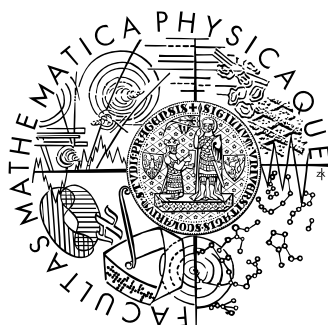


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Vladimír Kudelas

Evoluce umělého života

Kabinet software a výuky informatiky

Vedúci bakalárskej práce: RNDr. Tomáš Holan, Ph.D.
Študijný program: Informatika, programování

2009

Ďakujem pánovi RNDr. Tomášovi Holanovi, Ph.D. za odborné vedenie mojej bakalárskej práce, za ochotu a čas, ktorý mi pri vypracovávaní tejto práce venoval.

Prehlasujem, že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce.

V Prahe dňa

Vladimír Kudelas

Obsah

1 Úvod.....	6
1.1 Čo je Umelý život?	6
1.2 Načo evolúcia umelého života?	6
1.3 SofE – Simulátor evolúcie umelého života.....	7
2 Podobné programy	8
2.1 Avida.....	8
2.2 Gene Pool.....	9
2.3 Broucci a Abeetles	9
2.4 Záver	10
3 Umelý život v SofE.....	11
3.1 Svet simulácie	11
3.2 Popis parametrov organizmov	11
3.2.1 Potrava organizmov	11
3.2.2 Vek.....	11
3.2.3 Cena pohybov	11
3.2.4 Energia organizmov	12
3.2.5 Násobok pre spotrebu energie z mäsa.....	12
3.2.6 Limit hladu.....	12
3.2.7 Funkcia Energia vs. Vek	12
3.2.8 Schopnosť zabíjať	13
3.2.9 Parametre pre párenie	13
3.2.10 Algoritmus organizmov	14
3.3 Druhy organizmov	15
4 Popis implementácie	16
4.1 Riešenie grafického užívateľského rozhrania	16
4.1.1 Podpora viacerých jazykov	16
4.1.2 Rozvrhnutie dialógu pre novú simuláciu	16
4.1.3 Rozvrhnutie okna pre simuláciu	17
4.1.4 Uloženie a načítanie simulácie/-í.....	18
4.1.5 Vrátenie simulácie späť	18
4.1.6 Nápoveda	19
4.2 Problémy pri vytváraní samotnej simulácie.....	19
4.2.1 Komponenty simulácie	19
4.2.2 Algoritmus organizmov	20
4.2.3 Vlákna	20
4.2.4 Fond organizmov	21
4.2.5 Spolupráca organizmov	21
4.2.6 Výpočet štatistických údajov	22
5 Experimenty	23
5.1 Prvý experiment – vplyv pravdepodobnosti mutácie na vývoj populácie	23

5.1.1 Počiatočné nastavenia	23
5.1.2 Výsledky prvého experimentu	24
5.1.3 Záver	27
5.2 Druhý experiment – vplyv funkcie využitia energie vzhľadom k veku na vývoj populácie	27
5.2.1 Počiatočné nastavenia	28
5.2.2 Výsledky druhého experimentu	29
5.2.3 Záver	30
5.3 Tretí experiment - vývoj parametru určujúceho množstvo požadovanej energie od partnera a parametru určujúceho množstvo energie predanej partnerovi.	31
5.3.1 Počiatočné nastavenia	31
5.3.2 Výsledky tretieho experimentu	32
5.3.3 Pod experiment – klesajúca pravdepodobnosť potravy	33
5.3.4 Výsledky pod experimentu	34
5.3.5 Záver	34
5.4 Štvrtý experiment – analýza a vývoj algoritmu pre jeden stav organizmu.....	34
5.4.1 Počiatočné nastavenia	34
5.4.2 Analýza algoritmu pre jeden stav organizmu	35
5.4.3 Výsledky štvrtého experimentu	37
6 Záver	39
6.1 Budúcnosť programu SofE	39
Literatúra.....	40
Prílohy.....	41
Príloha A - Uživatelská dokumentácia	41

Názov práce: Evoluce umělého života

Autor: Vladimír Kudelas

Katedra (ústav): Katedra software a výuky informatiky

Vedúci bakalárskej práce: RNDr. Tomáš Holan, Ph.D.

e-mail vedúceho: Tomas.Holan@mff.cuni.cz

Abstrakt:

Predmetom práce je implementácia simulátoru evolúcie umelého života a uskutočnenie experimentov s týmto simulátorom. V simulácii sa nachádzajú jednoduché organizmy, ktoré sa vyvíjajú na základe prirodzeného výberu. Genóm týchto organizmov je tvorený zoznamom parametrov. V simulácii sa rozlišujú druhy organizmov a druhy potravy. Vytvorený program umožňuje spúšťanie simulácii pomocou skriptu a ukládanie štatistických údajov.

Kľúčové slová: umelý život, evolúcia, simulátor

Title: Evolution of artificial life

Author: Vladimír Kudelas

Department: Department of Software and Computer Science Education

Supervisor: RNDr. Tomáš Holan, Ph.D.

Supervisor's e-mail address: Tomas.Holan@mff.cuni.cz

Abstract:

The task of the work is to implement a simulator of evolution of artificial life and make experiments with this application. The evolution is based on simple organisms which are evolved by natural choice. Genome of organisms consists of set of some parameters. Simulator distinguishes species of organisms and kinds of food. Application could be run by a script and it provides saving statistics information.

Keywords: artificial life, evolution, simulator

1 Úvod

1.1 Čo je Umelý život?

Umelý život je rýchlo sa rozvíjajúca oblasť vedeckého výskumu, ktorá spája biológiu a informatiku. Študuje „reálny“ život, tým, že sa pokúša znovu vytvoriť biologický fenomén od začiatku, pomocou počítačov a iných „umelých“ médií. Umelý život dopĺňa tradičný analytický prístup biológie o syntetický prístup, v ktorom namiesto toho, aby študoval biologický fenomén rozoberaním žijúcich organizmov, a zisťovaním ako fungujú, tak sa pokúša vytvoriť systém, ktorý sa správa ako žijúce organizmy.[1]

Termín „Umelý život“ bol prvýkrát spomenutý Christopherom Langtonom v práci „International Conference on the Synthesis and Simulation of Living Systems“.

Rozlišujeme tri hlavné druhy Umelého života. „Soft“ vytvára simulácie alebo iné čisto digitálne konštrukcie, ktoré znázorňujú správanie podobné tomu, ktoré vidíme v živej prírode. „Hard“ využíva k tomuto účelu hardware. „Wet“ používa biochémiu.[2]

V tejto práci sa venujeme „Soft“ Umelému životu.

Zásadnou vlastnosťou Umelého života je jeho syntetický prístup, ktorý pristupuje k problému z druhého konca. Tradičný analytický prístup sa snaží rozdeliť komplexný problém (systém) na menšie (základné) časti, s ktorými sa ľahšie pracuje. Syntetický prístup sa pokúša vytvoriť komplexný systém (riešenie) zo základných jednotiek.

Tento prístup nachádza využitie v rôznych odvetviach, a pri riešení rôznych problémov. V chemickom priemysle sa využíva pre vytvorenie nových materiálov, ktoré nachádzajú veľké praktické využitie. V informatike je to riešenie zložitých problémov. Napríklad algoritmus simulovaného žihania, ktorý je možné použiť k riešeniu problému obchodného cestujúceho, alebo metóda zakázaného prehľadávania (Tabu Search), ktorú je možné využiť pre riešenie logistických problémov v ekonomike, alebo na rozvrhnutie práce pre stroje v továrňach.[1]

1.2 Načo evolúcia umelého života?

To, ako vznikli žijúce tvory je jeden z najzložitejších a najdlhšie skúmaných problémov nášho sveta. Je to kvôli tomu, že telá živočíchov sú natoľko zložené, že ich naša myseľ nedokáže pochopiť ako celok.

Príkladom zložitosti môže byť napríklad oko človeka alebo „sonar“ netopierov. Niečo podobné „echolokátoru“ netopierov zostrojil aj človek počas druhej svetovej vojny. Bol to sonar, ktorý slúžil na lokalizovanie ponoriek. Netopiere dotiahli tento systém k dokonalosti pred desiatkami miliónov rokov.

Ďalším príkladom je napríklad obyčajná molekula hemoglobínu (červené krvné farbivo). Skladá sa zo štyroch aminokyselinových reťazcov. Každý reťazec sa skladá zo 146 aminokyselín. V živých organizmoch nachádzame 20 rôznych druhov aminokyselín. Pokiaľ by sme chceli vytvoriť jednu molekulu hemoglobínu zložením jednotlivých aminokyselín, tak máme približne 10^{190} možností ako pospájať jednotlivé aminokyseliny.[3]

Telá organizmov sú nepredstaviteľne zložené, a za ich vytvorenie je zodpovedná evolúcia.

Organizmy majú celú svoju genetickú informáciu (génom) uloženú v DNA. Počas evolúcie dochádza ku kríženiu organizmov a k malému vplyvu náhody, ktorý sa nazýva mutácia. Náhodne dôjde k mutácii nejakej časti génomu. Takýmto spôsobom sa vyvíjali organizmy počas niekoľko miliárd rokov. Vývoj organizmov sa nezastavil, bude pokračovať s evolúciou ďalej. Pre nás je zaujímavé ako tento vývoj prebiehal. Čo má vplyv na evolúciu? Počas reálnej evolúcie dokázali prežiť len tí, ktorí sa najlepšie prispôsobili prostrediu (dokázali prežiť), a dokázali sa reprodukovať, či už sexuálne alebo asexuálne. Tieto dve základné kritéria pre vývoj nejakej populácie, dokážu vyselektovať „najlepších“ jedincov. Tieto naše poznatky vyplývajú z Darwinovej teórie.

Úlohou simulácie evolúcie umelého života je pomôcť pri skúmaní toho, ako prebiehala evolúcia organizmov. Analytickým prístupom to nedokážeme zistiť, pretože dekomponovanie mechanizmu organizmov je ohromne zložitý a náročný proces. Pomocou umelého života, ktorý môže simulovať chovanie a génom primitívnych organizmov, môžeme sledovať ako sa približne „učili“ a vyvíjali jednoduché organizmy pred stovkami miliárd rokov. So stále sa zlepšujúcou výpočtovou technikou a programami sa môžeme pokúšať simulovať vývoj zložitejších organizmov.

Takýmto spôsobom môžeme pomôcť evolučnej biológii pri študovaní evolúcie jednotlivých organizmov.

1.3 SofE – Simulátor evolúcie umelého života

Cieľom tejto bakalárskej práce je vytvoriť program SofE, ktorý bude simulovať evolúciu jednoduchého umelého života. Organizmy, žijú vo svojom prostredí, kde dochádza k ich sexuálnemu rozmnožovaniu. Tieto organizmy majú definovaný určitý génom. SofE sa dá využiť k experimentom s evolúciou organizmov, pričom umožňuje rôzne ovplyvňovať túto evolúciu, a vypisovať o nej štatistické údaje. Súčasťou tejto bakalárskej práce sú aj rôzne experimenty uskutočnené s týmto programom, ktoré sú uvedené v kapitole Experimenty.

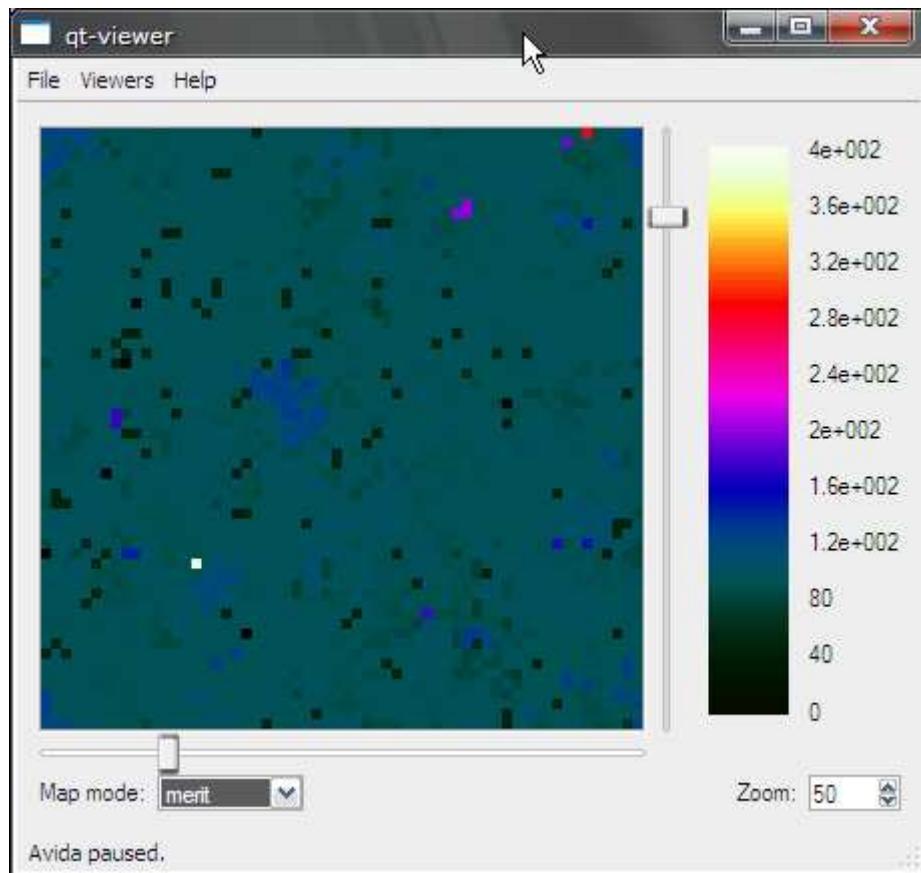
2 Podobné programy

Na internete môžeme nájsť veľa programov, ktoré umožňujú simulovanie života organizmov alebo ich evolúcie. Každý z týchto programov pristupuje k danej problematike iným spôsobom. Definuje a vyvíja rôzne parametre organizmov v rôznom prostredí. V tejto kapitole spomenieme niekoľko programov, ktoré boli inšpiráciou pri tvorbe programu SofE.

2.1 Avida

Avida študuje evolúciu „programov“. Organizmy sa pohybujú vo štvorčekovej sieti, ktorá je v tvare anuloidu. Jeden organizmus sa nachádza na jednom políčku sveta. Organizmy sú „malé počítače“, ktoré vykonávajú svoj kód. Kód je zoznam jednoduchých inštrukcií, ktoré vykonávajú. Rozmnožujú sa asexuálne a pri ich vzniku dochádza k poisson-random mutácii. Avida je implementovaná ako dva programy. Existuje konzolová a GUI verzia programu. Teda má oddelenú implementáciu GUI od samotnej simulácie.

Bola vyskúšaná verzia 2.0 beta6 pre operačný systém Windows. Avida je spoločný projekt Digital Life Laboratory v California Institute of Technology, <http://dllib.caltech.edu/>, a Digital Evolution Group, <http://devolab.cse.msu.edu/>. Domovská stránka projektu <http://dllib.caltech.edu/avida/>.



Obrázok 2.1: Avida 2.0 beta6 – GUI verzia.

2.2 Gene Pool

Program simuluje evolúciu plávajúcich organizmov v „Darwinovom akváriu“. Organizmy si vyberajú svojich partnerov na základe príťažlivosti. Program sa snaží porovnať rozdiely v evolúcii pri prirodzenom výbere a pri sexuálnom výbere (výbere podľa atraktívnosti). Telo organizmov sa skladá z niekoľkých častí. Organizmy sa živia potravou v akváriu. Nový organizmus vzniká spárením dvoch organizmov. Genotyp nového organizmu vznikne skrížením a mutáciou genotypu rodičov. Potomok získa 50% energie od obidvoch rodičov. [4]

Gene Pool bol vytvorený Jeffreyom Ventrellom. Projekt je odvodený od projektu Darwin Pond. V súčasnosti je dostupná šiesta verzia programu Gene Pool, ktorá tu bola vyskúšaná. Domovská stránka projektu <http://www.swimbots.com/>.



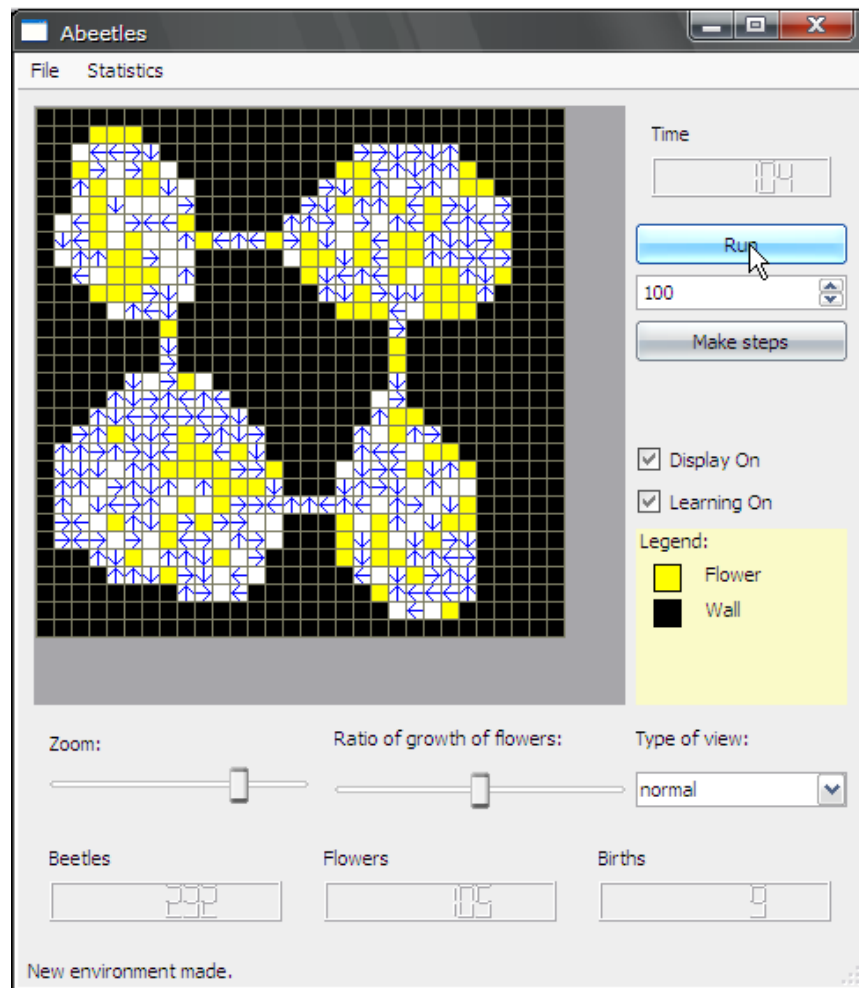
Obrázok 2.2: Gene Pool 6.

