

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

OBOR GEOMATIKA

DIPLOMOVÁ PRÁCE

**Předzpracování a analýza dat pro
vyrovnání v systému GNU Gama**

Vedoucí práce: Prof. Ing. Aleš Čepek, CSc.

Katedra geomatiky

PRAHA 2016

Bc. DOMINIK HLADÍK



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební

Thákurova 7, 166 29 Praha 6

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Hladík Jméno: Dominik Osobní číslo: 396906

Zadávací katedra: Katedra geomatiky

Studijní program: Geodézie a kartografie

Studijní obor: Geomatika

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Předzpracování a analýza dat pro vyrovnání v systému GNU Gama

Název diplomové práce anglicky: Pre-processing and analysis of data for adjustment in GNU Gama system

Pokyny pro vypracování:

Seznamte se s programovou dokumentací projektu GNU Gama <https://www.gnu.org/software/gama/>, především pak s implementací Qt uživatelského rozhraní. Navrhněte a implementujte softwarové řešení pro zpracování geodetických měření a souřadnic s cílem eliminace chyb, které lze detekovat před vlastním vyrovnáním sítě, jako jsou například kontroly dvojích měření délek, trojúhelníkové uzávěry apod.

Seznam doporučené literatury:

Dokumentace projektu <https://www.gnu.org/software/gama/>

Qt vývojové prostředí <http://www.qt.io/application-development/>

Jméno vedoucího diplomové práce: Prof. Ing. Aleš Čepěk, CSc.

Datum zadání diplomové práce: 22.2.2016 Termín odevzdání diplomové práce: 22.5.2016

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

17-3.2016

Datum převzetí zadání

Podpis studenta(ky)

Abstrakt

Tato diplomová práce se zabývá rozšířením grafického uživatelského rozhraní projektu GNU Gama. Pro tento projekt byly vytvořeny tři pluginy, které se zabývají analýzou délek a trojúhelníkových uzávěrů před vyrovnáním geodetické sítě.

Klíčová slova

GNU Gama, plugin, vyrovnání sítí, měření délek, trojúhelníkové uzávěry

Abstract

This diploma's work is focused on extension of graphical user interface for project GNU Gama. Three plugins are focused on analysis of distances and triangular closures before network adjustment were created for this project.

Key words

GNU Gama, plugin, network adjustment, measuring of distances, triangular closures

PROHLÁŠENÍ

Prohlašuji, že diplomovou práci na téma „Předzpracování a analýza dat pro vyrovnání v systému GNU Gama“ jsem vypracoval samostatně. Použité zdroje informací a podkladových dat uvádím v seznamu zdrojů.

V Praze dne

.....

(podpis autora)

PODĚKOVÁNÍ

Poděkovat bych chtěl zejména vedoucímu této diplomové práce panu Prof. Ing. Aleši Čepkovi, CSc. Dále bych rád poděkoval také oponentovi této diplomové práce panu Doc. Ing. Martinu Štronerovi, Ph.D. za odbornou konzultaci.

Obsah

Úvod	4
1 O projektu GNU Gama.....	5
2 Jak získat soubory.....	6
2.1 Git Bash	6
2.2 GitHub Shell	7
3 Grafické uživatelské rozhraní (gama-g2)	8
3.1 Hlavní ovládací panel (Main Control Panel)	8
3.1.1 Databáze (Database).....	8
3.1.2 Vyrovnání (Adjustment).....	10
3.1.3 Nástroje (Tools).....	11
3.1.4 Nápověda (Help).....	11
3.2 Panel pro vyrovnání sítí (Network Adjustment Panel).....	12
3.2.1 Soubor (File).....	12
3.2.2 Vyrovnání (Adjustment).....	12
3.2.3 Upravit (Edit).....	12
3.2.4 Nastavení (Settings).....	12
3.2.5 Nástroje (Tools).....	13
3.2.6 Nápověda (Help).....	13
4 Matematický základ.....	14
4.1 Mezní rozdíl.....	14
4.2 Mezní úhlový uzávěr.....	14
4.3 Koeficient spolehlivosti	16
4.4 Grubbsův oboustranný test	17
5 Qt	18
6 Qt - tabulky.....	19
6.1 Filtrace	19
6.2 Řazení	20

6.3	Zaškrťovací tlačítka	20
6.4	Tlačítka	21
6.5	Barva pozadí	21
7	Qt - Widgets Application.....	22
8	Qt – signály a sloty	24
9	Qt - pluginy.....	25
10	Pluginy – obecně	25
10.1	Pairwise Distances Plugin	26
10.2	Multiple Distances Plugin	26
10.3	Triangular Closures Plugin.....	26
11	Pluginy – z hlediska uživatele	27
11.1	Pairwise Distances Plugin	27
11.2	Multiple Distances Plugin	27
11.3	Triangular Closures Plugin.....	30
12	Pluginy – z hlediska programování	31
12.1	Pairwise Distances Plugin	31
12.2	Multiple Distances Plugin	32
12.3	Triangular Closures Plugin.....	33
12.4	Použité kontejnery.....	33
12.5	Vyhledání měřených délek.....	34
12.6	Vyhledání dvojic měřených délek.....	34
12.7	Vyhledání úhlových měření	35
12.8	Vyhledání trojice úhlů v trojúhelníku	35
13	Nahrávání souborů na GitHub	36
14	Vytvoření spustitelné aplikace.....	39
	Závěr.....	40
	Seznam proměnných.....	41
	Seznam zkratk.....	42
	Seznam obrázků.....	42

Seznam tabulek.....	43
Seznam grafů	43
Seznam příloh.....	43
Seznam zdrojů	44

Úvod

Tato diplomová práce se věnuje rozšíření funkcionality grafického uživatelského rozhraní systému GNU Gama [1]. To je projekt zaměřený na vyrovnání geodetických sítí, jehož autorem je Prof. Ing. Aleš Čepek, CSc.

Před vlastním vyrovnáním je třeba vyloučit chybná měření. Cílem této diplomové práce je vytvořit pluginy (zásuvné moduly) [2], které dokáží vyloučit právě taková měření.

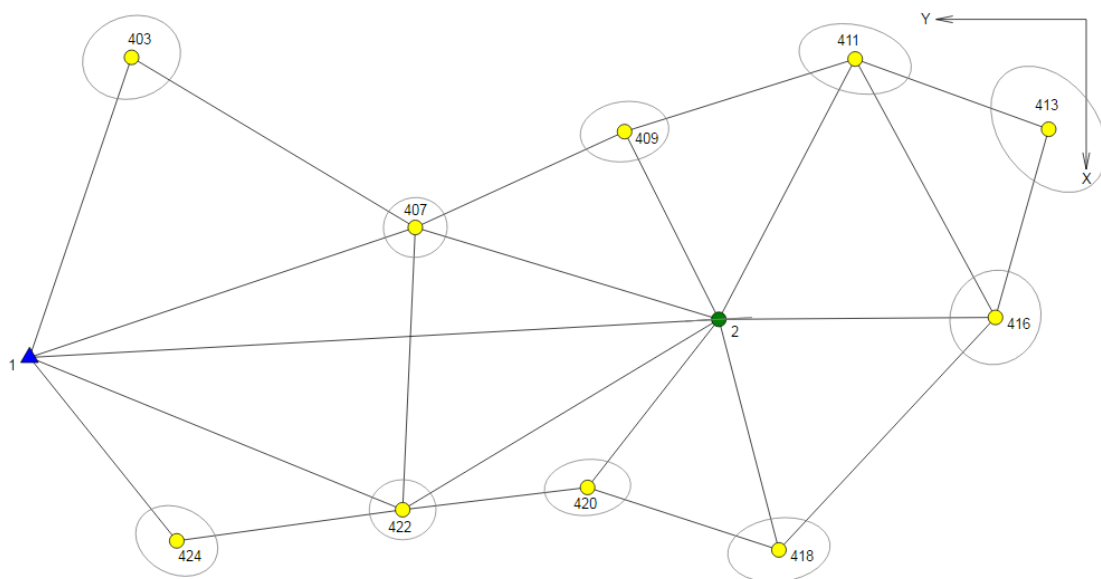
Prvním úkolem je identifikovat chybná měření délek, a to buď vodorovných anebo šikmých (redukovaných na spojnici stabilizačních značek). Odlehlá měření je poté nutné před samotným vyrovnáním vyloučit.

Druhým úkolem je zjistit zdali nebyly překročeny mezní úhlové uzávěry u všech kombinací trojúhelníků v síti. Úhly v trojúhelnících přitom mohou být určeny buď přímým měřením úhlů anebo pomocí rozdílu dvou směrů.

K samotnému vyhotovení pluginů je třeba použít platformu *Qt* [3], jelikož grafické uživatelské rozhraní projektu *GNU Gama* je vytvořeno právě na této platformě. Programovacím jazykem v tomto projektu je *c++* ve standardu *c++11*.

1 O projektu GNU Gama

Projekt je zaměřen na vyrovnání geodetických sítí. Skládá se ze dvou částí, a to z *gama-g2* (resp. *gama-local*) a *gama-g3*. Část *gama-g2* se věnuje výpočtům v lokálních souřadnicových systémech. A dále část *gama-g3* je určena pro výpočty v prostorových souřadnicových systémech.



Obr. 1.1 Ukázka geodetické sítě [1].

Původní verze programu byla spustitelná pouze pomocí příkazového řádku (*gama-local*). Vstupem byl XML soubor, ve kterém byly zadávány jak souřadnice, tak vlastní měření. V tomto souboru lze také volit různé parametry geodetické sítě.

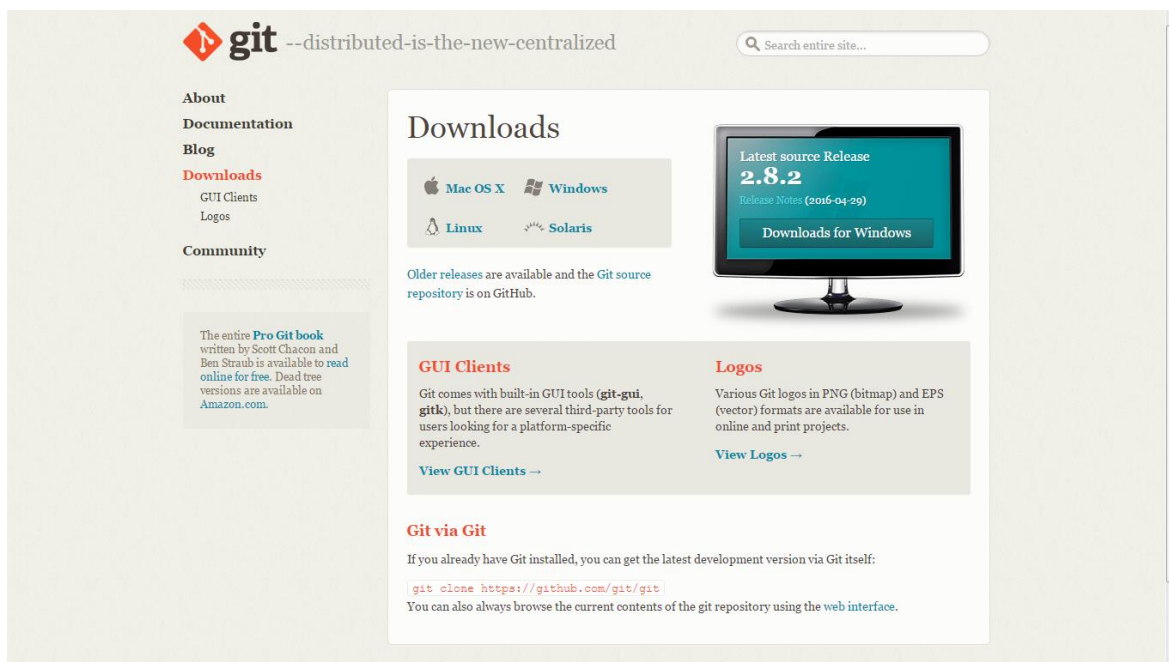
Grafická verze programu také pracuje s tímto typem souborů, ale ty je nutné nejprve naimportovat do databáze. Program je totiž založen na práci s databází. Pro grafickou verzi jsou také tvořeny pluginy, které jsou výsledkem této diplomové práce.

2 Jak získat soubory

Program je distribuován pomocí systému pro správu verzí *Git*, který byl původně vyvinut pro *Linux* Linusem Torvaldsem [4]. Dále zde uvádím několik možností jak používat *Git* k získání souborů.

2.1 Git Bash

První možností je nainstalovat si *Git Bash* [5]. K dispozici jsou verze pro různé operační systémy.



Obr. 2.1 Webová stránka, na které lze stáhnout program *Git Bash* [5].

