

Obsah

1	Úvod	2
2	Tvorba mapových výstupů	2
2.1	Porovnání možností programů pro tvorbu mapových výstupů	2
2.1.1	ArcGIS	2
2.1.2	QGis	2
2.1.3	gvSIG	2
2.1.4	GRASS GIS – ps.map	2
3	Zásady tvorby grafického uživatelského rozhraní	2
3.1	Čeho se vyvarovat	2
3.2	Co je vhodné	2
4	Systém GRASS GIS	2
4.1	Grafické rozhraní systému GRASS GIS	2
4.2	Modul ps.map	2
4.2.1	Konfigurační soubor	3
4.2.2	Volba rozsahu zobrazovaného území	3
4.2.3	Nekonzistence a chyby v modulu ps.map	4
5	Grafické rozhraní pro modul <i>ps.map</i>	7
5.1	Stažení programu z repozitáře add-ons	7
5.2	Podklady a použité knihovny	7
5.3	Funkcionalita aplikace	7
5.4	Základní práce s GUI aplikací	8
5.5	Problémy při tvorbě GUI	8
5.6	Ukázky	10
6	Závěr	10

1 Úvod

2 Tvorba mapových výstupů

2.1 Porovnání možností programů pro tvorbu mapových výstupů

2.1.1 ArcGIS

2.1.2 QGis

2.1.3 gvSIG

2.1.4 GRASS GIS – ps.map

3 Zásady tvorby grafického uživatelského rozhraní

3.1 Čeho se vyvarovat

3.2 Co je vhodné

4 Systém GRASS GIS

4.1 Grafické rozhraní systému GRASS GIS

4.2 Modul ps.map

Modul *ps.map* lze jako většinu modulů používat z příkazové řádky GRASSu nebo prostřednictvím wxGUI. Grafické rozhraní je vytvořeno stejným způsobem jako u většiny ostatních modulů, dovoluje tedy zadat parametry příkazu v dialogu. V tomto případě ale spouštění přes grafické rozhraní neskýtá mimořádnou výhodu, jelikož v zásadě většinou stačí zadat jméno již připraveného konfiguračního souboru a souboru výstupního, což je pro mnohé uživatele pohodlnější prostřednictvím příkazové řádky. V příkazové řádce vypadá spuštění *ps.map* takto:

```
ps.map input=/konfiguracni/soubor.txt output=/vystupni/soubor.ps
```

Jak již bylo zmíněno v části 2.1.4, modul vytváří mapový výstup na základě konfiguračního souboru, což je obyčejný textový soubor s instrukcemi. Vytvořit tento soubor je většinou poměrně pracné, umisťování mapových prvků na vhodnou pozici se často neobejde bez vedlejších výpočtů, což ke konformitě používání příliš nepřispívá. Z toho je zřejmé, že v současnosti *ps.map* využívá pouze poměrně úzká komunita uživatelů GRASSu. Nové

grafické rozhraní přímo vytváří konfigurační soubor, na základě něhož poté *ps.map* vygeneruje mapový výstup. Není pak třeba se konfiguračním souborem zabývat, což může zpříjemnit používání tohoto modulu, a tím i rozšířit řady jeho uživatelů.

4.2.1 Konfigurační soubor

Konfigurační soubor má poměrně jednoduchou strukturu. Skládá se z instrukcí, které musí být na samostatných řádcích. Instrukce mohou obsahovat další dílčí instrukce umístěné opět na samostatných řádcích. Tyto víceřádkové instrukce musí být ukončeny instrukcí **end**, která musí být uvedena i na konci celého konfiguračního souboru. Dílčí instrukce je možné ve většině případů vynechat a pro nastavení dané vlastnosti se použije implicitní hodnota. Komentáře lze zapisovat pomocí znaku **#**, prázdné řádky jsou ignorovány. Pořadí instrukcí většinou nehraje roli s výjimkou instrukcí **vpoints**, **vlines** a **vareas**, které vykreslují vektorové vrstvy. První uvedená vektorová vrstva bude vykreslena nejvýše. Více napoví ukázka jednoduchého konfiguračního souboru:

```
paper a3
end
raster soils          # priklad jednoradkove instrukce
border y              # priklad viceradkove instrukce ...
    color 255:0:0
    width 3
end                    # ... ukoncene instrukci end
vpoints archsites
    symbol basic/diamond
    size 10
end
end                    # konec instrukci
```

Význam jednotlivých instrukcí je popsán v manuálové stránce k modulu, která je dostupná buď z dosavadního GUI nebo na internetových stránkách [1].