2.3 Broucci a Abeetles

Program Abeetles vznikol v rámci diplomovej práce „Nástroj pro modelování evoluce umělého života“ Ivety Bartůňkovej. Vychádzal z programu Broucci, ktorého autorom je pán RNDr. Tomáš Holan, PhD. Tento program bol predstavený v rámci predmetu Programování 2 (na Matematicko-fyzikálnej fakulte Univerzity Karlovej v Prahe), ako ukážka rôznych prístupov k riešeniu komplexných problémov. V programe sa nachádza svet s potravou, kde žijú organizmy (chrobáky). Organizmy sa môžu pohybovať, vidia, čo sa nachádza okolo nich. Pri pohybe dopredu môžu zjesť potravu, iný organizmus, alebo sa len postaviť na voľné políčko. Cieľom tohto programu bolo nájsť najlepší algoritmus organizmov, aby nevymreli, a aby sa nenechali zjesť iným organizmom. K tomuto účelu bolo použité genetické programovanie. [5]

Program Abeetles vyšiel z myšlienok programu Broucci. Organizmy sa taktiež rozhodujú na základe svojho okolia (čo je pred organizmom, na ľavo a na pravo od organizmu) a môžu vykonať určité druhy pohybov. Pridáva však ďalšie možnosti

a parametre pre evolúciu organizmov. V programe Broucci nedochádzalo k páreniu. Pri poklese počtu pod určitú hranicu sa vygenerovali z najlepšieho organizmu nový jedinci. Abeetles doplnil párenie organizmov, schopnosť organizmov učiť sa a množstvo ďalších.



Obrázok 2.3: Abeetles

2.4 Záver

Základné myšlienky programu SofE vychádzajú z programov Broucci a Abeetles. Sú nimi napríklad tvar a rozdelenie sveta organizmov, schopnosť organizmov vykonať určité pohyby, schopnosť organizmov rozlišovať veci vo svojom okolí a množstvo ďalších.

Pre rozdelenie programu na dve časti bol inšpiráciou program Avida.

Okrem týchto prevzatých myšlienok obsahuje program SofE aj nové myšlienky a prístupy. Napríklad rozlišuje rôzne druhy potravy a druhov organizmov, organizmy sa rozhodujú na základe štyroch políčok vo svojom okolí a množstvo ďalších.

3 Umelý život v SofE

Táto kapitola popisuje model života v simulátore SofE a niektoré jeho implementačné detaily. Tento model vychádza z programov Broucci a Abeetles. Postupne popíšeme jednotlivé prvky života v tejto simulácii.

Organizmy sa v tejto simulácii pohybujú v danom svete, a to ich stojí určité množstvo energie. Energiu získavajú z potravy, ktorú nachádzajú vo svete. Vývoj organizmov sa uskutočňuje prostredníctvom zmien parametrov organizmov. Tieto zmeny nastávajú pri vzniku nového organizmu.

3.1 Svet simulácie

Celá simulácia sa odohráva na štvorčekovej sieti. Potrebné informácie o svete získa program z rastrového obrázku, ktorý mu zadáme v nastaveniach. Veľkosť sveta je daná týmto obrázkom. Jedno políčko sa rovná jednému pixelu. Na jednom políčku vo svete sa môže tvoriť len jeden druh potravy. Každý druh potravy je reprezentovaný inou farbou. Čierna a biela farba sú však rezervované pre iné účely. Čierna farba reprezentuje steny, teda miesta, cez ktoré sa organizmy nedostanú. Biela farba reprezentuje políčka, na ktorých nerastie žiadna potrava. Organizmus môže získať z jednej potravy 40 jednotiek energie. Ukážka rastrového obrázku sveta vid' Obrázok 5.1.

3.2 Popis parametrov organizmov

Organizmy majú niekoľko parametrov, ktoré predstavujú ich génom. Tieto parametre boli navrhnuté, aby sa čo najviac podobali génomu skutočných jednoduchých organizmov. Postupne si rozoberieme všetky parametre organizmov.

3.2.1 Potrava organizmov

Organizmy sa živia potravou, ktorá je rastlinná alebo živočíšna (organizmy, ktoré sa okrem rastlinnej potravy živia aj mäsom môžeme nazývať všežravé. Pre zjednodušenie budeme takto nazývať aj organizmy, ktoré sa živia výhradne inými organizmami.). Jednotlivé druhy organizmov sa nemusia živiť všetkými druhmi potravy. Množina potrav, ktorými sa živí organizmus, nepodlieha kríženiu pri párení rovnakých druhov organizmov. Nikdy nedochádza k mutácii tejto množiny.

3.2.2 Vek

Vek organizmov sa počíta v rokoch (v nastaveniach sa určí akému množstvu krokov zodpovedá jeden rok). Maximálna dĺžka života jednotlivých druhov organizmov (vid' Druhy organizmov) je obmedzená určitou hranicou, ktorú definujeme v nastaveniach pre druh organizmov. Tento parameter nepodlieha kríženiu pri párení rovnakých druhov organizmov. Nikdy nedochádza k mutácii tohto parametru.

3.2.3 Cena pohybov

Organizmy môžu vykonať štyri rôzne pohyby. Urobiť krok vpred, otočiť sa vľavo, otočiť sa vpravo a neurobiť nič. Po vykonaní každého pohybu sa organizmu odoberie určité množstvo energie (cena daného pohybu). Cena pohybu sa nesmie rovnať nule. Pokiaľ nemá organizmus dostatok energie pre vykonanie pohybu, tak hynie. Ceny pohybov sú rovnaké pre jeden druh organizmov (vid' Druhy

organizmov). Tieto parametre nepodliehajú kríženiu pri párení rovnakých druhov organizmov. Nikdy nedochádza k ich mutácii.

3.2.4 Energia organizmov

Tento parameter obsahuje dve hodnoty. Prvá určuje množstvo energie, ktorú získal organizmus z konzumovania rastlín. Druhá určuje množstvo energie, ktorú získal z konzumovania iných organizmov (mäsa). Energia z mäsa sa spotrebúva prednostne. Druhý parameter má význam iba pre všežravé organizmy. Organizmus skonzumuje potravu, keď sa postaví na políčko s touto potravou (alebo ak na danom políčku stojí a potrava sa tam vytvorí). Pre získanie energie z iných organizmov musí všežravý organizmus urobiť krok vpred na políčko, kde sa nachádza nejaký organizmus. Potom dochádza k súboju vid' Schopnosť zabíjať'.

3.2.5 Násobok pre spotrebu energie z mäsa

Tento parameter má význam pre všežravé organizmy. Určuje číslo, ktorým sa prenášobí cena pohybu, pokiaľ sa energia odpočítava z energie získanej z mäsa. Takto môžeme zvýhodňovať alebo znevýhodňovať organizmy, ktoré sa živia mäsom, podľa našej potreby.

3.2.6 Limit hladu

Všežravé organizmy sa nemusia živiť inými organizmami počas celého svojho života. Tento parameter určuje hladinu energie. Pokiaľ sa energia všežravého organizmu dostane pod túto hladinu, tak môže skonzumovať iný organizmus. Pokiaľ je však jeho energia nad touto hladinou, tak nebude napádať iné organizmy, bude sa živiť normálnou potravou. Na základe tohto parametru sa určí pre všežravé organizmy, či je organizmus nebezpečný alebo nie (vid' Algoritmus organizmov).

3.2.7 Funkcia Energia vs. Vek

V skutočnosti sú to dve funkcie. Obidve súvisia s energiou a vekom. Prvá určuje percento získanej energie z potravy v určitom veku.

Napríklad organizmus v 20 rokoch získa z potravy 80% energie a v 50 rokoch len 60%. Z jednej potravy získa 40 jednotiek energie. Ak zje túto potravu v 20 rokoch, tak v skutočnosti získa 32 jednotiek energie. Ak v 50 rokoch, tak získa 24 jednotiek energie.

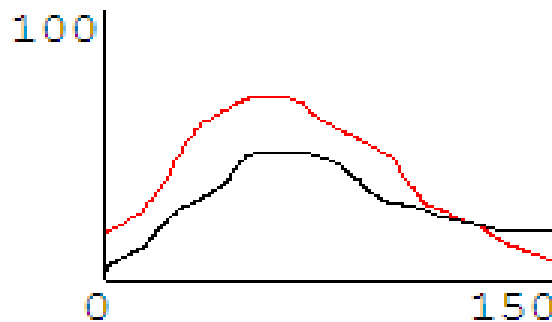
Druhá určuje percento energie organizmu, ktoré je schopný vložiť do boja, vzhľadom k jeho veku.

Napríklad organizmus v 10 rokoch je schopný vložiť do boja len 20% svojej energie, ale organizmus v 30 rokoch je schopný vložiť do boja 70% svojej energie.

Táto funkcia sa používa pri súbojoch organizmov (vid' Schopnosť zabíjať').

Tieto funkcie sa zadávajú do simulácie prostredníctvom rastrového obrázku (vid' Obrázok 3.1), kde x-ová os predstavuje vek a y-ová os je v rozsahu 1-100 a predstavuje percentá. Za bod [0, 0] sa považuje ľavý dolný roh obrázku. Každý pixel je jeden rok. Čierna krivka predstavuje prvú funkciu a červená krivka predstavuje druhú funkciu. Pokiaľ je niektorá krivka dlhšia ako maximálny vek daného druhu organizmov, tak sa nadbytočné hodnoty nepoužívajú. Pokiaľ je niektorá krivka kratšia ako maximálny vek, tak sa používa posledná hodnota až do smrti organizmu. Pokiaľ sa krivky krížia, tak na ich prieniku musí byť zobrazená čierna farba. Pokiaľ

pre daný vek nie je definovaná červená krivka, tak sa použije hodnota pre čiernu krivku. Povinná je len čierna krivka. Červenú nemusíme vôbec zadať.



Obrázok 3.1: Ukážka funkcie Energia vs. Vek

Tieto hodnoty sú spoločné pre jeden druh organizmov. Nemenia sa pri krížení rovnakých druhov organizmov a nikdy nedochádza k ich mutácii.

3.2.8 Schopnosť zabíjať

Tento parameter má význam pri súbojoch organizmov. K súboju dochádza, keď sa všežravý organizmus snaží skonzumovať iný organizmus. Súboj je realizovaný ako výpočet funkcie, v ktorej sa zohľadňuje schopnosť organizmov zabíjať, energia a vek (schopnosť využitia energie v boji vzhľadom k veku vid' Funkcia Energia vs. Vek). Schopnosť zabíjať predstavuje skúsenosť organizmu s bojom. V nastaveniach uložených v súbore (vid' Príloha A - Užívateľská dokumentácia, podkapitola Nastavenia druhu organizmov) musí byť táto hodnota zadaná. Pri používaní grafického rozhrania je v dialógu pre novú simuláciu táto hodnota nastavená na nulu.

Z funkcie Energia vs. Vek sa podľa veku zistí koľko percent energie vloží organizmus do boja. Touto hodnotou sa prenásobí súčasná energia organizmu. Výslednú hodnotu budeme nazývať ťažkosť boja. Ťažkosť boja sa prenásobí schopnosťou zabíjať (pokiaľ je schopnosť nulová, tak sa výsledok prenásobí jednotkou). Tento výpočet sa uskutoční pre oba bojujúce organizmy. V súboji vyhráva organizmus, ktorému vyšla väčšia hodnota. Pokiaľ vyhrá v súboji útočiaci organizmus, tak usmrť brániaci sa organizmus. Ako výhru si pripočíta energiu porazeného organizmu prenásobenú percentom z funkcie Energia vs. Vek. Ak prehrá útočiaci organizmus, tak sa obidvom organizmom zníži energia o hodnotu ťažkosti boja.

Po každej výhre sa víťazovi zvýši schopnosť zabíjať o jedna.

3.2.9 Parametre pre párenie

Párenie prebieha medzi pasívnym a aktívnym organizmom. Pasívny organizmus stojí na mieste (matka) a po spárení ostáva tehotný. Aktívny organizmus (otec) urobí krok vpred k pasívnemu organizmu (matka) a po oplodnení ostáva na svojom pôvodnom políčku.

Parametre pre párenie obsahujú dva typy informácií. Pomocou prvých organizmy selektujú vhodných partnerov pre párenie:

- Minimálna energia, ktorú musí mať partner.
- Minimálny a maximálny vek partnera.

- Požadovaná energia od partnera – pri párení požaduje pasívny organizmus určité množstvo energie od aktívneho organizmu.
- Energia predaná partnerovi – pri párení predáva aktívny organizmus určité množstvo energie pasívnemu organizmu. Tento parameter určuje hornú hranicu tejto predávanej energie.

Druhý typ informácií predstavujú dva parametre. Prvý parameter určuje koľko energie predá pasívny organizmus svojmu potomkovi pri počatí (energia predaná potomkovi). Druhý parameter určuje dĺžku tehotenstva.

K spáreniu môže dôjsť medzi organizmami, ktoré navzájom splňajú prvý typ parametrov.

Párenie je realizované genetickými algoritmami. Dochádza ku kríženiu niektorých parametrov organizmov pomocou operácie one-point crossover. Následne dochádza k mutácii týchto parametrov podľa určitej pravdepodobnosti. Táto pravdepodobnosť je spoločná pre jeden druh organizmov. Meniť ju môže len užívateľ.

3.2.10 Algoritmus organizmov

Algoritmus organizmov sa skladá z niekoľkých stavov. Ich počet sa pre jednotlivé druhy definuje v nastaveniach pre druh organizmov. Tento počet sa pre daný druh nemení počas celej simulácie. K zmene dochádza len pri spárení dvoch rôznych druhov organizmov. Jeden stav organizmu je tvorený 2401 hodnotami. Organizmus si na základe svojho vnútorného stavu a svojho okolia (čo je pred ním, za ním, naľavo a napravo od neho) vyberie z tabuliek pohyb, ktorý vykoná. S každým pohybom je spojený prechod do nejakého stavu.

Organizmus označí každé políčko vo svojom okolí jednou zo siedmich značiek:

- Moja potrava – potrava (rastlinka), ktorou sa živí.
- Potrava – potrava (rastlinka), ktorou sa neživí.
- Vhodný organizmus – organizmus, s ktorým sa môže spáriť.
- Nebezpečný organizmus – všežravý organizmus, ktorého energia klesla pod limit hladu.
- Organizmus – iný organizmus.
- Stena – políčko na mape, na ktorom sa nachádza stena.
- Nič – voľné políčko na mape.

Rozlíšenie organizmu na nebezpečný, vhodný a iný sa deje podľa nasledujúceho postupu:

- Nebezpečný organizmus – ak je to všežravý organizmus a jeho energia je pod limitom hladu.
- Vhodný organizmus – pokiaľ nemôže konzumovať iné organizmy, tak to je organizmus, ktorý splňa podmienky pre spárenie. Ak je to všežravý organizmus, tak musí splňať podmienky pre spárenie a jeho energia musí byť nad limitom hladu.
- Iný organizmus – všetky ostatné organizmy.

Pre uľahčenie práce je v inštalačnom balíku uložený implicitný algoritmus. Ten sa využíva aj pri vytváraní novej simulácie v programe s grafickým rozhraním. Tento implicitný algoritmus má jeden stav. A pohyby organizmu sa v ňom postupne striedajú.

3.3 Druhy organizmov

Program rozlišuje druhy organizmov, na základe potravy, ktorou sa živia. Jediný úkon, ktorý je ovplyvnený rozdelením na druhy je párenie. Pri vysvetľovaní párenia rôznych druhov budeme uvažovať o druhu ako o množine potrav, ktorými sa živia. Užívateľ má možnosť pre každú dvojicu rôznych druhov nastaviť pravdepodobnosť ich spárenia. Tým, že sme povolili párenie rôznych druhov organizmov, nám môžu vzniknúť úplne nové druhy, ktoré sme v nastaveniach ani neuviedli. Preto pri nastavovaní pravdepodobnosti spárenia dvoch rôznych druhov musíme uvažovať o všetkých dvojiciach rôznych druhov organizmov vzhľadom k množine potrav, ktorými sa organizmy živia.

Príklad

Všetky druhy potravy, ktorými sa organizmy živia (nie všetky v danom svete):

1, 2, Mäso

Všetky možné dvojice:

{1}, {2}; {1}, {Mäso}; {1}, {1, 2}; {1}, {1, Mäso}; {1}, {2, Mäso};
{1}, {1, 2, Mäso}; {2}, {Mäso}; {2}, {1, 2}; {2}, {1, Mäso};
{2}, {2, Mäso}; {2}, {1, 2, Mäso}; {Mäso}, {1, 2}; {Mäso}, {1, Mäso};
{Mäso}, {2, Mäso}; {Mäso}, {1, 2, Mäso}; {1, 2}, {1, Mäso};
{1, 2}, {2, Mäso}; {1, 2}, {1, 2, Mäso}; {1, Mäso}, {2, Mäso};
{1, Mäso}, {1, 2, Mäso}; {2, Mäso}, {1, 2, Mäso}

4 Popis implementácie

Program SofE bol napísaný v jazyku C++. Pre tvorbu grafického užívateľského rozhrania (GUI) využíva multiplatformnú GUI knižnicu wxWidgets [6]. Táto knižnica v spolupráci s jazykom C++ nám umožňuje, aby pri čo najmenších zmenách bolo možné v budúcnosti program skompilovať aj pre iné platformy ako je Windows. V súčasnosti bol program testovaný pre platformy Windows XP a Windows Vista. Táto knižnica bola vybraná aj kvôli svojej komplexnosti.

Simulácia evolúcie a jej grafická vizualizácia sú časti aplikácie, ktoré sú logicky a implementačne oddelené. Toto oddelenie umožňuje spúšťať simulácie bez použitia grafického rozhrania, ktoré do značnej miery spomaľuje aplikáciu. Rýchlosť je pre simulácie, ktoré požadujú vykonať veľké množstvo krokov, veľmi dôležitá. Zdrojové kódy pre simuláciu a jej grafické rozhranie sú taktiež oddelené.

Zdrojové súbory pre simuláciu sa nachádzajú v adresári *Project\src\sim* a súbory pre grafické rozhranie sa nachádzajú v adresári *Project\src\graphic*.

Celý zdrojový kód a komentáre v kóde sú napísané v anglickom jazyku. Pri pomenovaní tried a štruktúr je používaná maďarská notácia, kde sa názvy tried začínajú písmenom C, názvy štruktúr a typov sa začínajú písmenom T.

Implicitne povoľuje program spustiť len jednu inštanciu. Pomocou konfiguračnej položky *oneInstance* v súbore *config\config.txt* môžeme povoliť viac inštancií programu.

4.1 Riešenie grafického užívateľského rozhrania

4.1.1 Podpora viacerých jazykov

Podpora viacerých jazykov spríjemňuje užívateľom používanie programov. Nikdy nevieme, kto bude daný software využívať. Knižnica wxWidgets obsahuje mechanizmus pre používanie katalógov správ. Tieto katalógy sú binárne kompatibilné s gettext katalógmi, ktoré využíva GNU Translation Project [7].

