

Kompass

Erich Brenner, Herbert Gursky

Das Programm sollte vorerst einen Magneten durch die Tastatur am Bildschirm (Textmodus) bewegen können, und eine Magnetnadel sollte die Position des Magneten anzeigen.

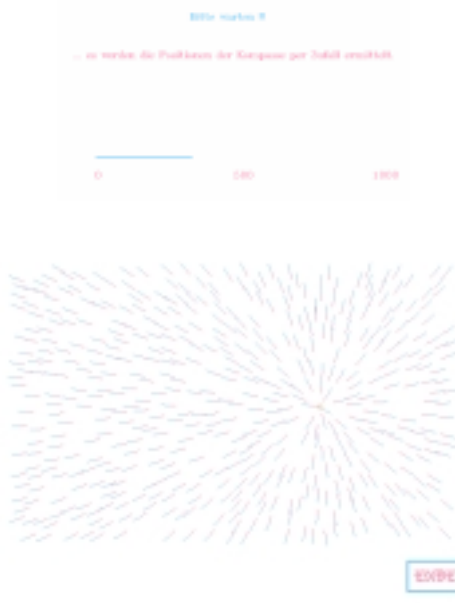
Das Programm wurde im Laufe des Unterrichtes immer mehr ausgebaut, und es wurden folgende Features hinzugefügt:

- steuern des Magneten durch die Maus und beenden des Programmes durch einen Mausklick auf das 'ENDE' - Feld
- anzeigen im Graphikmodus
- eröffnen eines Projektes: aufteilen des Programmes in eine Header - Datei und in zwei CPP - Dateien
- programmieren einer polymorphen Struktur
- frei wählbare Anzahl von Kompaßnadeln (1 - 1000)

Die Maus wurde mit Hilfe der Interrupt 33 -Funktion angesteuert und ihr Status eingelesen. Ein Container-Array `ort[]` enthält bis zu 1000 Magnete, repräsentiert durch den Basisklassenzeiger `ORT` sowie die Maus (=Magnet) an der Position `ort[0]`. Jeder Kompass kennt die Position des Magneten über den Maus-Pointer, der über den Konstruktor einem Kompass mitgegeben wird. Die virtuelle Funktion `bewege()` behandelt Magneten und Kompass polymorph.

Ausführungsbeispiel

Als Anzahl der Kompassse wurde 500 eingegeben.



