Obsah

[1 Zadání 2](#_Toc56158054)

[1.2 Údaje o bonusových úlohách 2](#_Toc56158055)

[2 Popis a rozbor problému 3](#_Toc56158056)

[3 Popis algoritmů 3](#_Toc56158057)

[3.1 Jarvis Scan 3](#_Toc56158058)

[3.2 Quick Hull 4](#_Toc56158059)

[3.3 Sweep Line 5](#_Toc56158060)

[3.4 Graham Scan 6](#_Toc56158061)

[4 Problematické situace, generátory 7](#_Toc56158062)

[4.1 Konstrukce striktní konvexních obálek 7](#_Toc56158063)

[4.2 Generátor gridu 8](#_Toc56158064)

[4.3 Generátor kružnice 8](#_Toc56158065)

[4.4 Generátor náhodných bodů 8](#_Toc56158066)

[4.5 Generátor elipsy 8](#_Toc56158067)

[4.6 Generátor čtverce 8](#_Toc56158068)

[4.7 Generátor obdélníků 8](#_Toc56158069)

[4.8 Výpočet směrů pro Graham Scan 8](#_Toc56158070)

[5 Popis Aplikace 9](#_Toc56158071)

[5.1 Vstupní data 9](#_Toc56158072)

[5.2 Výstupní data 9](#_Toc56158073)

[6 Dokumentace 12](#_Toc56158074)

[6.1 Třída Algorithms 12](#_Toc56158075)

[6.2 Třída Draw 13](#_Toc56158076)

[6.3 Třída Generatorforpoint 13](#_Toc56158077)

[6.4 Pomocné třídy 14](#_Toc56158078)

[7 Porovnání časové náročnost algoritmů 14](#_Toc56158079)

[7.1 Jarvis Scan 14](#_Toc56158080)

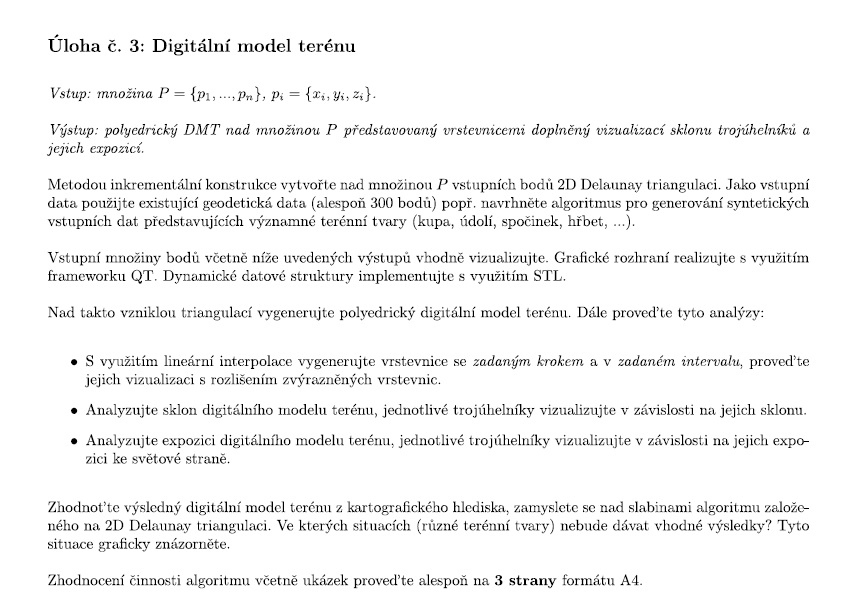
[7.2 Quick Hull 17](#_Toc56158081)

[7.3 Sweep Line 19](#_Toc56158082)

[7.4 Grafy 21](#_Toc56158083)

[8 Závěr 23](#_Toc56158084)

### 1 Zadání



## 1.2 Údaje o bonusových úlohách

|  |  |
| --- | --- |
| **Krok** | **Řešeno/neřešeno** |
| Delaunay triangulace, polyedrický model terénu. |  |
| Konstrukce vrstevnic, analýza sklonu a expozice. |  |
| Triangulace nekonvexní oblasti zadané polygonem. |  |
| Výběr barevných stupnic při vizualizaci sklonu a expozice. |  |
| Automatický popis vrstevnic. |  |
| Automatický popis vrstevnic respektující kartografické zásady (orientace, vhodné rozložení). |  |
| Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet, ...). |  |
| 3D vizualizace terénu s využitím promítání. |  |
| Barevná hypsometrie. |  |

# 2 Popis a rozbor problému

Cílem úlohy je vytvoření aplikace, která nad množinou bodů vytvoří polyedrický model terénu pomocí Delaunay triangulace. Vstupní body lze do aplikace importovat ze souboru nebo zadávat kliknutím myši do prostoru.

# Popis algoritmů

## 3.1 Delaunay triangulace

Delaynay triangulace je metoda pro vytvoření trojúhelníkové sítě. Metoda funguje na principu inkrementální konstrukce, která je založena na postupném přidávání bodů do již vytvořené množiny DT. Nad existující Delaunayovsko hranou *e* je hledán bod p minimalizují poloměr

Tato hraně je orientovaná, bod je hledán pouze nalevo od ní. Pokud se nalevo žádný takový bod nenachází, dojde ke změně orientace hrany.

