

TILE MAP PATHFINDING

Pavel Pospíšil

Bachelor Degree Programme (1), FEEC BUT

E-mail: xpospi63@stud.feec.vutbr.cz

Supervised by: -

E-mail: -

ABSTRACT

This project describes the theme of pathfinding in simple tile map. It contains a review of some basic pathfinding algorithms, for example blind-search or breadth-first search. Also a remark about heuristic is included. The foremost task is to develop program that is able to find the shortest and easiest path from point A to point B.

1 ÚVOD

Hledání cesty jeden ze základních principů umělé inteligence. Jedná se o prosté nalezení nejlepší cesty z místa A do místa B. Využití algoritmů hledání cesty můžeme najít v robotice, vojenství, počítačových hrách nebo v navrhování plošných spojů. Žádný autonomní robot schopný pohybu v terénu se neobejde bez schopnosti najít správnou cestu. V počítačových hrách má pathfinding velké uplatnění zejména ve strategiích nebo v hrách s počítačem řízenými protihráči. Zde je simulováno chování člověka při hledání cesty nebo se požaduje hledání nejkratších cest.

2 ALGORITMY HLEDÁNÍ CESTY

Algoritmů pro hledání cesty je mnoho. Některé jsou jednoduché a snadno se programují. V některých případech mohou mít velice uspokojivé výsledky, ale ve složitější mapě selžou nebo jsou příliš výpočetně náročné. Příkladem jednoduchých algoritmů je Hledání naslepo (blind-search) nebo hledání metodou rozděl a panuj (divide&conquer).

Pro různé praktické aplikace se potom častěji využívají komplikovanější algoritmy. Jsou mnohem spolehlivější a řeší i problém nalezení nejkratší možné cesty k cíli.

Zde popisované algoritmy pracují s mapovou reprezentací prostředí. Mapou zde rozumíme dvojrozměrné pole čtverců. Dvě buňky mapy jsou označeny za start a cíl. Zbytek je buďto průchozí nebo tvoří překážku. Cestou na mapě rozumíme uspořádanou množinu průchozích sousedících polí.

2.1 HEURISTIKA

Heuristika všeobecně znamená postup, který nemusí nutně vést vždy nejrychleji k nejlepšímu řešení daného problému. Ze zkušenosti či citu však víme, že tento postup vede ve většině případů k řešení efektivněji.

Algoritmy hledání cesty využívají heuristiku, aby eliminovaly prohledávání zbytečného množství cest. Preferovány jsou cesty, které pravděpodobně vedou k cíli. Zpravidla tak dojde k urychlení algoritmu. U „záludných bludišť“, kde cesta vedoucí přímo k cíli, je nakonec slepá, může použití heuristiky znamenat naopak časové ztráty.

2.2 PROHLEDÁVÁNÍ DO ŠÍŘKY (BREADTH-FIRST SEARCH)

Algoritmus začíná ve startovním políčku. V prvním cyklu prohledá jeho sousedy, poté prohledá pozice vzdálené 2 políčka od startu, v dalším cyklu pozice vzdálené 3 políčka a tak pokračuje, dokud nedojde k cíli nebo neprobral všechna dostupná políčka.

Přitom se o každém prohledaném políčku uchová cesta od startu k němu. Když se dostaneme k cíli, pouze přečteme uloženou cestu, která vede jako dosud od startu, ale tentokrát až k cíli. Na podobném principu pracují i další algoritmy.

Protože počet prozkoumaných políček roste kvadraticky s délkou cesty, je dobré prohledávat do šířky jak od startu, tak i od cíle. Postupně provádí vždy jeden cyklus jedné a jeden cyklus druhé šířky. Ve chvíli, kdy se šířky setkají, je možné zkonstruovat cestu ze startu do cíle. Pokud jedna z šířek dojde do stavu, kdy už nemá nová políčka na prozkoumání, pak cesta v mapě neexistuje a algoritmus může skončit.

3 PROGRAM PATHFINDING

Pro praktickou ukázkou algoritmů hledání cesty byl vytvořen program Pathfinding. Je napsán v jazyce Visual Basic. Používá algoritmus prohledávání do šířky. Je možné ohodnotit jednotlivá pole mapy podle obtížnosti jejich průchodu. Program pak hledá nejsnazší cestu k cíli.