4.2.2 Volba rozsahu zobrazovaného území

V konfiguračním souboru lze nastavit velikost mapového okna, čímž je myšlen obdélník s vykreslenou rastrovou či vektorovou mapou. K nastavení slouží jednořádková instrukce **maploc**, která umístí levý horní roh okna na dané místo a volitelně nastaví i šířku a výšku mapového okna. Rozměry lze nepřímo ovlivnit také instrukcí **scale**, která na základě měřítka upraví rozměry okna.

Nicméně při jakýchkoli rozměrech a umístění mapového okna, se vykresluje stále stejný územní rozsah. Ten je dán tzv. *výpočetním regionem*. Proto je třeba před vlastním generováním mapového výstupu nastavit výpočetní region tak, aby odpovídal území, které

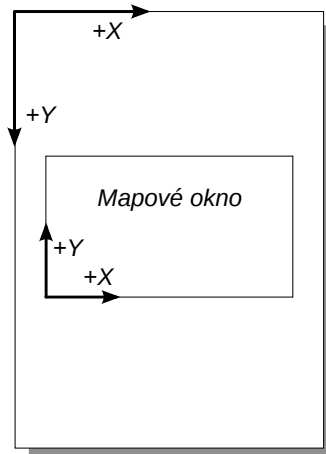
chceme zobrazit. K tomu slouží modul *g.region*. V konfiguračním souboru tedy nelze rozsah území ovlivnit. Modul *ps.map* tedy nejprve zjistí požadované rozměry a měřítko z konfiguračního souboru (jsou-li uvedeny) a následně se je pokusí aplikovat na rozměry současného výpočetního regionu. Když by se mapové okno při požadovaném měřítku nevešlo na daný formát papíru, modul měřítko upraví. Podobně upraví rozměry mapového okna, aby se vešlo do zadaných rozměrů.

Nové grafické rozhraní *ps.map* řeší volbu zobrazovaného území ve vlastní režii, není tedy třeba volat modul *g.region* a zároveň se nemění současný výpočetní region pro ostatní moduly.

4.2.3 Nekonzistence a chyby v modulu *ps.map*

Při tvorbě grafického rozhraní bylo nutné prozkoumat chování modulu *ps.map* o něco detailněji, než je nutné pro jeho běžné užívání. Vycházela jsem především z manuálové stránky [1] a vlastního experimentování s modulem.

Souřadnicové systémy a jednotky Jedním z nejnepríjemnějších problémů, se kterými bylo třeba se při tvorbě GUI vypořádat, byla nekonzistence v jednotkách a souřadnicových systémech. Modul *ps.map* totiž pro umístění jednotlivých prvků (jako legenda, text, měřítko) používá dva různé systémy, viz obr. č. 1.



Obrázek 1: Souřadnicové systémy používané modulem *ps.map*

První z nich – souřadnicový systém papíru – má počátek v levém horním rohu, kladná osa x směřuje doprava a osa y směrem dolů (tedy odpovídá systému obvyklému v počítačové grafice). Je-li poloha určena v tomto systému, očekávanou jednotkou je inch¹.

Druhý systém – systém mapového okna – má počátek souřadnic v levém dolním rohu mapového okna s pravotočivou orientací os obvyklou v matematice. Souřadnice lze v tomto

¹ česky palec, odpovídá 2,54 cm

systému zadávat dvěma způsoby. Buď v procentech výpočetního regionu nebo přímo v souřadnicích mapy. Například, při souřadnicích [100%, 0%] je objekt umístěn v pravém dolním rohu mapového okna, ať už je mapové okno na papíře umístěno kdekoli. Alternativně bychom toho samého umístění dosáhli zadáním mapových souřadnic bodu položeného co nejvíce na východ a jih v rozsahu výpočetního regionu. Oběma způsoby lze dosáhnout i umístění objektu mimo mapové okno, a to zadáním záporné hodnoty nebo hodnoty větší než 100 u procent, případně mapových souřadnic mimo výpočetní region.

