

KSI 2012

Úloha 1-1-2: Karel horolezcem

Jan Horáček
Gymnázium, Brno, Vídeňská 47; jan.horacek@seznam.cz

19. září 2012

1 Úvod

Toto řešení úlohy "Karel horolezcem" je založeno na rozdělení algoritmu do dílčích funkcí, které jsou opakovaně volány.

Program předpokládá:

- robota Karla umístěného na libovolný řádek, ze kterého vystupuje pohoří.
- domov Karla umístěný na spodním řádku za všemi pohořími. Zde robot Karel skončí svou cestu.
- takové pohoří, které je možno přejít.

2 Zdrojový kód

Zdrojový kód k tomuto problému je přiložen v souboru *Reseni2.txt* v kódování UTF-8 a byl vytvořen <http://ksi.fi.muni.cz/karel/> osobně.

3 Popis řešení

Program se spouští funkcí *HOROLEZ*.

Následuje kompetní slovní popis funkce programu - od dílčích funkcí až k celku.

3.1 Funkce *DOLE_START*

Funkce *DOLE_START* zajistí úspěšný přesun Karla z výchozí pozice k nejbližší zdi, tedy k začátku pohoří.

3.2 Funkce *NAHORU*

Funkce nahoru předpokládá robota umístěného čelem k pohoří a jejím výstupem je robot umístěný v prvním "sedle", či vrcholu pohoří. Ať už tento bod nazveme jakkoliv, prakticky se jedná o první bod, ve kterém Karel na zdi stojí (nikoliv šplhá po zdi).

Na začátku funkce je známo, že je před Karlem zeď (předpoklad). Karel provede výstup o 1 políčko nahoru a zkontrolujeme, zda-li je vedle Karla stále zeď. Jakmile se vedle Karla zeď nenachází, dojde k přesunu robota na toto prázdné místo. Pokud před Karlem není zeď, robot pokračuje o pole výš.

3.3 Funkce *NAHORE*

Funkce *NAHORE* se stará o přesun robota Karla po horní části pohoří. Ukončí se, jakmile má Karel sestupovat.

Nyní je Karel v "sedle". Dojde ke kontrole, zda-li je před Karlem zeď. Pokud ano, vyšplhá robot do následujícího sedla. (Jelikož funkce *NAHORU* před svým ukončením nepředá řízení další funkci, máme jistotu, že se program nachází stále ve funkci *NAHORE*) Jakmile je před robotem prázdné pole, provede se krok na toto pole. Vyjděme z předpokladu, že se robot pohybuje po volné rovině. Proto je implemetován cyklus, který skončí jakmile je robot na poli, pod kterým se nenachází obsazené pole, tedy pokud je "ve vzduchu". Pokud ovšem robot ve vzduchu není, pokračuje dál ve své cestě: kontroluje se, zda-li před ním není hora, na kterou by potřeboval vyšplhat. Pokud ano, funkce *NAHORU* to zajistí. Celá funkce tedy skončí, jakmile robot úspěšně přejde vrchoky a dostaví se na políčko, pod kterým je prázdné pole.

Jistě jste si všimli, že ve funkci jsou 2 zdánlivě podobné cykly. Tato konstrukce je zcela záměrná a zajišťuje časovou efektivitu tím, že se na prvním poli ("sedlo") nekontroluje, zda-li se pod Karlem nachází zeď, protože by tato kontrola byla zcela zbytečná.

3.4 Funkce *DOLU*

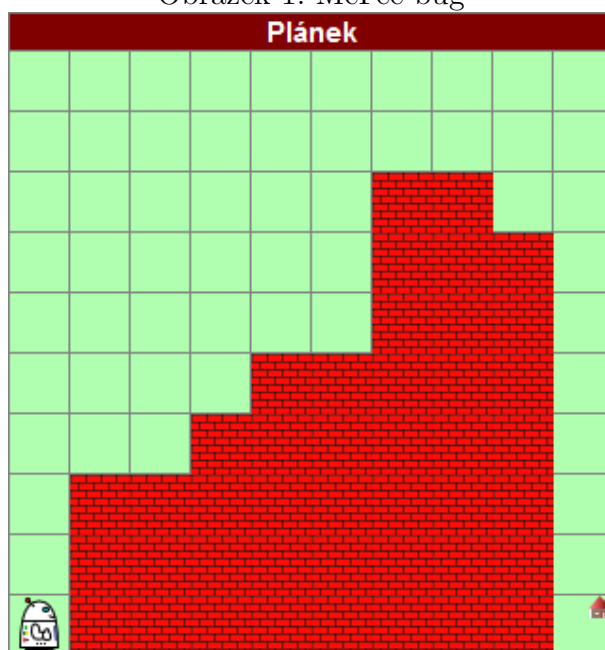
Funkce *DOLU* zajistí úspěšný přesun Karla z vrcholků hor na nejnižší bod daného sloupce.

3.5 Funkce *DOLE*

Karel se nyní nalézá buď v sedle při cestě dolů, anebo na spodním řádku plánku - tedy už pod pohořím. Je otočený směrem k cíli - tedy doprava. Tato funkce zajistí přesun Karla přes další možná pohoří až do domova.

Základní myšlenkou této funkce je neustálé hledání cesty dolů. Celá Karlova cesta v této funkci je ale zapouzdřena do cyklu "DOKUD NENÍ DOMOV" a to proto, aby Karel včas skončil s horolezením a nepřešel domov. Uvnitř cyklu Karel provede krok směrem doprava a funkce zkontroluje, zda-li náhodou pod Karlem není prázdné pole. Pokud ano, zavolání funkce "DOLU" zajistí přesun Karla na pevnou zem. Ale před robotem samozřejmě může být také další hora, takže je potřeba tuto skutečnost kontrolovat. Pokud zde hora je, provedeme sérii již známých funkcí, které zajistí přesun Karla do prvního sedla při sestupu.

Obrázek 1: McFee bug



Podmínka "KDYŽ NENÍ DOMOV" musela být do programu přidána, protože kdyby plánek vypadal podobně, jako na obr. 1, algoritmus by skončil s chybou. To je způsobeno tím, že hranice hracího plánu se chovají jako zeď.

3.6 Funkce *HOROLEZ*

Tato funkce tvoří pojítka mezi dílčími funkcemi. Touto funkcí se celý program spouští.

4 Závěr

Ačkoliv se jedná o rozcvičku na reálné příklady, připadá mi, že tento příklad nebyl zcela triviální. A hlavně jeho slovní popis.

Pevně doufám, že jsem ošetřil všechny speciální případy a tedy, že je program plně funkční. Principiálně je algoritmus funkční a použitelný na každý validní plánek.

5 Poděkování

Děkuji Martinu Florianovi (McFee) za testování algoritmu.

Reference

Tento algoritmus byl vytvořen pouze autorovou vlastní silou - bez použití externích zdrojů.