

```

program movimiento_en_campo_gravitatorio;
{C lculo en tiempo real movimiento de ocho asteroides}
{situados entre Marte y Jupiter}
{Influencia gravitatoria de Jupiter e inestabilidad en el cinturon}
{tiempo aconsejable const t=10000}
uses
  Crt, Graph;
const
  t=10000;           {constantes generales}
  pi=3.141592653598793;
  g=6.67392E-11;
  mb=1.9891E+30;     {masa del sol}
  ma=1.8987E+27;     {probar25.1. masa de jupiter}
  e=150/6E+11;       {factor de escala: 2 veces la rbita de
Jupiter}
var
  GraphDriver : integer; { The Graphics device driver }
  GraphMode   : integer; { The Graphics mode value }
  MaxX, MaxY  : word;    { The maximum resolution of the screen }
  ErrorCode   : integer; { Reports any graphics errors }
  OldExitProc : Pointer; { Saves exit procedure address }

  {asteroides}
  aas,aaas,abas,aasx,aasy,raspa0,raspax0,raspay0,raspb0:array[1..8]of
real;
  raspbx0,raspby0,ras0,rasx0,rasy0,vasx0,vasy0,phyas0,psias:array[1..8]of
real;

  aju,axju,ayju,rju0,rxju0,ryju0,vxju0,vyju0,phyju0,psiju:real; {jupiter}
  time:real;

  {$F+}
procedure MyExitProc;
begin
  ExitProc := OldExitProc; { Restore exit procedure address }
  CloseGraph;              { Shut down the graphics system }
end; { MyExitProc }
{$F-}

procedure Inicia_modografico;
{ Inicializa el modo grafico e informa de los errores que puedan ocurrir}
var
  InGraphicsMode : boolean; { Flags initialization of graphics mode }
  PathToDriver   : string;   { Stores the DOS path to *.BGI & *.CHR }
begin
  { when using Crt and graphics, turn off Crt's memory-mapped writes }
  DirectVideo := False;
  OldExitProc := ExitProc;           { save previous exit proc }
  ExitProc := @MyExitProc;          { insert our exit proc in chain }
}
  PathToDriver := '';
  repeat

  {$IFDEF Use8514}           { check for Use8514 $DEFINE }
    GraphDriver := IBM8514;
    GraphMode := IBM8514Hi;
  {$ELSE}

```

```

    GraphDriver := Detect;                { use autodetection }
{$ENDIF}

    InitGraph(GraphDriver, GraphMode, PathToDriver);
    ErrorCode := GraphResult;             { preserve error return }
    if ErrorCode <> grOK then               { error? }
    begin
        Writeln('Graphics error: ', GraphErrorMsg(ErrorCode));
        if ErrorCode = grFileNotFound then { Can't find driver file }
        begin
            Writeln('Enter full path to BGI driver or type <Ctrl-Break> to
quit:');
            Readln(PathToDriver);
            Writeln;
        end
        else
            Halt(1);                      { Some other error: terminate }
        end;
    until ErrorCode = grOK;
    MaxX := GetMaxX;                      { Get screen resolution values }
    MaxY := GetMaxY;
end; { Initialize }

procedure dibuja_los_ejes;
begin
    { ejes }
    SetLineStyle(DottedLn,0,NormWidth);
    SetColor(4);
    Line(MaxX div 2,MaxY div 2 +230,MaxX div 2,MaxY div 2 -230);{
vertical }
    Line(MaxX div 2 -310,MaxY div 2,MaxX div 2 +310,MaxY div 2);{
horizontal }
    { sol }
    SetLineStyle(SolidLn,0,NormWidth);setColor(14);
