**BÀI 2: GIỚI THIỆU ĐỒ HỌA MÁY TÍNH**

**-Đồ họa máy tính** là 1 ngành khoa học Tin học chuyên nghiên cứu về các phương pháp và kỹ thuật để có thể  mô tả và thao tác trên các đối tượng của thế giới thực bằng máy tính.

* Về bản chất: đó là 1 quá trình xây dựng và phát triển các công cụ trên cả 2 lĩnh vực phần cứng và phần mềm hỗ trợ cho các lập trình viên thiết kế các chương trình có khả năng đồ họa cao.
* Với việc mô tả dữ liệu thông qua các hình ảnh và màu sắc đa dạng của nó, các chương trình đồ họa thường thu hút người sử dụng bởi tính thân thiện, dễ dùng,… kích thích khả năng sang tạo và nâng cao năng suất làm việc. Do vậy, đồ họa máy tính được ứng dụng trong nhiều lĩnh vực: giáo dục, thương mại..

**-Các kỹ thuật đồ họa:**

***\**Kỹ thuật đồ họa điểm*:*** ( ở bộ môn ĐHMT thì ta sẽ học kỹ thuật đồ họa này)

* Các hình ảnh được hiển thị thông qua từng Pixel ( từng mẫu rời rạc).
* Với kỹ thuật này, chúng ta có thể tạo ra, xóa hoặc thay đổi thuộc tính của từng Pixel của các đối tượng.
* Các hình ảnh được hiển thị như 1 lưới điểm rời rạc (grid), từng điểm đều có vị trí xác định được hiển thi với 1 giá trị nguyên biểu thị màu sắc hoặc độ sáng của điểm đó. Tập hợp tất cả các Pixel của grid tạo nên hình ảnh của đối tượng mà ta muốn biểu diễn.

**\*Kỹ thuật đồ họa vector:**

* Xây dựng mô hình hình học cho hình ảnh đối tượng, xác định các thuộc tính của mô hình hình học, sau đó dựa trên mô hình này để thực hiện quá trình tô trát để hiển thị từng điểm của mô hình,hình ảnh của đối tượng.
* Kỹ thuật này chỉ lưu trữ mô hình toán học của các thành phần trong mô hình hình học cùng với các thuộc tính tương ứng mà không cần lưu lại tòa bộ tất cả các Pixel của hình ảnh đối tượng.

**-Màn hình đồ họa:**

* Mỗi máy tính đều có 1 CARD dùng để quản lý màn hình, gọi là Video Adapter hay Graphics Adapter.
* Có nhiều loại adapter như: CGA, MCGA, EGA, VGA.
* Các adapter có thể làm việc ở 2 chế độ: văn bản ( Text Mode) và đồ họa (Graphics Mode).

**-Biểu diễn tọa độ:**

**-Vẽ điểm:**

* Trong các hệ thống đồ họa, 1 điểm (Pixel) được biểu thị bởi các tọa độ bằng số (nguyên)

Vd: trong mặt phẳng, một điểm là một cặp (x,y). Trong không gian 3 chiều, một điểm là bộ ba (x,y,z).

* Trên màn hình của máy tính, mỗi điểm là 1 vị trí trong vùng nhớ màn hình , dùng để lưu trữ các thông tin về độ sang của điểm tương ứng trên màn hình.
* Số điểm vẽ trên màn hình được gọi là độ phân giải của màn hình (320×200, 480×640, 1024×1024,..)

**Các hàm trong chế độ đồ họa (phần 1)**

Khi bắt đầu học về môn Đồ họa máy tính tại các trường đại học, cao đẳng , hầu như sinh viên sẽ được tiếp cận và tìm hiểu về các thuật toán vẽ đường trong đồ họa máy tính. Tuy nhiên, khi tiến hành làm project thì chúng ta nên sử dụng các hàm được hỗ trợ sẵn trong thư viện đồ họa. Và sau đây, mình xin nêu các hàm tiêu biểu và ý nghĩa của chúng.

Đầu tiên , ta nên biết các màu cơ bản .

