

Obsah

4. TRANSFORMACE OBRAZŮ	117
4.1. Změna umístění (translace)	118
4.1.1 Přesouvání bodu	119
4.1.2 Přesouvání obrazu	119
4.1.3 Přesouvání grafu	124
4.2. Změna rozměrů (scaling)	128
4.2.1 Měřítko přímky	129
4.2.2 Měřítko zobrazení	133
4.3. Změna orientace (rotace)	139
4.3.1 Rotace bodu	139
4.3.2 Rotace obrazu	141
4.4. Kombinované transformace	146
5. ANIMACE	155
5.1. Body a kružnice	156
5.1.1 Přímočará dráha	156
5.1.2 Křivkové dráhy	165
5.2. Přímky a mnohoúhelníky	170
5.2.1 Přímky	171
5.2.2 Mnohoúhelníky (polygony)	180
5.3. Složený pohyb	184
5.4. Pohyb pozadí	192
6. OKÉNKA NA displeji	201
6.1. Zvýrazňování	202
6.2. Mazání a vyřezávání	208
6.2.1 Mazání	208
6.2.2 Vyřezávání	209
6.3. Oblast pozorování (viewport)	225

STŘEDNÍ GRAFIKA

(Co ještě můžeme dělat?)

V předešlých kapitolách jsme se seznámili se základními technikami počítačové grafiky. Umíme už vytvářet grafy a obrázky různých druhů. Nyní se budeme zabývat metodami manipulace s nimi. Uvidíme, jak můžeme modifikovat jejich části a jak můžeme animace.



4. TRANSFORMACE OBRAZŮ



Transformační metody, o nichž bude v této kapitole řeč, jsou základními nástroji pro manipulaci s obrazy. Pochopíme-li principy potřebné pro modifikaci nebo transformaci vytvořeného obrazu, značně se tím zvýší vaše schopnost vytvářet efektivní počítačovou grafiku. Specifickým využitím transformačních metod je např. přemístění obrázků a grafů z jednoho místa stínítka obrazovky na jiné, vytváření složitějších obrazů z jednodušších modulových částí, vytváření doplňků do hotových obrazů, zvětšování obrazu nebo grafu kvůli lepší srozumitelnosti. Dále se využívá např. při vkládání dalších detailů, zmenšování obrazu kvůli možnému doplnění vysvětlujícími informacemi a při animaci obrazů.

K proměnám jednoduchých obrázků a grafů dospějeme aplikací tří základních transformací:

- 1) přesunem zobrazených objektů z jednoho místa stínítka na jiné (translaci)
- 2) zvětšením nebo zmenšením velikosti zobrazeného obrázku nebo grafu (měřítkem)
- 3) změnami smyslu orientace grafické reprezentace (rotací).

4.1. Změna umístění (translace)

Techniky pro změnu umístění na stínítku obrazovky nám umožňují vytvářet obraz nebo graf ve zvolené části stínítka obrazovky a pak jej přesouvat na libovolné jiné místo stínítka. Můžeme buď změnit uspořádání celého obrazu, nebo vytvoříme nový obraz z určité množiny složek.

4.1.1 Přesouvání bodu

Přemístění zobrazovaného bodu se děje pouze změnou souřadnic bodu na hodnoty odpovídající novému umístění. Přesunutí bodu horizontálně o 15 jednotek provedeme přičtením 15 k hodnotě souřadnice X bodu. Pokud chceme také přesun ve směru souřadnice Y o 40 (nahoru), musíme odečíst 40 od souřadnice Y. Obecné vztahy pro přesun bodu na stínítku o souřadnicích (X,Y) do nové pozice o souřadnicích (XT,YT), daném vzdálenostmi posunu H a V, můžeme napsat ve tvaru

$$\begin{aligned} XT &= X + H \\ YT &= Y + V \end{aligned} \tag{6-1}$$

Kladná hodnota H znamená horizontální posun bodu doprava, kladná hodnota V znamená vertikální posun směrem dolů. Záporné hodnoty H nebo V představují přesun bodu vlevo nebo nahoru.

Při přesouvání zobrazeného bodu užíváme hodnoty H a V; dovolí nám zobrazení bodu na obrazovce. Pokud jsou hodnoty H a V menší než 0,5, zobrazí se původní bod na stejném místě. Větší počet vícenásobného zobrazení jednotlivých bodů může vést k výraznému zpomalení programu.

4.1.2 Přesouvání obrazu

Přesunutí obrazu z jednoho místa stínítka na jiné dosáhneme tak, že přemístíme všechny body obrazu a pak znova nakreslíme všechny čáry propojující přesunuté body. Abychom zabránili zkreslení objektu, musíme posunout všechny body o stejnou vzdálenost. To znamená, že vztahy

/6-1/ jsou aplikovány se stejnou hodnotou H na všechny horizontální souřadnice a se stejnou hodnotou V na všechny vertikální souřadnice obrazu. Užijeme-li pro různé body obrazu různé hodnoty H nebo V dojde ke zkreslení původního obrazu. Ovyklo chceme přesouvat objekty bez deformace, ovšem někdy může být zavádění takových zkřivení základem experimentů při navrhování tvarů nebo při hraničer.

Souřadnice koncových bodů každé čáry mohou být jednoduše zapsány do nějakého pole. Pro jednoduchý obrázek to může být pole jednorozměrné, do něhož uložíme body v tom pořadí, v němž se mají propojit. Pokud však pracujeme se složitějším obrazem, kde jsou body spojeny několika čarami nebo v němž jsou části, které nejsou spojeny čarami s jinými částmi obrazu, jak je uvedeno na obr. 6-1, můžeme zapsat souřadnice bodů do dvourozměrných polí. Jeden index pole pak identifikuje část obrazu, druhý index označuje určitý bod této části. Takže hodnota (X(2,1), Y(2,1)) se vzťahuje pro uložení souřadnic počátečního bodu druhé části obrazu (jako je chlapec v obr. 6-1). Dvourozměrné pole je užito v prog. 6-1 pro zápis souřadnic definujících detaily z obr. 6-1. Tento program nám umožňuje přenést složky obrazu na libovolné místo stínítka obrazovky tolikrát, kolikrát si to přejeme.

Program prog. 6-1 zapisuje posunuté souřadnice do původních polí X a Y. Pokud chceme zachovat původní pozici, zaznamenáme přesunuté souřadnice do jiných polí, než jsou XT a YT. Zachování původní pozice obrazu je třeba pouze tehdy, pokud chceme vyzkoušet, jak bude nové umístění vypadat, nebo chceme-li zobrazit tento objekt v obou pozicích. Pokud není žádný důvod pro zachování původní pozice obrazu, ušetříme paměť zápisem přepočítaných hodnot souřadnic do původních polí X a Y.

Pro objekty se symetrií nebo s hranicemi vypočítanými podle rovnic nemusíme přičítat vzdálenosti posuvu ke všem bodům objektu, abychom jej přesunuli. Např. pro přesunutí kružnice nebo elipsy stačí přesunout střed obrazu a pak vypočítat nové body na této hranici podle rovnic pomocí podprogramů uváděných v kapitole 5. Pro přemístění pravoúhelníku stačí přemístit jeden vrchol a pak znovu nakreslit čtyřúhelník při užití hodnot šířky a výšky.

```

10  'PROGRAM 6-1. PRESUNUTI CASTI OBRAZU.
20  'KRESLENI OBRAZKU A PAK MUZE UZIVATEL PRESOUVAT
30  'CASTI OBRAZU NA JINA MISTA. CASTI OBRAZU JSOU
40  'ZAPSANY V POLICHE X A Y. PRESUNUTE BODY NAHRADI
50  'PUVODNI BODY V POLICHE X A Y.

60  CLS
70  DIM X(5,50), Y(5,50), N(5)
80  XM = 319  'XM,YM JSOU MAXIMALNI HODNOTY SOURADNIC
90  YM = 199
100   '----- NACTENI A NAKRESLENI CASTI OBRAZU -----
110 PN = 1    'PN JE CISLO CASTI
120 E = 0
130 READ XD, YD
140 IF XD >= 0 THEN 200
150   'ZAPSANI POCTU ELEMENTU CASTI PN DO POLE N(PN)
160 N(PN) = E
170 IF XD = -100 THEN 240
180 PN = PN + 1
190 GOTO 120
200 E = E + 1
210 X(PN,E) = XD
220 Y(PN,E) = YD
230 GOTO 130
240 SCREEN 1
250 GOSUB 510  'NAKRESLENI OBRAZKU
260 '
270   '----- ZADANI POSUNUTI -----
280 LOCATE 22,1
290 PRINT "      1-CLAPEC 2-PES 3-HYDRANT"
300 PRINT "      PRESUNOUT CAST OBRAZU (0 PRO KONEC)";
310 INPUT P

```

```

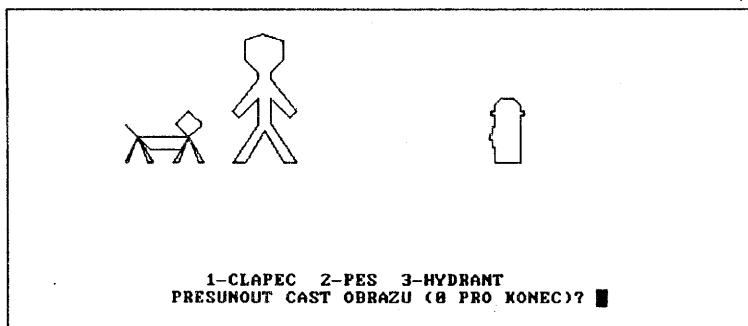
320 LOCATE 22,1
330 PRINT "
340 PRINT "
350 IF P = 0 THEN 730
360 LOCATE 23,1
370 PRINT " HODNOTY PRO PRESUN";
380 INPUT H, V
390 '
400 '----- PREPOCET BODU -----
410 FOR J = 1 TO N(P)
420     X(P,J) = X(P,J) + H
430     Y(P,J) = Y(P,J) + V
440     IF X(P,J) >= 0 AND X(P,J) <= XM AND Y(P,J) >= 0
        AND Y(P,J) <= YM THEN 470
450     PRINT "SOURADNICE MIMO STINITKO !"
460     GOTO 730
470 NEXT J
480 GOSUB 510
490 GOTO 280
500 '
510 '----- PODPROGRAM PRO KRESLENI -----
520 CLS
530 FOR K = 1 TO PN
540     FOR J = 1 TO N(K)-1
550         LINE(X(K,J),Y(K,J))-(X(K,J+1),Y(K,J+1))
560     NEXT J
570 NEXT K
580 RETURN
590 '
600 DATA 85,70,90,75,105,60,105,80,85,110,95,110,110,85
610 DATA 125,110,135,110,115,80,115,60,130,75,135,70,115,45
620 DATA 115,40,125,30,125,15,110,10,95,15,95,30,105,40,105
630 DATA 45,85,70,-1,-1
640 DATA 50,90,62,110,58,110,50,90,42,110,38,110,50,90
650 DATA 45,100,20,100,10,90,22,110,18,110,10,90,2,110
660 DATA 0,110,10,90,0,80,10,90,50,90,40,80,50,70,55,75,60
670 DATA 78,60,82,50,90,-1,-1
680 DATA 290,110,290,98,288,98,288,92,286,92,286,88
690 DATA 286,88,288,88,288,82,290,82,290,73,287,73,287,70
700 DATA 290,70,290,65,295,60,305,60,310,65,310,70,313,70
710 DATA 313,73,310,73,310,110,290,110,-100,-100

```

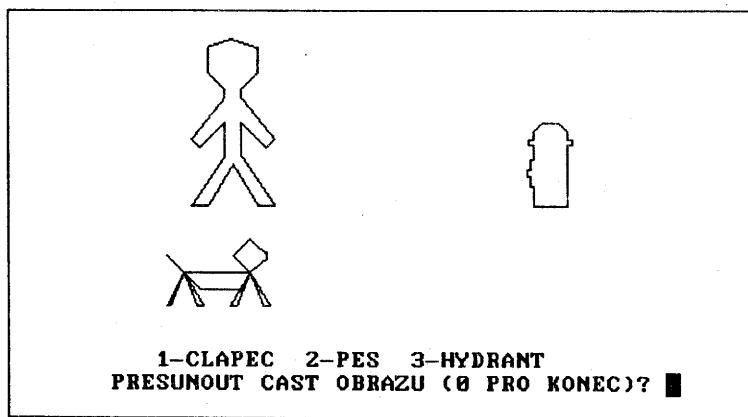
720 -----

730 END

Prog. 6-1 Přenášení obrázků (chlapec, pes a hydrant).



(a)



(b)



(c)

Obr. 6-1 Přesunutí složky obrazu z původní pozice (a) na novou pozici (b) a pak na pozici (c) programem Prog. 6-1.

4.1.3 Přesouvání grafu

Stejné metody a pravidla jako pro přesouvání obrazů platí samozřejmě i pro grafy - všem bodům grafu přiřadíme nové hodnoty (při respektování hranic stínítka obrazovky) a všechny čáry se překreslí na nové místo. Pak můžeme doplnit i popis na příslušná místa v novém grafu nebo posunout pozice popisu společně s ostatními souřadnicemi obrazu.

Pro posunutí znakových popisů musíme velikostí vzdáleností posuvu H a V vyjádřit v jednotkách změn znakových pozic. Dostaneme horizontální posuv popisu dělením H počtem bodů na znak horizontálně. Vertikální posuv dostaneme podobně dělením V počtem bodů na znak vertikálně. Pro znaky 8x12 by byla řádková pozice popisu zobrazena ve vzdálenosti V/12 a horizontální pozice (sloupec) popisu by byla posunuta o vzdálenost H/8. Pokud chceme posunout popisy, měli bychom volit H a V tak, aby byly

celočíselnými násobky rozměrů znakového pole. Potom budou popisy ve stejné relativní pozici vzhledelem ke grafu.

Metodu pro přesouvání sloupcového grafu s popisem uvádí prog. 6-2. Pro posouvání popisu se používá znakových polí 8x12. Funkce LEN se používá pro určení délky každého řetězce znaků popisu při kontrole správnosti přesunu, tzn. zda nedojde ke zobrazení popisu mimo stínítko obrazovky.

```
10  'PROGRAM 6-2. PRESOUVANI HORIZONTALNIHO SLOUPCOVEHO
20    'GRAFU S POPISEM. VELIKOST STINITKA SE PREDPOKLADA
30    '320x200 S 40 ZNAKY NA RADEK A 25 RADKY (KADZY ZNAK
40    'SE ZOBRAZI NA POLI 8x8 PIXELU). PROMENNY POCET
50    '(AZ 8) POPISU A HODNOT JE CTEN PRIKAZEM DATA. PO-
60    'PISY JSOU ULOZENY V POLI L$. TISKova POZICE KAZDE-
70    'HO POPISU JE ZAPSANA V POLICHE R A C. POPISY MOHOU
80    'BYT AZ 8 ZNAKU DLOUHE A ZABIRAJI PRVYCH 8 POZIC
90    'TISKOVEHO RADKU. HODNOTY JSOU ZOBRAZENY MERITKEM
100   'TAK, ABY VYSLEDNE SOURADNICE BYLY V INTERVALU 64-
110   '160. KONCOVE BODY SLOUPCE PRO KAZDOUN CAST GRAFU
120   'JSOU ZAPSANY V POLICHE X A Y. VYTVORENY GRAF
130   'MUZE BYT PRENESEN NA NEJAKE JINE MISTO NA STINIT-
140   'KU. PRESUNUTE BODY JSOU ZAPSANY DO NOVYCH POLI.
145 '
150 CLS
160 DIM X(8,2), Y(8,2), L$(9), R(9), C(9)
170 DIM XT(8,2), YT(8,2), LT$(9), RT(9), CT(9)
180 '
190      '----- NACTENI DAT GRAFU A KRESLENI -----
200 READ MI, MA 'NACTENI MIN. A MAX. HODNOTY DAT
210 RM = (160 - 64) / (MA - MI)
220 'NACTENI DAT GRAFU
230 READ N      'NACTENI POCTU CASTI GRAFU (SLOUPCU)
240 FOR K = 1 TO N
250   READ L$(K), M
260   R(K) = K + 2 'PONECHANI PRVYCH DVOU RADKU PRO
      NAZEV A MEZERU
270   C(K) = 1
280   X(K,1) = 64
290   X(K,2) = INT((M-MI) * RM + 64 + .5)
```

```

300      Y(K,1) = (K + 1) * 8 + 1  'VRSEK SLOUPCE,
310      Y(K,2) = Y(K,1) + 4        'SPODEK SLOUPCE,
320 NEXT K
330 READ L$(N + 1)    'NACTENI POPISU SLOUPCE,
340 R(N + 1) = 1
350 C(N + 1) = 10 - LEN(L$(N+1))/ 2  'CENTROVANI PODLE
           SLOUPCE 10
360      'KOPIROVANI PUvodNICH HODNOT DO POLI UZITYCH PRO
370      'ZAPIS' PREMISTENYCH BODU, PROTOZE TY JSOU UZITY PRO
375      'KRESLENI
380 FOR K = 1 TO N
390      RT(K) = R(K)
400      CT(K) = C(K)
410      FOR J = 1 TO 2
420          XT(K, J) = X(K, J)
430          YT(K, J) = Y(K, J)
440      NEXT J
450 NEXT K
460 RT(N + 1) = R(N + 1)
470 CT(N + 1) = C(N + 1)
480 SCREEN 1
490 GOSUB 830      'NAKRESLENI GRAFU
500 '
510      '----- VOLBA PRESUNU -----
520 LOCATE 20, 1
530 PRINT " HODNOTY H A V PRO PRESUH";
540 PRINT " ORIGINALU. UVED 0,0 PRO ORIGINAL";
550 PRINT " -999,-999 PRO KONEC";
560 INPUT H, V
570      'VYMAZANI POKYNU
580 LOCATE 20, 1
590 PRINT " ";;
600 PRINT " ";;
610 PRINT " ";;
620 PRINT " ";;
630 IF H = -999 THEN 1080
640 '
650      '----- PREPOCET BODU -----
660 FOR K = 1 TO N
670     RT(K) = R(K) + INT(V / 8)
680     IF RT(K) < 1 OR RT(K) > 15 THEN 980

```

```

690     CT(K) = C(K) + INT(H / 8)
700     IF CT(K) < 0 OR CT(K) + LEN(L$(K)) > 35 THEN 980
710     FOR J = 1 TO 2
720         XT(K, J) = X(K, J) + H
730         IF YT(K, J) < 0 OR YT(K, J) > 279 THEN 1000
740         YT(K, J) = Y(K, J) + V
750         IF YT(K, J) < 0 OR YT(K, J) > 159 THEN 1000
760     NEXT J
770 NEXT K
780 RT(N + 1) = R(K) + INT(V / 12)
790 CT(N + 1) = C(K) + INT(H / 8)
800 GOSUB 830
810 GOTO 520
820 '
830 '----- PODPROGRAM PRO KRESLENI -----
840 CLS
850 LOCATE RT(N + 1), CT(N + 1)
860 PRINT L$(N + 1)
870 PRINT
880 FOR K = 1 TO N
890     LOCATE RT(K), CT(K)
900     PRINT L$(K);
910     'KRESLENI SLOUPE PRO CAST
920     FOR YT = YT(K, 1) TO YT(K, 2)
930         LINE(XT(K,1), YT)-(XT(K,2), YT)
940     NEXT YT
950 NEXT K
960 RETURN
970 '-----
980 PRINT "POPIS MIMO OBRAZOVKU"
990 GOTO 1080
1000 PRINT "BOD GRAFU MIMO OBRAZOVKUP
1010 GOTO 1080
1020 '-----
1030 DATA 0,800
1040 DATA 4
1050 DATA FIRST,600,SECOND,500,THIRD,800,FOURTH,400
1060 DATA "QUARTERLY SALES"
1070 '-----
1080 END

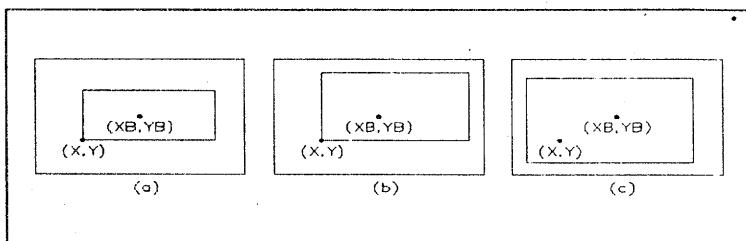
```

Prog. 6-2 Přenášení grafu.

KOREKTURA

4.2. Změna rozměrů (scaling)

Máme-li vytvořen nějaký obraz můžeme se, v zájmu zvýšení srozumitelnosti informací které znázorňuje nebo aby se tento obraz vešel do většího grafického obrazu, rozhodnout pro změnu jeho velikosti. Tato velikost obrazu nebo grafu se změní vynásobením všech vzdáleností mezi body hodnotou, která určuje zvětšení nebo zmenšení obrazu. Chceme-li zdvojnásobit velikost pravoúhelníka, vynásobíme délky všech čar 2x; má-li být velikost zmenšena na polovinu, jsou všechny délky násobeny 1/2. Změna měřítka nějakého objektu vyžaduje, abychom také uvedli umístění tohoto objektu po zvětšení, nebo zmenšení. Velikost objektu můžeme měnit ve vztahu k nějakému centrálnímu bodu objektu, k bodu na jeho obvodu, nebo k libovolnému bodu mimo tento objekt. Například, můžeme požadovat zvětšení pravoúhelníka při ponechání levého dolního rohu v téže pozici po zvětšení, nebo bychom mohli zvětšit pravoúhelník kolem jeho středního bodu (viz Obr. 6-2).



Obr. 6-2 (a) Pravoúhelník se středem v (XB, YB) a levým dolním rohem v (X, Y) ; (b) Změna měřítka s pevným bodem (X, Y) zvětší obdélník nahoru a vpravo; (c) změna měřítka s pevným bodem (XB, YB) zvětší obdélník směrem od středu ven.

4.2.1 Měřítko přímky

Horizontální přímka s levou hodnotou souřadnice X = X(1) a pravou hodnotou X(2) má délku

$$L=X(2)-X(1) \quad (6-2)$$

Změnit délku této přímky znamená násobit L hodnotou horizontálního měřítka, označenou HS, čímž vznikne nová hodnota délky:

$$\begin{aligned} LS &= L * HS \\ &= X(2)*HS - X(1)*HS \\ &= XS(2) - XS(1) \end{aligned} \quad (6-3)$$

kde pole XS obsahuje souřadnice zobrazené pomocí měřítka a HS musí být kladné číslo (HS > 0). Měřítko HS vytvoří delší přímku, pokud je HS > 1. Výsledná přímka bude kratší, pokud je HS < 1. Je-li HS = 1 přímka se nezmění.

Protože měřítko přímky mění hodnoty souřadnic koncových bodů, musíme určit, kde se má výsledná přímka nakreslit. V případě horizontální přímky můžeme považovat za fixní levý krajní bod, pravý krajní bod nebo střední bod přímky, nebo můžeme přemístit novou přímku na zcela jinou pozici. Předpokládejme, že chceme zachovat původní polohu levého krajiního bodu přímky. Nové souřadnice přímky, po změně měřítka, pak jsou

$$XS(1)=X(1) \quad (6-4)$$

$$XS(2)=X(1)+(X(2)-X(1))*HS$$

Pokud chceme zachovat pravý krajní bod přímky v původní poloze, jsou souřadnice přímky

$$\begin{aligned} XS(1) &= X(2)-(X(2)-X(1))*HS \\ XS(2) &= X(2) \end{aligned} \tag{6-5}$$

Pro zachování původní pozice středního bodu přímky dostaneme

$$\begin{aligned} XS(1) &= XM - \frac{X(2)-X(1)}{2}*HS \\ XS(2) &= XM + \frac{X(2)-X(1)}{2}*HS, \end{aligned} \tag{6-6}$$

kde XM je střední bod (Obr. 6-3), který můžeme vypočítat jako

$$XM = \frac{X(1)+X(2)}{2}.$$

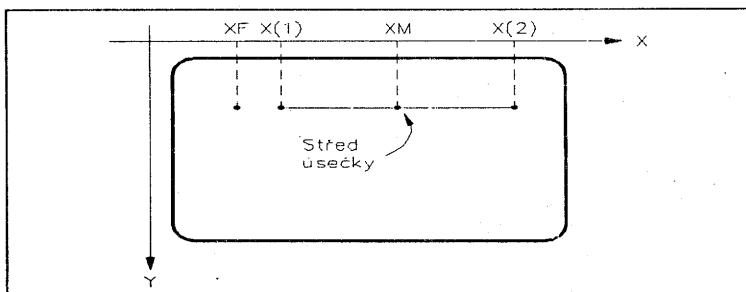
Pokud potřebujeme přesunout přímku se změněným měřítkem do nové pozice na obrazovce, přemístíme nejprve původní přímku do nové pozice a pak provedeme změnu měřítka přímky buď podle koncových bodů, nebo podle středního bodu. Obecně se postupuje tak, že nejprve vytvoříme referenční bod, jako je např. XF na Obr. 6-3.

Nové koncové body pak dostaneme změnou měřítka vzdáleností mezi koncovými body a tímto referenčním bodem

pomocí horizontálního měřítka HS. Tyto nové body lze vypočítat jako

$$\begin{aligned} XS(1) &= (X(1) - XF) * HS + XF \\ XS(2) &= (X(2) - XF) * HS + XF \end{aligned} \quad (6-7)$$

Jako referenční bod XF lze zvolit libovolný bod na obrazovce nebo mimo ní. Pokud je zvolen XF jako X(1) anebo X(2) nebo XM, dostaneme výsledky dané vztahy (6-4), (6-5), nebo (6-6).



Obr. 6-3 Horizontální přímka s koncovými body X(1) a X(2) má délku X(2) - X(1) a střední bod XM = X(1) + (X(2) - X(1))/2.