Práce s ním je celkem snadná. Stačí zadat adresu za příkaz *git clone*. Zde je krátká ukázka:

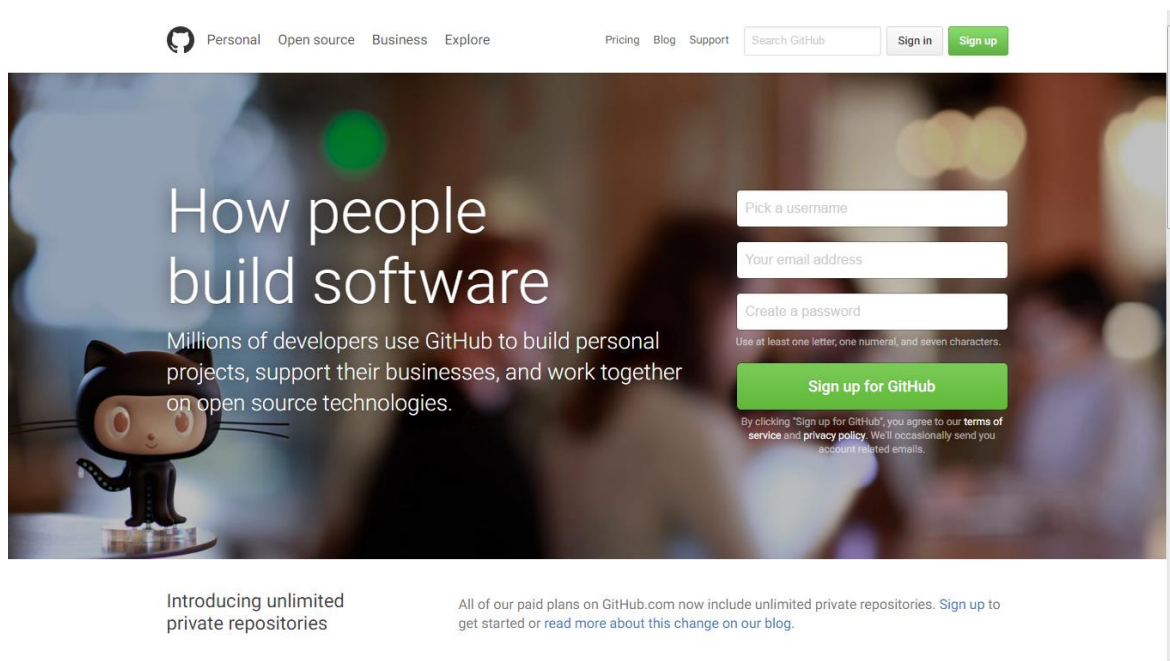
```
MINGW64:/
Celtian@Celtian-PC MINGW64 /
$ git clone git://git.sv.gnu.org/gama/examples.git
Cloning into 'examples'...
remote: Counting objects: 346, done.
remote: Compressing objects: 100% (131/131), done.
remote: Total 346 (delta 208), reused 346 (delta 208)
Receiving objects: 100% (346/346), 151.35 KiB | 138.00 KiB/s, done.
Resolving deltas: 100% (208/208), done.
Checking connectivity... done.
Celtian@Celtian-PC MINGW64 /
$ |
```

Obr. 2.2 Ukázka softwaru *Git Bash*.

Soubory budou poté k dispozici v adresáři, ve kterém je *Git* nainstalován, pokud není explicitně zadán jiný adresář.

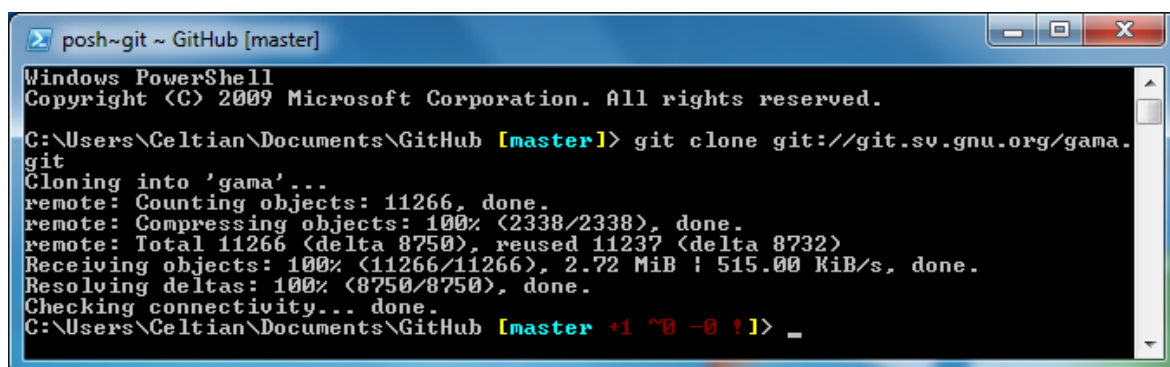
2.2 GitHub Shell

Druhou možností je použít webovou službu *GitHub* [6], která podporuje nástroj *Git*. Služba byla spuštěna v roce 2008 [7]. Tam je však nutné se nejprve registrovat.



Obr. 2.3 Webová stránka služby *GitHub* [6].

Podobně jako *Git Bash* funguje i příkazová řádka (*Shell*) služby *GitHub*.



Obr. 2.4 Ukázka softwaru *GitHub Shell*.

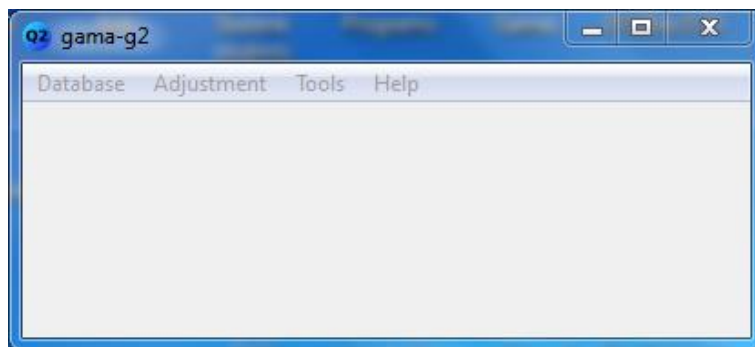
Soubory jsou staženy do aktivního adresáře. Ve výchozím stavu do složky *GitHubu* v dokumentech.

3 Grafické uživatelské rozhraní (gama-g2)

Program se skládá ze dvou hlavních oken a tj. hlavní ovládací panel (*Main Control Panel*) a panel pro vyrovnání sítě (*Network Adjustment Panel*), kterých může být otevřeno více najednou.

3.1 Hlavní ovládací panel (Main Control Panel)

Po spuštění programu se zobrazí okno, obsahující pouze nástrojovou lištu. Obsahuje čtyři položky: Databáze (*Database*), Vyrovnání (*Adjustment*), Nástroje (*Tools*) a Nápověda (*Help*).

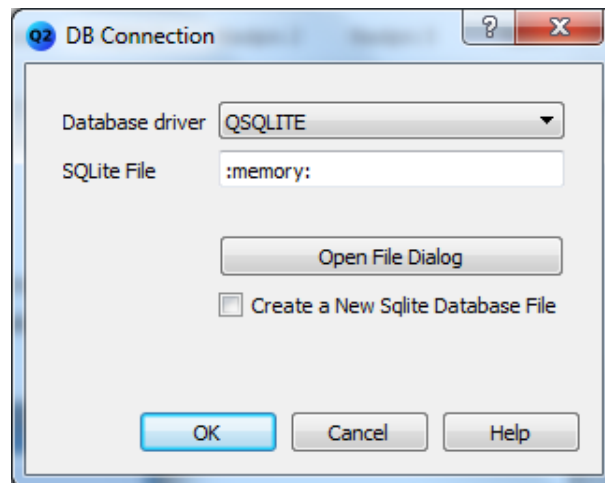


Obr. 3.1 Hlavní okno programu *gama-g2*.

3.1.1 Databáze (Database)

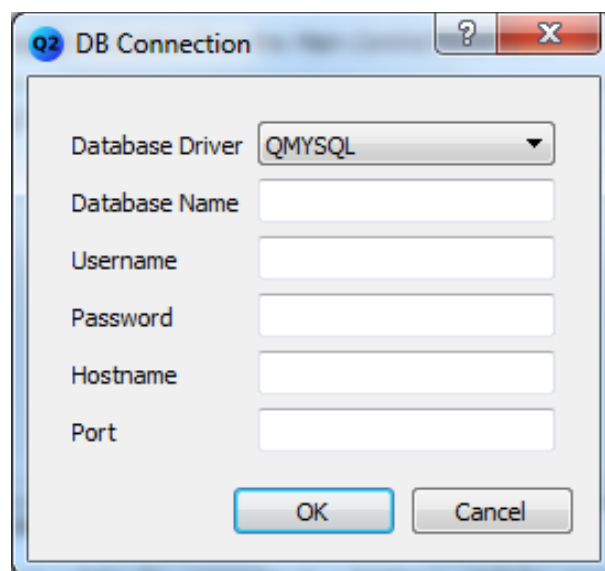
V položce databáze jsou nástroje pro manipulaci s databázemi.

Před tím než bude zahájena práce s programem je nutné buď otevřít, nebo založit novou databázi. Po spuštění *Database/Connect to Database* se zobrazí dialogové okno. Tam je nutné zadat databázový ovladač a jméno databáze. Ve výchozím stavu je nastaveno na hodnotu *:memory:*, což znamená, že se databáze vytvoří pouze interně do paměti nikoliv fyzicky na disk.



Obr. 3.2 Dialogové okno pro připojení databáze pro *SQLite* databázi.

Je-li vybrán jiný ovladač než *QSQLITE*, tak vypadá dialogové okno jinak, jelikož jiné databáze vyžadují odlišné vstupní informace.

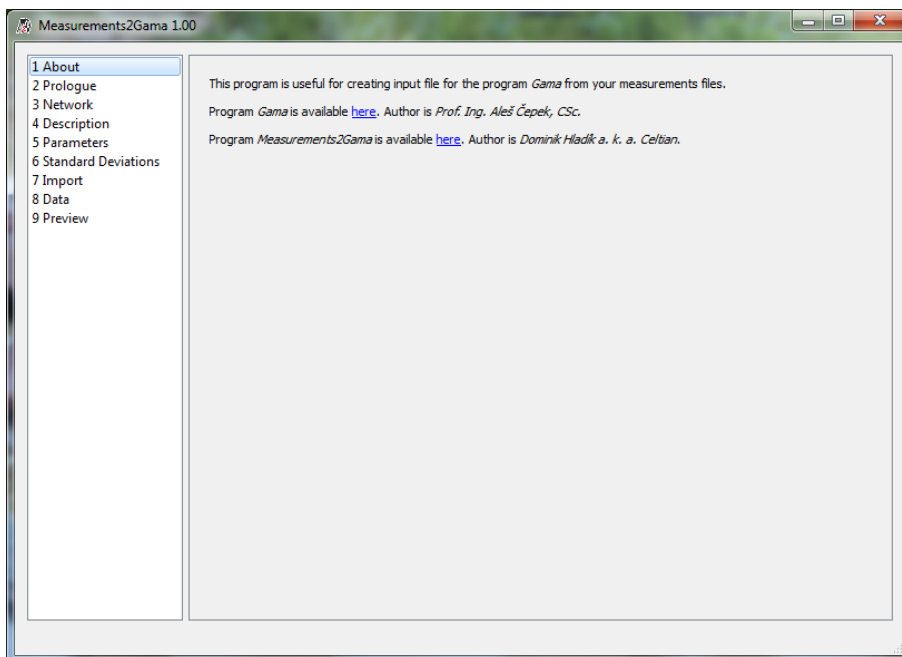


Obr. 3.3 Dialogové okno pro připojení databáze pro *MySQL* databázi.

Databázi je možné po jejím otevření naplnit pomocí XML souboru. A dále je možné z databáze odstranit všechny tabulky nebo pouze data v nich. Další možností je odstranit pouze data pro vybranou geodetickou síť nebo je tu ještě možnost načíst do databáze příklady.

Pokud není k dispozici XML soubor, ale máme k dispozici data, můžeme využít program *Measurements2Gama* [8], který vytvoří tento soubor z CSV souboru. Výhodné je

to například tehdy, máme-li data v nějakém tabulkovém procesoru, nebo když chceme zabránit tomu, aby při ruční tvorbě vznikla chyba.



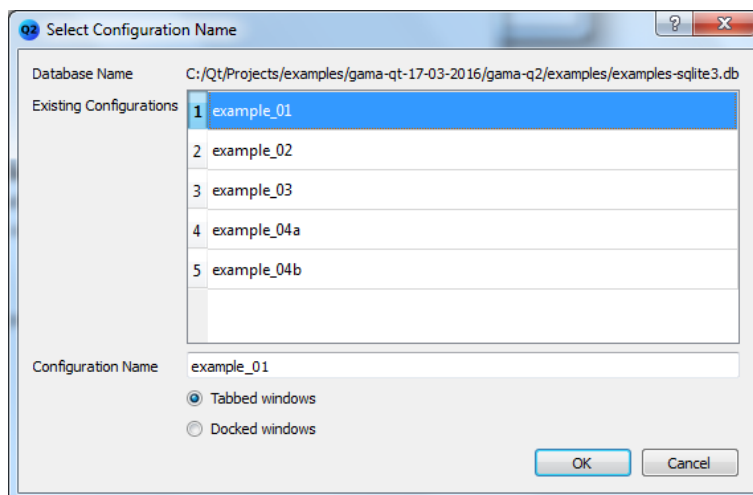
Obr. 3.4 Grafické uživatelské rozhraní programu *Measurements2Gama* [8].

Tento program nevznikl v rámci této diplomové práce, ale autorem jsem též já. Testován byl na konsolové verzi (příkazový řádek) programu *gama-local* ve vybraných verzích tudíž nemusí být plně kompatibilní se všemi verzemi.

V této diplomové práci byl využit pro vytvoření testovacích příkladů.

3.1.2 Vyrovnání (Adjustment)

Po výběru geodetické sítě je zde možné otevřít panel pro vyrovnání sítí, kde lze provést vyrovnání a další operace.

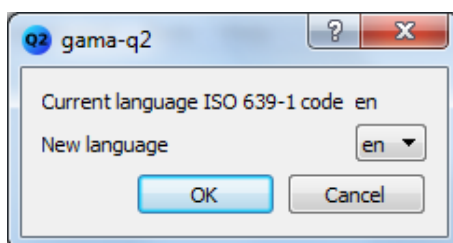


Obr. 3.5 Dialog pro výběr geodetické sítě z databáze.

