Cvičení 1

Obsah:
- reprezentace polygonu
- problém lokalizace bodu v obecném polygonu
- problém lokalizace bodu v konvexním polygonu
- planární graf, Eulerova formule
- planární mapa

Motivace:
V mnoha průmyslových aplikacích nacházíme problémy spoje s geometrií. Motivací pro první i následující cvičení nechť jsou aplikace typu ray-tracing 3D scén nebo výpočety a simulace metodou konečných prvků. Takovéto aplikace je možno řešit ve zlomkových časech oproti přímočarým (a méně sofistikovaným) řešením. Úlohy lokalizace bodu vycházejí z potřeb ray-tracingu. Planární mapu možno použít pro řešení úloh konečnými prvky. V našich cvičeních se omezíme na úlohy v 2D.

Reprezentace polygonu:
V našich úlohách bychom si většinou měli vystačit s datovou strukturou pole, která obsahuje x a y složku jednotlivých vrcholů polygonu. Při implementaci většinou předpokládáme, že první a zároveň poslední bod polygonu je v poli pouze jednou.

Problém lokalizace bodu v obecném polygonu:
Tato úloha se zabýva řešením otázky, zda se zadaný bod naléza uvnitř nebo vně nějakého obecného polygonu. Obecným polygonem máme na mysli polygon, který není konvexní. V následující části se seznámíme s řešením tohoto problému.

<u>Metoda průsečíku paprsku</u>
Jednoduchá metoda pro zjištění, zda je bod uvnitř, či vně polygonu vychází z Jordanovy věty o křivce. Veďme paprsek ze zadaného bodu jakýmkoli směrem od něj. Podle počtu průsečíku s polygonem můžeme určit, zda je bod uvnitř nebo vně polygonu. Při lichém počtu průsečíků je bod uvnitř polygonu. V opačném případě (počet průsečíků je sudý) je zadaný bod vně polygonu.

Průsečík paprsku s jednotlivými hranami polygonu lze vypočítat jako soustavu dvou rovnic, kde dané rovnice jsou parametrickým vyjádřením úseček paprsku a hrany polygonu. Pokud má soustava rovnic jedno řešení, paprsek a hrana se protínají.

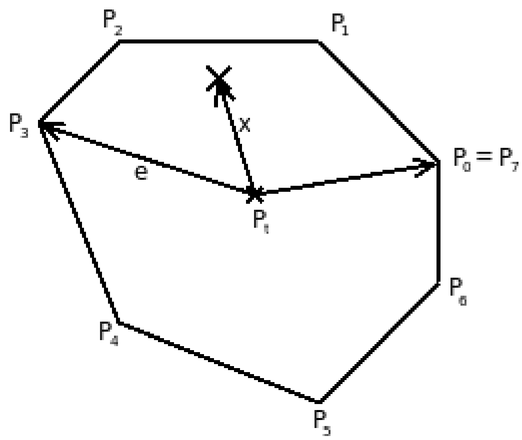
Časovou složitost toho to postupu lze snadno vyčíst z algoritmu, který musí vypočíst průsečík paprsku se všemi hranami polygonu. Z tohoto snadno vyvodíme složitost O(n), Theta(n), Omega(n).

Problém lokalizace bodu v konvexním polygonu
Tento problém je již používán častěji. Používá se v CAD a modelovacích nástrojích nebo v ray-tracingu.

<u>Metoda půlení intervalu</u>
Na začátku předpokládejme, že zadaný polygon má seřazené body tak, že jejich průchodem můžeme vytvořit zadaný polygon. Dále si vytvoříme težiště polygonu. Vytvoříme si dvě poloviny ze zadaného polygonu a to tak, že z bodu těžiště vedeme jeden vektor do počátečního bodu a druhý vektor opět z těžiště do bodu, který je v polovině seznamu bodů polygonu. Tímto dostaneme dvě výseče. Pro další

postup si vezmeme tu z výsečí, která svírá úhel meší než 180 stupňů. Poté přistoupíme k samotné lokalizaci. Vektory ***e*** a ***x*** podrobíme vektorovému součinu. Výsledek vektorového součinu určuje, zda je hledaný bod vlevo nebo vpravo od vektoru ***e***. Pokud je výsledek vektorového součinu kladný, resp. záporný, je bod *X* vpravo, resp. vlevo od vektoru ***e***. Tímto určíme ve které polovině polovině polygonu se bod nachází. Dále postupujeme rekurzivně až nalezneme výsledný trojúhelník, ve kterém se hledaný bod nachází.

Časovou složitost tohoto algoritmu určíme jednoduše pomocí toho, že algoritmus je de facto metodou binárního vyhledávání aplikovanou na geometrickou úlohu. Složitost je tedy O(log n).

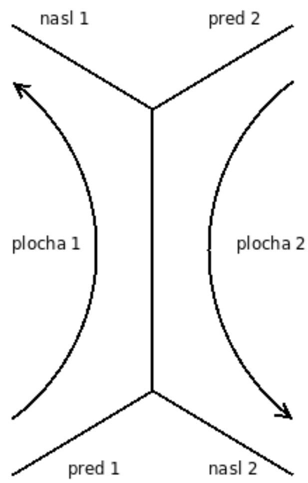


Planární graf
Planární graf je takový, jenž je možno nakreslit na rovnině tak, aby se jeho hrany nekřížily. Takovým grafem je například úplný graf K4.

Eulerova formule
f + v = e + 2

f – počet ploch, které jsou ohraničeny uravřenými hranami + jedna plocha obklopující graf
v – počet vrcholů grafu
e – počet hran grafu

Planární mapa
Planární mapa popisuje topologickou strukturu, která je zadána vrcholy, hranamí a plochami. Můžeme se s ní setkat např. při výpočtech metodou konečných prvků. Je dána jako graf, který se skládá z vrcholů, hran, ploch a popisem jejich vztahu.



- plocha
  - dalsiPlocha
  - hrana
    - pred1, nasl1 (hrana)
    - pred2, nasl2 (hrana)
    - plocha1, plocha2
    - vrchol1, vrchol2