Fakulta Strojního Inženýrství VUT V Brně, Ústav Automatizace a Informatiky

Technická správa k semestrálnemu projektu z predmetu VAI - Algoritmy umělé inteligence

Pyxel Snake

Sára Sujová

182999@vutbr.cz

2020/2021

1. Teoretický popis úlohy

Snake je zaužívané meno pre koncept video hier, ktorý vznikol v roku 1976 ako arkáda Blockade. Hra spočíva v manévrovaní čiarou (hadom) v priestore, ktorý pri zbieraní odmien rastie a stáva sa sám sebe prekážkou. V dnešnej dobe sa koncept rozšíril do mnohých podôb, z ktorých najznámejšia je hra na mobilných telefónoch Nokia.

Základ hry pozostáva z prostredia, pohyblivého hada a formy odmeny (v primitívnejších hrách farebný pixel, v zložitejších vajce, jablko...). V niektorých verziách sa pridávajú prekážky, steny, pohyblivé prekážky atď.

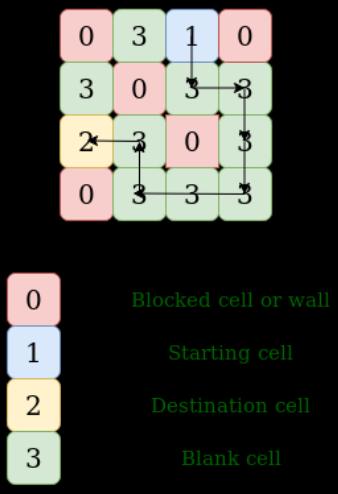
Ukončenie hry nastáva dvomi spôsobmi - had narazí do prekážky, alebo had narazí do svojho vlastného tela.

Úlohou bolo vytvoriť autonómneho Snake Agenta, ktorý by zvládal hrať hru a nahradil tak vstup tlačítok klávesnice/telefónu. Veľa vytvorených Snake Agentov počíta s použitím neurónových sietí (väčšinou na princípe zaznamenávania polohy, dĺžky hada a Manhattanovskej vzdialenosti hlavy od odmeny), čo nebolo v tomto predmete žiadúce, preto boli uvažované algoritmy prehľadávania priestoru.

Kvôli dynamickému prostrediu (pohyb hada) musí výpočet optimálnej trasy prebiehať každý snímok hry. Bol zvolený algoritmus prehľadávania do šírky - Breadth First Search.

Prehľadávanie do šírky je algoritmus prehľadávania stromových grafov, ktorý prechádza všetky vrcholy v danej komponente súvislosti až kým nenájde uzol splňujúci stanovené podmienky. Vždy prechádza všetkých nasledvníkov jediného uzla, až potom prejde na ďalší uzol a prehľadáva jeho nasledovníkov. Pri tom si zaznamenáva cesty k jednotlivým uzlom, z ktorýchje následne získaná najkratšia cesta od koreňa (štartu) k cieli.

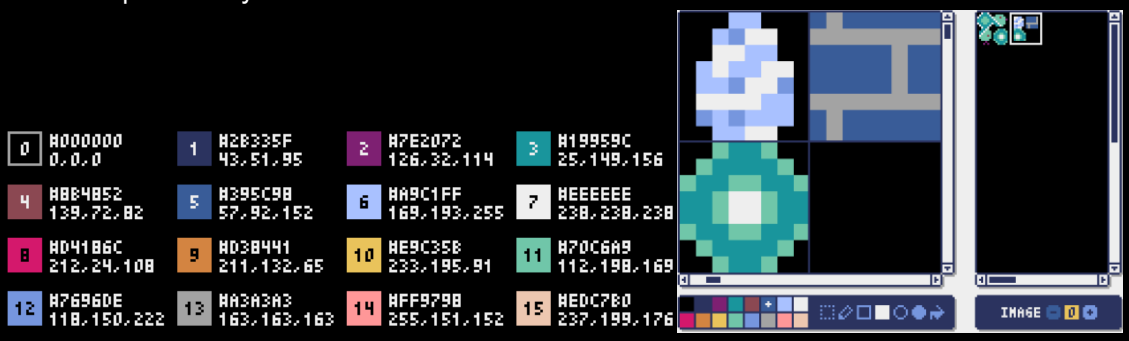
Pre 2D graf tvaru mriežky je možné použiť formu binárneho bludiska, kde sú priechodné bunky označené nulou (0) a nepriechodné bunky označené jednotkou (1) (na obrázku naopak). Bludisko je výhodné kvôli prehľadnosti a prípadnej kontrole zaznamenávať v maticovom tvare. Agent prechádza bludiskom a všetky skontrolované bunky označuje. Zároveň zapisuje ich polohu. Z týchto údajov potom vypočíta najkratšiu cestu k odmene.