    Line(MaxX div 2,MaxY div 2 +6,MaxX div 2,MaxY div 2 -6);
    Line(MaxX div 2 -4,MaxY div 2,MaxX div 2 +4,MaxY div 2);
    PutPixel(MaxX div 2 +6, MaxY div 2, 15);
    PutPixel(MaxX div 2 -6, MaxY div 2, 15);
    PutPixel(MaxX div 2, MaxY div 2 +9, 15);
    PutPixel(MaxX div 2, MaxY div 2 -9, 15);
    PutPixel(MaxX div 2 +1, MaxY div 2 +1, 15);
    PutPixel(MaxX div 2 +1, MaxY div 2 -1, 15);
    PutPixel(MaxX div 2 -1, MaxY div 2 +1, 15);
    PutPixel(MaxX div 2 -1, MaxY div 2 -1, 15);
    PutPixel(MaxX div 2 +3, MaxY div 2 +3, 15);
    PutPixel(MaxX div 2 +3, MaxY div 2 -3, 15);
    PutPixel(MaxX div 2 -3, MaxY div 2 +3, 15);
    PutPixel(MaxX div 2 -3, MaxY div 2 -3, 15);
end;

procedure presentacion;
begin
    SetTextStyle(TriplexFont,0,3);SetTextJustify(CenterText,CenterText);
    SetColor(15);OutTextXY(320,70,'GRAVITY.EXE');
    SetColor(15);
    OutTextXY(540,410,'Pulsa Enter');
    Readln;ClearViewPort;
end;

```

```

procedure DatosIniciales0;
begin
    rasx0[1]:=6.27936E+11; rasy0[1]:=0; {asteroide 1}
    ras0[1]:=6.27936E+11;
    vasx0[1]:=0; vasy0[1]:=14500;
    rxju0:=0; ryju0:=7.78412E+11;           {jupiter}
    rju0:=ryju0;
    vxju0:=-13069; vyju0:=0;
end;
procedure DatosIniciales;
var vel: real;
begin
    vel:=15850;
    rasx0[1]:=6.27936E+11; rasy0[1]:=0; {asteroide 1}
    ras0[1]:=6.27936E+11;
    vasx0[1]:=0; vasy0[1]:=14500;

    rasx0[1]:=5.27936E+11; rasy0[1]:=0; {asteroide 1}
    ras0[1]:=5.27936E+11;
    vasx0[1]:=0; vasy0[1]:=vel;

    rasx0[2]:=0; rasy0[2]:=-5.27936E+11; {asteroide 2}
    ras0[2]:=5.27936E+11;
    vasx0[2]:=vel; vasy0[2]:=0;

    rasx0[3]:=-5.27936E+11; rasy0[3]:=0; {asteroide 3}
    ras0[3]:=5.27936E+11;
    vasx0[3]:=0; vasy0[3]:=-vel;

    rasx0[4]:=0; rasy0[4]:=5.27936E+11; {asteroide 4}
    ras0[4]:=5.27936E+11;
    vasx0[4]:=-vel; vasy0[4]:=0;

    rasx0[5]:=5.27936E+11*sin(pi/4); rasy0[5]:=5.27936E+11*sin(pi/4);
    ras0[5]:=sqrt(2)*5.27936E+11;
    vasx0[5]:=-vel*sin(pi/4); vasy0[5]:=vel*sin(pi/4);

    rasx0[6]:=-5.27936E+11*sin(pi/4); rasy0[6]:=5.27936E+11*sin(pi/4);
    ras0[6]:=sqrt(2)*5.27936E+11;
    vasx0[6]:=-vel*sin(pi/4); vasy0[6]:=-vel*sin(pi/4);

    rasx0[7]:=-5.27936E+11*sin(pi/4); rasy0[7]:=-
5.27936E+11*sin(pi/4);
    ras0[7]:=sqrt(2)*5.27936E+11;
    vasx0[7]:=vel*sin(pi/4); vasy0[7]:=-vel*sin(pi/4);

    rasx0[8]:=5.27936E+11*sin(pi/4); rasy0[8]:=-5.27936E+11*sin(pi/4);
    ras0[8]:=sqrt(2)*5.27936E+11;
    vasx0[8]:=vel*sin(pi/4); vasy0[8]:=vel*sin(pi/4);

    rxju0:=0; ryju0:=7.78412E+11;           {jupiter}
    rju0:=ryju0;
    vxju0:=-13069; vyju0:=0;
end;
procedure calcula_phy;
var n: integer;
begin
    for n:=1 to 8 do

```

```

        begin
            if (rasy0[n]>0)and(rasx0[n]>0)then
phyas0[n]:=arctan(rasy0[n]/rasx0[n]);
            if (rasy0[n]>0)and(rasx0[n]<0)then
phyas0[n]:=arctan(rasy0[n]/rasx0[n])+pi;
            if (rasy0[n]<0)and(rasx0[n]<0)then
phyas0[n]:=arctan(rasy0[n]/rasx0[n])+pi;
            if (rasy0[n]<0)and(rasx0[n]>0)then
