

## State Space Search

Po spuštění appletu se na pracovní ploše zobrazí stavový prostor první předpřipravené úlohy:

- Zeleným kroužkem je označen počáteční stav úlohy, který nemůže být změněn.
- Červeným kroužkem je označen cílový stav úlohy. Alespoň jeden cílový stav je vyžadován pro algoritmy/metody BS, Greedy, A\*, Hill-climbing a Simulated Annealing. Pokud je u těchto metod zadáno více cílových stavů, je při prohledávání uvažován pouze cílový stav, který má nejkratší přímou vzdálenost od počátečního stavu.
- Černými kroužky jsou označeny běžné stavy úlohy.

Po spuštění úlohy (tlačítka **Step** nebo **Run**) se uzly ve frontě/zásobníku OPEN vybarví žlutě, kromě uzlu na čele fronty, resp. vršku zásobníku, který se vybarví zeleně – u všech uzlů v OPEN se zobrazí i číslo, udávající aktuální pozici příslušného uzlu v OPEN. Expandované uzly přesunuté do seznamu CLOSED (metody BFS, BS, USC) se vybarví červeně (u metody BS jsou použity dvě fronty OPEN a dva seznamy CLOSED – uzly v OPEN jsou vybarveny stejně, uzly ve druhém seznamu CLOSED jsou vybarveny fialově). Společný uzel (pokud je nalezen) je vybarven růžovou barvou.

### Editace úloh

Tlačítka **Task1** a **Task2** lze nastavit výchozí stavy dvou předdefinovaných úloh, Tlačítkem **Init** se nastaví výchozí stav aktuální úlohy (totéž se stane při jakékoliv akci rozbalovacího políčka, nebo obou přepínačů) a tlačítkem **Clear** se nastaví výchozí stav pro tvorbu nové úlohy (na pracovní ploše zůstane pouze počáteční stav úlohy – zelený kroužek).

Levý přepínač (**Node/Goal**) je aktivní a je zobrazen pouze v případě, že pravý přepínač (**Add/Shift/Remove**) je ve stavu **Add**:

- Je-li levý přepínač ve stavu **Node** a stiskne se levé tlačítko myši uvnitř existujícího uzlu (kolečka) a pokud byl tento uzel uzlem cílovým, změní se na uzel běžný. Stiskne-li se levé tlačítko myši mimo existující uzly, vytvoří se nový běžný uzel. Při stisknutí tlačítka myši je v obou případech možné tažením myši vytvořit hranu do libovolného jiného uzlu. Na posledním řádku pracovní plochy vlevo dole se zobrazí aktuální údaje o příslušném uzlu (viz níže).
- Je-li levý přepínač ve stavu **Goal** a stiskne se levé tlačítko myši uvnitř existujícího uzlu (kolečka) a pokud byl tento uzel uzlem běžným, změní se na uzel cílový. Stiskne-li se levé tlačítko myši mimo existující uzly, vytvoří se nový cílový uzel. V obou případech se levý přepínač vrátí do stavu **Node** a rovněž v obou případech lze tažením myši vytvořit hranu do libovolného jiného uzlu. Na posledním řádku pracovní plochy vlevo dole se zobrazí aktuální údaje o příslušném uzlu (viz níže).

Je-li pravý přepínač ve stavu **Shift** je levý přepínač neaktivní (nezobrazuje se). Stiskem levého tlačítka myši uvnitř existujícího uzlu a následujícím tažením myši lze tento uzel libovolně přemísťovat. Na posledním řádku pracovní plochy vlevo dole se zobrazují aktuální údaje o pozici příslušného uzlu (viz níže).

Je-li pravý přepínač ve stavu **Remove**, je levý přepínač neaktivní (nezobrazuje se). Kliknutím levého tlačítka myši na hranu nebo uzel se tyto objekty odstraní (s uzlem se samozřejmě odstraní i všechny příslušné hrany). Na posledním řádku pracovní plochy vlevo dole se zobrazí aktuální údaje o počátečním uzlu (viz níže).

Při jakékoliv akci tlačítek nebo rozbalovacího políčka se pravý přepínač vrací do stavu **Add** a levý do stavu **Node**.

