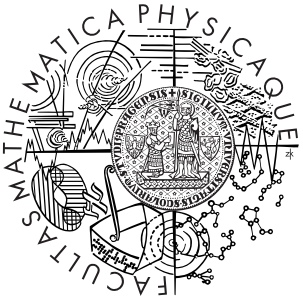
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tomáš Kraut

Evoluce parametrů páření

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Tomáš Holan, Ph.D.,

Studijní program: Informatika, obor Obecná informatika

2010

Na tomto místě bych chtěl poděkovat především vedoucímu mé bakalářské práce RNDr. Tomáši Holanovi, Ph.D. za poskytnuté konzultace a rady při psaní této práce. Janě Zákorové za pomoc při finálním odstraňování překlepů. Dále pak všem, kteří mě při jejím psaní jakýmkoliv jiným způsobem podpořili.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6. srpna 2010 Tomáš Kraut

Obsah

Název práce: Evoluce parametrů páření
Autor: Tomáš Kraut
Katedra (ústav): Kabinet software a výuky informatiky
Vedoucí bakalářské práce: RNDr. Tomáš Holan, Ph.D.
e-mail vedoucího: tomas.holan@mff.cuni.cz

Abstrakt:
Tato práce se zabývá implementací prostředí pro simulaci umělého světa a zkoumání vývoje v něm žijících jednoduchých umělých organismů. V první části jsou popsán základní principy tohoto modelu světa, jeho vytváření a změny, růst potravy a průběh života jedinců, způsob získávání potravy a způsob jejich rozmnožování. Dále je implementováno tohoto prostředí v jazyce C++, je vysvětleno, jak se implementovaný program používá, popsán je způsob automatických změn v simulaci pomocí skriptů a způsob, kterým lze z tohoto programu získat statistická data pro další zpracování.

Klíčová slova: evoluce, umělý život, simulace, páření

Title: Mating criteria evolution
Author: Tomáš Kraut
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: This work is concerned on implementating an environment for simulation of artificial world and studying the evolution of simple artificial creatures. First, the basic principles od this model are described, as its creation, changing, growth of food, lifetime of individual creatur and means ot their reproduction. Next, this environment is implemented in C++ language, there is explained, how to use implemented program, automatic changes in simulation by scripts is described and also how to get statistical data for next elaboration.

Keywords: evolution, artificial life, simulation, mating

Kapitola 1

Úvod

Tato práce se zabývá umělým životem, ve smyslu vytvoření prostředí pro simulaci umělého světa, s předem danými pravidly. Využitím analogií se skutečným světem, ovšem se značným zjednodušením, které umožní za velmi krátký časový úsek proběhnout velké možství generací *umělých* organismů by bylo možné vzdáleně simulovat změny chování organismů *skutečných* a zkusit odpovědět na různé otázky *jak* a *proč* u jejich chování při shánění prostředků k přežití a k rozmnožování. [1]

Světem pro simulaci je čtvercová síť předem daných rozměrů. V této síti se na základě svých parametrů, algoritmu a paměti pohybují organismy. Navíc na každém poli, pokud zrovna není obsazeno organismem, může růst potrava. Simulace probíhá po jednotlivých krocích, každý krok je rozdělen na tři fáze. V první fázi se provedou naplánované změny světa, ve druhé na prázdných polích roste potrava a ve třetí každý organismus provede jeden tah. Vždy ve svém tahu organismus provede akci na základě své genetické informace, paměti a okolí. To, jak organismus vnímá své okolí, také částečně závidí na jeho parametrech. Všechny organismy se řídí stejným algoritmem.

Genetická informace nového jedince při rozmnožování vzniká zkřížením genetických informací jeho rodičů s následnou mutací, která umožňuje, aby se organismy dále zlepšovaly za pomoci *kumulativní selekce* – tedy série drobných zlepšení, směrovaných přirozeným výběrem [2].

Kapitola 2

Simulace

2.1 Prostředí

Simulace probíhá ve čtvercové síti předem daných rozměrů, kde pravým sousedem posledního sloupce je první sloupec, stejně jako za posledním řádkem opět následuje první. Každé políčko má svoji úrodnost, tedy pravděpodobnost, s jakou na ní vyroste potrava (ta je dále násobena globální úrodností - globálním parametrem simulace) a počet kroků, po kterých potrava roste. Políčko může být prázdné, může na něm být umístěna potrava, nebo na něm může být v jednom ze čtyř směrů umístěn organismus.

2.2 Organismy

Život organismů, vyskytujících se v simulaci se skládá z jednotlivých tahů. Každý organismus je umístěn na jednom poli čtvercové sítě, je nasměrován jedním směrem, ve svém tahu vidí pouze tři sousední pole – ve směru, kterým je umístěn, a vlevo a vpravo od tohoto směru. Nevidí tedy na pole za ním ani diagonálně.

Každý organismus si s sebou nese genetickou informaci v podobě parametrů páření (*nabízí*, *požaduje* a *pro potomka*) a tabulky. Tabulka obsahuje informace o tom, jakou akci má organismus provést, pro všechny kombinace toho, co může na sousedních polích organismus vidět, a pro každý vnitřní stav organismu (má k dispozici čtyři).

Každý organismus má navíc své pořadové číslo a pamatuje si číslo otce i matky.

2.2.1 Algoritmus

Jeden tah organismu tedy vypadá takto:

Pokud má organismus nastaven příznak *čeká*, pak pouze zmenší hodnotu parametru *doba čekání* o 1 a prováděnou akcí je akce *nic*.

