České vysoké učení technické v Praze
Fakulta stavební

155ADKG: Digitální model terénu

Michael Kala
Anna Zemánková

1 Zadání

Vstup: množina $P = \{p_1, ..., p_n\}$, $p_i = \{x_i, y_i, z_i\}$.

Výstup: polyedrický DMT nad množinou P představovaný vrstevnicemi doplněný vizualizací sklonu trojúhelníků a jejich expozicí.

Metodou inkrementální konstrukce vytvořte nad množinou P vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů) popř. navrhněte algoritmus pro generování syntetických vstupních dat představujících významné terénní tvary (kupa, údolí, spočinek, hřbet, ...).

Vstupní množiny bodů včetně níže uvedených výstupů vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku QT. Dynamické datové struktury implementujte s využitím STL.

Nad takto vzniklou triangulací vygenerujte polyedrický digitální model terénu. Dále proveďte tyto analýzy:

- S využitím lineární interpolace vygenerujte vrstevnice se *zadaným krokem* a v *zadaném intervalu*, proveďte jejich vizualizaci s rozlišením zvýrazněných vrstevnic.
- Analyzujte sklon digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich sklonu.
- Analyzujte expozici digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich expozici ke světové straně.

Zhodnoťte výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabinami algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích (různé terénní tvary) nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

Zhodnocení činnosti algoritmu včetně ukázek proveďte alespoň na tři strany formátu A4.

Hodnocení:

Krok	**Hodnocení**
Delaunay triangulace, polyedrický model terénu.	10b
Konstrukce vrstevnic, analýza sklonu a expozice.	10b
Triangulace nekonvexní oblasti zadané polygonem.	*+5b*
Výběr barevných stupnic při vizualizaci sklonu a expozice.	*+3b*
Automatický popis vrstevnic.	*+3b*
Automatický popis vrstevnic respektující kartografické zásady (orientace, vhodné rozložení).	*+10b*
Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet, ...).	*+10b*
3D vizualizace terénu s využitím promítání.	*+10b*
Barevná hypsometrie.	*+5b*
Max celkem:	**65b**

Čas zpracování: 3 týdny

2 Údaje o bonusových úlohách

Z bonusových úloh byly zpracovány:

- automatický popis vrstevnic
- barevná hypsometrie

3 Popis a rozbor problému

Mějme množinu bodů $P\{p_i\}$, $p_i = \{x_i, y_i, z_i\}$. Nad touto množinou chceme vytvořit síť trojúhelníků t_j pomocí Delaunay triangulace DT, následně vytvořit vrstevnice a pro vizualizaci DMT určit sklon a expozici jednotlivých trojúhelníků.

3.1 Delaunay triangulace

Vlastnosti:

- Uvnitř kružnice opsané trojúhelníku $t_j \in DT$ neleží žádný jiný bod množiny P.
- DT maximalizuje minimální úhel v $\forall t_j$, avšak DT neminimalizuje maximální úhel v t_j.
- DT je lokálně optimální i globálně optimální vůči kritériu minimálního úhlu.
- DT je jednoznačná, pokud žádné čtyři body neleží na kružnici.

3.2 Vrstevnice

Vrstevnice byly určeny za využití lineární interpolace, při které se předpokládá, že spád terénu je mezi podrobnými body p_i, mezi nimiž se provádí interpolace, konstantní.

Mějme trojúhelník t_j, tvořený hranami e_1, e_2, e_3 a rovinu vrstevnice ρ o dané výšce. Vztah hrany tojúhelníku a roviny vrstevnice:

1. $(z - z_i) * (z - z_{i+1}) < 0 \longrightarrow e_i \cap \rho$
2. $(z - z_i) * (z - z_{i+1}) > 0 \longrightarrow e_i \notin \rho$
3. $(z - z_i) * (z - z_{i+1}) = 0 \longrightarrow e_i \in \rho$