Při výběru je možné si zvolit, zdali bude možné z hlavního okna jednotlivé karty vytáhnout či nikoliv. Pokud chceme, aby je bylo možné vytahovat, zvolíme dokovací okna (*Docked windows*) pokud nikoliv zvolíme druhou volbu (*Tabbed windows*). Dokovací okna jsou taková okna, která je možné vytáhnout panelu pro vyrovnání sítí (*Network Adjustment Panel*) a dále s nimi pracovat jako se samostatnými okny. Zvolíme-li druhou možnost, znamená to, že se nám tato okna zobrazí ve formě karet, které nelze vytáhnout z panelu pro vyrovnání sítí (*Network Adjustment Panel*).

3.1.3 Nástroje (Tools)

Zde je možné změnit jazykovou lokalizaci výstupních souborů. Více nástrojů zde nebylo naimplementováno.



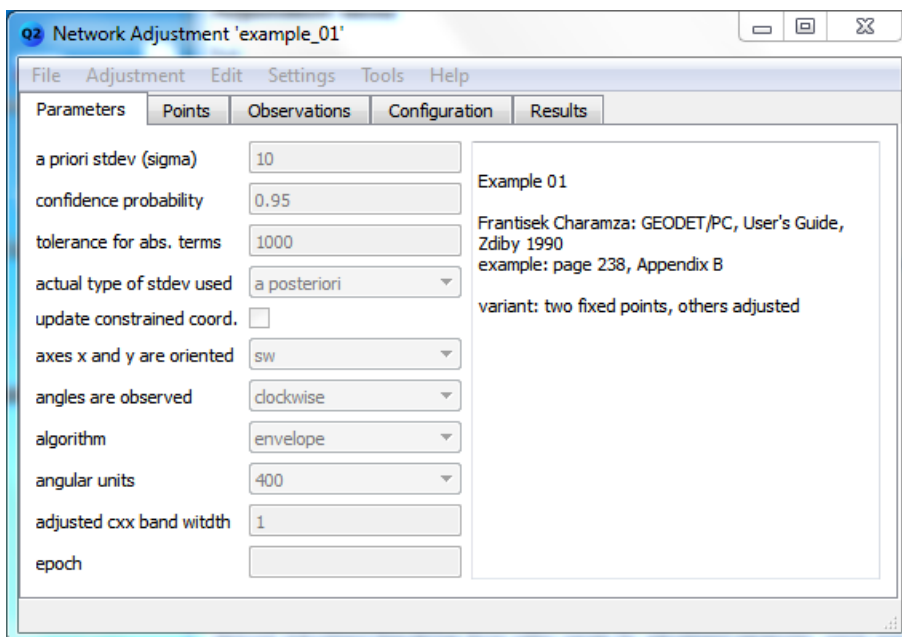
Obr. 3.6 Dialogové okno, kde je možné změnit jazyk výstupních souborů.

3.1.4 Nápověda (Help)

Tady je možné získat informace o jednotlivých funkcích tohoto programu, dále obecně informace o tomto programu a dále také o platformě *Qt* [3].

3.2 Panel pro vyrovnání sítí (Network Adjustment Panel)

Hlavní okno se skládá z pěti karet (resp. dokovacích oken). V horní části je opět umístěna nástrojová lišta.



Obr. 3.7 Panel pro vyrovnání sítí (*Tabbed Windows*).

3.2.1 Soubor (File)

V tomto menu je možné uložit nebo vytisknout různé výstupy. Na výběr je hned z několika formátů (např. HTML, XML, SQL či jako text).

3.2.2 Vyrovnání (Adjustment)

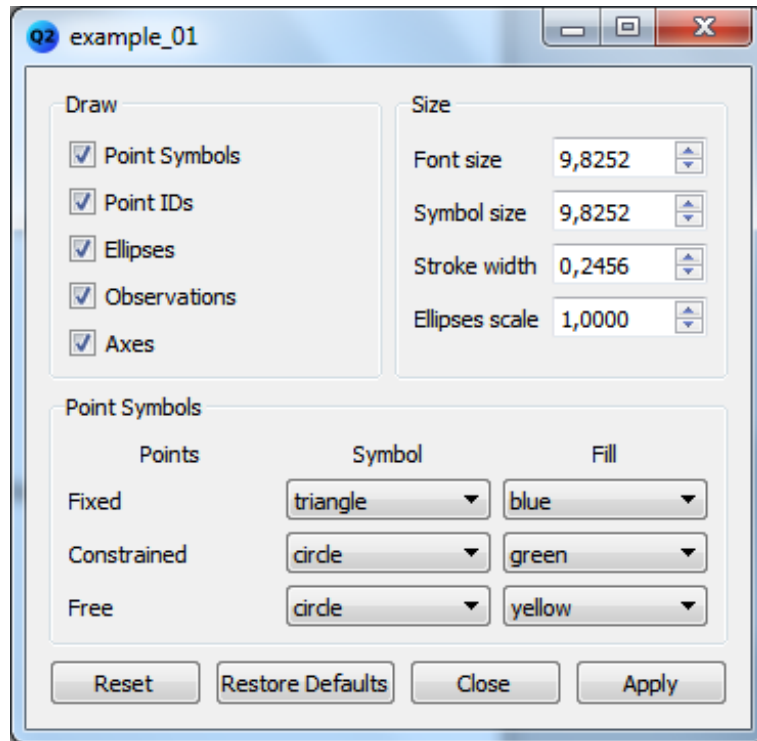
Zde se spouští výpočet vyrovnání.

3.2.3 Upravit (Edit)

Tady je možné povolit editaci parametrů sítě, seznamu bodů či seznamu měření.

3.2.4 Nastavení (Settings)

Nastavení je možné měnit pro zobrazení sítě. Lze zde zvolit obsah, který bude zobrazen. Dále je tu možné měnit bodové symboly a jejich velikost, tloušťku čar, velikost fontu a měřítko elips chyb.



Obr. 3.8 Nastavení zobrazení.

3.2.5 Nástroje (Tools)

Tato položka je věnována pluginům.

3.2.6 Nápověda (Help)

Tady je opět místo, ze kterého je možné se dostat do stejné nápovědy jako z hlavního ovládacího panelu.

Zdrojem informací pro tuto kapitolu byla nápověda programu *gama-g2*.

4 Matematický základ

V této kapitole popisují, jaký matematický aparát stojí za pluginy vyhotovené v rámci této diplomové práce.

4.1 Mezní rozdíl

Pro testování dvojic obousměrně měřených délek bylo třeba zjistit, zdali je možné tuto dvojici nadále použít do vyrovnání či nikoliv. Pro tento účel byl jako testovací kritérium použit tzv. mezní rozdíl:

$$\Delta_M = u_p \cdot \sigma_\Delta. \quad 4.1$$

Ve výrazu je použit koeficient spolehlivosti u_p a jak jej určit je uvedeno v samostatné kapitole.

Ve vzorci pro mezní rozdíl se dále také vyskytuje směrodatná odchylka mezního rozdílu. K tomu, aby byla vypočtena, potřebujeme znát směrodatné odchylky obou dvojic měření:

$$\sigma_\Delta = \sqrt{\sigma_1^2 + \sigma_2^2}. \quad 4.2$$

Jsou-li obě dvě hodnoty měřeny se stejnou přesností, vzorec se zjednoduší:

$$\sigma_\Delta = \sqrt{2} \cdot \sigma. \quad 4.3$$

Vlastní test probíhá tak, že je-li absolutní hodnota rozdílu vyšší než mezní rozdíl, je nutno přidat další měření. Taková hodnota je poté v programu označena červeně.

4.2 Mezní úhlový uzávěr

Hodnocení přesnosti úhlových uzávěrů byl použit, podobně jako u mezního rozdílu, mezní uzávěr:

$$U_M = u_p \cdot \sigma_U. \quad 4.4$$

Opět se zde vyskytuje koeficient spolehlivosti u_p , který je popsán v samostatné kapitole.

A dále je tu směrodatná odchylka úhlového uzávěru, kterou vypočteme jako součet čtverců směrodatných odchylek jednotlivých úhlů (resp. směrů) pod odmocninou.

Jedná-li se o výhradně úhlové měření v trojúhelníku, vzorec vypadá takto:

$$\sigma_U = \sqrt{\sigma_{\omega_1}^2 + \sigma_{\omega_2}^2 + \sigma_{\omega_3}^2}, \quad 4.5$$

nebo, v případě rovnosti přesnosti jednotlivých úhlů, například takto

$$\sigma_U = \sqrt{3} \cdot \sigma_{\omega}. \quad 4.6$$

Jedná-li se výhradně o měření vodorovných směrů, vzorec získá tuto podobu:

$$\sigma_U = \sqrt{\sigma_{\varphi_1}^2 + \sigma_{\varphi_2}^2 + \sigma_{\varphi_3}^2 + \sigma_{\varphi_4}^2 + \sigma_{\varphi_5}^2 + \sigma_{\varphi_6}^2}, \quad 4.7$$

a při rovné přesnosti jednotlivých měřených směrů tuto:

$$\sigma_U = \sqrt{6} \cdot \sigma_{\varphi}. \quad 4.8$$

Nastat může ovšem i případ, že některý úhel v trojúhelníku je určen pomocí rozdílu dvojic směrů. Proto bylo nutné znát i vztah pro převod směrodatných odchylek dvou směrů na směrodatnou odchylku úhlu:

$$\sigma_{\omega} = \sqrt{\sigma_{\varphi_1}^2 + \sigma_{\varphi_2}^2}, \quad 4.9$$

nebo pokud mají směry shodnou přesnost:

$$\sigma_{\omega} = \sqrt{2} \cdot \sigma_{\varphi}. \quad 4.10$$

V pluginu bylo využito vzorce s odlišnými směrodatnými odchylkami. Pokud se v trojúhelníku vyskytl úhel určený jako rozdíl dvou směrů, byla jejich směrodatná odchylka přepočtena na směrodatnou odchylku úhlu, který svírají a poté byla dosazena do vzorce.

Vlastní testování probíhá naprosto stejně jako u mezního rozdílu. Absolutní hodnota úhlového uzávěru je porovnána s mezní hodnotou, a když je překročena, pak lze předpokládat, že v trojúhelníku se vyskytuje odlehlé měření.

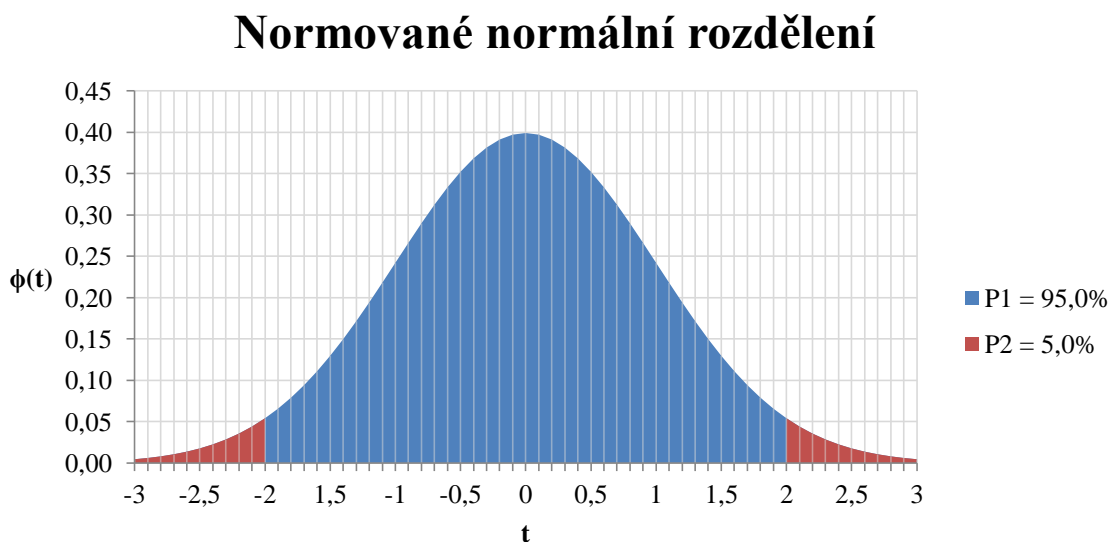
4.3 Koeficient spolehlivosti

Ve vzorcích pro mezní rozdíl a mezní úhlový uzávěr se vyskytuje koeficient spolehlivosti u_p . Ten by měl být zpravidla volen dle následující tabulky:

P	u_p	Zásady volby
95,0%	2,0	jednoduchá, snadno kontrolovatelná měření
98,8%	2,5	složitá, těžko kontrolovatelná měření
99,7%	3,0	měření za nepříznivých podmínek nebo měření zatížená obtížně vylučitelnými systematickými měřeními
více	více	výjimečně, u velmi důležitých měření

Tab. 4.1 Volba koeficientu spolehlivosti pro jednorozměrné veličiny [9].

Jelikož v geodetických sítích v programu gama-g2 je již obsažena hodnota pravděpodobnosti (*Confidence level*), byla tato hodnota použita i ve vzorci pro výpočet mezního rozdílu. Vzájemné vztahy mezi hodnotami pravděpodobnosti a u_p jsou vidět z tohoto grafu:



Graf 4.1 Normální rozdělení - $N(\mu = 0, \sigma = 1)$ [9].

Obsah modré plochy je v tomto případě roven pravděpodobnosti 95,0% (*Confidence level*). V našem případě hledáme koeficient spolehlivosti u_p na vodorovné ose ($u_p = t$).

