

Samoorganizující se systémy

D. Roesel

Hálkova 11, 120 00 Praha 2

roesel@gmail.com

Abstrakt

Projekt s cílem od začátku naprogramovat demonstraci samoorganizujícího se systému na příkladu mravenců, kteří přenášejí jehličí v lese. Hlavní snahou bylo dosáhnout programováním jednotlivce organizace na globální úrovni. U finální simulace bylo dbáno na grafické zpracování, reprodukovatelnost výsledků a dostupnost použitých zdrojů.

1 Úvod

Na simulaci samoorganizujících se systémů existují robustní profesionální programy, které umožňují měnit rozsáhlé spektrum parametrů. Ve všech případech se však jedná o už hotové algoritmy, ve kterých není možné kromě parametrů měnit nic dalšího a “do simulace” není vidět.

2 Cíl

Právě z této vlastnosti již napsaných programů plyne hlavní náplň projektu: od začátku si simulaci napsat. Důležité je, aby uživatel mohl dle chuti měnit základní parametry, ale také snadno upravovat vlastní programování jednotlivců.

Hlavním cílem tedy bylo napsat simulaci pohybu mravenců po šachovnici a umožnit jim podle daných pravidel přenášet jehličí (sbírat ho a opět pokládat). Mravenci nesměli mít “kolektivní myšlení” (být řízeni na globální úrovni), ale i přesto mělo jejich působení na šachovnici vést ke zorganizování systému (nakupení jehličí na jedno místo). Simulace musela být prezentována přizpůsobitelným grafickým rozhraním a měla umožňovat změnu základních parametrů i pokročilé zásahy do fungování jednotlivce. V neposlední řadě měl projekt za cíl jednoduchost a srozumitelnost kódu a finální program vytvořit multiplatformní a snadno dostupný.

3 Program

3.1 Základní informace

Program je psán v jazyce Python [1] verze 2.7 a grafické rozhraní zajišťuje knihovna Pygame [2]. Způsob provedení byl zvolen na základě dobré dostupnosti a skvělé přenostielnosti obou prostředků mezi nejrůznějšími systémy. Oba nástroje se navíc dají na základní úrovni velmi snadno ovládat a umožňují tak i začátečníkům snadno provádět úpravy.

3.2 Hlavní běh programu

Program nejprve vytvoří šachovnici o definovaných rozměrech a zacyklí ji. V případě, že by některý z mravenců chtěl opustit plochu simulace nahore, objeví se dole (analogicky všemi směry). Následně na políčka šachovnice umístí náhodně definovaný počet jehliček a mravenců. Poté se spustí hlavní cyklus a postupně prochází každého mravence a náhodně s ním pohne nahoru, dolů, doleva nebo doprava. Tento cyklus se opakuje, dokud uživatel simulaci nezastaví.

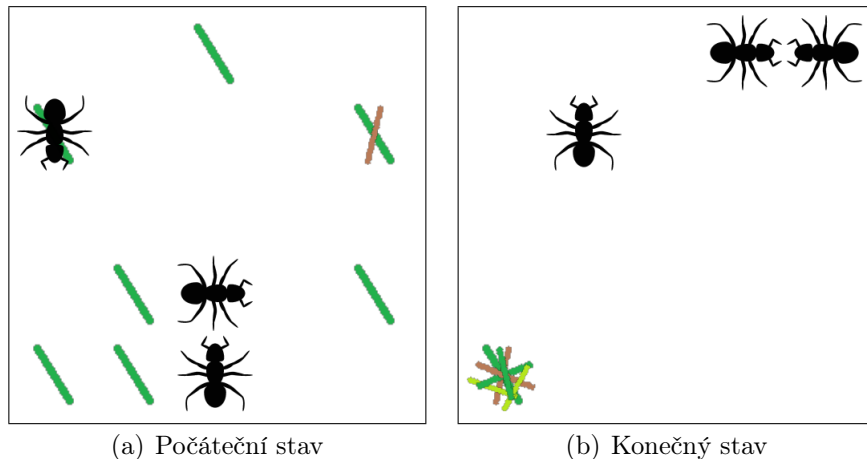
3.3 Chování mravence

Všichni mravenci jsou instance jednoho objektu a chovají se tedy stejně. U každé instance se drží mravencova pozice a to, zda nese jehličku, nebo ne. Po každém přemístění se podívají, zda je na jejich nové pozici na zemi nenulový počet jehliček. Pokud ne, nic neudělají.

