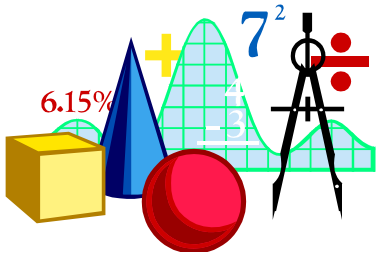
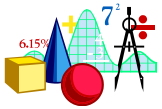
A grafika programozás alapjai

- *A grafika programozás módozatai*
- *A képernyő koordinátarendszere*
- *A graphics.h header-fájl fontosabb függvényei*
- *Mintaprogram*



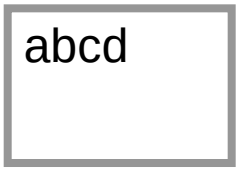
- ***Grafika programozás módozatai***



A ma már csak ablakban elérhető szöveges (text) üzemmód helyett a monitorokat grafikus üzemmódban használjuk.

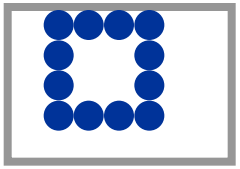
Szöveges üzemmód:

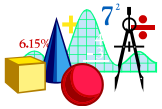
80 oszlop, 25 sor, beállított kódtábla karakterei.



Grafikus üzemmód:

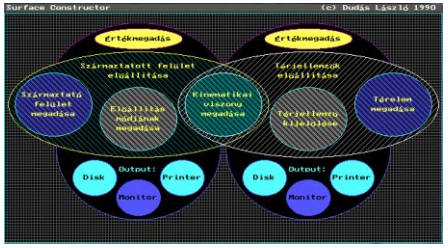
Képpontonkénti (pixelenkénti) rajzolás. Egy képernyőpont a színmélységtől függő számú szín egyikével gyújtható ki.
Minden rajzi elem kapcsolódó képpontok együtteseként jelenik meg.

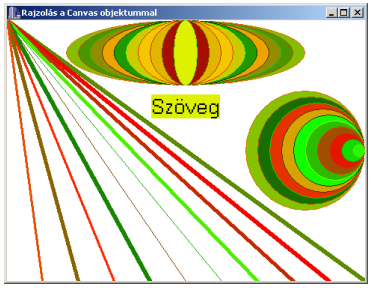


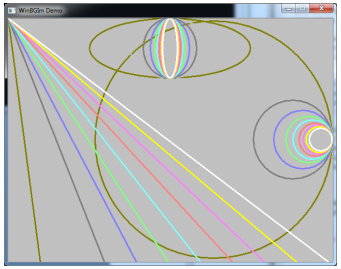


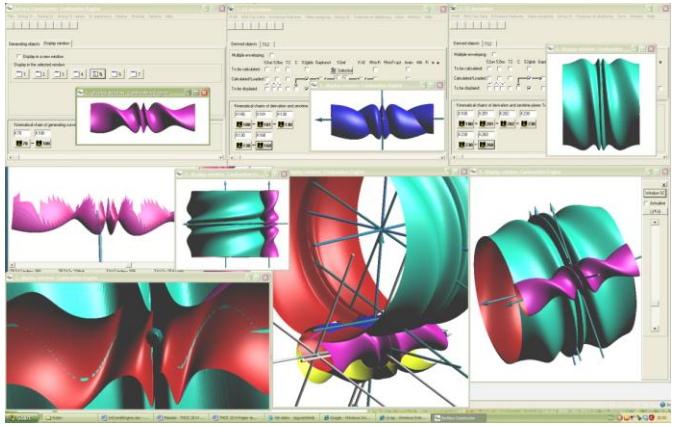
A C programozási nyelv használatával elérhető módozatok:

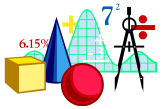
- *alapvetően 2D vonalas grafika DOS szinten, a graphics.h könyvtár függvényeivel,*
- *alapvetően 2D vonalas grafika Windows ablakban, a graphics.h könyvtár függvényeivel,*
- *2D grafika Windows környezetben a Canvas függvényeivel,*
- *3D árnyalt grafika Windows környezetben, OpenGL, DirectX grafikus könyvtárakkal.*











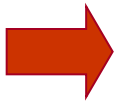
Grafika létrehozása a képernyőmemóriába való írást jelenti. A képernyőmemória tartalma alapján a videoelektronika a képernyőképet másodpercenként 60-130 alkalommal újrarajzolja, frissíti.