Pre vytváranie katalógov správ bol použitý program Poedit (<http://www.poedit.net/>), ktorý vyhľadáva texty pre preklad v zdrojových súboroch na základe makra `gettext _()`. Následne poskytne celý zoznam nájdených textov, kam môžeme uviesť jednotlivé preklady. Pri uložení súboru priamo vygeneruje binárnu verziu katalógu.

Program prekladá všetky texty zobrazované v oknách a dialógoch. Okrem toho prekladá aj texty výnimiek, ktoré môžu byť v niektorých prípadoch zobrazené užívateľovi ako varovné alebo výstražné hlásenie.

V súčasnosti sú tieto texty preložené do anglického a slovenského jazyka. Pre pridanie nového katalógu (a tým aj nového jazyka programu) je potrebné vytvoriť nový katalóg, ktorý sa uloží do podadresára *lang* inštalačného adresára. Pri vytváraní katalógu je potrebné nastaviť správny názov jazyka. Tento názov sa potom uvedie do konfiguračnej položky *lang* v súbore *config\config.txt* a oddelí sa bodkočiarkou od ostatných jazykov. Implicitným jazykom programu je prvý jazyk v tomto zozname.

4.1.2 Rozvrhnutie dialógu pre novú simuláciu

Pre vytvorenie novej simulácie je potrebné nastaviť veľké množstvo parametrov, preto bolo potrebné navrhnuť tento dialóg veľmi prehľadne.

Pre opätovné použitie nastavení má užívateľ možnosť uložiť a načítať tieto nastavenia do súborov vid' Príloha A - Užívateľská dokumentácia, podkapitola Súbor s nastaveniami. Vykonalie týchto operácií mu umožňujú tlačidlá umiestnené v lište nástrojov pri hornom okraji dialógu. Dialóg si neukladá posledné použité nastavenia simulácie. Pri každom spustení je dialóg prázdny. Užívateľ má možnosť spustiť viac týchto dialógov súčasne, čo mu umožní porovnávanie vkladných hodnôt.

Nastavenia simulácie môžeme rozdeliť na dve skupiny. Prvú tvoria všeobecné nastavenia simulácie. Tie sa nachádzajú v hornej časti dialógu a sú graficky oddelené. Druhú tvoria nastavenia druhov organizmov, ktoré sa nachádzajú vo zvyšnej časti dialógu. Prvky využívané pre vkladanie a odoberanie druhov organizmov sú taktiež graficky oddelené.

Počiatočný algoritmus druhov je natoľko rozsiahly, že je ho možné zadať iba prostredníctvom súboru.

Zoznam potrav pre druh organizmov, závisí od sveta, v ktorom sa pohybujú. Počet druhov potrav vo svete nie je obmedzený. Preto pre zobrazenie všetkých potrav bolo potrebné využiť grafický prvok, ktorý umožňuje uložiť dynamický počet textov. Medzi štandardné grafické prvky, ktoré to umožňujú patrí listbox a combobox. Kvôli malému priestoru a potrebe prehľadnosti sa využíva pre zobrazenie všetkých potrav vo svete upravený combobox. Ten zobrazuje okrem názvu aj farebný štvorec, ktorého farba zodpovedá danému druhu potravy. Pre vybratie danej potravy je použitý jednoduchý checkbox.

4.1.3 Rozvrhnutie okna pre simuláciu

Okno so simuláciou slúži pre sledovanie aktuálneho vývoja evolúcie, preto bol kladený dôraz na zobrazovanie sveta organizmov a poskytnutie, čo najväčšieho množstva informácií o vývoji evolúcie.

Svet sa kreslí na špeciálne rolovateľné okno. Je prekresľované na základe tikotu časovača, ktorého rýchlosť môže užívateľ ovplyvniť. Pred vykreslením tohto okna sa uskutoční jeden krok simulácie. Pri každom tiku časovača sa prekreslia len zmenené políčka sveta. Program umožňuje približovať a oddiaľovať svet, čo je sprostredkované prepočtom veľkosti jednej bunky sveta a zmenou veľkosti obrázkov organizmov.

Pre zadávanie pravdepodobností tvorby potravy a mutácie jednotlivých druhov boli použité comboboxy, pretože umožňujú uložiť dynamicky sa meniace množstvo textov, a sú priestorovo úsporné. Pre jednu pravdepodobnosť sa používajú dva prepojené comboboxy. Bolo potrebné umožniť užívateľovi meniť tieto pravdepodobnosti. Pokiaľ by sme chceli na okno umiestniť ďalší grafický prvok pre vkladanie nových pravdepodobností, tak by okno začalo byť neprehľadné. Preto sa pre zadávanie nových pravdepodobností využívajú rovnaké prvky ako pre ich zobrazovanie. Do comboboxu s pravdepodobnosťami je možné zadať novú hodnotu, ktorá sa po stlačení klávesy *Enter* nastaví.

Program umožňuje zmeniť zobrazenie sveta organizmov, kvôli poskytnutiu názorných informácií o vývoji evolúcií vid' Príloha A - Užívateľská dokumentácia podkapitola Okno simulácie. Niektoré z týchto zobrazení upravujú vykresľovanie na základe niektorého parametru organizmov. Rozmedzia pre toto vykresľovanie sú umiestnené v konfiguračnej položke `legend.graph.*.value`, kde * predstavuje názov daného parametru. Pomocou konfiguračnej položky

legend.graph.*.color je možné meniť farbu daného vykresľovania. Obrázky organizmov sú v spustenej simulácii uložené v triede wxBitmap. Pri vykresľovaní sa mení farba organizmov prostredníctvom masky a zmeny kresliaceho štetca. Masku určuje, ktoré časti obrázku sa nemajú vykresľovať, tieto časti sú potom vyplnené pomocou štetca. Okrem iného vykresľovania sveta bolo potrebné poskytnúť užívateľovi legendu, aby vedel, čo dané vykreslenie znamená. Pre tento účel bol v tomto prípade zvolený grafický prvok listbox, aj napriek jeho väčšiemu rozmeru, pretože legendu potrebuje užívateľ neustále porovnávať s vykresľovaným svetom.

Kvôli tomu, že majú organizmy veľké množstvo parametrov bolo potrebné vytvoriť dialóg pre zobrazenie týchto informácií pre jeden organizmus. V okne simulácie je možné zobraziť len základné parametre organizmu.

4.1.4 Uloženie a načítanie simulácie/-í

Aplikácia SofE umožňuje v grafickom rozhraní ukladať jednu alebo všetky simulácie. Táto funkcia je užitočná pre uloženie aktuálneho stavu simulácie, v ktorom budeme môcť pokračovať po jeho načítaní. Pri ukladaní simulácie sa okrem informácií pre samotnú simuláciu kopírujú do zvoleného adresára aj pomocné súbory so štatistickými údajmi, rastrové obrázky funkcie Energia vs. Vek, a pokiaľ je možné vracajú simuláciu späť, tak sa kopírujú aj pomocné súbory pre túto funkciu.

Rastrový obrázok sa kopíruje, pretože sa zobrazuje v dialógu s informáciami o organizme (viď Obrázok A.3). Pomocné údaje sa ukladajú preto, aby boli štatistické údaje kompletne, a aby bolo možné vrátiť simuláciu do kroku pred jej uložením.

Informácie o simulácii sú uložené v textovej podobe v jednom súbore. Údaje nie sú nijak predspracované. Svet organizmov a organizmy sú uložené so všetkými svojimi atribútmi. V budúcnosti by bolo vhodné predspracovať údaje pre uloženie, aby došlo k zmenšeniu ukladaného súboru. Pri aktuálnom ukladaní vznikajú duplicity údajov.

Pri implementovaní tejto funkcie bolo potrebné vyriešiť niekoľko problémov. Jeden z nich nastane, keď uložíme simuláciu/-ie v aplikácii, ktorá má zapnutú funkciu vracania sa späť a načítame ju v aplikácii, ktorá ju má vypnutú. V tomto prípade sa pre túto načítanú simuláciu pridá táto funkcia do aplikácie. Pre ostatné simulácie však bude funkcia vracania sa späť vypnutá. Podobný problém nastane, ak uložíme simuláciu/-ie v aplikácii, ktorá má funkciu vracania sa späť vypnutú a načítame ju v aplikácii, ktorá má túto funkciu zapnutú. V tomto prípade sa pre načítanú simuláciu táto funkcia vypne. Pre ostatné simulácie je však zapnutá.

4.1.5 Vrátenie simulácie späť

Pri využívaní grafického rozhrania je užitočná funkcia programu pre vrátenie simulácie naspäť do určitého kroku/roku. Pri simulácii dochádza k veľkému množstvu zmien, ktoré by bolo treba ukladať pri klasickej implementácii Undo, podľa návrhového vzoru Memento. Každý organizmus zmení v jednom kroku niekoľko parametrov, organizmy hynú a vytvárajú sa nové, hynú alebo sú zjedené jednotlivé rastliny. Preto SofE nepoužíva tento návrhový vzor.

Využíva to, že celý krok simulácie je riadený generátorom náhodných čísel. Na začiatku jedného kroku sa vždy inicializuje generátor náhodných čísel novým náhodným semienkom. Pre každú náhodnú činnosť, ktorá sa deje v tomto kroku

vytvorí generátor náhodné číslo. Program si ukladá tieto náhodné semienka, čím je možné presne odsimulovať priebeh simulácie od začiatku do určitého kroku.

Okrem toho ukladá pravidelne celú simuláciu. Períodu ukladania určuje konfiguračná položka `store.saveSimEach`, ktorá sa nachádza v súbore `config\graphic_sim.txt`.

To ešte stále nestačí pre presné odsimulovanie pôvodnej simulácie, pretože užívateľ môže meniť pravdepodobnosti tvorby potravy a mutácie druhov organizmov. Preto program ukladá aj tieto zmeny.

Pri požiadavke o vrátenie sa späť do určitého kroku program spočíta, v ktorom súbore nájde najbližšie uloženú simuláciu. Načíta si ju do pamäte, a v pozadí ju odsimuluje do žiadaného kroku/roku. Nakoniec sa len aktualizujú hodnoty v okne so simuláciou. Pokiaľ chce užívateľ zopakovať simuláciu, tak sa používajú pôvodné náhodné semienka a zmeny pravdepodobností, ktoré spravil užívateľ predtým.

Výhodnosť tohto riešenia závisí na intervaloch uloženia simulácie a na veľkosti ukladaných údajov. Pre veľký počet organizmov dochádza k podstatnému spomaleniu pri ukladaní. Pri veľkej perióde ukladania nedochádza k častému spomaleniu simulácie, ale pre vrátenie sa späť je potreba viac času pre opätovné odsimulovanie.

Túto funkciu je možné úplne vypnúť (viď Príloha A - Užívateľská dokumentácia podkapitola Vrátenie simulácie späť), aby sme sa zbavili zbytočného spomalenia.

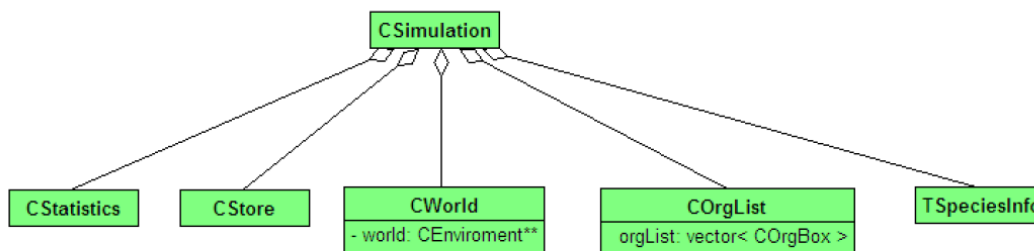
4.1.6 Nápvoda

Knižnica `wxWidgets` poskytuje multiplatformné riešenie pre vytváranie a zobrazovanie HTML nápovedy. Pre vytvorenie HTML nápovedy v programe SofE bol použitý software `HelpBlocks` (<http://www.helpblocks.com>), ktorý umožňuje jednoduché vytváranie nápovedy, a jej kompiláciu vo formáte MS HTML pre platformu Windows (tento formát používa program SofE) a formáte `wxHTML`, ktorý je multiplatformný.

4.2 Problémy pri vytváraní samotnej simulácie

4.2.1 Komponenty simulácie

Funkcionalita simulácie je rozdelená medzi tieto triedy `CSimulation`, `CWorld`, `CStatistics`, `CStore`, `COrgList` a štruktúru `TSpeciesInfo` (viď Obrázok 4.1).



Obrázok 4.1: Diagram hlavných tried a štruktúr simulácie

Trieda `CSimulation` predstavuje jednu simuláciu. Obsahuje jednu metódu `DoStep` pre uskutočnenie jedného kroku simulácie. Tým umožňuje grafickému rozhraniu, aby pre jeden tikot časovača urobila simulácia jeden krok. V tejto metóde sa uskutoční jeden krok, pre každé políčko sveta a pre každý organizmus. Ako jediná má prístup k triedam `CStatistics` a `CStore`. Prvá sa stará o uchovávanie a správu štatistických údajov. Druhá o ukládanie a správu informácií pre vrátenie simulácie späť.

Jednotlivé organizmy sú uložené vo svete a v zozname organizmov, ktorý je uložený v triede `CSimulation`. Toto dvojité uloženie je vhodné pre náhodné prechádzanie organizmov v jednom kroku simulácie.

4.2.2 Algoritmus organizmov

Algoritmus organizmov pre jeden stav je uložený v samostatnej triede `CAlgorithmTable`. Tento algoritmus je veľmi rozsiahly, preto do značnej miery ovplyvňuje práca s ním rýchlosť celej aplikácie. V simulácii sa uskutočňujú dve základné operácie s týmto algoritmom. Dochádza k jeho kríženiu, a potom k jeho mutácii.

Počas vývoja aplikácie bol algoritmus pre jeden stav reprezentovaný štvorrozmerným poľom štruktúr `pair` z knižnice `STL`. Táto implementácia bola vhodná pre ladiace účely. V praxi však program strávil pri simulácii 50% času v metódach, ktoré pracovali s týmto algoritmom. Preto sa jeho reprezentácia zmenila na jednorozmerné pole, v ktorom je zlinearizované pôvodné štvorrozmerné pole. Pri jeho krížení a kopírovaní sa používa nízkoúrovňová funkcia `memcpy`, ktorá nerobí žiadne kontroly, ale umožňuje veľmi rýchle kopírovanie dát. Táto úprava zrýchlila celú simuláciu.

Stále je však tento algoritmus Achillovou päťou celej aplikácie. Pri mutácii tohto algoritmu strávi simulácia viac ako 40% času. Preto boli vytvorené konfigurovatelné obmedzenia pre mutáciu algoritmu organizmov pre jeden stav. Pomocou konfiguračnej položky `standardOrg.mutateAlgo` je možné určiť, či sa má program pokúsiť zmutovať každú položku algoritmu, alebo či sa má použiť iný spôsob mutácie algoritmu. Iný spôsob sa pokúsi zmutovať iba určitý počet políčok. Tento počet je definovaný v konfiguračnej položke `standardOrg.maxFieldToMutate`. Implicitne je v simulácii nastavený druhý spôsob mutovania algoritmu, a počet políčok pre zmutovanie je nastavený na 800.

Tieto obmedzenia na mutáciu algoritmu organizmov sú diskutabilné, pretože môžu ovplyvniť evolúciu danej simulácie. Obmedzujú totiž mutovanie organizmov. Záleží však na konkrétnej simulácii, či dôjde k výraznému ovplyvneniu evolúcie alebo nie. V mnohých prípadoch dôjde len k spomaleniu vývoja algoritmu organizmov.

4.2.3 Vlákna

Pri spustení programu pomocou skriptu sú spustené všetky simulácie v samostatných vláknach. Jedna simulácia je navrhnutá tak, aby bola schopná pracovať samostatne. Jediným bodom, v ktorom musia simulácie spolu komunikovať je fond organizmov. V ostatných prípadoch je každá simulácia samostatný program. Jedno vlákno je reprezentované triedou `CSimThread`. Tá zabezpečuje volanie metódy `DoStep` pre uskutočnenie jedného kroku simulácie,

zmenu pravdepodobnosti tvorby potravy určitého druhu a o ukládanie štatistických údajov.

Pomocou identifikačného čísla vlákna a názvu simulácie je pomenovaný súbor, v ktorom sa zapisuje priebeh simulácie, ktorá bola spustené pomocou skriptu.

4.2.4 Fond organizmov

V simulácii dochádza k častému vzniku a zániku organizmov. Jeden organizmus obsahuje veľké množstvo parametrov, ktoré musia byť uložené v jeho atribútoch. Pokiaľ by sme neustále alokovali a dealkovali pamäť potrebnú pre organizmus došlo by k značnému spomaleniu celej aplikácie. Zbytočne by sa zaťažoval správca pamäte.

Z toho dôvodu je v programe vytvorený fond organizmov. Za každým, keď simulácia potrebuje vytvoriť nový organizmus, tak požiada triedu `COrgFactory`. Keď organizmus zahynie, tak ho simulácia vráti tejto triede. Táto trieda je implementovaná ako Singleton. Je to jediný spoločný bod pre dve rôzne simulácie.

`COrgFactory` udržiava zoznamy vytvorených organizmov, ktoré sa hierarchicky delia na základe niekoľkých kritérií. Ako prvé je identifikačné číslo vlákna, v ktorom je spustená simulácia, ktorá žiada o nový organizmus. Toto delenie je vytvorené, kvôli rôznym vláknam pri spustení aplikácie pomocou skriptu. Po ukončení práce vlákna sa oznámi fondu, že toto vlákno ukončilo prácu, a fond uvoľní všetky zdroje pre simuláciu v tomto vlákno. Potom pokračuje delenie na základe implementácie žiadaného organizmu. V súčasnosti existuje v programe len jedna implementácia organizmov. Toto delenie je vytvorené skôr pre budúce úpravy programu. Ako posledné sa uplatní delenie na základe počtu stavov organizmu.

Podľa týchto kritérií má trieda `COrgFactory` rozdelené zoznamy organizmov. Jeden zoznam uchováva v triede `COrgFond`. Táto trieda obsahuje zoznam všetkých vytvorených organizmov (pre dané kritéria). Na začiatku tohto zoznamu sú voľné organizmy, a trieda obsahuje index na posledný voľný organizmus. Pri požiadavke o organizmus vráti posledný voľný organizmus. Ak voľný organizmus neexistuje, tak vytvorí nový pomocou metódy `Clone`. Fond organizmov používa návrhový vzor `Prototype` pre vytváranie organizmov. Pri vracaní organizmu sa zavolá na organizme metóda `Reset`, ktorá má za účel uvoľniť prostriedky, ktoré sú zviazané s konkrétnym organizmom v nejakej simulácii a inicializovať atribúty organizmu (napríklad ak organizmus obsahuje odkaz na svojho potomka, tak ho musí vrátiť fondu organizmov).

