155ADKG: Geometrické vyhledávání bodu

Datum odevzdání: 13.11.2017

Petra Millarová, Bc. Oleksiy Maybrodskyy

Contents

1 Zadání

Následuje kopie oficiálního zadání úlohy. Autoři z nepovinných bodů zadání implementovali všechny kromě algoritmu pro automatcké generování nekonvexních polygonů.

Úloha č. 1: Geometrické vyhledávání bodu

Vstup: Souvislá polygonová mapa n polygonů $\{P_1, ..., P_n\}$*, analyzovaný bod* q*.*

Výstup: P_i, $q \in P_i$.

Nad polygonovou mapou implementujete následující algoritmy pro geometrické vyhledávání:

- Ray Crossing Algorithm (varianta s posunem těžiště polygonu).
- Winding Number Algorithm.

Nalezený polygon obsahující zadaný bod q graficky zvýrazněte vhodným způsobem (např. vyplněním, šrafováním, blikáním). Grafické rozhraní vytvořte s využitím frameworku QT.

Pro generování nekonvexních polygonů můžete navrhnout vlastní algoritmus či použít existující geografická data (např. mapa evropských států).

Polygony budou načítány z textového souboru ve Vámi zvoleném formátu. Pro datovou reprezentaci jednotlivých polygonů použijte špagetový model.

Hodnocení:

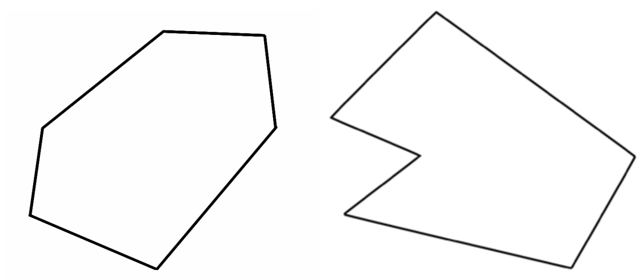
Krok	**Hodnocení**
Detekce polohy bodu rozlišující stavy uvnitř, vně na hranici polygonu.	10b
Ošetření singulárního případu u Winding Number Algorithm: bod leží na hraně polygonu.	*+2b*
Ošetření singulárního případu u obou algoritmů: bod je totožný s vrcholem jednoho či více polygonů.	*+2b*
Zvýraznění všech polygonů pro oba výše uvedené singulární případy.	*+2b*
Algoritmus pro automatické generování nekonvexních polygonů.	*+5b*
Max celkem:	**21b**

2 Popis a rozbor problému

Tato úloha se věnuje řešení praktického problému určování pozici zvoleného bodu q, vůči zadaným polygonům. Jako implementaci lze nejjednodušeji představit zjišťování polohy jistého bodu kliknutím na digitální mapě.

Nechť existuje pole ve dvojrozměrné kartézské soustavě s ***n*** *body. Uzavřením tohoto pole vznikne polygon. Polygon může nabývat jak konvexní, tak nekonvexní tvar.*

Polygon je konvexní právě tehdy, když poloha všech bodů je vůči jaké koliv přímce procházející vedle polygonu, vždy na stejné straně.
Rozdíl mezi *k*onvexním a *n*ekonvexním polygonem je možno pozorovat dle níže uvedených obrázku.



obr 1.:*konvexní polygon a nekonvexní polygon* [1]

Pokud následovně polygon rozdělíme na menší polygonové úpravy, pak polohu zvoleného bodu q můžeme popsat následovně:

1. Bod q se nachází uvnitř polygonu $q \in P_i$
2. Bod q se nachází vně všech polygonů $q \notin P_i$
3. Bod se nachází na hraně jednoho $q \notin P_i$ nebo dvou polygonů $q \in P_{i,i+1}$
4. Bod je totožný s vrcholem jednoho polygonu nebo více polygonů $q \in P_{i,i+1,\ldots,i+n}$

Výpočet se bude provádět na základě metody **Ray crossing algorithm** a **Winding Number algorithm**. Jdjich výpočet je popsat v následujících kapitolách.

3 Popisy algoritmů

Existuje hned několik způsobů řešení této úlohy, v dané úloze výpočet byl proveden na základě níže uvedených algoritmů.

