

Zpravodaj Československého sdružení uživatelů TeXu

Jan Šustek

Práce s vrstvami v METAPOSTu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 23 (2013), No. 2, 94–99

Persistent URL: <http://dml.cz/dmlcz/150218>

Terms of use:

© Československé sdružení uživatelů TeXu, 2013

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Abstrakt

Cílem článku je ukázat možnosti práce s vrstvami v METAPOSTu. Části obrázku je možné kreslit v různých vrstvách. Jednotlivé vrstvy obrázku lze zpracovávat nezávisle na sobě. Vrstvy lze zobrazit v libovolném pořadí, bez ohledu na to, v jakém pořadí byly vytvořeny.

Klíčová slova: METAPOST, vrstvy.

1. Úvod

V počítačové grafice se vrstvy používají k oddělení různých částí obrázku, které se vytvářejí a zpracovávají samostatně. Na závěr se obrázek seskládá ze všech, nebo jen z některých vrstev.

V článku [1] autor ukázal použití vrstev na příkladu nakreslení křížovatky cest v mapě. Při vytváření části obrázku kreslil pomocí METAPOSTového příkazu `withpostscript`. Tento příkaz vloží přímo do postscriptového výstupu zadaný speciální text. Dále autor musel mít program, který zpracovává postscriptový výstup, naprogramovaný tak, aby daný speciální text správně interpretoval.

V tomto článku budou ukázána makra pro práci s vrstvami naprogramovaná přímo v METAPOSTu.

2. Implementace

Pro jednotlivé vrstvy budeme používat proměnné `vrstva[i]` typu `picture`. Hlavní obrázek, který se nakonec vyexportuje do postscriptového výstupu, bude v proměnné `vrstva[0]`. Problémem je, že pro kreslení do proměnných typu `picture` nemůžeme přímo použít obvyklá makra `draw`, `fill` a další, ale primitivní příkaz `addto`, což není pro uživatele zcela přirozené.

V současné implementaci využijeme toho, že se všechna makra na kreslení expandují na primitivní příkaz `addto`, kterým přidávají materiál k proměnné `currentpicture`. Token `currentpicture` však nadefinujeme jako makro, které se bude expandovat na tokeny `vrstva [i]`, takže po úplné expanzi budou všechna makra na kreslení přidávat materiál k proměnným `vrstva[i]`. Uživatel pak může do vrstvy `i` kreslit makrem

1 $V(i)$ ($\langle text \rangle$)

kde $\langle text \rangle$ je příkaz nebo příkazy, které by použil při obvyklém kreslení.

Makro V je definováno následovně.

```
2 def V(expr i)(text t)=
3   if (i=floor i) and (i>0):
4     if unknown vrstva[i]:
5       vrstva[i]:=nullpicture;
6       if i>maxcislovrstvy: maxcislovrstvy:=i; fi
7     fi
8     def currentpicture=vrstva[i] enddef;
9     t;
10    def currentpicture=vrstva[0] enddef;
11  else:
12    errmessage("Cislo vrstvy "&decimal(i)&
13      " musi byt kladne cele cislo");
14  fi
15 enddef;
```

Makro umožní zápis pouze do vrstvy, jejíž číslo je kladné celé. Pokud do vrstvy ještě nebylo zapisováno, vytvoří se na řádce 5 vrstva jako prázdný obrázek. Makro `currentpicture` je implicitně definováno jako `vrstva[0]` a po zápisu objektů do vrstvy na řádce 9 se makro takto opět definuje.

Makro `beginfig` drobně upravíme, aby navíc provedlo potřebnou inicializaci lokálních proměnných. Proměnná `zobrazrucne` typu `boolean` informuje, zda se mají vrstvy vykreslovat automaticky v přirozeném pořadí, nebo zda uživatel určí jejich pořadí ručně. Podrobnosti se dočtete v sekci .

```
16 def beginfig(expr n)=
17   maxcislovrstvy:=0;
18   def currentpicture=vrstva[0] enddef;
19   begingroup
20   charcode:=n;
21   save vrstva,zobrazrucne;
22   picture vrstva[];
23   boolean zobrazrucne; zobrazrucne:=false;
24   clearxy; clearit; clearpen;
25   pickup defaultpen;
26   drawoptions();
27 enddef;
```

Makro `endfig` také drobně upravíme. Nejdříve se v cyklu přidá obsah všech vrstev do proměnné `vrstva[0]` (reprezentované makrem `currentpicture`).

Nakonec se proměnná `vrstva[0]` makrem `shipit` exportuje do postscriptového výstupu.

```
28 def endfig=  
29   if not zobrazrucne:  
30     for i:=1 upto maxcislovrstvy:  
31       if known vrstva[i]:  
32         addto currentpicture also vrstva[i];  
33       fi  
34     endfor  
35   fi  
36   shipit;  
37 endgroup  
38 enddef;
```

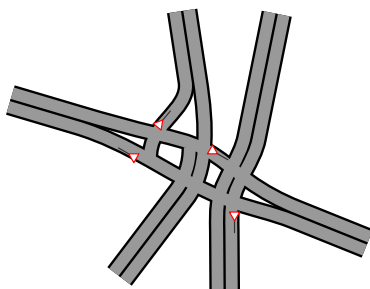
Výhodou této implementace spočívá v tom, že pokud nebudeme používat makro `V`, prostředí `beginfig...endfig` se bude chovat jako obvykle.

