz atomov. Fizikalno obnašanje atomov lahko predstavimo z matematičnimi enačbami, ki jih računalnik lahko izračunava. Torej lahko v računalniku simuliramo resničen svet, in ugotovimo kako bi se atomi obnašali v realnosti. Vzemimo torej vse atome iz katerih je sestavljen človek, in jih virtualno postavimo v računalnik. Na računalniku zaženemo simulirano realnost in gledamo človeka. Predpostavljamo, da človeška zavest izvira iz sinaptičnih povezav v možganih, ki so tudi sestavljeni iz atomov. Ali bi bil človek v taki simulaciji tudi živ in zaveden? Na taka vprašanja je težko odgovoriti, toda z raziskovanjem evolucije in življenja smo vsak dan bližje odgovoru.

## Literatura in viri

- <http://www.libsdl.org/>
- [http://www.ferzkopp.net/Software/SDL\\_gfx-2.0/](http://www.ferzkopp.net/Software/SDL_gfx-2.0/)
- [http://www.lazyfoo.net/SDL\\_tutorials/](http://www.lazyfoo.net/SDL_tutorials/)
- [http://en.wikipedia.org/wiki/List\\_of\\_pseudorandom\\_number\\_generators](http://en.wikipedia.org/wiki/List_of_pseudorandom_number_generators)
- <http://www.ai-junkie.com/ann/evolved/nnt1.html>
- [http://en.wikipedia.org/wiki/Artificial\\_life](http://en.wikipedia.org/wiki/Artificial_life)
- [http://www.youtube.com/watch?v=m97\\_kL4ox0](http://www.youtube.com/watch?v=m97_kL4ox0)