

Programátorský manuál k programu Tafel

Jan Sedlák

4. června 2009

Obsah

1	O programu	2
1.1	Kuželosečky	2
2	O kódu	3
2.1	Kompilace pod Unix* like systémy, např. Linuxem	3
2.2	Kompilace pod MS Windows	3
3	Výňatky z kódu	4
4	Soubor main.cpp	4
4.1	Funkce handle_arguments	4
4.2	Funkce main	4
5	Soubor equation.cpp a equation.h	5
5.1	Metoda return_ifExistsX a return_ifExistsY	5
5.2	Metoda draw	5
6	Soubory version.h a defines.h	5
7	Ostatní	6

1 O programu

Program Tafel slouží k vykreslování kuželoseček do kartézské souřadné soustavy.

1.1 Kuželosečky

Mezi obecné kuželosečky patří kružnice, elipsa, parabola a hyperbola. Bod a přímka se dají také vyjádřit obecnou rovnicí, ale nepatří mezi obecné kuželosečky (pro naprosté objasnění problému doporučuji článek kuželosečky na české wikipedii) a program Tafel se jimi nezabývá.

Všechny kuželosečky se dají vyjádřit obecnou rovnicí

$$a * x^2 + b * x * y + c * y^2 + d * x + e * y + f = 0$$

a v této obecné rovnici očekává program vstup¹.

Pro získání souřadnic x a y můžeme použít následující kvadratické rovnice (které získáme vypočítáním z obecné rovnice kuželoseček):

$$y_{1,2} = \frac{-(b*x+e) \pm \sqrt{(b*x+e)^2 - 4*c*(d*x+f)}}{2*c}$$

a

$$x_{1,2} = \frac{-(b*y+d) \pm \sqrt{(b*y+d)^2 - 4*a*(e*y+f)}}{2*a}$$

Idea byla taková, že po vykreslení souřadné soustavy program projede veškeré pixely ve směru osy x a poté ve směru osy y , zjistí zda-li pro daný bod existuje (popřípadě jestli je jeden, nebo dva) odpovídající bod a poté ho vykreslí. Protože pro zjištění používá předchozí rovnice, je patrné, že musí určitě být zadány hodnoty a a c , jinak je v programu dělení nulou (to je také ten důvod, proč program nemůže vykreslovat bod a přímku). Pravda je však taková, že u některých parabol je hodnota a NEBO hodnota c nulová. Naštěstí není případ, že by byly nulové obě zároveň, proto vše program kontroluje a poté vykresluje jenom v ose x (popřípadě jenom v ose y). Z tohoto důvodu jsou ale některé paraboly vykreslené nedokonale (jejich vrchní části jsou pouze vytečkovány). Z důvodu chybejících hodnot a a c také nelze vykreslit speciální případ hyperboly, rovnosou hyperbolu.

¹Vstup musí být v naprostě dopočítané obecné rovnici, program pouze pracuje s hodnotami, ale nesnaží se je nijak dopočítávat.

2 O kódu

Program Tafel je psaný v jazyce C++. Jazyk C++ je objektově orientovaný vyšší jazyk, dobře uzpůsobený práci na náročnějších projektech. Podpora objektů je v C++ opravdu na vysoké úrovni a proto není divu, že objekty využívá i můj program. Program Tafel dále využívá profesionální grafickou knihovku SDL (zveřejněnou pod licencí GNU/GPL). Program je psaný hlavně pro operační systém Linux (ač zkompilování pro M\$ Windows je možné také), který je doopravdy vhodně připraven a uzpůsoben pro programování a spouštění programů napsaných v jazycích C/C++.

Program byl psán v IDE Code::Blocks, opravdu velmi pohodlném, open-source a multiplatformním IDE pro psaní C/C++, dále v jednoduchém editoru Geany a textovém editoru vim určeným pro konzoli. Program byl kompilován kompilerem gcc (Gnu C Compiler), který je součástí projektu GNU.

2.1 Kompilace pod Unix* like systémy, např. Linuxem

Pro zkompilování programu na Unix systémy potřebujete knihovnu SDL, knihovnu SDL_gfx, C++ kompilér (např. gcc s podporou C++) a pro snazší kompilaci i program Gnu make. Obvyklý postup je, že ze stromu Vaší distribuce stáhnete balíčky SDL, SDL-devel a SDL_gfx, v terminálu se přepnete do adresáře s programem a napíšete „make“.