Různé způsoby určení polohy mají své opodstatnění. V některých případech je výhodné umístit popis k určitému místu, jehož mapové souřadnice známe, a jeho poloha vzhledem k papíru není podstatná. Pozicování pomocí procent je vhodné například pro nadpis mapy, který chceme umístit přesně doprostřed nad mapové okno.

Problém nastává v případě, když chce uživatel určit polohu objektu relativně k papíru a tento objekt podporuje umístění pouze relativně k mapovému oknu, případně naopak. Bylo by proto vhodné, aby všechny objekty podporovaly umístění všemi těmito způsoby. Uživatel by si pak mohl vybrat, který z nich je pro danou situaci nejvhodnější.

Referenční bod objektu Modul *ps.map* u většiny objektů nepodporuje volbu referenčního bodu, tedy bodu objektu, ke kterému se vztahuje zadaná poloha. Výjimkou je text, u nějž si lze vybrat jeden z devíti referenčních bodů. Je zřejmé, že u textu má tato volba smysl, u ostatních objektů je otázka, zda by tato vlastnost byla využívána. U ostatních objektů je referenční bod dán a nelze měnit. Uživatelé však může překvapit, že není u všech objektů stejný. Ve většině případů se poloha vztahuje k levému hornímu rohu objektu, nicméně grafické měřítko (instrukce *scalebar*) má referenční bod ve středu. Bylo by vhodné tento přístup sjednotit, na druhou stranu nejde o nijak zásadní problém.

Volba barev V mnoha instrukcích se vyskytuje nastavení barvy. Bohužel velká část instrukcí omezuje výběr barev tím, že vyžaduje zadat barvu ve formě jejího jména, tedy např. *aqua*, *black*, *blue*, ... Některé jiné instrukce však umožňují zadat barvu i ve tvaru R:G:B. Bylo by proto logické, kdyby RGB podporovaly všechny instrukce.

Jednořádkové a víceřádkové instrukce Instrukce *border* (ohraničení mapového okna) a *colortable* (rastrová legenda) mohou být jednořádkové i víceřádkové, což mimo jiné znamená, že někdy je *end* za příkazem vyžadováno a někdy naopak následovat nesmí. Záleží, zda za zmíněné instrukce do stejného řádku napíšeme **y** nebo **n**, tím ji lze zapnout a vypnout.

```
border n      # jednoradkova instrukce – nesmi nasledovat end

border y      # viceradkova instrukce – musi nasledovat end
width 2
end
```

Je nepříjemné, že si uživatel musí být vědom tohoto chování, jinak by se mu mohlo snadno stát, že napíše instrukci **end**, kde být nemá, a *ps.map* to bude považovat za konec souboru a další instrukce bude ignorovat. Navíc u **colortable** nemá možnost zapínání a vypínání valného smyslu, když tuto instrukci neuvedu, legenda se prostě nevykreslí.

Mapinfo Mapový prvek *mapinfo*, se v některých případech špatně vykreslí. Problém nastává, když jej uživatel chce umístit nalevo od mapového okna. Modul *ps.map* přesune *mapinfo* tak, že se vertikálně zarovná k mapovému oknu, neboli převezme *x*-ovou sořadnici okna. Další chybou je špatné barevné vykreslení barvy pozadí a ohraničující linie. Barvy se nevykreslí, když je *mapinfo* mimo mapové okno.