Pokud $e_1, e_2, e_3 \in \rho$, jedná se o trojúhelník náležící rovině ρ a není nutné vrstevnici pro tento trojúhelník řešit.
Jestliže $e_i \cap \rho$, je vypočten průsečík hrany $e_i = (p_1, p_2)$ a roviny vrstevnice ρ o výšce z:

$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1,$$

$$y = \frac{(y_2 - y_1)}{(z_2 - z_1)}(z - z_1) + y_1.$$

3.3 Sklon

Sklon je úhel φ mezi svislicí n a normálou trojúhelníku n_t. Rovina trojúhelníku t_j je určena vektory u,v.

$$n = (0, 0, 1)$$

$n_t = \vec{u} \times \vec{v}$

$$\varphi = \arccos(\frac{n_t \cdot n}{|n_t||n|})$$

Sklon je vizualizován odstíny šedi.

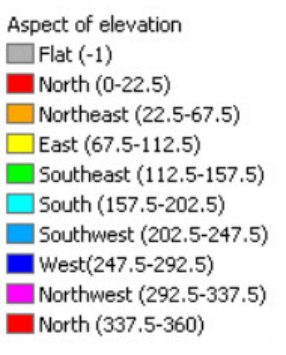
3.4 Expozice

Expozice je orientace trojúhelníku vůči světovým stranám.

$$A = \arctan 2(\frac{n_x}{n_y});$$

kde n_x, n_y jsou vektorové součiny u a v.

Expozice je vizualizována pomocí barevného spektra, barvy byly vybrány stejně jako v SW ArcMAP - ESRI:



Obrázek 1: Vizualizace expozice

4 Popis algoritmů

4.1 Delaunayova triangulace

Triangulace byla realizována metodou inkrementální konstrukce, body jsou tedy do triangulace přidávány postupně a to tak, aby vybraný bod ležel v levé polorovině od orientované hrany, poloměr opsané kružnice byl minimální a zároveň jsou preferovány body, jejichž střed opsané kružnice leží v pravé polorovině. Pokud žádný bod těmto kritériím nevyhovuje, je orientace hrany obrácena a bod je vybírán znovu. Jakmile je bod nalezen, jsou k němu vytvořeny orientované hrany a vše je uloženo do triangulace.
Pro "manipulaci"s hranami se používá struktura Active Edges List AEL, do ní jsou ukládány hrany, ke kterým je třeba nalézt třetí bod a vytvořit trojúhelník. Jakmile je AEL prázdná, algoritmus končí.

4.1.1 Implementace metody

1. $p_1 = rand(P), ||p_2 - p_1|| = min$ ....náhodný a nebližší bod
2. Vytvoř hranu $e = (p_1, p_2)$
3. Inicializuj: $p_{min} = argmin_{\forall p_i \in \sigma_L(e)} r'(k_i), k_i = (a, b, p_i), e = (a, b)$
4. Pokud $\nexists p_{min}$, prohoď orientaci $e \longleftarrow (b, a)$. Jdi na 3)
5. $e_2 = (p_1, p_{min}), e_3 = (p_{min}, p_1)$...zbývající hrany trojúhelníku
6. $AEL \longleftarrow e, AEL \longleftarrow e_2, AEL \longleftarrow e_3$
7. $DT \longleftarrow e, DT \longleftarrow e_2, DT \longleftarrow e_3$
8. while AEL not empty:
9. $AEL \longrightarrow e, e = (p_1, p_2)$....vezme první hranu z AEL
10. $e = (p_2, p_1)$ ...prohodí její orientaci
11. $p_{min} = argmin_{\forall p_i \in \sigma_L(e)} r'(k_i), k_i = (a, b, p_i), e = (a, b)$
12. if $\exists p_{min}$:
13. $e_2 = (p_1, p_{min}), e_2 = (p_{min}, p_1)$...zbývající hrany trojúhelníku
14. $DT \longleftarrow e$
15. $add(e_2, AEL, DT), add(e_3, AEL, DT)$

Dílčí algoritmus Add:

1. Vytvoř hranu $e' = (b, a)$
2. if $(e' \in AEL)$
3. $\quad AEL \longrightarrow e'$...Odstraň z AEL
4. else:
5. $\quad AEL \longleftarrow e$ ...Přidej do AEL
6. $DT \longleftarrow (a, b)$ Přidej do DT

4.1.2 Problematické situace

Pokud je vstupními daty grid, dochází k nejednoznačnosti Delaunay triangulace - na opsané kružnici leží čtyři body. V tom případě nefunguje výpočet vrstevnic ani následných analýz.