Hodnota u_p bude rovna po zaokrouhlení 1,96, což přibližně odpovídá hodnotě 2,0 z výše uvedené tabulky. Dále grafu použit oboustranný intervalový odhad, protože mezní rozdíl může nabývat kladné i záporné hodnoty.

4.4 Grubbsův oboustranný test

Má-li soubor měřených veličin charakter normálního rozdělení, lze testovat jejich extrémní hodnoty pomocí Grubbsova testu (někdy také nazývané jako Pearson-Sekharův test). Test lze provést, máme-li počet veličin v souboru vyšší než dva a směrodatná odchylka jednotlivých veličin je shodná. Pro nízký počet veličin není zřejmé, že má soubor charakter normálního rozdělení, proto je vhodné používat tento test při vyšším počtu opakovaných měření.

Vlastní test probíhá tak, že nejprve vypočteme střední opravu:

$$m_v = \sqrt{\frac{\sum v^2}{n}}. \quad 4.11$$

Dále je určena kritická hodnota Grubbsova oboustranného testu, dle vzorce:

$$K_{G(\alpha,n)} = \frac{n-1}{\sqrt{n}} \cdot \sqrt{\frac{t(\alpha,n)}{n-2+t(\alpha,n)^2}}, \quad 4.12$$

kde se vyskytuje kritická hodnota Studentova rozdělení pro $n - 2$ stupně volnosti. (Ta byla získána výpočtem z již vytvořených knihoven programu *gama-g2*.)

Poté byla určena maximální oprava, která byla použita do vzorce pro výpočet testovacího kritéria:

$$K_1 = \frac{v_{max}}{m_v}, \quad 4.13$$

které bylo posléze porovnáváno s kritickou hodnotou:

$$K_{1(\alpha,n)} = K_{G(\alpha,n)} \cdot \sqrt{\frac{n}{n-1}}. \quad 4.14$$

Platí-li:

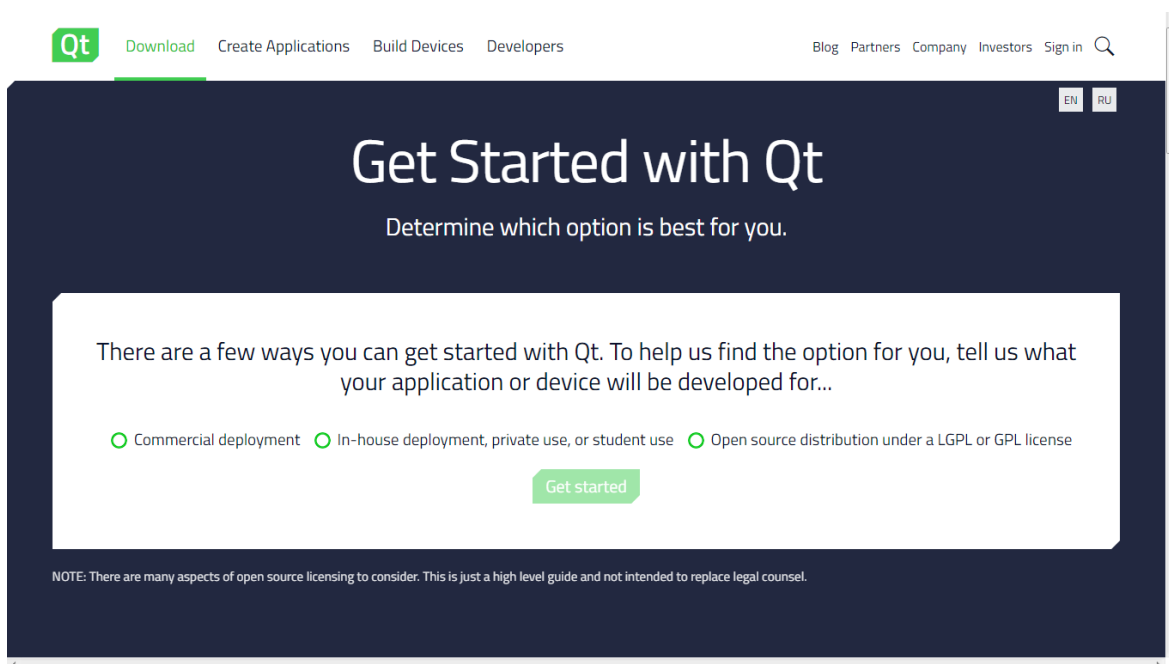
$$K_1 > K_{1(\alpha, n)}, \quad 4.15$$

Nejodlehlejší měření můžeme ze souboru vyloučit.

Test je poté nutné test zopakovat, dokud soubor obsahuje odlehlá měření.

5 Qt

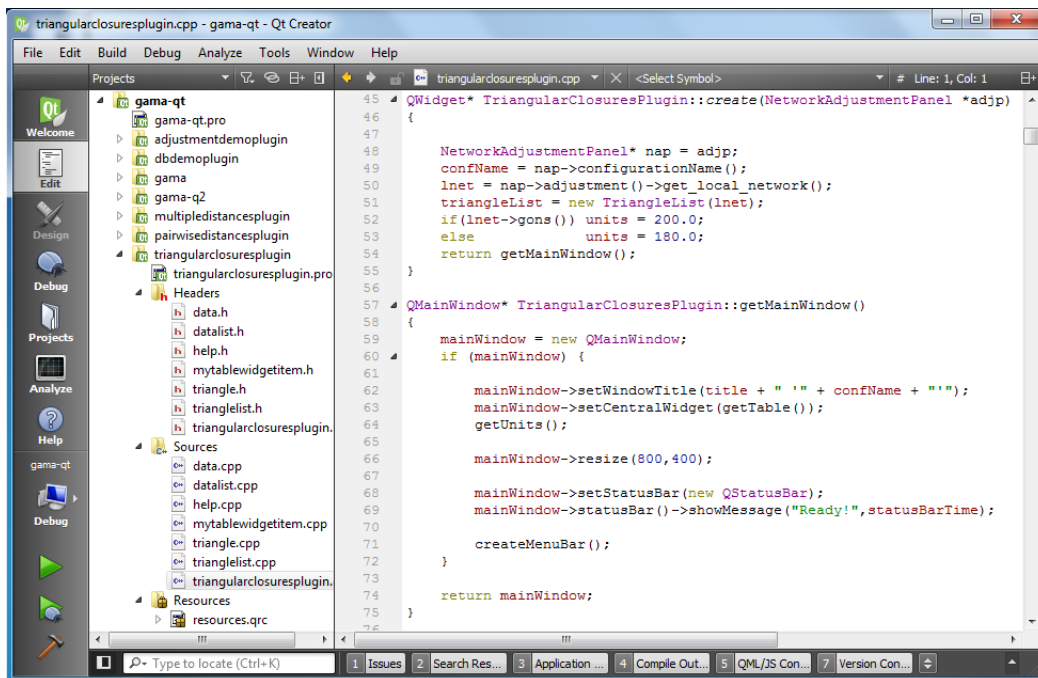
Pro vlastní tvorbu pluginů pro program *gama-g2* byla zvolena platforma *Qt* [3]. Její výhodou je možnost využít grafické uživatelské rozhraní a velké množství knihoven. K dispozici je v několika verzích podle licenčních podmínek a operačních systémů.



Obr. 5.1 Webová stránka platformy *Qt* [3].

Ke stažení je však nutná registrace, kterou je potřeba zadat do instalačního průvodce při vlastní instalaci. Na stránkách této platformy lze kromě *online* instalačního programu, je možné nalézt i jeho *offline* verzi.

Součástí platformy *Qt* je program *Qt Creator* a ten lze ovládat tak, že si nejprve v levé části vyberete to, co chcete zobrazit a obsah se poté objeví v pravé části.



Obr. 5.2 Program *Qt Creator*.

Pro vývoj pluginů byla použita verze *Qt Creator 3.6.1 (based on Qt 5.6.0)*.

6 Qt - tabulky

Před tím než je zvoleno, jaká tabulka v konkrétním pluginu bude použita, je dobré si rozmyslet, odkud budou čerpána data a jakého rozsahu bude práce s ní. Máme k dispozici dva základní druhy tabulek, které můžeme použít:

1. *QTableWidget*

Tuto tabulku používáme, když chceme prezentovat jednoduchá data a na pozadí nemáme žádný model.

2. *QTableView*

Toto je tabulka založená na modelu. Modelem může být v podstatě cokoliv (např. tabulka z databáze).

Jelikož bylo náplní mé práce tvorba pluginů a pluginy mají být co možná nejjednodušší, zvolena byla první možnost. Pokud bych se pouštěl do nějakého rozsáhlejšího programu, pravděpodobně bych zvolil druhou možnost.

6.1 Filtrace

Během vývoje pluginů vznikl požadavek na odfiltrování řádků nebo sloupců v tabulkách. To ovšem neznamená, že jsou řádky nebo sloupce z tabulky odstraněny, ale pouze nejsou viditelné. V pluginech byla zachována možnost vybrané sloupce vypínat a

zapínat. Dále je tam několik pomocných sloupců, které slouží k identifikaci řádků nebo k nějakému složitějšímu řazení. Ty nejsou pro běžného uživatele nikdy vidět.

6.2 Řazení

Třída *QTableWidget* řadí hodnoty jako text. Proto bylo nutné zajistit řazení jednotlivých položek *QTableWidgetItem* jako číslo. Proto bylo třeba přetížít operátor pro řazení.

QTableWidget::operator<		
porovnání	číslo	prázdný text
číslo	porovnání	text < číslo
prázdný text	text < číslo	neporovnáváme

Tab. 6.1 Operátor pro řazení záznamů v tabulce.

Jelikož chceme, aby při vzestupném řazení byly na prvním místě texty a teprve poté čísla, při porovnání textu a čísla bude mít vždy text nižší hodnotu. Pokud porovnáváme dvě čísla, porovnáme jejich hodnoty. Když porovnáváme texty, neprovádíme nic, jelikož tabulky obsahují pouze prázdné texty (texty o nulové délce).

6.3 Zaškrťovací tlačítka

Bylo třeba, aby tabulka obsahovala zaškrťovací tlačítka (*QCheckBoxes*). K dosažení tohoto cíle byly k dispozici tři cesty.

1. *QTableWidgetItem::checkState(Qt::CheckState)*

U této metody je do pole v tabulce kromě textu umístěno i zaškrťovací tlačítko.

2. *QTableWidget::(int row, int colulmn, QWidget* widget)*

V tomto případě je do pole v tabulce umístěn libovolný *widget*, který není nutné aktivovat.

3. *QAbstractItemView::setItemDelegateForColumn(QAbstractItemDelegate *delegate)*

Tato volba je asi nejpracnější jelikož je nutné si nejprve vytvořit vlastní třídu odvozenou od *QAbstractItemDelegate*. Odlišují se zde data, která jsou zobrazována v tabulce (*EditorData*) a ty které zdrojem (*ModelData*).

Jelikož tabulka v našem případě nebyla založena na modelu, tak byla zvolena druhá možnost. Druhým důvodem byl ten fakt, že *widgety* použité skrze druhou možnost je

možné používat i při zakázané editaci dat v tabulce, čehož bylo využito. Jinak bych nejspíše volil třetí možnost, kterou považuji za nejčistší řešení.

6.4 Tlačítka

Při vývoji bylo dále nutné, aby z jednoho řádku z tabulky byla vyvolána další tabulka. K tomu byly použity tlačítka (*QPushButtons*). Možnosti jak toho docílit byly dvě, a to stejně jako u zaškrťovacích tlačítek (*QCheckBoxes*):

1. *QTableWidget::(int row, int colulmn, QWidget* widget)*
2. *QAbstractItemView::setItemDelegateForColumn(QAbstractItemDelegate *delegate)*

Zvolena byla, stejně jako u zaškrťovacích tlačítek, první možnost.

6.5 Barva pozadí

Pro lepší přehlednost je dobré buď sloupce anebo řádky barevně odlišit. Pro naše účely připadají v úvahu tři varianty:

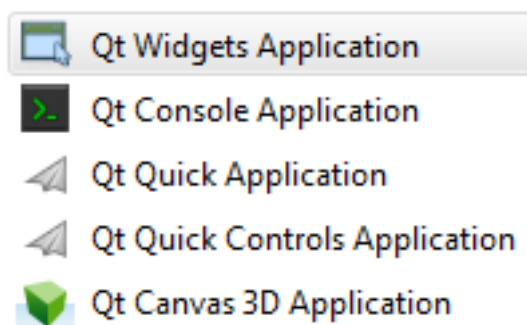
1. *QTableWidget::setAlternatingRowColors(bool enable)*
Toto je možnost, která nám obarví střídavě jednotlivé řádky. Toho je dobré využít, když se jedná o rozsáhlé tabulky s úzkými řádky, kde není jednoduché najít požadovaná data.
2. *QTableWidgetItem::setData(int role, const QVariant &value)*
Nastavíme-li roli na *Qt::BackgroundRole*, lze nastavit libovolnému záznamu v tabulce barvu pozadí.
3. *QWidget::setStyleSheet(const QString & styleSheet)*
Barvu pozadí *widgetu*, který jsme použili v tabulce lze změnit pomocí tzv. kaskádových stylů (*CSS*) [10]. Kaskádové styly jsou hojně používány například při psaní *HTML*. Slouží například k zobrazení elementů na webových stránkách.

V tabulkách byla využita druhá a třetí možnost. Voleny byly světlé barvy, aby nerušily obsah jednotlivých záznamů v tabulce, které byly naopak tmavé.

Zdrojem informací pro tuto kapitolu byla nápověda programu *Qt Creator* [11].

