

Referát do PB016 Úvod do umělé inteligence

**Implementace A* algoritmu na
konkrétní problém orientace v prostoru
budov**

Popis problému

- Jak se dostat z bodu A do B
- Jak se vyhnout překážkám na cestě
- Jak najít nejkratší možnou cestu
- Jak najít příslušnou cestu rychle

Informované hledání cesty

- Při tomto hledání cesty má robot kompletní informace o prostoru
 - V počítačových hrách
 - Pomoc postiženým lidem při orientaci v budově
 - Hledání cest na mapách
 - Vojenská technika

Neinformované hledání cesty

- Například automatické sondy na Marsu. Kvůli zpoždění při komunikaci se sondou, musela být vybavena programy řešící některé situace při pohybu bez pomoci základny.
- Záchranářská technika
- Vojenská technika

Hledání cest v počítačových hrách

- U algoritmů, které se používají v počítačových hrách, se k požadavku nalezení cesty připojují i další kritéria, jako:
 - Rychlost a nenáročnost výpočtu
 - Inteligence pohybu
 - Interakce mezi hráčem a roboty nebo mezi roboty navzájem

Výhody umělého světa (1)

- Svět vytvořený v počítačové hře má na rozdíl od skutečného světa jednu základní výhodu. Můžeme si ho totiž sami vytvořit tak, aby co nejvíce vyhovoval danému algoritmu. Je konečný. Můžeme si před spuštěním programu některé věci přepočítat nebo přednastavit tak, aby samotný výpočet byl co nejrychlejší.

Výhody umělého světa (2)

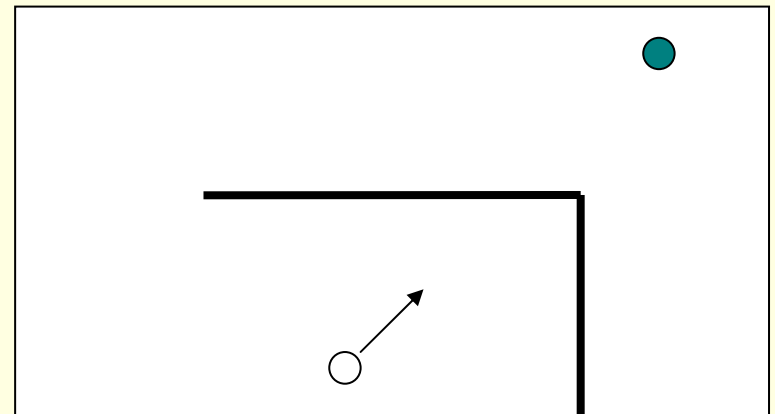
- Herní prostředí se dnes svojí složitostí přibližují realitě, a umožňují proto nejen velmi dobře prověřit jednotlivé techniky umělé inteligence, ale také schopnost tyto techniky integrovat do kooperujícího celku - což představuje zásadní problém, kterému je ve výzkumu akademické AI věnována stále větší pozornost. Na rozdíl od reality lze ale složitost herních světů dle libosti zmenšovat. Je proto možné nejprve vyvinout inteligentní systém, který se vypořádá s jednodušším herním světem, a pak postupně složitost světa i úroveň inteligence zvyšovat.

Využití neinteligentních algoritmů

- Někdy je vhodné z hlediska výkonu využít ten nejjednodušší algoritmus
- Jejich další výhodou je, že robot nemusí znát detailně své okolí, stačí mu jen základní informace, typu dál nemůžu, přede mnou je zed'.

Algoritmus „Přímo k cíli“

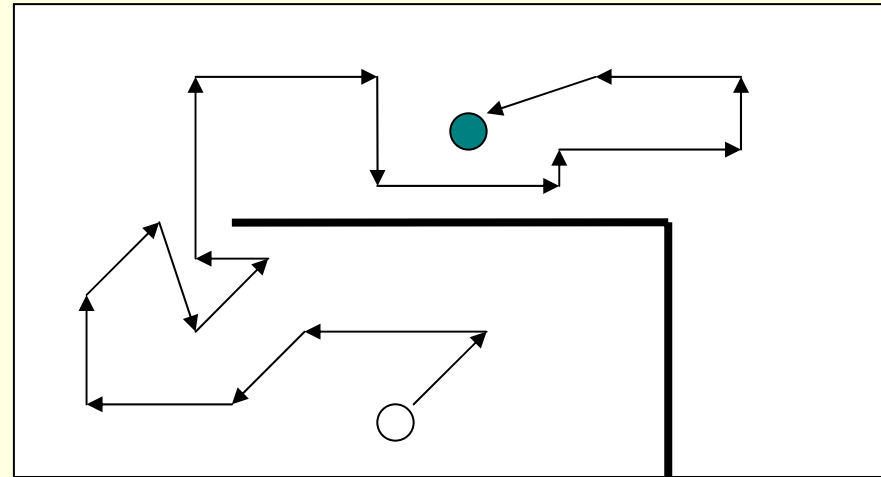
- Nejjednodušší algoritmus vypočítá směr k cíli a po této přímce se robot vydá.
- V každém případě je tento algoritmus základní.