Jinak na základě obsazení pozic ve směru, vlevo a vpravo a vnitřního stavu organismu se provede jedna akce a zároveň se nastaví nový vnitřní stav. Algoritmus lze

tedy popsat následujícím zobrazením

$$f : W^3 \times S \to A \times S$$

kde $W = \{vhodny, nevhodny, potrava, nic\}$ je množina obsazení pozice sousedního pole, $S = \mathbb{Z}_4$ množina vnitřních stavů a $A = \{krok, vlevo, vpravo, nic\}$ množina akcí. Sousední pole je pro organismus *vhodny*, pokud je na daném poli jiný organismus, jenž nyní bezprostředně po páření nečeká na nového potomka a jeho parametr *požaduje* je menší nebo roven parametru *nabízí* aktivního organismu. Je-li parametr *nabízí* větší, než aktuální energie organismu, zmenšená o hodnotu parametru *pro potomka* a cenu kroku, pak je místo něj použita tato. Pokud je na sousedním poli organismus a není *vhodny*, pak je *nevhodny*. Není-li na sousedním poli organismus, pak zbývají možnosti *potrava* a *nic* v závislosti na přítomnosti potravy.

Je-li výstupem f akce *vlevo* nebo *vpravo*, organismus pouze změní směr, ve kterém je umístěn (a tím se mu pro příští tah změní pole, která vidí), ovšem zůstane na stejném poli. Při akci *nic* zůstane organismus na stejném poli a ani nezmění směr. Provedení alce *krok* může probíhat více způsoby. Buď je pole před organismem volné, pak se na něj přesune. Je-li na tomo poli navíc potrava, zpracuje ji a přemění na energii. Jinak je na daném poli jiný organismus a je *nevhodný* k páření, pak je výsledkem akce to samé, jako při akci *nic* - tedy neprovede nic. Ovšem zaplatí množství energie za akci *krok*.

Je-li na daném poli organismus *vhodný* k páření, pak proběhne samotné páření.

2.2.2 Páření

Páření nastane, pokud organismus (dále F), který právě provádí svůj tah, chce provést akci *krok* na pole obsazené organismem, který je pro F *vhodný* k páření (dále M). Organismu F se odečte počet energie, určený aritmetickým průměrem jeho parametru *nabízí* a parametru *požaduje* organismu M. Organismu M se tato energie přičte. Následně je vytvořen organismus, jehož genetická informace vznikne použitím parametrů a jednotlivých položek tabulky vždy se stejnou pravděpodobností $1/2$ buď od F nebo od M. Pouze parametr *pro potomka* se vždy bere od F. Následně je s pravděpodobností, zadanou parametrem světa *pravděpodobnost mutace*, provedena v jedné části genetické informace mutace. V případě mutace parametru nastává změna o 10% a poté ještě o celočíselnou hodnotu, vybranou rovnoměrně náhodně z intervalu $\langle -10; 10 \rangle$.

Následně se organismu M nastaví *doba čekání* podle stejnojmenného parametru světa a nastaví příznak *čeká*.

Tyto tři parametry (*nabízí*, *požaduje* a *pro potomka*) umožňují rozdělit ztrátu energie předané nově vzniklému jedinci mezi oba rodiče tím, že parametry nejsou omezeny na nezáporná čísla, rodičovskou investicí matky do potomka pak je právě onen záporný poplatek za páření.

2.2.3 Umístění potomka do světa

Organismus, který se dostane na tah, má nastaven příznak *čeká* a sníží se mu v jeho tahu čítač *doba čekání* na 0, před svým tahem umístí svého potomka do světa na nejbližší volné pole. Pokud jsou všechna pole obsazená organismy, pak je potomek bez náhrady ztracen. Organismu je poté zrušen příznak *čeká* a okamžitě provádí další tah.

2.2.4 Barvy

Organismus může být pro sledování označen jednou z osmi barev. Barvy se také předávají potomkům a to tímto způsobem:

červená
: potomek je obarven, pokud je obarven libovolný z rodičů

zelená
: potomek je obarven, pokud jsou obarveni oba rodiče

modrá
: potomek je obarven, pokud je obarven otec

žlutá
: potomek je obarven, pokud je obarvena matka

purpurová
: potomek je obarven, pokud je obarven pouze otec

tyrkysová
: potomek je obarven, pokud je obarvena pouze matka

bílá
: potomek je obarven, pokud je obarven právě jeden rodič

černá
: potomek je obarven, pokud není obarven ani jeden rodič

2.3 Průběh

2.3.1 Změny světa

Před každým krokem se provedou události[1], které jsou naplánovány na tento krok, případně na předchozí kroky. To může mít za následek i ukončení simulace (události *exit*, *load*), jinak se po skončení této části kroku odstraní události, naplánované dříve, než na aktuální čas, jsou-li nějaké.

[1] Seznam možných událostí je uveden v příloze A

2.3.2 Růst potravy

Pro všechna pole v mapě postupně po sloupcích (ovšem na pořadí nezáleží, protože neprobíhá interakce s jinými poli) se provede následující: Pokud na poli není brouček (a je nastaven parametr *doba růstu*), zmenší dobu čekání na růst potravy. Pokud je doba čekání na růst nulová (nebo parametr *doba růstu* není nastaven) a potrava na poli není, pak nastává na tomto poli růst potravy. Určí se, zda růst potravy byl úspěšný, tedy vybere se náhodné celé číslo z intervalu $\langle 0; 255 \rangle$. Pokud je menší, než úrodnost pole, násobená globální úrodností, pak se na pole umístí potrava. Nakonec nastaví aktuální dobu čekání na růst podle parametru pole *doba růstu*.