4.2 Barevná hypsometrie

Myšlenka výpočtu barevné hypsometrie vychází z procházení všech trojúhelníků triangulace i s jejich vrstevnicemi. Výpočet pak vždy probíhá nad jedním trojúhelníkem. Jediný moment při běhu programu, kdy lze získat trojúhelníky a jejich vrstevnice dohromady, je při výpočtu vrstevnic. Proto výpočet hypsometrie probíhá zároveň s výpočtem vrstevnic. Samostatný algoritmus vypadá následovně:

1. Vytvoř vektor polygonů **pols** s údajem o jejich střední výšce
2. Procházej po jednom všechny trojúhelníky:
   (a) Do vektoru bodů **pts** ulož vrcholy trojúhelníku a krajní body vrstevnic
   (b) Vytvoř vektor vektorů bodů **vec_pts** o velikosti počtu intervalů vrstevnic v rámci trojúhelníku (počítají se i krajní intervaly, ve kterých může ležet vrchol trojúhelníku)
   (c) Procházej vektor bodů **pts** a ulož body podle výšky do příslušného vektoru bodů z **vec_pts**.
   (d) Vypočítej konvexní obálku každého vektoru bodů z **vec_pts**, vypočti jejich průměrnou výšku a ulož do **pols**.
3. Seřaď **pols** bodle středních výšek.
4. Naškáluj tabulku barev podle počtu polygonů v **pols**.
5. Vykresli polygony.

Pozn.: Pro vytvoření polygonů bylo třeba každou množinu bodů seřadit tak, aby byly ve správném pořadí. K tomu byl využit algoritmus Sweep Line z minulé úlohy (možná dělo na vrabce, ale funkční a již ověřené).

4.3 Automatický popis vrstevnic

Při vytváření vrstevnic je zároveň ukládána jejich výška. Vektor výšek je pak procházen společně s vektorem vrstevnic a ke každé hlavní je přidán popisek.

5 Vstupní data

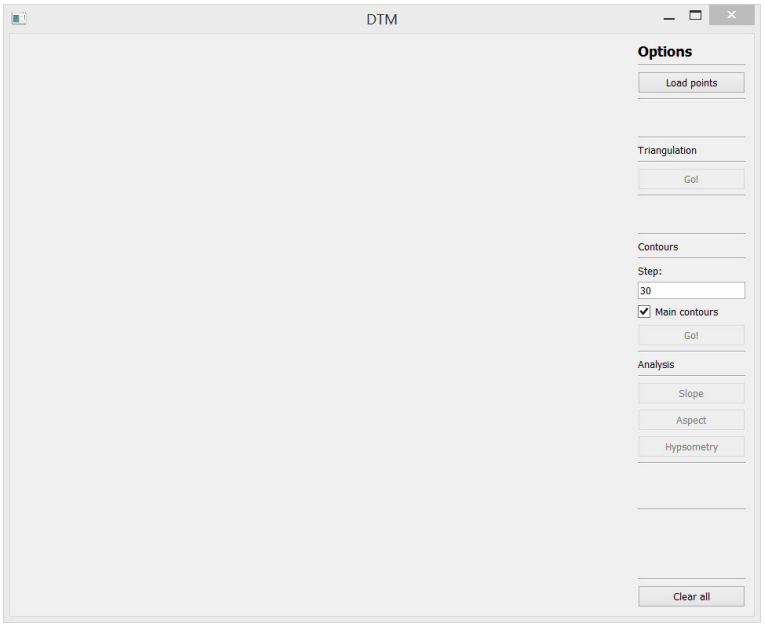
Vstupními daty je textový soubor **.txt* se souřadnicemi bodů ve tvaru [X,Y,Z]. Lze jej do aplikace nahrát pomocí tlačítka *Load points*.