Nekonzistence u instrukce *vlines* U instrukce *vlines* která přidává liniovou vektorovou vrstvu, lze použít dílčí instrukci *rgbcolumn*. Ta vykreslí linii barvou uvedenou ve zvoleném sloupci atributové tabulky na příslušném řádku. Podobně lze použít dílčí instrukci *cwidth*, která ovlivňuje šířku vykreslené linie. Šířka je pak určena zadanou hodnotou pronásobenou číslem kategorie (sloupec *cat* v atributové tabulce), do níž linie patří. V tomto případě si tedy uživatel nemůže zvolit, podle jakého atributu se bude šířka řídit, což v prvním případě jde. Bylo by vhodné tento přístup sjednotit.

Nepřesnosti v dokumentaci Manuálová stránka [1] k modulu *ps.map* je poměrně podrobná a vyčerpávající, přesto lze narazit na některé nepřesnosti. Například u instrukce *text* je k popisu dílčí instrukce *width* napsáno:

width of the lines used to draw the text to make thicker letters

Prakticky však tato instrukce ovlivňuje šířku rámečku kolem textu.

Popis instrukce **colortable** (vytváří rastrovou legendu) je místy matoucí. Jsou totiž dva typy rastrové legendy, což záleží na typu dat (*CELL* – celočíselný rastr a *FCELL*, *DCELL* – neceločíselný rastr). Některé dílčí instrukce mají pak smysl pouze pro jeden typ, případně se jejich význam pro jednotlivé typy liší. Situace se dále komplikuje dílčí instrukcí *discrete*, která umožňuje změnit typ legendy. Manuál se sice snaží problematiku vysvětlit, ale ne příliš úspěšně, účinnější je prakticky vše vyzkoušet.

Orientace stránky Volbu orientace stránky zajišťuje přepínač *-r* (*rotate*), který je-li uveden, mění orientaci stránky na šířku. Nevidím důvod, proč by se tato tato volba neměla provést již v konfiguračním souboru v rámci instrukce **paper**, kam logicky patří. Význam by přepínač měl, kdybychom chtěli vytvořit ze stejného konfiguračního souboru výstupy s různou orientací stránky, to ale smysl příliš nedává.

Titulek výstupního souboru Výstupní soubor má titulek *Mapset* = *<aktuální mapset>*, což mnoho neříká a navíc rovnítko v titulku nepůsobí hezky. Uživatel by mohl mít možnost změnit titulek, případně by se titulek měl převzít z aktuálního rastru, případně vektoru².

5 Grafické rozhraní pro modul *ps.map*

Důvody, proč je nové grafické rozhraní pro modul *ps.map* zapotřebí, jsou zmíněny již v části 4.2. Účelem je především oprostít uživatele od nutnosti ručně vytvářet konfigurační soubor.

5.1 Stažení programu z repozitáře add-ons

5.2 Podklady a použité knihovny

Nové grafické uživatelské rozhraní pro modul *ps.map* bylo vyvinuto pomocí knihovny wx-Python, proto bylo nutné se s knihovnou podrobněji seznámit. Jedním z nejvýznamnějších zdrojů informací je demo pro wxPython [3], ve kterém je názorně ukázáno chování všech dostupných widgetů. Při hledání vhodného widgetu pro aplikaci je demo velice užitečné.

Při psaní programu jsem pracovala s knihovnou *GRASS Python Scripting Library* [2], která umožňuje a zjednodušuje volání modulů GRASSu. Zdrojem informací mi byly také již napsané GRASS wxGUI moduly. Seznámila jsem se také s moduly GRASSu, zjišťovala jsem především, co umožňují, a jak je mohu pro práci využít.

5.3 Funkcionalita aplikace

V následujícím textu jsou shrnuty hlavní vlastnosti GUI aplikace a možnosti jejího využití.

Podporované instrukce Aplikace v současné době podporuje pouze vybrané instrukce, a to: *border*, *colortable*, *mapinfo*, *maploc*, *paper*, *raster*, *scale*, *scalebar*, *text*, *vareas*, *vlines*, *vpoints* a *vlegend*. V budoucnu se počítá s podporou dalších instrukcí. Pokud uživatel potřebuje použít některé dosud nepodporované instrukce, má možnost část práce vykonat v GUI aplikaci a nechat si vygenerovat konfigurační soubor. Tam tyto instrukce doplní a spustí přímo modul *ps.map*.