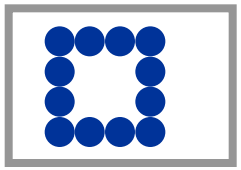
A grafikának ilyen hardverközeli programozása helyett mi a magasszintű C függvényeket használjuk.

Mozgást, animációt a kettő, vagy több képernyőnyi képernyőmemóriák tartalmának megjelenítését váltva jeleníthetünk meg. (Double Buffer)

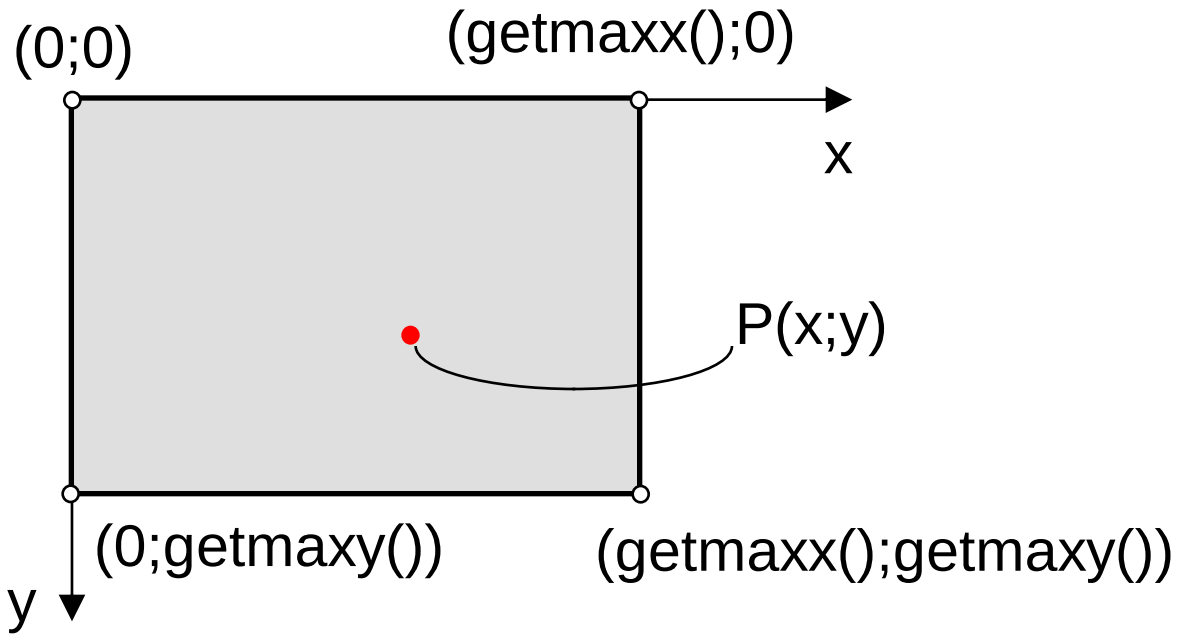
Videokártya RAM

01111000100100010010000111100

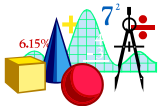




- ***A képernyő koordinátarendszere***
  (graphics.h használata esetén)



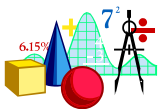
- ***Monitortípusok, meghajtókártyák, meghajtóprogramok, módok***



A monitornak, a meghajtókártyának (vezérlőkártya) és a meghajtószoftvernek (interfész) egymáshoz illeszkednie kell! Ezt a beállítást elvégzi az operációs rendszer.

A C magasszintű grafikus függvényeit, melyek a <graphics.h> header-fájlban találhatók, a különféle monitorokhoz, ill. meghajtókártyákhoz a meghajtószoftverek illesztik.

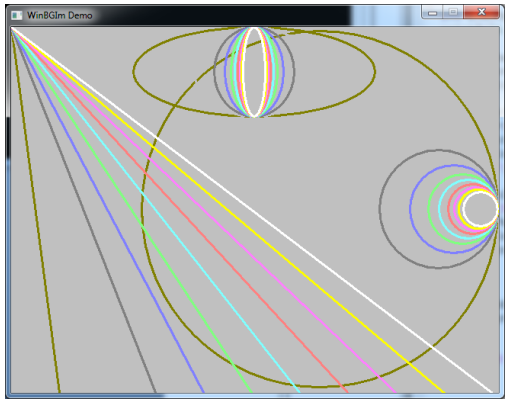
Mód van a monitor és a meghajtókártya alkotta grafikus hardver típusának automatikus meghatározására, detektálására.
Egy adott grafikus hardver többféle módon beállítható a felbontás, a színek száma szempontjából. Ezt a beállítást a grafikus mód (Gm) változó mutatja.