7 Qt - Widgets Application

Platforma *Qt* umožňuje vytvářet několik druhů aplikací:



Obr. 7.1 Různé druhy aplikací v prostředí *Qt*.

Gama-g2 používá *QtWidgetsApplication* a stejně tak tomu bylo u i pluginů, které jsou součástí této diplomové práce.

Při vytváření jsme dotázáni, jakou základovou třídu máme použít. Na výběr máme tři možnosti:

1. *QMainWindow*

Tato volba nám umožňuje použít okno, které lze zvětšovat či zmenšovat. Pomocí tlačítka v pravém či levém horním rohu (dle *OS*). Dále obsahuje stavový řádek (*QStatusBar*), který je ovšem ve výchozím stavu skryt.

2. *QDialog*

Na rozdíl od *QMainWindow* dialog nemá v pravém či v levém horním rohu tlačítka pro zvětšení či zmenšení okna. Dále se zde často využívá standardních tlačítek umístěných ve svém vlastním boxu (*QDialogButtonBox*), kde jsou tlačítka řazena podle jejich role. Lišit se tak bude jejich pořadí na různých *OS*.

3. *QWidget*

QWidget může být v podstatě cokoliv. Zároveň je základovou třídou obou dvou předchozích možností.

Dalšími možnostmi jsou například:

1. *QFrame*

Který se chová podobně jako *QMainWindow*.

2. *QMessageBox*

Toho se využívá při krátkých dialogích či varovných zprávách, informacích atp.

Platforma *Qt* dále umožňuje kombinovat práci s grafikou nebo je možné vše napsat přímo do zdrojového kódu. Kód může vypadat například takto:

```
QDialog *dialog = new QDialog;
dialog->setWindowModality(Qt::ApplicationModal);
dialog->setWindowTitle("Title");
dialog->show();
```

Obr. 7.2 Ukázka toho, jak vytvořit a zobrazit dialog.

V ukázce je na prvním řádku alokován dialog v paměti (*heap*). Na druhém řádku je nastavena modalita dialogového okna, což znamená, jak se bude chovat dialog po jeho zobrazení.

Qt::WindowModality		
Konstanta	Hodnota	Význam
NonModal	0	Okno neblokuje využívání žádných oken.
WindowModal	1	Okno blokuje pouze okna, ze kterých je odvozeno.
ApplicationModal	2	Okno blokuje všechna ostatní okna.

Tab. 7.1 Modalita jednotlivých oken a její význam [11].

Na dalším řádku je pak nastaven název okna. A na posledním řádku je pak ukázáno jak vlastní dialogové okno zobrazit.

Pro veškeré pluginy a dokonce i pro celý program *gama-g2*, nebylo využíváno grafického rozhraní platformy *Qt*.

Zdrojem informací pro tuto kapitolu byla nápověda programu *Qt Creator* [11].

8 Qt – signály a sloty

Objekty mohou generovat tzv. signály (*signals*) a přijímat tzv. sloty (*slots*). Ty byly v pluginech velmi často využívány.

1. Signály (*signals*)

Představují akci, na kterou se má vyvolat nějaká reakce. Může to být například kliknutí na tlačítko (*QPushButton::clicked(bool)*) nebo spuštění akce (*QAction::triggered()*).

2. Sloty (*slots*)

Představují reakci na určitý signál. Jako příklad mohu uvést zavření dialogového okna (*QDialog::close()*) nebo nastavení textu do widgetu *QLineEdit* (*QLineEdit::setText(QString)*).

V pluginech toho bylo využito například při změně stavu zaškrťávacích tlačítek. Nejčastěji bylo, namísto již existujících slotů v knihovně *Qt*, využito vlastních slotů.

Jelikož program *gama-g2* je psán ve standardu *c++11*, je možné využít pro psaní slotů lambda funkce. Rozdíl je vidět na tomto obrázku:

```
connect(action, SIGNAL(triggered()), this, SLOT(close()));  
connect(action, &QAction::toggled, [this]() {close();});
```

Obr. 8.1 Signály a sloty bez a s využitím lambda funkce.

Při použití lambda funkcí může nastat situace, kdy je signál přetížen. Jak to řešit je vidět na následující ukázce:

```
connect(spinBox, static_cast<void (QSpinBox::*)(int)>(&QSpinBox::valueChanged),  
       [spinBox, anotherSpinBox]() {anotherSpinBox->setValue(spinBox->value());});
```

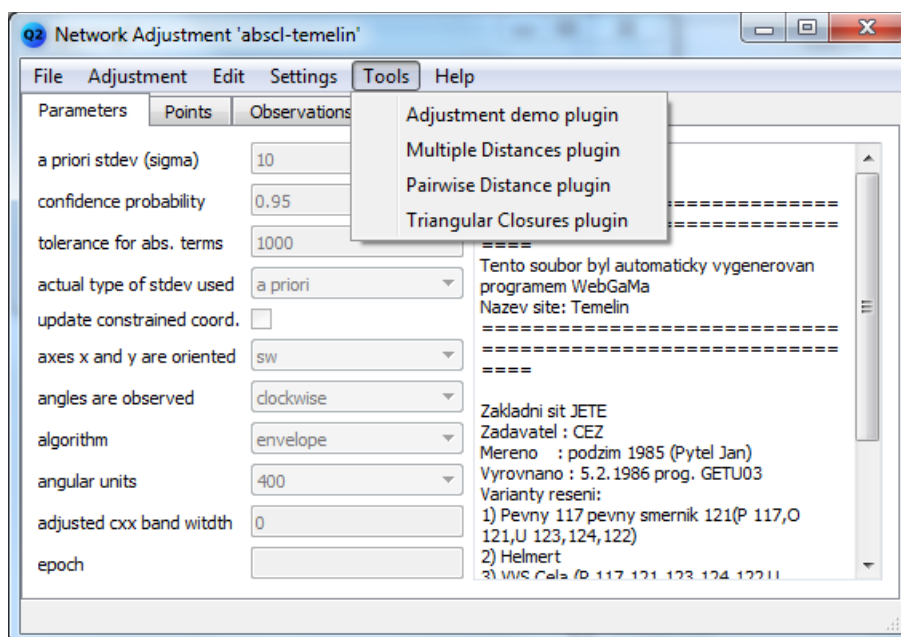
Obr. 8.2 Ukázka přetíženého signálu a jeho určení při použití lambda funkce.

Obecně tam, kde byly využity sloty z knihoven *Qt* a všude tam, kde by vlastní sloty využity více než jednou, byla použita varianta bez lambda funkcí. Všude tam, kde bylo použití jednorázové, byly využity lambda funkce.

Zdrojem informací pro tuto kapitolu byla nápověda programu *Qt Creator* [11].

9 Qt - pluginy

Plugin (nebo také plug-in) se někdy překládá jako zásuvný modul [2]. Je to vlastně program, který není schopen pracovat samostatně a je součástí jiného programu, který rozšiřuje o nový obsah.



Obr. 9.1 Pluginy v programu *gama-g2*.

V prostředí *Qt* jsou pluginy zařazeny do hlavního projektu jako samostatné části (*subprojects*).

```
lessThan(QT_MAJOR_VERSION, 5): error("requires Qt 5 or later")

TEMPLATE = subdirs
CONFIG += ordered
SUBDIRS = gama gama-q2

debug {
    SUBDIRS += gama-q2-plugins/adjustmentdemoplugin
    SUBDIRS += gama-q2-plugins/dbdemoplugin
    SUBDIRS += gama-q2-plugins/pairwisedistancesplugin
    SUBDIRS += gama-q2-plugins/multipledistancesplugin
    SUBDIRS += gama-q2-plugins/triangularclosuresplugin
}
```

Obr. 9.2 Schéma projektu *gama-g2* v prostředí *Qt* (*project file*).

10 Pluginy – obecně

Výsledkem této diplomové práce je několik pluginů. Tato kapitola obsahuje jejich popis.

10.1 Pairwise Distances Plugin

Tento plugin slouží k vylučování odlehlých dvojic měření. Pracuje s vodorovnými a šikmými délkami. Šikmé délky však musí být redukovány na spojnici stabilizačních značek. To však neprovádí tento plugin. Proto tedy do pluginu vstupují pouze ty šikmé délky, které mají nulovou výšku stroje a cíle. Dvojice jsou tvořeny ve všech kombinacích.

Mám-li mezi body A a B měřeno 10 vodorovných délek a například 6 šikmých délek redukovanych na spojnici stabilizačních značek. Výsledkem bude 45 dvojic pro vodorovné délky a 15 dvojic pro šikmé délky.

O tom zdali daná dvojice vyhovuje či nikoliv rozhoduje tzv. mezní rozdíl. Plugin umožňuje libovolnou délku vyskytující se v některé dvojici vyřadit.

10.2 Multiple Distances Plugin

Pokud máme více délek než pouze 2, můžeme testovat jejich opravy od průměru. Existuje několik statistických testů, které to umožňují vylučovat odlehlá měření. Tento plugin obsahuje testování opakovaných měření délek na základě Grubbsova testu. Uživatel by měl vědět, za jakých podmínek si může dovolit použít tento test – a to při vyšším počtu měření a za předpokladu, že má měření charakter normálního rozdělení.

Stejně jako u *Pairwise Distance Pluginu*, tak i tento plugin pracuje s vodorovnými a šikmými délkami, kde šikmé délky bere pouze takové, které jsou redukovány na spojnici stabilizačních značek.

Jeho sekundární funkce může být taková, že je zde přehled všech délek a je možné je zde opět vyřazovat. Zobrazují se zde i délky měřené jednou či dvakrát, ale není samozřejmě možné na ně aplikovat Grubbsův test.

10.3 Triangular Closures Plugin

Další plugin se týká trojúhelníkových uzávěrů a to bez ohledu na to zdali jsou měřeny přímo úhly nebo jsou úhly určeny jako rozdíl dvou směrů. Je zde zanedbán sférický exces, tudíž by nemělo být pracováno s trojúhelníky s příliš dlouhými stranami. To by však v inženýrské geodézii nemělo nastat.

Plugin umožňuje také vyřazovat jednotlivé směry či úhly. V hlavním okně pluginu jsou zobrazovány vždy úhly a to jak měřené tak vypočtené. Chceme-li vyřadit směr, pomocí něhož byl určen konkrétní úhel, zobrazí se nám další tabulka, kde můžeme vyřadit jakýkoliv z obou směrů.

11 Pluginy – z hlediska uživatele

V této kapitole je popsáno jak pracovat s jednotlivými pluginy.

11.1 Pairwise Distances Plugin

Tento plugin načte data (vodorovné délky či případně šikmé délky redukované na spojnici stabilizačních značek) při jeho samotném spuštění. Uživatel si poté může filtrovat řádky nebo sloupce dle svého uvážení. Další funkcí tohoto pluginu je řazení dle libovolného sloupce nebo případně podle poměru mezního rozdílu a absolutní hodnoty rozdílu. A třetí základní funkcí je možnost hromadně aktivovat nebo deaktivovat všechny načtené délky.

Máme-li toto měření:

Číslo měření	Stanovisko	Cíl	Délka	Sm. od.
			[m]	[mm]
1	1	2	50,005	10
2	1	2	50,025	10
3	2	1	50,000	10
4	2	1	50,005	10

Tab. 11.1 Příklad (kritická hodnota je rovna hodnotě 0,05).

Pomocí tohoto pluginu zjistíme, že 4 měření nám utvoří 6 dvojic. Z toho u dvou dvojic je překročen mezní rozdíl.

	A	B	Value [m]			Value [m]			I - II [mm]	ΔM [mm]
1	1	2	50.0000	A-B	<input checked="" type="checkbox"/>	50.0300	A-B	<input checked="" type="checkbox"/>	-30.0	27.7
2	1	2	50.0300	A-B	<input checked="" type="checkbox"/>	50.0020	B-A	<input checked="" type="checkbox"/>	28.0	27.7
3	1	2	50.0300	A-B	<input checked="" type="checkbox"/>	50.0050	B-A	<input checked="" type="checkbox"/>	25.0	27.7
4	1	2	50.0000	A-B	<input checked="" type="checkbox"/>	50.0050	B-A	<input checked="" type="checkbox"/>	-5.0	27.7
5	1	2	50.0020	B-A	<input checked="" type="checkbox"/>	50.0050	B-A	<input checked="" type="checkbox"/>	-3.0	27.7
6	1	2	50.0000	A-B	<input checked="" type="checkbox"/>	50.0020	B-A	<input checked="" type="checkbox"/>	-2.0	27.7

Obr. 11.1 Příklad (*Pairwise Distances Plugin*).

Obě inkriminované dvojice obsahují měření číslo 2 (50,025 m), tudíž je možné usoudit, že toto měření je odlehlé a je třeba jej vyloučit.

11.2 Multiple Distances Plugin

Tento plugin je obdobou předchozího. Opět je zde možné filtrovat řádky či sloupce. Sloupce lze také řadit. Tento plugin neobsahuje možnost hromadné aktivace či deaktivace

všech načtených měření na rozdíl od předchozího pluginu. Kolik měření je aktivních je vidět ve sloupci počet (*Count*). Vyřazovat jednotlivá měření je možné po vyvolání další tabulky po stisknutí tlačítka v posledním sloupci tabulky. Vyřazovat lze i měření, jejichž počet není vhodný pro Grubbsův test. V této tabulce se vše přepočítá automaticky, takže není potřeba žádné tlačítko.