2. Popis programu - funkčná časť

Na tvorbu prostredia bol zvolený Pyxel Retro game engine pre jazyk Python. Prostredie je inšpirované vzhľadom retro herných konzolí, tj. môže zobrazovať len 16 farieb a hrať len 4 zvuky naraz. Prostredie beží na systémoch Mac, Linux a Windows, je kompatibilné s Python 3, má 3 image banky o veľkosti 256x256 a rovnako veľkých 8 máp, definované vstupy z klávesnice, myši a gamepadu a editor obrázkov a zvukov. Všetky používané obrázky sa ukladajú do súboru s príponou **.pyxres**, ktorý je súčasťou programu.

Tieto zdroje z Pyxelu sa ďalej načítajú do prostredia a programuje sa s nimi ako s importovaným modulom.



Program má dva súbory a niekoľko tried podľa funkcie. Na začiatku súboru main sú definované smery pohybu a dve skupiny game modov, z toho jedna definuje stav hry - beží/nebeží a druhá kto hrá - AI/Hráč/Nikto - do modu Nikto sa hra vracia po každom ukončení zo strany hráča/štarte aplikácie. Ďalšia trieda vykresľuje tilemapu herného prostredia pomocou funkcie bltm enginu Pyxel. V hre sa vyskytujú dva pohybujúce/premiestňujúce sa objekty - vajíčko a had. Vajíčko sa premiestni na novú náhodnú polohu vždy, keď je zjedené hadom a zaznamenáva pri tom svoje nové súradnice. Had má tri časti - hlavu, stred tela a chvost. Hlava sa vždy pohybuje prvá v smere hry a ďalšie časti hada ju nasledujú.

Samotná aplikácia má na začiatku zadefinované východzie game mody a premenné, počiatočný vzhľad a umiestnenie hada (hlava, 2x telo a chvost), hra je spustená na poli 128x128 pixelov a spustenie hry pomocou enginu Pyxel príkazom run.

Aplikácia má funkciu pre vykresľovanie, ktorá beží na základe príkazu run. Vždy vykreslí level, vajíčko a všetky časti hada v danom momente. Zároveň vypisuje skóre a vykresľuje menu v prípade začiatku novej hry.

Hra v každom snímku kontroluje kolízie vajíčka s hadom, kedy pri „kolízii “ vykreslí nové vajíčko a skontroluje, či je na mieste, kde sa momentálne nachádza had alebo steny. V prípade že áno, priradí vajíčku nové náhodné súradnice. Zároveň hra kontroluje zrážku hlavy hada s telom. V prípade, že dôjde k zrážke, hra končí. Rovnaký princíp je aplikovaný aj v prípade zrážky hada s hranicami herného poľa.