3. Příklad

Jako příklad nakreslíme křižovatku u Portail Rouge v Saint-Étienne. Při vhodném nadefinování bodů `b[i]` ji lze popsat následovně:

```
39 def krizovatka=  
40   silnice b1---b2..b3..b4..b5---b6;  
41   silnice b7---b8..b9..b10..b11..b12---b13; ...  
42   znacka((5.86,2.6),0); znacka((3.3,4.08),240); ...  
43 enddef;
```

Silnice nakreslíme jako tlustou černou čáru ve vrstvě 1 překrytou tenčí šedou čárou ve vrstvě 2. Dopravní značky budou ve vrstvě 4.



```

44 def silnice expr p=
45   begingroup interim linecap:=0;
46   V(1)(draw p withpen pencircle scaled (sirkasilnice+2sirkacary)
47     withcolor black;)
48   V(2)(draw p withpen pencircle scaled sirkasilnice
49     withcolor .6white;)
50   endgroup
51 enddef;
52 def znacka(expr b,a)=
53   V(4)(
54     draw ((0,-1)--(0,-3)) scaled 0.15 rotated a shifted b
55     withpen pencircle scaled sirkaznacky withcolor 0.2white;
56     fill ((0,-1)--((0,-1)rotated120)--((0,-1)rotated240))--cycle
57     scaled 0.15 rotated a shifted b withcolor white;
58     draw ((0,-1)--((0,-1)rotated120)--((0,-1)rotated240)--cycle)
59     scaled 0.15 rotated a shifted b
60     withpen pencircle scaled sirkaznacky withcolor red;
61   )
62 enddef;

```

Vidíme, že v argumentu makra *V* můžeme přirozeně používat všechny konstrukce pro kreslení včetně jejich poměrně volné syntaxe.

Celý obrázek v METAPOSTu nakreslíme jako obvykle:

```

63 beginfig(1)
64   krizovatka;
65 endfig;

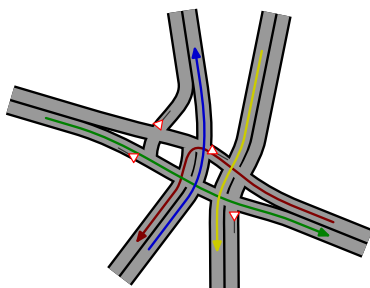
```

Přidáme-li ještě vrstvu 3, ve které naznačíme projíždějící auta, můžeme snadno vytvořit otázku z autoškoly.

```

66 beginfig(2)

```



```

67   krizovatka;
68   V(3)(drawarrow ...; drawarrow ...; ...)
69 endfig;

```

4. Další náměty

4.1. Ruční zobrazení vrstev

Níže definovaným makrem `zobrazvrstvy` můžeme do obrázku vložit jenom některé vrstvy, případně změnit pořadí vložení vrstev, nebo s určitými vrstvami provést nějakou transformaci. Po použití makra `zobrazvrstvy` bude potlačeno automatické vložení všech vrstev a makro `endfig` bude mít svůj původní význam.

```

70 def zobrazvrstvy(text s) text t=
71   for i:=s:
72     if known vrstva[i]:
73       addto currentpicture also vrstva[i] t;
74     fi
75   endfor
76   zobrazrucne:=true;
77 enddef;
78 beginfig(3)
79   krizovatka;
80   zobrazvrstvy(1) shifted (0.03,-0.03);
81   zobrazvrstvy(2 thru 5);
82 endfig;

```

Syntaxe makra `zobrazvrstvy` je poměrně volná. Seznam vrstev se makru předává v závorce ve stejném tvaru, jako by se použil uvnitř cyklu `for`. Za závorkou je nepovinný text určující další činnosti, které se mají s uvedenými vrstvami udělat před jejich vložení. Myslím, že z ukázek je naprosto jasné, co makro v daných případech provede.

```

83 zobrazvrstvy(1, 5 thru 7, 10 thru 20);
84 zobrazvrstvy(1,4,3,2) rotated 30;
85 zobrazvrstvy(3 step 2 until 9) scaled 2;
86 zobrazvrstvy(2 upto 4) shifted (cm,cm) withcolor black;

```

4.2. Vrstva s popisky

Je vhodné mít speciální vrstvu s popisky obrázků. Řekněme, že to bude vrstva 9, aby popisky byly dostatečně nahoře. Po následujících definicích pak každé použití makra `label` ve všech možných zápisech automaticky umístí popisek do vrstvy 9.

```
87 let orilabel=label;  
88 def label text t=V(9)(orilabel t); enddef;
```

Seznam literatury

- [1] Budaj, Martin. Using METAPOST as a library. *Zpravodaj Československého sdružení uživatelů T_EXu* [The Bulletin of the Czechoslovak T_EX Users Group], 22(1):2–8, 2012. ISSN 1211-6661.

Summary: Working with Layers in METAPOST

The paper shows possibilities of working with layers in METAPOST. It is possible to draw a picture in several layers. Users can work with one layer independently of the other layers. The layers can be drawn in an arbitrary order, regardless of the order of their creation.

Key words: METAPOST, layers.

*Jan Šustek, jan.sustek@osu.cz
Ostravská univerzita, Přírodovědecká fakulta, Katedra matematiky
30. dubna 22, CZ-701 03 Ostrava*