phyas0[n]:=arctan(rasy0[n]/rasx0[n]);
        end;
    end;
procedure calcula_psi(n:integer);
var psia,psib,axa,aya,axb,ayb:real;
begin
    if ((rasy0[n]-ryju0)>0)and((rasx0[n]-rxju0)>0)then
psia:=arctan((rasy0[n]-ryju0)/(rasx0[n]-rxju0))+pi;
    if ((rasy0[n]-ryju0)>0)and((rasx0[n]-rxju0)<0)then
psia:=arctan((rasy0[n]-ryju0)/(rasx0[n]-rxju0));
    if ((rasy0[n]-ryju0)<0)and((rasx0[n]-rxju0)<0)then
psia:=arctan((rasy0[n]-ryju0)/(rasx0[n]-rxju0));
    if ((rasy0[n]-ryju0)<0)and((rasx0[n]-rxju0)>0)then
psia:=arctan((rasy0[n]-ryju0)/(rasx0[n]-rxju0))+pi;
    axa:=aaas[n]*cos(psia);aya:=aaas[n]*sin(psia);
    if (rasy0[n]>0)and(rasx0[n]>0)then
psib:=arctan(rasy0[n]/rasx0[n])+pi;
    if (rasy0[n]>0)and(rasx0[n]<0)then
psib:=arctan(rasy0[n]/rasx0[n]);
    if (rasy0[n]<0)and(rasx0[n]<0)then
psib:=arctan(rasy0[n]/rasx0[n]);
    if (rasy0[n]<0)and(rasx0[n]>0)then
psib:=arctan(rasy0[n]/rasx0[n])+pi;
    axb:=abas[n]*cos(psib);ayb:=abas[n]*sin(psib);
    aasx[n]:=axa+axb;aasy[n]:=aya+ayb;
end;
procedure dibuja_jupiter(x,y,c:integer);
begin
    PutPixel(MaxX div 2 +x, MaxY div 2 +y, 1);
    SetLineStyle(SolidLn,0,NormWidth);setColor(c);
    Line(MaxX div 2 +x, MaxY div 2 +y+6,MaxX div 2 +x,MaxY div 2 +y-6);
    Line(MaxX div 2 +x-4,MaxY div 2 +y,MaxX div 2 +x+4,MaxY div 2 +y);
    PutPixel(MaxX div 2 +x+6, MaxY div 2 +y, c);
    PutPixel(MaxX div 2 +x-6, MaxY div 2 +y, c);
    PutPixel(MaxX div 2 +x, MaxY div 2 +9 +y, c);
    PutPixel(MaxX div 2 +x, MaxY div 2 -9 +y, c);
    PutPixel(MaxX div 2 +x+1, MaxY div 2 +y+1, c);
    PutPixel(MaxX div 2 +x+1, MaxY div 2 +y-1, c);
    PutPixel(MaxX div 2 +x-1, MaxY div 2 +y+1, c);
    PutPixel(MaxX div 2 +x-1, MaxY div 2 +y-1, c);
    PutPixel(MaxX div 2 +x+3, MaxY div 2 +y+3, c);
    PutPixel(MaxX div 2 +x+3, MaxY div 2 +y-3, c);
    PutPixel(MaxX div 2 +x-3, MaxY div 2 +y+3, c);
    PutPixel(MaxX div 2 +x-3, MaxY div 2 +y-3, c);
end;
procedure dibuja_asteroide(x,y,c:integer);
begin
    PutPixel(MaxX div 2 +x, MaxY div 2 +y, c);
    PutPixel(MaxX div 2 +x+2, MaxY div 2 +y, c);
    PutPixel(MaxX div 2 +x, MaxY div 2 +y+2, c);

```

```

        PutPixel(MaxX div 2 +x-2, MaxY div 2 +y, c);
        PutPixel(MaxX div 2 +x, MaxY div 2 +y-2, c);
    end;
procedure borra_asteroide(x,y,c:integer);
begin
    PutPixel(MaxX div 2 +x, MaxY div 2 +y, c);
    PutPixel(MaxX div 2 +x+2, MaxY div 2 +y, 0);
    PutPixel(MaxX div 2 +x, MaxY div 2 +y+2, 0);
    PutPixel(MaxX div 2 +x-2, MaxY div 2 +y, 0);
    PutPixel(MaxX div 2 +x, MaxY div 2 +y-2, 0);
end;
procedure calculos2;
var n:integer;
{Movimiento de los tres asteroides}
begin
    {jupiter}
    dibuja_jupiter(round(rxju0*e*0.8),-round(ryju0*e),0);
    phyju0:=phyju0-t*vxju0/rju0;if phyju0>2*pi then phyju0:=phyju0-
2*pi;
    rxju0:=rju0*cos(phyju0);ryju0:=rju0*sin(phyju0);
    dibuja_jupiter(round(rxju0*e*0.8),-round(ryju0*e),7);
    {asteroides}
    for n:=1 to 8 do
    begin
        raspa0[n]:=sqrt((rasx0[n]-rxju0)*(rasx0[n]-rxju0)+(rasy0[n]-
ryju0)*(rasy0[n]-ryju0));