2.3.3 Tahy organismů

Následně má každý organismus (v pořadí vzestupně podle svého identifikátoru) k dispozici jeden tah. Průběh jednoho jeho tahu je již popsán v části 2.2.1.

Kapitola 3

Implementace

Simulace je implementována v jazyce C++. Ke grafickému zobrazení simulace, interakci s uživatelem a ukládání mezivýsledků využívá funkcí knihovny MFC[1]. Umožňuje jak vytvoření světa přímým zadáním parametrů a následné spuštění, při kterém se dění světa může rovnou vykreslovat na obrazovku, tak i spuštění z příkazové řádky, které simulaci pustí na pozadí a pouze ukládá případné statistické výstupy pro další zpracování.

3.1 Kostra aplikace

Základní kostru aplikace tvoří třída `CEPPApp`, odvozená od MFC třídy `CWinApp`, která zajišťuje propojení třídy pro svět (viz 3.2) s hlavním oknem programu. Jeho ovládání je zprostředkováno pomocí třídy `CMainFrame`, dědící od `CFrameWnd`, která navíc vlastní třídy jednotlivých ovládacích panelů a přijímá zprávy vyvolané jejich použitím. Grafické zobrazení světa je implementováno třídou `CEPPView`, která je potomkem knihovní třídy `CView`. Při vykreslování světa není vykreslováno přímo na obrazovku, ale do pomocného bufferu, který je následně celý najednou překopírován na plochu obrazovky.

3.2 Svět

Svět je implementován třídou `CWorld`, která je potomkem třídy `CDocument` z knihovny MFC. Tato třída zajišťuje všechny činnosti, které nějakým způsobem zjišťují nebo mění parametry světa nebo obsah jednotlivých polí. Zároveň má na starosti spouštění kroků – tedy zpracovává události a volá metodu třídy `CCell` pro růst potravy a třídy `CBug` pro tahy organismů. Udržuje si seznam organismů, mapu světa i frontu událostí.

Vytváření světa

Při vytváření světa se nejdřív případná data z předchozích světů, načtených v jednom běhu programu, smažou, následně se nastaví zadané parametry. Pokud byla zadána

[1] Microsoft Foundation Class Library

mapa, načtou se jednotlivá pole, pokud byly zadány rozměry světa, vytvoří se pole s náhodnou úrodností a dobou růstu v intervalu v intervalu $\langle 0; 255 \rangle$.

Zásahy do světa

Zásahy do světa jsou prováděny pomocí metod třídy `CWorld`. Jedná se o změny parametrů světa (buď pomocí metody `setParam()`, která nastaví parametr, určený klíčovým řetězcem[2], případně metodou, příslušející přímo ke konkrétnímu parametru) a případně přidávání nových organismů (metoda `addBug()`).

3.3 Události

Pro zpracování událostí je v programu vytvořena abstraktní třída `IEvent`, která deklaruje virtuální metodu `apply`, která dostane jako svůj jediný parametr ukazatel na běžící svět, a pro výpis událostí navíc operátor přetypování na řetězec.

```
virtual void apply(CWorld * world) = 0;
virtual operator CString() = 0;
```

Načítání příkazů probíhá v metodě `Event::parse` pomocí parseru, implementovaného třídou `CCsvLine` (viz 3.4), s nastavením mezery jako oddělovače záznamů.

Pro plánování událostí se využívá metody `scheduleAt` pro naplánování na čas od začátku simulace a `scheduleAfter` pro naplánování na čas od aktuálního kroku.

Všechny třídy, dědící od `IEvent`, jsou pojmenovány podobně jako jejich příkazy ve skriptech plus koncovka `Event`, tedy např. `RepeatEvent` pro opakovanou událost, `RaiseEvent` pro událost, která zvedne hodnotu parametru apod.

3.4 CSV

Pro jednodušší ukládání statistických výstupů a také pro načítání skriptů s parametry je použita třída `CCsvLine`, která umožňuje načíst řádku ve formátu CSV[3] a z této řádky vytáhnout jednotlivé záznamy (řetězce nebo celá čísla). Stejně jako naopak umožňuje postupně přidávat záznamy a následně vyexportovat řetězec pro uložení do souboru. Umožňuje nastavit oddělovací znak, tedy pokud místo oddělovače „`,`“ nastavíme „␣“ (znak mezery), pak lze tuto třídu použít k načítání parametrů událostí v načítaných skriptech. Pro přidávání záznamů je přetypován operátor `>>`, pro výběr záznamů operátor `<<`.

[2]Seznam možných klíčů je uveden v příloze B

[3]Comma Separated Values

3.5 Organismus

Organismus je v programu reprezentován třídou `CBug`[4]. Ta má na starosti uchování genetické informace (parametrů a tabulky) a aktuálního stavu, tedy umístění ve světě, vnitřního stavu, označení barvami, počtu potomků. Vykonává algoritmus organismu, tedy vyhodnocuje vhodnost případných partnerů pro páření a na základě tabulky rozhoduje, jakou akci organismus provede.

3.6 Ukládání a načítání

Pro ukládání stavu světa na disk a jeho pozdější načtení je využita serializace pomocí třídy `CArchive` z knihovny MFC. Třídy `CWorld`, `CCell` a `CBug` proto implementují virtuální metodu `Serialize(CArchive &ar)`, ve které jsou do archivu `ar`, zprostředkovaného frameworkem MFC, operátorem `<<` uložena následující data.