Součástí příloh je textový soubor s testovacími daty *testovaci_data.txt*.

6 Výstupní data

Výsledky aplikace jsou vizualizovány v kanvasu grafického rozhraní.

7 Ukázka vytvořené aplikace



Obrázek 2: Rozvržení aplikace

Nejprve je třeba nahrát vstupní data pomocí tlačítka *Load points* a následně je možné vypočítat Delaunay triangulaci, teprve poté jsou zpřístupněna i ostatní tlačítka aplikace.

Při výpočtu vrstevnic je třeba zadat krok vrstevnic a (ne)zaškrtnout zvýraznění hlavních vrstevnic. Po výpočtu vrstevnic je zpřístupněno tlačítko pro barevnou hypsometrii.

V sekci analysis je možné vypočíst a vizualizovat sklon a expozici či barevnou hypsometrii. Po opětovném stisknutí příslušného tlačítka se daná vizualizace opět skryje.

Vše je uvedeno do původního stavu (smazány body i vypočtené výsledky) kliknutím na tlačítko Clear.

8 Dokumentace

8.1 QPoint3D

Definice nového datového typu **QPoint3D**, který dědí od třídy **QPointF**. Rozšíření o výškovou souřadnici **z**. V defaultním konstruktoru jsou nasteveny implicitní hodnoty souřadnic: $[0, 0, 0]$.

- členská proměnná **double z** - výška bodu
- public metody:
  - **getZ** - získání souřadnice z. Návratovým typem je **double**.
  - **setZ** - uložení hodnoty do členské proměnné z. Návratovým typem je **void**.

8.2 QPolygonFZ

Definice nového datového typu **QPolygonFZ**, který dědí od třídy **QPolygonF**. Rozšíření o průměrnou výšku polygonu.

- členská proměnná **double z_mid** - průměrná výška polygonu
- public metody:
  - **getZ** - získání průměrné výšky polygonu. Návratovým typem je **double**.
  - **setZ** - uložení průměrné výšky. Návratovým typem je **void**.

8.3 Edge

Definice nového datového typu **Edge**. Uspořádaná dvojice vrcholů simulující orientovanou hranu (startovní a koncový vrchol).

- členské proměnné **QPoint3D s, e** - počáteční a koncový bod hrany
- public metody:
  - **getS resp. getE** - získání počátečního resp. koncového bodu hrany. Návratovým typem je **QPoint3D**.
  - **switchOr** - metoda slouží ke změně orientace hrany - prohodí počáteční a koncový bod. Návratovým typem je **void**.
- Přetížení operátoru == pro porovnání dvou hran (hrany se rovnají, pokud mají shodný počáteční a koncový bod).

8.4 Triangle

Definice nového datového typu **Triangle**. Ukládá trojici vrcholů a sklon a expozici jimi tvořeného trojúhelníku.

- členské proměnné:
  QPoint3D p1,p2,p3 - tři vrcholy trojúhelníku
  double slope, aspect - sklon a expozice trojúhelníku
- public metody **get**<*příslušná členská proměnná*> - vrací danou proměnnou.

8.5 Sortovací kritéria

- **SortByXAsc** - řazení bodů typu **QPoint3D** podle rostoucí souřadnice x, při rovnosti podle y.
- **SortByZAsc** - řazení bodů typu **QPoint3D** podle rostoucí souřadnice z, při rovnosti podle y, při rovnosti podle x.
- **SortPolByZAsc** - řazení polygonů typu **QPolygonFZ** podle střední výšky polygonu.

8.6 Algorithms

V třídě Algorithms jsou staticky implementovány algoritmy počítající Delaunay triangulaci, vrstevnice, analýzu DMT (sklon a expozici) - a barevnou hypsometrii.