Pravidla pro změnu měřítka vertikální přímky se podobají pravidlům pro horizontální přímku. Délka vertikální přímky s koncovými body Y(1) a Y(2) je

$$L = Y(2) - Y(1) \quad (6-8)$$

Změnou měřítka této vertikální čáry pomocí koeficientu vertikálního měřítka VS vznikne přímka o nové délce

$$\begin{aligned}
 LS &= L + VS \\
 &= Y(2) * VS - Y(1) * VS(6 - 9) \\
 &= VS(2) - VS(1),
 \end{aligned} \tag{6-9}$$

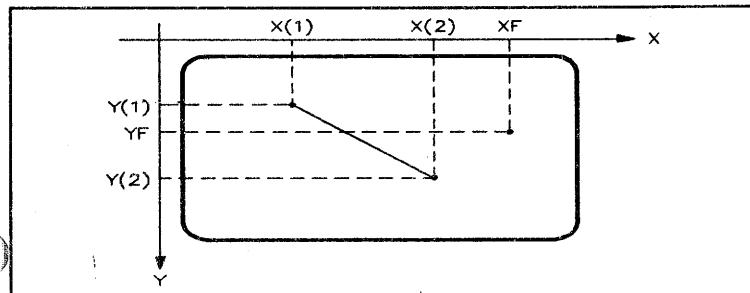
kde pole VS je užito pro uložení nových koncových bodů. Je-li měřítko VS kladné, může být přímka překreslena podle libovolného pevného bodu, jako jsou Y(1) nebo Y(2), nebo střední bod. Určením pevného bodu jako YF (vertikální referenční bod) dostaneme pro souřadnice nových koncových bodů

$$\begin{aligned}
 VS(1) &= (Y(1) - YF) * VS + YF \\
 VS(2) &= (Y(2) - YF) * VS + YF
 \end{aligned} \tag{6-10}$$

Přímky skloněné pod libovolným úhlem (Obr. 6-4) prodlužujeme nebo zkracujeme změnou měřítek jak horizontální X(2) - X(1), tak vertikální složky Y(2) - Y(1) relativně k pevnému referenčnímu bodu o souřadnicích (XF, YF). Vztahy (6-7) a (6-10) můžeme pak napsat v následující formě

$$\begin{aligned}
 XS(1) &= X(1) * HS + XF * (1 - HS) \\
 VS(1) &= Y(1) * VS + YF * (1 - VS) \\
 XS(2) &= X(2) * HS + XF * (1 - HS) \\
 VS(2) &= Y(2) * VS + YF * (1 - VS)
 \end{aligned} \tag{6-11}$$

Výrazy XF * (1 - HS) a YF * (1 - VS) jsou konstantní, tj. pro všechny souřadnice je musíme vyhodnotit pouze jednou. Pro prodloužení nebo zkrácení šikmé přímky beze změny směrnice musí platit HS = VS.



Obr. 6-4 Měřítko přímky s horizontální složkou délky $X(2) - X(1)$ a vertikální složkou délky $Y(2) - Y(1)$.
Ize měnit relativně podle libovolného pevného bodu (XF, YF) .

4.2.2 Měřítko zobrazení

Nastavíme-li HS a VS na shodnou hodnotu S, zvětšíme nebo zmenšíme obrázky a grafy. Vztahy (6-11) jsou užívány pro výpočet nových souřadnic všech bodů nového zobrazení po změně měřítka. Zvětšený nebo zmenšený obrázek se pak utváří propojováním čar těchto nových bodů. Pokud jde o obrázky složené z matematicky snadno popsatelných křivek, není nutné aplikovat (6-11) na každý bod obrazu. Chceme-li změnit změnit kružnice o poloměru R, můžeme vypočítat pouze nové souřadnice středu kružnice pomocí vztahů (6-11) a určit nový poloměr jako $R * S$. Pro zobrazení kružnice se změněným měřítkem pak můžeme použít podprogramy z části 5-1.

Někdy potřebujeme protáhnout obraz v jednom směru (např. ve vertikálním) a zachovat původní velikost v druhém směru (horizontálním). Chceme-li změnit čtverec na obdélník, můžeme např. zdvojnásobit pouze výšku, a to tak, že nastavujeme vertikální složku měřítka VS pro výpočty vertikálních souřadnic na hodnotu 2 a současně horizontální složku HS na hodnotu 1 pro výpočty horizontálních souřadnic.

Tato metoda může být užitečná u experimentů nebo u nastavování proporcí složek obrázku. Také při vytváření složených obrazů z nějaké množiny standardních tvarů můžeme použít nejednotnou změnu měřítka a např. ze čtverce můžeme vytvořit obdélník,

Program Prog. 6-3 uvádí příklad změny měřítka obrázku. Vstupní data programu určují obrázek a obě složky měřítka HS a VS. Tento program mění měřítko celého obrázku nebo libovolné vybrané části obrázku. Příklad výstupu je uveden na Obr. 6-5.

```
10  'PROGRAM 6-3. ZMENA MERITKA CELEHO NEBO CASTI OBRAZKU.
20  'NAKRESLI AUTO A UMOZNI UZIVATELI ZMENIT MERITKO
30  'CASTI NEBO CELEHO AUTA. POKUD SE MENI MERITKO CAS-
40  'TI, NASTAVI PROGRAM PEVNE BODY. POKUD SE MENI
50  'CELE, MUZE UZIVATEL UVEST BOD, KTERY MA BYT PEVNY.
60  'NOVE BODY NAHRADI PUVOVDNI HODNOTY V POLICHE X A Y.
70  CLS
80  DIM X(5, 100), Y(5, 100), N(5)
90  PRINT "UVED MAXIMALNI HODNOTU HORIZONTALNI A VERTIK."
100 PRINT "SOURADNICE PRO TENTO ROZLISOVACI REZIM"
110 INPUT XM, YM
120 '
130  '----- NACTENI CASTI OBRAZKU A KRESLENI -----
140 PN = 1  'PN JE CISLO CASTI,
150 E = 0  'E JE PRVEK CASTI,
160 READ XD, YD
170 IF XD >= 0 THEN 240  'JE TO NOVY DATOVY BOD ?
180  'JINAK JE TO KONEC CASTI. ZAPAMATUJ OKAMZITOU
     HODNOTU E V N(PN).
190 N(PN) = E
200 IF XD = -100 THEN 280 'JE-LI KONEC DAT, NAKRESLI OBR.
210  'JINAK NOVA CAST. NACTI PRVKY TETO CASTI.
220 PN = PN + 1
230 GOTO 150
240 E = E + 1
250 X(PN, E) = XD
```

```

260 Y(PN, E) = YD * 5/6      'KOREKCE SOURADNICE Y
    DIFERENCI ROZLISENI
270 GOTO 160
280 SCREEN 1
290 GOSUB 770
300 '
310      '----- INSTRUKCE PRINT -----
320 LOCATE 21, 1
330 PRINT "1-PREDEK 2-TELO 3- ZADEK 4-VSE 0-KONEC";
340 LOCATE 22, 1
350 PRINT "KTEROU CAST ZMENIT";
360 INPUT P
370 LOCATE 21, 1
380 PRINT "          ";
390 PRINT "          ";
400 IF P = 0 THEN 1100
410 LOCATE 21, 1
420 PRINT "VELIKOST H A V PRO ZMENU";
430 INPUT HS, VS
440 LOCATE 21, 1
450 PRINT "          ";
460 '
470      '--- PREPOCET BODU PRO PRISLUSNOU CAST OBRAZKU ---
480      'POKUD SE MENI MERITKO CASTI OBRAZKU, VOLI SE REFE-
490      'RENCFNI BOD VE STREDU AUTA. POKUD SE MENI MERITKO
500      'CELEHO AUTA, MUSI UZIVATEL UVEST BOD, MA BYT
PEVNÝ.
510 IF P = 4 THEN 670      'ZMENA MERITKA VSECH CASTI
520      'ZMENA MERITKA JEDNE CASTI
530 IF P = 1 THEN 560
540 IF P = 2 THEN 590
550 IF P = 3 THEN 620
560 XF = X(1, 2)
570 YF = Y(1, 2)
580 GOTO 640
590 XF = X(2, 2)
600 YF = Y(2, 2)
610 GOTO 640
620 XF = X(3, 2)
630 YF = Y(3, 2)
640 GOSUB 870      'PROVED ZMENU MERITKA

```

```

650 GOSUB 770      'KRESLI
660 GOTO 320
670 LOCATE 21, 1  'ZMENA MERITKA CELEHO AUTA
680 PRINT "ZMENIT OKOLO KTEREHO BODU";
690 INPUT XF, YF
700 FOR P = 1 TO PN    'ZMENA MERITKA KAZDE CASTI
710   GOSUB 870
720 NEXT P
730 GOSUB 770      'KRESLI
740 GOTO 320
750 '
760 '----- PODPROGRAM KRESLENI -----
770   'KRESLI OBRAZEK
780 CLS
790 FOR K = 1 TO PN
800   FOR J = 1 TO N(K) - 1
810     IF X(K,J)<0 OR X(K,J)>XM OR Y(K,J)<0 OR
Y(K,J)>YM THEN 1090
820     LINE(X(K,J), Y(K,J))-(X(K,J+1), Y(K,J+1))
830   NEXT J
840 NEXT K
850 RETURN
860 '----- PODPROGRAM PRO ZMENU MERITKA -----
870   'MERITKO CASTI OBRAZKU
880 FOR J = 1 TO N(P)
890   X(P,J) = X(P,J) * HS + XF * (1 - HS)
900   Y(P,J) = Y(P,J) * VS + YF * (1 - VS)
910 NEXT J
920 RETURN
930 '
940   'PREDEK AUTA
950 DATA 220,90,210,90,210,60,250,65,260,90,250,90
960 DATA 250,100,240,110,230,110,220,100,220,90
970 DATA 230,80,240,80,250,90
980 DATA -1,-1
990   'STRED TELA
1000 DATA 210,90,160,90,160,60,205,60,187,35,160,35
1010 DATA 160,30,190,30,210,60
1020 DATA -1,-1
1030   'ZADEK AUTA
1040 DATA 120,90,160,90,160,60,120,60,135,35,160,35

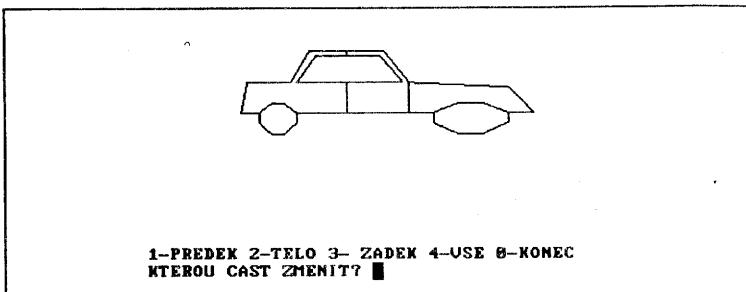
```

```
1050 DATA 160,30,130,30,115,60,80,60,75,90,90,90  
1060 DATA 90,100,100,110,110,110,120,100,120,90,110,80,100  
1065 DATA 80,90,90,90,100  
1070 DATA -100,-100  
1080 -----  
1090 PRINT "SOURADNICE MIMO ROZSAH"  
1100 END
```

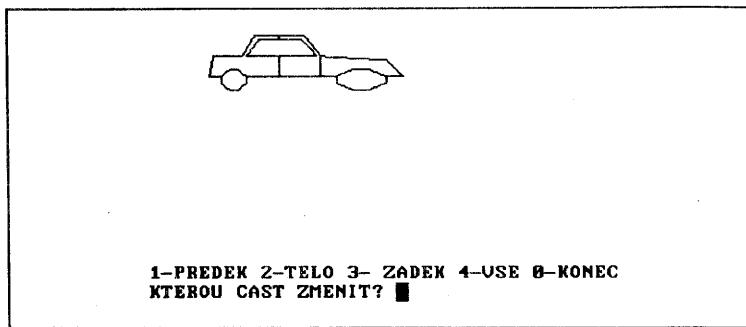
Prog. 6-3 Změny měřítka částí obrazu (auto).



(a)



(b)



(c)

Obr. 6-5 Změna měřítka složek obrázku z původního (a), na obrázek se zvětšenou přední částí (b), a celkové zmenšení (c) programem Prog. 6-3.

U grafů jsou metody, použité pro změnu měřítka, shodné s metodami ilustrovanými v Prog. 6-3. Protože popis nelze měnit (velikost znaků je pevná), musíme se o umístění popisů po změně měřítka rozhodnout podle vzdáleností. To můžeme provést manuálně prohlídkou grafu po změně měřítka nebo můžeme vypočítat nové počáteční pozice pro řetězce

relativně k některému bodu grafu. Jako alternativu bychom mohli vytvořit popis pomocí bodů. Takový popis by mohl měnit měřítka spolu se všemi částmi obrázku.

4.3. Změna orientace (rotace)

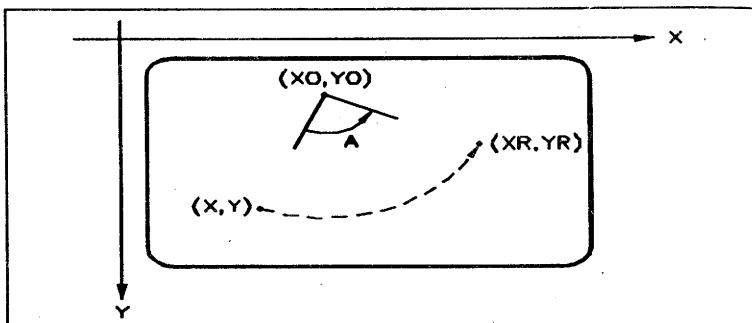
Mnohde bychom rádi měnili orientaci obrázků. Třeba změnili sloupcový graf tak, aby byly sloupce kresleny horizontálně místo vertikálně. V tom případě jednoduše natočíme obraz o 90 stupňů - obejdeme se bez rekonstrukce grafu. Při simulaci a hrách často potřebujeme zobrazovat rotující objekty.

4.3.1 Rotace bodu

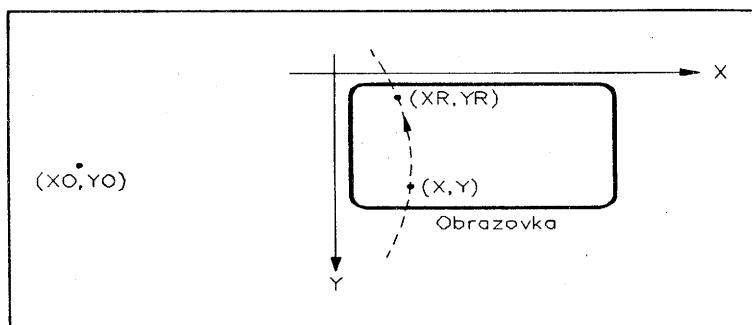
Obr. 6-6 ilustruje rotaci bodu. Dráha rotace je přitom oblouk kružnice z pozice (X,Y) do pozice (XR,YR) okolo středu (osy) (XO,YO). Úhel A v tomto obrázku specifikuje velikost rotace (otočení) z bodu (X,Y) do bodu (XR,YR) jako

$$\begin{aligned} XR &= XO + ((X - XO) * \cos(A)) + ((Y - YO) * \sin(A)) \\ YR &= YO + ((Y - YO) * \cos(A)) - ((X - XO) * \sin(A)) \end{aligned} \quad (6-12)$$

Souřadnice středu rotace (XO,YO) lze volit tak, aby byl umístěn na nějaké vhodné místo buď na stínítku obrazovky, nebo mimo něj v libovolném směru. Tento bod není kreslen a má pouze pomocný význam jako referenční bod určující kruhovou dráhu rotace, jak je uvedeno na Obr. 6-7.

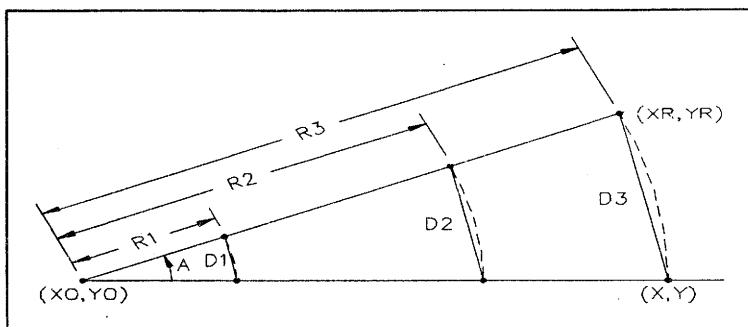


Obr. 6-6 Rotace bodu na pozici (X, Y) do pozice (X_R, Y_R) okolo bodu (X_0, Y_0) . Bod rotuje o úhel A podél kruhové cesty



Obr. 6-7 Souřadnice středu rotace (X_0, Y_0) mohou být voleny mimo stínítko obrazovky, stejně tak jako na stínítku obrazovky.

Úhel rotace A je měřen proti směru otáčení hodinových ručiček od počáteční pozice (X, Y) otáčeného bodu. To předpokládá, že počátek našeho souřadnicového systému je v levém



Obr. 6-8 Pro malé hodnoty úhlu rotace A je vzdálenost D od (X, Y) ke (XR, YR) přibližně rovna vzdálenosti od středu krát úhel rotace A . S rostoucím R roste také D při stejném úhlu A

horním rohu stínítka obrazovky. Pro počítačové systémy, kde je počátek souřadnicového systému v levém dolním rohu stínítka obrazovky, měříme úhel rotace A ve směru otáčení hodinových ručiček. Bude mít hodnotu mezi 0 radiány a $2 * \pi = (6.283185)$ radiány. Mohou být také uvedeny jiné úhly, které však jednoduše opakují rotace v tomto intervalu.

Vzdálenost, o kterou se bod přesouvá po kruhové cestě při určitém úhlu A , závisí na vzdálenosti tohoto bodu (X, Y) od středu rotace (X_0, Y_0) . Čím dále je (X_0, Y_0) od (X, Y) , tím větší je vzdálenost, kterou urazí bod (X, Y) do (X_R, Y_R) . Obr. 6-8 ukazuje vzájemné vztahy mezi vzdáleností R od středu rotace, úhlem A a vzdáleností D od (X, Y) k (X_R, Y_R) . Pro malé úhly rotace je D přibližně rovno součinu $R * A$. Pokud je $R = 50$, mohl by být bod otočen o vzdálenost okolo jedné jednotky (na sousední bod) s úhlem $A = 0.02$ radiánu.

4.3.2 Rotace obrazu

Pro změnu orientace zobrazovaného objektu nejprve zvolíme střed rotace. Pak přepočítáme hodnoty souřadnic

všech bodů tohoto objektu podle vztahů (6-12). Nakonec vymažeme stínítko obrazovky a celý objekt nakreslíme znovu s těmito přepočítanými (natočenými) hodnotami souřadnic všech bodů objektu.

Program 6-4 natáčí obrázek okolo libovolného zvoleného středu rotace (XO,YO) o libovolný specifikovaný úhel natočení A za předpokladu, že tato natočená pozice je na stínítku obrazovky. Výstup tohoto programu pro úhly natočení 90 a 180 stupňů je uveden na Obr. 6-9. Střed rotace je v tomto případě ve středu figurky.

```
10  'PROGRAM 6-4. OTACENI OBRAZKU.
20      'KRESLI OBRAZEK A UMOZNI UZIVATELI UVEST UHEL POZA-
30      'DOVANEHO NATOCENI. NATACENE BODY JSOU ZAPSANY V
40      'POLICH XR A YR.
45  CLS
50  DIM X(7,100), Y(7,100), N(7)
60  DIM XR(7,100), YR(7,100)
70  PRINT "UVED MAXIMALNI HODNOTU HORIZONTALNI A VERTIK."
80  PRINT "SOURADNICE PRO TENTO ROZLISOVACI REZIM"
90  INPUT XM, YM
100 '
110      '----- NACTENI A NAKRESLENI CASTI OBRAZKU -----
120  PN = 1    'PN JE CISLO CASTI
130  E = 0    'E JE PRVEK CASTI
140  READ XD, YD
150  IF XD >= 0 THEN 220  'JE TO NOVY DATOVY BOD ?
160      'NENI-LI, KONCI CAST. ZAPIS POCET PRVKU E DO N(PN)
170  N(PN) = E
180  IF XD = -100 THEN 260 'JE-LI KONEC DAT, NAKRESLI
190      'OBRAZEK, JINAK NOVA CAST. NACTI PRVKY TETO' CASTI.
200  PN = PN + 1
210  GOTO 130
220  E = E + 1
230  X(PN,E) = XD
240  Y(PN,E) = YD * 5 / 6    'KOREKCE SOURADNICE Y NA
        DIFERENCI ROZLISENI
250  GOTO 140
```

```

260      'KOPIROVANI PUVODNICH HODNOT DO POLI UZITYCH PRO
270      'NATOCENE BODY, PROTOZE KRESLICI PODPROGRAM POUZIVA
275      'TATO POLE
280 FOR P = 1 TO PN
290     FOR E = 1 TO N(P)
300       XR(P,E) = X(P,E)
310       YR(P,E) = Y(P,E)
320     NEXT E
330 NEXT P
340 SCREEN 1
350 GOSUB 590  'KRESLENI OBRAZKU
360 '
370      '----- UVEDENI UHLU NATOCENI -----
380 LOCATE 22,1
390 PRINT "UHEL OTOCENI KOLEM POCATECNI POLOHY";
400 LOCATE 23,1
410 PRINT "(0-360, NEBO -1 KONEC)";
420 INPUT A
430 LOCATE 22,1
440 PRINT "
450 PRINT "
460 IF A < 0 THEN 900
470 A = A * 3.14159 / 180   'PREVOD STUPNU NA RADIANY
480 LOCATE 22,1
490 PRINT "KOLEM KTEREHO BODU";
500 INPUT XO, YO
510 '
520      '----- PREPOCET BODU -----
530 FOR P = 1 TO PN      'NATOCENI KAZDE CASTI
540     GOSUB 680
550 NEXT P
560 GOSUB 590
570 GOTO 370
580 '
590 '----- PODPROGRAM KRESLENI -----
600 CLS
610 FOR P = 1 TO PN
620     FOR E = 1 TO N(P) - 1
630       IF XR(P,E)<0 OR XR(P,E)>XM OR YR(P,E)<0 OR
          YR(P,E)>YM THEN 890
640       LINE(XR(P,E),YR(P,E))-(XR(P,E+1),YR(P,E+1))

```

```

650      NEXT E
660 NEXT P
670 RETURN
680 '----- PODPROGRAM PRO NATACENI -----
690 FOR E = 1 TO N(P)
700   XR(P,E) = XO + (X(P,E) - XO) * COS(A) + (Y(P,E) -
    YO) * SIN(A) * 6 / 5
710   YR(P,E) = YO + (Y(P,E) - YO) * COS(A) - (X(P,E) -
    XO) * SIN(A) * 5 / 6
720 NEXT E
730 RETURN
740 '-----
750 DATA 160,60,170,50,170,40,160,30,150,30,140,30,140,50
760 DATA 150,60,178,60,165,63,173,70,160,67,155,78,150,67
770 DATA 140,72,145,65,130,65,150,60,100,70,90,70,90,80,100
780 DATA 80,110,90,130,80,120,140,120,150,130,160,120,170
790 DATA 125,175,140,170,140,140,150,150,155,110,170,150
800 DATA 180,160,180,170,200,170,200,165,190,160,195,150
810 DATA 195,140,175,80,200,30,190,20,190,10,180,10,180,20
820 DATA 170,60
830 DATA -1,-1
840 DATA 147,42,153,42,-1,-1
850 DATA 150,38,150,45,-1,-1
860 DATA 157,42,163,42,-1,-1
870 DATA 160,38,160,45
875 DATA -100,-100
880 '-----
890 PRINT "SOURADNICE MIMO POVOLENY INTERVAL"
900 END

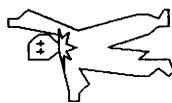
```

Prog. 6-4 Otáčení obrázku (klaun).



UHEL OTOCENI KOLEM POCATEKNI POLOHY
(0-360, MEBO -1 KONEC)? ■

(a)



UHEL OTOCENI KOLEM POCATEKNI POLOHY
(0-360, MEBO -1 KONEC)? ■

(b)



(c)

Obr. 6-9 Natáčení obrázku z původní pozice (a) o 90 stupňů (b) a o 180 stupňů (c) programem Prog. 6-4.

Rozdílné rozlišování souřadnic X a Y mohou způsobit zkreslení (pokřivení) obrázku nebo grafu. V programu Prog. 6-4 je prováděna korekce rozdílu rozlišovacích schopností násobením členu $(Y - Y_0) * \text{SIN}(A)$ na řádku 700 poměrem rozlišení X ku Y a násobením členu $(X - X_0) * \text{SIN}(A)$ na řádku 710 poměrem rozlišení Y ku X. To je ekvivalentní s natočením obrázku podél eliptické cesty, která na stínítku obrazovky vypadá jako kruhová.

4.4. Kombinované transformace

Obecný transformační program pro kombinování posunu, změny měřítka a rotace je uveden v Prog. 6-5. Tento program ilustruje postup pro vytváření sloupcových diagramů při použití transformací. Data jsou vynesena do grafu v levém rohu stínítka a pak je možno volit transformace nebo kód pro ukončení. To umožňuje zkoušet umístění a velikost tak dlouho, dokud nejsme spokojeni s výsledným zobrazením.