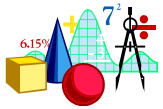
A graphics.h fájl fontosabb függvényei

Lehetőségek:

- *grafikus rendszer vezérlése,*
- *rajzolás*
- *kifestés, kitöltés,*
- *szövegkiírás,*
- *képernyőrész, ill. ablakkezelés,*
- *lapozás*
- *színkezelés*
- *hibakezelés*
- *állapotlekérdezés.*



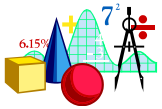
Lásd még: http://www.cs.colorado.edu/~main/cs1300/doc/bgi/index.html



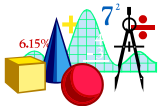
Grafikus rendszer kezelése:

- *detectgraph();* — *Grafikus hardver detektálás (DOS)*
- *initgraph();* — *Grafikus mód beállítás (DOS)*
- *initwindow();* — *Grafikus mód beállítás (Windows, CodeBlocks)*
- *closegraph();* — *Grafikus mód lezárása*
- *setgraphmode();* — *Más grafikus mód beállítása, ablaktörlés.*

initwindow();

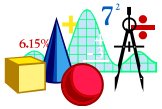


closegraph();



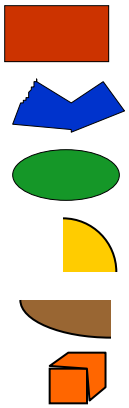
Rajzolás:

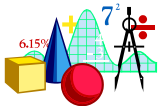
- *moveto(); moverel();* — *Grafikus kurzor pozicionálása*
- *putpixel();* — *Pont rajzolása*
- *line(); lineto(); linerel();* — *Szakaszrajzolás*
- *drawpoly();* — *Sokszögrajzolás*
- *circle(); arc();* — *Kör, körcikk rajzolás*
- *ellipse();* — *Ellipszisrajzolás*
- *rectangle();* — *Téglalaprajzolás*
- *setlinestyle();* — *Vonaljellemző beállítás*
- *stb.*



Kifestés, kitöltés:

- *setfillstyle();* — *Kitöltőminta+szín kiválasztás*
- *setfillpattern();* — *Kitöltőminta+szín megadás.*
- *bar();* — *Kitöltött téglalap rajzolás*
- *fillpoly();* — *Kitöltött sokszög rajzolás*
- *fillellipse();* — *Kitöltött ellipszis rajzolás*
- *pieslice();* — *Tortaszelet rajzolás*
- *sector();* — *Elliptikus szektor rajzolás*
- *bar3d();* — *Téglatest rajzolás*



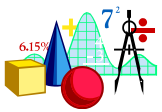


Szövegkiirás:

- *outtext(); outtextxy();* — *Szövegkiirás*
- *settextjustify();* — *Szövegigazítás beállítás*
- *settextstyle();* — *Karaktertípus, irány, méret* (.chr kell)

Képernyőrész, ill. ablakkezelés:

- *imagesize();* — *Képernyőrész helyfoglalása*
- *getimage();* — *Képernyőrész elmentése*
- *setviewport();* — *Ablak létrehozás+vágás*
- *clearviewport();* — *Ablak törlés háttérszínre*
- *putimage();* — *Képernyőrész kirakása*
- *stb.*



Lapozás:

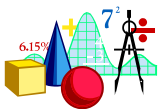
- *setactivepage(),* — *Érvényes lap kijelölése*
- *setvisualpage(),* — *Látható lap kijelölése*
- *cleardevice(),* — *Képernyő törlése*

Színkezelés:

- *setcolor(),* — *Vonalszín megadása*
- *setbkcolor(),* — *Háttérszín megadása*

Hibakezelés:

- *graphresult(),* — *Hibakódot ad*
- *grapherrormsg(),* — *Hibaszöveget ad a kódhoz*