Bei 500 Kompassen rechnet ein 486 DX 66 ungefähr 10 Sekunden, bis er alle gesetzt hat.

```

/*****
/*          KOMPASS.H          */
/* Herbert Gursky, Erich Brenner */
/* Datum: 3.3.98              Version: 1*/
*****/
    
```

```

#define PI 3.141592654
#define MOUSE 0x33

class Ort
{
protected:
    int x;
    int y;
public:
    Ort (int, int);
    int get x() const { return x; }
    int get_y() const { return y; }
    virtual void bewege() = 0;
};
    
```

```

class Maus:

class Magnet: public Ort
{
    Maus *maus;
    float z;
public:
    Magnet
        (Maus *m, int px=0, int py=0, float pz=0);
    float get z() const { return z; }
    void zeich();
    void loesch(float zalt);
    void bewege();
};
    
```

```

class Maus: public Ort
{
    int t; //Taste
public:
    Maus (int px=0, int py=0);
    void zeich();
    void loesch(int xalt=0, int yalt=0);
    void bewege();
    void abfrage();
};
    
```

```

/*****
/*          KOMPASS.CPP          */
/* Erich Brenner, Herbert Gursky */
/* Datum: 17.3.98              */
/* Version: 1.1                */
*****/
    
```

```

#include <conio.h>
#include <iostream.h>
#include <math.h>
#include <dos.h>
#include <graphics.h>
#include <STDLIB.h>
#include "kompass.h"
    
```

```

Ort::Ort (int px=0, int py=0) //Klasse Ort
{
    x=px;
    y=py;
}
    
```

```

Magnet::Magnet
(Maus *m, int px, int py, float pz) :
Ort(px,py)
{
    maus = m;
    //Klasse Magnet ( + Klasse Ort)
    z=pz;
}
    
```

```

Maus::Maus (int px, int py) : Ort(px, py)
//Klasse Maus ( + Klasse Ort)
{
    union REGS regs;

    setcolor(4);
    //zeichnet das Symbol 'ENDE'
    setlinestyle(0,0,3);
    rectangle(550,420,620,460);
    setcolor(2);
    settextrstyle(GOTHIC_FONT,HORIZ_DIR,2);
    outtextxy(560,425,"ENDE");
    setlinestyle(0,0,1);
    
```

```

regs.x.ax = 0; // Maustreiber geladen?
int86(MOUSE, &regs, &regs);

if (regs.x.ax != 0xFFFF)
{
    cout <<
        "Der Moustreiber ist nicht geladen!";
    getch();
    closegraph();
    exit (1);
}
    
```

```

regs.x.ax = 2; // Maus ausschalten
int86(MOUSE, &regs, &regs);
    
```

```

abfrage();
zeich();
}
    
```

```

void Magnet::loesch(float zalt)
//löscht einen Magneten
{
    int a,b,c,d;
    float e,f;
    e=cos (zalt);
    f=sin (zalt);
    a=x+10*e;
    b=y+10*f;
    c=x-10*e;
    d=y-10*f;
    setcolor(0);
    line (x,y,a,b);
    line (x,y,c,d);
}
    
```

```

void Maus::loesch(int xalt,int yalt)
//löscht die Maus
{
    setcolor(0);
    circle(xalt,yalt,3);
}
    
```

```

void Magnet::zeich()
//zeichnet einen Magneten
{
    int a,b,c,d;
    float e,f;
    e=cos (z);
    f=sin (z);
    a=x+10*e;
    b=y+10*f;
    c=x-10*e;
    d=y-10*f;
    setcolor(2);
    line (x,y,a,b);
    setcolor(4);
    line (x,y,c,d);
}
    
```

```

void Maus::zeich()
//Zeichnet die Maus
{
    setcolor(9);
    circle(x,y,3);
}
    
```

```

void Maus::abfrage()
// Abfragen Cursorposition
{
    union REGS regs;
    // linke Taste gedrückt: t = 1
    // rechte Taste gedrückt: t = 2
    regs.x.ax = 3;
    // beide Tasten gedrückt: t = 3
    int86(MOUSE, &regs, &regs);
    // keine Taste gedrückt: t = 0
    t = regs.x.bx;
    x = regs.x.cx;
    y = regs.x.dx;
}

void Maus::bewege() //bewegen der Maus
{ int xalt,yalt;
  xalt=x;
  yalt=y;
  abfrage();
  if (xalt!=x || yalt!=y)
  { loesch(xalt,yalt);
    if (x>540 && x<630 && y>410 && y<470)
    { setcolor(4);
      setlinestyle(0,0,3);
      //zeichnet das Symbol 'ENDE',
      rectangle(550,420,620,460);
      //wenn der Cursor dieses überschreibt
      setcolor(2);
      settextstyle(GOTHIC_FONT,HORIZ_DIR,2);
      outtextxy(560,425,"ENDE");
      setlinestyle(0,0,1);
    }
    zeich();
  }

  if
  (x>554 && x<616 && y>424 && y<456 && t==1)
  { closegraph();
    exit(0);
  }
}

void Magnet::bewege()
//ausrichten der Magnete
{
    float dx, dy, zalt;

    zalt=z;

    dx=maus->get_x()-x;
    dy=maus->get_y()-y;

    if (dx==0)
    {
        if (dy>0) z=PI/2;
        if (dy==0) z=0;
        if (dy<0) z=(3*PI)/2;
    }
    if (dx!=0)
    {
        z=atan(dy/dx);
        if (dx<0) z=z+PI;
    }

    if (zalt!=z)
    {
        loesch(zalt);
        zeich();
    }
}

/*****
/*      KOMPMAIN.CPP      */
/*  Erich Brenner, Herbert Gursky  */
/*  Datum: 17.3.98      */
/*  Version: 1.1      */
*****/

#include <conio.h>
#include <iostream.h>
#include <math.h>
#include <dos.h>
#include <STDLIB.h>
#include <graphics.h>
#include "kompass.cpp"

#define MAGNETE 1001 //1000 Magnete + 1 Maus