Obrázek Obr. 6-10 ukazuje původní graf a jeho konečnou pozici na stínítku.

```
10 'PROGRAM 6-5. TRANSFORMACE SLOUPCOVEHO GRAFU.
20 'PREDPOKLADANA VELIKOST STINITKA JE 280x160.
30 'PROMENNY POSET DATOVYCH HODNOT (NEJVYSE 8) JE
40 'NACTEN PRIKAZEM DATA. JE NAKRESLENA OSA A MERITKO
50 'DAT URCENO TAK, ABY VYSLEDNE SOURADNICE X BYLY
60 '10 - 110. VRCHOLY SLOUPCU KAZDEHO GRAFU JSOU ULO-
70 'ZENY DO POLI X A Y. PO VYTVORENI GRAFU LZE POUZIT
80 'LIBOVOLNU KOMBINACI POSUVU, MERITKA A ROTACE.
90 'TRANSFORMOVANE BODY NAHRADI PUVODNI HODNOTY ULOZE-
95 'NE V POLICH X A Y.
100 -----
110 CLS
120 DIM X(8,4), Y(8,4), N(8)
130 '
140 '----- NACTENI HODNOT PRO GRAF -----
150 READ MI, MA 'NACTENI MIN. A MAX. HODNOTY UDAJU
160 RM = (110 - 10) / (MA - MI)
170 READ D 'D JE POSET ODDILU
180 IF D <= 8 THEN 210
190 PRINT "POUZE 8 ODDILU JE POVOLENO"
200 GOTO 980
210 FOR K = 1 TO D
220     READ M
230     N(K) = 4
240     X(K,1) = 10
250     Y(K,1) = 15 + (K - 1) * 10 'DALSI SLOUPCE JSOU
      NIZE NA STINITKU
260     X(K,2) = INT((M - MI) * RM + 10 + .5)
270     Y(K,2) = Y(K,1)
280     X(K,3) = X(K,2)
290     Y(K,3) = Y(K,1) + 6
300     X(K,4) = 10
310     Y(K,4) = Y(K,3)
320 NEXT K
330 'ULOZENI BODU PRO OSY GRAFU
340 K = D + 1
350 N(K) = 3
360 X(K,1) = 110
```

```

370 Y(K,1) = 10
380 X(K,2) = 10
390 Y(K,2) = 10
400 X(K,3) = 10
410 Y(K,3) = 15 + 10 * K
420 SCREEN 1
430 GOSUB 610      'KRESLENI GRAFU
440
450 '----- VOLBA TRANSFORMACI -----
460 LOCATE 22,1
470 PRINT "1-POSUN 2-VELIKOST 3-ROTACE 4-KONEC";
480 LOCATE 23,1
490 PRINT "KTERA TRANFORMACE";
500 INPUT T
510 LOCATE 22,1
520 PRINT "
530 PRINT "
540 'VETVENI NA PODPROGRAM ZVOLENE TRANSFORMACE
550 IF T = 1 THEN GOSUB 710
560 IF T = 2 THEN GOSUB 820
570 IF T = 3 THEN GOSUB 1010
580 IF T = 4 THEN 1220
590 GOSUB 610      'KRESLENI
600 GOTO 450
610 '----- PODPROGRAM PRO KRESLENI -----
620 CLS
630 FOR K = 1 TO D + 1
640   FOR J = 1 TO N(K) - 1
650     IF X(K,J) < 0 OR X(K,J) > 279 OR X(K,J+1) < 0
       OR X(K,J+1) > 279 THEN 1200
     IF Y(K,J) < 0 OR Y(K,J) > 159 OR Y(K,J+1) < 0
       OR Y(K,J+1) > 159 THEN 1200
     LINE(X(K,J),Y(K,J))-(X(K,J+1),Y(K,J+1))
660   NEXT J
670 NEXT K
680 RETURN
690
700
710 '----- PODPROGRAM POSUVU -----
720 LOCATE 22,1
730 PRINT "UVED HODNOTY POSUVU H A V";
740 INPUT H, V
750 FOR K = 1 TO D + 1

```

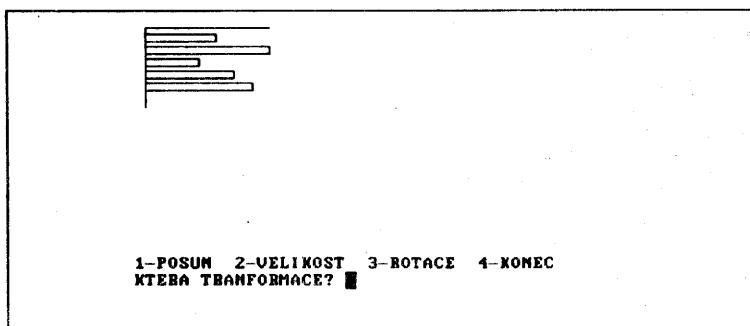
```

760      FOR J = 1 TO N(K)
770          X(K,J) = X(K,J) + H
780          Y(K,J) = Y(K,J) + V
790      NEXT J
800 NEXT K
810 RETURN
820 '----- PODPROGRAM PRO MERITKO -----
830 LOCATE 22,1
840 PRINT "UVED HODNOTY MERITKA H A V";
850 INPUT HS,VS
860 LOCATE 22,1
870 PRINT "
880 LOCATE 22,1
890 PRINT "REFERENCNI BOD";
900 INPUT XF, YF
910     'VYPOCET KONSTANTNI CASTI ROVNIC
920 XC = XF * (1 - HS)
930 YC = YF * (1 - VS)
940 FOR K = 1 TO D + 1
950     FOR J = 1 TO N(K)
960         X(K,J) = X(K,J) * HS + XC
970         Y(K,J) = Y(K,J) * VS + YC
980     NEXT J
990 NEXT K
1000 RETURN
1010 '----- PODPROGRAM PRO ROTACI -----
1020 LOCATE 22,1
1030 PRINT "UVED UHEL NATOCENI";
1040 INPUT A
1050 A = A * 3.14159 / 180 'PREVOD UHLU NATOCENI NA RADIANY
1060 LOCATE 22,1
1070 PRINT "
1080 LOCATE 22,1
1090 PRINT "UVED STRED ROTACE";
1100 INPUT XO, YO
1110 FOR K = 1 TO D + 1
1120     FOR J = 1 TO N(K)
1130         XH(K,J) = X(K,J)    'ZACHOVEJ X(K,J) PRO
        VYPOCET Y
1140         X(K,J) = XO + (X(K,J) - XO) * COS(A) +
        (Y(K,J) - YO) * SIN(A) * 6 / 5

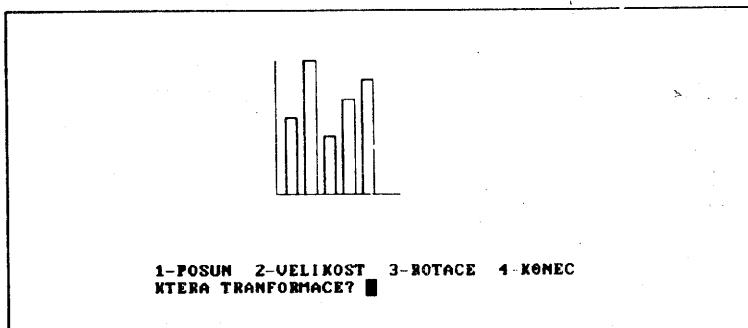
```

```
1150      Y(K,J) = YO + (Y(K,J) - YO) * COS(A) -  
           (XH(K,J) - XO) * SIN(A) * 5 / 6  
1160      NEXT J  
1170 NEXT K  
1180 RETURN  
1190 '-----  
1200 PRINT "SOURADNICE MIMO POVOLENY INTERVAL"  
1210 DATA 0,700,5,400,700,300,500,600  
1220 END
```

Prog. 6-5 kombinované transformace na grafech.



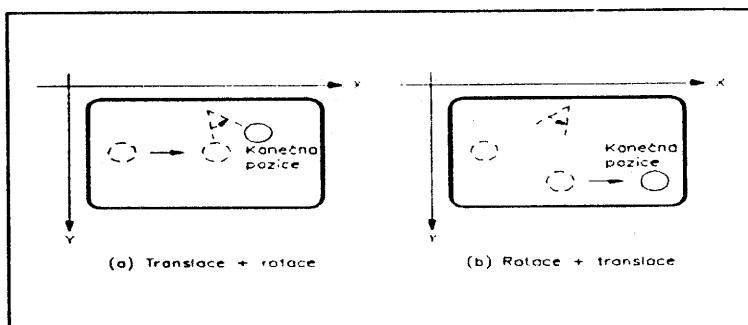
(a)



(b)

Obr. 6-10 Graf původně zobrazený v levém horním rohu obrazovky, jak je uvedeno v části (a) obrázku, byl programem Prog. 6-5 změněn do podoby uvedené v (b) pomocí kombinovaných transformací posuvu, změny měřítka a rotace.

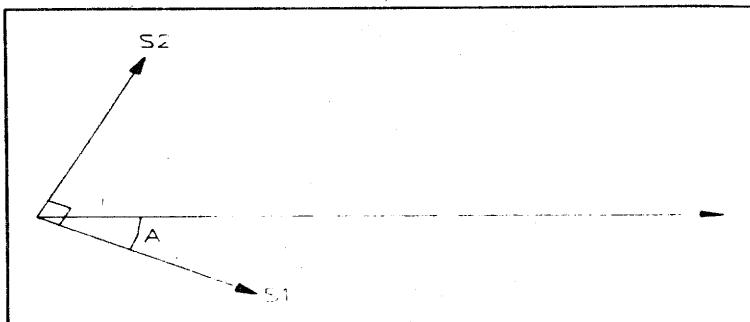
Pořadí, v němž posuv a rotaci provádíme, může ovlivnit pozici zobrazeného objektu. Jak je uvedeno na Obr. 6-11, otáčíme-li nějaký objekt okolo bodu ležícího vně objektu, bude výsledná pozice záviset na tom, zda provedeme nejprve posuv a pak rotaci, nebo naopak.



Obr. 6-11 Konečná pozice objektu po posuvu (translaci) a otočení (rotaci) okolo bodu ležícího mimo tento objekt závisí na pořadí těchto transformací. V případě (a) je provedena jako první translace; v případě (b) dojde k translaci až na druhém místě.

PROJEKTY PROGRAMOVÁNÍ

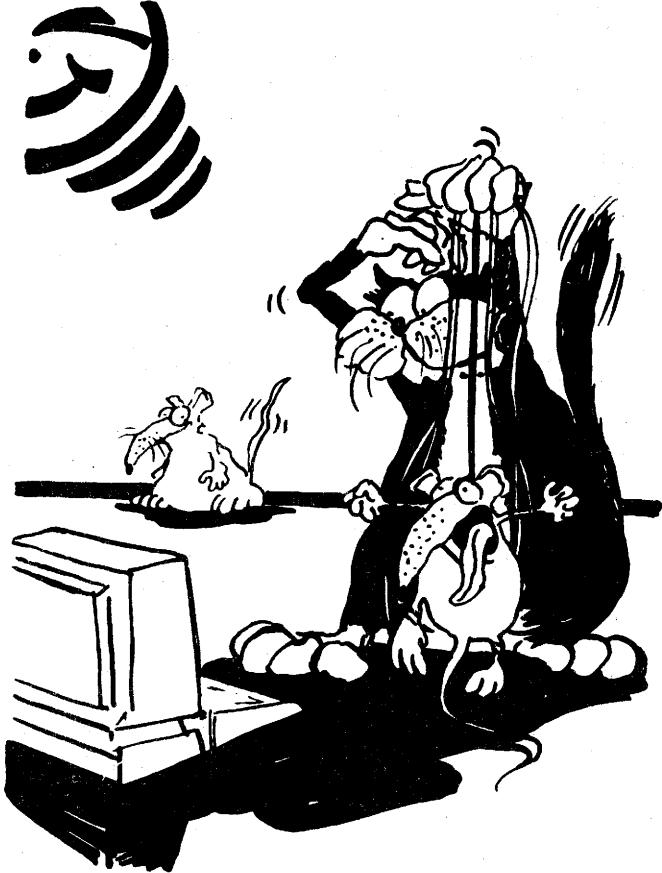
- 6-1. Napište program pro posuv (translaci) kružnice na libovolné místo stínítka obrazovky, zadávané jako vstupní hodnota programu, posunutím středu této kružnice a nakreslením nové kružnice na požadovaném místě.
- 6-2. Upravte program Prog. 6-1 pro změnu měřítka kružnice (relativně k jejímu středu) a vyplnění této kružnice po přesunutí. Parametry pro přesunutí, změnu měřítka a vybarvení kružnice budou zadávány z klávesnice.
- 6-3. Napište program pro přesunutí libovolného znakového řetězce z jednoho místa obrazovky na jiné pomocí metody uvedené v Prog. 6-2. Velikost rastru znaků a vzdálenosti posuvu budou určovány z klávesnice. Žádná část znakového řetězce by neměla být přesunuta mimo stínítko obrazovky.
- 6-4. Napište program, který bude měnit měřítka a zobrazovat libovolný mnohoúhelník. Referenčním bodem bude střed mnohoúhelníka. Souřadnice středu X se vypočítají jako průměrná hodnota souřadnice X všech vrcholů. Souřadnice středu Y se vypočítají jako průměrná hodnota souřadnice Y všech vrcholů.
- 6-5. Napište program pro změnu měřítka figurky v libovolném směru. Směr bude specifikován úhlem A, měřeným od horizontály, jak je uvedeno na Obr. 6-12. Ke změně měřítka dojde po rotaci (pootočení) objektu proti směru hodinových ručiček o úhel A, při současném použití složek měřítka HS = S₁ a VS = S₂, a pak po rotaci objektu zpět o úhel -A do jeho původní pozice. Parametry A, S₁ a S₂ budou zadávány z klávesnice.



Obr. 6-12 Změna měřítka určená úhlem A a složkami S1 a S2.

- 6-6. Upravte Prog. 6-5 tak, aby znakové řetězce popisů byly automaticky přemístěny při změně měřítka grafů. Souřadnice nového počátečního bodu pro umístění popisu určete změnou měřítka vzdálenosti referenčního bodu od původní pozice tohoto počátku.
- 6-7. Napište program, který provede libovolnou transformaci (translaci, změnu měřítka, rotaci) na slově napsaném velkými písmeny (z bodů). Všimněte si, že roztažení písmen podél diagonály (jak bylo uvedeno v Prog. 6-5) zešikmí tato písmena.
- 6-8. Napište program pro obecnou transformaci, který provede translaci, změnu měřítka, nebo rotaci libovolné figurky nebo scény. Figurka nebo scéna může být uložena ve dvourozměrných polích podle návrhu na grafickém papíru. Vstupní parametry určí typ transformace, která má být provedena (budou určovány z klávesnice).

5. ANIMACE



Metody základních transformací (translace, změny měřítka, rotace) můžeme používat i pro animaci zobrazení. Půjde o zobrazení objektu ve výchozí poloze, vymazání tohoto zobrazení, užití transformací a následné zobrazení transformovaného objektu. Opakujeme-li tento postup několikrát, dostaneme pohyb. Tak vznikají kreslené filmy. Umělec nakreslí každé okénko filmového pásu pro kreslený film s drobnými změnami v pozicích a velikostech objektů. Rychlým prohlížením jednoho obrázku po druhém vzniká dojem pohybu objektů.

5.1. Body a kružnice

Uvedené programy pro vytváření pohybujícího se bodu na stínítku obrazovky vznikly úpravou programů uváděných v kapitole 6. Místo jednoho přemístění bodu nastává neustálý pohyb. Bod se může pohybovat po přímkové nebo křívkové dráze.

5.1.1 Přímočará dráha

K jednoduchému příkladu přímočaré animace patří horizontální pohyb bodu přes stínítko obrazovky. Pokud chceme, aby se bod pohyboval zleva doprava, může být počáteční pozice bodu např. (X_1, Y) a koncová pozice (X_2, Y) , kde $X_2 > X_1$. K animaci dochází po sérii translací od X_1 po X_2 , o jednu jednotku za určitou dobu. V každém kroku smažeme dřívější bod (X, Y) a vyneseme nový bod $(X + 1, Y)$. Pohyb vlevo se provede snižováním hodnoty souřadnice X o jednu jednotku v každém kroku. Kombinací těchto dvou pohybů můžeme "odrážet" pohybující se bod zpět a vpřed mezi X_1 a X_2 . Tento pohyb ilustruje program Prog. 7-1. Bod na stínítku obrazovky se nejprve pohybuje vpravo s kladným jednotkovým přírůstkem ($DX = 1$). Jakmile dosáhne bodu (X_2, Y) , obrátíme přírůstek ($DX = -DX$) na negativní, což

způsobí pohyb vlevo. V bodě (X1,Y) opět změníme přírůstek. Program opakuje tento pohyb nepřetržitě.

Program 7-1 může být rozšířen o nakreslení vertikálních hranic na pozicích X1 a X2, takže to vypadá jako by se bod o tyto zdi odrázel. Pak ale musíme změnit body otočení tak, aby se žádná ze zdí při odražení bodu nevymazala. Toho dosáhneme, otočíme-li bod o jednu jednotku dřív, než se dostane ke zdi. Při pohybu bodu zpět a vpřed po přímočaré dráze ve směru osy Y nastavíme vertikální hranice (Y1 a Y2) a provedeme animaci bodu s pevnou hodnotou souřadnice X. Jde-li nám o pohyb v libovolném jiném směru, inkrementujeme obě souřadnice X a Y. Jde o proces podobný kreslení přímky, ovšem s tou výjimkou, že mažeme každý bod před vynesením dalšího bodu. Dráhu pohybu můžeme specifikovat různými způsoby. Můžeme zvolit koncové body této dráhy, můžeme užít rovnici přímky (směrnice a průsečík s osou Y) nebo můžeme zvolit libovolné přírůstky souřadnic X a Y (DX,DY). V prvých dvou případech zvolíme hodnoty DX a DY tak, aby poměr DY/DX byl roven směrnici přímky.

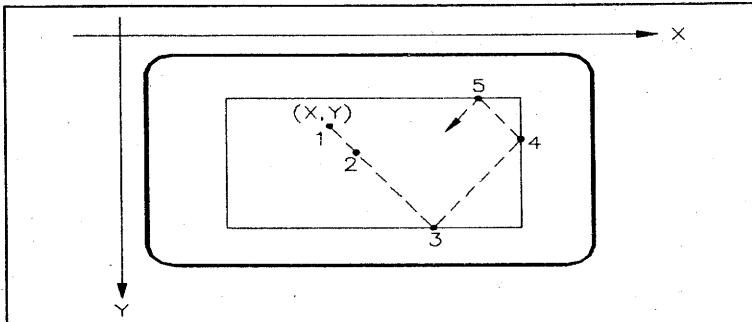
Příklad diagonálního přímočáreho pohybu je uveden v Prog. 7-2. Bod se zde přímočáre pohybuje uvnitř obdélníku a odráží se diagonálně od jeho stran. Po zvolení velikosti obdélníka začíná pohyb bodu diagonálně směrem vpravo dolů po stínítku obrazovky, s přírůstky jedné jednotky pro obě souřadnice. Dosáhne-li strany obdélníka, změní směr, jak je uvedeno na Obr. 7-1. Pokud se pohybující bod odrazí od vertikální strany, změní se znaménko přírůstku X ($DX = -DX$). Pokud se bod odrazí od horizontální strany, dojde ke změně znaménka přírůstku Y ($DY = -DY$). Oba přírůstky změní znaménko v rohu.

Rychlosť pohybu odrážejícího se bodu v Prog. 7-2 můžeme zvětšit volbou větších hodnot přírůstků DX a DY. Zvolíme-li

u přírůstků hodnotu 5 jednotek, pohybuje se bod pětkrát rychleji.

```
10 'PROGRAM 7-1. ODRAZENI BODU MEZI VERTIKALNIMI STENAMI.
20 CLS
30 PRINT "UVED MAXIMALNI HODNOTY HORIZONTALNI A
        VERTIKALNI"
40 PRINT "SOURADNICE PRO TENTO ROZLISOVACI REZIM"
50 INPUT XM, YM
60 PRINT "UVED HODNOTY SOURADNICE X PRO LEVOU A PRAVOU
        STENU"
70 INPUT XL, XR
80 IF XL >= XR OR XL < 0 OR XR > XM THEN 210
90 DX = 1
100 X = XL + INT((XR-XL) / 2)      'POCATECNI BOD JE UPROSTRED
110 Y = INT(YM / 2)                  'MEZI STENAMI
120 CLS
130 SCREEN 1
140 ****
150 PSET(X,Y)
160 IF X > XL AND X < XR THEN 180    'JE-LI MEZI STENAMI
        POKRACUJ
170 DX = -DX                          'JINAK ZMEN ZNAMENKO DX
180 PRESET(X,Y)                      'VYMAZ BOD
190 X = X + DX                        'PRESUN BOD
200 GOTO 150
210 END
```

Prog. 7-1 Horizontální odrazy bodu.



Obr. 7-1 Dráha odrážejícího se bodu uvnitř obdélníka. Počínaje hodnotou souřadnic (X, Y) v pozici 1 bude hodnota souřadnic bodu $(X + DX, Y + DY)$ v pozici 2. V pozici 3 se směr pohybu změní nastavením $DY = -DY$. Směr pohybu se opět změní v pozici 4 změnou $DX = -DX$ a v pozici 5 změnou $DY = -DY$.

```

10  'PROGRAM 7-2. ODRAZENI BODU UVNITR OBDELNIKA.
20      'DX = DY = 1.
30  CLS
40  PRINT "UVED MAXIMALNI HODNOTU HORIZONTALNI A
      VERTIKALNI"
50  PRINT "SOURADNICE PRO TENTO ROZLISOVACI REZIM"
60  INPUT XM, YM
70  PRINT "UVED HODNOTY SOURADNICE X PRO LEVOU A PRAVOU
      STENU"
80  INPUT XL, XR
90  IF XL >= XR OR XL < 0 OR XR > XM THEN 330
100 PRINT "UVED HODNOTY SOURADNICE Y PRO HORNI A DOLNI
      STENU"
110 INPUT YT, YB
120 IF YT >= YB OR YT < 0 OR YB > YM THEN 330
130 DX = 1
140 DY = 1
150 X = XL + INT((XR-XL) / 2)           'POCATECNI BOD
160 Y = YT + INT((YB-YT) / 2)           'UPROSTRED OBDELNIKA
170 CLS
180 SCREEN 1
190      ***** KRESLENI OBDELNIKA
      *****
200 LINE(XL, YT)-(XR, YT)

```

```

210 LINE(XR,YT)-(XR,YB)
220 LINE(XR,YB)-(XL,YB)
230 LINE(XL,YB)-(XL,YT)
240 ***** ODRÁZENÍ BODU ****
***** PSET(X,Y)
250 IF X = XL+1 OR X = XR-1 THEN DX = -DX 'ZBYVA 1 KE
     STENE ?
270 IF Y = YT+1 OR Y = YB-1 THEN DY = -DY 'OTOC ZNAMENKO
280 PRESET(X,Y)      'VYMAZ VYNESENÝ BOD
290 X = X + DX      'VÝPOCET DALŠIHO BODU
300 Y = Y + DY
310 GOTO 250
320
***** PRINT "CHYBA PRI VOLBE STEN OBDELNIKA"
340 END

```

Prog. 7-2 Odrázy bodu uvnitř obdélníku pomocí jednotkových inkrementů.

Zvolíme-li jeden přírůstek větší než než druhý (například $DX > DY$) dojde k rychlejšímu pohybu bodu ve směru většího přírůstku. Alternativně můžeme zpomalit pohyb bodu v jednom směru tím, že hodnoty přírůstku pro tento směr menší než jedna (například 0.5). Změna velikosti přírůstků během zpracování programu může urychlit nebo zpomalit odrážení bodu. Změníme-li velikosti pouze jednoho přírůstku dostaneme při odskoku bodu od stěny efekt smyku (skluzu).

Program 7-2 může být upraven tak, aby pracoval s libovolnou hodnotou obou inkrementů DX nebo DY a se změnou testu odrazu. Pohyb bodu musí být reverzován jakmile by buď souřadnice X nebo Y při inkrementaci dosáhla stěny obdélníka. Tuto změnu můžeme udělat v Prog. 7-2 náhradou řádek 260 - 270 následujícími řádkami:

```

260 IF DX > 0 THEN 268 'SMER VPRAVO
262   'JINAK SMER VLEVO A DX JE NEGATIVNI
264 IF X + DX <= XL THEN DX = - DX

```

```

266 GOTO 269
268 IF X + DX >= XR THEN DX = - DX
269   'TEST Y
270 IF DY > 0 THEN 278      'SMER DOLU
272   'JINAK SMER NAHORU A DY JE NEGATIVNI
274 IF Y + DY <= YT THEN DY = - DY
276 GOTO 280
278 IF Y + DY >= YB THEN DY = - DY

```

Výše uvedený segment programu způsobí, že se bod odrazí dřív, než se dostane ke stěnám obdélníka, při DX nebo DY větším než 1. Abychom dostali realističtější odraz, můžeme vynést bod před změnou směru pohybu těsně vedle stěny. Toho dosáhneme volbou velikosti inkrementu a velikosti obdélníka: vzdálenost napříč obdélníkem v obou směrech bude celočíselným násobkem inkrementu pro tento směr. Toto řešení je pro některé aplikace poněkud omezující. Obecnějším řešením tohoto problému je navržení dráhy bodu ke stěně a určení průsečíku této dráhy se stěnou obdélníka. Pak můžeme odrážit bod od stěn pro libovolnou hodnotu inkrementu.

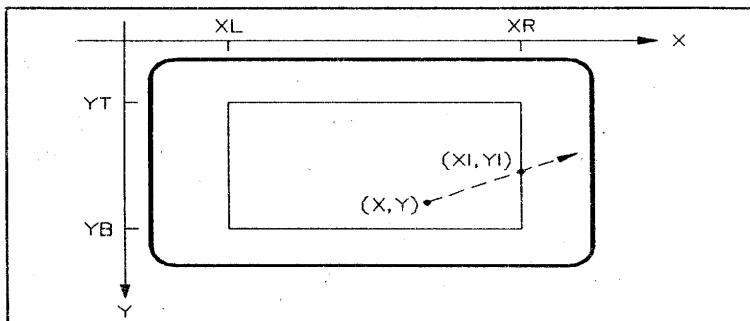
Obrázek 7-2 popisuje dráhu pohybu bodu ve směru k vertikální hranici obdélníka. Při souřadnicích počátečního bodu (X, Y) jsou souřadnice průsečíku (XI, YI) s touto hranicí určeny podle rovnice přímky takto:

$$YI = M \cdot XI + (Y - M \cdot X) \quad (7-1)$$

s hodnotami $XI = XR$ na pravé hranici a $XI = XL$ na levé hranici. Směrnice přímky M se vypočítá z velikostí přírůstků souřadnic podle vztahu

$$M = \frac{DY}{DX} \quad (7-2)$$

Rovnice (7-1) a (7-2) může být použito k určení průsečíků pro libovolnou hranici a libovolný směr pohybu. Pro průsečíky s horní nebo dolní stěnou obdélníka je X_I určené podle vztahu (7-1) jako $X_I = X + (Y_I - Y)/M$, kde $Y_I = Y_T$ nebo $Y_I = Y_B$ (Obr. 7-2). Na obou těchto pozicích odrazu umístíme bod o 1 jednotku blíž k hranici a obrátíme směr pohybu. Aplikace této metody na odrážení míčku je uvedena v Prog. 7-3.