3.1 Ray crossing algorithm

*Nechť existuje uzavřený polygon ve dvojrozměrné kartézské soustavě, tvořený pomocí **n** bodů. Nechť následně existuje bod q, polohu kterého se snažime určit. Proložíme-li bodem q nekoneční počet paprsků směrem k polygonu, pak pro jednotlivý paprsek nastané následující situace:*

1. počet průsečíků paprsku k je roven sudému počtu, pak bod q se nachází vně polygonu $q \notin P_i$
2. počet průsečíků paprsku k je roven lichému počtu, pak bod q se nachází uvnitř polygonu $q \in P_i$

Zároveň mohou nastat jisté možností singularity, respektive jisté situace, kdy algoritmus nefunguje nedokáže přímo dat správný výsledek. V algoritmu ray crossing konkretně se jedna o:

1. Bod se nachází na hraně jednoho $q \in P_i$ nebo dvou polygonů $q \in P_{i,i+1}$
2. Bod je totožný s vrcholem jednoho polygonu nebo více polygonů $q \in P_{i,i+1,\ldots,i+n}$

Řešením je posun, respektivě redukce vrcholů polygonů směr k poloze bodu q.
Hledáný algoritmus je možné popsat následovně:

1. Inicializace bodů polygonu p_i, počet průsečíku $= 0$;
2. Redukce souřadnic x bodů polygonu k bodu q, respektivě k paprskovému segmentu, $x_i^{'} = x_i - x_q$.
3. Redukce souřadnic y bodů polygonu k bodu q, respektivě k paprskovému segmentu, $y_i^{'} = y_i - y_q$.
4. Znovu pro ostatní body daného polygonu $p_i/$
5. if $(y_i^{'} <= 0) \&\& (y_i + 1^{'} > 0) \| (y_i^{'} > 0) \&\& (y_i + 1^{'} <= 0)$.
6. $x_m^{'} = (x_i + 1^{'} y_i^{'} - x_i^{'} y_i + 1^{'})/(y_{i+1}^{'} - y_i^{'})$.
7. Sčítaní počtu redukováných bodů, pro $x_i^{'} > 0$
8. Poté nastáva situace, že pokud počet průsečíku je sudý, pak $q \in P$, pokud není, pak $q \notin P$

3.2 Winding Number Algorithm

*Nechť existuje uzavřený polygon ve dvojrozměrné kartézské soustavě, tvořený pomocí **n** bodů. Nechť následně existuje bod q, polohu kterého se snažíme určit. Z pohledu bodu q provedeme orientaci směru ze které pak následně se určí součet všech úhlů na jednotlivé body uvedeného polygonu.*

Součtový úhel bude dále uváděn jako w. Výpočet je lepší provádět proti směru pohybu hodinových ručiček, jelikož v případě toho směru hodnota počítaných oběhů Winding Number Ω dosahuje kladných hodnot. Je třeba taky pamatovat, že hodnota Ω je uváděna v počtech oběhů a je záporná při oběhu ve směru hodinových ručiček a kladná ve směru opačném. Do výpočtu taky vstupuje tolerance ϵ, která zahrnuje vlivy v zaokrouhlování. Dle uvedené matematické podmínky, může nastát následující výsledky:

1. w=**2R**, pak $q \in P_i$
2. $w <$ **2R**, pak $q \notin P_i$

Níže je uveden algoritmus výpočtu:

1. Vstup $\omega = 0$, tolerance ϵ
2. Orientace z bodu q káždého následujícího bodu p_{i+1} od orientaci na bod p_i
3. Určení úhlu $\omega_i = \angle p_i, q, p_{i+1}$
4. $\omega = \omega + \omega_i$, pro bod vprávo od orientace na bod p_i, pokud bod vlevo od orientace na bod p_i, pak $\omega = \omega - \omega_i$
5. Pokud platí podmínka $(|\omega - 2\Pi| < \epsilon)$, pak $q \in P$
6. Pokud neplatí podmínka $(|\omega - 2\Pi| < \epsilon)$, pak $q \notin P$

4 Problematické situace a singularity

5 Vstupní data

5.1 Formát vstupních dat

5.2 Popis dat

6 Výstupní data

6.1 Formát výstupních dat

6.2 Popis dat

7 Ukázka aplikace

8 Závěr

8.1 Náměty na vylepšení