Použijeme-li příklad z Tab. 4.1, zjistíme, podle Grubbsova testu je měření číslo 2 odlehlé:

	Type	A	B	Count	Count A-B	Count B-A	Mean [m]	Test
1	horizontal	1	2	4 / 4	2	2	50.0093	75 %

Test '1-2'						
	From	To	Value [m]	StdDev [m]	Outliers [m]	Active
1	1	2	50.0000	0.0100	0.0093	<input checked="" type="checkbox"/>
2	1	2	50.0300	0.0100	-0.0207	<input checked="" type="checkbox"/>
3	2	1	50.0020	0.0100	0.0072	<input checked="" type="checkbox"/>
4	2	1	50.0050	0.0100	0.0042	<input checked="" type="checkbox"/>

Obr. 11.2 Příklad (*Multiple Distances Plugin*).

Tento příklad však není příliš vhodný pro Grubbsův test, jelikož je počet měření příliš nízký. Po odstranění nejodlehlejší hodnoty, podle něho měření stále obsahuje odlehlé hodnoty.

	Type	A	B	Count	Count A-B	Count B-A	Mean [m]	Test
1	horizontal	1	2	3 / 4	1	2	50.0023	67 %

Test '1-2'						
	From	To	Value [m]	StdDev [m]	Outliers [m]	Active
1	1	2	50.0000	0.0100	0.0023	<input checked="" type="checkbox"/>
2	1	2	50.0300	0.0100	-0.0277	<input type="checkbox"/>
3	2	1	50.0020	0.0100	0.0003	<input checked="" type="checkbox"/>
4	2	1	50.0050	0.0100	-0.0027	<input checked="" type="checkbox"/>

Obr. 11.3 Příklad (*Multiple Distances Plugin*).

Vhodnějším příkladem pro použití Grubbova testu je tento:

Číslo měření	Stanovisko	Cíl	Délka	Sm. od.
			[m]	[mm]
1	1	2	50,005	10
2	1	2	50,004	10
3	1	2	50,003	10
4	1	2	50,005	10
5	1	2	50,002	10
6	1	2	50,003	10
7	2	1	50,002	10
8	2	1	50,005	10
9	2	1	50,004	10
10	2	1	50,004	10
11	2	1	50,003	10
12	2	1	50,005	10

Tab. 11.2 Příklad (kritická hodnota je rovna hodnotě 0,05).

Plugin nejprve detekoval jedno odlehlé měření:

Type	A	B	Count	Count A-B	Count B-A	Mean [m]	Test
1 horizontal	1	2	12 / 12	6	6	50.0046	92 %

Test '1-2'						
	From	To	Value [m]	StdDev [m]	Outliers [m]	Active
1	1	2	50.0050	0.0100	-0.0004	<input checked="" type="checkbox"/>
2	1	2	50.0040	0.0100	0.0006	<input checked="" type="checkbox"/>
3	1	2	50.0030	0.0100	0.0016	<input checked="" type="checkbox"/>
4	1	2	50.0150	0.0100	-0.0104	<input checked="" type="checkbox"/>
5	1	2	50.0020	0.0100	0.0026	<input checked="" type="checkbox"/>
6	1	2	50.0030	0.0100	0.0016	<input checked="" type="checkbox"/>
7	2	1	50.0020	0.0100	0.0026	<input checked="" type="checkbox"/>
8	2	1	50.0050	0.0100	-0.0004	<input checked="" type="checkbox"/>
9	2	1	50.0040	0.0100	0.0006	<input checked="" type="checkbox"/>
10	2	1	50.0040	0.0100	0.0006	<input checked="" type="checkbox"/>
11	2	1	50.0030	0.0100	0.0016	<input checked="" type="checkbox"/>
12	2	1	50.0050	0.0100	-0.0004	<input checked="" type="checkbox"/>

Obr. 11.4 Příklad (*Multiple Distances Plugin*).

A po jeho vyloučení již soubor neobsahoval, žádná odlehlá měření:

Type	A	B	Count	Count A→B	Count B→A	Mean [m]	Test
1 horizontal	1	2	11 / 12	5	6	50.0036	100 %

	From	To	Value [m]	StdDev [m]	Outliers [m]	Active
1	1	2	50.0050	0.0100	-0.0014	<input checked="" type="checkbox"/>
2	1	2	50.0040	0.0100	-0.0004	<input checked="" type="checkbox"/>
3	1	2	50.0030	0.0100	0.0006	<input checked="" type="checkbox"/>
4	1	2	50.0150	0.0100	-0.0114	<input type="checkbox"/>
5	1	2	50.0020	0.0100	0.0016	<input checked="" type="checkbox"/>
6	1	2	50.0030	0.0100	0.0006	<input checked="" type="checkbox"/>
7	2	1	50.0020	0.0100	0.0016	<input checked="" type="checkbox"/>
8	2	1	50.0050	0.0100	-0.0014	<input checked="" type="checkbox"/>
9	2	1	50.0040	0.0100	-0.0004	<input checked="" type="checkbox"/>
10	2	1	50.0040	0.0100	-0.0004	<input checked="" type="checkbox"/>
11	2	1	50.0030	0.0100	0.0006	<input checked="" type="checkbox"/>
12	2	1	50.0050	0.0100	-0.0014	<input checked="" type="checkbox"/>

Obr. 11.5 Příklad (*Multiple Distances Plugin*).

Výsledkem tedy je, že bylo použito 5 měření z bodu 1 na bod 2 a 6 měření směrem opačným. Průměr z těchto 11 měření poté činil 50,0036 m.

11.3 Triangular Closures Plugin

U tohoto pluginu jsou použity vesměs podobné nebo naprosto totožné funkce jako u *Pairwise Distances Pluginu*. Řádky i sloupce lze také filtrovat. Jednotlivé sloupce je možné řadit a zase se zde vyskytuje funkce pro řazení dle poměru mezního úhlového uzávěru a vlastního úhlového uzávěru. Stejně jako u *Pairwise Distances Pluginu* je tu také funkce pro hromadnou aktivaci a deaktivaci všech načtených měření.

Jak plugin funguje, si můžeme ukázat na tomto příkladu:

Číslo vrcholu	Úhel	Sm. od.
	[gon]	[cc]
1	70,0005	5
2	80,0004	5
3	50,0010	5

Tab. 11.3 Příklad (kritická hodnota je rovna hodnotě 0,05).

Máme úhly v trojúhelníku určené buď z rozdílů směrů, nebo jsou měřené přímo.

	A	B	C	Angle [gon]		Angle [gon]		Angle [gon]		Closure [cc]	ΔU [cc]
1	1	2	3	70.00050	<input checked="" type="checkbox"/>	80.00040	<input checked="" type="checkbox"/>	50.00100	<input checked="" type="checkbox"/>	-19.0	17.0

Obr. 11.6 Příklad (*Triangular Closures Plugin*).

V tomto případě byl překročen mezní úhlový uzávěr. Chceme-li vyřadit úhel určený pomocí dvojicí směrů, dále je zobrazena takováto tabulka:

	A	B	C	Angle [gon]		Angle [gon]		Angle [gon]		Closure [cc]	ΔU [cc]
1	1	2	3	70.00050	<input checked="" type="checkbox"/>	80.00040	<input checked="" type="checkbox"/>	50.00100	<input checked="" type="checkbox"/>	-19.0	17.0

Observation	Target	Value [gon]	Use	
1	1	2	0.00000	<input checked="" type="checkbox"/>
2	1	3	70.00050	<input checked="" type="checkbox"/>

Obr. 11.7 Příklad (*Triangular Closures Plugin*).

Pokud chceme vyřadit úhel, který byl přímo měřen, žádná tabulka samozřejmě není potřeba. Po vyřazení jednoho úhlu (resp. směru) může tabulka vypadat například takto:

	A	B	C	Angle [gon]		Angle [gon]		Angle [gon]		Closure [cc]	ΔU [cc]
1	1	2	3	70.00050	<input type="checkbox"/>	80.00040	<input checked="" type="checkbox"/>	50.00100	<input checked="" type="checkbox"/>		

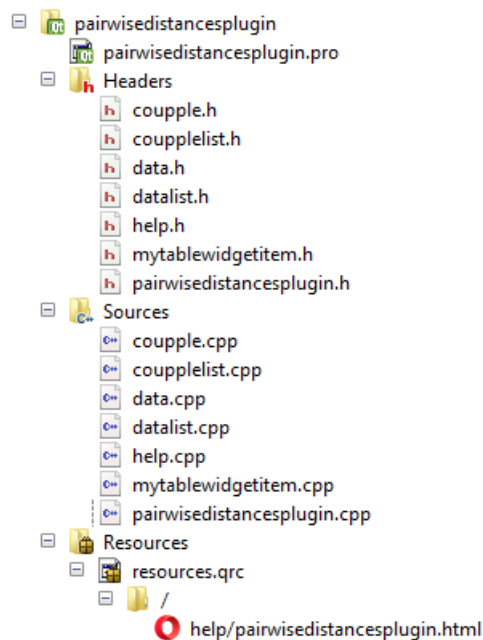
Obr. 11.8 Příklad (*Triangular Closures Plugin*).

12 Pluginy – z hlediska programování

Tato kapitola obsahuje popis jednotlivých tříd pro všechny pluginy.

12.1 Pairwise Distances Plugin

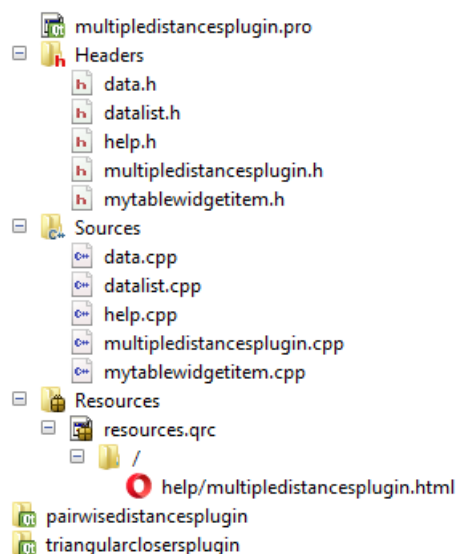
Třída, ze které se spouští plugin nese název pluginu (*PairwiseDistancesPlugin*). Na ní jsou přímo navázány třídy: *MyTableWidgetItem*, *Help* a *CoupleList*. *MyTableWidgetItem* je třída odvozená od *QTableWidgetItem* a obsahuje pouze jeden operátor pro řazení dat v tabulce. Třída *Help* obsahuje pouze privátní konstruktor a veřejnou metodu, která jej volá. Stejným způsobem je vytvořena i třída *GamaQ2Help* v hlavním ovládacím panelu (*MainControlPanel*). Třída *CoupleList*, jak již název napovídá, obsahuje seznam všech dvojic délek. To znamená, že je tato třída navázána na třídu *Couple*, která obsahuje pouze jeden konkrétní pár. Na třídu *CoupleList* je dále ještě navázána třída *DataList*, do které jsou ukládány jednotlivá data. V této třídě jsou vždy data přiřazena k jednotlivým délkám data. Třída *Data* představuje jednu konkrétní délku.



Obr. 12.1 Struktura pluginu *Pairwise Distances Plugin*.

12.2 Multiple Distances Plugin

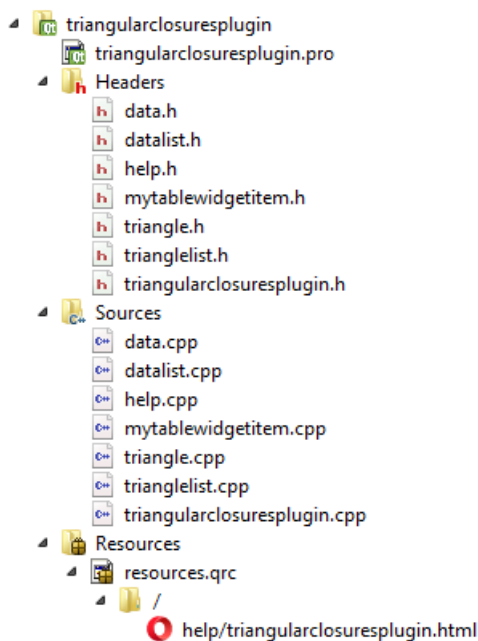
Třída *MultipleDistancesPlugin* je opět třída, ze které je spouštěno hlavní okno. Stejně jako u *PairwiseDistancesPluginu* je tato třída navázána na třídy *MyTableWidgetItem* a *Help*, které jsou téměř totožné. U tohoto pluginu je třída *MultipleDistancesPlugin* přímo navázána na třídu *DataList*, které je naprosto totožná jako u pluginu pro dvojice délek. Stejně tak je na tuto třídu navázána třída *Data*.



Obr. 12.2 Struktura pluginu *Multiple Distances Plugin*.

12.3 Triangular Closures Plugin

Hlavní třídou je v tomto případě *TriangularClosuresPlugin*. Na ní jsou navázány třídy *MyTableWidgetItem*, *Help* a *TriangleList*. Třída *TriangleList* slouží k ukládání trojúhelníků a dále obsahuje metodu na vyhledání trojúhelníků ze seznamu dat *DataList*. Do třídy *Triangle* jsou ukládány ukazatele na jednotlivá měření úhlů či směrů. Ve třídě *Data* jsou k jednotlivým stanoviskům přiřazována jednotlivá měření.



Obr. 12.3 Struktura pluginu *Triangular Closures Plugin*.

12.4 Použité kontejnery

V pluginech byly použity různé druhy kontejnerů. V této části jsou vybrané popsány. Ze standardních kontejnerů byly použity tyto kontejnery:

1. *std::pair*

Tento kontejner slouží k ukládání dvojic.