Obr. 7-2 Bod v pozici (X, Y) se pohybuje uvnitř obdélníka po diagonální dráze, která se protíná s pravou hranicí v pozici (X_I, Y_I) .

Přesouvání kružnice je základem mnoha animačních aplikací pracujících s odrázejícím se míčem. Metody užité u bodů lze použít i pro odrážení kružnic: v tom případě považujeme kružnici za velký bod. Přesouváme střed kružnice a testujeme průsečík bodů kružnice s okolními body. Odrážení uvnitř obdélníka zobrazuje program Prog. 7-3, který používá techniky uváděné při odrážení bodů. Míč se odrazí od stěny při zjištění průsečíku dráhy míče se stěnou

obrácením směru pohybu jednu jednotku před dosažením stěny.

```
10 'PROGRAM 7-3. ODRAZENI MICE Uvnitr OBDELNIKA.
20     'DX = DY = 5. POKUD SE MIC BLIZI K HRANICI,
30     'URCI PRUSECIK MICE A HRANICE A PAK MIC NAKRESLI
40     'REVERZUJ SMER POHYBU MICE A POKRACUJ.
50     *****

60 CLS
70 PRINT "UVED MAXIMALNI HODNOTY SOURADNIC X A Y"
80 PRINT "PRO TENTO ROZLISOVACI REZIM"
90 INPUT XM,YM
100 PRINT "UVED HODNOTY X PRO LEVOU A PRAVOU STENU"
110 INPUT XL,XR
120 IF XL >= XR OR XL < 0 OR XR > XM THEN 940
130 PRINT "UVED HODNOTY Y PRO HORNI A DOLNI STENU"
140 INPUT YT,YB
150 IF YT >= YB OR YT < 0 OR YB > YM THEN 940
160 DX = 5      'MIC POSKOCI O 5 JEDNOTEK V KAZDEM KROKU
170 DY = 5
180 R = 3      'R JE POLOMER MICE
190 XN = XL + INT((XR - XL) / 2)  'START MICE JE UPROSTRED
200 YN = YT + INT((YB - YT) / 2)
210 CLS
220 SCREEN 1
230     ***** KRESBA OBDELNIKA *****
240 LINE(XL,YT)-(XR,YT)
250 LINE(XR,YT)-(XR,YB)
260 LINE(XR,YB)-(XL,YB)
270 LINE(XL,YB)-(XL,YT)
280     ***** ODRAZENI MICE *****
290 REM
300 CIRCLE(X,Y),R,0  'VYMAZANI MICE NA PUvodNI POZICI
310 REM
320 CIRCLE(XN,YN),R,1  'KRESBA MICE NA NOVE POZICI
330 X = XN
340 Y = YN
350 BX = 0          'BX A BY JSOU PREPINACE INDIKUJICI
360 BY = 0          'STENU DO KTERE MIC NARAZI
370 M = DY / DX    'M JE SMERNICE DRAHY MICE
380 '
```

```

390      ****
400      'NARAZI DO VERTIKALNI STENY ?
410 IF DX > 0 THEN 480  'POKRACOVANI VPRAVO
420      'JINAK POKRACOVANI VLEVO (DX JE NEGATIVNI)
430 IF X + DX - R > XL THEN 530  'JESTE JSME V OBDELNIKU
440 BX = 1          'NARAZ NA STENU X
450 XN = XL + R + 1    'NOVE X JE VPRAVO OD LEVE STENY
460 GOTO 530
470      -----
480      'POKRACOVANI VPRAVO
490 IF X + DX + R < XR THEN 530  'JSME JESTE V OBDELNIKU
500 BX = 1          'NARAZ NA STENU X
510 XN = XR - R - 1    'NOVE X JE VLEVO OD PRAVE STENY
520      ****
530      'NARAZI DO HORIZONTALNI STENY ?
540 IF DY > 0 THEN 610  'POKRACOVANI DOLU
550      'JINAK POKRACOVANI NAHORU (DY JE NEGATIVNI)
560 IF Y + DY - R > YT THEN 670  'JSME JESTE V OBDELNIKU
570 BY = 1          'NARAZ NA STENU Y
580 YN = YT + R + 1    'NOVE Y JE POD HORNÍ STENOU
590 GOTO 670
600      -----
610      'POKRACOVANI DOLU
620 IF Y + DY + R < YB THEN 670  'JSME JESTE V OBDELNIKU
630 BY = 1          'NARAZ NA STENU Y
640 YN = YB - R - 1    'NOVE Y JE NAD DOLNI STENOU
650      ,
660      ****
670      'ODRÁZ OD ZADNE STENY, STENY X, Y, NEBO OD OBOU ?
680 IF BX = 0 AND BY = 0 THEN 730  'BEZ ODRAZU
690 IF BX = 0 AND BY = 1 THEN 770  'ODRÁZ OD Y
700 IF BX = 1 AND BY = 0 THEN 810  'ODRÁZ OD X
710 IF BX = 1 AND BY = 1 THEN 850  'ODRÁZ V ROHU
720      '(OD OBOU STEN)
730 **** BEZ ODRAZU ****
740 XN = X + DX
750 YN = Y + DY
760 GOTO 290
770 **** ODRAZ OD STENY Y ****
780 XN = (YN - Y) / M + X
790 DY = - DY

```

```

800 GOTO 290
810 *****ODRAZ OD STENY X *****
820 YN = (XN - X) * M + Y
830 DX = - DX
840 GOTO 290
850 *****ODRAZ V ROHU *****
860 'DO KTERE STENY NARAZI DRIVE?
870 IF ABS(XN - X) < ABS(YN - Y) THEN 810 'ODRAZ OD X
880 IF ABS(YN - Y) < ABS(XN - X) THEN 770 'ODRAZ OD Y
890 'MIC JE STEJNE DALEKO OD STEN X A Y V ROHU
900 DX = - DX
910 DY = - DY
920 GOTO 290
930 *****
940 PRINT "CHYBA VYBERU STEN OBDELNIKA"
950 END

```

Prog. 7-3 Odrážení míče uvnitř obdélníka.

V programu Prog. 7-3 by bylo možné použít realističtější míč vyplněním vnitřku kružnice barvou popředí. Program by mohl být také modifikován tak, aby se jako vstupní hodnoty zadávaly hranice obdélníku, velikost a počáteční poloha míče.

5.1.2 Křivkové dráhy

K animaci bodu nebo kružnice podél určené křivkové dráhy dochází změnou poloh podle rovnice této křivky. Pohyb začíná v určené počáteční pozici a končí v určeném koncovém bodu. Bodem můžeme pohybovat po kruhové nebo elliptické dráze pomocí technik pro kreslení kružnic o nichž jsme mluvili v Části 5-1, nebo metod rotace z Části 6-3. Pro plynulý pohyb po kružnici je potřeba, aby pozice bodu byly rozmištěny rovnoměrně podél této dráhy. To znamená, že použijeme trigonometrických rovnic pro výpočet souřadnic těchto pozic se shodnou úhlovou vzdáleností mezi dvěma sousedními pozicemi. Pro rychlý pohyb se používají velké přírůstky úhlu (např. 30 stupňů). Jednoduchými úpravami

docílíme překvapivé efekty. Například plynulou změnou poloměru kružnice vzniká spirálový pohyb. Rotace podél kruhových drah tvoří základ modelování oběžných drah držíc sluneční soustavy nebo strojních součástí.

Pohyb bodu podél paraboly simuluje dráhu předmětu vrženého do vzduchu. Dolet předmětu závisí na počáteční rychlosti, s níž byl vržen do vzduchu a na úhlu tohoto vrhu, měřeného od horizontální (viz Obr. 7-3). Z hodnot rychlosti S a úhlu A vrhu můžeme vypočítat dolet R a maximální výšku HT.

$$R = \frac{S * S * \sin(2 * A)}{G} \quad (7-3)$$

$$HT = \frac{(S * \sin(A))2}{2 * G}$$

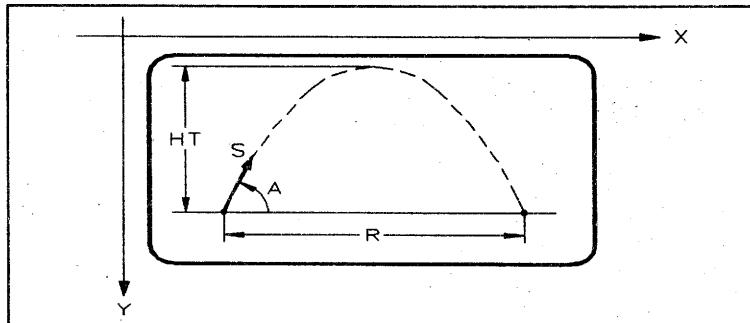
kde G je gravitační zrychlení (980 cm/sec * sec). Pro konstantní hodnotu S dostaneme maximální dolet při úhlu 45 stupňů. Výška HT bude největší bude-li předmět vržen přímo vzhůru (A = 90 stupňů). Pozice bodu podél této křivky se získají plynulou změnou X od 0 do R a výpočtem příslušných hodnot Y podle rovnice

$$Y = C1 * X^2 + C2 * X + Y_0 \quad (7-4)$$

kde Y₀ je libovolná počáteční hodnota zvolená na stínítku obrazovky a konstanty C₁ a C₂ jsou určeny jako

$$C1 = \frac{G}{2 * (S * \cos(A))^2} \quad (7-5)$$

$$C2 = -\tan(A)$$



Obr. 7-3 Předmět vržený do vzduchu s počáteční rychlosí S a pod úhlem A doletí do výšky HT a dopadne ve vzdálosti R od bodu vrhu. Výšku HT a dolet R vypočítáme z hodnot rychlosí S a úhlu A vrhu.

Program Prog. 7-4 animuje bod podél parabolické dráhy. Souřadnice počáteční pozice (X_0, Y_0) jsou zadávány jako vstupní hodnoty. Úhel vrhu A a počáteční rychlosí S jsou také zadávány jako vstupní údaje. Velikost úhlu A je omezena na hodnoty mezi 0 a 90 stupňů, takže pohyb směřuje vpravo (Obr. 7-3). Hodnoty pro S v rozsahu 200 až 600 poskytuje křivky s maximální výškou HT a doletem R okolo 200. Jiné rozsahy pro S mohou být voleny změnou hodnoty G . Můžeme měnit počáteční a koncovou pozici v tomto programu pro simulování jiných podobných typů pohybu. Uvědomme si, že vržení předmětu z vrcholu nějaké budovy nebo kopce znamená, že pohyb začíná a končí s rozdílnými hodnotami souřadnice Y ; pro simulování dráhy předmětu vrženého z letícího letadla, začíná předmět na maximální výšce HT a padá k zemi.

Simulace skákajícího míče je zobrazována Prog. 7-5. Míč je puštěn z určité výšky H a skáče na obrazovce. Každý odraz má poněkud menší výšku než předcházející (Obr. 7-4). Funkce

SIN je užita pro vytvoření kmitajícího pohybu a funkce EXP je použita pro zmenšování amplitudy. Protože chceme, aby pohyb začínal na souřadnicích (0,H), nastavíme D = PI / 2 ve funkci SIN. Dále zvolíme vzdálenost 40 mezi odrazy, takže na obrazovce může být zobrazeno několik odrazů. Toho dosáhneme nastavením W = PI / 40. Konečně chceme zobrazovat míč přesně, když naráží na zem při každém odrazu.

```

10 'PROGRAM 7-4. POHYB PO PARABOLE
20 CLS
30 PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
40 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
50 INPUT XM, YM
60 PRINT "UVED SOURADNICE POCATECNIHO BODU"
70 INPUT XO, YO
80 IF XO>=0 AND XO<=XM AND YO>=0 AND YO<=YM THEN 110
90 PRINT "UVED ZNOVA POCATECNI POZICI"
100 GOTO 60
110 PRINT "UVED UHEL (0 - 90)"
120 INPUT A
130 A = A * 3.14159 / 180      'PREVOD A NA RADIANY
140 PRINT "UVED RYCHLOST (100 - 600)"
150 INPUT S
160 G = 980                      'G JE GRAVIT. ZRYCHLENI
170 R = S * S * SIN(2 * A) / G    'R JE DOLET NA OSĚ X
180 IF XO + R <= XM THEN 210    'VEJDE SE NA OBRAZOVKU ?
190 PRINT "NOVY UHEL, RYCHLOST"   'NEVEJDE SE - NOVE UDAJE
200 GOTO 110
210 HT=((S * SIN(A))^2)/(2 * G) 'HT JE VYSKA KŘIVKY
220 IF HT>0 AND HT<=YM THEN 260 'VEJDE SE NA OBRAZOVKU ?
230 PRINT "NOVY UHEL, RYCHLOST"
240 GOTO 110
250      'VYPOCET KOEFICIENTU ROVNICE
260 C1 = G / (2 * (S * COS(A)) ^ 2)
270 C2 = - TAN(A)
280 CLS
290 SCREEN 1
300 ***** POHYB BODU PO KŘIVCE *****
310 FOR X = 0 TO R

```

```

320      Y = C1 * X * X + C2 * X + YO
330      PSET(X+XO,Y)
340      PRESET(X+XO,Y)
350  NEXT X
360  END

```

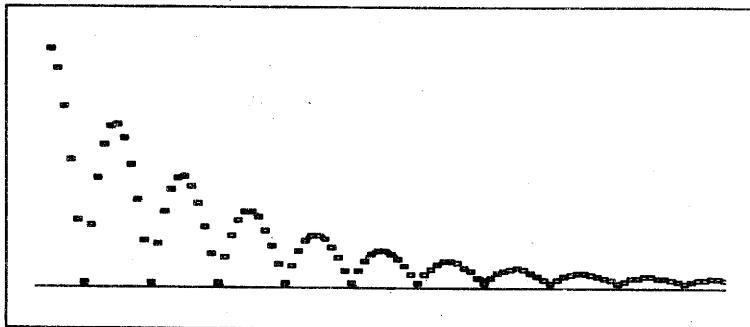
Prog. 7-4 Pohyb bodu po parabolické dráze.

```

10  'PROGRAM 7-5. SKAKANI MICE PADAJICIHO Z URCITE VYSKY.
20      ' PROGRAM SIMULUJE SKAKANI PADAJICIHO MICE.
30  CLS
40  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
50  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
60  INPUT XM,YM
70  PRINT "Z JAKE VYSKY JE MIC PUSTEN";
80  INPUT H
90  W = 3.14159 / 40          'VZDALENOST MEZI ODRAZY JE 40
100 D = 90 * 3.14159 / 180   'POSUV O 90 STUPNU (V RAD.)
110 K = .01                   'K JE CINITEL TLUMENI
120 CLS
130 SCREEN 1
140 LINE(0,YM)-(XM,YM)  'KRESLENI ZEME
150 '
160 ***** PAD MICE A ODRAZ *****
170 FOR XN = 0 TO XM-10 STEP 4
180 YN = H * SIN(W * XN + D) * EXP(-K * XN)
190 YN = YM - ABS(YN) - 3
200 REM
210 CIRCLE(X,Y),2,0          'VYMAZ OKAMZITE POZICE
220 REM
230 CIRCLE(XN+10,YN),2,1  'KRESLENI NOVE POZICE
240 X = XN + 10              'ULOZ POZICI DO X A Y
250 Y = YN
260 NEXT XN
270 END

```

Prog. 7-5 Skákání padajícího míče.



Obr. 7-4 Pohyb odražejícího se míče zleva doprava zobrazený programem Prog. 7-5.

To znamená, že přírůstek X musí být volen tak, aby byl celočíselným dělitelem 40. Pro tento příklad je přírůstek X nastaven na 4 a míč je zobrazen pokaždé, když se odrazí na $X = 20, 60, 100$, atd. Konečné zobrazení je posunuto o 10 bodů doprava, takže pohyb nezačíná na okraji obrazovky.

Kružnice mohou být využívány v animovaných zobrazeních různými způsoby. Translace středu kružnice a rotace paprsků kola simuluují pohybující se kola. Změníme-li měřítka kol, můžeme simulovat pohyb kol směřující k nám nebo od nás. Interaktivní vstup lze užít pro vytváření pohybu podél libovolné dráhy. Pomocí klávesnice, elektronické myši nebo světelného pera potom můžeme řídit objekty v libovolném směru s libovolnou specifikovanou rychlosťí.

5.2. Přímky a mnohoúhelníky

Jednoduchý pohyb přímky nebo mnohoúhelníku na obrazovce se uskutečňuje stejnými základními technikami jako pohyb bodu nebo kružnice. Zobrazíme všechny přímky objektu, objekt vymažeme a znova nakreslíme všechny přímky na nové pozici. Opakováním tohoto postupu vzniká

dojem pohybu objektu. Při animaci objektu používáme další pravidla, jestliže chceme provádět rotaci, změnu měřítka objektu nebo chceme-li pohybovat různými částmi objektu různě.

5.2.1 Přímky

Přímka vymezená body (X,Y1) a (X,Y2) se přesouvá vertikálně, mění-li se souřadnice Y. Program Prog.7-6 zobrazuje vertikální přímku pohybující se nahoru a dolů mezi dvěma pevnými hranicemi. Přímka se pohybuje rychleji nebo pomaleji volbou DY v rozsahu 1 až 20. Tento program nikdy nedovolí přímce dosáhnout nebo překročit hranice nastavené nahoře (YT) a dole (YB).

Vertikální přímkou můžeme pohybovat horizontálně, změníme-li souřadnice X. V tomto případě by mohl pohyb probíhat mezi vertikálními stěnami umístěnými na XL a XR.

```
10 'PROGRAM 7-6. VERTIKALNI POHYB PRIMKY.
20 CLS
30 PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
40 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
50 INPUT XM, YM
60 PRINT "UVED HODNOTY Y PRO HORNI A DOLNI HRANICE"
70 INPUT YT, YB
80 IF YT < YB AND YT >= 0 AND YB <= YM THEN 110
90 PRINT "CHYBNE HRANICE. ZVOL YT < YB A OBE MEZI 0 A ";YM
100 GOTO 60
110 PRINT "UVED POSET PRO ZMENY SOURADNIC Y PRIMKY"
120 INPUT DY
130 X = INT(XM / 2)           'STREDNI PRIMKA OBRAZOVKY
140 Y1 = YT + INT((YB - YT) / 2) - 20 'KONCE JSOU 20 JEDN.
150 Y2 = YT + INT((YB - YT) / 2) + 20 'NAHORU A DOLU
160 CLS
170 SCREEN 1
180 '*****REZIM*****
190 REM
```

```

200 LINE(X,Y1)-(X,Y2),1           'PRIMKA SE POHYBUJE DOLU
210 IF DY > 0 THEN 250          'JINAK SE POHYBUJE NAHORU (DY JE NEGATIVNI)
220   'VYMAZ PRIMKU, VYPOCET NOVE POZICE A KRESLENI
230 IF Y1 + DY <= YT THEN DY = -DY    'ZMENA SMERU
240 GOTO 270
250   'PRIMKA SE POHYBUJE DOLU
260 IF Y2 + DY >= YB THEN DY = -DY    'ZMENA SMERU
270   'VYMAZ PRIMKU, VYPOCET NOVE POZICE A KRESLENI
280 REM
290 LINE(X,Y1)-(X,Y2),0
300 Y1 = Y1 + DY
310 Y2 = Y2 + DY
320 GOTO 180
330 END

```

Prog. 7-6 Vertikální pohyb přímky.

Inkrementace obou souřadnic přesouvá přímku diagonálně. Potom bychom mohli odrážet přímky od čtyř stěn stejně jako v případě jednoho bodu. Horizontální přímky nebo přímky kreslené pod libovolným úhlem se přesunují podobnými metodami. Přesouváme oba koncové body přímky o stejně inkrementy DX a DY. Přímky se pohybují rychleji při větší hodnotě inkrementů a mění směr pohybu, změníme-li znaménko jednoho nebo obou inkrementů.

Změníme-li při pohybu přímek velikost DX a DY, budou se tyto přímky pohybovat po křivkách. Pro pohyb přímky po určité zakřivené dráze vždy udržujeme nějaký určitý bod přímky na křivce dráhy. Např. Prog. 7-7 pohybuje jedním koncem horizontální přímky po kružnici. Přímka se pohybuje vzad, vpřed, nahoru a dolu a simuluje tak pohyb horizontální tyče připojené k rotoru (Obr. 7-5).

Někdy potřebujeme pohyb otáčející se přímky po zakřivené dráze. Obr. 7-6 ukazuje pozice a orientace přímky (se šípkou na konci), jež levý konec se pohybuje po parabolické dráze, a přímka se otáčí tak, že je v každém kroku

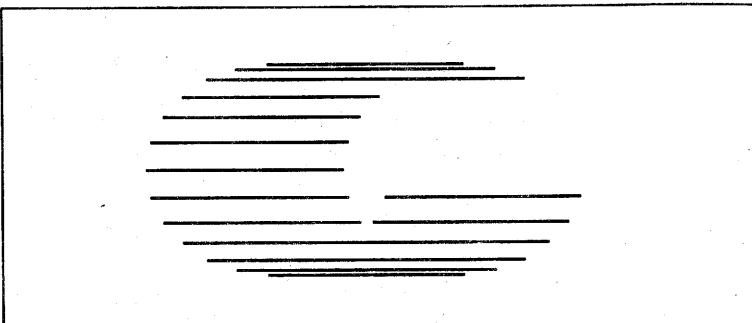
tečnou křivky. Tento pohyb by mohl např. simulovat pohyb šípu vystřeleného do vzduchu. Pro modelování tohoto pohybu musíme měnit směrnici přímky během pohybu. Pro rovnici paraboly (7-4) je směrnice tečny (tangenciální přímky) této křivky počítána z hodnoty souřadnice X pozice:

$$M=2 \cdot C_1 \cdot X + C_2$$

(7-6)

```
10  'PROGRAM 7-7. KRUHOVY POHYB HORIZONTALNI PRIMKY.
20  CLS
30  PRINT "UVED MAXIMALNI HORIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
50  INPUT XM, YM
60  XC = INT(XM / 2) - 40      'XC A YC BUDE STRED KRUZNICE
70  YC = INT(YM / 2)
80  R = 50                      'R JE POLOMER KRUZNICE
90  YA = 5/6                     'YA JE KOREKCE ROZLISOVACI SCHOPNOSTI
100   'VYPOCET BODU X1 PRO KAZDICH 15 STUPNU NA KRUZNICI
110  RE = 360 * 3.14159 / 180   'PREVOD 360 NA RADIANY
120  DA = 15 * 3.14159 / 180   'PREVOD 15 NA RADIANY
130  CLS
140  SCREEN 1
150  ****
160 FOR A = DA TO RE STEP DA
170   REM
180   LINE(X,Y)-(X+80,Y),0  'VYMAZANI PUVODNI CARY
190   X = XC + R * COS(A)
200   Y = YC + R * SIN(A) * YA
210   REM
220   LINE(X,Y)-(X+80,Y),1 'CARA JE 80 JEDNOTEK
230 NEXT A                      'DLOHA
240 GOTO 160
250 END
```

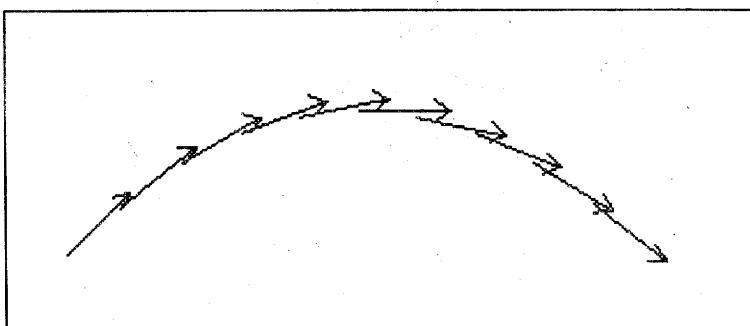
Prog. 7-7 Pohyb přímky po kružnici.



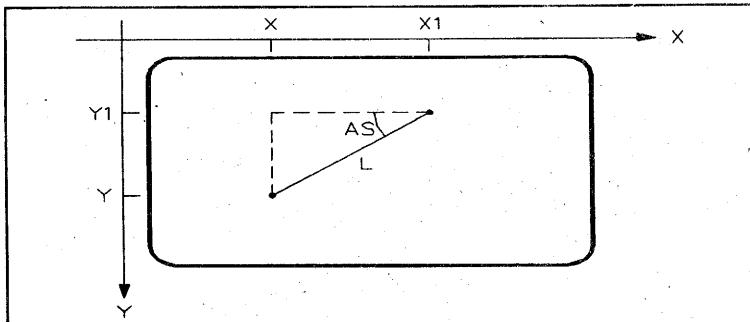
Obr. 7-5 Pozice přímky, jejíž levý koncový bod se pohybuje po kružnici, jako výstup Prog. 7-7. Tato horizontální přímka se začíná pohybovat na pravé straně, pokračuje dolů a doleva a přesouvá se po kruhové dráze zpět do výchozí pozice.

Souřadnice druhého konce přímky (X_1, Y_1) mohou být určeny ze směrového úhlu AS a délky přímky L, jak je naznačeno na Obr. 7-7.

$$\begin{aligned} X_1 - X &= L \cdot \cos(AS) \\ Y_1 - Y &= L \cdot \sin(AS) \end{aligned} \quad (7-7)$$



Obr. 7-6 Pohyb tečny paraboly, jejíž levý koncový bod se pohybuje po této parabole (letící šíp). Obrázek byl zobrazen Prog. 7-8.



