

# 第十章 图形程序设计基础

## 【教学目标】

基本掌握在 TC 环境中的作图方法  
图形函数的使用方法和像素概念  
图形与文本方式的交换  
了解图形模式的种类

## 【教学内容】

显示模式和图形初始化;  
设置视口和清除视口;  
线型和填充模式、基本绘图函数;  
设置前景色和背景色;  
曲线方程的图形化表示等。

## 【教学重点和难点】

图形与文本的模式交换;  
图形初始化和绘图参数的准确使用;  
色彩设置;  
平面曲线方程的图形表示  
动画在 C 语言中的实现

## 【问题的提出】

C 中对数据处理能力很强，是否同样具有较强的图形处理能力，怎样处理？

### 1. 平面曲线方程的图形表示

【例 10.1】利用 C 的强大绘图功能，动态绘制一条平面曲线——小蜜蜂。

```
#include<malloc.h>
#include<stdio.h>
#include<graphics.h> /* 包含定义所有绘图函数的头文件——绘图时必须有 */
#include<math.h>      /* 包含数学处理函数，以计算三角函数值 */

main()
{
    int gdriver, gmode=0,i,x,y;
    gdriver=0;
    initgraph(&gdriver, &gmode, "e:\\tc\\bgi"); /* 图形初始化 */
    setbkcolor(0); /* 设置背景色为黑色 */
    setcolor(RED); /* 前景色为红色 */
    settextstyle(1,0,0); /* 设置文本以 3 倍字型，水平方向，缺省方式显示 */
    outtextxy(50,50,"BEE"); /* 在（50，50）坐标点输出字符串 BEE */
}
```

```

for(i=0;i<2280;i++)
{
    x=(cos(i*1./360)+cos(3*i*1./360))*100+300;    /* 计算 X 坐标值 */
    y=(sin(i*1./360)+sin(5*i*1./360))*100+250;    /* 计算 Y 坐标值 */
    putpixel(x, y, 14);                            /* 以黄色 (14) 在 (X, Y) 坐标点处画点 */
    delay(4000);                                    /* 延时函数, 或 sleep(1); */
}
getch();
closegraph( );                                    /* 关闭图形方式, 回到文本方式工作 */
}

```

#### 说明:

C 绘图时, 首先需要包含图形头文件 `graphics.h`。

绘图前, 应该使用函数 `initgraph(int* driver,int *mode,char *path)`对图形进行初始化, 再设置前景色、背景色等。

绘曲线图时, 首先确定曲线方程, 计算坐标点, 使用绘点函数 `putpixel(int x,int y,int color)`画点。

图上如果需要显示文本, 则使用设置文本格式函数 `settextstyle(int font,int direction,int charsize)`设置, 利用 `outtextxy(int x,int y,char *string)`在 (X, Y) 位置显示串 `string`。

为了看清楚绘图过程, 可以在每绘制出一个点后添加延时函数 `delay( )`或 `sleep( )`进行延时处理。

绘图结束, 应该关闭图形方式 `closegraph( )`, 以便于进行其他操作。

## 2. 动画在 C 语言中的实现

【例 10.2】使用 C 的绘图功能, 制作实现一个简易时钟示意。

```

#include<graphics.h>
#include<stdio.h>
void main( )
{int driver,mode;
int i,j,k,start,end,start1,end1,start2,end2,maxx,maxy;
driver=DETECT;    /* 自动检测计算机的图形方式 , 值是 DETECT 或 0 */
mode=0;          /* 以可以使用的最大分辨率模式, 其值为 0 */
initgraph(&driver,&mode,"e:\\tc");
                /* 图形初始化, 图形驱动程序在 e:\\tc 目录下, 名称是 BGI */

start1=0;                /* 指针转角值初始化 */
end1=-2;
start2=0;
end2=-2;
start=0;
end=-2;
maxx=getmaxx( );        /* 获取最大 X 坐标值 */
maxy=getmaxy( );        /* 获取最大 Y 坐标值 */
for(k=0; ;k++)
{ start2=30; /* 时针每次以 1 小时为单位顺时针方向转动一次, 转角为 360/12=30 */
  end2=30;
  for(j=1;j<=60;j++)
    /*死循环, 使时钟一直走下去*/

```

```

{ start1-=6; /* 分针每次以 1 分钟为单位顺时针方向转动一次，转角为 360/60=6 */
  end1-=6;
for(i=1;i<=60;i++)
{ setfillstyle(SOLID_FILL,i%14); /* 设置以 i%14 为色号，实体填充 */
pieslice(maxx/2,maxy/2,start,end,200);
/* 绘制半径为 200、圆心角为 2 度的扇形作为秒针 */
pieslice(maxx/2,maxy/2,start1,end1,150);
/* 绘制半径为 150、圆心角为 2 度的扇形作为分针 */
pieslice(maxx/2,maxy/2,start2,end2,100);
/* 绘制半径为 150、圆心角为 2 度的扇形作为时针 */
circle(maxx/2,maxy/2,200);
/* 以点(maxx/2,maxy/2)为圆心，半径为 200，绘制时钟外圈圆 */
gotoxy(maxx/2-200,maxy/2); /* 将绘图笔移动到点 (maxx/2-200,maxy/2) */
line(maxx/2-200,maxy/2,maxx/2+200,maxy/2);
/* 从点(maxx/2-200,maxy/2)到点(maxx/2+200,maxy/2)划线，表示 3 点与 9 点位置 */
gotoxy(maxx/2,maxy/2-200); /* 将绘图笔移动到点 (maxx/2,maxy/2-200) */
line(maxx/2,maxy/2-200,maxx/2,maxy/2+200);
/* 从点(maxx/2,maxy/2-200)到点(maxx/2,maxy/2+200)划线，表示 0 点与 6 点位置 */
start-=6; /* 秒针每次以 1 秒为单位顺时针方向转动一次，转角为 360/60=6 */
end-=6;
sleep(1); /* 每循环一次，延时 1 秒 */
clearviewport();
/* 清除视口，进行下一次绘图，整个循环以绘图、延时、清除来实现图形的运动 */
}
}
}
restorecrtmode(); /* 恢复文本显示模式函数 */
}

```

### 说明：

实现动画的基本方法是：绘图、延时、清除，移动位置再绘图、延时、清除，以此类推。为使动画效果更佳，可以使每帧在运动时，改变一次颜色。

### 【小结】

绘制图形时，首先应该包含图形头文件 `graphics.h`，测试计算机的图形显示模式，再对图形进行初始化。

绘图前应该考虑图形元素中的线型、线宽、颜色、填充方式等。

使用画图函数，绘点、线、曲线、多边形等时，重点掌握函数的使用方法和参数，绘制函数曲线时，应该首先确定函数，通过循环计算曲线上对应的点坐标值，使用画点函数画点，连续的点表示出曲线。绘点过程中，加入延时，可以看出绘图的过程。

制作动画首先应该考虑图形的组成、运动的过程与方式。

动画的基本思路是：配合循环，绘图、延时、清除，再绘图、延时、清除，以此实现图形的运动。