Pro třídu `CWorld`

- počet proběhlých tahů
- zisk energie z jedné potravy
- globální úrodnost
- počáteční energie nového broučka
- maximální energie broučka
- pravděpodobnost mutace
- cena akce *krok*
- cena akce *vlevo*
- cena akce *vpravo*
- cena akce *nic*
- čas čekání
- šířka mapy
- výška mapy
- po sloupcích postupně všechna pole mapy
- počet organismů
- postupně všechny organismy

Pro třídu `CCell`

[4] pracovní název organismu je „brouček“

- úrodnost pole
- zda je na poli uložena potrava
- interval růstu potravy
- čas aktuálně zbývající do růstu potravy

Pro třídu `CBug`

- indikátor, zda má potomka, ještě nevypuštěného do světa
- pokud má potomka, pak jej zde rekurzivně uloží
- ID broučka
- X-ová souřadnice ve světě
- Y-ová souřadnice ve světě
- směr broučka
- energie broučka
- množství zpracované potravy
- vnitřní stav broučka
- maximální platba za páření
- minimální platba za poskytnutí páření
- energie, předávaná potomkovi
- ID otce
- ID matky
- indikátor, zda čeká
- čas do konce čekání
- pro každý stav, pro každou situaci na poli před ním, na poli vlevo od něho a na poli vpravo od něho, akce, kterou provede a stav, do kterého se přepne, celkem tedy $4 \cdot 4 \cdot 4 \cdot 4 \cdot 2 = 512$ hodnot

Označení barvami se neukládá.

Zpětné načtení takto uloženého světa probíhá stejně, jen se hodnoty neukládají do archivu `ar`, ale načítají se z něj operátorem `>>`.

Třídy `CBug` a `CWorld` mají navíc metody pro uložení do formátu CSV (`toCSV()`) a načtení z něj (`fromCSV()`).

3.7 GUI

Grafické uživatelské rozhraní je vytvořeno pomocí nástrojů pro vytváření GUI, integrovaných ve vývojovém prostředí Visual Studio 2010 [3]. Jeho funkčnost je pak dále implementována obslužnými metodami tříd `CMainFrame`, `CEPPView` a `CEPPApp`, které dále volají příslušné metody tříd světa(`CWorld`), políčka (`CCell`) nebo organismu (`CBug`).

Kapitola 4

Použití programu

4.1 Spuštění světa

Po spuštění programu nebo kdykoliv po vybrání položky menu Svět → Vytvoř svět (klávesová zkratka *N*) se objeví okno s parametry nového světa (Obrázek 4.1). Jsou to tyto:

Šířka
: šířka náhodně generované mapy. *Kladné celé číslo.*

Výška
: výška náhodně generované mapy. *Kladné celé číslo.*

Doba čekání
: doba kterou brouček čeká na narození potomka. *Nezáporné celé číslo.*

Energie/potrava
: množství energie, kterou organismus získá zpracováním jednoho kusu potravy. *Kladné celé číslo. Je vhodné, aby bylo větší, než cena kroku.*

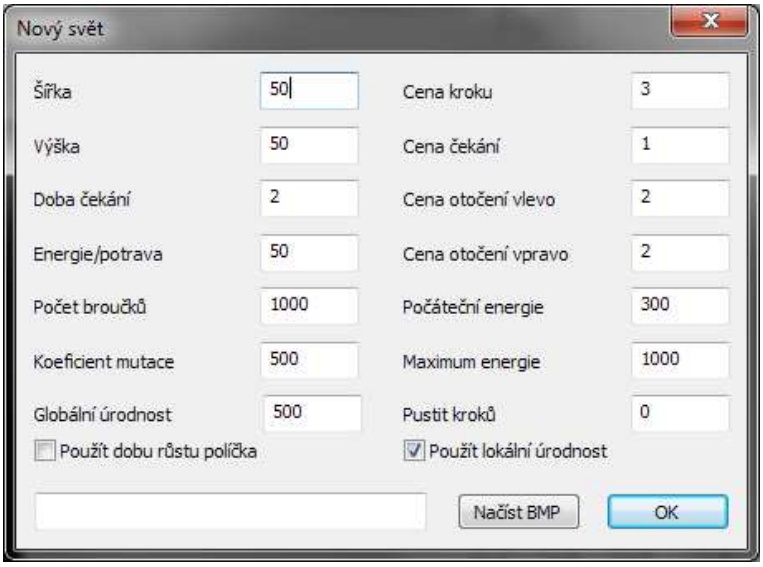
Počet broučků
: počet náhodně generovaných organismů, kteří se objeví ve světě hned od začátku. *Nezáporné celé číslo.*

Koeficient mutace
: pravděpodobnost v promile, že při vzniku organismu nastane mutace. *0 – 1000*

Globální úrodnost
: pravděpodobnost v promile, že se objeví potrava, poté, co na políčku vyroste. *0 – 1000*

Cena kroku
: energie, kterou organismus spotřebuje na provedení akce krok vpřed. *Celé číslo.*

Cena čekání
: energie, kterou organismus spotřebuje na provedení akce čekání. *Celé číslo.*



Obrázek 4.1: Vytvoření světa

Cena otočení vlevo
: energie, kterou organismus spotřebuje na provedení akce otočení vlevo. *Celé číslo.*

Cena otočení vpravo
: energie, kterou organismus spotřebuje na provedení akce otočení vpravo. *Celé číslo.*