Obr. 7-7 Souřadnice druhého konce přímky (X_1, Y_1) mohou být vypočítány z hodnot délky přímky L , směrového úhlu AS a hodnot souřadnic (X, Y) prvého konce přímky. Směrový úhel AS je určen z hodnoty směrnice M jako $AS = ATN(M)$.

Směrový úhel AS je úhel, který svírá přímka s horizontálou. Tento úhel se vypočítá podle směrnice M

$$AS = ATN(M) \quad (7-8)$$

Pohyb šípu na Obr. 7-6 po parabole je simulován programem Prog. 7-8, pomocí rovnic (7-7) a (7-8). V každé pozici přímky se kreslí hrot šípu, který se skládá ze dvou krátkých úseček dotýkajících se v bodu (X_1, Y_1) . Tyto krátké úsečky mají směrnice odvozené ze směrnice paprsku šípu. V tomto programu byla zvolena jedna z těchto směrnic o 0.75 větší než M a druhá o 0.75 menší než M . Délka krátkých úseček je 8 a délka paprsku 40.

```

10 'PROGRAM 7-8. VYSTRELENY SIP NA PARABOLICKE DRAZE.
20 'OPAKOVANE KRESLI A MAZE SIP JEROZ KONEC JE BOD
30 'NA PARABOLE. ZBYVAJICI CAST SIPU JE NAKRESLENA
40 'POMOCI TOHOTO BODU, SMERNICE TECNY PARABOLY V
50 'TOMTO BODE A DELKY SIPU.
60 '
70 ' ****

```

```

80 CLS
90 PRINT "UVEDTE MAXIMALNI HORIZONTALNI A VERTIKALNI"
100 PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
110 INPUT XM, YM
120 PRINT "UVEDTE SOURADNICE POCATECNI POZICE"
130 INPUT XO, YO
140 IF XO>=0 AND XO<=XM AND YO>=0 AND YM<=YM THEN 170
150 PRINT "POCATECNI POZICE MIMO STINITKO"
160 GOTO 120
170 PRINT "UVEDTE UHEL (0 - 90);"
180 INPUT A
190 A = A * 3.14159 / 180 'VYJADRENI UHLU A V RADIANECH
200 PRINT "UVEDTE RYCHLOST (100 - 600);"
210 INPUT S
220 G = 980 'G JE GRAVITACNI SILA
230 LA = 40 'LA JE DELKA SIPKY
240 LT = 8 'LT JE DELKA SPICE SIPKY
250 'NALEZENI ROZSAHU A VYSKY LETU SIPKY
260 R = S * S * SIN(2 * A) / G 'R JE ROZSAH LETU SIPKY
265 'NA OSE X
270 IF XO + R <= XM THEN 300 'VEJDE SE KONCOVY BOD
275 'SIPKY NA STINITKO ?
280 PRINT "UVEDTE ZNOVA UHEL A RYCHLOST"
290 GOTO 170
300 HT = ((S * SIN(A)) ^ 2) / (2 * G) 'HT JE VYSKA SIPKY
310 IF HT > 0 AND HT <= YM THEN 340 'VEJDE SE DRAHA
315 'SIPKY NA STINITKO ?
320 PRINT "UVEDTE ZNOVA UHEL A RYCHLOST"
330 GOTO 170
340 'URCENI KOEFICIENTU ROVNICE PARABOLY
350 C1 = G / (2 * (S * COS(A)) ^ 2)
360 C2 = - TAN(A)
370 CLS
380 SCREEN 1
390 ***** POHYB SIPKY *****
400 'URCENI POLOHY KONCOVEHO BODU SIPKY NA PARABOLE
405 'A NAKRESLENI SIPKY
410 FOR X = 0 TO R STEP R/10 'UMISTENI SIPKY NA
415 'DUJICI DESETINU R
420 Y = C1 * X * X + C2 * X + YO
430 'X A Y JSOU SOURADNICE KONCOVEHO BODU NA PARABOLE

```

```

440      'URCENI DALSIHO KONCOVEHO BODU SIPKY
450      M = 2 * C1 * X + C2 'M JE SMERNICE SIPKY
460      A1 = ATN(M)          'INVERSNI TANGENS M DAVA
465                  'VELIKOST UHLU A1
470      Y1 = Y + LA * SIN(A1)
480      X1 = X + LA * COS(A1)
490      IF X1 > XM OR Y1 > YM THEN 800 'JE DALSI KONCOVY
500      GOSUB 730             'VYMAZANI SIPKY
510      'VYPOCET VRCHOLU SIPKY
520      M2 = M + .75          'SMERNICE JEDNE STRANY SPICE
530      A2 = ATN(M2)
540      X2 = X1 - LT * COS(A2)
550      Y2 = Y1 - LT * SIN(A2)
560      M3 = M - .75          'SMERNICE DRUHE STRANY SPICE
570      A3 = ATN(M3)
580      X3 = X1 - LT * COS(A3)
590      Y3 = Y1 - LT * SIN(A3)
595                  'BOD NA STINITKU ?
600      GOSUB 670             'KRESLENI SIPKY
610      XS = X               'ULOZENI OKAMZITE POLOHY DO
615                  'XS, YS, X1S, Y1S
620      YS = Y
630      X1S = X1
640      Y1S = Y1
650 NEXT X
660 GOTO 800
670 ##### KRESLENI SIPKY #####
680 REM
690 LINE(X,Y)-(X1,Y1),1
700 LINE(X1,Y1)-(X2,Y2),1
710 LINE(X1,Y1)-(X3,Y3),1
720 RETURN
730 ##### VYMAZANI SIPKY #####
740 REM
750 LINE(XS,YS)-(X1S,Y1S),0
760 LINE(X1S,Y1S)-(X2,Y2),0
770 LINE(X1S,Y1S)-(X3,Y3),0
780 RETURN
790 #####
800 END

```

Prog. 7-8 Pohyb šípu po parabolické dráze.

Pro pohyb tangenciální přímky libovolné křivky lze použít metod programu Prog.7-8. Jeden konec úsečky posouváme podél křivky, vypočítáme směrnici v každé poloze, určíme z ní polohový úhel a pomocí rovnice (7-7) vypočítáme souřadnice druhého konce úsečky. Směrnice tangenty křivky v libovolném bodu se určí z rovnice této křivky. Například polynomiální rovnice třetího stupně

$$Y=C_1 \cdot X^3 + C_2 \cdot X^2 + C_3 \cdot X + C_4 \quad (7-9)$$

má směrnici tečny v bodě X danou vztahem:

$$M=3 \cdot C_1 \cdot X^2 + 2 \cdot C_2 \cdot X + C_3 \quad (7-10)$$

Obecná sinusová křivka

$$Y=H \cdot \sin(W \cdot X + D) \quad (7-11)$$

má směrnici tečny v bodě X určenou pomocí funkce COS:

$$M=H \cdot W \cdot \cos(W \cdot X + D) \quad (7-12)$$

Pohyb tangenciální přímky po kružnici nebo elipse lze realizovat pomocí rotačních transformací.

Úsečka může rotovat kolem libovolného pevného bodu s pomocí opakování rotací. Program Prog. 7-9 vytváří přímku rotující kolem svého středu. Tento typ pohybu by mohl být použit např. pro znázornění loukotí otáčejícího se kola. Užité rovnice rotace byly přepsány do tvaru, který redukuje čas výpočtu.

Přesouvat úsečky pro animaci zobrazení několika způsoby. Pohybující se úsečku můžeme používat v hrách ke znázornění pálky nebo rakety odrážející skákající míče. Můžeme ale také různými způsoby přesouvat úsečkové složky obrázku a tak vznikne složitější pohyb, jako např. při simulaci kráčející figurky.

```

10  'PROGRAM 7-9. USECKA ROTUJICI KOLEM SVEHO STREDU.
20  CLS
30  PRINT "UVEDTE MAXIMALNI HORIZONTALNI A VERTIKALNI"
40  PRINT "HODNOTY PRO TENTO ROZLISOVACI REZIM"
50  INPUT XM, YM
60  XC = INT(XM / 2)      'CENTROVANI A ROTACE USECKY
70  YC = INT(YM / 2)      'KOLEM STREDU STINITKA
80  X1 = XC - 10          'USECKA JE DLOUHA 20 JEDNOTEK
90  X2 = XC + 10
100 Y1 = YC
110 Y2 = YC
120 YA = 5/6              'YA JE KOREKCE ROZLISENI PRO Y
130 XA = 6/5              'XA JE KOREKCE ROZLISENI PRO X
140 A = 15 * 3.14159 / 180 'VYJADRENI 15 STUPNU V RADIANECH
150 'VYPOCET KONSTANTNICH CASTI ROVNIC ROTACE
160 CA = COS(A)
170 SX = SIN(A) * XA
180 SY = SIN(A) * YA
190 XE = XC - XC * CA - YC * SX  'URCENI KONSTANTNICH CASTI
200 YE = YC - YC * CA + XC * SY  'ROVNIC ROTACE
210 CLS
220 SCREEN 1
230 '***** ROTACE KONCOVYCH BODU & KRESLENI *****
240 X1R = XE + X1 * CA + Y1 * SX
250 Y1R = YE + Y1 * CA - X1 * SY
260 X2R = XE + X2 * CA + Y2 * SX
270 Y2R = YE + Y2 * CA - X2 * SY
280 'VYMAZANI USECKY VE STAVAJICI POLOZE
290 REM
300 LINE(X1,Y1)-(X2,Y2),0
310 'NAKRESLENI NOVE USECKY
320 REM
330 LINE(X1R,Y1R)-(X2R,Y2R),1

```

```
340      'ULOZENI AKTUALNICH KONCOVYCH BODU PRO POZDEJSI
345      'VYMAZANI USECKY
350  X1 = X1R
360  Y1 = Y1R
370  X2 = X2R
380  Y2 = Y2R
390 GOTO 230
400 END
```

Prog. 7-9 Pohyb rotující úsečky.

5.2.2 Mnohoúhelníky (polygony)

Mnohoúhelník lze určit souřadnicemi vrcholů zaznamenanými v pořadí, v němž mají být propojeny. Přesunout mnohoúhelník znamená přesunout všechny vrcholy a pak mnohoúhelník překreslit. Program Prog. 7-10 přemisťuje obrázek nákladního auta po stínítku zleva doprava, jak je ilustrováno na obrázku Obr. 7-8. Tento program bude přesouvat libovolný objekt definovaný v příkazech DATA. Čím bude více úseček v objektu, tím pomalejší bude výsledná animace. Tento pohyb zobecníme na libovolný směr, použijeme-li příručky pro horizontální i vertikální směr jako v předcházejících příkladech.

Změněná měřítka mnohoúhelníka simulují pohyb směrem k pozorovateli nebo od něj. Plachetnice plovoucí k zapadajícímu slunci na Obr. 7-9, je kreslena programem Prog. 7-11. Měřítko plachetnice je opakovaně zmenšováno vzhledem k bodu na horizontu, což vyvolává dojem zvětšující se vzdálenosti.

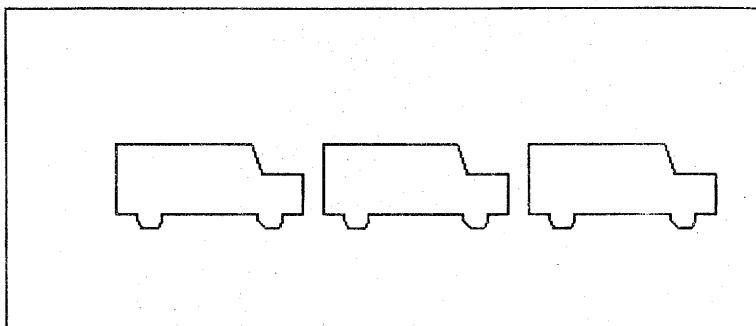
```
10  'PROGRAM 7-10. NAKLADAK POHYBUJICI SE NAPRIC STINITKEM
20  CLS
30  DIM X(20), Y(20), XT(20)
40  XM = 300   'XM JE MAXIMALNI HODNOTA X PRO TENTO SYSTEM
50  READ N      'N JE POSET BODU NA OBRAZKU
60  FOR K = 1 TO N
```

```

70      READ X(K), Y(K)
80      'ULOZENI X DO ALTERNATIVNIHO POLE, KTERE JE UZIVANO
85      'PRO KRESLENI
90      XT(K) = X(K)
100 NEXT K
110 H = 15
120 SCREEN 1
130 GOSUB 260  'KRESLENI
140      ***** TRANSLACE, VYMAZANI, KRESLENI *****
150 FOR K = 1 TO N
160      XT(K) = X(K) + H      'Y SE NEBUDA PRESOUVAT
170 NEXT K
180 GOSUB 320  'VYMAZANI OBRAZKU V AKTUALNI POLOZE
190 GOSUB 260  'NAKRESLENI OBRAZKU V NOVE POLOZE
200 IF XT(14) + H > XM THEN 430 'POKUD PRESUN BODU NEJVICE
210          'V PRAVO PADNE MIMO STI-
215          'NITKO ZASTAVIT, JINAK DAL
220 FOR K = 1 TO N
230      X(K) = XT(K)          'ULOZENI AKTUALNI POZICE X
240 NEXT K
250 GOTO 140
260 ##### PODPROGRAM KRESLENI #####
270 REM
280 FOR K = 1 TO N-1
290      LINE(XT(K),Y(K))-(XT(K+1),Y(K+1)),1
300 NEXT K
310 RETURN
320 ##### PODPROGRAM MAZANI #####
330 REM
340 FOR K = 1 TO N-1
350      LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
360 NEXT K
370 RETURN
380 #####
390 DATA 19
400 DATA 10,90,20,90,20,93,22,97,30,97,32,93,32,90
410 DATA 78,90,78,93,82,97,88,97,90,93,90,90,100,90
420 DATA 100,70,80,70,75,55,10,55,10,90
430 END

```

Prog. 7-10 Pohyb nákladního auta po přímé dráze.



Obr. 7-8 Jedoucí nákladní auto napříč stínítkem, animované programem Prog. 7-10.

```

10  'PROGRAM 7-11. PLACHTE NI K ZAPADAJICIMU SLUNCI.
20      'OPAKOVANE MENI MERITKO LODKY VZHLEDEM K PEVNEMU
30      'BODU (-30,70). ZMENA MERITKA ZMENSUJE LODKU
40      'A PRESOUVA JI ZPRAVA DO LEVA.
50  CLS
60  DIM X(10), Y(10), XN(10), YN(10)
70  YA = 5/6    'YA JE KOREKCE ROZLISOVACI SCHOPNOSTI
80  SCREEN 1
90      ***** VYTVORENI POZADI *****
100 LINE(0,60)-(319,60)      'KRESLENI HORIZONTU
110      'KRESLENI PULKRUZNICE JAKO ZAPADAJICI SLunce
120 R = 25      'R JE POLOMER SLUNCE
130 XS = 40     'STRED SLUNCE JE XS, YS
140 YS = 59
150 DA = 1 / R 'DA JE PRIRUSTEK UHLU
160 C1 = 180 * 3.14159 / 180
170 C2 = 360 * 3.14159 / 180
180 FOR A = C1 TO C2 STEP DA
190      PSET(XS+R*COS(A),YS+R*SIN(A)*YA)
200 NEXT A
210      ***** NACTENI BODU CLUNU A NAKRESLENI *****
220 FOR K = 1 TO 9
230      READ X(K), Y(K)
240      Y(K) = Y(K) * YA
250      XN(K) = X(K)
260      YN(K) = Y(K)
270 NEXT K

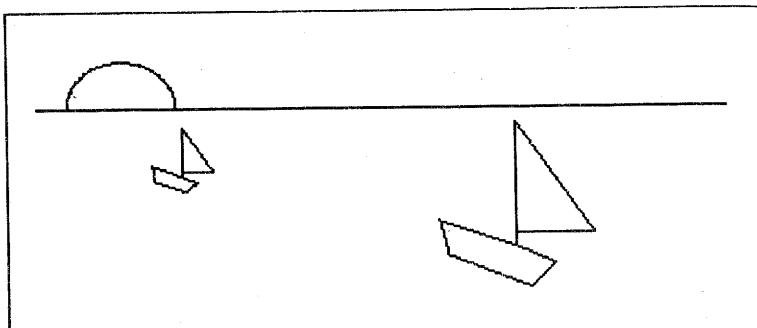
```

```

280 GOSUB 510      'KRESLENI PLACHETNICE
290      ***** PLACHETNI DO ZAPADAJICIHO SLUNCE ****
300 XF = -30        'ZMENA MERITKA VZHLEDEN K XF,YF
310 YF = 70
320 HS = .9         'ZMENA MERITKA LODKY NA 9/10
330 VS = .9
340      'VYPOCET KONST. CASTI ROVNIC PRO ZMENU MERITKA
350 XE = XF * (1 - HS)
360 YE = YF * (1 - VS)
370 FOR T = 1 TO 12
380      FOR K = 1 TO 9
390          XN(K) = X(K) * HS + XE + H
400          YN(K) = Y(K) * VS + YE + V
410      NEXT K
420 GOSUB 570      'VYMAZANI PUVODNI
430 GOSUB 510      'NAKRESLENI V NOVE POLOZE
440 FOR K = 1 TO 9  'ULOZENI AKTUALNI POLOHY DO X A Y
450     X(K) = XN(K)
460     Y(K) = YN(K)
470      NEXT K
480 NEXT T
490 GOTO 650
500 '
510 ##### KRESLENI PLACHETNICE #####
520 REM
530 FOR K = 1 TO 8
540     LINE(XN(K),YN(K))-(XN(K+1),YN(K+1)),1
550 NEXT K
560 RETURN
570 ##### VYMAZANI PLACHETNICE #####
580 REM
590 FOR K = 1 TO 8
600     LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
610 NEXT K
620 RETURN
630 ##### 
640 DATA 250,148,290,148,250,80,250,156,270,167
650 DATA 258,182,215,162,210,140,250,156
660 END

```

Prog. 7-11 Animace změnou měřítka (plachetnice)



Obr. 7-9 Plachtění lodky do zapadajícího slunce. Animace je vytvářena programem Prog. 7-11 změnou měřítka.

5.3. Složený pohyb

Obrazy mohou být animovány přemístěním různých částí obrazu "současně". Tak vznikají vozidla pohybující se v různých směrech, zvířata s pohyblivýma nohami, lidé s pohyblivýma rukama a nohami, elektrické nebo jiné sítě s "tokem" podél několika cest současně nebo složitá zařízení s větším počtem pohyblivých částí. Při animování různých částí je důležitá koordinace délčích pohybů tak, aby vznikl realistický model pohybu.

Program 7-12 kreslí vagón uvedený na Obr. 7-10. Kola vagónu se otáčejí střídavě při přesunu vagónu vlevo. Loukotě se v každé fázi pohybu vždy vymažou a překreslí v pootočené poloze, takže tyto prázdné plochy se zobrazí za minimální dobu. Aby se zkrátila doba výpočtu, používáme minimální počet lounotí. Čím víc úseček má být v obrázku nakresleno, tím delší dobu animace trvá. Kráčející robot je ilustrován programem Prog. 7-13. Zde se translace kombinuje s pohybem nohou. Obrázek 7-11 ukazuje výstup tohoto

programu pro pět poloh pohybu, kdy se robot přesouvá
vpravo.

```
10  'PROGRAM 7-12. POHYB VAGONU A KOL.
20      'KRESLI VAGON A PAK JEJ POSOUVA ZPRAVA DOLEVA.
30      'PAPRSKY KOL SE OTOCI DVAKRAT ZA KAZDY KROK
40      'POSUNU VAGONU.
50  CLS
60  DIM X(20), Y(20), XN(20), YN(20)
70  YA = 5/6      'YA JE KOREKCE ROZLISENI HODNOT Y
80  XA = 6/5      'XA JE KOREKCE ROZLISENI HODNOT X
90  FOR K = 1 TO 18      'NACTENI BODU VAGONU
100     READ X(K), Y(K)
110     Y(K) = Y(K) * YA
120     XN(K) = X(K)
130     YN(K) = Y(K)
140 NEXT K
150 H = -5      'H JE HORIZONTALNI VZDALENOST POSUNU
160 A = 25 * 3.14159 / 180  'VYJADRENI 15 V RADIANECH
170      'VYPOCET KONSTANTNICH CASTI ROVNIC ROTACE
180 CA = COS(A)
190 SX = SIN(A) * XA
200 SY = SIN(A) * YA
210 SCREEN 1
220 GOSUB 360      'KRESLENI VAGONU
230      ***** POSUV, VYMAZANI A NAKRESLENI *****
240 GOSUB 480      'ROTACE PAPRSKU
250 IF XN(2) + H < 0 THEN 940  'KDYZ BY LEVY KRAJNI BOD
260 FOR K = 1 TO 18      'PADL MIMO STINITKO STOP,
265          'JINAK POKRACUJ.
270     X(K) = XN(K)      'ULOZENI AKT. POLOHY DO X A Y
280     Y(K) = YN(K)
290 NEXT K
300 FOR K = 1 TO 18
310     XN(K) = X(K) + H      'PRICTENI VELIKOSTI POSUNU
320 NEXT K
330 GOSUB 770      'VYMAZANI AKTUALNI POLOHY
340 GOSUB 360      'NAKRESLENI NOVE POLOHY
350 GOTO 240
360 ##### PODPROGRAM PRO KRESLENI #####
370 REM
```

```

380 FOR K = 1 TO 5
390     LINE(XN(K),YN(K))-(XN(K+1),YN(K+1)),1
400 NEXT K
410 LINE(XN(7),YN(7))-(XN(8),YN(8)),1
420 CIRCLE(XN(9),YN(9)),11,1 'STREDY KOL JSOU PRVKY POLE
430 CIRCLE(XN(10),YN(10)),11,1
440 FOR K = 11 TO 17 STEP 2   'KRESLENI PAPRSKU
450     LINE(XN(K),YN(K))-(XN(K+1),YN(K+1)),1
460 NEXT K
470 RETURN
480 '***** PODPROGRAM ROTACE PAPRSKU *****
490 YO = Y(9)                  'SOURADNICE Y STREDU OBOU KOL
500 FOR S = 1 TO 2             'PAPRSKY SE OTOCI DVAKRAT
510     FOR K = 11 TO 18         'ULOZENI AKT. POLOHY PAPRSKU
520         X(K) = XN(K)
530         Y(K) = YN(K)
540     NEXT K
550     XO = XN(9)            'SOURADNICE X PRVEHO KOLA
560     FOR K = 11 TO 14
570         XS = XN(K)          'ULOZENI XN(K) PRO VYPOCET YN(K)
580         XN(K) = XO + (XN(K)-XO) * CA + (YN(K)-YO) * SX
590         YN(K) = YO + (YN(K)-YO) * CA - (XS - XO) * SY
600     NEXT K
610     XO = XN(10)           'SOURADNICE X DRUHEHO KOLA
620     FOR K = 15 TO 18
630         XS = XN(K)          'ULOZENI XN(K) PRO VYPOCET YN(K)
640         XN(K) = XO + (XN(K)-XO) * CA + (YN(K)-YO) * SX
650         YN(K) = YO + (YN(K)-YO) * CA - (XS - XO) * SY
660     NEXT K
670 REM
680 FOR K = 11 TO 17 STEP 2   'VYMAZANI AKTUALNI POLOHY
690     LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
700 NEXT K
710 REM
720 FOR K = 11 TO 17 STEP 2   'NAKRESLENI NOVE POLOHY
730     LINE(XN(K),YN(K))-(XN(K+1),YN(K+1)),1
740 NEXT K
750 NEXT S
760 RETURN
770 '***** PODPROGRAM PRO VYMAZANI *****
780 REM

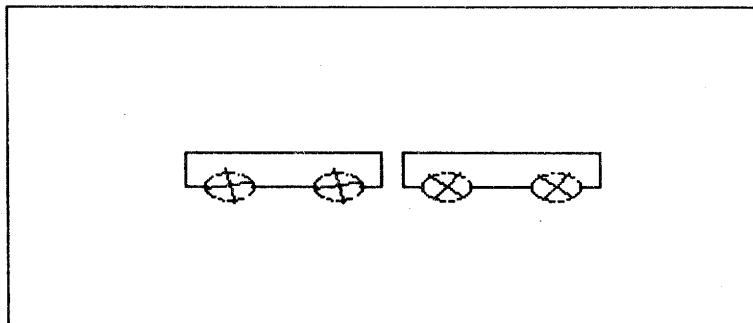
```

```

790 FOR K = 1 TO 5
800     LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
810 NEXT K
820 LINE(X(7),Y(7))-(X(8),Y(8)),0
830 CIRCLE(X(9),Y(9)),11,0
840 CIRCLE(X(10),Y(10)),11,0
850 FOR K = 11 TO 17 STEP 2
860     LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
870 NEXT K
880 RETURN
890 '#####
900 DATA 190,140,180,140,180,120,270,120,270,140,260,140
910 DATA 240,140,210,140,200,140,250,140
920 DATA 210,140,190,140,200,130,200,150
930 DATA 260,140,240,140,250,130,250,150
940 END

```

Prog. 7-12 Složený pohyb: posun vagónu s otáčením kol.



