

SZÁMÍTÓGÉPI GRAFIKA

MARKEREK

FELADAT

Egy *grafikus markert* kezelő program elkészítése, amely képes

1. a markert a képernyőn *mozgatni* (a kurzormozgató billentyűk segítségével),
2. a *méretét* növelni, illetve csökkenteni (a méret a mozgás lépésköze),
3. *alakját* megváltoztatni (amely külső jegye lehet az esetleges speciális funkciójának).

TUDNIVALÓK

1. Mindenekelőtt próbálja ki: mire lehet gondolni ([Marker.exe](#))! („Kezeléséhez”: mozgítás a kurzormozgató billentyűkkel, méretnövelés „+” billentyű, csökkentés „-” billentyű, alakváltoztatás az „0”...„4” billentyűkkel, kilépni az „Esc” billentyűvel.)

2. A marker Pascal típusa:

```
TMarker=Record
    typ:TMarkerType;
    size,color:Integer;{alaki jellemzők}
    x,y:Integer;{hollét}
End;
TMarkerType=(MT_Pont,MT_Plusz,MT_Rombusz,MT_X,MT_Nyil);
              {0⇒pont, 1⇒+; 2⇒∅; 3⇒x; 4⇒↖ (ferde nyíl)}
```

3. A markert kezelő műveletek:

```
Procedure SetMarker(
    Var mk:TMarker;
    Const typ:TMarkerType; siz,col,x,y: Integer);
    {beállítja a marker tulajdonságait}
Procedure SetMarkerType(
    Var mk:TMarker; Const typ:TMarkerType);
    {átállítja a marker típusát}
Procedure SetMarkerSize(
    Var mk:TMarker; Const siz:Integer);
    {átállítja a marker méretét}
Procedure SetMarkerColor(
    Var mk:TMarker; Const col:Integer);
    {átállítja a marker színét}
```

Számítógépi grafika – Markerek

```
Procedure Marker(  
    Var mk:TMarker; Const x,y:Integer);  
    {kirajzolja a markert az (x,y) bázisponttól}
```

... és ezek közül a legtöbbnek *GetOperáció* nevű függvénypárja, mint pl. az alábbi.

```
Function GetMarkerType(Const mk:TMarker):TMarkerType;  
    {visszadja a marker típusát}
```

...

4. A XORPut vonalrajzoló mód beállításával elérhető, hogy könnyen visszaállíthassuk a marker „alatti” képeket annak elmozdulásakor. Elegendő ugyanott „újrarajzolni” a markert, és máris előtűnik az a kép, amelyre rákerült a kirajzoláskor.¹ A vonalrajzoló mód beállítása a SetWriteMode(XORPut) utasítással történik. A rajzoló mód figyelembevételével működik a Line, a LineTo, a Rectangle, de **nem** így a PutPixel és a Circle... L. a Turbo Grafika súgóját.
5. A Free Pascal esetén két ablakban fut az alkalmazás: egyik a „normál” konzolablak, a másik a grafikus ablak. Mivel a beavatkozás műveleteit (ReadKey, KeyPressed) szokásosan a Crt unit tartalmazza, ezért a beavatkozáshoz a konzolablakot kell aktívvá tenni, ekkor viszont részben takarva lesz a „főszereplő” ablak. Ezen segíthetünk a WinCrt unittel. Azonban fontos a Crt, WinCrt sorrendje a Uses sorban: előbb legyen a Crt.

Először megírandó a markert kezelő unit (MarkerUn.pas néven)! E unitra építve a fent körvonalazott egyszerű feladatot az alábbi program megoldja. (Letöltheti a [Marker.pas](#)-t.)

AZ ELSŐ PRÓBAPROGRAM

```
1      Program Markerek;  
2      (*  
3      Markerek témához próbaprogram.  
4      Egy marker kezelése.  
5      Egy marker kezelése: mozgatás, nagyítás/kicsinyítés, típusváltás.  
6      A marker típusa (alakja) és billentyűje:  
7      * "pont"      -- 0  
8      * "+"         -- 1  
9      * "rombusz"  -- 2  
10     * "X"         -- 3  
11     * "nyíl"     -- 4  
12     *)
```

¹ Kitalálható, hogy e rajzoló mód háttérében a pixelek színbitjeinek XOR („kizáró-vagy”) művelete húzódik, amely –mint köztudomású– saját maga inverz művelete.

