توابع گرافيكي در محيط Turbo C++ 3.0 Dos

intgraph

اين تابع براي تغيير حالت برنامه از حالت متني به حالت گرافيكي استفاده مي كند. شكل كلي اين دستور به صورت زير مي باشد:

void initgraph(int *driver , int *mode , char * bgiaddress)

در اين الگو driver به مبدل گرافيكي اشاره مي كند كه مي خواهيم تبديل مورد نظر را انجام دهد.كه مي توانيم مقدار آن را با كلمه كليدي DETECT به طور خودكار پيدا كرد ونيازي به حفظ كردن آنها نيست.

هر حالت driver چند mode دارد كه البته مشخص كردن آنها تاثير زيادي در خروجي ندارد و فقط ممكن است Resolation خروجي را تغيير دهد.

آرگومان سوم آدرس دقيق فايل مبدل گرافيكي را مشخص مي كند كه مي تواند به صورت نسبي يا دقيق صورت گيرد.در حالت دقيق آدرس را از Root كه در ويندوز My Computer مي باشد مي نويسند مثلاً"C:\\TC\\BGI" . توجه نيد كه از دو backslash استفاده كرديم چون مانند دستور Cout از يكي از آنها صرف نظر مي شود.در تعيين آدرس نسبي آدرس نسبت به محلي كه فايل اجرايي كامپايلر (TC.exe) در آن قرار دارد مي نويسيم.مثلاً " " يعني فايل مورد نظر در همان شاخهٔ كامپايلر قرار دارد.

مثال:

```
int driver=DETECT;
int mode=0;
initgraph( &driver , &mode ,  "\\BGI" );
```

اين دستورات driver را به طور خودكار تشخيص مي دهد كه در اغلب كامپيوتر هاي امروزي IBM8514 است و mode آنرا صفر قرار مي دهد.بعد تابع initgraph سعي ميكند از شاخه نسبي BGI تابع مبدل را پيدا كرده و بارگذاري كند ، در اين حالت resolation صفحه 640*480 مي باشد.

graphresult

اين تابع اغلب بعد از initgraph براي كنترل ورود موفقيت آميز به محيط صورت مي گيرد، اگر خروجي آن grOk باشد ورود موفقيت آميز بوده است.

شكل كلي اين تابع به صورت زير مي باشد:

int graphresult ( );

مثال:

```
int driver=DETECT;
int mode=0;
initgraph( &driver , &mode ,  "" );
if ( graphresult() != grOk ){
        cout<<"Graphic error!!";
        getch();
        exit(0);
        }
```

در اين مثال اگر ورود موفقيت آميز نباشد دستورات داخل حلقه اجرا مي شود و برنامه با يك خروجي مناسب پايان مي پذيرد.

line

شكل كلي اين تابع به صورت زير مي با شد:

void line (int x1 , int y1 , int x2 , int y2)

اين تابع خطي را با رنگي كه مشخص مي شود از (x1,y1) تا (x2 , y2) رسم مي كند

lineto

خطي را از موقعيت جاري تا نقطه دلخواه رسم مي كند و داراي الگوي زير است:

```
void lineto(int x , int y);
```

---

linerel

خطي را از موقعيت جاري تا نقطه اي كه مختصات آن نسبت به خط جاري داده شده است رسم مي كند.

```
void linerel(int deltax , int deltay );
```

---

setcolor

اين تابع مقدار رنگي را كه با آن خط ها و شكل ها رسم مي شوند مشخص مي كند كه عددي بين ۰ تا ۱۵ است.البته اگر عدد بزرگتر داده شود باقيمانده آن به ۱۶ محاسبه مي شود.اين اعداد طبق جدول زير مشخص مي شوند

.