V rozbalovacím políčku se zvolí metoda (úplný název vybrané metody se zobrazí v pravém horním rohu pracovní plochy), úloha se uvede do výchozího stavu (tlačítka **Task1**, **Task2**, nebo **Init**) a spustí se její řešení – buď tlačítkem **Run**, nebo po krocích stisky tlačítka **Step**.

Varovná a informativní hlášení se zobrazují v levém horním rohu pracovní plochy.

V levém dolním rohu pracovní plochy jsou zobrazeny informace o počtu uzlů a hran na pracovní ploše a na posledním řádku informace o vybraném uzlu:

1. Číslo uzlu,
2. x-ová souřadnice uzlu,
3. y-ová souřadnice uzlu,
4. typ uzlu (běžný/počáteční/cílový ... 0/1/2),
5. umístění uzlu (volný/v OPEN/v CLOSED ... 0/1/2),
6. předchůdce uzlu,
7. - číslo 0 u BFS, BS, GS, Hill-climbing, Simulated Annealing  
- číslo hrany vedoucí do aktuálního následníka u Backtracking  
- hloubka  $l$  u DFS, IDS  
- hodnota  $g$  (součet délek hran z počátečního do aktuálního uzlu) u UCS,  $A^*$
8. - číslo 0 u BFS, BS, DFS, IDS, UCS, Backtracking  
-  $h$  (délka přímé spojnice z aktuálního do cílového uzlu) u Greedy,  $A^*$ , Hill-climbing, Simulated Annealing (pouze u aktivního uzlu)
9. - číslo 0 u BFS, BS, DFS, IDS, Backtracking  
-  $f = g + h$  (tj. součet dvou předchozích hodnot) u UCS, GS,  $A^*$ , Hill-climbing.  
-  $f = h$  u Simulated Annealing (pouze u aktivního uzlu)

### Stručné popisy jednotlivých algoritmů/metod

#### ***BFS (Breadth First Search - Slepé prohledávání do šířky)***

1. Sestroj **frontu** OPEN (bude obsahovat všechny uzly určené k expanzi) a seznam CLOSED (bude obsahovat všechny expandované uzly). Do fronty OPEN umístí počáteční uzel.
2. Je-li fronta OPEN prázdná, pak úloha nemá řešení, a proto ukončí prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber z čela fronty OPEN první uzel a umísti tento uzel do seznamu CLOSED.
4. Je-li vybraný uzel uzlem cílovým, ukončí prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Vybraný uzel expanduj, všechny jeho bezprostřední následníky, kteří nejsou ve frontě OPEN, ani v seznamu CLOSED, umísti do fronty OPEN a vrať se na bod 2.

#### ***DFS (Depth First Search - Slepé prohledávání do hloubky)***

1. Sestroj **zásobník** OPEN (bude obsahovat všechny uzly určené k expanzi) a umísti do něj počáteční uzel.
2. Je-li zásobník OPEN prázdný, pak úloha nemá řešení, a proto ukončí prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber z vrcholu zásobníku OPEN první uzel.
4. Je-li vybraný uzel uzlem cílovým, ukončí prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Vybraný uzel expanduj, všechny jeho bezprostřední následníky, kteří nejsou v zásobníku OPEN a nejsou ani předky generovaného uzlu, umísti do zásobníku OPEN a vrať se na bod 2.

### ***IDS (Iterative Deepening Search - Slepé prohledávání iterativním zanořováním)***

1. Zadej maximální hloubku prohledávání.
2. Nastav aktuální hloubku prohledávání na hodnotu 1.
3. Zavolej proceduru DLS s počátečním a koncovým stavem (seznamem koncových stavů) a s omezením na aktuální hloubku. Skončí-li tato procedura úspěchem (tj. vrátí cestu od počátečního uzlu k uzlu cílovému), ukonči prohledávání jako úspěšné (vyznač cestu nalezenou procedurou DLS). Jinak pokračuj.
4. Inkrementuj aktuální hloubku.
5. Je-li nová aktuální hloubka menší než zadaná maximální hloubka, vrať se na bod 3. Jinak ukonči prohledávání jako neúspěšné.