Számítógépi grafika – Markerek

```
13     Uses
14     {$IFDEF FPC -- TurboPascal}
15         Newdelay, {hogy ne szálljon el a Crt-ben}
16     {$ENDIF}
17         Crt,
18     {$IFDEF FPC -- FreePascal}
19         WinCrt, {hogy a GUI-ban billentyűkkel lehessen vezérelni; Crt után!!!}
20     {$ENDIF}
21         Graph, MarkerUn;
22
23     Const
24         path='c:\LANGS\bp\bgi\' ;
25         ESC=#27;
26         Le=#80; Fel=#72; Bal=#75; Jobb=#77;
27     Var
28         bfx,bfy,jax,jay:Integer; {rajzterület}
29         mk      :TMarker;
30         ch      :Char;
31         x,y     :Integer;
32         meret:Integer;
33
34     Procedure Karkezd;
35     Begin
36         Window(1,1,80,25);
37         TextBackGround(Green);
38         ClrScr;
39     End;
40
41     Procedure Szovegesre;
42     Begin
43         CloseGraph;
44         RestoreCrtMode;
45     End;
46
47     Procedure Grafikusra;
48     Var
49         gd,gm:Integer;
50     Begin
51         gd:=detect; gm:=0;
52         InitGraph(gd,gm,path);
53         SetBkColor(Green);
54         ClearDevice;
55         SetLineStyle(SolidLn,02,ThickWidth);
56         Rectangle(2,2,GetMaxX-2,GetMaxY-2);
57         bfx:=5; bfy:=5; jax:=GetMaxX-5; jay:=GetMaxY-5;
58         SetWriteMode(XORPut);
59     End;
60
```

² Ez a paraméter a „vonalmintázat”, amelynek szerepe csak az első paraméter „UserBitLn” esetében van.

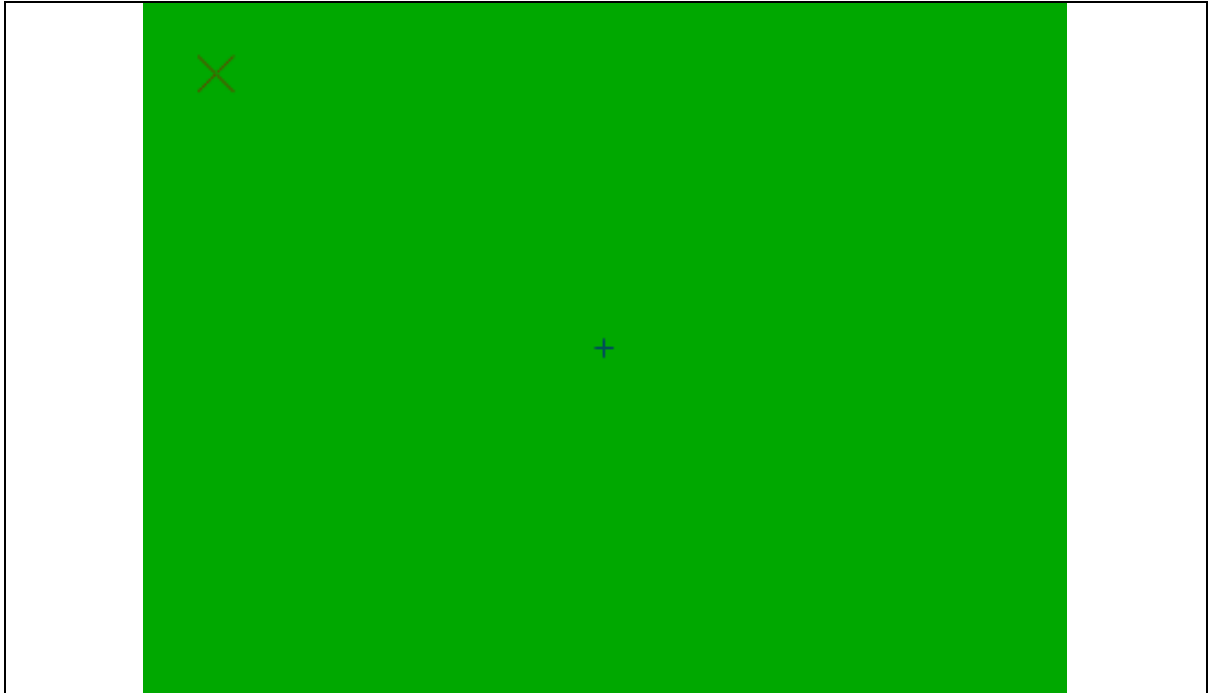
Számítógépi grafika – Markerek

```
61 Begin {főprogram}
62   Karkezd;
63   Grafikusra;
64   x:=GetMaxX Div 2; y:=GetMaxY Div 2; meret:=8;
65   SetMarker(mk,MT_Plusz,meret,Red,x,y);
66   Marker(mk,x,y); {marker kirajzolása}
67   Repeat
68     ch:=ReadKey;
69     Marker(mk,x,y); {a régi marker levétele}
70     Case ch of
71       '0'..'4': SetMarkerType(mk,TMarkerType(ord(ch)-ord('0')));
72       '+': Begin
73         meret:=meret+1; {alighanem célszerű lenne korlátozni!!!}
74         SetMarkerSize(mk,meret);
75       End;
76       '-': if meret>1 then Begin
77         meret:=meret-1;
78         SetMarkerSize(mk,meret);
79       End;
80       #0: Begin {funkció billentyűk esetén}
81         ch:=ReadKey;
82         case ch of
83           Le: Begin
84             If (y+meret<=jy) then y:=y+meret;
85           End;
86           Fel: Begin
87             If (y-meret>=by) then y:=y-meret;
88           End;
89           Bal: Begin
90             If (x-meret>=bx) then x:=x-meret;
91           End;
92           Jobb: Begin
93             If (x+meret<=jx) then x:=x+meret;
94           End;
95         End;
96       End;
97     End; {Case}
98     Marker(mk,x,y); {a marker újbóli kirajzolása -- az új állapotában}
99   Until ch=Esc;
100   Szovegesre;
101 End.
```