V opačném případě ověří, zda už nějakou jehličku nesou, a pokud ano, odloží ji na zem. Pokud ale nic nenesou, jednu jehličku z nalezené hromádky zvednou a čekají na další kolo.

4 Výsledky

Demonstrace se povedla a mravenci úspěšně nanosili jehličky na jedno místo. Simulace trvala v závislosti na parametrech různě dlouho (minuty až hodiny) a vždy vedla buď k postavení jediné věže, nebo k jednomu z níže popsaných případů.



Obr. 1: Ukázka výsledku simulace.

4.1 Více mravenců než tyčinek

V případě, že bylo v parametrech simulace nastaveno více mravenců než tyčinek, neskončila simulace postavením jedné hromádky, ale naopak rozebráním všech hromádek. V jednu chvíli totiž došlo k tomu, že byly všechny jehličky přenášeny mravenci a žádná nezůstala ležet na zemi. V ten moment neměli mravenci podle svého programování právo tyčinky odložit a tím pádem se simulace v této podobě přestala vyvíjet.

4.2 Dvě a tři hromádky

Dalším typickým případem bylo, že se vytvořily dvě či tři hromádky a mravenci neustále nosili jehličky mezi nimi. Podle počtu mravenců a tyčinek může tento stav trvat velmi dlouho, po konečném čase však musí dojít ke statistickému vychýlení a jedna z hromádek musí být rozebrána a navršena na druhou.

5 Diskuse

5.1 Nereálnost

Simulace bere v potaz pouze nejzákladnější vlastnosti mravenců a diametrálně se liší od reality. Mravenci spolu narozdíl od reality nekomunikují žádným způsobem, pohyb mravenců je (pseudo)náhodný (nemají cíl ani preferovaný směr) a nedochází ke kolizím (mravenci chodí přes sebe a přemýšlí po jednom). Asi nejpalcivější problém je, že staví věže místo hromádek – šance na to, že mravenec narazí na jednotlivou tyčinku je stejně velká, jako že narazí na největší hromádku na šachovnici.

5.2 Rychlost programu

Program neběží závratně rychle. Jedním z důvodů je to, že je psán v Pythonu, který se nepyšní největší efektivitou. Dalším (a patrně nejmarkantnějším) důvodem je, že Pygame zobrazuje stav simulace minimálně desetkrát za vteřinu, což běh programu rapidně zpomaluje. V neposlední řadě není kód psán za účelem efektivnosti, ale snaží se být spíše demonstrativní a snadný na pochopení, což se promítá na svižnosti jeho běhu.

5.3 Náměty na zlepšení

Jak už bylo zmíněno v diskusi výše, dalo by se vymyslet programování mravence tak, aby nestavěl věže, ale nějak reálněji vypadající struktury (například hromádky). Kód by se dal na mnoha místech zefektivnit a programu umožnit změny parametrů simulace za běhu, nebo alespoň přes grafické rozhraní. Dalším dobrým nápadem je zkusit mravence vypustit do 3D šachovnice, což by nevyžadovalo přílišné úpravy kódu. Posledním nalezeným problémem je generování náhodných čísel, které probíhá přes pseudonáhodný generátor `randint()`. Pohyb mravenců by se dal totiž zdokonalit použitím čísel náhodnějších, například generovaných vesmírným šumem.

6 Závěr

Projekt dosáhl svého cíle a výsledný program je k dispozici online [4], stejně jako prezentace projektu [5] (obsahující videa z průběhu simulací).

Poděkování

Rád bych poděkoval své sestře za věčné připomínky k programu i prezentaci a trpělivé vyslechnutí nápadů. Dále pak ing. Vojtěchu Svobodovi, CSc. za skvělý předmět a motivaci ke zpracování tohoto tématu.

Reference

- [1] G. Rossum a kol., *Programovací jazyk Python*,
<http://www.python.org/>
- [2] P. Shinnars, *PyGame*,
<http://www.pygame.org/>
- [3] G. Rossum, F. L. Drake, *Python Reference Manual*,
<http://docs.python.org/2/reference/>
- [4] D. Roesel, *Repozitář projektu “mravenci”*,
<http://www.github.com/roesel/mravenci/>
- [5] D. Roesel, *Prezentace “Jak mravenci staví věže”*,
<http://david.roesel.cz/server/mravenci.zip>
- [6] V. Svoboda, *Pravidla proceedings pro FyzSem*,
<http://fyzsem.fjfi.cvut.cz/public/html/proceedings-rules.html>