### ***Procedura DLS (Depth Limited Search – Slepé prohledávání do omezené hloubky)***

1. Sestroj **zásobník** OPEN (bude obsahovat všechny uzly určené k expanzi) a umísti do něj počáteční uzel.
2. Je-li zásobník OPEN prázdný, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber z vrcholu zásobníku OPEN první uzel.
4. Je-li vybraný uzel uzlem cílovým, ukonči prohledávání jako úspěšné a vrať cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Je-li hloubka vybraného uzlu menší než zadaná maximální hloubka, tak tento uzel expanduj, všechny jeho bezprostřední následníky, kteří nejsou jeho předky a nejsou dosud v zásobníku OPEN, umísti do zásobníku OPEN a vrať se na bod 2.

### ***BS (Bidirectional Search - Obousměrné prohledávání)***

1. Sestroj **fronty** OPEN1 a OPEN2 (budou obsahovat všechny uzly určené k expanzi) a seznamy CLOSED1 a CLOSED2 (budou obsahovat všechny expandované uzly). Do fronty OPEN1 umísti počáteční uzel a do fronty OPEN2 cílový uzel.
2. Je-li fronta OPEN1 prázdná, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber z čela fronty OPEN1 první uzel a umísti tento uzel do seznamu CLOSED1.
4. Vybraný uzel expanduj. Pokud některý bezprostřední následník je prvkem fronty OPEN2 (je tzv. „můstek“), ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému (v CLOSED1 jsou uzly od počátečního stavu k můstku, v CLOSED2 jsou uzly od můstku k cílovému stavu), jinak ulož tohoto následníka do fronty OPEN1.
5. Vyber z čela fronty OPEN2 první uzel a umísti tento uzel do seznamu CLOSED2.
6. Vybraný uzel expanduj. Pokud některý bezprostřední následník je prvkem fronty OPEN1 („můstek“), ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému (v CLOSED1 jsou uzly od počátečního stavu k můstku, v CLOSED2 jsou uzly od můstku k cílovému stavu), jinak ulož tohoto následníka do fronty OPEN2 a jdi na bod 2.

### ***UCS (Uniform Cost Search - Slepé prohledávání do šířky s respektováním stejných cen)***

1. Sestroj **seznam** OPEN (bude obsahovat všechny uzly určené k expanzi) a seznam CLOSED (bude obsahovat všechny expandované uzly). Do seznamu OPEN umísti počáteční uzel včetně jeho (nulového) ohodnocení.
2. Je-li seznam OPEN prázdný, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber ze seznamu OPEN uzel s nejnižším ohodnocením a umísti tento uzel do

seznamu CLOSED.

4. Je-li vybraný uzel uzlem cílovým, ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Vybraný uzel expanduj a ohodnoť všechny jeho bezprostřední následníky (cena generovaného uzlu = cena expandovaného uzlu + vzdálenost mezi oběma uzly v pixelech), kteří nejsou v seznamu CLOSED a ulož je do seznamu OPEN. Z uzlů, které se v seznamu OPEN vyskytují vícekrát, ponech v tomto seznamu pouze uzel s nejnižším ohodnocením a vrať se na bod 2.

### ***Backtracking (Slepé prohledávání se zpětným navracením)***

1. Sestroj **zásobník** OPEN (bude obsahovat všechny uzly určené k expanzi) a umísti do něj počáteční uzel.
2. Je-li zásobník OPEN prázdný, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Jde-li na uzel na vrcholu zásobníku aplikovat první, resp. jiný dosud neaplikovaný operátor, tak tento operátor aplikuj a pokračuj bodem 4, v opačném případě testovaný uzel z vrcholu zásobníku odstraň a vrať se na bod 2.
4. Je-li vygenerovaný uzel (uzel vzniklý aplikací operátoru) uzlem cílovým, ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak ulož nový uzel na vrchol zásobníku (pouze však, pokud se tento uzel již v zásobníku OPEN nenachází) a vrať se na bod 2.

### ***GS (Greedy Search - Informované „hltavé“ prohledávání)***

1. Sestroj **seznam** OPEN (bude obsahovat všechny uzly určené k expanzi) a seznam CLOSED (bude obsahovat všechny expandované uzly). Do seznamu OPEN umísti počáteční uzel včetně jeho ohodnocení (cena uzlu = přímá vzdálenost k cílovému uzlu v pixelech).
2. Je-li seznam OPEN prázdný, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber ze seznamu OPEN uzel s nejnižším ohodnocením.
4. Je-li vybraný uzel uzlem cílovým, ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Vybraný uzel expanduj, ohodnoť všechny jeho bezprostřední následníky, kteří nejsou jeho předky a ulož je do seznamu OPEN (cena uzlu = přímá vzdálenost k cílovému uzlu v pixelech). Z uzlů, které se v seznamu OPEN vyskytují vícekrát, ponech v tomto seznamu pouze uzel s nejnižším ohodnocením a vrať se na bod 2.