4.2.5 Spolupráca organizmov

Organizmy potrebujú zistiť, čo sa nachádza v ich okolí. Od triedy `CSimulation` dostanú len `vector` odkazov na triedy `CEnvironment`, ktoré predstavujú políčka vo svete. Organizmy si ich sami označia príslušnými značkami. Pokiaľ sa na niektorom políčku nachádza iný organizmus, tak ho požiadajú, aby im povedal, či je nebezpečný, vhodný pre párenie alebo len obyčajný organizmus. V tomto bode sa predpokladá, že si organizmy nebudú navzájom klamať (je to len spôsob implementácie, princípy fungovania evolúcie sú vysvetlené v kapitole Umelý život v SofE). Súboj a párenie sú tiež riešené ako určitá forma komunikácie medzi dvoma organizmami.

Pri návrhu aplikácie sa uvažovalo aj o pridávaní nových implementácií organizmov. Pridávanie nových implementácií je podporené samotným riešením jednotlivých úkonov organizmov prostredníctvom komunikácie medzi organizmami. Trieda `COrganism` je okrem toho odvodená od abstraktnej triedy `CAOrganism`, ktorá definuje potrebné operácie pre komunikáciu organizmov. Pre rozlíšenie implementácií organizmov je definovaný typ enum `TTypeOfOrganisms`.

4.2.6 Výpočet štatistických údajov

O ukladanie a výpočet štatistických údajov sa stará trieda `CStatistics`. Je súčasťou každej simulácie. Aplikácia podporuje štyri druhy štatistických údajov. Krokové a rokové štatistické údaje majú rovnaký obsah. Líšia sa len časom, pre ktorý sú počítané. Agregované štatistické údaje sú súčasťou krokových a rokových údajov. Histogramové údaje sa počítajú v momente, keď o to užívateľ požiada. Sú to údaje o všetkých organizmoch v danom stave.

Agregované štatistické údaje sa počítajú na konci kroku. Trieda `CStatistics` získa zoznam organizmov na konci daného kroku. Pre každý druh organizmov spočíta ich agregované štatistické údaje. Pre podporu rôznych výpočtov štatistických údajov bol použitý návrhový vzor `Strategy`. Pomocou konfiguračnej položky `statistic.counter` sa určí, akým spôsobom majú byť počítané agregované údaje. `CStatistics` pracuje len s abstraktným rozhraním `CACounterAggrData`. Momentálne sú implementované dve možnosti počítania údajov. Pomocou mediánu a pomocou priemeru.

Niektoré krokové údaje sa počítajú priebežne. Sú to počty organizmov a potravy. Vždy pri vzniku a zániku organizmu alebo potravy sú dané počty upravené.

5 Experimenty

So simulátorom SofE môžeme uskutočniť veľké množstvo experimentov. Umožňuje nám to množstvo parametrov, ktoré spolu súvisia a ovplyvňujú sa. Napríklad mapa sveta, uloženie stien vo svete, parametre pre párenie, pravdepodobnosť mutácie organizmov a potravy, ...

V tejto kapitole si ukážeme a vysvetlíme štyri experimenty uskutočnené pomocou simulátoru SofE.

Simulátor môže vytvárať štatistické údaje dvoma rôznymi spôsobmi. Buď bude počítat priemery, alebo strednú hodnotu pre jednotlivé parametre organizmov, ktoré žijú v danom kroku. Konkrétny spôsob počítania je možné nastaviť pomocou konfiguračnej položky `statistic.counter`.

Pri spúšťaní jednotlivých experimentov sú niektoré hodnoty pri počítaní priemeru skreslené (konkrétne počet detí organizmu), preto sú výsledky experimentov počítané pomocou strednej hodnoty.

Štatistické súbory, ktoré vytvorí SofE sú pre spracovanie pomocou tabuľkového editoru príliš veľké, preto ich spracovávame pomocným programom `DataProcessor`, ktorý je na priloženom CD. Tento program spočíta priemer alebo strednú hodnotu (typ počítania sa určí pomocou argumentu na príkazovom riadku) určitého množstva dát, podľa toho koľko chceme mať hodnôt vo výsledku.

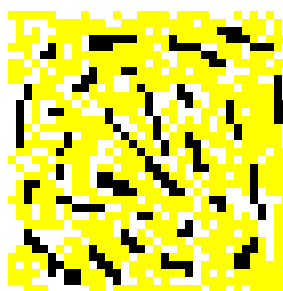
Výsledky všetkých experimentov sú na priloženom CD v adresári `experimenty`.

5.1 Prvý experiment – vplyv pravdepodobnosti mutácie na vývoj populácie

Cieľom experimentu je zistiť, ako vplýva pravdepodobnosť mutácie na organizmy. Ako veľmi sa budú líšiť ich parametre pri zmene tejto pravdepodobnosti? Vývoj pri malej mutácii môže stagnovať, a pri veľkej mutácii nemusí mať žiaden smer.

5.1.1 Počiatočné nastavenia

Mapa má rozmery 35x35 bodov. Pravdepodobnosť tvorby potravy je 50%.



Obrázok 5.1: Mapa sveta pre prvý experiment.

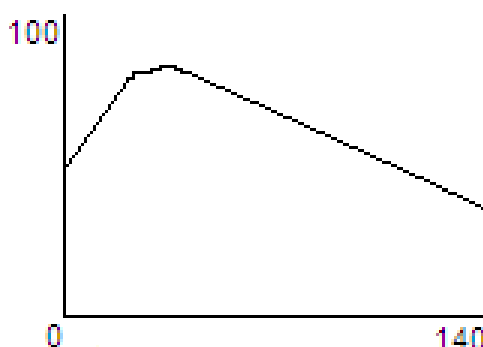
Je len jeden druh organizmov, ktorý sa živí žltou potravou. Počiatočný počet organizmov je 500.

Cena pohybov:

- Krok vpred: 8
- Otočenie vľavo/vpravo: 4

- Nič: 3

Krivka funkcie, ktorá určuje využitie potravy vzhľadom k veku:



Obrázok 5.2: Priebeh funkcie Energia vs. Vek v prvom experimente. Horizontálna os = vek: 0...140, vertikálna os = percento energie z potravy: 0...100.

Maximálny vek organizmov je 140 rokov. Majú implicitný algoritmus.

Parametre pre párenie:

- Minimálny vek partnera: 10
- Maximálny vek partnera: 120
- Energia partnera: 60
- Energia pre potomka: 30
- Energia pre partnera: 40
- Požadovaná energia od partnera: 40
- Dĺžka tehotenstva: 10

Pravdepodobnosť mutácie organizmov je testovaným parametrom. Pravdepodobnosť mutácie pre jednotlivé testy: 5%, 30%, 50%, 70%, 100%. Simulácia je pre každú pravdepodobnosť spustená na 500 000 krokov.

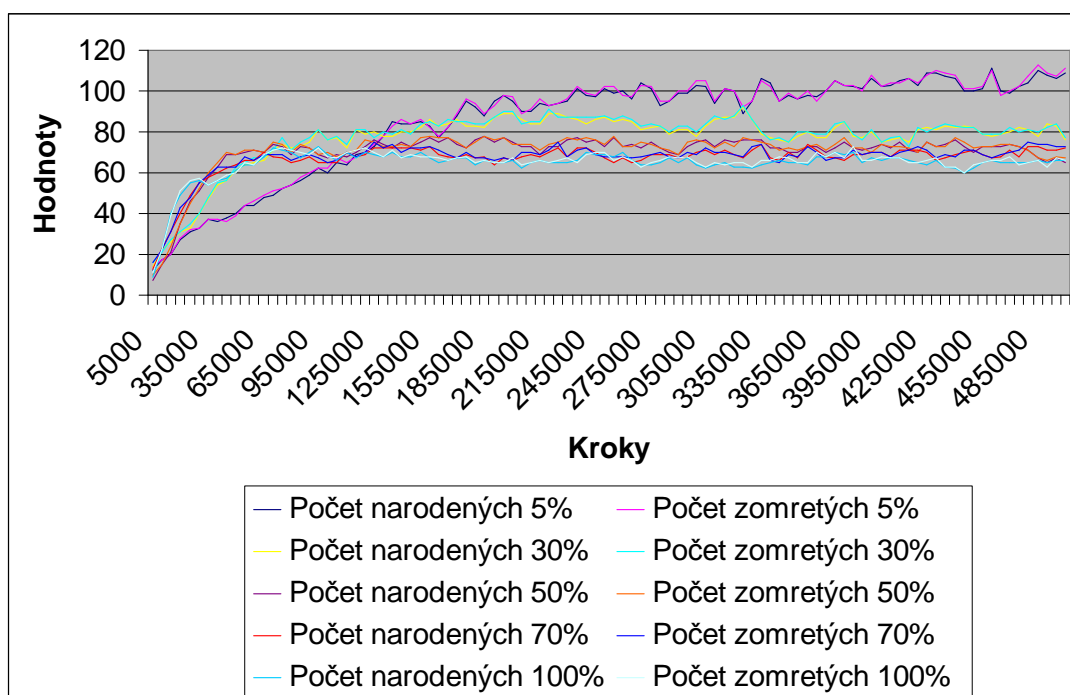
5.1.2 Výsledky prvého experimentu

Počet organizmov pre všetky simulácie kolíše okolo 900. Organizmy totiž obsadili všetky políčka s potravou, tých je na mape 712. Okrem toho sa na mape vyskytujú aj organizmy, ktoré sa pohybujú mimo potravu, tí však väčšinou neprežijú.

Simulácia s 5% pravdepodobnosťou mutácie má okolo 200 jednotiek energie na organizmus. Pre ďalšie štyri simulácie sa energia pohybuje okolo 300 jednotiek.

Vek organizmov sa pre všetky simulácie pohybuje okolo 50 rokov.

Pravdepodobnosť mutácie ovplyvňuje počet narodených a zomretých organizmov vid' Obrázok 5.3. Pri 5% mutácii rastie počet narodených a zomretých pomalšie ako pri vyšších mutáciách, ale populácia je dlhší čas progresívna, čo zabezpečuje populácii rýchlejšiu obmenu genetickej informácie. Populácie s vyššou pravdepodobnosťou mutácie nemusia, tak často obmieňať svoju genetickú informáciu, pretože ich genetická informácia mutuje viac. Preto dosiahnu určitý bod vo vývoji genofondu s menšou obmenou populácie, ako organizmy s nižšou pravdepodobnosťou mutácie, ktoré ju musia mať pre dosiahnutie tohto bodu vyššiu.



Obrázok 5.3: Beh prvého experimentu, graf vývoja počtu narodených a zomretých organizmov pre päť simulácií.

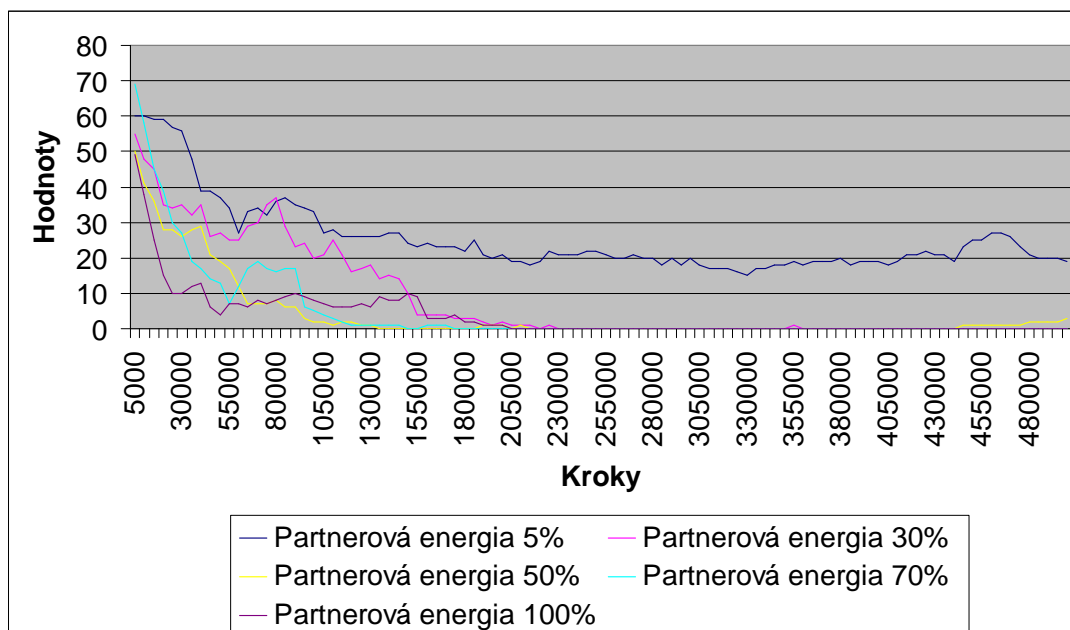
Vývoj množstva partnerovej energie je taktiež ovplyvnený mutáciou organizmov. Tento parameter udáva minimálne množstvo energie, ktoré musí mať partner, aby sa s ním organizmus vôbec chcel spáriť.

Vo všetkých simuláciách okrem prvej, kde je 5% pravdepodobnosť mutácie, majú organizmy okolo 300 jednotiek energie. Preto je tento parameter pre nich nepodstatný, a klesá k nule vid' Obrázok 5.4.

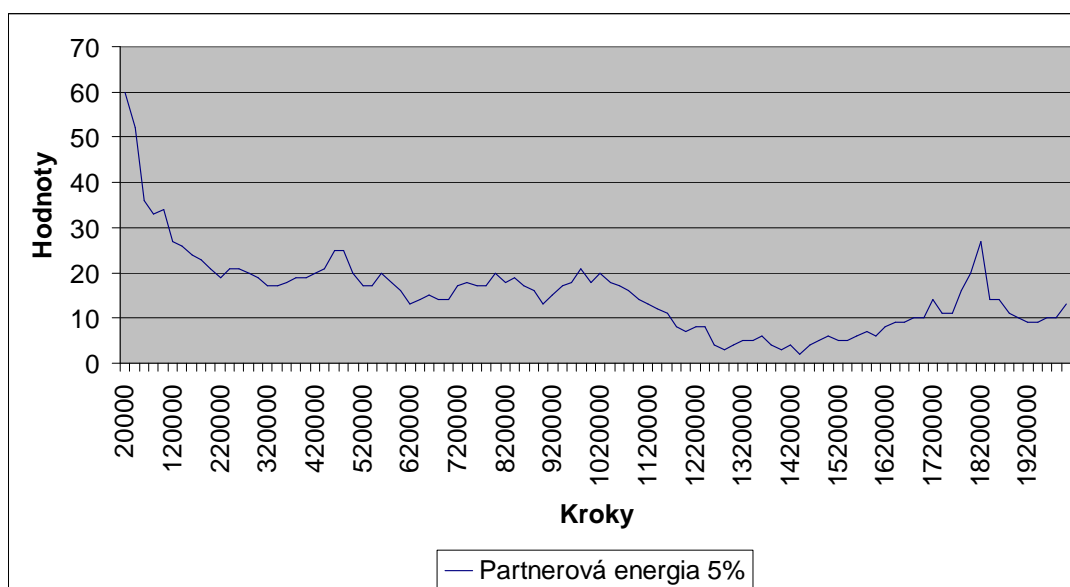
Pre simuláciu s 5% pravdepodobnosťou mutácie neklesá hodnota parametru k nule pre 500 000 krokov, preto bola simulácia spustená znovu na 2 000 000 krokov. Vývoj tohto parametru pre nové spustenie simulácie vid' Obrázok 5.5. Hodnota sa pohybuje medzi nulou a dvadsiatkou, ale nikdy sa neustáli. Posledných 500 000 krokov začala dokonca stúpať.

Ako sme si mohli všimnúť, tak vývoj tejto hodnoty je tým pomalší, čím menšie percento mutácie majú organizmy.

Možným vysvetlením toho, prečo simulácie s menšou mutáciou udržiavajú tento parameter dlhšie nad nulou je, že vzniká viac organizmov (ktoré ale nie sú schopné prežiť v danom svete), čím dochádza k častejšej obmene genotypu. Udržaním požiadavku na partnerovu energiu nad nulou by mohli zabezpečiť prežitie svojej populácie, a vytváranie jedincov s lepšími predpokladmi pre prežitie.

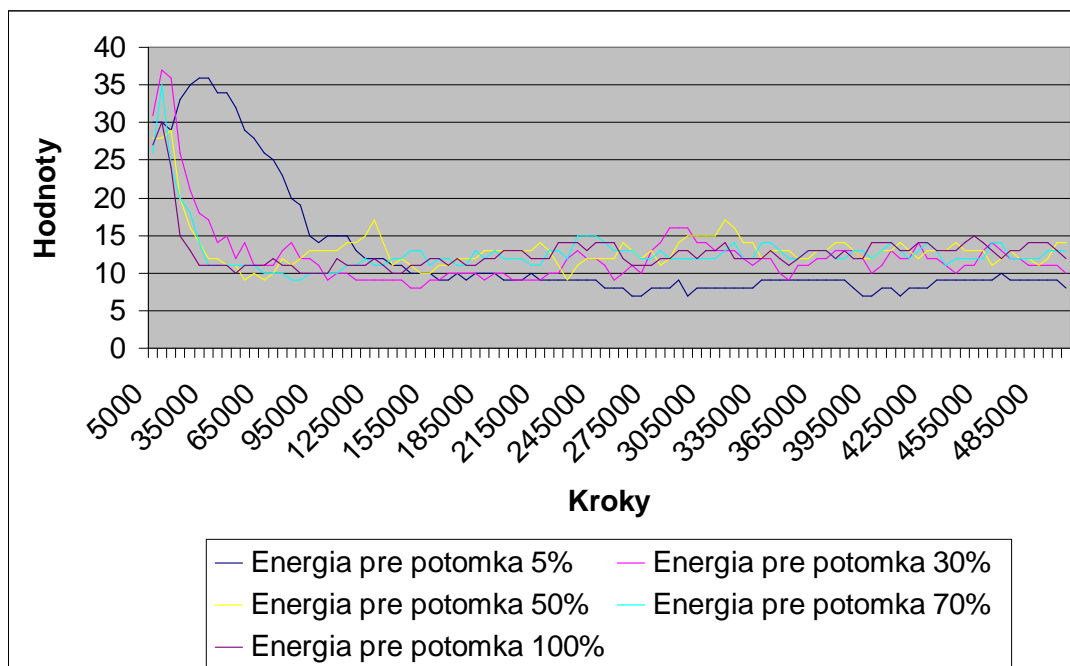


Obrázok 5.4: Beh prvého experimentu, graf vývoja partnerovej energie pre päť simulácií.



Obrázok 5.5: Beh prvého experimentu, graf vývoja partnerovej energie pre 5% mutáciu na 2 000 000 krokov.