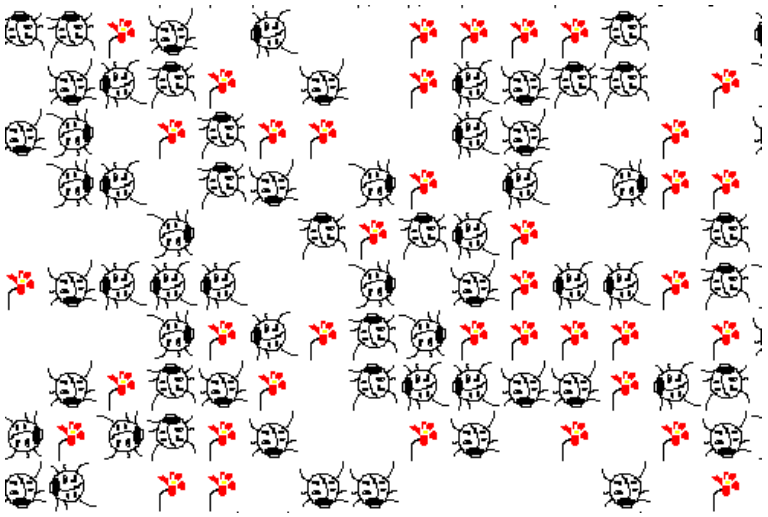
Počáteční energie
: energie, kterou má nově vygenerovaný organismus k dispozici od začátku. *Nezáporné celé číslo. Je vhodné, aby bylo větší, než cena kroku.*

Maximum energie
: maximální množství energie, které může mít organismus k dispozici, jakákoliv získaná energie nad tuto hodnotu bude ztracena. *Celé číslo. 0 znamená neomezeně.*

Pustit kroků
: kolik kroků se má ihned po vytvoření provést. *Celé číslo.* -1 *znamená neomezeně.*

Použít dobu růstu políčka
: zda použít dobu růstu, určenou políčkem, nebo potrava roste každý tah

Použít lokální úrodnost
: zda použít úrodnost políčka, nebo potrava roste vždy



Obrázek 4.2: Plocha světa

Načíst BMP
: stisk tohoto tlačítka vyvolá dialogové okno pro výběr obrázku. Vybraný obrázek se použije jako mapa světa místo náhodně generované ze zadaných rozměrů.

Zde lze parametry změnit, případně tlačítkem načíst mapu světa. Poté se nebere v úvahu šířka a výška mapy. Po vyplnění příslušných políček a stisknutí tlačítka se vytvoří nový svět.

4.2 Zobrazení světa

Náhled do světa zprostředkovává volná plocha programu, která není zabrána ovládacími a jinými panely. (Obrázek 4.2)

Zobrazuje se čtvercová síť, na políčku může být:

Potrava
: Zobrazí se obrázek kytičky-

Organismus
: Zobrazí se obrázek broučka, ve směru, do kterého je otočen.

Prázdno
: Pole zůstane vyplněné pouze barvou pozadí.

Pro každý rozměr platí, že pokud je v něm velikost světa větší, zobrazí se výřez, je-li velikost menší, políčka se opakují. Výřez světa lze posouvat buď tlačítky na ovládacím panelu ($<$ vlevo, $>$ vpravo, $\vee$ dolů, $\wedge$ nahoru) nebo klávesami (W nahoru, A doleva, S dolů, D doprava). Vzdálenost posunu lze nastavit na ovládacím panelu v poli vpravo dole od tlačítek posunu.

Po kliknutí na políčko se do panelu „Brouček“ načtou informace o organismu na aktuálním poli, je-li tam nějaký, a do panelu „Pole“ informace o políčku samotném.

4.2.1 Panel „Brouček“

Zde se ukazují základní informace o organismu. Jeho identifikační číslo, číslo jeho otce a matky (pokud se zde zobrazuje 0, pak brouček nemá rodiče, je tedy ve světě od jeho vytvoření, případně byl vložen někdy v průběhu simulace vnějším zásahem), jeho stáří (počet kroků, které provedl), počet potomků, aktuální stav energie, jeho parametry – požadavky na partnera, zda právě čeká a kolik tahů čekání mu ještě zbývá.

Ve spodní části se pak zobrazuje algoritmus organismu, vždy pro jeden vybraný stav. Jejich přepínání probíhá pomocí číslovaných radiobuttonů. (Nemá vliv na aktuální stav organismu v simulaci, pouze zobrazí příslušnou část jeho algoritmu.)

Zaškrtnutí checkboxu „Sleduj“ zajistí, že střed mapy bude vždy nastaven na pozici tohoho broučka. Tlačítko „Ulož“ vyvolá dialogové okno pro výběr souboru, do kterého se následně uloží brouček pro pozdější načtení. Tlačítkem „Odstraň“ lze broučka odebrat ze simulace.

Panel lze zobrazit či skrýt výběrem položky menu „Zobraz“ → „Brouček“ nebo klávesovou zkratkou Ctrl+B.

4.2.2 Panel „Pole“

Zde se zobrazuje úrodnost vybraného pole (v rozmezí 0 – 255), interval, po jakém se rozhoduje, zda má vyrůst potrava (také 0 – 255) a to, zda je na políčku aktuálně potrava. Tyto hodnoty lze přímým vepsáním nových údajů měnit. Změna se projeví okamžitě, pokud je simulace zastavena, pokud běží, změna nastane po konci probíhajícího tahu.

Panel lze zobrazit či skrýt výběrem položky menu „Zobraz“ → „Buňka“ nebo klávesovou zkratkou Ctrl+C.