2. *std::vector/std::list*

Tyto kontejnery patří mezi sekvenční kontejnery. *std::vector* se doporučuje používat jako implicitní. *std::list* list by měl být použit tam, kde jsou často odebírány prvky uprostřed kontejneru.

3. *std::map/std::multimap*

Tyto kontejnery patří mezi asociativní. To že k prvkům (*value*) znamená, že je k nim přistupováno pomocí klíčů (*key*). U *std::map* je klíč unikátní a naopak *std::multimap* může obsahovat duplicitní hodnoty.

4. *std::set/std::multiset*

Opět se jedná o asociativní kontejnery se stejnými vlastnostmi jako u *std::map* resp. *std::multimap*. Jediným rozdílem je to, že neobsahuje prvky (*value*), ale pouze klíče (*key*).

Dále bylo použito i několik kontejnerů z knihoven *Qt*:

1. *QVector*

Podobné jako *std::vector*.

2. *QList*

Podobné jako *std::list*.

3. *QStringList*

Podobné jako *QList*, ale jen pro datový typ *QString*.

12.5 Vyhledání měřených délek

Vlastní vyhledání se skládá ze tří kroků.

1. Vytvoření seznamu dvojic (bez opakování) s použitím kontejneru:

```
std::set<std::pair<std::string, std::string>>
```

2. Vytvoření seznamu všech délek s použitím kontejneru:

```
std::multimap<std::pair<std::string, std::string>, Observation*>
```

3. Přiřazení jednotlivých délek (pomocí dvou cyklů) k seznamu dvojic, kde je pro jednotlivé dvojice použit kontejner: *QList<Observation*>*

(První dva kroky byly provedeny současně.)

12.6 Vyhledání dvojic měřených délek

Vyhledání dvojic délek proběhlo pomocí dvou cyklů, přičemž bylo nutné zvlášť vyhledat dvojice vodorovných a zvlášť dvojice šikmých délek (redukovaných na spojnicí stabilizačních značek).

```
for(int j = 0; j < data[i]->size(); j++){  
    for(int k = j+1; k < data[i]->size(); k++){  
        Couple* couple = new Couple(data[i]->a(), data[i]->b(), true,  
                                     data[i]->observation(j), data[i]->observation(k), _lnet);  
        addData(couple);  
    }  
}
```

Obr. 12.4 Vyhledání dvojic měřených délek.

12.7 Vyhledání úhlových měření

Vyhledání se skládá v tomto případě ze dvou kroků:

1. Vyhledání stanovisek (bez opakování) s použitím kontejneru:

```
std::set<std::string>
```

2. Vyhledání cílů a jejich přidání do kontejneru příslušného stanoviska:

```
std::set<std::string>
```

12.8 Vyhledání trojice úhlů v trojúhelníku

Trojice jednotlivých měření, které tvoří trojúhelníky, byly nalezeny pomocí třech cyklů:

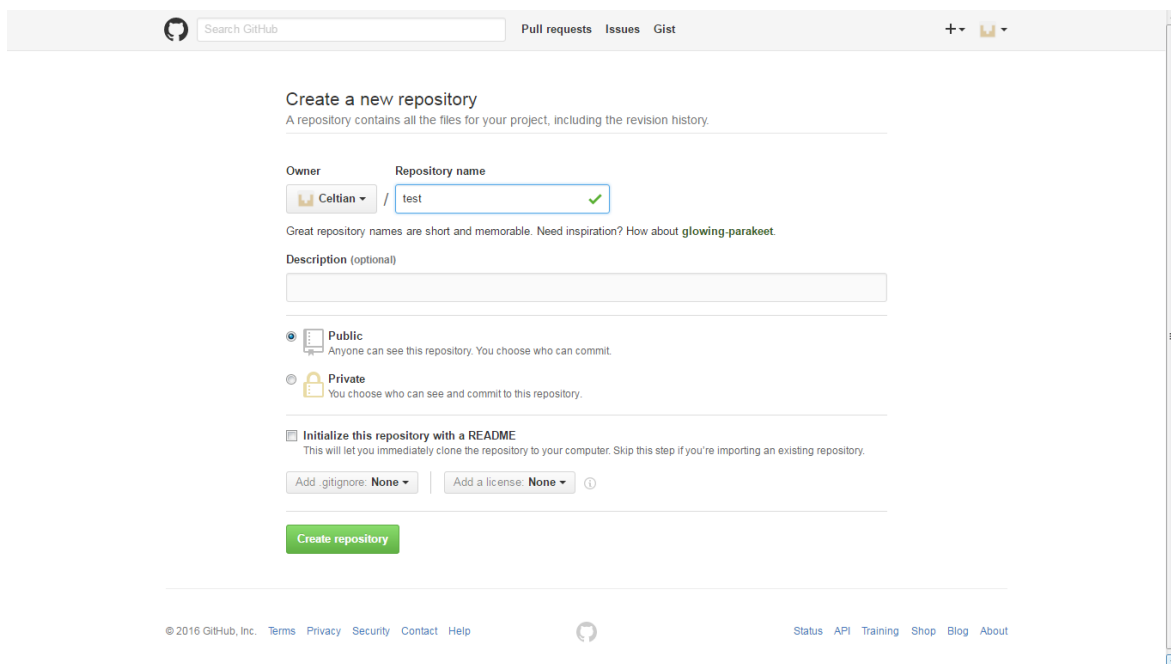
```
int N = dataList->count();
for (int i=0; i<N-2; i++) {
    Data A = *(dataList->operator [] (i));
    for (int j=i+1; j<N-1; j++) {
        Data B = *(dataList->operator [] (j));
        for (int k=j+1; k<N; k++) {
            Data C = *(dataList->operator [] (k));
            if (A.pid() == B.pid() || A.pid() == C.pid() || B.pid() == C.pid())
                continue;
        }
        if(A.targets().find(B.pid()) == A.targets().end() ||
            A.targets().find(C.pid()) == A.targets().end() ||
            B.targets().find(A.pid()) == B.targets().end() ||
            B.targets().find(C.pid()) == B.targets().end() ||
            C.targets().find(A.pid()) == C.targets().end() ||
            C.targets().find(B.pid()) == C.targets().end()) {
                continue;
            }
        addData(new Triangle(A.pid(),B.pid(),C.pid(),_lnet));
    }
}
}
```

Obr. 12.5 Vyhledání trojúhelníků.

Poté bylo však nutno zjistit, jsou směry v takových trojúhelnících obousměrné a pokud ne, tak je třeba takový trojúhelník vyřadit.

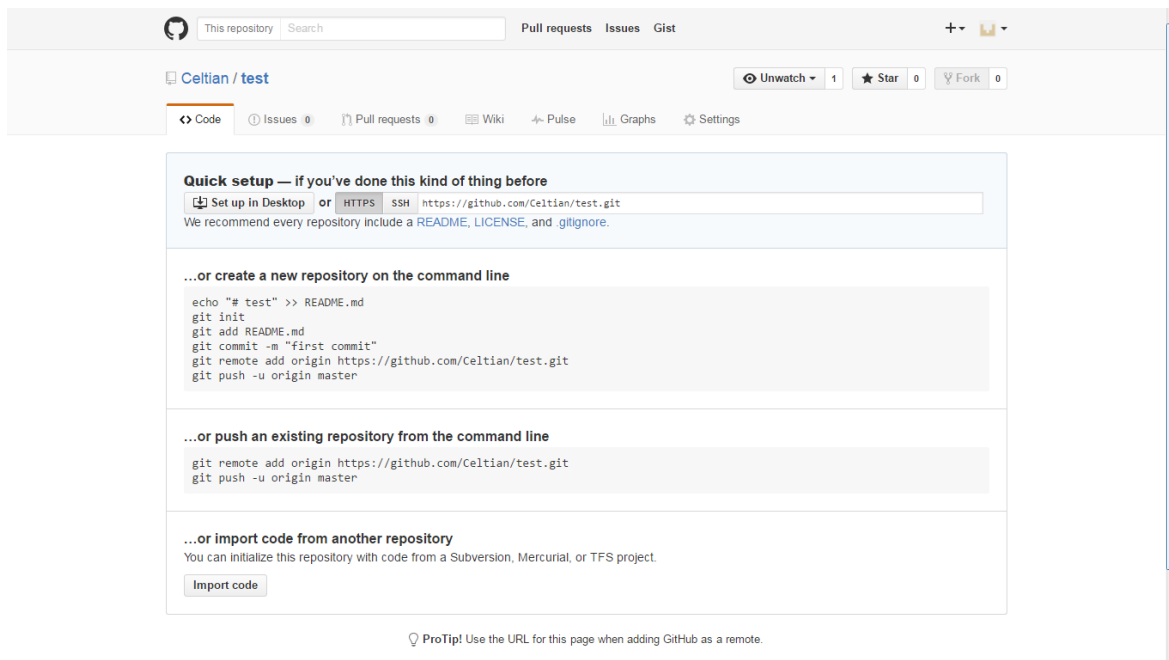
13 Nahrávání souborů na GitHub

Jedním úkolů v rámci této diplomové práce bylo nahrání hotových pluginů na *GitHub*. Nejprve je nutné se zaregistrovat a stáhnout si *GitHub* [6]. Poté je třeba se přihlásit k účtu a tam vytvořit adresář (*repository*):



Obr. 13.1 Založení nového adresáře ve webovém prostředí *GitHubu* [6].

Když potvrdíme jeho vytvoření, zobrazí se nám stránka, na které jsou již připravené některé základní příkazy, které je potřeba vykonat pro nahrání souborů:



Obr. 13.2 Obsah Adresáře na *GitHubu* po jeho úspěšném vytvoření [6].

Význam základních příkazů je následující:

1. *git init* – inicializace prázdného adresáře
2. *git add* – přidávání souborů
3. *git commit* – potvrzení provedených změn
4. *git remote add* – přidání vzdáleného adresáře ve službě *GitHub*
5. *git push* – nahrání souborů do vzdáleného adresáře

Příkazy, které tam jsou připraveny, je možné použít v příkazovém řádku *GitHubu* (*GitHub Shell*):

```

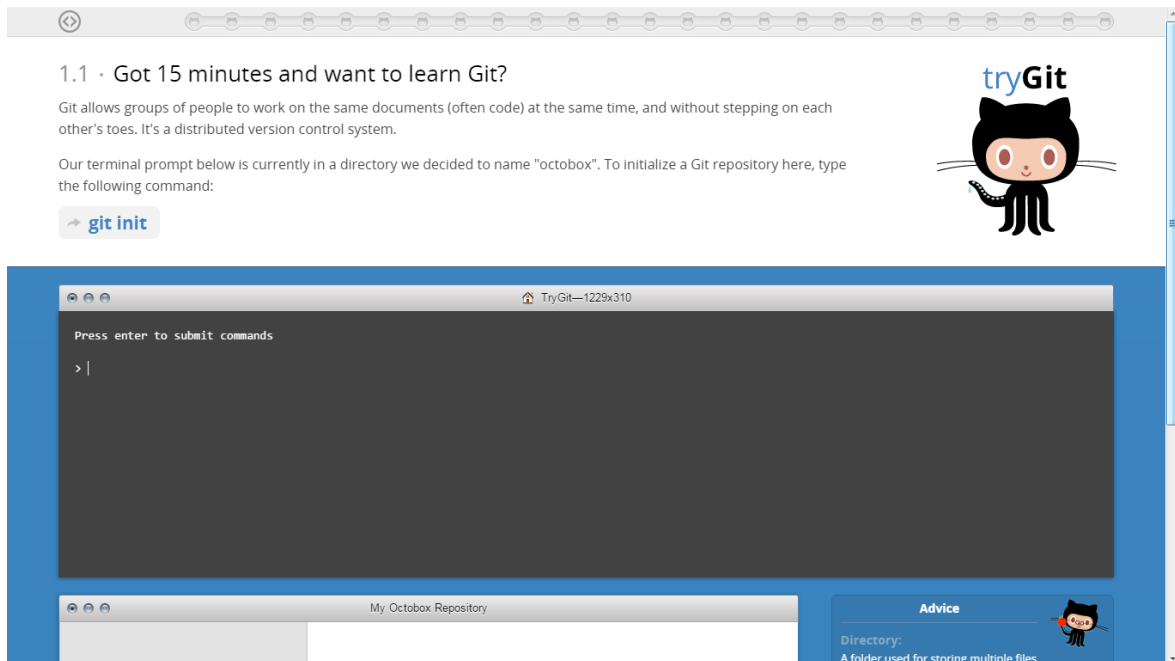
posh-git ~ multipledistancesplugin [master]
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Celtian\Documents\GitHub [master] > cd ..\..
C:\Users\Celtian > cd .\Desktop\aaaa\multipledistancesplugin
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin > git init
Initialized empty Git repository in C:/Users/Celtian/Desktop/aaaa/multipledistan
cesplugin/.git/
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin [master +14 ^0 -0 !] > git
add
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin [master +14 ^0 -0] > git co
mmit -m "first commit"
[master (root-commit) 518210a] first commit
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
14 files changed, 1668 insertions(+)
 create mode 100644 README.md
 create mode 100644 data.cpp
 create mode 100644 data.h
 create mode 100644 datalist.cpp
 create mode 100644 datalist.h
 create mode 100644 help.cpp
 create mode 100644 help.h
 create mode 100644 help/multipledistancesplugin.html
 create mode 100644 multipledistancesplugin.cpp
 create mode 100644 multipledistancesplugin.h
 create mode 100644 multipledistancesplugin.pro
 create mode 100644 mytablewidgetitem.cpp
 create mode 100644 mytablewidgetitem.h
 create mode 100644 resources.qrc
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin [master] > git remote add o
rigin https://github.com/Celtian/multipledistancesplugin.git
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin [master] > git push -u orig
in master
Counting objects: 17, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (17/17) done.
Writing objects: 100% (17/17), 14.71 KiB | 0 bytes/s, done.
Total 17 (delta 5), reused 0 (delta 0)
To https://github.com/Celtian/multipledistancesplugin.git
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.
C:\Users\Celtian\Desktop\aaaa\multipledistancesplugin [master] > _

```

Obr. 13.3 Nahrání souborů na *GitHub* v prostředí *GitHub Shell*.