Ďalším parametrom, ktorý je ovplyvnený mutáciou organizmov je množstvo energie predanej potomkovi. Matka pri narodení predá časť svojej energie potomkovi (ona pri párení získala určité množstvo energie od partnera). Vývoj tohto parametru vid' Obrázok 5.6. Pri porovnaní s vývojom počtu zomretých a narodených organizmov vid' Obrázok 5.3 zistíme, že ide o inverzný graf, čo sa týka usporiadania simulácií. Pri nižších mutáciách dávajú organizmy svojim potomkom menej energie ako pri vyšších mutáciách, čo je pravdepodobne spôsobené tým, že majú viac zomretých a narodených organizmov počas simulácie.



Obrázok 5.6: Beh prvého experimentu, graf vývoja množstva energie predanej potomkovi pre päť simulácií.

Maximálny vek partnera je ďalším parametrom, ktorého vývoj ovplyvnila mutácia organizmov. Tento parameter určuje maximálny vek partnera vhodného pre spárenie. Vývoj tohto parametru nie je pre všetky simulácie rovnaký. Parameter však stráca zmysel, keď presiahne hodnotu 140, pretože maximálny vek organizmov je 140 rokov (staršie organizmy nebudú v simulácii existovať). Pre všetky simulácie je spoločné to, že parameter na začiatku vystúpi nad 140, a už nikdy pod túto hodnotu neklesne. Z tohto hľadiska je už nepodstatné, čo sa deje s parametrom po prekročení 140, pretože to nemá žiaden vplyv na evolúciu organizmov. Preto môžeme povedať, že priebeh tohto parametru je pre všetky simulácie v podstate rovnaký. Graf vývoja tohto parametru nájdete na priloženom CD v súbore 1.experiment.xls.

5.1.3 Záver

Pre vyššie pravdepodobnosti mutácie je vývoj veľmi podobný. Rozdiely sme zaznamenali hlavne pre nižšie mutácie.

Pri nižších mutáciách je vývoj jednotlivých parametrov genotypu pomalší. Organizmy sa to snažia vyrovnáť častejšou obmenou genetickej informácie, ale nevyrovná sa to väčšej pravdepodobnosti mutácie.

To, aké pravdepodobnosti zvoliť, závisí od modelu, ktorý vytvárame. Na koľko to má zodpovedať realite, a akou rýchlosťou sa majú organizmy prispôbovať podmienkam sveta.

5.2 Druhý experiment – vplyv funkcie využitia energie vzhľadom k veku na vývoj populácie

Cieľom experimentu je zistiť, aký vplyv má krivka, ktorá určuje percento využitia energie z potravy, na vývoj populácie organizmov. Pre tie isté nastavenia vyskúšame rôzne krivky a budeme sledovať vývoj populácie (jej prežitie, počet organizmov k danému kroku, vek organizmov a energiu organizmov).

5.2.1 Počiatočné nastavenia

Táto funkcia vplýva rovnako na byľinožravé, mäsožravé a všežravé jedince. Taktiež vplýva rovnako na všetky druhy rastlín (keďže sa líšia len typom).

Mapa je rovnaká ako v prvom experimente. Pravdepodobnosť tvorby potravy je v jednotlivých spusteniach simulácie nastavená postupne na 30%, 50%, 80%. Obsahuje len jeden druh organizmov, ktorý sa živí žltou potravou. Neživí sa mäsom. Počiatočný počet organizmov je 700.

Cena pohybov:

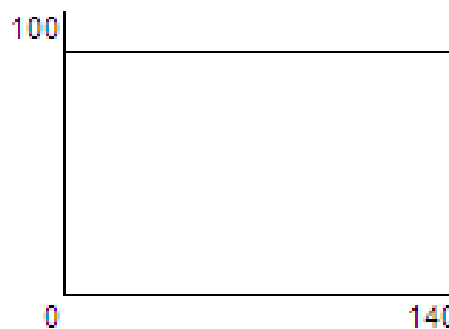
- Krok vpred: 9
- Otočenie vľavo/vpravo: 7
- Nič: 5

Parametre pre párenie sú rovnaké ako v prvom experimente. Maximálny vek organizmov je 140 rokov. Majú implicitný algoritmus. Pravdepodobnosť mutácie organizmov je 20%.

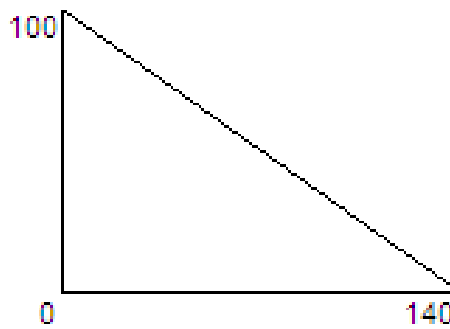
Pre jednotlivé simulácie sa bude meniť krivka funkcie, ktorá určuje percento využitia energie z potravy:



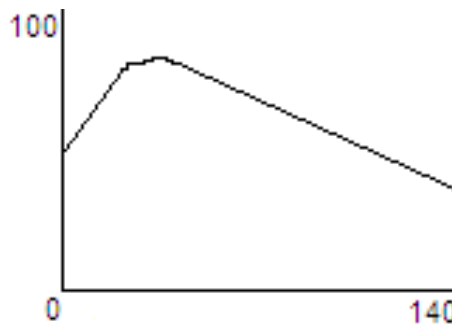
Obrázok 5.7: Pribeh prvej funkcie Energia vs. Vek v druhom experimente. Horizontálna os = vek: 0...140, vertikálna os = percento energie z potravy: 0...100.



Obrázok 5.8: Pribeh druhej funkcie Energia vs. Vek v prvom experimente. Horizontálna os = vek: 0...140, vertikálna os = percento energie z potravy: 0...100.



Obrázok 5.9: Priebek tretej funkcie Energia vs. Vek v prvom experimente. Horizontálna os = vek: 0...140, vertikálna os = percento energie z potravy: 0...100.



Obrázok 5.10: Priebek štvrtej funkcie Energia vs. Vek v prvom experimente. Horizontálna os = vek: 0...140, vertikálna os = percento energie z potravy: 0...100.

Simulácia je pre každú krivku a pravdepodobnosť tvorby potravy spustená na 500 000 krokov.

5.2.2 Výsledky druhého experimentu

Pred analýzou výsledkov je potrebné spomenúť, že posledná krivka by mohla najviac zodpovedať reálnej schopnosti získavať energiu z potravy pre žijúce organizmy. Je to však iba náš predpoklad.

Pre 30% pravdepodobnosť tvorby potravy neprežila ani jedna populácia. Populácie prežili iba pri 50% a 80% pravdepodobnosti tvorby potravy.

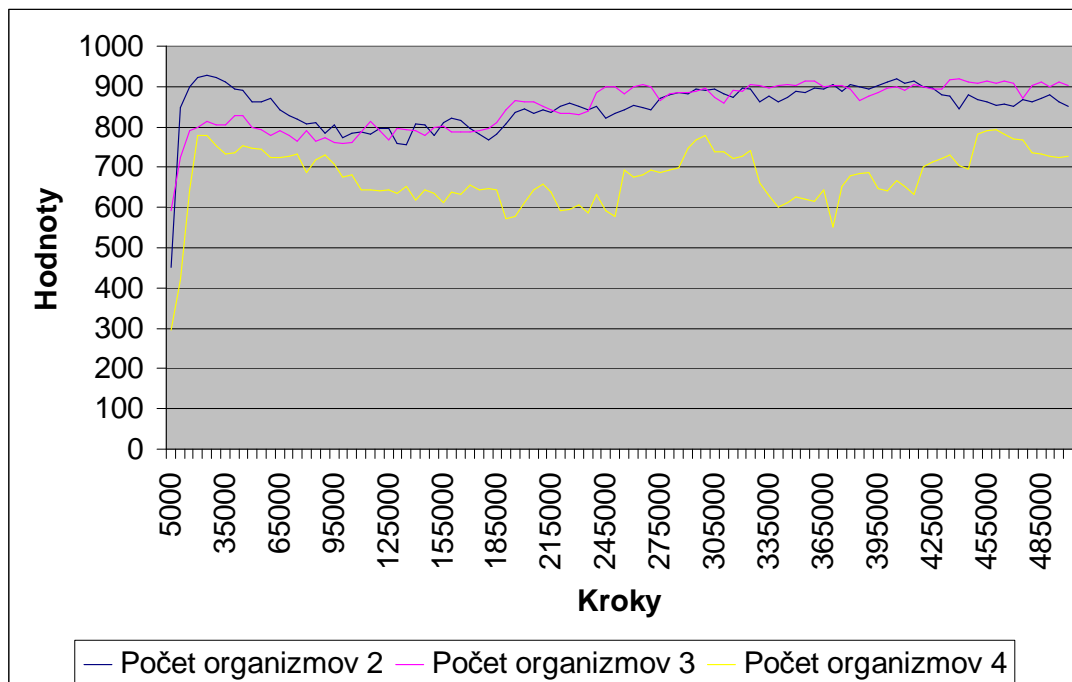
Pre prvú krivku neprežila ani jedna populácia. Pre posledné tri krivky prežili všetky populácie s 50% a 80% pravdepodobnosťou tvorby potravy.

Krivka funkcie pre využitie energie vzhľadom k veku výrazne ovplyvňuje počet organizmov v populácii pri 50% pravdepodobnosti tvorby potravy vid' Obrázok 5.11. Pre druhú krivku rastie počet organizmov veľmi rýchlo a udržiava si rovnakú strednú hodnotu. Podobné to je aj pre tretiu krivku. Prekvapujúco posledná krivka má strednú hodnotu oveľa menšiu ako predošlé dve. Tým, že organizmy v mladosti nezískavajú veľké percento energie, tak väčšinou prežijú len tie organizmy, ktoré sa v mladosti zdržiavajú na políčkach s potravou.

Vek organizmov je pre simulácie s 50% a 80% pravdepodobnosťou približne rovnaký, pohybuje sa okolo 50 rokov.

Energia organizmov, pri 50% pravdepodobnosti tvorby potravy, je pre tretiu a štvrtú krivku približne rovnaká. Je to preto, že organizmy s treťou krivkou majú v mladosti veľké percento získanej energie z potravy, takže si energiu nahromadia. Organizmy so štvrtou krivkou majú v mladosti menšie percento získanej energie, ale to sa potom zväčšuje, a organizmy začnú postupne hromadiť energiu. V strednom veku

majú organizmy so štvrtou krivkou väčšie percento získanej energie z potravy vzhľadom k veku, ako organizmy s treťou krivkou. Hodnoty pre druhú krivku sú výrazne nad nimi, ale to sa od takého vysokého percenta dalo očakávať.



Obrázok 5.11: Beh druhého experimentu, graf vývoja počtu organizmov pre tri simulácie s 50% pravdepodobnosťou tvorby potravy.

Počet organizmov sa pre druhú a tretiu krivku pri 80% pravdepodobnosti pohybuje okolo 1000. Pre štvrtú krivku sa pohybuje okolo 900.

Hodnoty energie na jeden organizmus sú vyššie ako pri 50% pravdepodobnosti. Pre všetky simulácie sa pohybujú okolo 500 jednotiek energie na organizmus. Najväčšiu hodnotu majú organizmy so štvrtou krivkou.

Grafy pre spomínané hodnoty sú na priloženom CD v súbore 2.experiment.xls.

5.2.3 Záver

Krivky určujúce percento využitia energie z potravy výrazne ovplyvňujú počet organizmov v populácii. Pokiaľ chceme mať väčší počet organizmov v populácii, tak v mladom veku musia mať vysoké percento zisku energie z potravy. Pokiaľ chceme menej organizmov, tak zvolíme nižšie percento zisku. Musíme však dať pozor na veľmi nízke hodnoty. Mapa v tomto experimente bola bohatá na potravu, ale pri prvej krivke neprežila populácia ani pri 80% pravdepodobnosti tvorby potravy.

Táto krivka výrazne ovplyvňuje priemernú energiu organizmov.

Rozdiely spôsobené touto krivkou sa dajú vyrovnáť zvýšením pravdepodobnosti tvorby potravy. Pokiaľ chceme mať náš model realistický, tak na začiatku simulácie nastavíme pravdepodobnosť tvorby potravy na vyššie percento, aby sa populácia udržala, a potom ju môžeme znížiť.

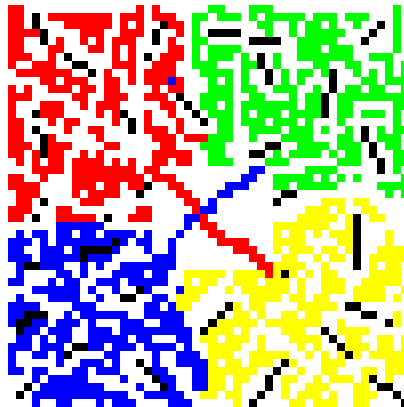
Chovanie organizmov pri „realistickej“ (štvrtej) krivke zodpovedá aj reálnemu vývoju organizmov vo svete. Mladé organizmy musia mať dostatok potravy, aby prežili, a mohli sa vyvíjať.

5.3 Tretí experiment - vývoj parametru určujúceho množstvo požadovanej energie od partnera a parametru určujúceho množstvo energie predanej partnerovi.

Cieľom experimentu je analyzovať vývoj týchto parametrov. O tom ako sa budú vyvíjať tieto parametre môžeme vysloviť určité predpoklady. Množstvo požadovanej energie, je to množstvo energie, ktoré musí predať „otec“ pri párení „matke“. Táto hodnota by mohla stúpať, pretože týmto spôsobom by mohli organizmy získavať energiu (organizmy nepredávajú celú túto energiu potomkovi, ale len jej časť). Množstvo energie predanej partnerovi, je množstvo energie, ktoré predá aktívny partner, ktorý urobí krok vpred (vyvolá párenie) pasívnemu partnerovi. Táto hodnota by mohla klesať, pretože týmto spôsobom organizmy prichádzajú o energiu. Budú sa tieto hodnoty vyvíjať podľa našich predstáv?

5.3.1 Počiatočné nastavenia

Mapa má rozmery 50x50 bodov. Pravdepodobnosť tvorby potravy je 50%.



Obrázok 5.12: Mapa sveta pre tretí experiment.

Sú dva druhy organizmov. Prvý sa živí zelenou a modrou potravou. Druhý sa živí červenou a žltou potravou. Počiatočný počet organizmov pre obidva druhy je 700.

Ceny pohybov sú pre obidva organizmy rovnaké:

- Krok vpred: 6
- Otočenie vľavo/vpravo: 4
- Nič: 3

Krivka funkcie, ktorá určuje využitie potravy vzhľadom k veku je rovnaká ako v prvom experimente.

Maximálny vek obidvoch druhov je 140 rokov a oba majú implicitný algoritmus.

Spoločné parametre pre párenie:

- Minimálny vek partnera: 10
- Maximálny vek partnera: 120
- Energia partnera: 60
- Energia pre potomka: 30

- Dĺžka tehotenstva: 10

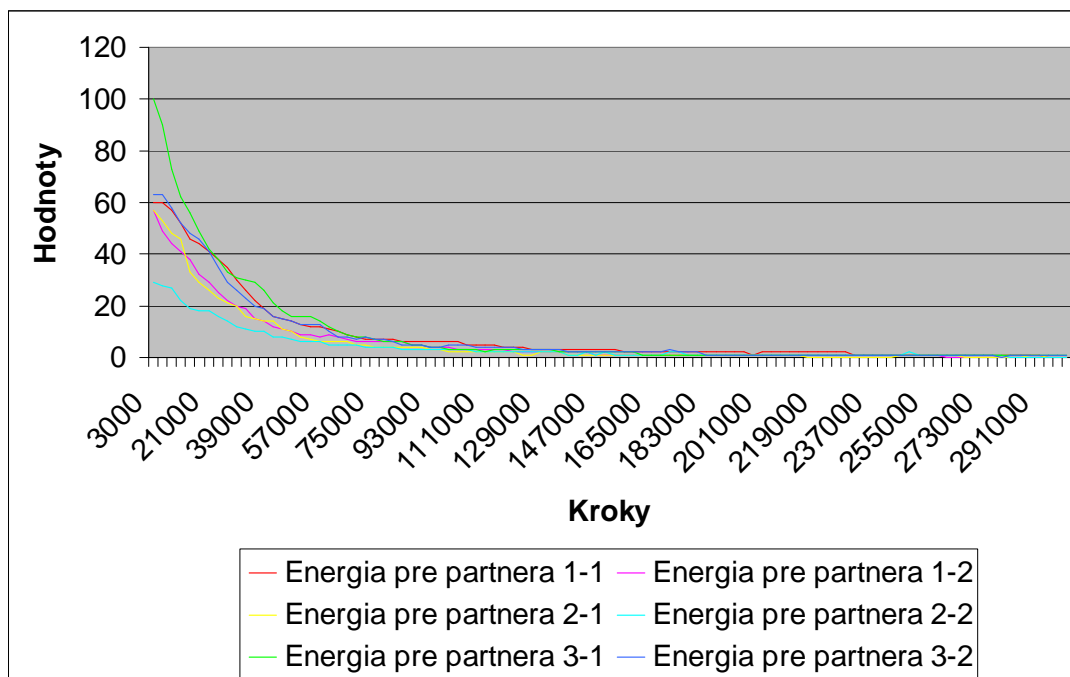
Odlíšné parametre pre jednotlivé spustenia:

- Prvá simulácia:
 - Požadovaná energia od partnera prvého druhu: 50
 - Požadovaná energia od partnera druhého druhu: 30
 - Energia predaná partnerovi prvého druhu: 60
 - Energia predaná partnerovi druhého druhu: 60
- Druhá simulácia:
 - Požadovaná energia od partnera prvého druhu: 10
 - Požadovaná energia od partnera druhého druhu: 10
 - Energia predaná partnerovi prvého druhu: 60
 - Energia predaná partnerovi druhého druhu: 30
- Tretia simulácia:
 - Požadovaná energia od partnera prvého druhu: 50
 - Požadovaná energia od partnera druhého druhu: 60
 - Energia predaná partnerovi prvého druhu: 100
 - Energia predaná partnerovi druhého druhu: 60

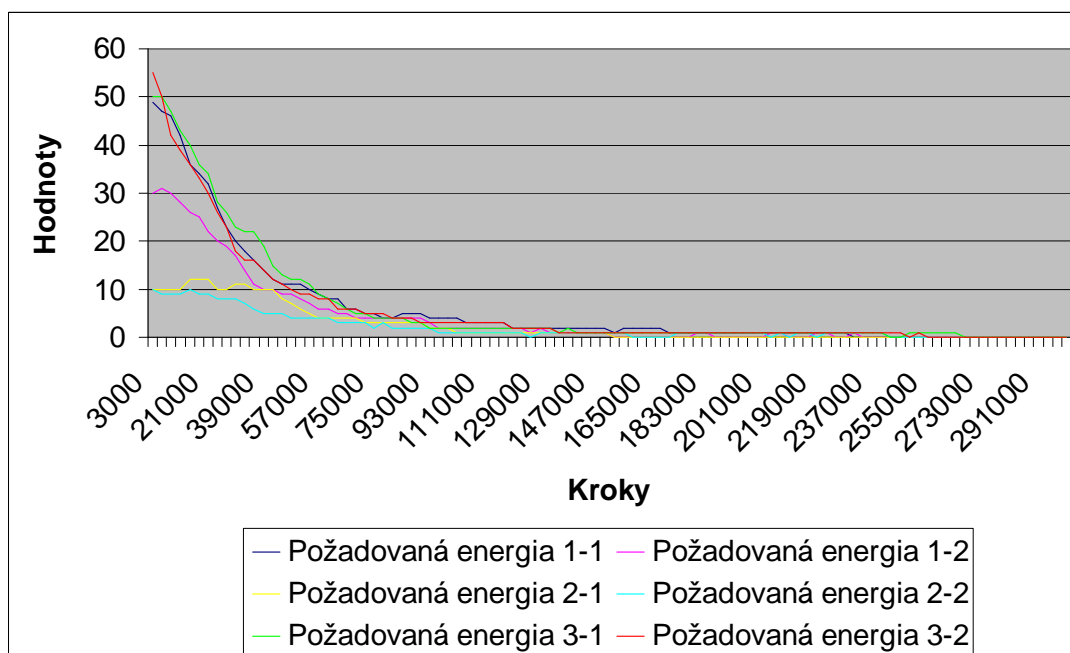
Pravdepodobnosť mutácie obidvoch druhov je 15%. Simulácia je pre jednotlivé hodnoty spustená na 300000 krokov.