FEJLESZTÉSEK

A marker unitját felhasználva készítse el a következő változatokat!

1. Egyidejűleg **két marker** legyen látható és vezérelhető (az egyik „aktív”, másik „inaktív”). A markerek közötti váltásra a `TAB`-billentyű legyen használható. A két marker „funkcionális” (aktív/inaktív) megkülönböztetése történjen színnel.³ (Kipróbálhatja: [Marker2.exe](#).)

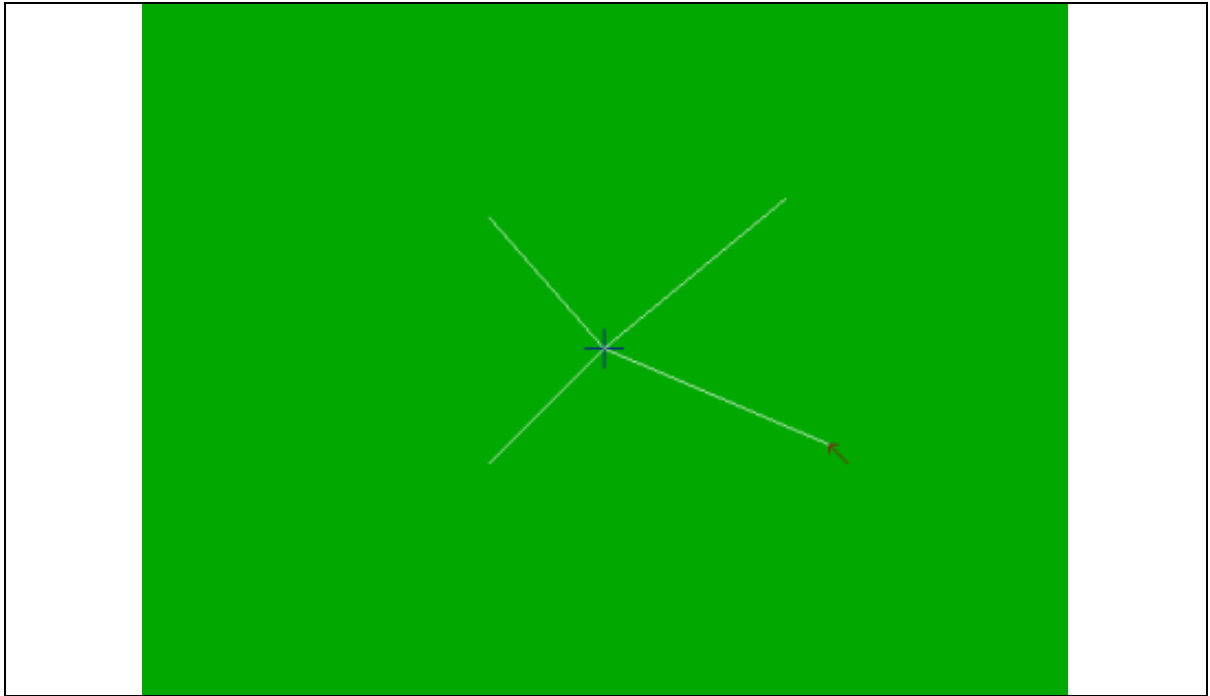


1. ábra. Kezdőállapot.

