

Punkt wewnątrz wielokąta

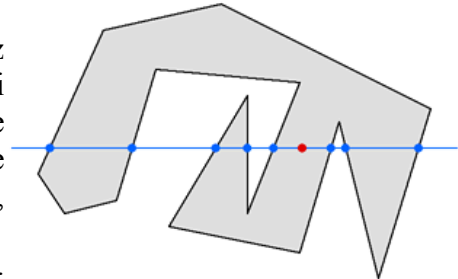
Referat z przedmiotu "Analiza i synteza algorytmów"

Adam Sawicki, gr. II

25 kwietnia 2006

Problem

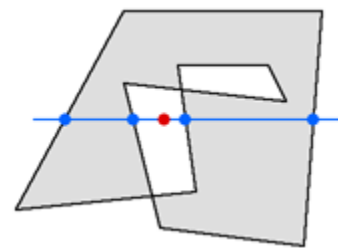
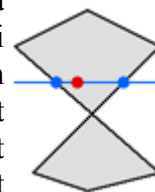
W wielu dziedzinach informatyki, szczególnie związanych z grafiką komputerową (jak np. programowanie gier) zachodzi potrzeba testowania tzw. kolizji, czyli nachodzenia na siebie różnego rodzaju obiektów geometrycznych na płaszczyźnie bądź w przestrzeni. Jednym z takich problemów jest testowanie, czy punkt leży wewnątrz, czy na zewnątrz wielokąta.



Dany jest wielokąt dowolny złożony z `polySides` krawędzi i zdefiniowany jako uporządkowany ciąg punktów opisanych współrzędnymi przechowywanymi w tablicy `polyX[]` i `polyY[]` oraz punkt o współrzędnych (x, y) . Zadaniem jest stwierdzenie, czy punkt należy do wielokąta.

Algorytm

Algorytm rozwiązujący ten problem opiera się na spostrzeżeniu, że jeśli po rzuceniu promienia (czyli półprostej) wychodzącej z tego punktu w dowolnym kierunku liczba jej przecięć z krawędziami wielokąta jest nieparzysta, punkt leży wewnątrz wielokąta, a jeśli jest parzysta, punkt leży na zewnątrz. Najprościej jest testować prostą poziomą przechodzącą przez ten punkt.

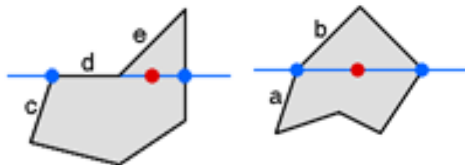


Parzystość lub nieparzystość ilości krawędzi jest zawsze taka sama po prawej jak i po lewej stronie.

W algorytmie tym, w przeciwieństwie do wielu innych znanych algorytmów geometrycznych, nie czyni się żadnych specjalnych założeń co do natury tego wielokąta – nie musi on być wypukły, może nachodzić sam na siebie, jego kolejne wierzchołki mogą być zorientowane w dowolnym kierunku itd.

Przypadki szczególne

Pewną trudność stanowią przypadki szczególne – kiedy testowany promień przechodzi przez wierzchołek bądź kiedy jest równoległy do krawędzi wielokąta. Na szczęście istnieje proste rozwiązanie – wystarczy jako krawędź przecinającą uznawać tylko taką, której jeden z wierzchołków leży poniżej, a drugi na lub powyżej testowanego, poziomego promienia.



Kod

```
bool pointInPolygon()
{
    int i, j = 0;
    bool oddNODES = false;
    for (i = 0; i < polySides; i++)
    {
        j++;
        if (j == polySides)
            j = 0;
        if (polyY[i] < y && polyY[j] >= y || polyY[j] < y && polyY[i] >= y)
        {
            if (polyX[i] + (y - polyY[i]) / (polyY[j] - polyY[i]) *
                (polyX[j] - polyX[i]) < x)
            {
                oddNODES = !oddNODES;
            }
        }
    }
    return oddNODES;
}
```

Podsumowanie

Przedstawiony algorytm pozwala testować, czy podany punkt na płaszczyźnie 2D należy do wnętrza podanego wielokąta, przy czym wielokąt ten może być dowolny.

Bibliografia

1. Point-In-Polygon Algorithm — Determining Whether A Point Is Inside A Complex Polygon. Darel Rex Finley.