Poslednou dôležitou funkciou je kontrola vstupov prevádzaná na pohyb, ktorá funguje na rovnakom princípe pre Agenta aj Hráča. Vždy je skontrolovaný vstup do funkcie (stlačenie tlačítka/súradnice z výpočtu), a ak nie je táto hodnota v rozpore

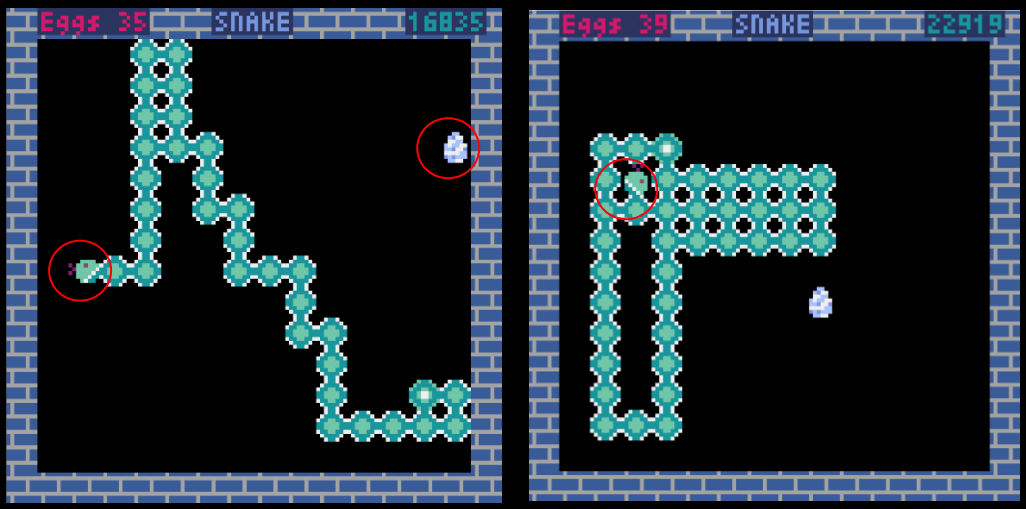
s pravidlami pohybu (keď ide had smerom doprava, nemôže sa otočiť v jednom kroku doľava), zaradí prislúchajúci pohyb do fronty.

Aplikácia BFS algoritmu do hry prebieha v dvoch funkciách. Prvá je súčasťou aplikácie a na začiatku vykoná potrebné výpočty, aby mohlo BFS prebehnúť. Zároveň vykreslí maticu, do ktorej zaznamená polohy prekážok a vytvorí premenné na zapisovanie momentálnych súradníc hlavy hada a vajíčka. Túto maticu potom prevedie na formu zoznamu (listu), s ktorou pracuje program BFS. Po tom, ako dostane z BFS zoznam bodov najkratšej cesty, nájde prvý bod cesty, porovná ho s bodom hlavy a výsledok po odčítaní je smer pohybu. Tento smer používa ako vstup do funkcie na kontrolu vstupov a vykonanie pohybov.

Program BFS v samostatnom súbore načíta premenné z funkcie v aplikácii: maticu, pozíciu hlavy hada a pozíciu odmeny. Algoritmus označí prvú dvojicu súradníc (hlavu hada) ako prečítanú, zapíše si jej súradnice s vrstvou rozširovania, ktorá je v tomto prípade 0. Následne rozšíri bod pozície hlavy do všetkých prípustných smerov a tieto súradnice zapíše do novej vrstvy 0 + 1 = 1. Ďalej sa rozširujú všetky body z vrstvy 1. Potomkom sa vždy pripíše číslo vrstvy o 1 väčšie, ako v predošlej vrstve. Algoritmus zastaví, keď nájde súradnicu odmeny.

V rámci BFS je potrebné nájsť najkratšiu cestu. Táto cesta sa hľadá spätne zo zapísaných dát pri prehľadávaní. Zoznam dcérskych a rodičovských buniek je napísaný tak, že index pre obe bunky je zhodný. Tak je možné nájsť porovnávaním s cieľom (v prvom kroku odmenou) poslednú dcérsku bunku, ktorej index bude zhodný s indexom jej rodičovskej bunky. Táto rodičovská bunka sa stane dcérskou bunkou a hľadá svoju rodičovskú bunku. Celý princíp sa opakuje do doby, než sa prehľadávanie zoznamov dostane na začiatok - k hlave hada. V tom momente je zoznam úplný a po prevrátení pripravený na použitie.