³ Megoldásnál a próbaprogramot úgy kell módosítani, hogy mindig az éppen „aktív” marker paraméterei legyenek kéznél (azaz az „alapszámítógéphez” képest sok `Get`-es műveletre lesz szükség.)

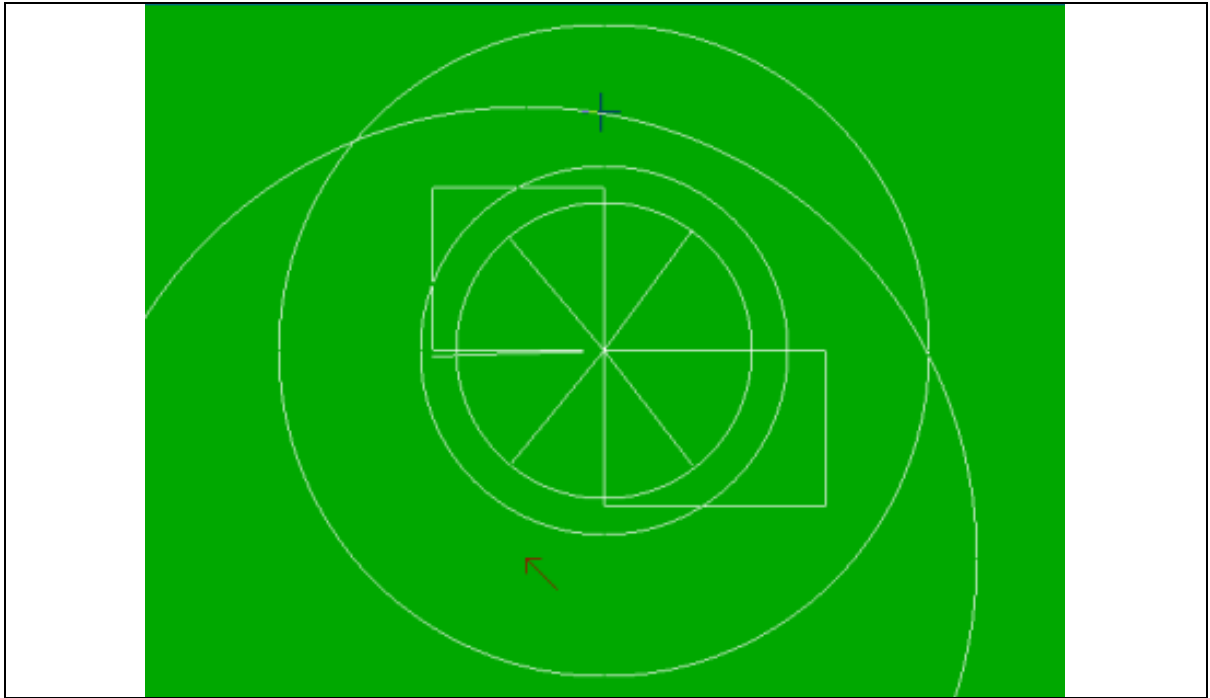
Számítógépi grafika – Markerek

2. Folytassuk a következővel: a két markert egy „**gumiszalag-szerű**” vonal állandóan kösse össze, amelyet ENTER-rel rögzíteni is lehet. Így egyenes szakaszokból ábrát is lehet rajzolni. A „gumiszalag” és a rögzített vonal színe (esetleg mintázata is) legyen eltérő. (Kipróbálhatja: [Marker3.exe.](#))



2. ábra. „Vonalazás” közben.

3. Bővítsük azzal, hogy **további fajta alakzatokat** is lehessen rajzolni. Pl. „K”-val kört, „D”-vel „dobozt”, és természetesen „V”-vel egyenes vonalat. Az aktuális alakzatot ENTER-rel lehet „rögzíteni”. A körív-rajzolás csak a Line-ra visszavezetve megy, ui. a CIRCLE-re a XORPut nem hat! Ezért az egyik korábbi [gyakorlaton](#) készített körív-rajzoló eljárást itt be kell építeni a programba. (Kipróbálhatja: [Marker4.exe](#).)