Robot směřuje přímo k cíli, ale pokud hráč neobejde stěnu, nikdy se k němu nedostane

Algoritmus „Náhodného pohybu“

- Tento algoritmu se dá použít i v případě, že robot nezná prostor, ve kterém se pohybuje.
- Navíc lze rozšířit o řadu heuristik, které zvýší pravděpodobnost a rychlost nalezení cesty.



Robot se náhodně pohybuje, dokud nenajde hráče

Využití algoritmů se znalostí prostředí

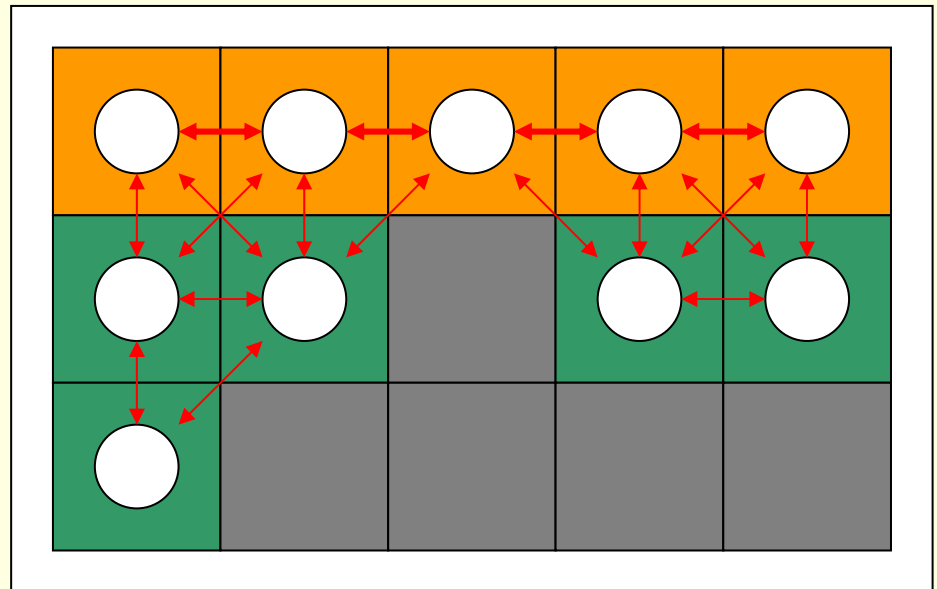
- Prostředí můžeme jej převést na graf, kde uzly jsou křižovatky a hrany cesty, po kterých se může robot pohybovat.

Generování grafů

- Při tvorbě prostředí v počítačových hrách jde logiku tvorby prostředí rozdělit do dvou základních skupin. Podle toho, kterou reprezentaci světa programátor zvolí, se řídí princip tvorby grafů.

Dlaždicové světy

- Jednotlivá pole tvoří uzly grafu a hrany mezi dlaždicemi.
- snadná dynamická změna



Jednoduchá mapka s cestou lesem a skalou.

Světy tvořené vektory

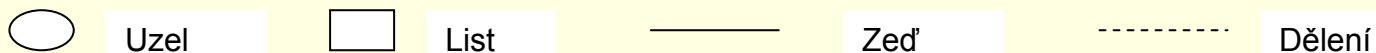
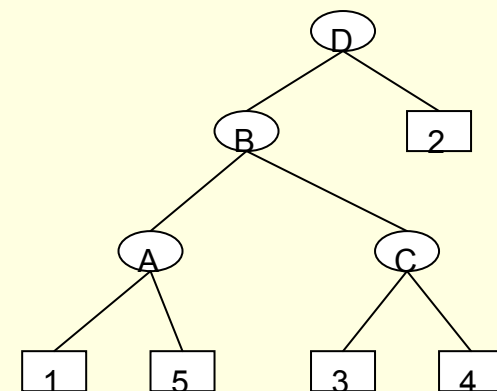
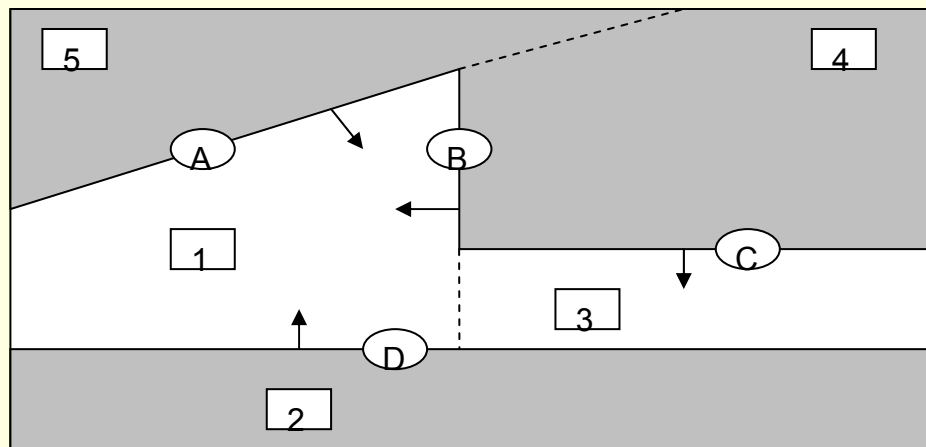
- do plánu takto tvořeného světa rozmístit uzly a hrany tak, aby graf zachycoval podstatu plánu a přitom, aby žádná hrana neprocházela stěnou.
- To znamená, že musíme plán rozdělit na konvexní polygony. Jejich hrany pak rozdělit na stěny a portály