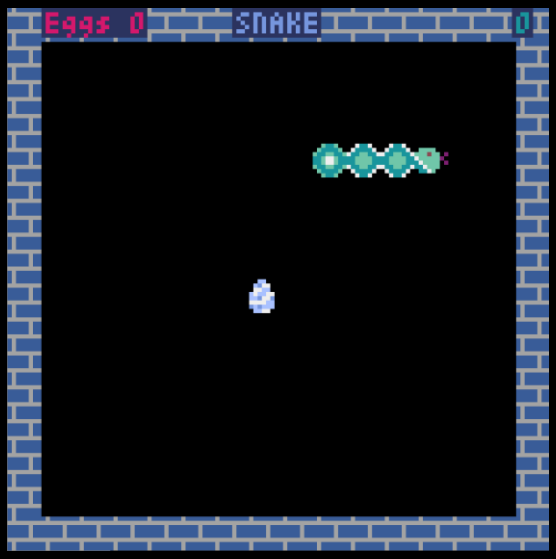
V prípade, že had cestu nenájde(hlava je oddelená od odmeny telom hada, pozri obrázok nižšie), Agent vytvorí cestu zo všetkých prejdených buniek v rámci prehľadávania. V tomto momente by bolo vhodné aplikovať na hada system učenia, aby sa naučil predchádzať podobným situáciám alebo hľadať inak optimálnu trasu, pretože v prípade, že sa mu cesta k odmene neotvorí v rámci tohto procesu, had vyčerpá svoje možnosti a narazí, takže ukončí hru.



3. Popis programu - vzhľad a ovládanie

Prostredie hry sa mení vo vzťahu k stavu hry - Running a Game Over. Východzí stav je Game Over, v ktorom sa na popredí aplikácie zobrazia inštrukcie k voľbe modu hry. Akonáhle hráč vyberie mod, hra sa spustí. Hra beží do doby, než hráč narazí. Potom sa dostane do modu Game Over a je možné znovu vyberať mod hry. Na hornej lište je v pravom hornom rohu počítané skóre, a v ľavom hornom rohu počet vyzbieraných odmien.

Had je ovládaný cez vstupy kláves UP, DOWN, LEFT, RIGHT prevedené na smery pohybu. Na rovnakom princípe funguje aj samostatný Snake Agent.



4. Záver

Cieľom projektu bolo vytvoriť autonómneho Snake Agenta, ktorý by suploval úlohu hráča hry. K tomu bol použitý algoritmus prehľadávania do šírky, Breadth First Search, z ktorého boli späne zistené súradnice najkratšej cesty. Hra sa odohráva v prostredí vytvorenom v Pyxel game engine.

Z testovania Agenta je možné konštatovať, že je schopný hru hrať aj vo vysokých rýchlostiach, ktoré by pre ľudského hgráča boli obtiažne zvládnuteľné. Je teda pravdepodobné, že by Agent porazil hráča v bodovom hodnotení. Zároveň agent pri vysokých rýchlostiach tiež môže naraziť, pretože mu dôjde zásobník, do ktorých sa môže presunúť.

5. Použité zdroje

- *Find whether there is path between two cells in matrix* [online]. 2021 [cit. 2021-6-29]. Dostupné z: https://www.geeksforgeeks.org/find-whether-path-two-cells-matrix/
- *Shortest Path In a Binary Maze* [online]. 2021 [cit. 2021-6-29]. Dostupné z: https://www.geeksforgeeks.org/shortest-path-in-a-binary-maze/
- *Breadth First Search grid shortest path | Graph Theory* [online]. 2018 [cit. 2021-6-29]. Dostupné z: https://www.youtube.com/watch?v=KiCBXu4P-2Y
- *Kitao/pyxel* [online]. 2019 [cit. 2021-6-29]. Dostupné z: https://github.com/kitao/pyxel