5.3.2 Výsledky tretieho experimentu

Pre všetky simulácie majú oba parametre rovnaký vývoj. Náš predpoklad pre energiu predanú partnerovi sa potvrdil vid' Obrázok 5.13. Úplne opačný vývoj, ako sme predpokladali, mala požadovaná energia od partnera vid' Obrázok 5.14. Obidva parametre klesajú k nule, pretože „matka“ môže počas tehotenstva zbierať potravu, preto nepotrebuje od „otca“ energiu. Iné by to bolo, pokiaľ by tehotný organizmus nemohol zbierať potravu, ale to by vo väčšine prípadov nezodpovedalo skutočnosti. Požadovaná energia ovplyvňuje vývoj energie pre partnera. Pokiaľ je požadovaná energia pasívneho organizmu väčšia ako energia, ktorú je aktívny partner ochotný predať, tak sa organizmy nemôžu spáriť, takže takéto organizmy okamžite vyhynú. Ďalším dôvodom, prečo tieto hodnoty klesajú k nule, je to, že organizmy mali dostatok potravy, preto nepotrebovali energiu pri párení.



Obrázok 5.13: Beh tretieho experimentu, graf vývoja energie pre partnera, pre tri simulácie.



Obrázok 5.14: Beh tretieho experimentu, graf vývoja požadovanej energie, pre tri simulácie.

5.3.3 Pod experiment – klesajúca pravdepodobnosť potravy

Vyskúšame spustiť tie isté simulácie, ale budeme postupne znižovať pravdepodobnosť tvorby potravy. Organizmy budú mať na začiatku dosť potravy, aby sa udržali. Tým, že budeme pravdepodobnosti znižovať, tak sa bude znižovať aj množstvo energie organizmov, preto by mohli začať využívať skúmané parametre pri párení.

Pravdepodobnosť tvorby potravy budeme znižovať nasledovne:

- v kroku 50 000 na 45%
- v kroku 100 000 na 40%
- v kroku 150 000 na 35%
- v kroku 225 000 na 30%
- v kroku 300 000 na 28%
- v kroku 350 000 na 27%

Simulácie spustíme na 500 000 krokov, aby sme mali dostatok miesta pre postupné znižovanie pravdepodobnosti, a aby organizmy mali čas sa prispôbiť.

5.3.4 Výsledky pod experimentu

Ani pri menšom množstve energie na jeden organizmus sa vývoj týchto parametrov nezmenili. Na začiatku klesli k nule, a potom sa ich hodnota už nezmenila. Organizmy sa snažili prispôbiť novým podmienkam zmenou iných parametrov. Konkrétne sa pokúšali upraviť požiadavky na energiu partnera a množstvo energie predanej potomkovi. To, že zmena množstva potravy ovplyvní vývoj skúmaných parametrov, bol náš predpoklad. Zmenšenie energie organizmov, však nedonútilo organizmy meniť ich hodnoty. Pravdepodobne kvôli tomu, že prostredníctvom týchto parametrov prichádzajú organizmy o energiu, a parametre sú vo vzájomnom vzťahu. Organizmy sa neuchyľujú k ich zmenám ani pri zmenšení energie.

Grafy pre skúmané hodnoty nájdete na priloženom CD v súbore 3.experiment.xls.

5.3.5 Záver

Vývoj skúmaných parametrov je rovnaký pre optimálne podmienky na život, aj pre postupne sa zhoršujúce podmienky. Evolúcia v tejto simulácii nepoužívala tieto parametre pre vývoj genofondu organizmov, pretože pasívny partner (matka) môže zbierať potravu počas tehotenstva, a tiež preto, lebo tým organizmy prichádzajú o energiu. Tieto parametre by mohli mať pravdepodobne zmysel v evolúcii, kde dochádza k zložitejším zmenám prostredia, a kde sú organizmy ovplyvňované viacerými faktormi.

5.4 Štvrtý experiment – analýza a vývoj algoritmu pre jeden stav organizmu

Cieľom experimentu je zanalyzovať algoritmus pre jeden stav organizmu. Algoritmy pre jeden stav organizmu majú rovnakú štruktúru. Podľa okolia organizmu (čo je pred organizmom, za ním, vľavo a vpravo od organizmu) sa definuje jeho pohyb a prechod do ďalšieho stavu (prípadne do toho istého stavu). Pre niektoré kombinácie tohto okolia môžeme predpokladať, aký pohyb by vykonal skutočný organizmus, aby prežil, čo najdlhšie. Napríklad, ak je vedľa organizmu nebezpečný organizmus, tak by sa mal pohnúť dopredu. Podľa výsledkov analýzy budeme percentuálne hodnotiť algoritmus organizmov. Na štyroch simuláciách vyskúšame, ako sa vyvíja algoritmus organizmov.

5.4.1 Počiatočné nastavenia

Mapa má rozmery 73x100 bodov. Pre prvú a druhú simuláciu je pravdepodobnosť tvorby potravy 50%. Pre tretiu a štvrtú simuláciu sa bude pravdepodobnosť tvorby potravy postupne znižovať ako v pod experimente tretieho experimentu.



Obrázok 5.15: Mapa sveta pre štvrtý experiment.

Pre prvú a tretiu simuláciu sú dva druhy organizmov. Počiatočný počet organizmov prvého druhu je 1200 a druhého druhu 1600 organizmov. Prvý druh sa živí žltou a červenou farbou. Okrem toho môže pojeďať iné organizmy. Jeho limit hladu je 10 a spotrebúva potravu z organizmov 1,5-krát rýchlejšie ako z rastlín. Druhý druh sa živí zelenou a modrou farbou.

Schopnosť zabíjať je pre obidva druhy 2.

Ceny pohybov pre obidva druhy:

- Krok vpred: 6
- Otočenie vľavo/vpravo: 4
- Nič: 3

Krivka funkcie, ktorá určuje využitie potravy vzhľadom k veku je pre všetky organizmy rovnaká ako v prvom experimente (viď Obrázok 5.2).

Maximálny vek všetkých organizmov je 140 rokov. Všetky organizmy majú implicitný algoritmus.

Parametre pre párenie majú všetky organizmy rovnaké:

- Minimálny vek partnera: 10
- Maximálny vek partnera: 120
- Energia partnera: 60
- Energia pre potomka: 30
- Energia pre partnera: 35
- Požadovaná energia od partnera: 30
- Dĺžka tehotenstva: 10

Pravdepodobnosť mutácie organizmov je pre všetky druhy 15%. Pre druhú a štvrtú simuláciu sú taktiež dva druhy organizmov. Sú rovnaké ako pre predošlé dve simulácie, len prvý druh sa neživí inými organizmami.

Všetky simulácie sú spustené na 500 000 krokov.

5.4.2 Analýza algoritmu pre jeden stav organizmu

Nebudeme tento algoritmus rozoberať úplne do podrobností, pretože je veľa možností a podmienok, na ktorých závisí pohyb organizmov. Preto rozoberieme algoritmus len podľa základných kritérií. Organizmus sa riadi podľa štyroch políčok

(čo je od organizmu vľavo, vpravo, pred ním a za ním) a môže urobiť len štyri rôzne pohyby (ísť vpred, otočiť sa vľavo/vpravo, neurobiť nič). Rozoberieme možnosti podľa priorít.

Pred organizmom

Najdôležitejšie je to, čo sa nachádza pred organizmom, pretože tam môže hneď ísť.

Pokiaľ sa tam nachádza nebezpečný organizmus, tak je dôležité, či skúmaný organizmus môže pojedať iné organizmy, a či je jeho energia pod limitom hladu. Ak áno, tak nebezpečný organizmus pred ním je potenciálny zdroj energie. Mohlo by však byť podstatné koľko má skúmaný organizmus energie (či vôbec potrebuje ďalšiu energiu), pretože je tu riziko, že môže zomrieť v boji. Čo je však dostatok energie závisí od nastavení sveta a jednotlivých druhov. Nebudeme sa energiou zaoberať, a zvolíme možnosť, že energia je potrebná vždy. Preto je preňho najlepšie urobiť krok vpred. Pokiaľ sa nemôže živiť inými organizmami alebo má energiu nad limitom hladu, tak je to zbytočný boj a najlepšie je sa otočiť (smerom, kam sa môže pohnúť), a nečakať ticho na boj.

Ak je pred organizmom stena alebo obyčajný organizmus, tak by sa mal taktiež otočiť smerom, kam sa môže pohnúť.

Ak je pred ním vhodný organizmus alebo jeho potrava, tak by mal spraviť krok vpred.

Ak je pred ním organizmus a on sa živí inými organizmami, tak by mal ísť vpred.

Vľavo/vpravo od organizmu

Pokiaľ sa nachádza na niektorých zo strán nebezpečný organizmus a vpredu nie je stena, tak by mal skúmaný organizmus ísť dopredu. Ak tam je stena, tak by sa mal otočiť smerom, kde sa nenachádza nebezpečný organizmus. Rovnaký postup platí aj pre skúmaný organizmus, ktorý sa živí inými organizmami a jeho energia je nad limitom hladu. Pokiaľ je však jeho energia pod limitom hladu a je pred ním stena, tak by sa mal otočiť k nebezpečnému organizmu.

Pokiaľ sa na niektorej strane nachádza vhodný organizmus alebo potrava skúmaného organizmu, a pred skúmaným organizmom nie je nič výhodnejšie, tak by sa mal otočiť smerom k organizmu alebo potrave. Tu nastáva zaujímavé dilemma, čo má robiť skúmaný organizmus, ak je na jednej strane vhodný organizmus, a na druhej strane je jeho potrava. Vhodný organizmus mu môže utiecť, ale aj potrava mu môže zahynúť, alebo ju môže zjesť niekto iný. Preto je jedno, ako sa otočí.

Za organizmom

Ak je za skúmaným organizmom nebezpečný organizmus, tak by mal ísť dopredu. Ak je tam stena, tak by sa mal otočiť smerom, kam sa môže pohnúť. Táto stratégia je výhodná, aj keď môže skúmaný organizmus požírať iné organizmy, pretože je veľká pravdepodobnosť, že mu organizmus utečie.

Ak je za skúmaným organizmom vhodný organizmus alebo jeho potrava, tak je veľmi veľká pravdepodobnosť, že organizmus utečie a potrava zahynie, alebo ju zje niekto iný, preto sú tieto políčka za organizmom nepodstatné.

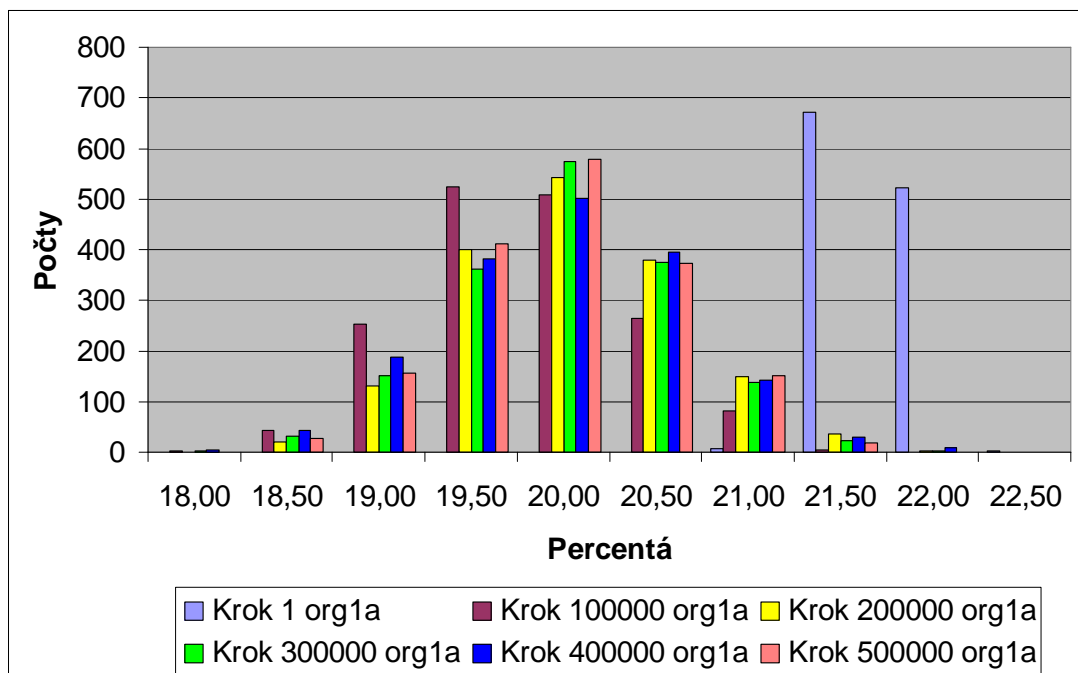
Súhrn

Najvyššiu prioritu má nebezpečný organizmus, pretože stret s ním sa môže rovnať smrti. Ako ďalšie sú vhodný organizmus a potrava skúmaného organizmu. Napokon to je vyhnutie sa prekážkam ako je stena alebo iný organizmus.

Program umožňuje spočítať percentuálnu zhodu jedného stavu algoritmu organizmu s týmito pravidlami. Tieto percentá sa následne vypíšu v histogramových štatistických údajoch. Tento výpis je možné vypnúť pomocou konfiguračnej položky `statistic.writeAlgoInfo`, ktorá sa nachádza v súbore `config\config.txt`. Taktiež je možné nastaviť výpis políček algoritmu, ktoré sa nezhodujú s hore popísanými pravidlami. Tento výpis je možné nastaviť pre organizmy, ktorých percentuálna zhoda prekročí určitú hranicu. Táto hranica je definovaná v konfiguračnej položke `statistic.writeBadOverPercent`.

5.4.3 Výsledky štvrtého experimentu

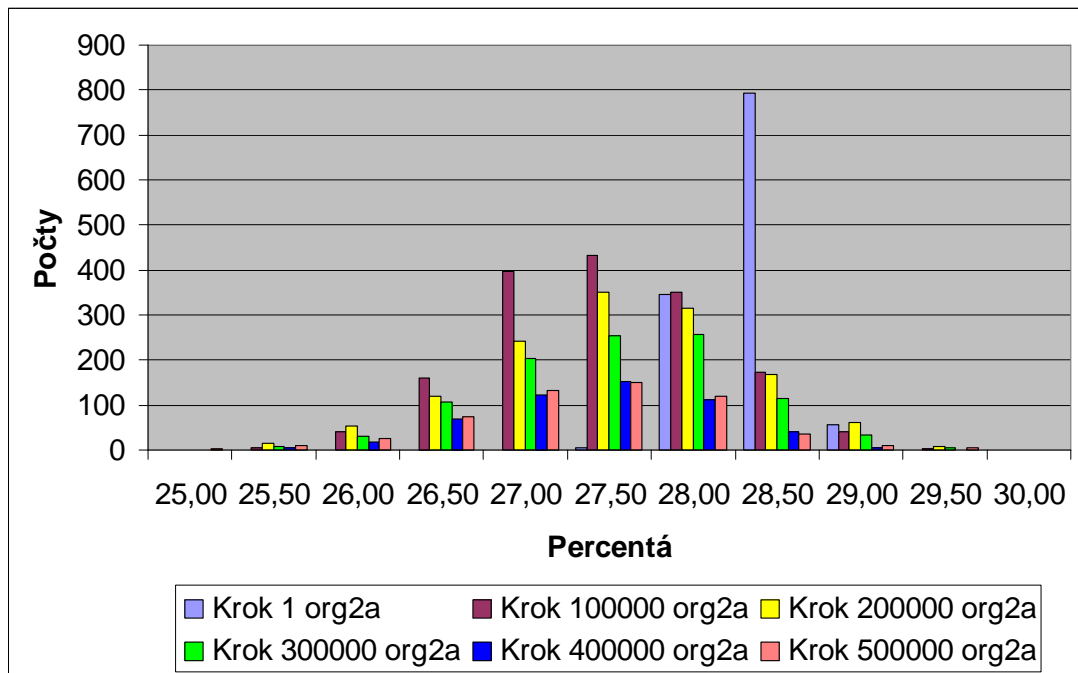
Populácie oboch druhov v tretej simulácii (klesajúca pravdepodobnosť tvorby potravy, jeden druh živiaci sa inými organizmami) vyhynuli okolo kroku 450 000. Vo štvrtej simulácii (klesajúca pravdepodobnosť tvorby potravy, ani jeden druh sa neživí inými organizmami) vyhynula populácia druhého druhu niekedy po kroku 400 000. Ostatné simulácie vykonali celých 500 000 krokov. Výsledky sú rovnaké pre simulácie s 50% pravdepodobnosťou tvorby potravy a pre simulácie s klesajúcou pravdepodobnosťou tvorby potravy. Klesajúca pravdepodobnosť ovplyvnila len počet organizmov, ktorý postupne klesal. Percentá hodnotenia algoritmov sú pre všežravé organizmy nižšie ako pre organizmy, ktoré sa neživili inými organizmami. Ich hodnotenie je totiž rozdielne vid' Analýza algoritmu pre jeden stav organizmu.



Obrázok 5.16: Beh štvrtého experimentu, prvá simulácia, prvý druh organizmov. Histogram zobrazujúci počet organizmov pre jednotlivé percentá v určitých krokoch simulácie. Percentá predstavujú zhodu algoritmu organizmov s našou analýzou.