- Výčtový typ **TPosition**
  - Typ využitý jako návratová hodnota členské metody **getPointLinePosition**.
  - **LEFT = 0**
  - **RIGHT = 1**
  - **ON = 2**
- Metoda **getPointLinePosition**
  - Tato metoda slouží k určení polohy bodu vůči přímce. Návratovou hodnotou je výčtový typ **TPosition**.
  - Vstup
    - **QPoint3D &q** - určovaný bod
    - **QPoint3D &a, &b** - body přímky
  - Výstup
    - **LEFT** - bod vlevo od přímky
    - **RIGHT** - bod vpravo od přímky
    - **ON** - bod na přímce
- Metoda **getCircleRadius**

  - Tato metoda slouží k výpočtu poloměru opsané kružnice. Návratovou hodnotou je **double**.
  - Vstup
    * **QPoint3D &p1, &p2, &p3, &c** - body p_i definují kružnici, do bodu c jsou ukládány souřadnice středu kružnice.
  - Výstup
    * Vypočtený poloměr kružnice.

- Metoda **getNearestPoint**
  - Tato metoda slouží k vyhledání nejbližšího bodu. Návratovou hodnotou je **int**.
  - Vstup
    * **QPoint3D &p, std::vector<QPoint3D> &points** - od bodu p je hledán nejbližší bod z vektoru bodů *points*.
  - Výstup
    * index nejbližšího bodu.

- Metoda **distance**
  - Tato metoda slouží k výpočtu vzdálenosti dvou bodů. Návratovou hodnotou je **double**.
  - Vstup
    * **QPoint3D &p1, QPoint3D &p2**
  - Výstup
    * vzdálenost dvou bodů.

- Metoda **getDelaunayPoint**
  - Tato metoda slouží k vyhledání bodu, který vyhovuje kritériím Delaunay triangulace. Návratovým typem je **int**.
  - Vstup
    * **QPoint3D &s, QPoint3D &e, std::vector<QPoint3D> &points**
  - Výstup
    * index bodu.

- Metoda **DT**
  - Tato metoda slouží k vytvoření vektoru, v němž jsou uloženy hrany Delaunay triangulace. Návratovým typem je **std::vector<Edge>** .
  - Vstup
    * **std::vector<QPoint3D> &points**
  - Výstup
    * vektor hran Delaunay triangulace.

- Metoda **getContourPoint**
  - Tato metoda slouží k vypočtení souřadnic průsečíku vrstevnice a hrany DT. Návratovým typem je **QPoint3D**.
  - Vstup
    - **QPoint3D &p1,&p2, double z**, z je výška vrstevnice, bod p_1 resp. p_2 je počáteční resp. koncový bod hrany.
  - Výstup
    - průsečík vrstevnice a trojúhelníku DT.
- Metoda **createContours**
  - Tato metoda slouží k vypočtení vrstevnic a barevné hypsometrie. Návratovým typem je **std::vector<Edge>**.
  - Vstup
    - **std::vector<Edge> &dt** - vektor hran
    - **double z_min, double z_max, double dz** - rozsah vrstevnic a jejich krok
    - **std::vector<double> &contour_heights** - vektor, do kterého jsou ukládány výšky vrstevnic, pořadí stejné jako pořadí hran vrstevnic ve výstupu z metody
    - **std::vector<QPolygonFZ> &hyps** - vektor, do kterého jsou ukládány polygony jednoho intervalu vrstevnic v rámci jednoho trojúhelníku (pro barevnou hypsometrii).
  - Výstup
    - vektor hran vrstevnic.
- Metoda **getSlope**
  - Tato metoda slouží k výpočtu sklonu trojúhelníku. Návratovým typem je **double**.
  - Vstup
    - **QPoint3D &p1, &p2, &p3** - vrcholy trojúhelníku.
  - Výstup
    - sklon trojúhelníku ve stupních.
- Metoda **getAspect**
  - Tato metoda slouží k výpočtu expozice trojúhelníku. Návratovým typem je **double**.
  - Vstup
    - **QPoint3D &p1, &p2, &p3** - vrcholy trojúhelníku.
  - Výstup

        * expozice trojúhelníku ve stupních.