شماره رنگ	رنگ
۰	سياه
۱	آبي
۲	سبز
۳	كبود
۴	قرمز
۵	بنفش
۶	قهوه اي
۷	خاكستري روشن
۸	خاكستري تيره
۹	آبي روشن
۱۰	سبز روشن
۱۱	كبود روشن
۱۲	قرمز روشن
۱۳	بنفش روشن
۱۴	زرد
۱۵	سفيد

شكل كلي دستور به صورت زير مي باشد:

```
void setcolor( int color)
```

مثال:

```
setcolor(4);
line(200,200,400,200);
line(400,200,300,200);
setcolor(1);
line(200,200,300,200);
```

در بالا ابتدا رنگ قرمز تعيين مي شود و سپس دو خط با رنگ قرمز كشيده مي شود و سپس رنگ آبي انتخاب ميشود و خط سوم كشيده مي شود كه اين سه خط تشكيل يك مثلث مي دهند.

getcolor

اين تابع رنگي را كه بر نامه فعلا با آن كار مي كند بر مي گرداند و در جا هايي كه رنگ به صورت متغير بوده يا از ثوابت استفاده مي كنيم كاربرد دارد.

```
int getcolor()
```

setbkcolor و getbkcolor

setbkcolor براي رنگ كردن پس زمينه از آن استفاده مي شود.

```
void setbkcolor(int color)
```

كه ورودي آن همان شماره رنگ هايي است كه در setcolor مشخص شد.
براي مثال setcolor(1) رنگ پس زمينه را آبي مي كند واگر دراين محيط شكلي را با رنگ آبي رسم كنيم آنرا نمي بينيم.
رنگ پيش فرض سياه مي باشد.
gebktcolor هم مثل getcolor كار مي كند و شماره رنگ پس زمينه را بر مي گرداند.

```
int getbkcolor()
```

---

getpixel

اين تابع رنگ پيكسل موجود در نقطه دلخواهي از صفحه نمايش را مشخص مي كند و داراي شكل كلي زير است:

```
int getpixel(int x , int y);
```

مثال:

```
setcolor( getcolor(110 ,110) );
```

دستور فوق رنگ انتخابي را رنگ نقطه (110,110) قرار مي دهد.

putpixel

اين تابع يك نقطه را بارنگ دلخواه در صفحه نمايش نشان مي دهد.

```
void putpixel(int x ,int y ,int color)
```

كه نقطه (x,y) را با رنگ color رنگ آميزي مي كند.

---

rectangle

```
void rectangle(int x1,int y1 , int x2, int y2);
```

اين تابع مستطيلي تو خالي را ايجاد ميكند كه نقطه ي بالاي آن y1 ونقطهٔ پايين آن y2 و نقطه سمت راست آن x1 ونقطه سمت چپ آن x2 مي باشد.

مثال: كد زير مستطيلي را ۱۰۰ پيكسل به سمت چپ حر كت مي دهد.

```
int bkcolor=getbkcolor();
for(i=100 ; i<200 ; ){
        setcolor(bkcolor);
        rectangle(100+i,100,200,200+i);
        i++;
        setcolor(4);
        rectangle(100+I,100,200,200+i);
        }
```

circle

```
void circle ( int x , int y , int radios )
```

دايره اي به مركز (x,y) و شعاع radios رسم مي نمايد.

مثال: دايره اي توپر به شعاع x با استفاده از رسم دايره هاي متوالي تو در تو ايجاد مي كند.

```
while( x>0 ){
        circle(xcenter , ycenter , x--);
        }
```

drawpoly

براي رسم چندضلعي ها به كار مي رود و داراي الگوي زير است:

void drawpoly (int numpoints , int *points)

در اين الگو numpoints تعداد اضلاع چندضلعي را مشخص مي كند و points به آرايه اي اشاره مي كند كه مختصات گوشه هاي چند ضلعي در آن قرار دارند. طول اين آرايه حداقل بايد دو برابر گوشه هاي چند ضلعي منظور شود.در اين آرايه هر گوشه با x و y مخصوص به خود تعريف مي شود كه ابتدا مقادير x و سپس مقادير y در آرايه منظور مي شود.

مثال:

```
int shape[]={10,10,100,80,200,200,350,90,0,0};
drawpoly(5 , shape );
```

setfillstyle

اين تابع براي تعيين رنگ و سبك پر شدن شكل هاي گرافيكي توپر مثل تابع setcolor كار مي كند و دو آرگومان مي گيرد. اولي حالت پر كردن و دومي رنگ پر كردن صفحه است.

void setfillstyle (int pattern , int color)

كه رنگ مي تواند يكي از رنگ هاي قبلي باشدو براي انتخاب حالت از جدول زير استفاده مي كنيم.