²GUI aplikace by sice mohla napravit tento nedostatek, nicméně je vhodnější toto upravit přímo v modulu *ps.map*.

Možné výstupy aplikace Výsledkem práce s GUI aplikací je mapový výstup ve formátu PostScript, případně Encapsulated PostScript, který je vygenerován modulem *ps.map* na základě konfiguračního souboru vytvořeném GUI aplikací. Dalším možným výstupem je konfigurační soubor samotný, který lze dále zpracovávat.

Čtení konfiguračních souborů Aplikace umožňuje také konfigurační soubory načítat. Vhodnější je ale načítat soubor vytvořený touto aplikací. To dáno tím, že aplikace při vytváření konfiguračního souboru zapisuje i informace o regionu do komentáře, což není standardní součástí konfiguračního souboru. Navíc je aplikace více otestovaná právě pro načítání vlastních souborů³.

Koncept a náhled Aplikace rozlišuje dva módy – koncept a náhled (*Draft mode* a *Preview mode*). Uživatel vytváří mapový výstup v módu konceptu, což znamená, že jednotlivé vykreslované prvky jako legenda či mapové okno jsou představovány pouze barevným obdélníkem s popisem typu objektu. Jejich vzhled tedy nijak nesouvisí se jejich skutečným vykreslením modulem *ps.map*, s výjimkou jejich rozměrů. Ty odpovídají rozměrům skutečným (alespoň se o to snaží, viz část **Problémy při tvorbě GUI**). Objekty lze posunovat po papíře, s výjimkou mapového okna však nelze měnit jejich rozměry. Při práci lze průběžně kontrolovat výsledek pomocí módu náhledu. Na pozadí je spuštěn modul *ps.map*, který vygeneruje dočasný PostScript soubor, ten je překonvertován do PNG a zobrazen v GUI aplikaci. To vše samozřejmě zabere poměrně dost času, záleží na složitosti mapového výstupu. Konverze do PNG je také poměrně časově náročná. Proto může být lepší si vygenerovat přímo PS soubor a ten si zobrazit v externím prohlížeči.

5.4 Základní práce s GUI aplikací

5.5 Problémy při tvorbě GUI pro *ps.map* a jejich řešení

Při práci jsem narazila na několik problematických oblastí, které se částečně kryjí se záležitostmi popsány v části 4.2.2 a 4.2.3.

Výpočetní region Připomeňme, že modul *ps.map* pracuje s aktuálním výpočetním regionem a vykresluje mapy pouze v tomto rozsahu. Jelikož by bylo nepohodlné, kdyby uživatel musel při práci v novém *ps.map* GUI nastavovat region externě pomocí GRASS modulu *g.region*, je vše řešeno v rámci této aplikace. Uživatel si v aplikaci určitým způsobem nastaví region, který pak *ps.map* použije k zobrazení. Stávající výpočetní region, který mezitím mohou používat další moduly GRASSu, zůstane nezměněný. Když si uživatel nechá vygenerovat konfigurační soubor, který chce dále editovat ručně, je v záhlaví v komentáři napsán příkaz pro modul *g.region*, který informuje o výpočetním regionu nastaveném v době vzniku

³Je nutné podotknout, že si aplikace nevytváří žádný vlastní odlišný formát.

souboru. Tato informace se využívá i při čtení konfiguračního souboru vytvořeného touto aplikací. Způsoby nastavení regionu jsou následující:

1. nastavit region na zvolenou rastrovou či vektorovou mapu
2. nastavit uložený region (*named region*)
3. nastavit aktuální výpočetní region
4. zvolit souřadnice středu mapy a měřítko (nastavení regionu se automaticky vypočítá)

Souřadnicové systémy a jednotky Nepříjemným problémem byly používané souřadnicové systémy a jednotky (viz část 4.2.3 na straně 4). Jelikož interaktivní umísťování objektů na stránce je jednou hlavních výhod grafického rozhraní, bylo nutné problém nejednotností systémů a jednotek řešit vlastními přepočty v programu tak, aby byl od něj uživatel co nejvíce odstíněn. Na druhou stranu by bylo logičtější, aby potřebné výpočty prováděl už samotný modul *ps.map*, který stejně už podobné výpočty provádí, stačilo by je pouze rozšířit. V důsledku toho se zřejmě určité výpočty provádí dvakrát – v *ps.map* a v GUI.

