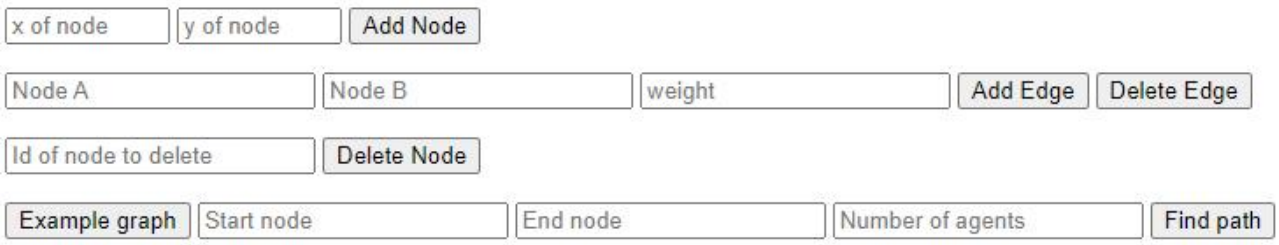
Technická správa

Primitive swarm path-finding / Primitívny hejnový algoritmus

Roman Červenka 208479 4.4.2022

Tento projekt sa zaoberal vyhľadávaním cesty v neorientovanom grafe, ktorý nemusí byť acyklický. Použitý prístup nie je ideálny z hľadiska optimálnej cesty. Cieľom bolo zostrojiť veľmi primitívnu inteligenciu, ktorá však čerpá z početnosti hejna na dosiahnutie daného cieľa. Narozdiel od iných hejnových algoritmov, ako napríklad Ant-colony, v tomto prípade jednotlivci v hejne spolu nekomunikujú až kým nenájdu cieľ.

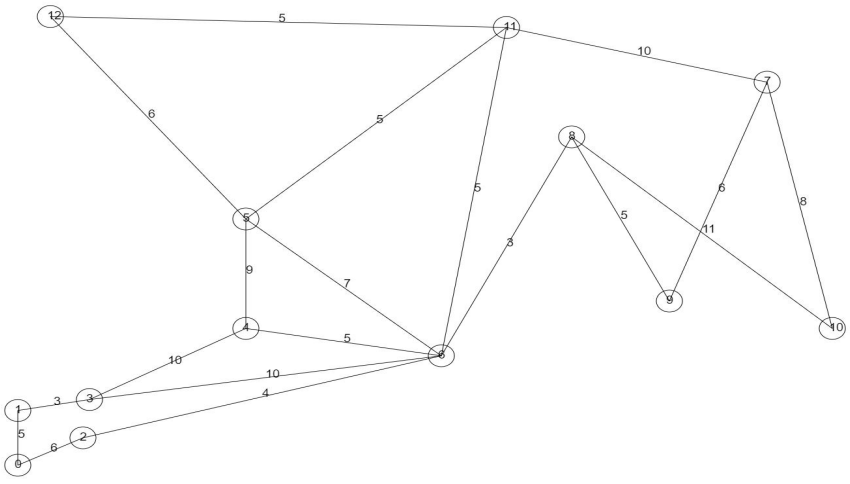
Program bol realizovaný formou web-stránky a teda v jazyku JavaScript a HTML.



Grafické rozhranie programu

Ako možno vidieť GUI obsahuje všetko na vytvorenie grafu, na pridanie uzlu treba uviesť súradnice, a číslovanie začína od nuly automaticky, t.j. prvý uzol bude 0 potom 1 a pod.
Hrany sa pridajú na základe čisla uzlov ktoré majú spojiť a ohodnotenie hrany.
Je možné zmazať hranu a to zadaním čísiel uzlov hrany. Takisto uzol sa zmaže podľa čísla uzlu.
Ďalej tu je tlačidlo ktoré vytvorí predpripravený graf. Následne rozhranie pre vyhľadávanie, počiatočný uzol, cieľový uzol a počet jednotiek hejna. Tento počet musí byť dostatočný, čím viac jednotiek, tým skôr získame optimálne riešenie.

Na nasledujúcom obrázku možno vidieť pripravený graf, je možné ho dodatočne upraviť.