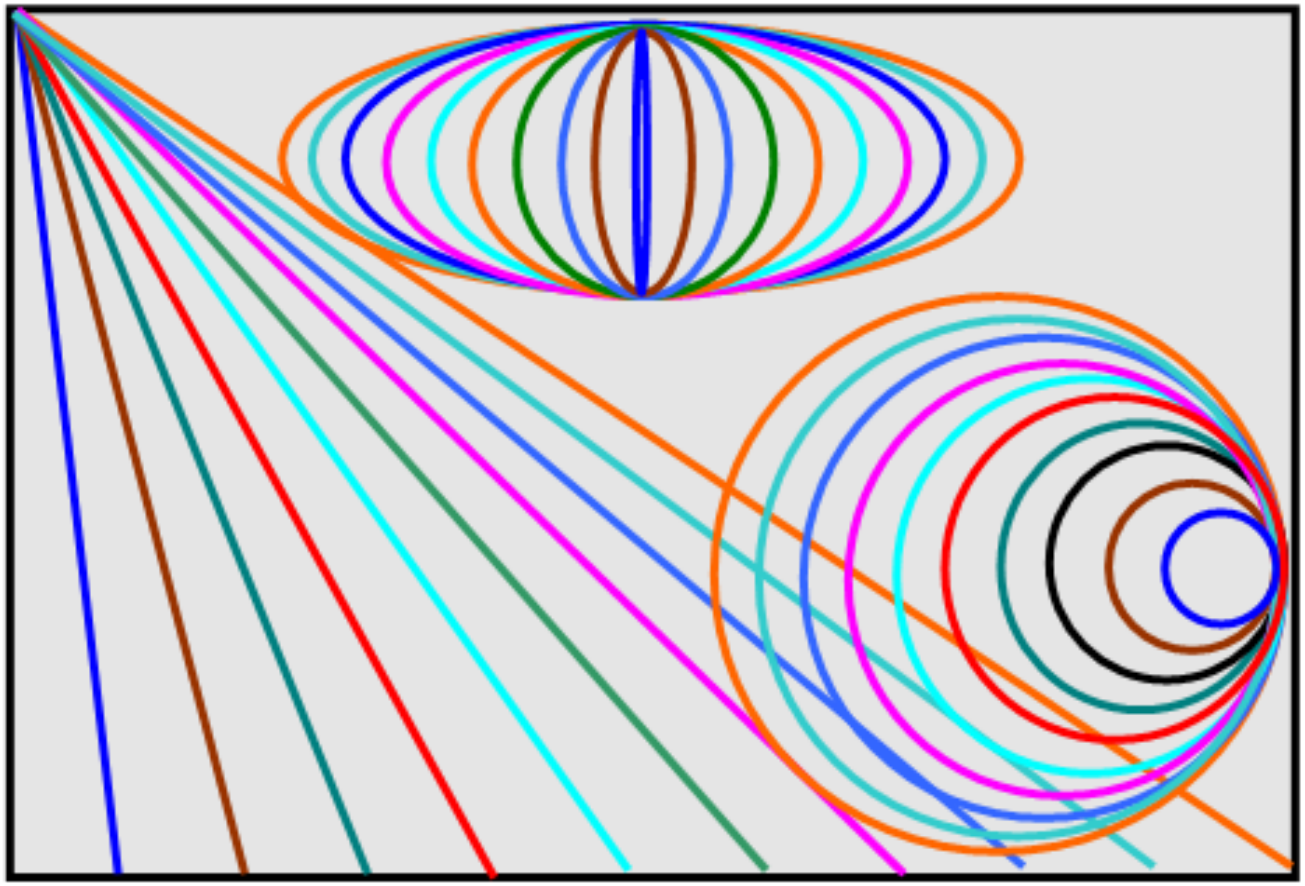


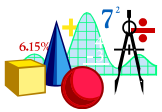
Állapotlekérdezés:

Minden beállítható érték lekérdezhető. A fontosabbak:

- *getgraphmode();* — *A mód kódszámát adja*
- *getmaxmode();* — *A max. mód számát adja*
- *getdrivername();* — *A grafikus hardver nevét adja*
- *getx(); gety();* — *A grafikus kurzor aktuális pozícióját adja*
- *getmaxx(); getmaxy();* — *A képernyő felbontását adja*
- *getcolor();* — *Az akt. vonalszínt adja*
- *getpixel();* — *Egy adott pont színét adja*
- *getmaxcolor();* — *A használható legnagyobb színkódot adja*
- *getbkcolor();* — *A háttérszín kódját adja.*

- ***Mintaprogram***
  Készítsünk az alábbi ábrához hasonló ábrát rajzoló programot!