### ***A\* (A Star Search - Informované prohledávání „A s hvězdičkou“)***

1. Sestroj **seznam** OPEN (bude obsahovat všechny uzly určené k expanzi). Do seznamu OPEN umísti počáteční uzel spolu s jeho ohodnocením (ohodnocení počátečního uzlu  $f_0 = g_0 + h_0$ , kde  $g_0 = 0$  a  $h_0$  = přímá vzdálenost počátečního uzlu k cílovému uzlu v pixelech).
2. Je-li seznam OPEN prázdný, pak úloha nemá řešení, a proto ukonči prohledávání jako neúspěšné. Jinak pokračuj.
3. Vyber ze seznamu OPEN uzel s nejnižším ohodnocením.
4. Je-li vybraný uzel uzlem cílovým, ukonči prohledávání jako úspěšné a vyznač cestu od počátečního uzlu k uzlu cílovému. Jinak pokračuj.
5. Vybraný uzel expanduj, ohodnoť všechny jeho bezprostřední následníky, kteří nejsou jeho předky a ulož je do seznamu OPEN (cena generovaného uzlu  $f_{gu} = g_{gu} + h_{gu}$ , kde

$g_{gu} = g_{eu} + \text{vzdálenost mezi oběma uzly (generovaným/potomkem } gu \text{ a expandovaným/rodičem } eu) \text{ v pixelech}$  a  $h_{gu}$  je přímá vzdálenost mezi generovaným uzlem a cílovým uzlem v pixelech). Z uzlů, které se v seznamu OPEN vyskytují vícekrát, ponech v tomto seznamu pouze uzel s nejnižším ohodnocením a vrať se na bod 2.

### ***Hill-Climbing (Horolezecký algoritmus – lokální prohledávání)***

1. Vytvoř uzel Current a ulož do něj počáteční uzel spolu s jeho ohodnocením (ohodnocení = přímá vzdálenost k cílovému uzlu v pixelech).
2. Expanduj uzel Current, ohodnot' jeho bezprostřední následníky a vyber z nich nejlépe ohodnoceného (nazvěme jej Next).
3. Je-li ohodnocení uzlu Current lepší než ohodnocení uzlu Next, ukonči řešení a vrať jako výsledek uzel Current (vyznač cestu od počátečního uzlu k uzlu Current). Jinak pokračuj.
4. Ulož uzel Next do uzlu Current a vrať se na bod 2.

### ***Simulated Annealing (Simulované žíhání – pravděpodobnostní prohledávání)***

1. Vytvoř předpis pro klesání teploty  $T$  v závislosti na kroku výpočtu  $k$  (v implementovaném algoritmu  $T = \max(0, 200-k)$ ).
2. Vytvoř pracovní uzel Current a ulož do něj počáteční uzel spolu s jeho ohodnocením (ohodnocení = přímá vzdálenost k cílovému uzlu v pixelech). Nastav krok výpočtu na nulu ( $k = 0$ ).
3. Je-li uzel Current uzlem cílovým, ukonči řešení jako úspěšné a vyznač cestu od počátečního uzlu k uzlu Current.
4. Je-li  $k > k_{\max}$  ukonči řešení, ukonči řešení jako neúspěšné a vyznač cestu od počátečního uzlu k uzlu Current.
5. Expanduj uzel Current a z jeho bezprostředních následníků vyber náhodně jednoho z nich (nazvěme jej Next).
6. Vypočítej rozdíl ohodnocení uzlů Current a Next:  

$$\Delta E = \text{ohodnocení(Next)} - \text{ohodnocení(Current)}.$$
7. Jestliže  $\Delta E < 0$ , tak ulož uzel Next do uzlu Current, jinak ulož uzel Next do uzlu Current s pravděpodobností  $e^{-\Delta E/T}$ .
8. Inkrementuj krok výpočtu  $k$  a vrať se na bod 3.