4.2.3 Panel „Svět“

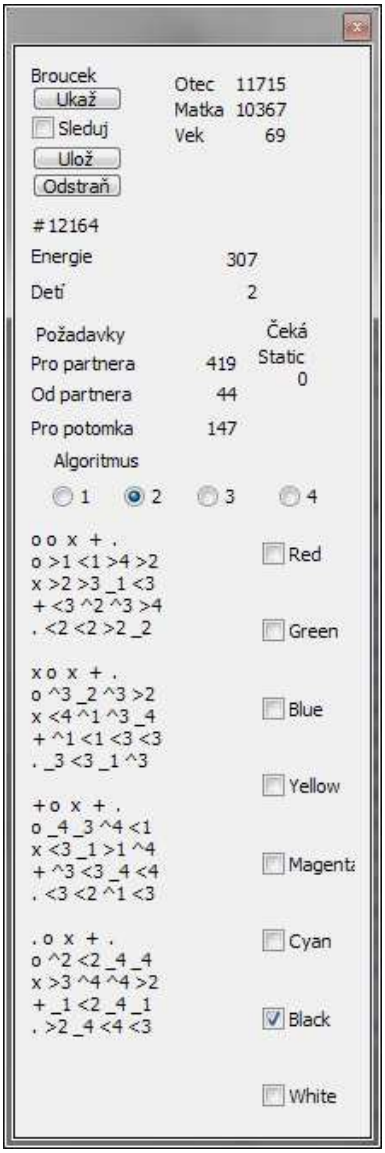
Zde se ukazují souhrnné informace o světě. Tedy počet broučků, množství potravy ve světě, celkové množství energie, kterým organismy disponují, počet tahů, které uběhly od spuštění simulace.

Dále se zde zobrazují parametry světa, které lze zároveň na tomto místě měnit s platností od následujícího tahu. Jsou to stejné parametry jako při spouštění světa (část 4.1), kromě výšky a šířky mapy, počtu broučků a jejich počáteční energie, které nelze během simulace měnit.

Ve spodní části panelu najdeme minimum, maximum, průměr a medián z věku, počtu potomků, parametru *nabízí*, parametru *požaduje* a množství potravy, zpracované organismy.

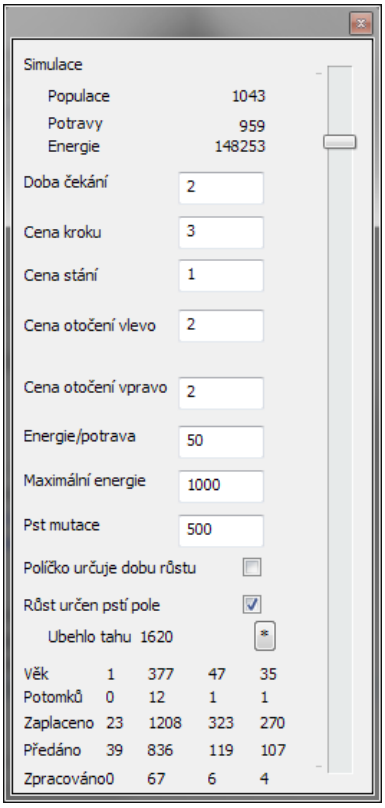
4.3 Ovládání simulace

Průběh simulace se ovládá prvky na ovládacím panelu (Obrázek 4.6) vlevo. Simulaci lze pustit stisknutím tlačítka „Pust“. Pokud je zaškrtnut checkbox „Neomezeně“,



Obrázek 4.3: Panel „Brouček“

Obrázek 4.4: Panel „Pole“



Obrázek 4.5: Panel „Svět“

Obrázek 4.6: Ovládání světa

pak simulace běží až do stisku tlačítka „Stop“, jinak běží zadaný počet tahů[1].

Zobrazení simulace se ovládá na ovládacím panelu vpravo. Lze nastavit, po kolika proběhlých tazích se obrazovka překreslí, případně překreslování úplně vypnout. Překreslení vždy proběhne při stisknutí tlačítka „Zobraz teď“, stisku klávesy *F5* a také vždy při změně zobrazované plochy jakými je zvětšení okna, zmenšení okna, skrytí ovládacího panelu apod.

Dále lze na panelu zobrazení světa (Obrázek 4.5) změnou příslušného textového pole nebo checkboxu měnit parametry simulace.

4.4 Uložení a načtení světa

Stisk Ctrl+S, případně výběr „Svět“ → „Uložit“ v menu vyvolá dialogové okno pro výběr souboru, do kterého se má uložit aktuální svět. Simulaci lze později načíst výběrem uloženého souboru v dialogovém okně, vyvolaném pomocí kombinace kláves Ctrl+O, případně výběrem menu „Svět“ → „Otevřít“.

4.5 Načtení skriptu

Klávesová zkratka Ctrl+L nebo výběr „Svět“ → „Načíst skript“ vyvolá dialogové okno pro výběr souboru se skriptem, který je po vybrání načten. (Seznam použitelných příkazů je uveden v příloze A.)

4.6 Zásahy do světa

Je možné zasahovat do dění ve světě několika způsoby. Prvním je změna parametrů světa, jak bylo popsáno v 4.2.3. Další možností je změnit parametry určitého políčka (4.2.2), tedy jeho úrodnost a čas, za který se potrava objeví, případně přímo přidat či odebrat potravu na políčku. Poslední možností je odstranit či přidat jednoho nebo několik broučků. Odstranění se provede kliknutím na broučka, čímž ho vybereme, a dále tlačítkem „Odstranit“ v panelu s informacemi o broučkovi. Přidání se provádí výběrem menu „Brouček“ → „Přidej“. Objeví se okno s možností zadání kolik broučků se má přidat. Broučci se generují náhodně a stejně tak se přidávají na náhodné políčko. Je také možné přidat dříve uloženého nebo vytvořeného broučka. Výběrem položky menu „Brouček“ → „Načti“ se vyvolá dialogové okno se zadáním počtu a po stisknutí tlačítka „OK“ další okno pro výběr souboru, ze kterého se broučci mají načítat. Broučci se přidávají na náhodné místo mapy.