Dalším možným způsobem je samozřejmě zkompilování veškerých balíčků ze zdrojových kódů. Pokud nemáte program Gnu Make, stačí napsat `g++ -lSDL -lm -lSDL_gfx -Wall main.cpp equation.cpp -o tafel`

2.2 Kompilace pod MS Windows

Pro zkompilování balíčků pod Windows musíte mít staženy a zkompilovány knihovny SDL, SDL-devel a SDL_gfx. Bohužel, v binární podobě jsou pro Windows k nesehnání (s obtížemi se ale sehnat dají, zkuste google) a tak jsou dvě možnosti. Buď stáhnout a zkompilovat SDL (s tím není moc velký problém, návod je na oficiálních stránkách) a SDL_gfx (zde už je problém větší) ze zdrojových kódů, stáhnout si C++ kompilér pro Windows (MS C++ Visual Studio apod.) a zkompilovat zdrojové kódy v závislosti na zvoleném kompileru, či stáhnout si IDE jménem Dev-C++ (s kompilerem Mingw), odsud stáhnout SDL a SDL_gfx z balíčků jako plugin (volba „Získání aktualizací“) a zkompilovat poté program v něm. Hodně štěstí při kompilaci na tuto zvláštní, téměř nepoužitelnou a divně se chovající platformu.

3 Výňatky z kódu

Celý program je rozdělen do jednoho hlavního programu („main.cpp“), kde je většina důležitých funkcí (včetně funkce `int main()`), dále souborů objektu `Eq` (soubory „equation.cpp“ a k němu jeho hlavičkový soubor „equation.h“), souboru `defines.h` a `version.h`. Dále jsou s programem distribuovány soubory „Makefile“ (který slouží pro Gnu Make) a soubor „readme“, který poskytuje zjednodušenou nápovědu k tomuto programu.

4 Soubor main.cpp

Poznámka: Celý program je docela přehledně a rozsáhle okomentován, nejlepší zdroj odpovědí je tedy mnohokrát i samotný kód. Soubor obsahuje funkci `main` (což je C++ obdoba Pascalovského hlavního programu, tuto funkci volá samotný operační systém a jako argumenty mu předává parametry, se kterými byl program volán), dále funkci `draw_axis` (která na obrazovku vykreslí osy X a Y), `handle_arguments`, která má na starosti čtení a parsování parametrů programu, a funkci `create_meta_conics`, která se stará o vytvoření tolika kuželoseček, kolik jí bylo pomocí parametru `-n` předáno.

4.1 Funkce handle_arguments

Funkce `handle_arguments` je funkce na parsování paramterů, se kterými byl program volán. Vzhledem k tomu, že parametry programu je jediný vstup, jak může být běh programu ovlivňován, jedná se o docela komplexní funkci. Je volána funkcí `main` a přebírá od ní pole `cstringů` a počet parametrů v poli. Ty pak převede do proměnných, nastaví příslušný počet objektů-kuželoseček a nakonec se rekurzivně zavolá na zbylé prvky pole.

4.2 Funkce main

I přes snahu funkci `main` co nejvíce zjednodušit se veškeré důležité věci dějí přímo v této funkci. Hned na začátku je vyřešeno volání programu bez parametrů (program pak nemá jaké kuželosečky vykreslit) a volání verze a `helpfile` (`helpfile` aktuálně zatím neexistuje). Poté je zavolána funkce `handle_arguments`, která vyřeší parametry, dále se vytvoří objekty kuželoseček a jsou nastaveny na získané hodnoty. Poté je pomocí knihovny `SDL` inicializované okno, rozměry `800x600` a ve smyčce jsou pak volány metody „draw“ příslušných objektů. Je zde také kontrola stisknuté klávesy `Escape`. Nakonec je vypsána zpráva o zdařilém dokončení programu a program se ukončí.

5 Soubor equation.cpp a equation.h

Tyto soubory jsou soubory objektu Eq. V nich je deklarován a definován objekt, který reprezentuje obecnou kuželosečku. Objekt Eq je poté funkcí main nastaven na určité hodnoty a poté používá vlastní metody pro dopočet čísel a nakonec, aby se sám vykreslil na obrazovku.

5.1 Metoda return_ifExistsX a return_ifExistsY

Tyto metody používají část rovnice pro získání x a y jak byli řečeny výše (oddíl 1.1, str. 2) a to on část s diskriminantem. Pokud pro dané X (Y) je diskriminant záporný, neodpovídá těmto souřadnicím žádný bod, pokud je nulový tak právě jeden a pokud je kladný tak právě dva body.

5.2 Metoda draw

Tato metoda postupně projíždí Xovou a (ne u parabol) Yovou osu, pomocí ifExists zjišťuje, kolik bodů odpovídá těmto souřadnicím, pomocí return_y1-x2 získává souřadnice a poté je po pixelech na dané souřadnice vykresluje.

6 Soubory version.h a defines.h

Soubor version.h si uchovává informace o aktuální verzi programu. Jeho promenné jsou volány, pokud spustíte program s parametrem „-v“. Soubor defines.h obsahuje konstanty rozlišení, barev a podobně.

7 Ostatní

Pro další informace si přečtěte uživatelskou příručku, zdrojové kódy, nebo kontaktuje přímo mě na boloomka@gmail.com.

Text licence GNU/GPL by měl být distribuován s programem.

Sázeno v programu \LaTeX

©Jan Sedlák, 2009