GUI aplikace je navržena tak, aby polohu objektů uživatel určil interaktivně kliknutím na objekt a táhnutím myši. Navíc lze v dialogích k objektům zadat přímo *x*-ovou a *y*-ovou souřadnici v systému papíru. Zde jsem rozšířila stávající možnosti *ps.map* o možnost zadat hodnoty v dalších jednotkách (mm, cm). U textového objektu, který se zadává v druhém systému (tedy mapové souřadnice či procenta region), jsou uživateli nabídnuty možnosti dvě. Buď zadat souřadnice v systému papíru, nebo pomocí mapových souřadnic. Možnost zadání procent regionu GUI aplikace zatím nepodporuje, myslím, že tato možnost není již tak potřebná, když lze polohu měnit jednoduchým tahem myši.

Referenční bod objektu Grafické měřítko má referenční bod ve svém středu, narozdíl od ostatních objektů, u kterých je to levý horní roh, viz 4.2.3. Rozhodla jsem se přístup sjednotit, proto GUI aplikace provádí drobný přepočet a pokud uživatel zadává ručně souřadnice, vztahují se k hornímu levému rohu. V dialogu je tato skutečnost uvedena, aby uživatelé zvyklí na původní ref. bod nebyli zmateni.

U textového objektu bylo řešení situace složitější. Zachovávám možnost zvolit si u textu jeho referenční bod, což znamená problémy při jeho zobrazování v GUI aplikaci. Situaci totiž komplikují další faktory jako rotace textu a odsazení (*offset*). GUI aplikace by musela na základě těchto parametrů provádět zbytečně složité výpočty, aby byla schopna určit přesné místo, kam se má text vykreslit a jaký je jeho ohraničující obdélník. V tomto místě jsem narazila na určité rezervy, které knihovna *wxPython* má. Očekávala jsem, že mi tyto výpočty knihovna aspoň částečně zjednoduší, ale jelikož jsem potřebné metody nenalezla, provedla jsem při výpočtech drobná zanedbání. Zobrazení v GUI aplikaci proto v některých případech není úplně přesné, nicméně je nutné zdůraznit, že to nijak nesouvisí se zobrazením textu ve vygenerovaném souboru.

Tvorba dialogů Každý mapový prvek jako např. mapinfo, grafické měřítko či legenda mají vlastní dialog, kde se nastavují jejich vlastnosti. Dialogy jsou různě rozsáhlé, ale mají podobný vzhled. Problém při tvorbě dialogů byla celková nejednotnost v nastavitelných vlastnostech dialogů. Například, rastrová legenda umožňuje změnit barvu písma a neumožňuje vykreslit ohraničení (*border*), zatímco u vektorové legendy je to přesně naopak. Takových případů je víc a značně komplikují snahy o ujednacení vzhledu dialogů. Proto by bylo příliš náročné vytvořit jakýsi jednotný dialog pro nastavení vlastností objektů, pravděpodobně by to vedlo k větší nepřehlednosti.

5.6 Ukázky

6 Závěr

Reference

- [1] GRASS Development Team. *GRASS GIS 7.0.svn Reference Manual* [online], c2003-2011 [cit. 2011-03-19].
URL: http://grass.fbk.eu/grass70/manuals/html70_user/ps.map.html
- [2] GRASS Development Team. *GRASS 7 Programmer's Manual* [online], c2000-2011 [cit. 2011-03-19].
URL: <http://grass.osgeo.org/programming7/>
- [3] Robin Dunn and Total Control Software. *wxPython demo* [program], c1997-2006 [cit. 2011-03-19].
URL: <http://downloads.sourceforge.net/wxpython/wxPython-demo-2.8.11.0.tar.bz2>