Algoritmus organizmov sa v priebehu simulácie ustálil okolo jednej hodnoty. Jeho vývoj neovplyvnili ani zhoršené podmienky pre život organizmov. Naša analýza bola uskutočnená na základe domnienok, ktoré vychádzali zo správania skutočných organizmov. Organizmy v tejto simulácii sa však správajú inak, pretože ich genotyp a podmienky pre život sú značne zredukované. Takisto svet, v ktorom žijú nezodpovedá úplne realite, pretože jeho veľkosť je obmedzená (nie je dost' pružný). Výsledky prvej simulácie prvého druhu vid' Obrázok 5.16, štvrtej simulácie prvého druhu vid' Obrázok 5.17.

Výsledky ostatných experimentov nájdete na priloženom CD v adresári experimenty.



Obrázok 5.17: Beh štvrtého experimentu, štvrtá simulácia, prvý druh organizmov. Histogram zobrazujúci počet organizmov pre jednotlivé percentá v určitých krokoch simulácie. Percentá predstavujú zhodu algoritmus organizmov s našou analýzou.

6 Záver

Cieľom tejto bakalárskej práce bolo vytvoriť program pre simuláciu evolúcie umelého života a uskutočniť niekoľko experimentov. Výsledkom práce je program SofE a štyri experimenty vid' Experimenty. Samotná simulácia a organizmy majú v tejto aplikácii množstvo parametrov, ktoré ovplyvňujú ich chovanie. Bolo by vhodné urobiť experiment pre každý parameter, ktorý môže ovplyvniť život organizmov. Rozsah práce je však obmedzený určitou hranicou, preto sme uskutočnili iba experimenty, ktoré ukazujú možnosti tohto programu a testujú vplyv základných parametrov na evolúciu.

6.1 Budúcnosť programu SofE

Program SofE by mohol byť v budúcnosti rozšírený o nové implementácie organizmov. V nových implementáciách by vytvorenie organizmov s jednoduchším algoritmom, umožnilo porovnať ich efektívnosť a vplyv na evolúciu organizmov.

Na základe uskutočnených experimentov by bolo vhodné upraviť spôsob chovania simulácie a zmeniť niektoré parametre organizmov.

Literatúra

- [1] Heller P., Matoušek V., 2009, Prezentácie k prednáške Umělý život, <http://www.kiv.zcu.cz/studies/predmety/uir/predn/P3/FThema3.pdf>.
- [2] Mark A. Bedau, 2003, „Artificial life: organization, adaptation and complexity from the bottom up“. TRENDS in Cognitive Sciences, <http://people.reed.edu/~mab/publications/papers/BedauTICS03.pdf>.
- [3] Dawkins R., 1991, „Slepý hodinář: Zázrak života očima evoluční biologie“, Paseka, ISBN 80-7185-445-X.
- [4] Ventrella J., GenePool: Exploring The Interaction Between Natural Selection and Sexual Selection, <http://www.ventrella.com/Alife/GenePool.pdf>.
- [5] Holan T., 2004, Jiné programování. In Vojtáš, Peter. ITAT 2004 Information Technologies – Applications and Theory. Košice: UPJŠ, 2004. s.139-148, <http://ksvi.mff.cuni.cz/~holan/jinak/itat.pdf>.
- [6] Knihovna wxWidgets, <http://www.wxwidgets.org/>.
- [7] GNU Transalton Project, 1998, <http://www.gnu.org/software/gettext/>.

Prílohy

Príloha A - Užívateľská dokumentácia

A.1 Inštalácia programu

Program SofE je napísaný pre operačný systém Windows. Bol testovaný na operačných systémoch Windows XP a Windows Vista.

Program je dodávaný v podobe inštalačného balíku *SofESetup.exe*. Ten nájdete na priloženom CD.

Všetko potrebné spraví samotný sprievodca, ktorého spustíme súborom SofESetup.exe. Môžeme si zvoliť umiestnenie, kam chceme inštalovať program. Je však potrebné, aby mohol program v inštalačnom adresári vytvárať súbory a adresáre. Sprievodca Vám vytvorí zástupcu na ploche a v ponuke Štart. Pre správne fungovanie programu sú potrebné všetky adresáre a súbory, ktoré sprievodca nainštaluje.

A.2 Spustenie programu

Program spustíme súborom SofE.exe, ktorý sa nachádza v inštalačnom adresári. Pokiaľ chceme používať grafické rozhranie programu, tak spustíme program bez parametru. Ak chceme spustiť len simuláciu/-ie, tak ako parameter uvedieme cestu k súboru s nastaveniami skriptu (viď Skript).

Program povoľuje spustiť len jednu inštanciu, ale podporuje rozhranie MDI, čím nám dovoľuje pracovať s viacerými simuláciami v grafickom rozhraní. V skriptovacom súbore máme tiež možnosť uviesť viac simulácií, ktoré budú spustené súčasne.

Pri spustení programu pomocou skriptovacieho súboru sa nám nezobrazí žiadne okno. Simulácie sú spustené na pozadí, a vypisujú len štatistické údaje. Pri spustení programu s grafickým rozhraním sa otvorí hlavné okno aplikácie. Pre vytvorenie novej simulácie vyberieme v menu *Súbor* záložku *Nová Simulácia*. Pre nahratie uloženej simulácie/-í vyberieme v menu *Súbor* záložku *Otvoriť simuláciu* alebo *Otvoriť simulácie*.

Program podporuje viac jazykov. Zatiaľ existuje preklad do anglického a slovenského jazyka. Po kliknutí na záložku *Jazyk* v menu *Súbor* sa zobrazí dialóg pre výber jazyka aplikácie.

Pre zoznámenie sa s programom nám slúži nápoveda, ktorá je uložená v súbore *doc.chm*. Ten sa nachádza v podadresári inštalačného adresára *doc*. Pre zobrazenie tejto nápovedy v programe, ktorý je spustený v grafickom rozhraní, použijeme klávesu F1.

Pre rýchle vyskúšanie programu môžeme použiť nastavenia pre uskutočnené experimenty (viď Experimenty). Tie sa nachádzajú v adresároch uložených v adresári *SimSettings*.

A.3 Dialóg pre novú simuláciu

Po kliknutí na záložku *Nová Simulácia* v menu *Súbor* sa nám zobrazí dialóg pre nastavenie parametrov pre novú simuláciu viď Obrázok A.1.

Pomocou tohto dialógu si môžeme ukladať a načítať nastavenia jednotlivých častí simulácie (viď Súbor s nastaveniami), čo nám uľahčí opätovné spustenie s rovnakými nastaveniami. K uloženiu nám slúžia tlačidlá v lište nástrojov (toolbar). Tlačidlá s písmenom L (Load) v názve slúžia pre uloženie, a tlačidlá s písmenom S (Save) slúžia pre načítanie nastavení. Po kliknutí na tlačidlo sa nám zobrazí štandardný dialóg pre uloženie/načítanie súboru. Po načítaní nastavení sa aktualizujú políčka dialógu.

Obrázok A.1: Dialóg pre novú simuláciu

- *SIM L* – načítanie súboru s nastaveniami pre simuláciu
- *SIM S* – uloženie súboru s nastaveniami pre simuláciu
- *ORG L* – načítanie súboru s nastaveniami pre druh organizmov, ktorý máme označený v listboxe *Druhy organizmov*. Pokiaľ nie je žiaden druh označený,

tak sa vytvorí nový, s implicitným názvom „Organizmus x“, kde x je poradové číslo implicitne vytvoreného názvu.

- *ORG S* – uloženie súboru s nastaveniami pre označený druh organizmov.
- *ALG L* – pre označený druh organizmu sa načíta nový algoritmus. Keď sa vytvorí nový druh organizmu, tak má implicitný algoritmus, ktorý je uložený v súbore *Def_alg.txt*.
- *ALG S* – uloženie algoritmu označeného druhu organizmov do súboru.

Pre ukladanie súborov s nastaveniami je vytvorený špeciálny podadresár v inštalačnom adresári *SimSettings*.

V hornej časti dialógu máme možnosť načítať svet organizmov, a nastaviť množstvo krokov, ktoré zodpovedá jednému roku.

Zvyšok dialógového okna slúži na nastavenie parametrov pre druhy organizmov.

Máme dve možnosti, ako vytvoriť nový druh organizmov. Buď načítaním nastavení pre druh organizmov, v prípade že nemáme žiaden druh označený v listboxe *Druhy organizmov*, alebo napísaním názvu do editboxu pod týmto listboxom a jeho pridaním do zoznamu druhov.

Pre daný druh organizmov môžeme vybrať druhy potravy, ktorými sa bude živiť. Pomocou comboboxu vyberieme daný druh potravy, a vpravo zaškrtneme checkbox.

V jednotlivých políčkach máme možnosť zadať počiatočné hodnoty parametrov pre daný druh. Hodnoty ceny pohybov, maximálneho veku organizmov daného druhu a počtu stavov algoritmu sa počas simulácie nemenia. Takisto zoznam potravín a krivka Energia vs. Vek počas simulácie ostávajú rovnaké. Pravdepodobnosť mutácie organizmov daného druhu môžeme meniť iba prostredníctvom grafického rozhrania. Ostatné parametre pre druh organizmov sa na začiatku simulácie zmutujú pre každý vytvorený organizmus, a počas simulácie sa menia pri párení alebo boji.

A.4 Dialóg pre pravdepodobnosť spárenia dvoch rôznych druhov organizmov

Po nastavení všetkých parametrov v dialógu pre novú simuláciu a kliknutí na tlačidlo *Ok* sa zobrazí dialóg pre nastavenie pravdepodobnosti spárenia dvoch rôznych druhov organizmov. Na výber máme všetky možné druhy organizmov, aj tie, ktoré sme nevytvorili, pretože pri spárení rôznych druhov môžu vzniknúť nové druhy organizmov. Pre zadanie pravdepodobnosti spárenia musíme vybrať dané druhy v comboboxoch a v editboxe zadať pravdepodobnosť ich spárenia. Po kliknutí na tlačidlo *Ok* sa zobrazí okno simulácie.

A.5 Okno simulácie

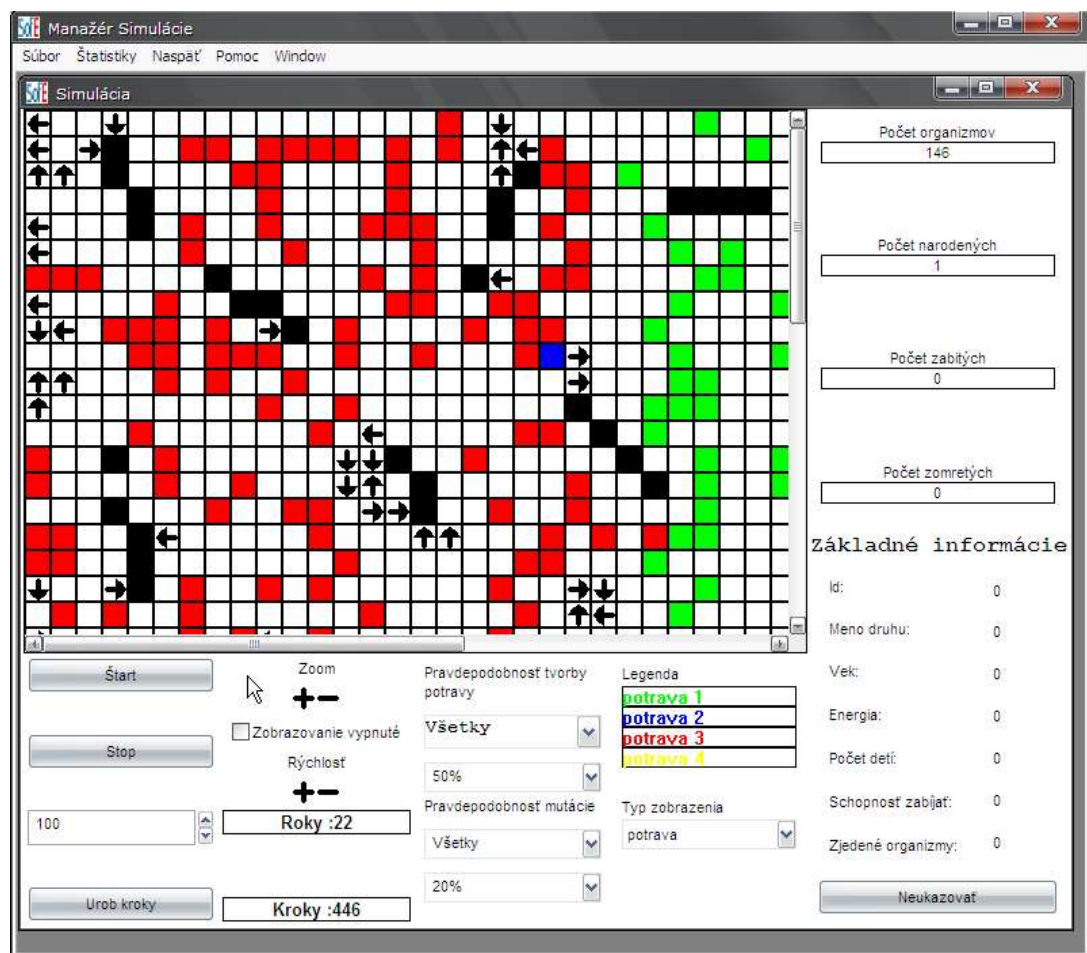
V tomto okne sa nám zobrazuje simulácia a my s ňou môžeme pracovať.

Pomocou záložiek v menu *Súbor* môžeme ukladať aktuálnu simuláciu/-e alebo ich načítať. Spolu so simuláciou sa ukladajú aj pomocné súbory a rastrový obrázok krivky Energia vs. Vek, preto sa pri ukladaní používajú adresáre. Súbor s uloženou simuláciou má rovnaké meno ako adresár. Program poskytuje dve možnosti ako uložiť simuláciu/-e. Prostredníctvom záložiek *Uložiť simuláciu* a *Uložiť všetko* sa uloží simulácia/-e priamo. Pokiaľ budeme ukladať simuláciu/-e prvýkrát, tak program uloží simuláciu pod názvom *Simulácia*. Pokiaľ daný adresár už existuje, tak sa nakoniec doplní číslo. Pri každom ďalšom ukladaní sa projekt uloží do tohto adresára. Pokiaľ chceme uložiť simuláciu/-e pod iným menom, tak môžeme použiť

záložky *Uložiť simuláciu ako* a *Uložiť všetko ako*. Po ich odkliknutí sa nám zobrazí štandardný dialóg pre výber adresára.

V ľavom hornom rohu je zobrazený svet simulácie. Organizmy sú kreslené dvoma rôznymi tvarmi. Pomocou šípky sa zobrazujú organizmy, ktoré sa neživia inými organizmami, naopak „všežravé“ organizmy sa kreslia pomocou trojuholníka. Čierne políčka predstavujú steny, cez ktoré nemôžu organizmy prechádzať. Biele políčka predstavujú miesta, kde nie je potrava. Farebné štvorcíky predstavujú potravu (rastlinu) daného druhu. Máme možnosť si tento svet približovať pomocou tlačidiel + a – pod nápisom *Zoom*.

Simuláciu môžeme spustiť na neurčitý počet krokov pomocou tlačidla *Štart*, a zastaviť ju potom pomocou *Stop*, alebo zadáme v spinbuttone počet krokov, a spustíme simuláciu kliknutím na tlačidlo *Urob kroky*. Môžeme vypnúť prekresľovanie sveta simulácie pomocou checkboxu *Zobrazovanie vypnuté*.



Obrázok A.2: Okno simulácie

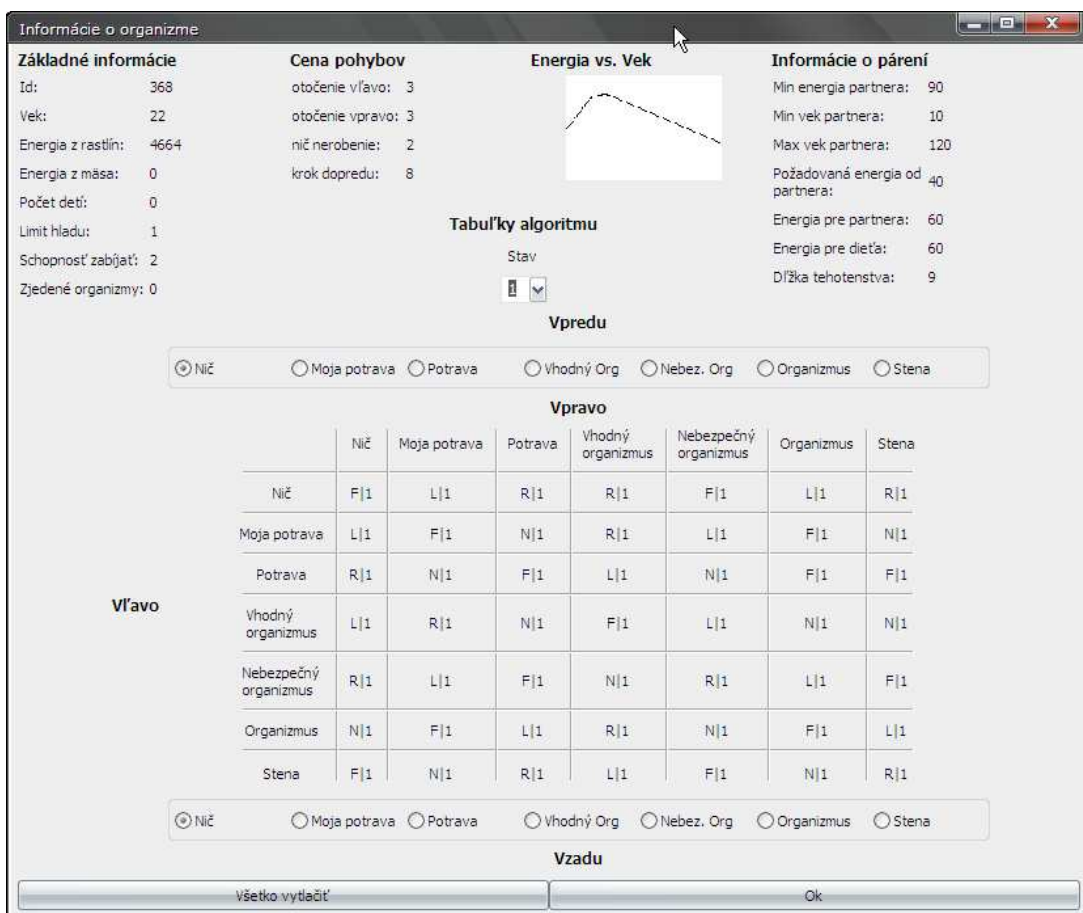
Máme možnosť meniť pravdepodobnosť tvorby rastlín jednotlivých druhov a pravdepodobnosť mutácie daných druhov organizmov pomocou comboboxov, ktoré sa nachádzajú v dolnej časti okna. Pre danú pravdepodobnosť sa nachádzajú v okne dva comboboxy. V jednom sa zobrazujú názvy druhov, a v druhom sa pre vybraný druh zobrazujú konkrétne pravdepodobnosti. Pokiaľ chceme zmeniť pravdepodobnosť, tak do druhého comboboxu napíšeme novú hodnotu a stlačíme klávesu *Enter*.