3.1 VYTVORENÍ MAPY

Nejprve je třeba určit rozměry mapy, tedy kolik polí je v řadě a sloupci. Pro tvorbu, případně editaci mapy slouží nástrojová lišta. S její pomocí lze jednotlivým polím přidělovat různé vlastnosti. Nejdůležitější je pole označené jako start respektive cíl. Další vlastností je průchodnost pole, které je v náhledu mapy zobrazeno bíle (hladká cesta), černě (překážka) nebo různým stupněm šedi. V algoritmu je průchodnost reprezentována číslem.

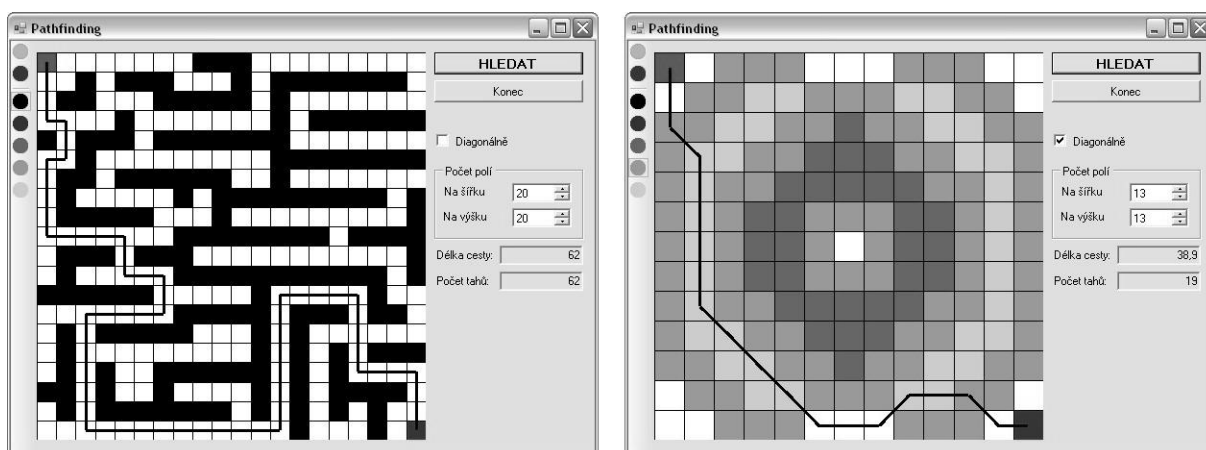
3.2 NALEZENÍ CESTY

Program používá prohledávání do šířky, ale navíc hlídá délku cesty. O každém prohledaném poli ukládá nejen celou cestu od startu, ale i její obtížnost. Při každém kroku pak hlídá, aby obtížnost cesty k němu byla co možná nejmenší.

Před samotným hledáním lze vybrat, jestli cesta může mít i směr diagonály. Pole pak má místo čtyř hned devět sousedních polí. Cesta pak vypadá přirozeněji. Délka šikmého kroku je $\sqrt{2}$ -krát větší než vodorovného nebo svislého.

První testovací mapa je tvořena pouze zdmi a cestami. Jedná se tedy o klasické bludiště. Algoritmus úspěšně našel cestu relativně komplikovaným bludištěm. Zobrazený počet tahů určuje, skrz kolik polí je nutno projít k cíli.

Druhá testovací mapa ukazuje schopnost algoritmu najít nejschůdnější cestu v mapě s různě prostupnými poli. Je velmi těžké na první pohled určit, zda je nalezená cesta opravdu ta nejlepší. Nelze ale přehlédnout, že algoritmus preferuje schůdnější terén (světlejší pole). Zde se pak liší počet tahů od délky cesty, která je zvětšena náročností průchodu.



Obrázek 1: Nalezení cesty v bludišti a ve složitějším terénu

4 ZÁVĚR

Práce představuje stručný přehled základních typů algoritmů hledání cesty. Jsou zmíněny výhody a nevýhody jednotlivých algoritmů. V kostce je přiblížena zajímavá problematika heuristiky. V poslední části práce je popsán program určený k editaci mapy a následném vyhledání nejkratší cesty.

REFERENCE

- [1] BAJER, Lukáš. Algoritmy pro pathfinding. [PDF dokument]. Duben 2006. Dokument dostupný na URL http://ksvi.mff.cuni.cz/~brom/hagents/H-likeAgents4_Bajer060410.pdf.
- [2] HABIBALLA, Hashim. Umělá inteligence. [PDF dokument]. Ostravská univerzita 2004. Dokument dostupný na URL <http://www.volny.cz/habiballa/publ/umint.pdf>.