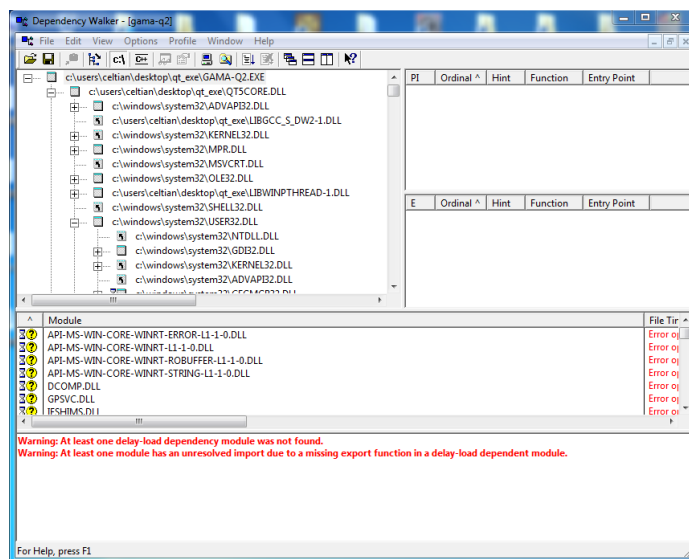
Pro pochopení jednotlivých příkladů je dobré si projít interaktivní tutoriál na [12]:



Obr. 13.4 Interaktivní tutoriál pro příkazy pro *Git* [12].

14 Vytvoření spustitelné aplikace

Pro vytvoření spustitelné aplikace bylo nutné změnit ladění (*Build*) na hodnotu *Release* v programu *Qt Creator*. Potom bylo potřeba zjistit, z jakých knihoven (*.dll) se program skládá. K tomu byl použit program *Dependency Walker* [13].



Obr. 14.1 *Dependency Walker* [13].

Většina knihoven bývá umístěna v adresáři, kde je nainstalována platforma *Qt*. Knihovny lze také nalézt na webu [14].

Nalezené knihovny je poté třeba nakopírovat do jednoho adresáře, kde je nutné také přidat spustitelnou aplikaci (*.exe). Pokud jsou knihovny zkopírovány ze složky *plugins* (pluginy) z adresářů platformy *Qt*, je nutné dbát na to, aby byly zkopírovány celé složky.

Závěr

Pro projekt GNU Gama byly vytvořeny tři pluginy. Dva se zabývají analýzou vodorovných či šikmých délek redukovaných na spojnici stabilizačních značek.

Pro analýzu dvojic délkových měření byl vytvořen *Pairwise Distances Plugin*. Ten vyhledá veškeré dvojice délek a porovná jejich rozdíl s mezním rozdílem. Výstupem je tedy informace o tom, zdali soubor obsahuje nějakou dvojici, kde je překročen mezní rozdíl.

Máme-li vyšší počet opakovaných měření délek, je možné na ně aplikovat Grubbsův oboustranný test v pluginu nazvaném *Multiple Distances Plugin*.

Chceme-li zjistit, zdali v nějakém trojúhelníku v geodetické síti byl překročen mezní úhlový uzávěr, k tomu byl vytvořen *Triangular Closures Plugin*.

Pluginy byly vytvořeny v programu *Qt Creator*, kde nebylo využito grafické rozhraní. Kód byl psán v jazyce *c++* ve standardu *c++11*.

Zdrojové kódy, příklady a spustitelná aplikace programu *gama-g2* byly umístěny do adresáře na webové službě *GitHub*, se kterou bylo nutné se nejprve seznámit.

Pro vytvoření spustitelné aplikace programu *gama-g2*, byl použit program *Dependency Walker*.

Seznam proměnných

Δ_M – mezní rozdíl

u_p – koeficient spolehlivosti

σ_Δ - směrodatná odchylka rozdílu dvojice měření

σ_1 - směrodatná odchylka prvního měření

σ_2 - směrodatná odchylka druhého měření

σ - směrodatná odchylka

U_M - mezní úhlový uzávěr

σ_U – směrodatná odchylka úhlového uzávěru

$\sigma_{\omega_1}, \sigma_{\omega_2}, \sigma_{\omega_3}$ – směrodatné odchylky jednotlivých úhlů v trojúhelníku

σ_ω – směrodatná odchylka měřeného úhlu

$\sigma_{\varphi_1}, \sigma_{\varphi_2}, \sigma_{\varphi_3}, \sigma_{\varphi_4}, \sigma_{\varphi_5}, \sigma_{\varphi_6}$ – směrodatné odchylky jednotlivých směrů v trojúhelníku

σ_φ – směrodatná odchylka měřeného směru

m_v - střední oprava

v – oprava od aritmetického průměru

n – počet měření

$K_{G(\alpha,n)}$ – kritická hodnota Grubbsova oboustranného testu

$t(\alpha, n)$ – kritická hodnota Studentova rozdělení

v_{max} – maximální oprava od aritmetického průměru

K_1 – testovací kritérium Grubbsova oboustranného testu pro nejodlehlejší měření

$K_{1(\alpha,n)}$ – kritická hodnota Grubbsova oboustranného testu pro nejodlehlejší měření

μ – střední hodnota

t – veličina

$\varphi(t)$ – hustota pravděpodobnosti normovaného normálního rozdělení

Seznam zkratk

GNU – grafické uživatelské rozhraní

OS – operační systém

Seznam obrázků

Obr. 1.1 Ukázka geodetické sítě [1].	5
Obr. 2.1 Webová stránka, na které lze stáhnout program <i>Git Bash</i> [5].	6
Obr. 2.2 Ukázka softwaru <i>Git Bash</i> .	6
Obr. 2.3 Webová stránka služby <i>GitHub</i> [6].	7
Obr. 2.4 Ukázka softwaru <i>GitHub Shell</i> .	7
Obr. 3.1 Hlavní okno programu <i>gama-g2</i> .	8
Obr. 3.2 Dialogové okno pro připojení databáze pro <i>SQLite</i> databázi.	9
Obr. 3.3 Dialogové okno pro připojení databáze pro <i>MySQL</i> databázi.	9
Obr. 3.4 Grafické uživatelské rozhraní programu <i>Measurements2Gama</i> [8].	10
Obr. 3.5 Dialog pro výběr geodetické sítě z databáze.	11
Obr. 3.6 Dialogové okno, kde je možné změnit jazyk výstupních souborů.	11
Obr. 3.7 Panel pro vyrovnání sítí (<i>Tabbed Windows</i>).	12
Obr. 3.8 Nastavení zobrazení.	13
Obr. 5.1 Webová stránka platformy <i>Qt</i> [3].	18
Obr. 5.2 Program <i>Qt Creator</i> .	19
Obr. 7.1 Různé druhy aplikací v prostředí <i>Qt</i> .	22
Obr. 7.2 Ukázka toho, jak vytvořit a zobrazit dialog.	23
Obr. 8.1 Signály a sloty bez a s využitím lambda funkce.	24
Obr. 8.2 Ukázka přetíženého signálu a jeho určení při použití lambda funkce.	24
Obr. 9.1 Pluginy v programu <i>gama-g2</i> .	25
Obr. 9.2 Schéma projektu <i>gama-g2</i> v prostředí <i>Qt</i> (<i>project file</i>).	25
Obr. 11.1 Příklad (<i>Pairwise Distances Plugin</i>).	27
Obr. 11.2 Příklad (<i>Multiple Distances Plugin</i>).	28
Obr. 11.3 Příklad (<i>Multiple Distances Plugin</i>).	28
Obr. 11.4 Příklad (<i>Multiple Distances Plugin</i>).	29
Obr. 11.5 Příklad (<i>Multiple Distances Plugin</i>).	30
Obr. 11.6 Příklad (<i>Triangular Closures Plugin</i>).	31
Obr. 11.7 Příklad (<i>Triangular Closures Plugin</i>).	31
Obr. 11.8 Příklad (<i>Triangular Closures Plugin</i>).	31
Obr. 12.1 Struktura pluginu <i>Pairwise Distances Plugin</i> .	32

Obr. 12.2 Struktura pluginu <i>Multiple Distances Plugin</i>	32
Obr. 12.3 Struktura pluginu <i>Triangular Closures Plugin</i>	33
Obr. 12.4 Vyhledání dvojic měřených délek.	34
Obr. 12.5 Vyhledání trojúhelníků.....	35
Obr. 13.1 Založení nového adresáře ve webovém prostředí <i>GitHubu</i> [6].....	36
Obr. 13.2 Obsah Adresáře na <i>GitHubu</i> po jeho úspěšném vytvoření [6].....	37
Obr. 13.3 Nahrání souborů na <i>GitHub</i> v prostředí <i>GitHub Shell</i>	38
Obr. 13.4 Interaktivní tutoriál pro příkazy pro <i>Git</i> [12].	38
Obr. 14.1 <i>Dependency Walker</i> [13].....	39

Seznam tabulek

Tab. 4.1 Volba koeficientu spolehlivosti pro jednorozměrné veličiny [9].	16
Tab. 6.1 Operátor pro řazení záznamů v tabulce.....	20
Tab. 7.1 Modalita jednotlivých oken a její význam [11].....	23
Tab. 11.1 Příklad (kritická hodnota je rovna hodnotě 0,05).....	27
Tab. 11.2 Příklad (kritická hodnota je rovna hodnotě 0,05).....	29
Tab. 11.3 Příklad (kritická hodnota je rovna hodnotě 0,05).....	30

Seznam grafů

Graf 4.1 Normální rozdělení - $N(\mu = 0, \sigma = 1)$ [9].	16
--	----

Seznam příloh

Veškeré výsledky a instrukce jsou k dispozici na adrese: <https://github.com/Celtian>.
Z tohoto adresáře jsou součástí této diplomové práce následující části:

1. Pairwise Distances Plugin (<https://github.com/Celtian/pairwisedistancesplugin>)
2. Multiple Distances Plugin (<https://github.com/Celtian/multipledistancesplugin>)
3. Triangular Closures Plugin (<https://github.com/Celtian/triangularclosuresplugin>)
4. Plugin Example Data (<https://github.com/Celtian/pluginexampledata>)
5. Spustitelná aplikace programu *gama-g2* (+ příklady)
(<https://github.com/Celtian/gama-exe>)

Seznam zdrojů

- [1] *GNU Gama* [online]. [cit. 2016-05-16]. Dostupné z: <http://www.gnu.org/software/gama/>
- [2] Plugin. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-05-16]. Dostupné z: <https://cs.wikipedia.org/wiki/Plugin>
- [3] *Qt* [online]. [cit. 2016-05-16]. Dostupné z: <https://www.qt.io/download/>
- [4] Git. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-05-16]. Dostupné z: <https://cs.wikipedia.org/wiki/Git>
- [5] *Git SCM* [online]. [cit. 2016-05-16]. Dostupné z: <https://git-scm.com/downloads>
- [6] *GitHub* [online]. [cit. 2016-05-16]. Dostupné z: <https://github.com>
- [7] GitHub. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-05-16]. Dostupné z: <https://en.wikipedia.org/wiki/GitHub>
- [8] Measurements2Gama. *Celtian* [online]. Praha, 2016 [cit. 2016-05-16]. Dostupné z: <http://www.celtian.wz.cz/measurements2gama.html>
- [9] HAMPACHER, Miroslav a Martin ŠTRONER. *Zpracování a analýza měření v inženýrské geodézii*. Vydání druhé, upravené a doplněné. Praha: Česká technika - nakladatelství ČVUT (CTN), Zikova 4, Praha 6, 2015. ISBN 978-80-01-05843.
- [10] Kaskádové styly. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-05-16]. Dostupné z: https://cs.wikipedia.org/wiki/Kaskádové_styly
- [11] *Qt Documentation* [online]. [cit. 2016-05-16]. Dostupné z: <http://doc.qt.io>
- [12] *TryGitHub* [online]. Praha, 2016 [cit. 2016-05-16]. Dostupné z: <https://try.github.io>
- [13] *Dependency Walker* [online]. [cit. 2016-05-16]. Dostupné z: <http://www.dependencywalker.com>
- [14] *dll-files.com* [online]. [cit. 2016-05-16]. Dostupné z: <http://www.dll-files.com>
- [15] HLADÍK, Dominik. *Software pro zpracování polárně zaměřených dat* [online]. 2014 [cit. 2016-05-18]. Dostupné z: <http://geo.fsv.cvut.cz/proj/bp/2014/dominik-hladik-bp-2014.pdf>. ČVUT v Praze. Vedoucí práce Prof. Ing. Aleš Čepek, CSc.
- [16] ČEPEK, Aleš. *Informatika: Úvod do c++*. Vydání první. Praha: Vydavatelství ČVUT - výroba, Zikova 4, 166 36 Praha 6, 2004. ISBN 80-01-03074.
- [17] GRUBBS, Frank E. *Sample Criteria for Testing Outlying Observations*. Ann. Math. Statist. 21 (1950), no. 1, 27--58. doi:10.1214/aoms/1177729885. <http://projecteuclid.org/euclid.aoms/1177729885>.