Do množiny *DT* jsou přidávány tři hrany tvořící trojúhelník. Algoritmus obsahuje seznam aktivních hran – *Active Edge List* (AEL), který obsahuje hrany, ke kterým ještě nebyl nalezen třetí bod. Pokud je nalezena nová hrana, je nutno otestovat, jestli se v seznamu již nenachází hrana s opačnou orientací. Pokud ne, je hrana vložena do seznamu. Pokud je k hraně z AEL nalezen třetí bod, je ze seznamu odstraněna. Celý algoritmus je ukončen, ve chvíli kdy seznam hran AEL je prázdný.

Určení Delaunay bodu p:

1. Určení polohy bodu od dané hrany:
2. Pokut t < 0 bod p se nachází v levé polorovině a je vypočten poloměr r:

kde

Algoritmus Delaunay triangulace:

1. Nalezení pivota *p1*, *p1 = min*(*x*), nalezení bodu *p2* nejbližšího k pivotu
2. Vytvoření hrany *e* = (*p1*, *p2*)
3. Nalezení optimálního Delaunay bodu *p*
4. Pokud *p* nenalezen, prohoď orientaci *e*, tj. *e* = (*p2*, *p1*), znovu krok 3
5. Vytvoř zbývající hrany trojúhelníku, tj. *e2* = (*p2*, *p*) a *e3* = (*p*, *p1*)
6. AEL ← *e*, AEL ← *e2*, AEL ← *e3*
7. DT ← *e*, DT ← *e2*, DT ← *e3*
8. dokud AEL není prázdná:
9. AEL → *e, e* = (*p1*, *p2*) // vezmi první hranu z AEL
10. Prohození orientace *e*, tj. *e* = (*p2*, *p1*)
11. Nalezení optimálního Delaunay bodu *p*
12. Pokud *p* existuje
13. *e2* = (*p2*, *p*), *e3* = (*p*, *p1*)
14. DT ← *e*, DT ← *e2,* DT ← *e3*
15. Prohození orientace *e2* a *e3, tj. e2‘* = (*p*, *p2*), *e3‘* = (*p1*, *p*)
16. Pokud *e2‘* je v AEL
17. AEL → *e2‘*
18. Jinak AEL ← *e2*
19. Pokud *e3‘* je v AEL
20. AEL → *e3‘*
21. Jinak AEL ← *e3*

## 

## 3.2 Tvorba vrstevnic

Vrstevnice byly vytvořeny pomocí lineární interpolace, která je založena na analytické geometrii. Je hledána průsečnice roviny tvořená trojúhelníkem a vodorovné roviny s výškou *z*. Koncové body *A,B* průsečnice jsou určeny z podobnosti trojúhelníků:

Z bodu *A,B* je vytvořena hrana určující v daném trojúhelníku vrstevnici o dané výšce z. Vrstevnice byly kresleny v případě, že strana trojúhelníku leží ve vodorovné rovině, vodorovná rovina protíná trojúhelník ve vrcholu a protilehlé straně nebo v případě, že vodorovná rovina protíná trojúhelník ve dvou stranách.

## 3.3 Sklon terénu

Sklon terénu je vyjádřen úhlem mezi normálovým vektorem (0,0,1) a normálovým vektorem roviny trojúhelníku. Pro určení úhlu byly použity tyto vzorce:

## 3.4 Orientace terénu

Orientace terénu je vyjádřena azimutem. Azimut je dán průmětem normálového vektoru z roviny trojúhelníku do roviny ρx,y. Pro určení azimutu byly použity tyto vzorce:

# 4 Problematické situace, generátory

# Popis Aplikace

V této kapitole je popsána funkcionalita aplikace.

## 5.1 Vstupní data

## 5.2 Výstupní data

# 

# 6 Dokumentace

## 6.1 Třída Algorithms

Třída obsahuje funkce:

## Třída Draw

Třída obsahuje funkce:

## 6.4 Pomocné třídy

*sortByAngle*

- Třída setřídí úhly podle velikosti, pokud se směry rovnají vrátí bod, který je vzdálenější od pivotu q

*sortByX*

- Třída setřídí body podle souřadnice x

*sortByY*

- Třída setřídí body podle souřadnice y

*removeByAngle*

- Třída setřídí vyhledá směry, které jsou si rovny

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 8 Závěr

## Seznam obrázku

[Obrázek 1: Jarvis Scan, zdroj: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf 4](#_Toc56161707)

[Obrázek 2: Quick Hull, zdroj: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf 5](#_Toc56161708)

[Obrázek 3: Quick Hull, zdroj: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf 5](#_Toc56161709)

[Obrázek 4: Sweep Line, zdroj: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf 6](#_Toc56161710)

[Obrázek 5: Graham Scan, zdroj: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk4.pdf 7](#_Toc56161711)

[Obrázek 6: Ukázka aplikace – vložení bodů myší a vykreslení konvexní obálky 9](#_Toc56161712)

[Obrázek 7: Ukázka aplikace – generované body (grid) 10](#_Toc56161713)

[Obrázek 8: Ukázka aplikace – generované body (náhodné) 10](#_Toc56161714)

[Obrázek 9: Ukázka aplikace – generované body (kružnice) 11](#_Toc56161715)

[Obrázek 10:: Ukázka aplikace – generované body (elipsa) 11](#_Toc56161716)

[Obrázek 11: Ukázka aplikace – generované body (čtverec) 12](#_Toc56161717)

[Obrázek 12: Graham Scan – špatné vykreslení konvexní obálky u gridu 23](#_Toc56161718)