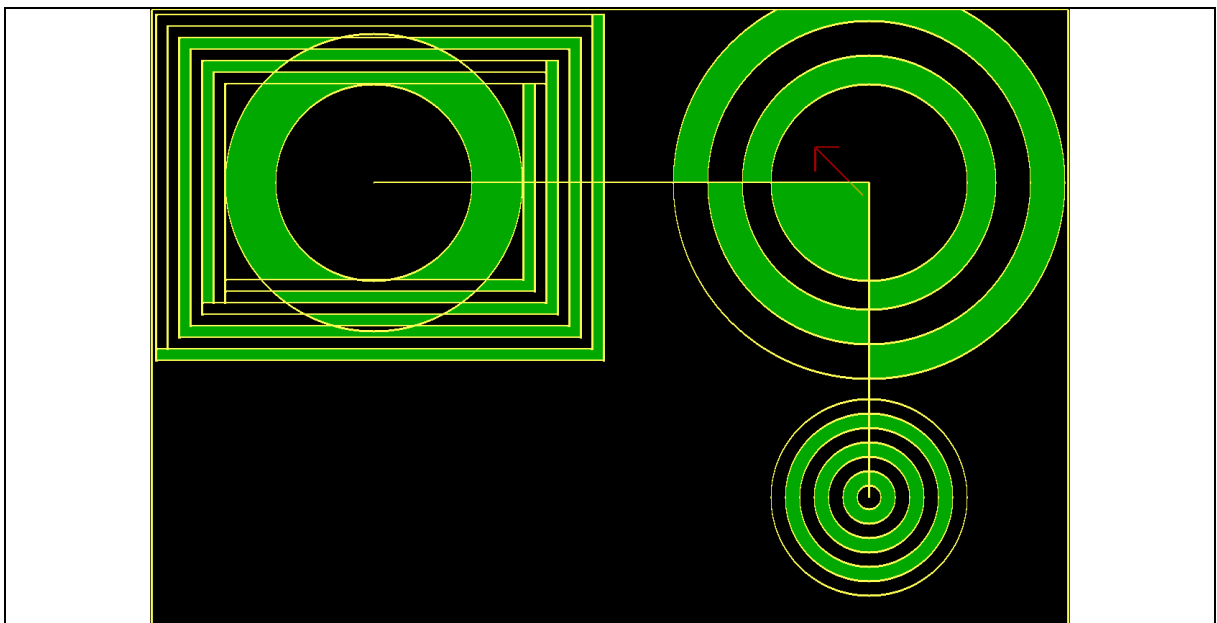
3. ábra. Rajzolás közben.

4. További bővítés lehetne a **tartományszínezés**. A lényeg, hogy a megrajzolt alakzatok vonal-színnel határolt tartományait kitöltő színnel lehessen színezni. Ehhez fel lehet használni a Turbo Grafika `FloodFill(x,y,vonalSzin)` eljárását.⁴ Természetesen nincs akadálya annak sem, hogy a korábbi [gyakorlat](#) tartományfestő eljárásainak valamelyikét itt felhasználhassuk.

Problémák a tartományszínezéskor:

- a **markerek ne látsszanak**, mert a színezést megzavarják, pl.:
 - eltűnik a tartományba eső részük (és az újrarajzoláskor csonkultnak),
 - miattuk nem lesz zárt a tartomány (és „kifolyik” a szín a tartományból)
 - **egyetlen** aktív marker, ezért nincs „gumi alakzat”
 - a végleges ábrát a **tényleges vonalszínnel** rögzíteni kell,
 - azaz ne `XORPut` módon, hanem `NormalPut` módon kerüljön kirajzolásra, de
 - a **markerek** e közben **se látsszanak**, mert a marker-újrarajzolás nem kívánt nyomot hagy.
- A háttérszint célszerű a legsemlegesebb színre (feketére) változtatni.

Kipróbálhatja: [Marker5.exe](#).



4. ábra. Rajzolás közben.

⁴ L. még a következő Turbo Grafika fogalmakat:

Type `FillPatternType`;

Procedure `GetFillPattern(pattern), SetFillPattern(pattern, rajzSzin)`.