- Metoda **analyzeDMT**
  - Tato metoda slouží k vytvoření trojúhelníků a výpočtu jejich sklonu a expozice. Návratovým typem je **std::vector<Triangle>**.
  - Vstup
    * **std::vector<Edge> &dt** - vektor hran Delaunay triangulace.
  - Výstup
    * vektor trojúhelníků a jejich sklon a expozice.

- Metoda **processPts**
  - Tato metoda slouží k vytvoření vektoru polygonů se střední výškou podle zadaného kroku pro barevnou hypsometrii. Návratovým typem je **void**.
  - Vstup
    * **std::vector<QPoint3D> &pts** - body, z nichž jsou polygony tvořeny
    * **double dz** - krok vrstevnic
    * **std::vector<QPolygonFZ> &pols** - vektor, do něhož jsou vytvořené polygony ukládané.

- Metoda **sweepLineCH**
  - Tato metoda slouží k výpočtu konvexní obálky pomocí algoritmu Sweep Line. Jejím výstupním typem je **QPolygonF**. Metoda použita při výpočtu barevné hypsometrie.
  - Vstup
    * **std::vector <QPointF> &points** - vektor bodů, kolem nichž má být vytvořená konvexní obálka.
  - Výstup
    * Polygon obsahující konvexní obálku.

- Metoda **avgH**
  - Metoda slouží k výpočtu průměrné výšky zadaných bodů. Návratovým typem je **double**.
  - Vstup - **std::vector<QPoint3D> &pts** - vstupní body
  - Výstup - průměrná hodnota výšek

8.7 Draw

Třída draw slouží k vykreslení načtených bodů, vypočtené Delaunay triangulace, vrstevnic, sklonu, expozice a hypsometrie. Třída dědí od třídy **QWidget**.

- Členské proměnné

- **std::vector <QPoint3D> points** vektor načtených bodů
- **std::vector <Edge> dt** - vektor hran Delaunay triangulace
- **std::vector <Edge> contours** - vektor hran vrstevnic
- **std::vector <double> contours_heights** - vektor výšek vrstevnic
- **std::vector <Triangle> dtm** - vektor trojúhelníků Delaunay triangulace
- **std::vector <QPolygonFZ> hyps** - vektor polygonů hypsometrie
- **int step** - rozestup vrstevnic
- **bool draw_main** - flag vykreslení hlavních vrstevnic
- **bool draw_slope** - flag vykreslení sklonu - v konstruktoru třídy nastaveno na false
- **bool draw_aspect** - flag vykreslení expozice - v konstruktoru třídy nastaveno na false
- **bool draw_hyps** - flag vykreslení barevné hypsometrie - v konstruktoru třídy nastaveno na false
- **std::vector <QColor> ctable** - tabulka barev pro hypsometrii, v konstruktoru třídy načtena ze souboru

- Metoda **paintEvent**
  - Tato metoda slouží k vykreslení načtených bodů, vypočtené Delaunay triangulace, vrstevnic, sklonu, expozice a hypsometrie. Metoda se volá pomocí metody **repaint()**. Návratovým typem je **void**.
  - Vstup
    - **QPaintEvent *e**
- Metoda **clearPoints**
  - Tato metoda slouží pro vymazání bodů.
- Metoda **clearDT**
  - Tato metoda slouží pro vymazání bodů, triangulace, vrstevnic a hypsometrie. Na vstupu není nic a je typu void.
- Metoda **getPoints** Tato metoda vrací členskou proměnnou **std::vector <QPoint3D> points**.
- Metoda **getDT** Tato metoda vrací členskou proměnnou **std::vector <Edge> dt** .
- Metoda **setDT** Tato metoda slouží k přiřazení hodnot členské proměnné dt.
- Metoda **setContours** Metoda slouží k přiřazení hodnot členským proměnným **std::vector <Edge> contours**, **std::vector <double> contours_heights**, **int step**, **bool draw_main** a **std::vector <QPolygonFZ> hyps**.
- Metoda **setDTM** Tato metoda slouží k přiřazení hodnot členské proměnné **std::vector <Triangle> dtm**.
- Metoda **loadPoints**