Obr. 7-10 Složený pohyb. Animování vagónu pomocí translace a rotace kol (Prog. 7-12).

```

10 'PROGRAM 7-13. KRACEJICI ROBOT.
20     'ROBOT SE POSUNUJE ZLEVA DOPRAVA PRES STINITKO.
30     'TELO ROBOTA SE POSUNE ZA KADY KROK O STEJNOU
40     'VZDALENOST. KAZDA NOHA JE POSUNUTA O HODNOTU
50     'V ZAVISLOsti NA TOM, ZDA BUDEM ZOBRAZOVAT
60     'KRACEJICI POZICI #1 NEBO #2. V POZICI #1 JSOU
70     'NOHY TESNE U SEBE. V POZICI #2 JSOU OD SEBE.
80 ****

```

```

90  CLS
100 DIM X(16), Y(16), XN(16), YN(16)
110 XM = 279      'XM JE MAXIMALNI HODNOTA X PRO UZITY SYSTEM
120 FOR K = 1 TO 16
130   READ XN(K), YN(K)
140 NEXT K
150 B = 15        'B JE VELIKOST POSUNU TELA PRI KAZDEM KROKU
160 P = 2          'P OZNACUJE ZOBRAZOVANOU KRACEJICI POZICI
170 SCREEN 1
180 GOSUB 510    'KRESLENI
190   ***** TRANSLACE *****
200   'ULOZENI AKTUALNI POLOHY PRO POZDEJSI VYMAZANI
210 FOR K = 1 TO 16
220   X(K) = XN(K)
230   Y(K) = YN(K)
240 NEXT K
250 FOR K = 1 TO 8      'POSUNUTI BODU TELA
260   XN(K) = X(K) + 15
270 NEXT K
280 IF P = 2 THEN 340  'MA JIT NA POZICI #2 ?
290   'JINAK SE ZOBRASI POZICE #1
300 FI = 10           'FI JE INKREMENT PREDNI NOHY
310 RI = 20           'RI JE INKREMENT ZADNI NOHY
320 P = 2             'PRISTE BUDE NASLEDOVAT POZICE #2
330 GOTO 380
340   'JDEME NA POZICI #2
350 FI = 20
360 RI = 10
370 P = 1             'PRISTE BUDE NASLEDOVAT POZICE #1
380   'POSUN ZADNI NOHY
390 FOR K = 9 TO 12
400   XN(K) = X(K) + RI
410 NEXT K
420   'POSUN PREDNI NOHY
430 FOR K = 13 TO 16
440   XN(K) = X(K) + FI
450 NEXT K
460 GOSUB 580        'VYMAZANI AKTUALNI POLOHY
470 GOSUB 510        'NAKRESLENI NOVE POLOHY
480 IF XN(1) + B > XM THEN 680 'PADNE-LI PRAVY KRAJNI BOD
490 GOTO 190          'MIMO STINITKO STOP, JINAK

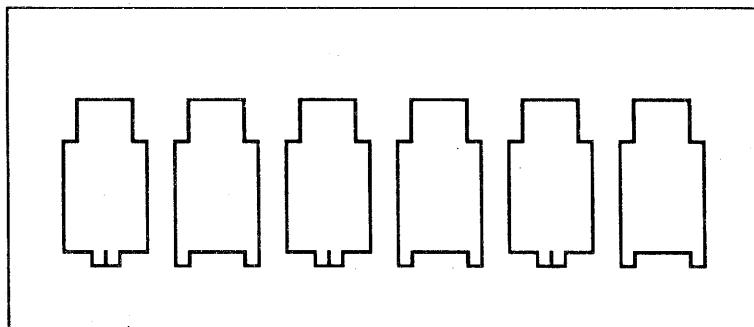
```

```

500          'POKRACOVAT DALE
510 '# ##### PODPROGRAM KRESLENI #####
520 REM
530 FOR K = 1 TO 15
540     LINE(XN(K),YN(K))-(XN(K+1),YN(K+1)),1
550 NEXT K
560 LINE(XN(16),YN(16))-(XN(1),YN(1)),1
570 RETURN
580 '# ##### PODPROGRAM MAZANI #####
590 REM
600 FOR K = 1 TO 15
610     LINE(X(K),Y(K))-(X(K+1),Y(K+1)),0
620 NEXT K
630 LINE(X(16),Y(16))-(X(1),Y(1)),0
640 RETURN
650 '# #####
660 DATA 40,150,40,110,35,110,35,95,15,95,15,110,10,110,10
670 DATA 150,20,150,20,155,25,155,25,150,25,150,25,155,30
680 DATA 155,30,150
690 END

```

Prog. 7-13 Kráčející robot.



Obr. 7-11 Složený pohyb. Robot kráčející přes stínítko obrazovky kreslený programem Prog. 7-13.

V programu Prog. 7-14 zobrazujeme běžící "tyčinkovou" figurku. Každá fáze pohybu byla nakreslena na grafický papír a pro účely reprezentace pohybu rukou a nohou byla

převedena na hodnoty souřadnic na stínítku obrazovky. Dvě fáze pro každou polohu jsou odděleně uloženy v polích. To nám umožňuje zjednodušit posun každé fáze. Tak můžeme realizovat pohyb přes stínítko obrazovky bez přepočítávání relativních pozic částí figurky. Obrázek Obr. 7-12 ilustruje výsledný pohyb

```

10  'PROGRAM 7-14. BEZEC.
20      'ZOBRAZUJE BEZCE VYMENOU DVOU POLOH (NEBO FAZI).
30      'POLE X1 A Y1 UCHOVAVAJI VSECHNY DATOVE BODY PRO
40      'POZICI #1; X2 A Y2 UCHOVAVAJI DATA PRO
50      'POZICI #2.
60      ****
70  CLS
80  DIM X1(15), Y1(15), X2(15), Y2(15)
90  XM = 320      'XM JE MAXIMALNI X PRO UZITY SYSTEM
100 FOR K = 1 TO 13  'NACTENI POZICE #1
110     READ X1(K), Y1(K)
120 NEXT K
130 FOR K = 1 TO 12  'NACTENI POZICE #2
140     READ X2(K), Y2(K)
150 NEXT K
160 XO = 0          'XO JE PRIRUSTEK X
170 SCREEN 1
180 IF XO + X1(12) > XM THEN 800 'PADNE #1 NA STINITKO ?
190 GOSUB 270        'NAKRESLENI POZICE #1
200 GOSUB 390        'VYMAZANI POZICE #1
210 XO = XO + 15    'POSUN O 15 JEDNOTEK
220 IF XO + X2(11) > XM THEN 800 'PADNE #2 NA STINITKO ?
230 GOSUB 510        'NAKRESLENI POZICE #2
240 GOSUB 630        'VYMAZANI POZICE #2
250 XO = XO + 20    'POSUN O 20 JEDNOTEK
260 GOTO 190
270 '***** KRESLENI POZICE #1 *****
280 REM
290 FOR K = 1 TO 3          'NAKRESLENI JEDNE NOHY A TELA
300     LINE(XO + X1(K), Y1(K))-(XO+X1(K+1),Y1(K+1)),1
310 NEXT K
320 LINE(XO + X1(5), Y1(5))-(XO+X1(6),Y1(6)),1  '2. NOHA
330 LINE(XO + X1(6), Y1(6))-(XO+X1(7),Y1(7)),1

```

```

340 FOR K = 8 TO 11      'KRESLENI RUKOU
350     LINE(XO + X1(K), Y1(K))-(XO+X1(K+1),Y1(K+1)),1
360 NEXT K
370 CIRCLE(XO + X1(13), Y1(13)), 10,1  'KRESLENI HLAVY
380 RETURN
390 ##### VYMAZANI POZICE #1 #####
400 REM
410 FOR K = 1 TO 3
420     LINE(XO + X1(K), Y1(K))-(XO+X1(K+1),Y1(K+1)),0
430 NEXT K
440 LINE(XO + X1(5), Y1(5))-(XO+X1(6),Y1(6)),0
450 LINE(XO + X1(6), Y1(6))-(XO+X1(7),Y1(7)),0
460 FOR K = 8 TO 11
470     LINE(XO+X1(K),Y1(K))-(XO+X1(K+1),Y1(K+1)),0
480 NEXT K
490 CIRCLE(XO + X1(13), Y1(13)), 10,0
500 RETURN
510 ##### KRESLENI POZICE #2 #####
520 REM
530 FOR K = 1 TO 2
540     LINE(XO + X2(K), Y2(K))-(XO+X2(K+1),Y2(K+1)),1
550 NEXT K
560 LINE(XO + X2(4), Y2(4))-(XO+X2(5),Y2(5)),1
570 LINE(XO + X2(5), Y2(5))-(XO+X2(6),Y2(6)),1
580 FOR K = 7 TO 10
590     LINE(XO + X2(K), Y2(K))-(XO+X2(K+1),Y2(K+1)),1
600 NEXT K
610 CIRCLE(XO + X2(12), Y2(12)), 10,1
620 RETURN
630 ##### VYMAZANI POZICE #2 #####
640 REM
650 FOR K = 1 TO 2
660     LINE(XO + X2(K), Y2(K))-(XO + X2(K+1),Y2(K+1)),0
670 NEXT K
680 LINE(XO + X2(4), Y2(4))-(XO + X2(5),Y2(5)),0
690 LINE(XO + X2(5), Y2(5))-(XO + X2(6),Y2(6)),0
700 FOR K = 7 TO 10
710     LINE(XO + X2(K), Y2(K))-(XO + X2(K+1),Y2(K+1)),0
720 NEXT K
730 CIRCLE(XO + X2(12), Y2(12)), 10,0
740 RETURN

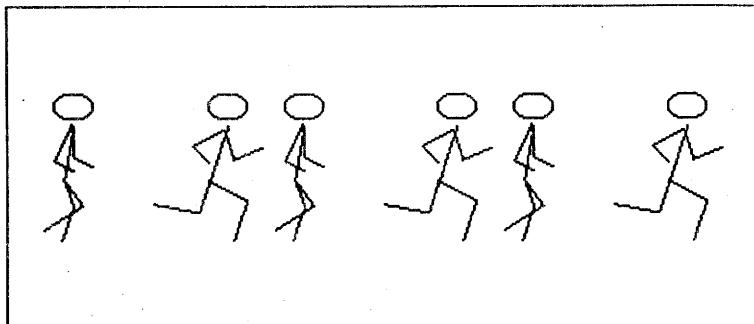
```

```

750 ' #####
760 DATA 14,150,20,133,15,120,20,93,5,145,25,133,15,120
770 DATA 20,115,10,110,19,92,20,108,30,113,20,83
780 DATA 2,132,25,136,40,93,43,150,50,130,30,120
790 DATA 30,111,22,103,38,95,43,110,58,104,40,83
800 END

```

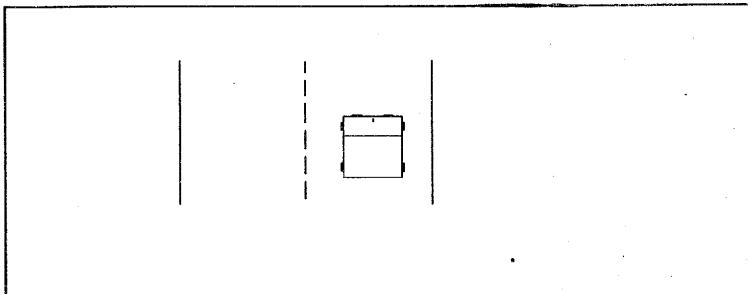
Prog. 7-14 Složený pohyb: běžící "hůlková" figurka kreslená pomocí "fází".



Obr. 7-12 Složený pohyb: dvě fáze užité při zobrazování běžící "hůlkové" figurky programem Prog. 7-14.

5.4. Pohyb pozadí

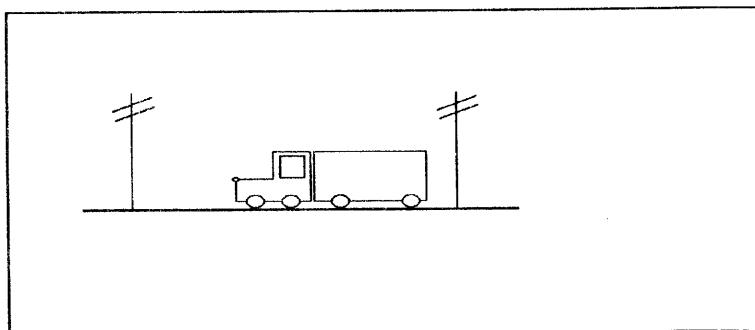
S rostoucím množstvím výpočtů a zobrazovaných úseček se zvětšuje také doba nutná pro animaci scény. U složitých objektů může trvat animace tak dlouho, že pohyb vytvářený na obrazovce přestane být srozumitelný. Urychlení pohybu dosáhneme tak, že animujeme pouze jednoduché útvary, s menším počtem kreslených úseček, anebo animaci omezíme pouze na jednoduché typy pohybu. Další metodou je posun pozadí scény vytvářející dojem pohybu.



Obr. 7-13 Simulace pohybu auta vzhůru po stínítku může být prováděna posunem středních čar silnice směrem dolů.

Obrázek Obr. 7-13 ilustruje jednoduchou formu pohybu pozadí. Pokud posunujeme středovou čáru silnice dolů po stínítku, vzniká dojem, že se auto pohybuje vzhůru. Přemísťování telefonních sloupů na Obr. 7-14 vpravo vyvolá dojem pohybu nákladního auta dovleva. Telefonní sloupy můžeme umístit a posouvat tak, aby nikdy neprotínaly obrázek auta. V opačném případě by se příslušná část obrázku auta vymazala.

Pokud chceme přesouvat pozadí přes nějaký objekt, můžeme modifikovat pozadí tak, že se v tomto místě kreslí pouze část pozadí nebo objektu. Tyto testy a modifikace však opět vedou ke zpomalení animace.



Obr. 7-14 Simulace pohybu nákladního auta vlevo může být znázorněna přesouváním telefonních sloupů doleva.

Existuje mnoho typů pohybu pozadí. Program Prog. 7-15 kombinuje pro simulaci animace složitějšího objektu pohyb pozadí s pohybem jednoduchého objektu. Výsledek je uveden na Obr. 7-15. Rotující ojnice, která propojuje kola lokomotivy a železniční kolej s pražci, simuluje pohyb vlaku. Pozadí se může pohybovat pomaleji nebo rychleji podle velikosti použitých přírůstků polohy objektů pozadí. Pohyb můžeme také zpomalit časovým zpožděním mezi fázemi.

```

10  'PROGRAM 7-15. LOKOMOTIVA S POHYBLIVOU OJNICI A PRAZCI.
20  'NAKRESLIL LOKOMOTIVU A ROTUJE S HORIZONTALNI CAROU
30  'PROPOUJUJICI DVE ZADNI KOLA. TAKE PRESOUVA BODY
40  'POD LOKOMOTIVOU SIMULUJICI PRAZCE.
50  '*****
60  CLS
70  YA = 5/6   'YA JE KOREKCE ROZLISENI
80  SCREEN 1
90  'KRESLENI LOKOMOTIVY
100 READ X1,Y1
110 FOR K = 1 TO 25
120   READ X2,Y2
130   LINE(X1,Y1)-(X2,Y2)
140   X1 = X2
150   Y1 = Y2

```

```

160 NEXT K
170   'DOPLNENI OKNA
180 READ XL,XR,YT,YB
190 FOR X = XL TO XR
200   LINE(X,YT)-(X,YB)
210 NEXT X
220   'DOPLNENI DETAILU
230 FOR K = 1 TO 6
240   READ X1,Y1,X2,Y2
250   LINE(X1,Y1)-(X2,Y2)  'POSLEDNI HODNOTY - OJNICE
260 NEXT K
270   'DOPLNENI KOL
280 FOR K = 1 TO 4
290   READ XC,YC,R
300   CIRCLE(XC,YC),R      'POSLEDNI XC,YC-ZADNI KOLO
310 NEXT K
320 '
330   ***** ROTACE OJNICE, POSUN PRAZCU *****
340 Y = 152      'POLOHA PRAZCU Y
350 R = 15       'R JE POLOMER ROTACE OJNICE
360 RE = 360 * 3.14159 / 180 'PREVOD 360 A 50 NA RADIANY
370 DA = 50 * 3.14159 / 180
380 FOR A = DA TO RE STEP DA
390   X1 = XC + R * SIN(A)
400   Y1 = YC + R * COS(A) * YA
410   REM
420   LINE(X1,Y1)-(X1-90,Y1),1 'KRESLENI NOVE POLOHY
425                           'OJNICE (90 JED. DLOUHE)
430   FOR X = XS TO 319 STEP 35 'VYNESENI PRAZCU VZD. 35
440     PSET(X,Y):PSET(X+1,Y):PSET(X+2,Y)
450   NEXT X
460   FOR X = XS TO 319 STEP 35 'VYMAZANI PRAZCU
470     PRESET(X,Y):PRESET(X+1,Y):PRESET(X+2,Y)
480   NEXT X
490   XS = XS + 7 'DALSI MNOZINA BODU POSUNUTA O 7 PIXELU
500   IF XS >= 30 THEN XS = 0 'ZMENA POLOHY 1. PRAZCE
510   REM
520   LINE(X1,Y1)-(X1-90,Y1),0 'VYMAZANI OJNICE
530 NEXT A
540 GOTO 380
550 #####

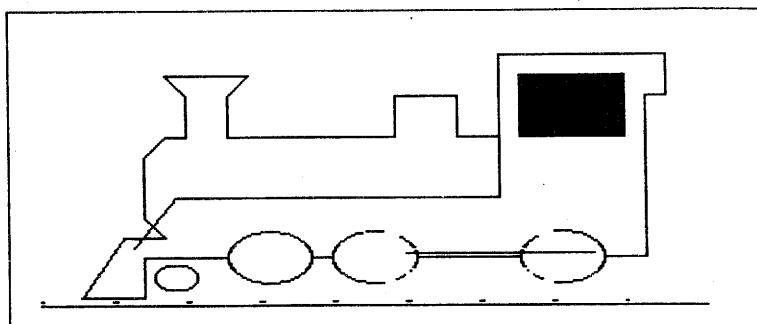
```

```

560      'OBRYS LOKOMOTIVY
570 DATA 270,130,290,130,290,50,300,50,300,30,220,30,220,70
580 DATA 200,70,200,50,170,50,170,70,90,70,90,50,100,40,60
590 DATA 40,70,50,70,70,60,70,50,80,50,110,60,120,40,120,20
600 DATA 150,50,150,50,130,90,130
610      'OKNO
620 DATA 230,280,40,70
630      'DETALY
640 DATA 220,70,220,100,220,100,65,100,65,100,45,125,130
650 DATA 130,140,130,0,154,319,154,180,130,230,130
660      'KOLA
670 DATA 65,140,10,110,130,20,160,130,20,250,130,20
680 END

```

Prog. 7-15 Simulace pohybu s pohybem pozadí: vlak s pohyblivou ojnici a posunem pražců.



Obr. 7-15 Simulace pohybu lokomotivy rotací ojnice a posunem pražců tvořená Prog. 7-15.

PROJEKTY PROGRAMOVÁNÍ

- 7-1. Napište program pro animaci bodu (nebo kružnice) mezi několika vertikálními a horizontálními hranicemi. Poloha a délka každé hranice může být volena náhodně. Posunujte bod o jednotkové přírůstky v horizontálním i vertikálním směru tak, aby se pohyboval po diagonále. Jakmile tento bod dosáhne vertikální hranice nebo okraje stínítka, změňte přírůstek X na opačný. Narazí-li tento bod na horizontální stěnu nebo okraj stínítka, reverzujte přírůstek Y. Pro určení okamžiku, kdy tento bod narazí na jednu z náhodně umístěných úseček, užijte funkci, která testuje polohu bodu o jeden bod dříve před posunem.
- 7-2. Modifikujte program z Prog. 7-1 tak, aby zobrazoval bod (nebo míč) v různých barvách, jak se odráží od okrajů stínítka. Dalšími možnými modifikacemi jsou změny barev, pouze když se bod odrazí, bod se však nevymaže (takže se zobrazuje barevný vzor), nebo zvuk, jakmile se bod odrazí od hranice.
- 7-3. Modifikujte program z Prog. 7-1 tak, aby přesouval bod rychleji nastavením horizontálních a vertikálních přírůstků na nějakou hodnotu větší než 1 (při shodné velikosti obou přírůstků). Každá bodová pozice na dráze od okamžité polohy do možné koncové polohy musí být testována, aby se určilo, zda existuje stěna na dráze pohybu 7-4. Napište program pro odrážení bodu (nebo kroužku) tam a zpět mezi dvěma určenými body (X_1, Y_1) a (X_2, Y_2). Přírůstky pro pohyb (DX a DY) volíme tak, že poměr DX/DY je roven směrnici dráhy pohybu. Toho se dosáhne dělením intervalů X_2-X_1 a Y_2-Y_1 např. deseti.
- 7-5. Napište program pro vytvoření bludiště uvnitř obdélníka z Prog. 7-3. Umístěte kružnice do nějaké počáteční polohy a umožněte interaktivní zadání směru pohybu (např. z

klávesnice). Programově přemísťuje kružnici po jednotkových krocích se změnou směru pohybu podle vstupu z klávesnice, nebo narazí-li bod na stěnu. Náraz může být určen pomocí funkce, která indikuje, zda je bod na stínítku nebo mimo stínítko obrazovky. Pohyb ustane, když kružnice dosáhne východu z bludiště. Pro vytvoření hry z tohoto programu odečtěte body za nárazy na stěny a vytiskněte výsledný počet bodů (skóre).

- 7-6. Modifikujte Prog. 7-3 tak, aby přírůstkům DX a DY mohly být přiřazeny libovolné počáteční hodnoty ($DX < > DY$). Dále zvětšete rychlosť pohybu zvětšením přírůstků o 5 po každém třetím odrazu od levé stěny. Pohyb může ustat, přesáhne-li jeden z přírůstků hodnotu 40.
- 7-7. Napište program, který zachytí pohyb míče (nebo jiného objektu) přes stínítko obrazovky po dráze určené libovolnou specifikovanou křivkou. Touto křivkou může být funkce SIN, polynom čtvrtého stupně nebo libovolná jiná rovnice, která simuluje pohyb nahoru a dolů.
- 7-8. Napište program pro zobrazení bodu (nebo míče), který se pohybuje po spirálové dráze se středem stínítkka. Změnami barev a nevymazáváním bodu může být stínítko obrazovky vyplňeno barevnou spirálou. Obrácením pohybu, dosáhne-li bod okraje stínítkka vzniká opakováný pohyb bodu po spirále směrem dovnitř a ven.
- 7-9. Napište program pro animaci kružnice kreslené uprostřed stínítkka změnou měřítka kružnice, dokud nedosáhne okrajů stínítkka, a pak reverzuje změnu měřítka. Pokračujte v opakování tohoto procesu se změnami barev, když se kružnice zvětšuje a zmenšuje.

- 7-10. Napište program, který simuluje pohyb objektu pohybem pozadí. Příklady jsou vlak nebo zaoceánská loď, které vypouští kouř nebo scény z Obr. 7-13 nebo Obr. 7-14.
- 7-11. Napište program pro zobrazení číselníku hodin uprostřed stínítka. Nakreslete ručičky (hodinovou a minutovou) jako šipky a otáčeje jimi tak, aby se menší ručička otočila o 30 stupňů na každou celou otáčku velké ručičky (360 stupňů). Dále můžete doplněnit o sekundovou ručičku, která se otočí o 360 stupňů na každých 6 stupňů otáčky minutové ručičky.
- 7-12. Napište program pro animaci tangenciální přímky pohybující se po následujících křivkách: a) elipse b) polynomu třetího stupně c) sinusoidě
- 7-13. Napište program simulující let letadla nebo kosmické lodě létající po kružnici

()

()

6. OKÉNKA NA displeji

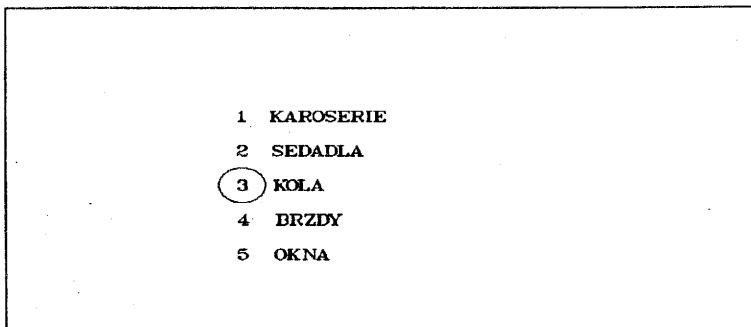


Grafická zobrazení mohou být upravována mnoha způsoby. Prozkoumali jsme metody změny velikosti a přesouvání figurek z jedné polohy nebo orientace do jiné. Těchto transformací lze použít na celé zobrazení nebo na jeho části. Tyto oblasti mohou být "prosvíceny" pro zvláštní zdůraznění, mohou být vymazány se zobrazením, nebo mohou být ponechány a transformovány zatím co zbývající část zobrazení je vymazána.

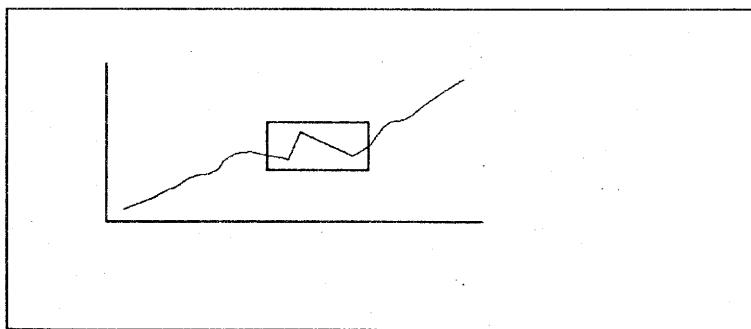
6.1. Zvýrazňování

Někdy potřebujeme zdůraznit jen určitou část obrázku: překreslíme ji jasnou barvou nebo prosvětlíme zvýšením jasu. Na některých systémech je k dispozici volba "vysoký jas". Použijeme-li ji, je text nebo grafika jasnější než ostatní místa na stínítku. Podobného efektu lze dosáhnout kreslením čar s dvojnásobnou tloušťkou a to tak, že místo jedné čary nakreslíme dvě čary o jednu jednotku vedle sebe. Další technikou pro zdůraznění oblasti stínítka je blikání části obrázku. Na určenou oblast můžeme také upozornit tak, že kolem ní nakreslíme kružnici nebo obdélník, jak je uvedeno na Obr. 8-1.

Pro přikreslení takové kružnice jednoduše zvolíme souřadnice středu (XC, YC) a poloměr R . Dále doplníme do našeho programu podprogram nebo příkaz pro kreslení kružnic, který je vyvolán pokaždé, přejeme-li si zvýraznit nějakou oblast. Tak můžeme psát programy, které nám umožní experimentovat s různými polohami a velikostmi kružnice, abychom dosáhli přesně požadovaného efektu zdůraznění, jako v Prog. 8-1.