        raspb0[n]:=sqrt(rasx0[n]*rasx0[n]+rasy0[n]*rasy0[n]);
        aaas[n]:=g*ma/(raspa0[n]*raspa0[n]);abas[n]:=g*mb/(raspb0[n]*raspb0[n]);
        calcula_psi(n);
        borra_asteroide(round(rasx0[n]*e*0.8),-round(rasy0[n]*e), n);

        rasx0[n]:=rasx0[n]+vasx0[n]*t+(aasx[n]*t*t)/2;rasy0[n]:=rasy0[n]+vasy0[n]
*t+(aasy[n]*t*t)/2;
        dibuja_asteroide(round(rasx0[n]*e*0.8),-round(rasy0[n]*e), 15);
        vasx0[n]:=vasx0[n]+aasx[n]*t;vasy0[n]:=vasy0[n]+aasy[n]*t;
    end;
    time:=time+t;
    TextColor(8);gotoxy(67,2);write(round(time/31557600),' years');
end;
procedure calculos1;
{Movimiento de un asteroide tan solo}
begin
    {jupiter}
    dibuja_jupiter(round(rxju0*e*0.8),-round(ryju0*e),0);
    phyju0:=phyju0-t*vxju0/rju0;if phyju0>2*pi then phyju0:=phyju0-
2*pi;
    rxju0:=rju0*cos(phyju0);ryju0:=rju0*sin(phyju0);
    dibuja_jupiter(round(rxju0*e*0.8),-round(ryju0*e),7);
    {asteroides azul}
    raspa0[1]:=sqrt((rasx0[1]-rxju0)*(rasx0[1]-rxju0)+(rasy0[1]-
ryju0)*(rasy0[1]-ryju0));
    raspb0[1]:=sqrt(rasx0[1]*rasx0[1]+rasy0[1]*rasy0[1]);
    aaas[1]:=g*ma/(raspa0[1]*raspa0[1]);abas[1]:=g*mb/(raspb0[1]*raspb0[1]);
    calcula_psi(1);
    borra_asteroide(round(rasx0[1]*e*0.8),-round(rasy0[1]*e), 1);

```

```

rasx0[1]:=rasx0[1]+vasx0[1]*t+(aasx[1]*t*t)/2; rasy0[1]:=rasy0[1]+vasy0[1]
*t+(aasy[1]*t*t)/2;
    dibuja_asteroide(round(rasx0[1]*e*0.8),-round(rasy0[1]*e), 15);
    vasx0[1]:=vasx0[1]+aasx[1]*t; vasy0[1]:=vasy0[1]+aasy[1]*t;

    time:=time+t;
    TextColor(8); gotoxy(67,2); write(round(time/31557600), ' years');
end;
begin
    Inicia_modos_grafico; Presentacion;
    {movimiento de un asteroide tan solo}
    dibuja_los_ejes;
    TextColor(4); gotoxy(43,11); Write('Marte');
    TextColor(7); gotoxy(67,4); Write('Jupiter');
    TextColor(1); gotoxy(67,20); Write('Asteroide');
    DatosIniciales0; time:=0; phyju0:=pi/2; {pi/2 guachi, pi, 3*pi/2}
    vasy0[1]:=14500; {14000_elipses, 14500_419anios, 15000, 15200_chachi}
    SetColor(8); Ellipse(MaxX div 2, MaxY div 2,
0,360,round(195*0.8),195);
    SetColor(4); Ellipse(MaxX div 2, MaxY div 2, 0,360,round(57*0.8),57);
    repeat calculos1 until keypressed;
    ReadLn; ClearViewPort;
    {movimiento de los tres asteroides}
    dibuja_los_ejes;
    TextColor(4); gotoxy(38,13); Write('Marte');
    TextColor(7); gotoxy(67,4); Write('Jupiter');
    TextColor(6); gotoxy(67,17); Write('Asteroide 1');
    TextColor(4); gotoxy(67,18); Write('Asteroide 2');
    TextColor(1); gotoxy(67,19); Write('Asteroide 3');
    TextColor(5); gotoxy(67,20); Write('Asteroide 4');
    TextColor(7); gotoxy(67,21); Write('Asteroide 5');
    TextColor(3); gotoxy(67,22); Write('Asteroide 6');
    TextColor(2); gotoxy(67,23); Write('Asteroide 7');
    TextColor(15); gotoxy(67,24); Write('Asteroide 8');
    DatosIniciales; time:=0; phyju0:=pi/2;
    SetColor(8); Ellipse(MaxX div 2, MaxY div 2,
0,360,round(195*0.8),195);
    SetColor(4); Ellipse(MaxX div 2, MaxY div 2, 0,360,round(57*0.8),57);
    repeat calculos2 until keypressed;
end.

```