FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

KPG

GENEROVÁNÍ BLUDIŠTĚ

Martin Hamet

8. března 2016

Obsah

1 Zadání

Vygenerujte podle algoritmů předvedených na přednášce bludiště a doplňte h o o libovolnou přidanou funkčnost.

2 Analýza úlohy

Pro realizaci jsem vybral bludiště jednoduchého tvaru mřížky kvůli snazší manipulaci a pochopení zkoušených algoritmů. Nejprve je tedy třeba rozhodnout o reprezentaci bludiště, aby bylo možné ho snadno generovat případně dále měnit.

Pro zpestření výsledného bludiště by se mohlo vnést otevření náhodných stěn bludiště pro větší variabilitu řešení. Toho lze docílit snadno například při inicializaci matice, kde jsou všechny stěny zavřené, některé otevřeme. Otevření stěn by mělo být řízeno vygenerováním náhodného čísla.

2.1 Reprezentace bludiště

Samotné bludiště bude vhodné rozdělit na jednotlivé buňky, kde každá buňka má čtyři stejně dlouhé stěny. Nabízí se uložení každé buňky jako samostatné třídy do dvourozměrného pole. Pokud by každá buňka obsahovala informaci o všech svých čtyřech stěnách, každé dvě sousední buňky by obsahovali stejnou informaci o jedné stěně která je dělí, proto bude vhodnější v každé buňce obsáhnout informaci pouze o jejích dvou stěnách. V mém případě jsem zvolil vrchní a levou.

Tímto vzniká problém s uzavřením (v mém případě) pravé strany bludiště, protože informace o pravé stěně jedné buňky je uložena v buňce sousední (napravo). Tento problém snadno vyřešíme přidáním jednoho sloupce a řádku které nebudou považovány za funkční buňky vlastního bludiště. Pro snazší manipulaci při zjišťování sousedních buněk v matici přidáme jeden sloupec a řádek i na obě strany matice. Zajistíme tím že nebude nutné zkoumat v průběhu zda jsme na okraji bludiště. Tyto buňky bude nutné označit jako neměnné a nebudou součástí bludiště.

2.2 Algoritmus generování

Pro generování bude jednoduší provést inicializaci všech buněk a nastavit všechny jejich stěny na uzavřené. Dále označíme všechny buňky v přidaných pomocných sloupcích za navštívené. Při průchodu se tedy nebudou uvažovat jako kandidáti pro cestu a zůstanou po celou dobu uzavřené. Algoritmus bude tedy procházet matici postupně po nalezených nenavštívených buňkách a bude korektně otevírat jejich stěny. Každou buňku, kterou algoritmus navštíví a otevře do ní stěnu musí označit za navštívenou, aby nedocházelo k zacyklení.

Snadno se může stát že při průchodu algoritmus uvízne ve "slepé uličce"a je potřeba aby navázal někde na již vytvořené cestě odkud je možné pokračovat jinam. Zde můžeme využít např. zásobníku nebo fronty pro ukládání všech nalezených sousedů, aby bylo možné snadno vybrat buňku pro navázání cesty. Pro snazší určení směru napojení si k buňce uložíme z jakého směru byla nalezena.

Zásobník se bude chovat způsobem odbočení při nejbližší možnosti pokud by jsme se pohybovali na vytvořené cestě zpět. Fronta by naopak vytvářela odbočení co nejblíže směrem k začátku na vytvořené cestě. Z hlediska obtížnosti bludiště pro uživatele by se zdála fronta jako obtížnější, protože by podle mého názoru mohlo vznikat delší slepé cesty. V této implementaci jsem nakonec zvolil zásobník.

Algoritmus končí pokud projde všechny nenavštívené buňky matice a tím tedy vytvoří bludiště.

2.2.1 Shrnutí algoritmu

1. Vlož počáteční buňku do zásobníku.
2. Vyber buňku ze zásobníku a odstraň stěnu ve směru ve kterém byla nalezena.
3. Vybranou buňku označ za navštívenou.
4. Přidej všechny nenavštívené sousedy vybrané buňky (tj. ve vodorovném a svislém směru) do zásobníku v náhodném pořadí a nastav jim směr ze kterého byli nalezeni.
5. Pokud není zásobník prázdný pokračuj na bod 2.
6. Pokud je zásobník prázdný algoritmus končí.

3 Uživatelská příručka

Po spuštění programu je nejprve potřeba zadat rozměry horizontální a vertikální požadovaného bludiště. Tyto rozměry jsou udávané v počtu buněk. Tlačítkem `Create` vytvoříme bludiště podle zadaných parametrů.

Dále je možné upravit parametr `Intensity of random opens`, který udává maximální počet náhodně otevřených stěn při inicializaci matice. Tento parametr lze nastavovat od 0 do 1000. K otevření ale dochází náhodně a proto nemusí při menších hodnotách k žádnému otevření dojít. Velká hodnota tohoto parametru může zajistit velmi "řídké"bludiště a tím snadnou cestu k cíli.

Pokud zaškrtneme přepínač `Gradual plot` potom generování bludiště bude vykreslované pomalu tak jak ho prochází algoritmus a otevírá stěny. Není zajištěno žádné omezení pro tuto volbu a tedy pro bludiště o velkých rozměrech může takové generování zabral dlouhou dobu, proto je doporučeno pouze pro menší rozměry.

Tlačítko `Play` vykreslí hráče (Modrý kruh) na startovní pozici. Pomocí kláves `W,S,A,D` můžeme ovládat jeho pohyb. Pokud se s hráčem dostaneme do cílové pozice označené zeleným čtvercem "hra"končí. Je možné začít od začátku opětovným stisknutím `Play`. Za hráčem se tvoří vodící čára kudy již hráč prošel zejména v rozměrných bludištích usnadňuje hledání cesty. Tato čára se smaže při restartovaní "hry"tlačítkem `Play`

4 Závěr

Program je implementován pouze pro ilustrační účely a je zde prostor pro optimalizaci zejména po paměťové stránce. Pro další použití by bylo vhodné doimplementovat funkci "zoom"pro velké rozměry bludiště. Pomocí funkce `Gradual plot` se dá snadno sledovat postup algoritmu především při jeho upravování. Program je tedy vhodný pro zkoušení různých algoritmů, které ale v této práci nebyli zatím implementovány z časových důvodů.