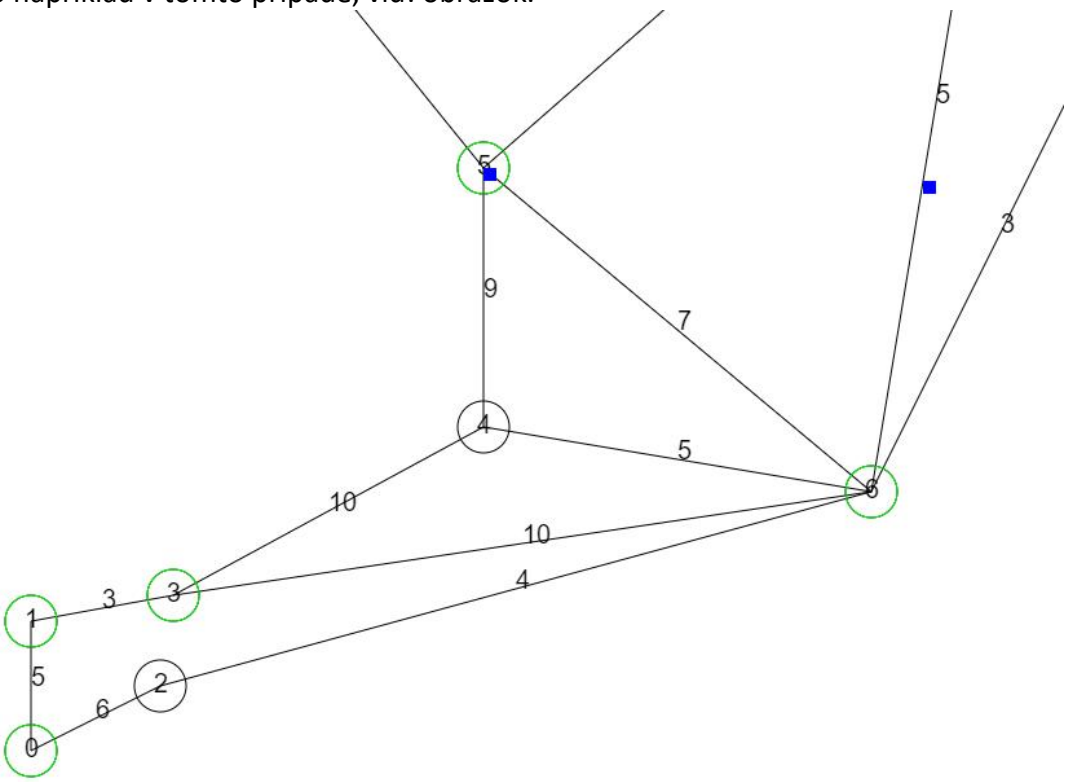


Technická správa

Primitive swarm path-finding / Primitívny hejnový algoritmus

Roman Červenka 208479 4.4.2022

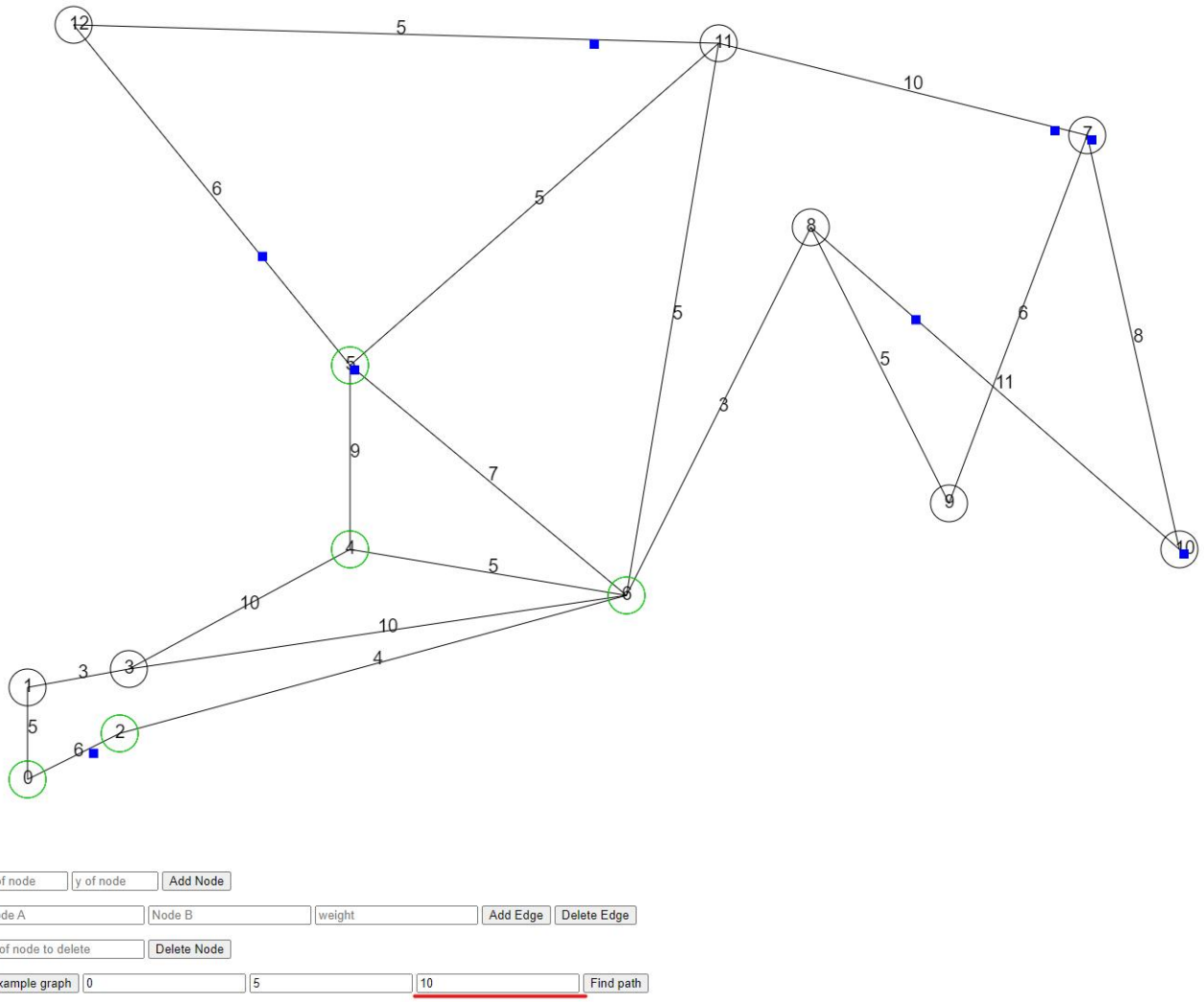
Ako bolo spomenuté v úvode, tento algoritmus nezaručuje vždy optimálny výsledok. Ako napríklad v tomto prípade, viď. obrázok.



Nájdená cesta je 0, 1, 3, 6, 5, kde suma hodnôt je 25, lepšia cesta by však bola 0, 2, 6, 5 kde suma by bola 17. Tento nedostatok je daný náhodnosťou ktorou si jednotlivé jednotky volia cestu.

Uvažovanie týchto jednotiek je nasledovné, ak nie je vybraná cesta/hrana, vyber hranu z ktorá ide z aktuálneho uzla, hrana však nemôže byť rovnaká ako predošlá. Tento jednoduchý predpoklad zabráni tomu aby jednotka cestoval po hrane tam a späť, pričom sa však stále jedná o primitívnu inteligenciu.

Ak jednotka príde do cieľového uzla, “oznámi” to hejnu a simulácia sa zastaví, následne sa označia uzly ktorými víťazná jednotka prešla. Všetky jednotky majú rovnakú rýchlosť a dĺžka cesty je rovná hodnote hrany, z čoho plynie, že daná víťazná jednotka našla najlepšiu cestu zo všetkých jednotiek. Tu je nutné zopakovať, že sa nejedná s istotou o najlepšiu trasu, ale o najlepšiu trasu z tých po ktorých šli jednotky hejna. Z toho opäť plynie potreba dostatočného počtu jednotiek.



Na následujúcom obrázku je možné vidieť ako sa pri zvýšení počtu jednotiek zlepšilo riešenie a našla sa optimálna cesta.

Zhrnutie:

Týmto projektom bol ukázaný možný prístup hľadania cesty v grafe. Je jasné že nie vždy sa nájde optimálne riešenie. Využiteľnosť môže byť však napríklad pre hry alebo simulácie pohybu organizmu v neznámom prostredí, inak povedané strateného človeka.
Jednotky nemajú žiadne povedomie o tom ako graf vyzerá a ani či už v danom uzle boli. Taktiež nepoznajú smer k cieľu.

Cieľom tohto projektu bolo vytvoriť primitívne hejno ako ukážku jednoduchej umelej inteligencie. Ak by šlo o hľadanie optimálnej cesty, bolo by vhodnejšie využiť algoritmy na hľadanie cesty (Djikstra, A*, IDA*,...).