

Protokol k semestrální práci z BI-ZUM

FIT ČVUT, LS 2019/2020

Jméno studenta: Eliška Svobodová
Username: svoboel5@fit.cvut.cz
Název semestrální práce: Genius Snake

OSNOVA

1. Zadání semestrální práce.
2. Stručný rozbor, analýza problému/ zadání.
3. Výběr metody.
4. Popis aplikace metody na daný problém.
5. Implementace.
6. Reference - odkaz na literaturu/ zdroj, ze kterého čerpáte. Tento bod do protokolu zahrňte pouze v případě, že se rozhodnete v rámci semestrální práce implementovat algoritmus/ metodu nad rámec přednášek z BI-ZUM nebo při tvorbě semestrální práce čerpáte z doporučené literatury.

Pozn: Pokud je metoda přímo definována v zadání semestrální práce, uveďte v tomto bodě pouze „Výběr metody je obsažen v zadání SP.“

1. Zadání semestrální práce.

Implementujte hledání co nejlepší řídicí funkce pro hru Had pomocí genetického programování.

Pravidla hry Had:

Herní pole má tvar obdélníku. Had se v každém kroku posune o políčko dopředu/doleva/doprava ve směru jeho stávajícího pohybu. Snaží se sebrat (a sníst) co největší počet jablek a nenarazit při tom ani do stěn, ani do částí svého těla. Jablka jsou postupně po jednom generována na náhodných prázdných místech herního pole. Za snědení jablka je hadovi připočítán bod a jeho tělo se prodlouží o jedno políčko. Pokud had po určitý počet kroků žádné jablko nesní, vyhladoví k smrti a hra končí. Hra končí, pokud nastane alespoň jeden z následujících případů:

- 1) Had narazí do stěny herního pole nebo do nějaké části svého těla
- 2) Had vyhladoví (provede větší počet kroků než je jeho výdrž)
- 3) Had dosáhne cílového skóre

Řídicí funkce bude hadovi dávat v každém kroku pokyn, jestli má zahnout doleva / zahnout doprava / pokračovat rovně. Kvalita řídicí funkce je dána počtem jablek, které had dokáže po dobu svého života sníst.

2. Stručný rozbor, analýza problému/ zadání.

Lidský hráč ovládá hada na základě vizuální informace o aktuálním stavu hry (pozice hada a jablka, tvar bludiště) a rozhoduje o dalším postupu na základě právě zvolené strategie. Řídicí funkce ale tuto možnost nemá, protože je vyhodnocena v každém kroku nezávisle na svých předchozích výsledcích (je bezstavová) a proto nedokáže plánovat dopředu. Navíc by neměla být závislá na konkrétním tvaru herní plochy, zvláště proto, že součástí bludiště je i hadovo tělo, které dynamicky vytváří překážky. Proto musí řídicí funkce obsahovat části, které jí poskytnou informace o okolí a musí umět na jejich základě rozhodnout o dalším postupu.

3. Výběr metody.

Metoda řešení je dána zadáním – problém bude řešen pomocí genetického programování.

4. Popis aplikace metody na daný problém.

Řídící funkce bude šlechtěna ve formě stromu. Vnitřní vrcholy budou tvořeny funkcemi poskytující hadovi informace o prostředí, listy budou obsahovat terminály představující směr pohybu. V každém kroku bude tento strom vyhodnocen, výsledkem bude vždy jeden terminál, který dá pokyn hadovi, kam se v tomto kroku pohnout. Jedinec v populaci bude tedy tvořen konkrétní řídicí funkcí. Jeho fitness bude skóre dosažené aplikací této funkce na hru.

5. Implementace.

Program obsahuje:

- menu : ze kterého je možno buď spustit hru pro lidského hráče, nebo začít šlechtit řídicí funkci pomocí genetického programování
- hru : implementaci logiky hry, které je podána instance třídy UI
- ui : které je podle potřeby složitější s obrázky (a pomaleji se vykresluje), nebo jednodušší, nebo vůbec žádné (pokud nechceme aby se konkrétní instance hry vůbec vykreslovala)
- kontroler : třída GeneticController představuje jedince v populaci a GeneticProgramming je třída provádějící proces šlechtění
- nastavení : v souboru Settings je celá řada proměnných ovládajících měnitelné vlastnosti programu (smysluplnost zadaných hodnot je ponechána na uživateli)

Řídící funkce:

- terminály : pokyn, kam má had vzhledem k současnému směru jeho pohybu pokračovat
: SNAKE_MOVE_FORWARD, SNAKE_MOVE_LEFT, SNAKE_MOVE_RIGHT
- funkce : přijímající současný stav hry, uzel, který se provede, pokud je funkce vyhodnocena jako True a uzel, který se provede, pokud je funkce vyhodnocena jako False
: bylo experimentováno s různými funkcemi a jejich různými kombinacemi
: např. if_food_forward, if_obstacle_left, if_moving_down, if_wall_right, if_body_left, if_just_eaten_apple, if_obstacle_two_forward, ...
: tyto funkce dávají hadovi informaci, jestli je na nejbližších políčkách nějaká překážka, jestli ta překážka je stěna nebo nějaká část těla, jestli je jablko před hlavou hada nebo kterým směrem se zrovna had pohybuje

Operátory použité pro šlechtění:

- inicializace : byly implementovány tři inicializační operátory, všechny vybírají prvky náhodně
: full (vygeneruje úplný binární strom)
: grow (při generování se v každém uzlu rozhodne podle dané pravděpodobnosti, jestli uzel bude terminal nebo funkce)
: ramped half-and-half (generuje postupně stromy o hloubce 1..max, polovinu metodou full a polovinu metodou grow)
- selekce : program obsahuje jak operátor ruletové selekce podle fitness, tak podle pořadí jedinců v rámci populace
- křížení : při křížení se náhodně vybere uzel z řídicích funkcí rodičů, a v potomcích se vymění podstromy kořenící z těchto uzlů
: šance, že vybraný uzel bude terminál, je nastavitelná v nastavení
- mutace : náhodně se vybere uzel v řídicí funkci, který se nahradí náhodně vygenerovaným podstromem

- náhrada : populace se „ořízne“ pouze na určitý počet jedinců, kteří dosáhli ve hře nejvyššího skóre

Další vlastnosti programu:

- šlechtění je ukončeno pokud nějaký jedinec dosáhne určeného maximálního skóre, nebo se dosáhne maximálního počtu generací (obě tyto hodnoty jsou ovlivnitelné v nastavení)
- při šlechtění se hadi pohybují nejvyšší rychlostí, který výkon počítače umožňuje, protože pro řídicí funkci je zrychlování hada v průběhu hry irelevantní
- různé tvary herní plochy během šlechtění nebyly implementovány, protože řídicí funkce ze své podstaty nemůže být závislá na konkrétním tvaru bludiště
- v nastavení je možné ovlivnit i kolikrát bude řídicí funkce jedince vyzkoušena v rámci generace, např. pokud se každý jedinec bude testovat 3x (jeho fitness se určí jako průměr dosažených skóre), nebudou se ke křížení a postupu do další generace dostávat jedinci, kteří měli štěstí a jen díky tomu nasbírali dostatečný počet jablek
- program je implementován v Pythonu za použití grafické knihovny Pyglet

6. Reference

Inspirace pro aplikaci genetického programování na hru Had:

<https://www.gamedev.net/tutorials/programming/artificial-intelligence/application-of-genetic-programming-to-the-snake-r1175/>

Pravděpodobnost křížení u terminálu-funkce:

http://www.cleveralgorithms.com/nature-inspired/evolution/genetic_programming.html

Inicializační operátory:

<https://books.google.cz/books?>

[id=yQVGAAAAQBAJ&pg=PA254&lpg=PA254&dq=initialize+ramped+half+and+half&source=bl&ots=wSx2p6g6e7&sig=ACfU3U3ME1ixSuwjoKvXvz1YOsMQilq-A&hl=cs&sa=X&ved=2ahUKEwjB2_eEtqvpAhXNyaQKHec-BUkQ6AEwAnoECAQQAQ#v=onepage&q=initialize+ramped+half+and+half&f=false](https://books.google.cz/books?id=yQVGAAAAQBAJ&pg=PA254&lpg=PA254&dq=initialize+ramped+half+and+half&source=bl&ots=wSx2p6g6e7&sig=ACfU3U3ME1ixSuwjoKvXvz1YOsMQilq-A&hl=cs&sa=X&ved=2ahUKEwjB2_eEtqvpAhXNyaQKHec-BUkQ6AEwAnoECAQQAQ#v=onepage&q=initialize+ramped+half+and+half&f=false)