(a)



(b)

Obr. 8-1 Zvýrazňování displejů pomocí kružnice (a) a obdélníka (b).

```
10 'PROGRAM 8-1. ZVYRAZNENI KRUZNICEMI.  
20 CLS  
30 DIM X(15), Y(15)  
40 XM = 299  
50 YM = 199  
60 '***** NACTENI DAT PRO ZOBRAZENI *****  
70 '  
80 '
```

```

90      '
100     '
110     '
120 GOSUB 160      'KRESLENI ZOBRAZENI,
130 GOSUB 250      'TVORBA ZVYRAZNUJICI KRUZNICE,
140 GOTO 600
150     '
160 ##### PODPROGRAM KRESLENI #####
170 CLS
180 SCREEN 1
190     '
200     '
210     '
220     '
230 RETURN
240     '
250 ##### DOPLNENI ZVYRAZNUJICI KRUZNICE #####
260 LOCATE 2,1
270 PRINT "STRED A POLOMER KRUZNICE";
280 INPUT XC, YC, R
290 IF XC+R<=XM AND XC-R>=0 AND YC-R>=0 THEN 340
300 LOCATE 1,1
310 PRINT "KRUZNICE MIMO STINITKO"
320 PRINT "
330 GOTO 260
340 CIRCLE(XC,YC),R,1
350 LOCATE 1,1
360 PRINT "
370 LOCATE 2,1
380 PRINT "
390 LOCATE 2,1
400 PRINT "UVED E=KONEC, C=ZMENU KRUZNICE";
410 INPUT M$
420 LOCATE 2,1
430 PRINT "
440 IF M$ = "E" THEN 580
450   'VYMAZANI PUVODNI KRUZNICE
460 REM
470 CIRCLE(XC,YC),R,0
480 REM
490   'POSKODILO VYMAZANI PUVODNI ZOBRAZENI ?

```

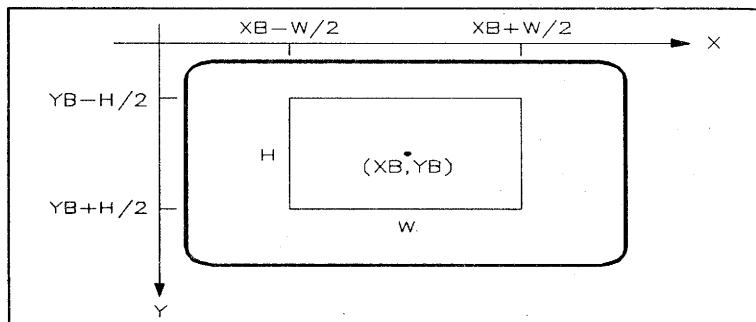
```

500 LOCATE 2,1
510 PRINT "CHCES ZNOVA NAKRESLIT OBRAZEK (A/N)";
520 INPUT D$
530 LOCATE 2,1
540 PRINT "
550 IF D$ = "N" THEN 250
560 GOSUB 160
570 GOTO 250
580 RETURN
590 '#####
600 END

```

Prog. 8-1 Zvýrazňování kružnicemi.

Pro zvýraznění obdélníkem můžeme specifikovat čtyři vrcholy nebo dva protilehlé vrcholy na úhlopříčce pravoúhelníka. Také bychom mohli odvodit podprogram, který se nás jednoduše ptá, kam má umístit střed pravoúhelníka a jak velký má být (Obr. 8-2). Tento typ podprogramu je ilustrován v Prog. 8-2: ten kreslí obdélník se středem v bodu (XB, YB) o šířce W a výšce H .



Obr. 8-2 Souřadnice vrcholů pravoúhelníka mohou být určeny na základě hodnot šířky W , výšky H a souřadnic středu (XB, YB) .

```

10 'PROGRAM 8-2. ZVYRAZNE NI OBDELNIKEM.
20 CLS
30 DIM X(15), Y(15)

```

```

40 XM = 279
50 YM = 159
60 ***** NACTENI DAT PRO ZOBRAZENI *****
70 '
80 '
90 '
100 '
110 '
120 GOSUB 160      'KRESLENI ZOBRAZENI
130 GOSUB 260      'TVORBA OBDELNIKOVEHO ZVYRAZNENI
140 GOTO 770
150 '
160 ##### PODPROGRAM KRESLENI #####
170 CLS
180 SCREEN 1
190 '
200 '
210 '
220 '
230 '
240 RETURN
250 '
260 ##### DOPLNENI ZVYRAZNUJICIHO OBDELNIKA #####
270 LOCATE 2,1
280 PRINT "STRED OBDELNIKA (X A Y)";
290 INPUT XB, YB
300 LOCATE 2,1
310 PRINT "
320 LOCATE 2,1
330 PRINT "SIRKA A VYSKA OBDELNIKA (W A H)";
340 INPUT BW, BH
350 'URCENI STRAN OBDELNIKA (LEVA, PRAVA, HORNI A DOLNI)
360 L = XB - BW / 2
370 R = XB + BW / 2
380 T = YB - BH / 2
390 B = YB + BH / 2
400 IF L<R AND L>=0 AND R<=XM AND T<B AND T>=0 AND B<=YM
     THEN 450
410 LOCATE 1,1
420 PRINT "OBDELNIK MIMO OBRAZOVKU"
430 PRINT "

```

```

440 GOTO 270
450 LINE(L,T)-(L,B)
460 LINE(L,B)-(R,B)
470 LINE(R,B)-(R,T)
480 LINE(R,T)-(L,T)
490 LOCATE 1,1
500 PRINT "
510 LOCATE 2,1
520 PRINT "
530 LOCATE 2,1
540 PRINT "UVĚDE PRO KONEC, C PRO ZMENU OBDELNIKA";
550 INPUT M$
560 LOCATE 2,1
570 PRINT "
580 IF M$ = "E" THEN 750
590   'VYMAZANI PUVODNIHO OBDELNIKA
600 REM
610 LINE(L,T)-(L,B) ,0
620 LINE(L,B)-(R,B) ,0
630 LINE(R,B)-(R,T) ,0
640 LINE(R,T)-(L,T) ,0
650 REM
660   'POSKODILO VYMAZANI PUVODNI ZOBRAZENI ?
670 LOCATE 2,1
680 PRINT "CHCES ZNOVA NAKRESLIT OBRAZEK (A/N)";
690 INPUT D$
700 LOCATE 2,1
710 PRINT "
720 IF D$ = "N" THEN 260
730 GOSUB 160
740 GOTO 260
750 RETURN
760 *****
770 END

```

Prog. 8-2. Zvýraznění pomocí obdélníků.

Zvýrazňování pomocí kružnic nebo obdélníků používáme, chceme-li zdůraznit část zobrazení, které doprovázejí zprávy nebo ukázky. Obrázek se zvýrazní na různých místech můžeme při každé změně polohy zvýraznění vytisknout i na

tiskárně . Tištěná zobrazení pak mohou být zařazena do zprávy. Můžeme také vytvořit pohybující se zvýraznění doprovázející určitou ukázku. Abychom mohli o diskutované části mluvit delší dobu (zvýraznění setrvá v téže poloze), lze využít nějaký typ časového zpoždění. Takto zvýrazňované oblasti mohou být např. typem seznamu, částí grafu nebo složek diagramu.

6.2. Mazání a vyřezávání

Kdybychom místo zdůrazňování bychom chtěli eliminovat zvýraznění, můžeme eliminovat, pomocí stejných metod, části vytvořeného zobrazení pomocí vymazání nebo výrezu. V prvním případě vymažeme vybranou oblast, v druhém zachováme vybranou oblast a vymažeme zbytek.

6.2.1 Mazání

Pro výběr oblastí určených k vymazání lze použít opačných metod jako u zvýrazňování pomocí kružnice nebo obdélníka. Tuto metodu použijeme potřebujeme-li zjednodušit složité zobrazení, získat místo pro doplňky zobrazení, zkoušet různá uspořádání, experimentovat s vizuálními efekty nebo potřebujeme místo pro rozšíření nebo nějaké jiné manipulace se zobrazením.

Části, které mají být zrušeny, bychom samozřejmě mohli odstranit změnou programu a novým zobrazením bez těchto částí. To by však mohlo představovat podstatné přepracování grafického zobrazovacího programu. Kdybychom chtěli později tyto části vrátit, museli bychom opět změnit program. Jsou-li nutná častá mazání, jako např. v oblasti designu, vytvoříme si obecný program pro vymazání všeho v označené oblasti.

Označení části, která má být vymazána, obdélníkem poskytuje jednoduchou metodu pro mazání znaků, bodů, a částí úseček uvnitř obdélníka. Vyplníme jednoduše tuto pravoúhlou oblast úsečkami téže barvy, jako je pozadí stínítka. Program pro vymazání musí mít jako vstup barvu pozadí stínítka a velikost a umístění obdélníka. Pokud není k dispozici volba barvy, použijeme příkaz pro vymazání každého bodu uvnitř obdélníka.

K vymazání může dojít uvnitř kružnice. To by však byl pomalejší proces než při použití obdélníka, protože takový program musí počítat body podél kruhové hranice. Tyto body lze pak užít jako koncové body úseček kreslených dovnitř kružnice. Některé systémy dovolují vyjádření barvy vnitřku obdélníků nebo kružnic pomocí parametru. Vnitřek těchto obrázků je při jejich kreslení automaticky vybarven zvolenou barvou.

6.2.2 Vyřezávání

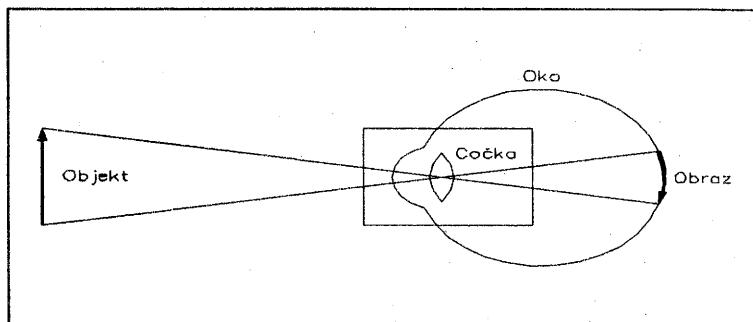
Někdy chceme zachovat pouze určitou oblast a všechno ostatní vymazat. Taková situace by nastala, kdybychom chtěli zvětšit malou část obrázku nebo grafu na velikost celého stínítka nebo kdybychom chtěli mít obrázky ukazující části zobrazení odděleně do nějaké zprávy nebo ukázky. K tomuto účelu můžeme zvolit oblast, která má být zachována v pravoúhelníkové oblasti neboli okénka (window). Vymazání všech částí zobrazení vně tohoto okénka se nazývá vyřezávání.

Program provádějící obecné vyřezávání musí identifikovat a zachovat všechny text, nepropojené body a úsečkové sekce prvků uvnitř okénka. Obrázek 8-1 ilustruje orámovanou oblast nebo okénko zvolené na diagramu. Vše, co je uvnitř okénka, má být zachováno; vše, co je vně, má být odříznuto.

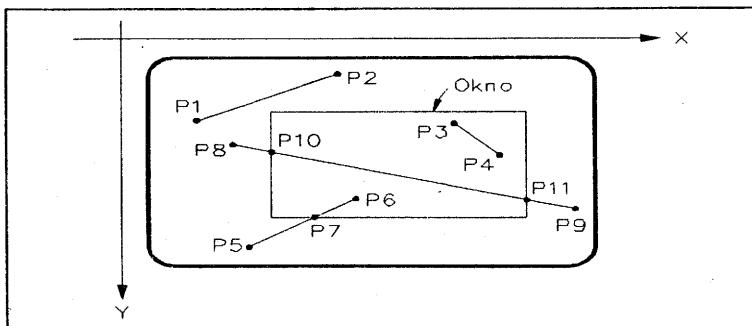
Program pro vyřezávání nejprve kontroluje koncové body úseček a textu, aby určil, co se má zachovat.

Nejprve se podívejme na vyřezávací program pro body a úsečky. Náš program bude požadovat souřadnice levého horního rohu okénka jako (XW,YW). Šířka a výška okénka bude vstupovat jako WW a HW. Souřadnice zobrazení budeme uchovávat v polích X a Y. Libovolná souřadnicová informace zachovaná vyřezávacím programem se bude ukládat do nových polí.

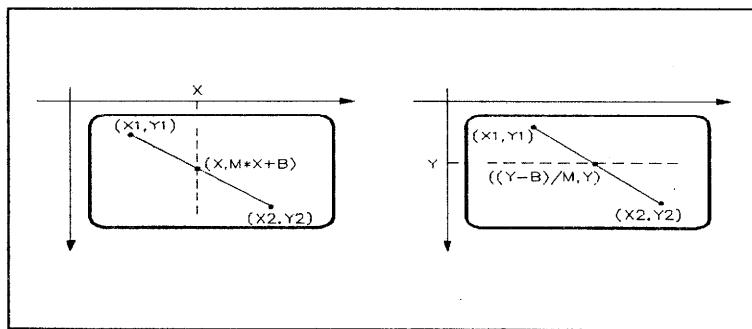
Izolovaný bod (ten, který není součástí úsečky) bude zachován vyřezávacím programem, pokud je umístěn uvnitř okénka. To znamená, že jeho souřadnice X je mezi XW a XW + HW. Úsečka bude zachována, pokud oba její koncové body leží uvnitř okénka. Celá úsečka bude vymazána, pokud leží celá vně okénka. Pokud leží části úsečky mimo okénko, odřízneme je a zachováme úsek uvnitř okénka. Průsečíky s vertikálními a horizontálními hranicemi okénka jsou vypočítány ze souřadnic okénka a rovníc přímky (směrnice M a průsečík B s osou Y), jak je uvedeno na Obr. 8-5.



Obr. 8-3 Okénko nakreslené kolem části zobrazení a textu, jež mají být zachovány vyřezávacím programem.



Obr. 8-4 Úsečka s koncovými body P1 a P2 vně okénka bude celá vymazána vyřezávacím programem. Úsečka s koncovými body P3 a P4 uvnitř okénka bude celá zachována. Vyřezávací program také zachová úsečky P7 - P6 a P10 - P11.

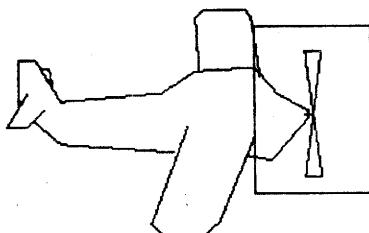


Obr. 8-5 Přímka se směrnicí M a průsečíkem B s osou Y protínající vertikální a horizontální hranice:
 (a) průsečík s vertikální hranicí má souřadnice $(X, M \cdot X + B)$; (b) průsečík s horizontální hranicí má souřadnice $((Y - B) / M, Y)$.

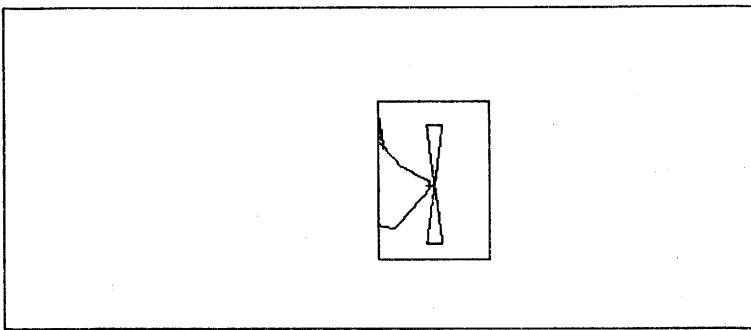
V Prog. 8-3, začnáme srovnáváním souřadnic s levým okrajem okénka. Body, jejichž souřadnice X je větší než hodnota souřadnice X levé strany okénka, jsou uloženy do polí X1 a Y1. Průsečíky úseček protínajících levou hranici okénka jsou také uloženy v X1 a Y1. Dále jsou odříznuty body uložené v X1 a Y1 podle horní hrany okénka. Všechny průsečíky a body se souřadnicí Y větší, než je tato hranice

jsou uloženy do polí X2 a Y2. Z těchto bodů se pak odříznou body podle pravé hrany. Pole X1 a Y1 jsou užita pro průsečíky a body, jejichž souřadnice X je menší než hodnota X pravého okraje. Nakonec odřízneme body nyní obsažené v polích X1 a Y1 podle dolní hranice okénka. Pole X2 a Y2 jsou užita pro výsledné souřadnice všech bodů a úseček které mají být znova nakresleny uvnitř okénka. Obr. 8-6 ukazuje zobrazení před a po výřezu programem Prog. 8-3.

UVED G PRO POKRACOVANI, C PRO ZMENU OKENKA? ■



(a)



(b)

Obr. 8-6 Obrázek zobrazený (a) před a (b) po výřezu programem Prog. 8-3.

```

10 'PROGRAM 8-3. VYREZAVANI VNE OKENKA. KRESLI OBRAZEK
20   'Z DAT ULOZENYCH V POLICHE X A Y. VOLI SE OBLAST
30   'OKENKA. USECKY A BODY VNE OKENKA JSOU ODRIZNUTY.
40   'ODREZAVANI SE PROVADI PODEL VSECH HRANIC V PORADI
50   'LEVA, HORNÍ, PRAVA, DOLNI. BEHEM ODREZAVANI JSOU
60   'BODY ULOZENE V X A Y ODRIZNUTY PODLE LEVE HRANICE;
70   'BODY UVNITR OKENKA JSOU ULOZENY V X1 A Y1. TYTO
80   'BODY JSOU PAK ODRIZNUTY PODLE HORNÍ HRANICE, SE
90   'ZACHOVAVANYMI BODY V X2 A Y2. BODY V X2,Y2 JSOU
100  'PAK ODRIZNUTY PODLE PRAVE HRANICE, A X1,Y1 JSOU
110  'ZNOVU POUZITY PRO ULOZENI BODU, KTERE JSOU JESTE
120  'UVNITR OKENKA. TYTO BODY JSOU NAKONEC ODRIZNUTY
130  'PODLE DOLNI HRANICE, A UVNITR LEZICI BODY JSOU
140  'ULOZENY DO X2,Y2. OBRAZOVKA JE VYMAZANA A OKENKO
150  'VCETNE VNITRNI CASTI OBRAZKU JSOU NAKRESLENY.
155  ****
160 CLS
170 DIM X(8,20), Y(8,20), X1(8,20), Y1(8,20),
     X2(8,20), Y2(8,20)
180 XM = 319      'XM, YM JSOU MAXIMALNI HODNOTY SOURADNIC
190 YM = 199      'PRO TENTO SYSTEM
200 ***** NACTENI CASTI OBRAZKU *****
210 READ N          'N JE POSET CASTI OBRAZKU
220 FOR P = 1 TO N

```

```

230      READ NE(P)      'POCET PRVKA V CASTI P
240      FOR E = 1 TO NE(P)
250          READ X(P,E), Y(P,E)
260      NEXT E
270 NEXT P
280 SCREEN 1
290 GOSUB 340      'KRESLENI OBRAZKU
300 GOSUB 450      'VYTVORENI OKENKA
310 GOSUB 850      'ODRIZNUTI BODU MIMO OKENKO
320 GOSUB 2480      'NAKRESLENI VYREZU
330 GOTO 2740
340 '***** PODPROGRAM KRESLENI *****
350 CLS
360 FOR P = 1 TO N
370     IF NE(P) > 1 THEN 400    'VICE NEZ JEDEN BOD
380     PSET(X(P,1),Y(P,1)) 'POUZE 1 BOD V TETO CASTI
390     GOTO 430              'POKRACUJ DALSI CASTI
400     FOR E = 1 TO NE(P)-1
410         LINE(X(P,E),Y(P,E))-(X(P,E+1),Y(P,E+1))
420     NEXT E
430 NEXT P
440 RETURN
450 '***** VYTVORENI OKENKA *****
460 LOCATE 1,1
470 PRINT "HORNI LEVY ROH OKENKA";
480 INPUT XW, YW
490 LOCATE 1,1
500 PRINT "                                     ";
510 LOCATE 1,1
520 PRINT "SIRKA A VYSKA OKENKA";
530 INPUT WW, HW
540 LOCATE 1,1
550 PRINT "                                     ";
560 L = XW
570 R = XW + WW
580 T = YW
590 B = YW + HW
600 IF L<R AND L>=0 AND R<=XM AND T<B AND T>= 0 AND B<=YM
    THEN 660
610 LOCATE 1,1
620 PRINT "OKENKO MIMO OBRAZOVKU. NOVY POKUS"

```

```

630 LOCATE 1,1
640 PRINT "
650 GOTO 460
660 LINE(L,T)-(L,B)
670 LINE(L,B)-(R,B)
680 LINE(R,B)-(R,T)
690 LINE(R,T)-(L,T)
700 LOCATE 1,1
710 PRINT "UVED G PRO POKRACOVANI, C PRO ZMENU OKENKA";
720 INPUT M$
730 IF M$ = "G" THEN 840
740   'VYMAZANI AKTUALNI POZICE
750 LOCATE 1,1
760 PRINT "
770 REM
780 LINE(L,T)-(L,B) ,0
790 LINE(L,B)-(R,B) ,0
800 LINE(R,B)-(R,T) ,0
810 LINE(R,T)-(L,T) ,0
820 REM
830 GOTO 450
840 RETURN
850 ##### VYREZAVACI PODPROGRAM #####
860   'ODRIZNUTI BODU V X A Y PODLE LEVE HRANY.
870   'ULOZENI DO X1 A Y1.
880 P1 = 1
890 FOR P = 1 TO N
900   E1 = 0
910   FOR E = 1 TO NE(P) - 1
920     IF X(P,E) >= L THEN 940
930     IF X(P,E) < L THEN 1020
940     'PRVY BOD USECKY JE UVNITR
950     E1 = E1 + 1
960     X1(P1,E1) = X(P,E)
970     Y1(P1,E1) = Y(P,E)
980     'A CO DRUHY BOD ?
990     IF X(P,E+1) < L THEN GOSUB 1190 'VNE-PRUSECIK,
1000       'JINAK JE UVNITR, TAK POKRACUJ
1010     GOTO 1070
1020   -----
1030   'PRVY BOD JE VNE

```

```

1040      'A CO DRUHY BOD ?
1050      IF X(P,E+1) >= L THEN GOSUB 1190 'V-PRUSECIK,
1060      'JINAK JE VNE, TAK POKRACUJ
1070      NEXT E
1080      'OSETRENI KONCOVEHO BODU
1090      IF X(P,NE(P)) < L THEN 1130
1100      E1 = E1 + 1
1110      X1(P1,E1) = X(P,NE(P))
1120      Y1(P1,E1) = Y(P,NE(P))
1130      IF E1 = 0 THEN 1160      'ZADNE PRVKY V OKENKU
1140      ME(P1) = E1      'JINAK ULOZ POSET OBSAZENYCH PRVKU
1150      P1 = P1 + 1      'POKRACUJ DALSI CASTI
1160      NEXT P
1170      MN = P1 - 1      'MN JE POSET CASTI S BODY UVNITR
1180      GOTO 1260
1190      ,^^^^^^ PODPROGRAM PRO URCENI PRUSECIKU ^^^
1200      E1 = E1 + 1
1210      M = (Y(P,E+1) - Y(P,E)) / (X(P,E+1) - X(P,E))
1220      Y1(P1,E1) = M * (L - X(P,E)) + Y(P,E)
1230      X1(P1,E1) = L
1240      RETURN
1250      ****
1260      'ODRIZNUTI BODU V X1 A Y1 PODLE HORNÍ HRANY.
1265      'ULOZENI DO X2 A Y2.
1270      P1 = 1
1280      FOR P = 1 TO MN
1290          E1 = 0
1300          FOR E = 1 TO ME(P) - 1
1310              IF Y1(P,E) >= T THEN 1330
1320              IF Y1(P,E) <= T THEN 1420
1330              'PRVY BOD USECKY JE UVNITR
1340              E1 = E1 + 1
1350              Y2(P1,E1) = Y1(P,E)
1360              X2(P1,E1) = X1(P,E)
1370              'A CO DRUHY BOD ?
1380              IF Y1(P,E+1) < T THEN GOSUB 1580
1390              'VNE-PRUSECIK,
1400              'JINAK JE UVNITR, TAK POKRACUJ
1400              GOTO 1460
1410              -----
1420              'PRVY BOD JE VNE

```

```

1430      'A CO DRUHY BOD ?
1440      IF Y1(P,E+1) >= T THEN GOSUB 1580 'V-PRUSECIK,
1450      'JINAK JE VNE, TAK POKRACUJ
1460      NEXT E
1470      'OSETRENI KONCOVEHO BODU
1480      IF Y1(P,ME(P)) < T THEN 1520
1490      E1 = E1 + 1
1500      Y2(P1,E1) = Y1(P,ME(P))
1510      X2(P1,E1) = X1(P,ME(P))
1520      IF E1 = 0 THEN 1550      'ZADNE PRVKY V OKENKU
1530      ME(P1) = E1      'JINAK ULOZ POSET OBSAZENYCH PRVKU
1540      P1 = P1 + 1      'POKRACUJ DALSI CASTI
1550      NEXT P
1560      MN = P1 - 1      'MN JE POSET CASTI S BODY UVNITR
1570      GOTO 1680
1580      ,***** PODPROGRAM PRO URCENI PRUSECIKU *****
1590      E1 = E1 + 1
1600      IF X1(P,E+1) <> X1(P,E) THEN 1630
1610      X2(P1,E1) = X1(P,E)      'VERTIKALNI USECKA
1620      GOTO 1650
1630      M = (Y1(P,E+1) - Y1(P,E)) / (X1(P,E+1) - X1(P,E))
1640      X2(P1,E1) = (T - Y1(P,E)) / M + X1(P,E)
1650      Y2(P1,E1) = T
1660      RETURN
1670      ****
1680      'ODRIZNUTI BODU V X2 A Y2 PODLE PRAVE HRANY.
1685      'ULOZENI VNITRICH BODU DO X1 A Y1.
1690      P1 = 1
1700      FOR P = 1 TO MN
1710          E1 = 0
1720          FOR E = 1 TO ME(P) - 1
1730              IF X2(P,E) <= R THEN 1750
1740              IF X2(P,E) > R THEN 1840
1750              'PRVY BOD USECKY JE UVNITR
1760              E1 = E1 + 1
1770              X1(P1,E1) = X2(P,E)
1780              Y1(P1,E1) = Y2(P,E)
1790              'A CO DRUHY BOD ?
1800              IF X2(P,E+1) > R THEN GOSUB 2000
1810              'VNE-PRUSECIK,
1810              'JINAK JE UVNITR, TAK POKRACUJ