```

```

void main (void)
{
    int Magnete,Karte,Modus,w=0,c;
    int a[MAGNETE],b[MAGNETE];

    clrscr();
    gotoxy(10,10);
    cout << "Wieviele Kompass (max. 1000) ?";
    gotoxy(64,10); //Auswählen der Kompass
    cin >> Magnete;
    if (Magnete > 1000 || Magnete < 1)
        Magnete=100;

    clrscr();
    registerbgidriver(EGAVGA_driver);
    //laden der Grafiktreiber
    registerbgifont(triplex_font);
    registerbgifont(gothic_font);
    initgraph (&Karte, &Modus, "");
    setcolor(4);
    settextstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    outtextxy (240,130,"Bitte warten !!!");
    setcolor(2);
    outtextxy (30,200,
        "Positionen der Kompass zufällig");
    outtextxy (70,420,"0");
    outtextxy (320,420,"500");
    outtextxy (570,420,"1000");
    c=sqrt(Magnete);
    c=250/c;

    setcolor(4);
    setlinestyle(0,0,3);

    moveto(70,400);

    randomize();
    //ermittelt die Position der Kompass
    //per Zufall
    for (int j=0;j<=Magnete;j++)
    {
        llneto(70+j/2,400);
        a[j]=30+random(580);
        b[j]=30+random(360);
        for (int k=0;k<j;k++)
        {
            for (int v=(-c);v<c;v++)
            {
                if ((a[j]+v)==a[k])
                {
                    for (int u=(-c);u<c;u++)
                    {
                        if ((b[j]+u)==b[k]) w++;
                    }
                }
            }
        }
        if (w>0)
        { w=0;
          j--;
        }
    }
    cleardevice();

    Maus* maus = new Maus(0,0);
    //erzeugen der Maus
    Ort* ort[MAGNETE+1];
    ort[0] = maus;

    for (int mi=1; mi<(Magnete+1); mi++)
    //erzeugen der Kompass
    ort[mi] =
        new Magnet(maus,a[mi],b[mi],0);

    do
    {
        for (int m=0; m<(Magnete+1); m++)
        //Bewegen der Maus
        //und Ausrichten der Kompass
        {
            ort[m]->bewege();
        }
    }
    while (!kbhit());
    //aussteigen mit Tastendruck
    closegraph();
}

```

Murphys Spielregeln

Highscore

1. Es fehlt Dir immer ein Punkt zum neuen Highscore.
2. Hast Du ein Spiel so lange gespielt, bis Dich niemand schlagen kann, wird ein Freund es zum ersten Mal spielen und problemlos Platz eins der Highscoreliste erreichen.
3. Bist Du der Beste, wird sich niemand dafür interessieren.

Erkenntnisse über Computerspiele

1. Das Computerspiel, das Du Dir kaufst, ist nicht den Preis wert, den es gekostet hat.
2. Kein Computerspiel ist so gut wie das Bild auf der Verpackung.
3. Eine gute Kritik in einer Spielzeitschrift besagt nur, daß das Spiel dem Redakteur gefallen hat. Du wirst immer einen anderen Geschmack haben als er.
4. Das merkst Du aber erst nach dem Kauf.
5. Dein größter Rivale wird immer ein besseres Spiel haben als Du.

Kinder an Deinem Computer

Das einzige, wofür sich heranwachsenden Kinder an Deinem Computer interessieren, sind die zahlreichen Spieleraubkopien, die sie während Deiner Abwesenheit auf die Festplatte kopieren.

Diese Tätigkeit wird jeden existierenden Virus im Umkreis von 400 Kilometern auf Deinen Computer befördern.

Homecomputer

Egal, wofür sich jemand einen Computer kauft. Spätestens nach einer Woche wird er 26 Stunden pro Tag Space Invaders damit spielen.