Program nám dovoľuje zmeniť zobrazenie sveta. V comboboxe *Typ zobrazenia* si môžeme vybrať z týchto možností:

- potrava – zobrazí organizmy, a na voľných políčkach zobrazí farebne potravu.
- druhy – zobrazí len farebne odlíšené druhy organizmov.
- vek – zobrazí farebne odlíšené organizmy podľa ich veku.
- energia – zobrazí farebne odlíšené organizmy podľa ich energie.
- počet detí – zobrazí farebne odlíšené organizmy podľa počtu ich detí.
- počet zjedených organizmov – zobrazí farebne odlíšené organizmy podľa počtu organizmov, ktoré zjedli.
- Schopnosť zabíjať – zobrazí farebne odlíšené organizmy podľa ich schopnosti zabíjať.

Pre každé z týchto zobrazení sa nad comboboxom vykreslí legenda.

Program vypisuje v pravom hornom rohu základné informácie o počtoch organizmov pre jeden krok simulácie.



Obrázok A.3: Dialóg s informáciami o organizme

V menu Štatistiky si môžeme vybrať medzi štyrmi druhmi štatistických údajov (viď Štatistické údaje). Po kliknutí na danú záložku si vyberieme súbor pre ich uloženie. Pre ukladanie súborov so štatistickými údajmi je vytvorený špeciálny podadresár v inštaláčnom adresári *Statistics*. Hlavička údajov sa pre agregované

a histogramové štatistické údaje prekladá podľa jazyka aplikácie. Pre krokové a ročné štatistické údaje sa vypíše hlavička v jazyku, v akom sa spustila aplikácia.

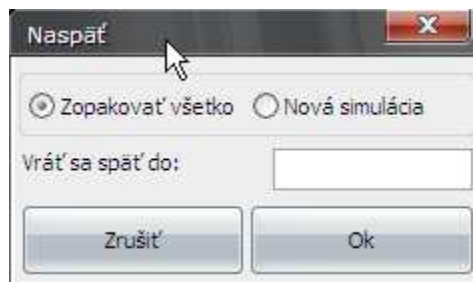
Ak chcem sledovať základné parametre niektorého organizmu, tak musíme na daný organizmus kliknúť ľavým tlačidlom myši. Potom sa nám budú v pravom dolnom rohu okna zobrazovať jeho základné aktuálne parametre. Okrem toho bude obrázok sledovaného organizmu na mape zväčšený.

Pre dôkladnejšie skúmanie organizmu klikneme pravým tlačidlom na organizmus. Zobrazí sa dialóg takmer so všetkými parametrami organizmu vid' Obrázok A.3. Hodnoty týchto parametrov sa však neaktualizujú. Sú to hodnoty, ktoré mal organizmus v čase zobrazenia tohto dialógu. V tomto dialógu si môžeme prezerať algoritmus organizmu. V comboboxe vyberieme stav organizmu, a tlačidlami v radioboxoch si vyberieme, čo sa nachádza pred a za organizmom. Zobrazí sa nám tabuľka pre políčka naľavo a napravo od organizmu. Všetky tieto údaje o jednom organizme môžeme vytlačiť pomocou tlačidla *Všetko vytlačiť*. Texty vo vytlačenej dokumente sa prekladajú podľa zvoleného jazyka aplikácie.

A.6 Vrátenie simulácie späť

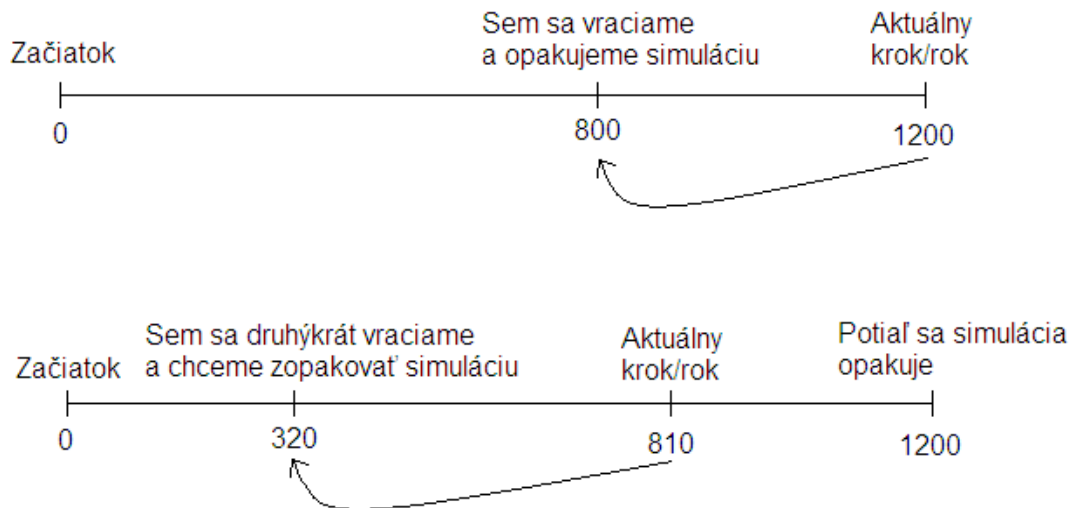
Program umožňuje vrátiť simuláciu do určitého kroku/roku. Táto funkcia je prístupná, ak nie je simulácia v nultom kroku/roku. Kliknutím na jednu zo záložiek v menu *Naspäť* sa zobrazí dialóg vid' Obrázok A.4, v ktorom nastavíme, do ktorého kroku/roku sa chceme vrátiť. Okrem toho si môžeme vybrať, či chceme zopakovať simuláciu rovnako ako prebiehala doteraz alebo chceme, aby od daného kroku prebiehala „nová simulácia“.

Pokiaľ zvolíme, aby sa simulácia zopakovala, tak nesmieme zabudnúť, že ak zmeníme pri opakovaní niektorú pravdepodobnosť, tak sa nám zmení aj priebeh simulácie. Preto, ak chceme presne zopakovať simuláciu, tak by sme po jej vrátení nemali meniť pravdepodobnosti až do kroku, pokiaľ sa simulácia opakuje.



Obrázok A.4: Dialóg pre vrátenie simulácie späť

Pokiaľ sa chceme vrátiť späť na už opakujúcej sa simulácii, a chceme opäť simuláciu zopakovať, tak sa musíme rozhodnúť, do ktorého kroku/roku chceme simuláciu opakovať. Lepšie to vysvetľuje Obrázok A.5. Pri dvojitom opakovaní sa zobrazí podobný dialóg ako pri jednom opakovaní. Jediný rozdiel je v tom, že pri opakovaní si musíme vybrať, do ktorého kroku/roku chceme simuláciu zopakovať.



Simulácia sa môže zopakovať do kroku/roku 810 alebo do 1200.

Obrázok A.5: Schéma dvojitého opakovania simulácie

Možnosť vrátenia sa späť je možné v programe úplne vypnúť. Odstráni sa tým spomalenie spôsobené ukladaním dodatočných informácií potrebných pre vrátenie sa späť. Stačí ak zmeníme hodnotu konfiguračnej položky `store.use` na nulu v súbore `config\graphic_sim.txt`. Nie je však možné vypínať a zapínať túto funkciu počas behu simulácie. Konkrétna simulácia bude používať túto metódu, pokiaľ bola zapnutá pri jej vytvorení.

A.7 Štatistické údaje

Pre ukládanie štatistických súborov obsahuje inštalačný adresár špeciálny podadresár *Statistics*. Všetky štatistické súbory sú uložené vo formáte csv (comma separated value), ktorý je možné upravovať napríklad pomocou tabuľkového editoru Microsoft Excel.

Údaje v štatistických súboroch môžu byť počítané pomocou priemeru alebo pomocou strednej hodnoty. Určuje to konfiguračná položka `statistic.counter` v súbore `config\config.txt`, ktorú môžeme zmeniť. Jej hodnotou môže byť `average` alebo `median`.

Program umožňuje ukladať štyri druhy štatistických údajov.

- Agregované štatistické údaje – obsahujú súhrnné údaje pre aktuálny krok simulácie. Pre druhy potravy a organizmov vypíše ich počty a pre druhy organizmov vypíše „spracované“ hodnoty ich parametrov.
- Krokové štatistické údaje – obsahujú informácie pre každý krok simulácie od jej začiatku. Pre každý krok vypíše počty druhov potravy a organizmov, a pre druhy organizmov vypíše „spracované“ hodnoty ich parametrov.
- Rokové štatistické údaje – sú rovnaké ako krokové, ale počítajú sa pre roky.
- Histogramové štatistické údaje – vypíše parametre všetkých organizmov žijúcich v danom kroku.

A.8 Súbory s nastaveniami

Tieto súbory sa využívajú pre ukladanie nastavení simulácie, a pre spustenie programu pomocou skriptu. Majú definovanú štruktúru a rôzne kľúčové slová. Je potrebné túto štruktúru dodržiavať. Hodnoty pre jednotlivé parametre sa musia nachádzať za daným kľúčovým slovom na tom istom riadku. Všetky súbory sú uložené vo formáte txt.

Niektoré položky v súboroch predstavujú odkazy na ďalšie súbory. Takéto položky majú v názve slovo REFERENCE. Tieto odkazy predstavujú cestu k danému súboru. Cesta môže byť relatívna vzhľadom k inštaláčnemu adresáru alebo môže byť absolútna.

A.8.1 Nastavenia algoritmu

V tomto súbore je uložený počiatočný algoritmus pre daný druh organizmov. Na začiatku súboru je potrebné uviesť počet stavov algoritmu. Každý stav musí byť očíslovaný 1 až n, kde n je počet stavov. Pre každý stav algoritmu sa tu nachádza 49 tabuliek. Pred každou tabuľkou je potrebné špecifikovať, čo sa nachádza pred a za organizmom. Obrázok A.6 popisuje jednu konkrétnu tabuľku a jej hodnoty.

		Okolie						
		N	MFO	FO	SO	DO	O	W
Vľavo	N	F1	L1	R1	N1	F1	L1	R1
	MFO	L1	F1	N1	R1	L1	F1	N1
	FO	R1	N1	F1	L1	R1	N1	F1
	SO	L1	R1	N1	F1	L1	R1	N1
	DO	R1	L1	F1	N1	R1	L1	F1
	O	N1	F1	L1	R1	N1	F1	L1
	W	F1	N1	F1	L1	F1	N1	R1
		Pohyb	Nový stav					

F = N
 B = SO

Kľúčové slová: F, B
 Okolie: N, MFO, FO, SO, DO, O, W

Pohyb: F, L, R

Nový stav: N, MFO, FO, SO, DO, O, W

Legenda okolie
 N: Nič
 MFO: Moja potrava
 FO: Potrava
 SO: Vhodný organizmus
 DO: Nebezpečný organizmus
 O: Organizmus
 W: Stena

Legenda pohyby
 N: Nič
 F: Krok vpred
 L: Otočiť vľavo
 R: Otočiť vpravo

Legenda kľúčové slová
 F: Vpredu
 B: Vzadu

Obrázok A.6: Popis konkrétnej tabuľky v nastaveniach algoritmu

A.8.2 Nastavenia druhu organizmov

V tomto súbore sú uložené počiatočné nastavenia pre jeden druh organizmov.

Príklad:

COUNT 500

MUTATION PROBABILITY 1

EVA REFERENCE SimSettings\test\eva_expr1.bmp

FOOD RGB(00FF00) RGB(0000FF)

GENERAL INFORMATION

MULTIPLE FOR MEAT ENERGY 2.0


```
MAX AGE 120
HUNGRY LIMIT 70
KILL ABILITY 2
COSTS OF ACTIONS L|3 R|3 N|2 F|8
```

```
ALGORITHM REFERENCE SimSettings\test\ok\a3.txt
```

```
MATING INFORMATION
```

```
MIN PARTNER ENERGY 50
MIN PARTNER AGE 10
MAX PARTNER AGE 120
MAX ENERGY FOR PARTNER 70
REQUIRED ENERGY 60
ENERGY FOR CHILD 50
LENGTH OF PREGNANCY 10
```

Význam jednotlivých hodnôt nájdeme v kapitole Umelý život v SofE.

Farba potravy, ktorou sa živia organizmy daného druhu sa uvádza ako číslo v šesnástkovej sústave, ktoré reprezentuje farebný model RGB. Zoznam potrav sa uvádza na jednom riadku, kde sú položky oddelené medzerou.

Na základe položky `MULTIPLE FOR MEAT ENERGY` program zistí, že sa daný druh živí aj inými organizmami. Pre lepšiu prehľadnosť môžeme zadať čiernu farbu do zoznamu potrav (táto farba reprezentuje v nastaveniach mäso).

Okrem hodnoty `MULTIPLE FOR MEAT ENERGY` sú všetky číselné hodnoty celočíselné. Táto hodnota môže byť desatinné číslo.

Položka `COSTS OF ACTIONS` definuje, koľko energie spotrebuje daný druh organizmov na vykonanie jednotlivých pohybov. Význam skratiek je rovnaký ako v nastaveniach algoritmu vid' Obrázok A.6. Hodnoty pre jednotlivé pohyby sú oddelené medzerou.

`ALGORITHM REFERENCE` predstavuje odkaz na súbor, kde je uložený algoritmus pre daný druh organizmov. Ak chceme, tak môžeme algoritmus definovať priamo v tomto súbore. Stačí, ak namiesto tejto položky vložíme rovnaký obsah, ako je opísaný v Nastavenia algoritmu.

A.8.3 Nastavenia simulácie

V tomto súbore sú uložené počiatkové nastavenia pre jednu simuláciu.

Príklad:

```
RGB(FFFF00) AS yellow
RGB(0000FF) AS blue
RGB(000000) AS meat
```

```
{ meat } AS predators
```

```
YEAR 20
```

```
WORLD REFERENCE Worlds\map_expr4.bmp
```

```

SPECIES
NAME org1
ORGANISM REFERENCE SimSettings\test\ok\o_.txt

NAME org2
COUNT 500
MUTATION PROBABILITY 20

EVA REFERENCE SimSettings\test\eva_expr1.bmp

FOOD yellow blue

GENERAL INFORMATION

MULTIPLE FOR MEAT ENERGY 3.0
MAX AGE 120
HUNGRY LIMIT 100
KILL ABILITY 2
COSTS OF ACTIONS L|3 R|3 N|2 F|8

ALGORITHM REFERENCE SimSettings\test\ok\a3.txt

MATING INFORMATION

MIN PARTNER ENERGY 90
MIN PARTNER AGE 10
MAX PARTNER AGE 120
MAX ENERGY FOR PARTNER 60
REQUIRED ENERGY 40
ENERGY FOR CHILD 60
LENGTH OF PREGNANCY 10

MATING PROBABILITY
org1 x org2 30
org2 x { yellow RGB(FF0000) } 50
{ RGB(FF0000) yellow blue } x predators 100

```

Na začiatku súboru môžeme definovať synonymá pre jednotlivé farby potravy k ich ďalšiemu používaniu, a pod nimi synonymá pre množiny potrav. Farby potrav v množine sú oddelené medzerou.

Na začiatku definujeme v položke YEAR množstvo krokov, ktoré zodpovedá jednému roku.

Pod nápisom SPECIES začínajú definície druhov organizmov pre túto simuláciu. Každá definícia obsahuje názov druhu v položke NAME. Pokiaľ nie je názov za slovom NAME uvedený, tak sa vytvorí implicitný názov druhu (viď Dialóg pre novú simuláciu). Okrem toho obsahuje aj samotnú definíciu druhu, ktorá môže byť uvedená buď ako odkaz na súbor s nastaveniami pre jeden druh organizmov, alebo môže byť priamo uvedená pod názvom druhu organizmov.

Na konci súboru môžeme definovať pravdepodobnosť spárenie pre rôzne druhy organizmov. Jeden druh organizmov môže byť definovaný menom, synonymom, alebo množinou potravín.

A.8.4Skript

Pomocou tohto súboru môžeme spustiť program bez grafického rozhrania. Obsahuje nastavenia pre jednotlivé simulácie, ktoré môžu byť uvedené buď ako odkaz na súbor s nastaveniami pre simuláciu, alebo priamo v tomto súbore.

Príklad:

```
SIMULATION mff-sim
STEP ENDTIME 1200

SETTINGS REFERENCE mff-settings.txt

FOOD PROBABILITY
RGB(1245AC) 40
something 80 90/120 50/1000

STATISTICS
AGGRSTAT NAME aggr.csv
HISTSTAT NAME hist.csv
STEPSTAT NAME step.csv
YEARSTAT NAME year.csv

STEP SAVETIME AGGRSTAT 150
YEAR SAVETIME HISTSTAT 40
YEAR SAVETIMES STEPSTAT 40 80 1100 23000

END

SIMULATION new
YEAR ENDTIME 560

IN

YEAR 120
WORLD REFERENCE Worlds\new.bmp

...
FOOD PROBABILITY
RGB(1245AC) 40

STATISTICS
YEARSTAT NAME year.csv

STEP SAVETIME YEARSTAT 2
```

Na začiatku každej simulácie môžeme zadať jej meno. Pokiaľ meno za položkou SIMULATION neuvedieme, tak sa vygeneruje implicitne podobným spôsobom ako pre druhy organizmov. Namiesto textu „Organizmus“ sa však uvedie „Simulácia“.

Následne musíme uviesť počet krokov/rokov, ktoré bude simulácia spustená.

Pokiaľ chceme uviesť viac simulácií, tak ich definície musíme oddeliť položkou END. V prípade, že chceme nastavenia simulácie napísať priamo do tohto súboru, tak na začiatok týchto nastavení musíme vložiť položku IN.

Pri použití skriptu máme možnosť meniť pravdepodobnosť tvorby určitej potravy počas simulácie. Za definíciu farby (môže ňou byť aj synonymum definované v nastaveniach simulácie) napíšeme počiatočnú pravdepodobnosť (tá je implicitne nastavená na 50%). Za túto hodnotu ďalej napíšeme medzerou oddelené zmeny pravdepodobnosti, ktoré nastanú v konkrétnom kroku simulácie. Ako prvú zapisujeme pravdepodobnosť a za lomítkom krok, v ktorom má dôjsť k zmene.

Program nám umožňuje ukladať rôzne druhy štatistických údajov (viď Štatistické údaje). Pri ukladaní štatistických údajov je názov súboru vytvorený z kroku/roku, v ktorom sa údaje ukladajú a koncovky, ktorú môžeme zadať v skripte.

Čas, v ktorom sa jednotlivé štatistické údaje ukladajú môžeme definovať v krokoch alebo rokoch. Môžeme definovať buď periódu (SAVETIME), alebo jednotlivé kroky oddelené medzerami (SAVETIMES), v ktorých sa majú údaje ukladať.