[1]Pokud během simulace dojde k události, mající za následek její ukončení, pak samozřejmě skončí dříve

4.7 Statistické výstupy

Výběrem „Záznam“ → „Start“ v menu se objeví okno pro zadání intervalu ukládání statistických dat, následně okno pro výběr souboru, do kterého se má ukládat a od toho okamžiku se v zadaných intervalech ukládají aktuální statistická data do souboru ve formátu CSV. Pokud vybraný soubor neexstuje, vytvoží se, jinak je jeho původní obsah smazán. Na první řádek se uloží názvy sloupců a postupně vždy po uběhnutí zadaného počtu tahů se do souboru připíše řádek s těmito údaji: počet tahů od začátku simulace, počet broučků, počet potravy, celkové množství energie broučků a dále součet minimum, maximum, průměr, medián, první a třetí kvartil z těchto údajů:

- věk
- počet potomků
- počet zpracované potravy
- parametr *pro potomka*
- parametr *nabízí*
- parametr *požaduje*

Kapitola 5

Závěr

Vědeckou disciplínu Umělý život – Artificial life pojmenoval v roce 1986 Christopher Langton. [4]. Od té doby se na světě objevila minimálně desítka významných simulátorů umělého života. [5] Program, implementovaný v rámci této bakalářské práce, se může mezi ně zařadit, pokud se bude podílet na nějakém budoucím výzkumu oblasti chování jedinců při výběru partnerů.

Proto by bylo vhodné dále ještě vylepšit skriptovací jazyk, pomocí kterého lze nyní provádět pouze jednoduché změny, tak, aby bylo možné změnu parametrů lépe zautomatizovat, například zavedením proměnných.

V neposlední řadě by bylo možné také program upravit, aby fungoval například jako spořič obrazovky, aby byla využito grafické zobrazení světa a organismů v něm.

Literatura

[1] DAWKINS, Richard. *Sobecký gen.* Praha : Mladá fronta, 1998. 318 s.

[2] DAWKINS, Richard. *Slepý hodinář* : Zázrak života očima evoluční biologie. Litomyšl : Paseka, 2002. 357 s. ISBN 80-7185-445-X.

[3] *MFC Reference* [online]. ©2010 [cit. 2010-08-06]. Dostupné z WWW: <http://msdn.microsoft.com/en-us/library/d06h2x6e(v=VS.80).aspx>

[4] WILSON, Robert A.; KEIL, Frank C. *The MIT encyclopedia of the cognitive sciences* [online]. [s.l.] : [s.n.], 2001 [cit. 2010-08-06]. Artificial Life, s. . Dostupné z WWW: < http://books.google.com/books?id=-wt1aZrGXLYC&lpg=PA37&hl=cs&pg=PA37#v=onepage&q&f=false>

[5] Wikipedia : the free encyclopedia [online]. 20 July 2010 [cit. 2010-08-06]. *Artificial life.* Dostupné z WWW: <http://en.wikipedia.org/wiki/Artificial_life>.

Příloha A

Seznam událostí

none
Neprovádí nic.

message
Zobrazí okno se zprávou (1).

param
Změní parametr (1) světa. Tato událost má čtyři varianty v závislosti na druhém parametru

set – nastaví parametr na hodnotu (3)

add – přičte k parametru hodnotu (3)

raise – zvětší parametr o (3) %

lower – zmenší parametr o (3) %

repeat
Opakuje po (1) krocích, maximálně (2)× příkaz (3).

ifg
Pokud je parametr (1) větší, než (2), provede (3), jinak (4).

script
Načte skript ze souboru (1).

createworld
Vytvoří svět s těmito parametry:

1. šířka
2. výška
3. počet organismů
4. doba čekání
5. cena kroku vpřed

6. cena otočení vlevo
7. cena otočení vpravo
8. cena stání
9. počáteční energie
10. zisk energie z potravy
11. maximální energie
12. používat dobu růstu
13. používat lokální úrodnost
14. globální úrodnost
15. pravděpodobnost mutace

startlog
: Začne ukládat záznam do souboru (1), v intervalech (2).

load
: Načte svět ze souboru (1) a nechá běžet neomezeně dlouho.

save
: Uloží svět do souboru (1).

V závorkách je uvedeno pořadí parametru příkazu. Parametry se oddělují mezerou. Pokud má parametr obsahovat mezeru, je třeba jej uzavřít do uvozovek `"`.