```

```

1820      GOTO 1880
1830      -----
1840      'PRVY BOD JE VNE
1850      'A CO DRUHY BOD ?
1860      IF X2(P,E+1) <= R THEN GOSUB 2000 'V-PRUSECIK,
1870      'JINAK JE VNE, TAK POKRACUJ
1880      NEXT E
1890      'OSETRENI KONCOVEHO BODU
1900      IF X2(P,ME(P)) > R THEN 1940
1910      E1 = E1 + 1
1920      X1(P1,E1) = X2(P,ME(P))
1930      Y1(P1,E1) = Y2(P,ME(P))
1940      IF E1 = 0 THEN 1970      'ZADNE PRVKY V OKENKU
1950      ME(P1) = E1      'JINAK ULOZ POSET OBSAZENYCH PRVKA
1960      P1 = P1 + 1      'POKRACUJ DALSI CASTI
1970      NEXT P
1980      MN = P1 - 1      'MN JE POSET CASTI S BODY UVNITR
1990      GOTO 2070
2000      ,^^^^^^ PODPROGRAM PRO URCENI PRUSECIKU ^^^
2010      E1 = E1 + 1
2020      M = (Y2(P,E+1) - Y2(P,E)) / (X2(P,E+1) - X2(P,E))
2030      Y1(P1,E1) = M * (R - X2(P,E)) + Y2(P,E)
2040      X1(P1,E1) = R
2050      RETURN
2060      ****
2070      'ODRIZNUTI BODU V X1 A Y1 PODLE DOLNI HRANY.
2075      'ULOZENI DO X2 A Y2.
2080      P1 = 1
2090      FOR P = 1 TO MN
2100          E1 = 0
2110          FOR E = 1 TO ME(P) - 1
2120              IF Y1(P,E) <= B THEN 2140
2130              IF Y1(P,E) > B THEN 2230
2140                  'PRVY BOD USECKY JE UVNITR
2150                  E1 = E1 + 1
2160                  Y2(P1,E1) = Y1(P,E)
2170                  X2(P1,E1) = X1(P,E)
2180                  'A CO DRUHY BOD ?
2190                  IF Y1(P,E+1) > B THEN GOSUB 2390 'VNE-PRUSECIK
2200                  'JINAK JE UVNITR, TAK POKRACUJ
2210                  GOTO 2270

```

```

2220      -----
2230      'PRVY BOD JE VNE
2240      'A CO DRUHY BOD ?
2250      IF Y1(P,E+1) <= B THEN GOSUB 2390 'V-PRUSECIK,
2260      'JINAK JE VNE, TAK POKRACUJ
2270      NEXT E
2280      'OSETRENI KONCOVEHO BODU
2290      IF Y1(P,ME(P)) > B THEN 2330
2300      E1 = E1 + 1
2310      Y2(P1,E1) = Y1(P,ME(P))
2320      X2(P1,E1) = X1(P,ME(P))
2330      IF E1 = 0 THEN 2360      'ZADNE PRVKY V OKENKU
2340      ME(P1) = E1      'JINAK ULOZ POSET OBSAZENYCH PRVKU
2350      P1 = P1 + 1      'POKRACUJ DALSI CASTI
2360      NEXT P
2370      MN = P1 - 1      'MN JE POSET CASTI S BODY UVNITR
2380      RETURN      'KONEC PODPROGRAMU PRO VYREZAVANI
2390      ,~~~~~ PODPROGRAM PRO URCENI PRUSECIKU ^^^
2400      E1 = E1 + 1
2410      IF X1(P,E+1) <> X1(P,E) THEN 2440
2420      X2(P1,E1) = X1(P,E)      'VERTIKALNI USECKA
2430      GOTO 2460
2440      M = (Y1(P,E+1) - Y1(P,E)) / (X1(P,E+1) - X1(P,E))
2450      X2(P1,E1) = (B - Y1(P,E)) / M + X1(P,E)
2460      Y2(P1,E1) = B
2470      RETURN
2480      ##### KRESLENI VYRIZNUTYCH BODU #####
2490      CLS
2500      LINE(L,T)-(L,B)
2510      LINE(L,B)-(R,B)
2520      LINE(R,B)-(R,T)
2530      LINE(R,T)-(L,T)
2540      FOR P = 1 TO MN
2550          FOR E = 1 TO ME(P) - 1
2560              LINE(X2(P,E),Y2(P,E))-(X2(P,E+1),Y2(P,E+1))
2570          NEXT E
2580      NEXT P
2590      RETURN
2600      #####
2610      DATA 7
2620      'TELO

```

```

2630 DATA 12,195,127,210,130,232,105,232,100,213,90,200,75
2640 DATA 165,77,145,90,85,95,70,70,60,70,60,97
2650 DATA 3,69,106,85,120,152,125
2660      'KRIDLÁ
2670 DATA 6,160,110,138,168,145,175,170,175,178,168,200,115
2680 DATA 6,203,80,198,45,195,40,170,40,165,45,167,76
2690      'OCAS
2700 DATA 3,73,75,78,75,80,85
2710 DATA 4,65,90,53,110,65,110,75,100
2720      'VRTULE
2730 DATA 7,230,103,23 5,103,230, 65, 240, 65,230,140,
         240,140,235,103
2740 END

```

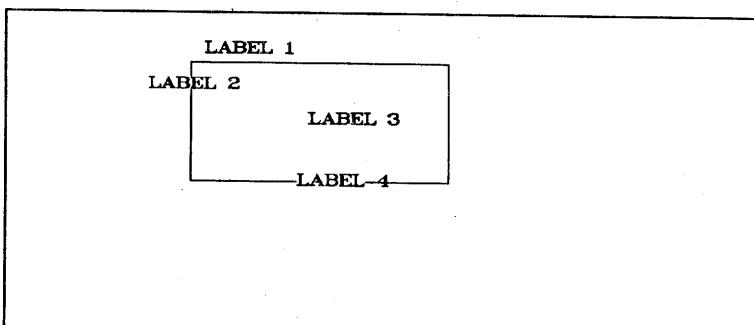
Prog. 8-3 Výřezy bodů a přímek (letadlo).

Nyní rozšíříme program Prog. 8-3 o vyříznutí libovolných textů na obrázku. Texty můžeme chápat jako úsečky a ponechat část uvnitř okénka, ale pro zjednodušení ponecháme znakové řetězce (texty), jen jsou-li celé uvnitř hranic okénka. Na Obr. 8-7 by byla pro uvedené okénko zachována pouze textová informace LABEL 3. Náš test pro zachování řetězce bude určovat, zda řetězec začíná za levou a končí před pravou hranicí okénka. Řetězec musí být také na řádku, který leží mezi horní a dolní hranicí okénka.

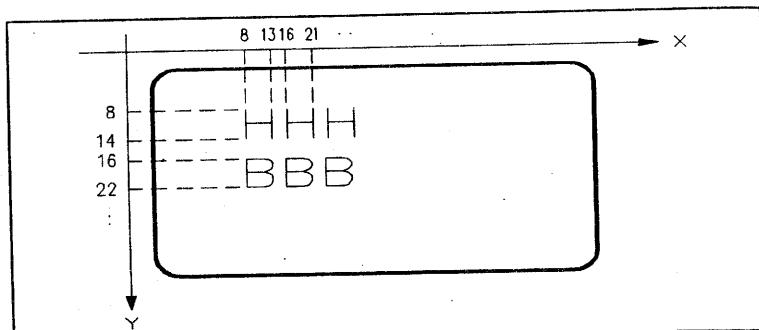
Náš program bude používat pole T\$ pro ukládání znakových řetězců. Pole XL a XR užijeme pro uložení pozic X začátků a konců každého řetězce. Pole YT a YB slouží k ukládání souřadnic Y horní a dolní hranice řetězců. Hodnoty souřadnic řetězců XL, XR, YT a YB získáme z rozměrů bodové mřížky znaku pro daný systém (Část 3-2).

Pro náš program na výrezávání textu budeme předpokládat, že každému znaku odpovídá mřížka 8 x 8 bodů. A také že každý znak ve skutečnosti zabere plochu mřížky 6 x 7 bodů. Zůstanou tedy minimálně dva body mezi dvěma znaky horizontálně a jeden bod mezi dvěma řádky vertikálně

(oddělení, viz Obr. 8-8). Znaky zobrazované na řádce tedy začínají na souřadnici 0, 8, 16, 24, ... a končí na souřadnici 5, 13, 21, atd. Horní okraje znaků mají souřadnice 0, 8, 16 ... a dolní okraje 6, 14, 22, atd. K určení souřadnice začátku znaku odečteme 1 od tiskové pozice znaku a násobíme 8. Pro určení koncové pozice znaku přičteme 5 k počáteční pozici. Pro znak tisknutý v pozici LOCATE 6,21 je souřadnice X začátku 160 a konce 165. Abychom určili souřadnici Y horního okraje libovolného znaku, odečteme 1 od čísla tiskové řádky a násobíme 8. Přičtením 6 k souřadnici horního okraje dostaneme souřadnici Y dolního okraje znaku. Pro znakovou pozici určenou jako LOCATE 6,21 je souřadnice horního okraje 40 a dolního okraje 46. Úpravy programu Prog. 8-3 pro vyřezávání textů jsou uvedeny v Prog. 8-4. Výstup tohoto programu je uveden na Obr. 8-9.



Obr. 8-7 Vyřezávání libovolných textových řetězců, které nejsou celé obsaženy uvnitř okénka, vymaže řetězce "LABEL 1", "LABEL 2" a "LABEL 4".



Obr. 8-8 Souřadnice znaků v mřížce 8 x 8 bodů. Horizontálně znaky začínají na pozicích 0, 8, 16, ... a končí na pozicích 5, 13, 21 atd. Horní okraje tiskových řádek jsou na pozicích 0, 8, 16, ... a dolní okraje tiskových řádek na 6, 14, 22 směrem dolů po obrazovce.

```

10 'PROGRAM 8-4. VYREZAVANI TEXTU A OBRAZKU
20      '(DODATEK K PROG. 8-3)
171 DIM T$(10),RO(10),CO(10),TL(10),TR(10),TT(10),TB(10)
271   'NACTENI TEXTOVYCH POLOZEK, RADKU A SLOUPCE
272 READ TN      'TN JE POSET TEXTOVYCH POLOZEK
273 FOR K = 1 TO TN
274   READ T$(K),RO(K),CO(K) 'RO & CO JE RADEK A SLOUPEC
275   TL(K) = (CO(K) - 1) * 8 'PREVOD NA SOURADNICE
276   TR(K) = TL(K) - 1 + LEN(T$(K)) * 8
277   TT(K) = (RO(K) - 1) * 8
278   TB(K) = TT(K) + 7
279 NEXT K
315 GOSUB 2461  'VYREZ TEXTU
431   'UMISTENI TEXTOVYCH POLOZEK (CAST PODPROG. KRESL.)
432 FOR K = 1 TO TN
433   LOCATE RO(K), CO(K)
434   PRINT T$(K);
435 NEXT K
2461 C1 = 0      'C1 JE POSET TEXTOVYCH POLOZEK V OKENKU
2462 FOR K = 1 TO TN
2463   IF TL(K)<L OR TR(K)>R OR TT(K)<T OR TB(K)>B THEN
2468
2464   C1 = C1 + 1
2465   T$(C1) = T$(K)
2466   RO(C1) = RO(K)

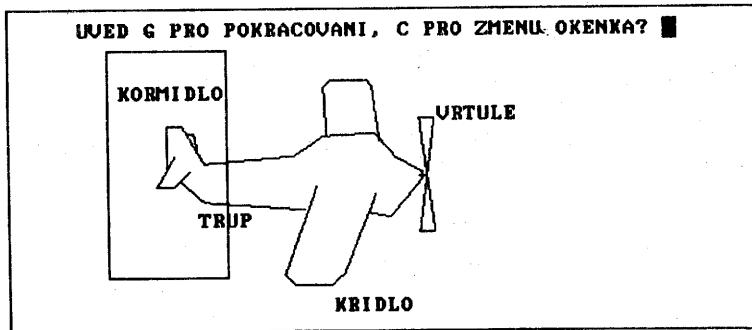
```

```

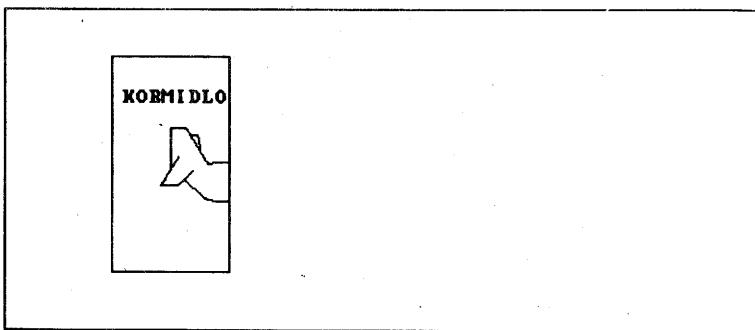
2467      CO(C1) = CO(K)
2468 NEXT K
2469 RETURN
2581      'DOPLNENI TEXTOVYCH POLOZEK
2582 FOR K = 1 TO C1
2583      LOCATE RO(K), CO(K)
2584      PRINT T$(K);
2585 NEXT K
2731      'TEXTOVE POLOZKY
2732 DATA 4
2733 DATA TRUP,14,10
2734 DATA KORMIDLO,7,4
2735 DATA KRIDLO,20,20
2736 DATA VRTULE,6,28
2750 END

```

Prog. 8-4 Vyřezávání bodů, úseček a textů (jetadio).



(a)



Obr. 8-9 Obrázek s textem zobrazený (a) před a (b) po výřezu programem Prog. 8-4.

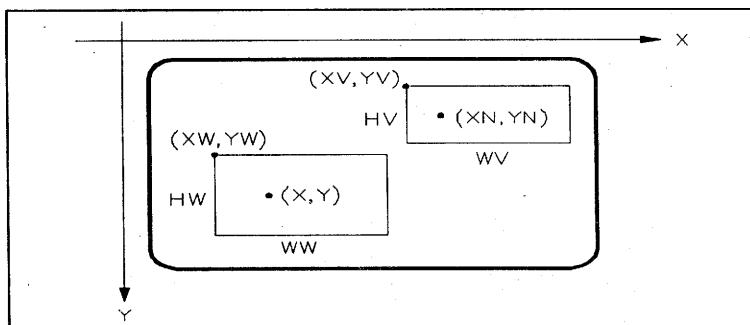
Program Prog. 8-4 můžeme upravit pro jiné systémy užitím jiných rozměrů znakové mřížky. Také můžeme upravit vyřezávání textu tak, že zachováme všechny části textu uvnitř okénka. Například můžeme zpracovávat každý bod obrázku jednou, místo po jednotlivých úsečkách. Pro každý bod lze stanovit kód, určující pozici tohoto bodu vzhledem ke hranici okénka. Tento kód pak lze užít pro test polohy úseček vzhledem k okénku. V některých grafických systémech se vyřezávání provádí automaticky podle specifikovaného okénka.

Po výřezu obrázku transformujeme oblast okénka. Bud' okénko zobrazíme na jiném místě (translace), nebo změníme měřítko, nebo natočíme (rotace) část obrázku, která zbude v okénku. Lze také zachovat původní obrázek a překrýt okénkem - např. zvětšeným - v jednom rohu stínítka obrazovky.

6.3. Oblast pozorování (viewport)

Okénko na obrázku můžeme přesouvat a libovolně měnit jeho měřítko přemístěním do určené pravoúhelníkové oblasti na stínítku obrazovky. Tato oblast se nazývá OBLAST POZOROVÁNÍ neboli VIEWPORT. Jak okénko, tak viewport můžeme na obrazovce vytvořit uvedením souřadnic levého horního rohu a velikosti každé pravoúhelníkové oblasti, jak ukazuje Obr. 8-10.

Okénko určuje "CO" chceme vidět a viewport určuje "KDE" bychom to rádi viděli na stínítku obrazovky. Viewport může být větší, nebo menší než okénko, nebo může mít stejnou velikost. Viewport může vyplnit celé stínítko obrazovky, nebo může tvořit malý doplněk zobrazení. Okénko a viewport lze bud' vytvářet v oddělených oblastech stínítku, nebo se mohou překrývat. Pokud je viewport užit pro zvětšení části obrázku, lze přitom zvětšit detaily, které jsou příliš malé, než aby byly vidět na původním zobrazení.



Obr. 8-10 Specifikace okénka a viewportu. Horní levý roh okénka má souřadnice (X_W, Y_W) ; horní levý roh viewportu (X_V, Y_V) . Velikost každé oblasti je určena šířkou a výškou (W_V, H_V) a (W_W, H_W) . Bod se souřadnicemi (X, Y) v okénku bude transformován do nové polohy (X_N, Y_N) ve viewportu.

Program pro transformaci oblasti okénka na oblast viewportu musí transformovat souřadnice každého bodu okénka, jako je (X,Y) na Obr. 8-10, na odpovídající nový bod (XN,YN) ve viewportu. Tato transformace se provádí téměř shodným způsobem, jaký jsme uvedli pro vytváření grafů ve specifikovaných oblastech v KAPITOLE 4. Vztah nových souřadnic ve viewportu k původním souřadnicím bodu okénka lze vyjádřit jako

$$XN = (X - XW) * \frac{WV}{WW} + XV \quad (8-1)$$

$$YN = (Y - YW) * \frac{HV}{HW} + YV$$

Koefficienty (WV/WW) a (HV/HW) v rovnících (8-1) představují změnu měřítka. Viewport má odlišné rozměry, pokud jsou tyto koefficienty různé od 1. Hodnota větší než 1 zvětšuje okénko, hodnota menší než 1 zmenšuje. Pokud není poměr WV/WW roven poměru HV/HW, dochází k pokřivení oblasti okénka. To odpovídá různé změně měřítek ve směru X a Y, což bylo diskutováno v předchozích kapitolách. Hodnoty XV a YV představují translaci. Pokud jsou tyto souřadnice odlišné od XW a YW, dochází k přemístění oblasti.

Zobrazení z určité oblasti okénka na specifikovaný viewport se provádí programem Prog. 8-5, rozšířením programu prog. 8-3. Obrázek Obr. 8-11 ilustruje výstup tohoto programu. Tento příklad demonstruje, jak lze viewport použít pro zvětšování detailů obrázku. Takové zvětšování je užitečné při práci s mapou nebo složitými diagramy pro zobrazování detailů.

2740 'PROGRAM 8-5. ZOBRAZENÍ OKENKA VE VIEWPORTU ####

```

2750           DOPLNEK K PROGRAMU PROG. 8-3
2760           'VYTvari VIEWPORT NA STINITKU. CASTI OBRAZKU
2770           'UVNITR OKENKA PREMISTI DO VYBRANEOHO VIEWPORTU
2780           '
2790           ****
2800 DIM XN(8,20), YN(8,20)
2810 GOSUB 2850      'VYTVORENI VIEWPORTU
2820 GOSUB 3070      'PREVOD OBLASTI OKENKA DO VIEWPORTU
2830 GOSUB 3150      'NAKRESLENI CASTI OBRAZKU VE VIEWPORTU
2840 GOTO 3280
2850 '##### VYTVORENI VIEWPORTU #####
2860 LOCATE 1,1
2870 PRINT "HORNI LEVY ROH VIEWPORTU";
2880 INPUT XV, YV
2890 LOCATE 1,1
2900 PRINT "
2910 LOCATE 1,1
2920 PRINT "SIRKA A VYSKA VIEWPORTU";
2930 INPUT WV, HV
2940 LOCATE 1,1
2950 PRINT "
2960 L = XV
2970 R = XV + WV
2980 T = YV
2990 B = YV + HV
3000 IF L<R AND L>=0 AND R<=XM AND T<B AND T>=0 AND B<=YM
     THEN 3060
3010 LOCATE 1,1
3020 PRINT "VIEWPORT MIMO STINITKO. NOVY POKUS"
3030 LOCATE 1,1
3040 PRINT "
3050 GOTO 2860
3060 RETURN
3070 '## KONVERZE OBLASTI OKENKA NA OBLAST VIEWPORTU ##
3080 FOR P = 1 TO MN
3090     FOR E = 1 TO ME(P)
3100         XN(P,E) = (X2(P,E) - XW) * (WV / WW) + XV
3110         YN(P,E) = (Y2(P,E) - YW) * (HV / HW) + YV
3120     NEXT E
3130 NEXT P
3140 RETURN

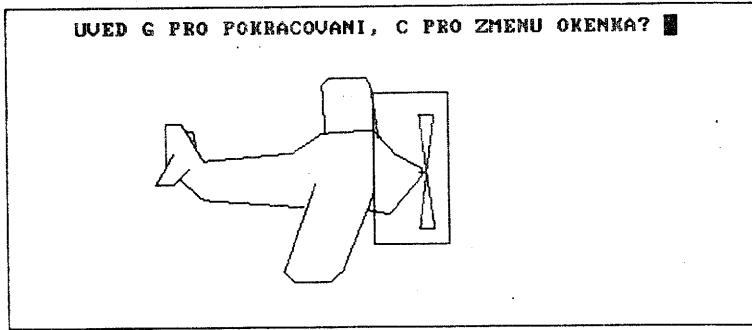
```

```

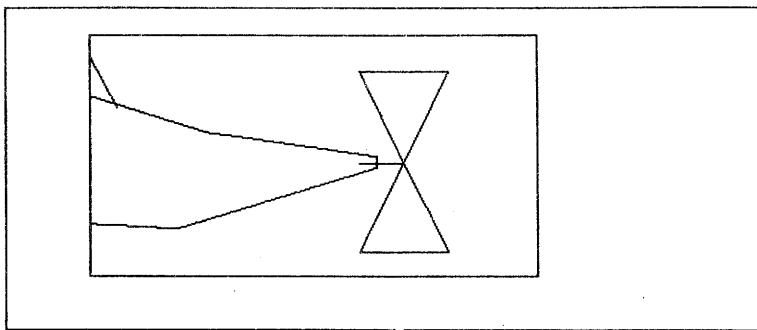
3150 ' ##### KRESLENI VYRIZNUTYCH BODU #####
3160 CLS
3170 LINE(L,T)-(L,B)
3180 LINE(L,B)-(R,B)
3190 LINE(R,B)-(R,T)
3200 LINE(R,T)-(L,T)
3210 FOR P = 1 TO MN
3220     FOR E = 1 TO ME(P) - 1
3230         LINE(XN(P,E),YN(P,E))-(XN(P,E+1),YN(P,E+1))
3240     NEXT E
3250 NEXT P
3260 RETURN
3270 ' ##### #####
3280 END

```

Prog. 8-5 Doplněk k prog 8-3



(a)



(b)

Obr. 8-11 Oblast okénka v původním zobrazení (a) je zvětšena do oblasti viewportu (b) pomocí programu Prog. 8-5.

U některých aplikací je nutno zobrazovat na stínítku obrazovky jak původní scénu, tak viewport. Můžeme také vytvářet postupně větší počet okének a zobrazovat současně i více než jeden viewport. Jako poslední transformace okénka může být viewport natáčen pro zobrazení oblasti v jiné orientaci.

PROJEKTY PROGRAMOVÁNÍ

- 8-1. Napište zvýrazňovací program, který bude zdůrazňovat úsečky tím, že je nakreslí jasnější (nebo odlišnou) barvou, při použití pravoúhelníkového nebo kruhového zvýrazňování. Jako další vlastnost bude možnost změny barvy pozadí ve zvýrazňované oblasti.
- 8-2. Vytvořte program, který bude zvýrazňovat oblasti zobrazení pomocí elipsy o libovolných specifikovaných rozměrech. Použijte podprogram pro vymazání elliptické oblasti, jako vstupní volbu.
- 8-3. Napište podprogram pro vyřezávání znakových řetězců podle hranic pravoúhelníkové oblasti vymazáváním pouze té části řetězce, která je uvnitř okénka.
- 8-4. Napište vyřezávací program, který vymaže všechny části vně specifikované kruhové oblasti.
- 8-5. Napište vyřezávací program, který užívá viewport pro zvětšování některé oblasti obrázku, pak překryje touto zvětšenou částí původní obrázek v jednom rohu stínítka obrazovky. Místo vymazání celého stínítka vymaže pouze oblast, která má obsahovat viewport.
- 8-6. Upravte Prog. 8-3 tak, aby každý bod obrázku byl srovnáván s hranicí okénka pouze jednou. Poloha každého bodu vzhledem k okénku může být určena v nějakém poli, které určuje zda příslušný bod je UVNITŘ (IN) nebo VNĚ (OUT), NAD (ABOVE) nebo POD (BELLOW) a VPRAVO (RIGHT) nebo VLEVO (LEFT). Tyto informace se mohou kódovat znakovými řetězci délky 3 (např. "OBR" by mohl hlásit, že odpovídající bod je mimo = OÚT = O, pod = BELLOW = B, vpravo = RIGHT = R). Po zpracování

všech bodů, určujícím polohové řetězce pro každý bod, mohou být úsečky obrázku vyřezávány testováním odpovídajících polohových řetězců všech koncových bodů. Úsečka zůstane zachována pokud jsou oba její koncové body uvnitř = IN = I. Zruší se, jsou-li oba její koncové body vlevo = LEFT = L nebo nad = ABOVE = A atd. Průsečíky pak nalezeny překládáním úseček. Polohové řetězce mohou být redukovány na dva znaky nastavením prvého znaku na "X" (uvnitř), "A" (nad) nebo "B" (pod) a nastavením druhého znaku na "L" (vlevo) nebo "R" (vpravo).

-)
- 8-7. Napište program, který bude zobrazovat libovolný počet viewportů oblastí obrázku na stínítku obrazovky. 8-8. Napište program, který bude natáčet (rotace) viewport do libovolné specifikované orientace.