معادل عددي	مفهوم
۰	پر كردن شكل با رنگ زمينه
۱	پر كردن شكل با رنگي پر رنگ
۲	پر كردن شكل توسط خطوط
۳	پر كردن شكل توسط / روشن
۴	پر كردن شكل توسط /
۵	پر كردن شكل توسط \
۶	پر كردن شكل توسط \ روشن
۷	پر كردن شكل توسط هاشورزني
۸	پر كردن شكل توسط هاشور زني روشن
۹	پر كردن شكل با كاراكتر هاي گوناگون
۱۰	پر كردن شكل توسط نقاط تو خالي
۱۱	پر كردن شكل توسط نقاط تو خالي
۱۲	الگوي انتخابي توسط توابع ديگر

از اين تابع فقط مي توان براي تعيين رنگ شكل هاي توپر استفاده كرد و نمي توان براي line و rectangle و circle استفاده كرد.

---

bar , bar3d

bar مثل rectangle است فقط شكل آن توپر است و رنگ آن با setfillstyle مشخص مي شود.

void bar(int right , int top , int left , int bottom)

bar3d براي رسم مكعب است كه چهار آرگومان آن مثل مستطيل است و آرگومان پنجم براي تعيين ميزان فرورفتگي است و اگر عدد منفي وارد شود برامدگي خواهيم داشت.آرگومان پنجم اگر مساوي صفر باشد قسمت بالاي مكعب حذف خواهد شد.

void bar3d(int right , int top , int left , int bottom , int depth , int topflag)

مثال :

```
setfillstyle(1,4);
bar(0,0,100,100);
getch();
setfillstyle(1,getbkcolor());  ///   setfillstyle(0,4);
bar(0,0,100,100);
```

مستطيلي كاملاً قرمز در صفحه ايجاد مي كند و سپس آن را پاك مي كند.

---

fillpoly

براي رسم چند ضلعي توپر استفاده مي شود و رنگ آن با setfillstyle مشخص مي شود و كاركرد آن مثل drawpoly است.

outtextxy و outtext

outtext متني را در خروجي با فونت انتخاب شده نمايش مي دهد و متني را از جايي نمايش ميدهد كه در آنجا هستيم ولي outtextxy دو آرگومان x و y مي گيرد و متن مورد نظر را از آن نقطه نمايش مي دهد. براي نمايش اعداد به صورت گرافيكي با اين تابع بايد آنها را به رشته تبديل كنيم.

```
void outtext ( char * str )
void outtextxy ( int x , int y , char *str)
```

setfontstyle

براي تعيين نوع فونت مورد استفاده در تابع هاي بالا مورد استفاده قرار مي گيرد و سه آرگومان مي گيرد كه آرگومان اول Font است و يكي از اعداد صفر تا سيزده است و آرگومان دوم جهت است كه اگر صفر باشد از چپ به راست و اگر يك باشد از بالا به پايين است.و آرگرمان آخر size بزرگي متن مي باشد كه از صفر شروع مي شود و براي فونت هاي گوناگون متفاوت است.

```
void settsxtstyle ( int font , int direction ,int size )
```

---

cleardevice

صفحه را پاك مي كند.

```
void cleardevice();
```

---

closegraph

از محيط گرافيكي خارج ميشود و به محيط text باز مي گردد و تمام حافظه گرفته شده توسط گرافيك را برمي گرداند.

```
void closegraph();
```

---

moveto

موجب انتقال موقعيت جاري به يك نقطه دلخواه مي شود و داراي الگوي زير است:

```
void moveto ( int x ,int y);
```

---

moverel

موجب انتقال مكان نما به يك نقطه نسبي مي شود.

```
void moverel ( int deltax , int deltay );
```

---

getx و gety و getmaxx و getmaxy

تابع هاي getx و gety مختصات مكان جاري را بر مي گردانند و تابع هاي getmaxx و getmaxy مقادير ماكزيمم صفحه نمايش را كه در تشخيص resolution كاربرد دارد.

```
int getmaxx()
int getmaxy()
int getx()
int gety()
```

ابراهيمي
آذر ۱۳۸۴