    * Tato metoda slouží pro načtení bodů z textového souboru do vektoru **points** a pro určení minimální a maximální výšky bodů v souboru. Zároveň je zvoleno "měřítko"kanvasu dle velikosti kanvasu a rozsahu vstupních dat. Návratovým typem je **void**.
    * Vstup
        · **std::string points_path** - cesta k vstupnímu souboru
        · **QSizeF &canvas_size** - velikost kanvasu
        · **double &min_z, &max_z** - nejmenší/největší výška

- Metoda **setDrawSlope**
    * Tato metoda slouží k zapnutí/vypnutí vykreslování sklonu a k vypnutí vykreslování expozice a hypsometrie.
- Metoda **setDrawAspect**
    * Tato metoda slouží k zapnutí/vypnutí vykreslování expozice a k vypnutí vykreslování sklonu a hypsometrie.
- Metoda **drawHypsometry**
    * Tato metoda slouží k zapnutí/vypnutí vykreslování hypsometrie a k vypnutí vykreslování sklonu a expozice.
- Metoda **genAspColor**
    * Metoda slouží k nastavení barev pro expozici. Jejím návratovým typem je **std::vector<QColor>**. Na vstupu nemá nic, vrací vektor barev.

8.8 Widget

Tato třída slouží ke komunikaci s GUI. Třída dědí od třídy QWidget. Všechny její metody slouží jako sloty k signálům z GUI, nemají žádné vstupní hodnoty a jejich návratovým typem je void.

- Členské proměnné **double z_min, double z_max** - instance třídy si drží minimální a maximální výšku vstupních bodů
- Metoda **on_trg_button_clicked** - výpočet triangulace a její vizualizace.
- Metoda **on_clear_button_clicked** - maže obsah kanvasu.
- Metoda **on_cont_button_clicked** - výpočet vrstevnic a hypsometrie, vizualizace vrstevnic.
- Metoda **on_slope_button_clicked** - výpočet a vizualizace sklonu.
- Metoda **on_load_points_button_clicked** - načtení a vizualizace vstupních bodů. Nastavení minimální a maximální výšky vstupních bodů do členských proměnných.
- Metoda **on_aspect_button_clicked** - výpočet a vizualizace expozice.
- Metoda **on_hyps_button_clicked** - vizualizace barevné hypsometrie.

9 Přílohy

- Příloha č.1: Testovací data - *testovaci_data.txt*.

10 Závěr

10.1 Zhodnocení

Jistě že implementace barevné hypsometrie by šla provést určitě lépe a efektivněji. Nicméně s naším postupem jsme spokojeni, protože jsme si ho vymysleli sami. To, že implementace zabrala hodně času (zvlášť ladění) a asi o hodně víc času (hloupé chyby -> debugování (debeatlesování?) levelu Yoko Ono) než nepovinné úlohy za dvojnásobek bodů (resp. ta s generováním terénních útvarů), nám nevadí, protože jedeme spíš na to, co nás baví a barvičky baví každého.

10.2 Návrhy na vylepšení

Nynější automatické popisky vrstevnic jsou spíš ošklivé než hezké, přestože plní účel. Popisky by tedy chtělo vylepšit, aby respektovaly kartografické zásady. Dále by možná stála za to změna barevné mapy při vykreslování expozice. Expozice (ikdyž barvy jsou přebrané od ESRI) není takto moc intuitivní.

11 Zdroje

1. BAYER, Tomáš. 2D triangulace, DMT [online][cit. 1.12.2018].
   Dostupné z: https://web.natur.cuni.cz/ bayertom/images/courses/Adk/adk5.pdf

2. BAYER, Tomáš. 2D triangulace, DMT [online][cit. 30.11.2018].
   Dostupné z: https://web.natur.cuni.cz/ bayertom/images/courses/Adk/adkcv3.pdf

3. Tools: Aspect. ArcMap—ArcGIS Desktop [online]. Environmental Systems Research Institute, 2016 [cit. 2018-12-02].
   Dostupné z: http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/how-aspect-works.htm