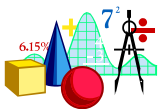
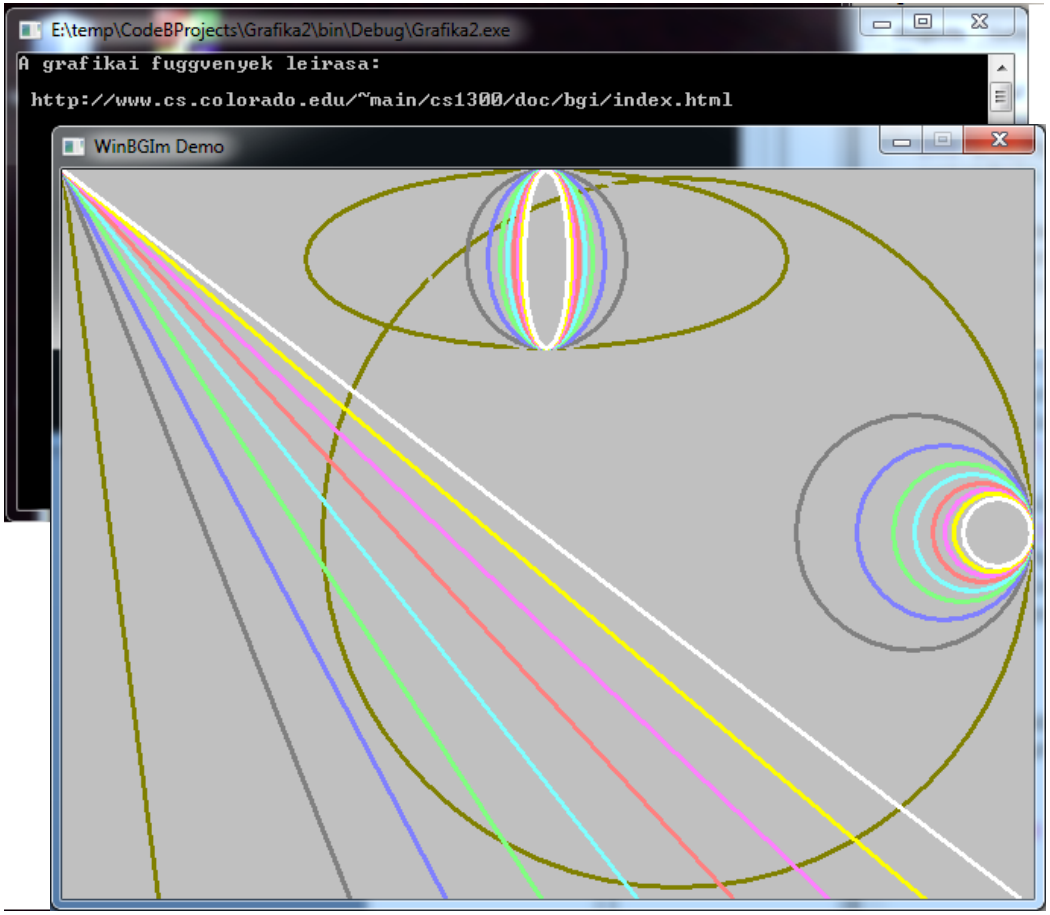




```
#include <stdio.h>
#include <graphics.h>

int main( )                                          /* a CodeBlocks C program főfüggvénye */
{
    int i,R,MaxX,MaxY;
    initwindow( 640 , 480 , "  WinBGIm Demo " );    /* 1024 , 768 is mûködik  */
    setbkcolor(LIGHTGRAY);                           /* háttérszín elõválasztás: világosszürke*/
    cleardevice();                                   /* háttér törlése háttérszínre */
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);          /* vonalstílus beállítása */
    MaxX= getmaxx(); MaxY= getmaxy();                /* ablakméret lekérdezése */

    for (i=1; i<=10; i++)
    {
        setcolor(i+5);                               /* aktuális vonalszín beállítása */
        line(0,0,(MaxX / 10)*i, MaxY);               /* szakaszrajzolás */
        R= (MaxY-10) / (2*i);                        /* körsugár beállítása */
        circle(MaxX-R, MaxY / 2, R);                 /* körrajzolás */
        ellipse(MaxX / 2, MaxY / 8, 0, 360, MaxX / (4*i), MaxY / 8);  /* ellipszis rajzolás */
    }
   while( !kbhit() );                                /* várakozás billentyűlenyomásig */
   closegraph( );                                    /* grafikus üzemmód/ablak bezárása */
   return( 0 );
}
```



A CodeBloks fejlesztőkörnyezetben futtatott program képe