Světy tvořené vektory



Co to jsou BSP stromy a jak nám pomůžou při generování portálů

- BSP stromy (binary space partitioned tree) je technika správy polygonů v prostoru tak, aby umožnila jejich snadné vykreslování
- Takto lze prostor rozdělit na konvexní polygony a snadno v něm vyhledávat.



Příklad mapky, jejího rozdělení a BSP stromu

Použití prohledávacího algoritmu

- Konečně se dostáváme k použití A* algoritmu. Ten potřebuje ke svému fungování odhadovací funkci. Nicméně jedna, celkem kvalitní, je přímý odhad vzdálenosti vzdušnou čarou od místa výskytu robota k jeho cíli. Vzhledem k povaze plánu, který neobsahuje takové rušivé elementy, jako jsou patra, je tato hodnotící funkce vyhovující.
- [AStarSearch.java](#)
- [AStarNode.java](#)

Výhody a nevýhody použití A* algoritmu při prohledávání cesty.

- **Tvar vypočítané trajektorie a chování robotu.**
 - Tvar trajektorie se skládá z přímek, to působí podivně, protože roboti zatačejí pouze v místech portálů.
 - Roboti se mohou rozhodnout vydat k nejbližšímu portálu místo, aby zamířili přímo k cíli
- **Změna prostředí** nebo pohyb cíle si vynutí přepočítání celé cesty.
- **Graf se v průběhu výpočtu mění** A* si do grafu ukládá informace o uzlech, které již prošel, to znamená, že k jednomu grafu nemohou zároveň přistupovat dva algoritmy.
- **Všichni roboti pohybující se podle jednoho grafu mají stejnou trajektorii.** Je zřejmé, že všichni roboti hledající cestu podle jednoho grafu, najdou vždy stejnou cestu.

- Ukázka

Orientace v počítačovém modelu budově

- Základní rozdíl mezi počítačovým modelem reálné budovy a labyrinthem počítačové hry je v tom , že plán budovy nelze tak jednoduše rozložit na konvexní polygony . Rozhodneme-li se tedy použít algoritmus založený na hledání cesty v grafu, musíme vymyslet speciální převod plánu budovy do grafu.

Příklad užití algoritmu pro orientaci v budově

- Pomoci zrakově postiženým lidem při orientaci ve složitých prostorech, jako je například tato budova.
- Při hledání optimální cesty v takových příkladech musí nalezená cesta splňovat mnohem náročnější požadavky.
 - **Její skutečná délka**, je sice důležitý faktor, ale není nejdůležitější.

Příklad užití algoritmu pro orientaci v budově

- **Počet prošlých místností.** Z lidského subjektivního hlediska je lepší cesta, který projde méně místností.
- **Specifické požadavky uživatele.**
- **Závislost na právech uživatele.** Uživatel naopak někdy nemůže projít některou částí budovy .
- **Propojení s budovou.**

Ohodnocovací a odhadovací funkce

- Finální fáze převedení získané cesty do podoby, ve které je nejvíce stravitelná pro člověka. U nás se to řeší pomocí zařízení, které umožňuje hmatem si „osahat“ objekt v počítači.

Reprezentace výsledků

- Finální fáze převedení získané cesty do podoby, ve které je nejvíce stravitelná pro člověka. U nás se to řeší pomocí zařízení, které umožňuje hmatem si „osahat“ objekt v počítači.

-
- Budeme-li procházet modelem budovy pouze v počítači, máme problém jednodušší, než kdybychom ho chtěli použít v robotovi, který se v reálné budově musí opravdu pohybovat. Ten se totiž musí umět vypořádat s neočekávanými situacemi, jako je třeba zatarasená cesta.