Příloha B

Seznam nastavitelných parametrů metodou setParam()

gf
: globální úrodnost

wt
: doba čekání

me
: maximální energie organismu

ie
: počáteční energie organismu

eg
: zisk energie z potravy

fp
: cena akce *krok*

lp
: cena akce *vlevo*

rp
: cena akce *vpravo*

sp
: cena akce *nic*

mp
: pravděpodobnost mutace

Příloha C

Veřejné metody a atributy tříd, implementujících vlastní simulaci

CWorld

Načítání a ukládání světa

fromCSV
: načtení světa z CSV řetězce

toCSV
: uložení světa do CSV řetězce

loadMap
: načtení mapy z obrázku

loadMapCsv
: načtení mapy, uložené programem

CreateWorld
: vytvoří svět z parametrů, zadaných uživatelem (pomocí dialogového okna)

loadBugs
: načte organismy ze souboru

saveBugs
: uloží organismy do souboru

saveMap
: uloží mapu do bitmapy nebo do CSV souboru

Serialize
: uložení/načtení světa

Ovládání světa

doStep
: provede 1 tah ve světě

go
: pustí svět v nekonečné smyčce

step
: pustí jeden tah simulace, pokud je zadáno číslo N jako parametr, pak pustí N tahů

stop
: zastaví simulaci

grow
: každému poli umožní růst potravy

move
: každému organismu umožní jeden krok

Manipulace s obsahem a parametry světa

addBug
: přidá organismus do světa

addBugL
: přidá organismus do světa, svět předtím zamkne

clear
: vyčistí svět

getCell
: vrátí odkaz na konkrétní buňku světa

removeActive
: odstraní aktivní organismus ze světa

removeBug
: odstraní daný organismus ze světa

killActive
: odstraní aktivní organismus ze světa a smaže ho

killBug
: odstraní daný organismus ze světa a smaže ho

setEnergyGain
: nastaví zisk energie z jedné potravy

setFwdPrice
nastaví cenu kroku vpřed

setGlobFert
nastaví globální úrodnost

setLeftPrice
nastaví cenu otočení vlevo

setMaxEnergy
nastaví maximální energii organismu

setMutationProb
nastaví pravděpodobnost mutace organismu

setNonePrice
nastaví cenu akce nedělat nic

setPrice
nastaví cenu zadané akce

setRightPrice
nastaví cenu otočení vpravo

setWaitTime
nastaví dobu čekání organismu

Zamykání světa

lock
čeká, dokud nezíská zámek na světě

unlock
odemkne svět

Získání informací o světě

getAgeVector
vrátí vektor věků všech organismů

getEnergyGain
vrátí zisk energie z jedné potravy

getFoodProcessedVector
vrátí vektor zpracované potravy všech organismů

getForChildVector
vrátí vektor parametrů *pro potomka* všech organismů

getGiveMaxVector
vrátí vektor parametrů *nabízí* všech organismů

getHeight
vrátí výšku světa

getChildCountVector
vrátí vektor počtu dětí všech organismů

getInitialEnergy
vrátí počáteční energii nových organismů

getMaxEnergy
vrátí maximální energii, kterou může organismus mít

getMutationProbability
vrátí pravděpodobnost mutace organismu při vytvoření

getPopulation
vrátí počet organismů

getPrice
vrátí pole cen akcí

getReceiveMinVector
vrátí vektor parametrů *požaduje* všech organismů

getStats
vrátí CSV řetězec s aktuálními statistkami

getStatsHeader
vrátí CSV řetězec s popisy statistických dat

getVectorStats
vrátí CSV řetězec se statistikami daného vektoru

getWaitTime
vrátí čas čekání organismů

getWidth
vrátí šířku světa

loaded
vrátí, zda je svět načten a v pořádku

Pomocné metody

get1Q
vybere první kvartil z vektoru

get3Q
vybere třetí kvartil z vektoru

getAvg
spočítá průměr čísel vektoru

getMax
vybere maximum z vektoru

getMed
vybere medián z vektoru

getMin
vybere minimum z vektoru

getSum
spočítá součet čísel vektoru

doWork
pustí pracovní vlákno, které vykoná tahy ve světě

infiniteLoop
metoda, která vykonává tahy, spouštěna v novém vlákně

logNow
provede zápis statistik do souboru

startLog
začne zapisovat statistiky do souboru

CCell

Atributy

bug
odkaz na organismus, pokud je přítomen

fert
úrodnost pole

toGrow
čas od teď do růstu potravy

growPeriod
interval mezi dvěma růsty potravy

food
indikátor, zda je na poli potrava

Metody

getImage
vrátí obrázek políčka pro vykreslení

grow(globalFertility, useCellProbability, useGrowPeriod)
akce políčka – zmenšení času čekání na růst potravy o jedna, případně samotný růst potravy, vrátí, zda je aktuálně na políčku potrava

putBug(bug)
umístí organismus na toto pole

removeBug()
odstraní organismus z pole

killBug()
odstraní organismus z pole a smaže ho

Serialize(ar)
uloží pole do archivu nebo jej z něj načte

CBug

Konstruktory

CBug()
vytvoří náhodný organismus

CBug(father, mother)
vytvoří organismus zkřížením dvou jiných

CBug(other)
zkopíruje organismus

Metody

toCSV()
exportuje organismus do řetězce pro uložení do CSV

fromCSV(csv)
načte organismus z CSV řetězce

offers()
kolik energie může zaplatit za páření, aby ještě přežil

canAfford()
kolik energie může ještě ztratit, aby přežil

getChildCount()
vrátí počet potomků

isOk(energy)
vrátí, zda je ochoten poskytnout páření za dané množství energie

operator&&(other)
kontrola vhodnosti protějšku k páření

operator>>(other)
provede páření

getPos()
vrátí souřadnice broučka ve světě

step()
provede krok, pokud organismus tímto tahem vyčerpá energii, vrátí *false*, jinak *true*

Příloha D

Obsah CD

bp.pdf
: tento dokument

src.zip
: zdrojové kódy programu

win32.zip
: spustitelný program pro 32bit verzi Windows

x64.zip
: spustitelný program pro 64bit verzi Windows