**\*Hằng biểu diễn màu:**

Sau đó, ta sẽ đi xem code đồng hồ kim này và phân tích các hàm đồ họa sử dụng trong nó nhá.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113 | #include <graphics.h>  #include <iostream>  #include <math.h>  #define PI 3.14159265359  **using** **namespace** std;  **float** tinhgoc(**float** goc){  **return** (goc\*PI/180);  }    **void** vekhung(**int** xtam,**int** ytam,**int** bk){ //ham ve hinh tron tam (x,y) ban kinh 250  **int** d,x1,y1,x2,y2;      setlinestyle(0,0,CYAN);      setcolor(RED);      circle(xtam,ytam,bk); //      setcolor(YELLOW);      circle(xtam,ytam,bk+2); //      setcolor(LIGHTCYAN);      circle(xtam,ytam,bk+4); //ve mat dong ho      setcolor(MAGENTA);      d = 270;  **while**(d<=630){          x1 = xtam + (**int**)bk\***cos**(tinhgoc(d));          y1 = ytam + (**int**)bk\***sin**(tinhgoc(d));          //circle(x,y,10);          x2 = xtam + (**int**)(bk-20)\***cos**(tinhgoc(d));          y2 = xtam + (**int**)(bk-20)\***sin**(tinhgoc(d));          line(x1,y1,x2,y2);          delay(10);          d+=30;      }      d = 270;      setcolor(LIGHTMAGENTA);      setlinestyle(0,0,GREEN);  **while**(d<=630){          x1 = xtam + (**int**)bk\***cos**(tinhgoc(d));          y1 = ytam + (**int**)bk\***sin**(tinhgoc(d));          //circle(x,y,10);          x2 = xtam + (**int**)(bk-8)\***cos**(tinhgoc(d));          y2 = xtam + (**int**)(bk-8)\***sin**(tinhgoc(d));          line(x1,y1,x2,y2);          delay(10);          d+=6;      }  }    **void** kimgiay(**int** bk,**int** cl,**int** goc){      //ve kim giay co ban kinh bk, mau cl, goc bat dau  **int** xgiay,ygiay;        setlinestyle(0,0,BLUE);      setcolor(cl);      xgiay = 300 + (**int**)(bk-50)\***cos**(tinhgoc(goc));      ygiay = 300 + (**int**)(bk-50)\***sin**(tinhgoc(goc));      line(xgiay,ygiay,300,300);  }    **void** kimphut(**int** bk,**int** cl,**int** goc){      //ve kim phut  **int** xphut,yphut;      setlinestyle(0,0,CYAN);      setcolor(cl);      xphut = 300 + (**int**)(bk-100)\***cos**(tinhgoc(goc));      yphut = 300 + (**int**)(bk-100)\***sin**(tinhgoc(goc));      line(xphut,yphut,300,300);  }    **void** kimgio(**int** bk,**int** cl,**int** goc){      //ve kim gio  **int** xgio,ygio;      setlinestyle(0,0,CYAN);      setcolor(cl);      xgio = 300 + (**int**)(bk-140)\***cos**(tinhgoc(goc));      ygio = 300 + (**int**)(bk-140)\***sin**(tinhgoc(goc));      line(xgio,ygio,300,300);  }    **void** vekim(**int** bk,**float** d1,**float** d2,**float** d3){  **while**(!kbhit()){          kimgiay(bk,BLUE,d3); //ve kim giay ung voi goc d1  **if** (d3==d2) kimphut(bk,GREEN,d2); // may ve cac net chong len nhau nen phai ve láº¡i net cu da mat  **if** (d3==d1) kimgio(bk,RED,d1);          kimphut(bk,GREEN,d2);  **if** (d2==d1) kimgio(bk,RED,d1);          kimgio(bk,RED,d1);          delay(1000);          kimphut(bk,BLACK,d2); //xoa di net cu, ve net moi de tao di chuyen          kimgiay(bk,BLACK,d3);          kimgio(bk,BLACK,d1);          d3+=6; d2+=0.1; d1+=1/120;          setcolor(LIGHTCYAN); // dat mau cho tam          circle(300,300,RED);      }  }  **void** gocdau(**int** gio,**int** phut,**int** giay, **float** &d1, **float** &d2, **float** &d3){      d3 = giay\*6 + 270; // 12h ung voi goc 270      d2 = phut\*6 + giay\*0.1 + 270;      d1 = gio\*30 + phut\*0.5 + giay\*1/120 +270;  }  **int** main(){  **int** bk,gio,phut,giay;  **float** d1,d2,d3;      cout<<"CAM ON DA SU DUNG CHIEC DONG HO NAY :)))) ";      cout<<"\nHay lan luot nhap vao so gio, phut, giay\n (cach nhau boi phim space) roi enter: ";      cin>>gio>>phut>>giay;      gocdau(gio,phut,giay,d1,d2,d3);      initwindow(600,600);//mo cua so do hoa      outtextxy(10,10,"CLOCK in graphics");      bk = 250;      vekhung(300,300,bk);      vekim(bk,d1,d2,d3);        closegraph();//dong cua so do hoa  } |

Trong code trên, đã sử dụng một số hàm đồ họa sau :

**Hàm về màu :**

**setcolor(int color);**

là hàm đặt màu vẽ hiện tại. Màu số 0 là màu nền. Khoảng màu chạy từ 0 ..15

Ngoài ra, còn có 1 số hàm về màu sắc khác:

**getmaxcolor();**

là hàm trả về giá trị màu cao nhất cho thiết bị và chế độ hiện tại. ( theo bảng màu trên getmaxcolor() =15 ).

**setbkcolor(int color);**

là hàm xác lập màu nền. Khoảng màu chạy từ 0 ..15

**Hàm về text:**

**outtextxy(int x, int y, char far \*textstring);**

sẽ hiển thị nội dung của xâu textstring tại tọa độ (x,y). Do đó lệnh outtextxy có thể tương đương với việc thực hiện 2 lệnh: moveto(x,y) và outtext(textstring).

**settextstyle(int font, int direction, int charsize);**

thiết lập kiểu chữ hiện ra trên màn hình.

-Kiểu font chữ:

trong đó, 5 kiểu font đầu đã được định nghĩa const trong thư viện. Còn mấy kiểu dưới là do mình tìm hiểu và biết thêm được nhiêu đó .

-Chiều viết (direction):

* HORIZ\_DIR=0 : nằm ngang, từ trái qua phải
* VERT\_DIR=1 : thẳng đứng, từ dưới lên trên

-Kích thước chữ (charsize): nhận giá trị từ 1 đến 10, là hệ số phóng to chữ.

Ngoài ra , còn có 1 số hàm về text khác như:

**outtext(char far \*textstring);**

sẽ hiển thị nội dung của xâu textstring tại vị trí hiện tại của con trỏ đồ họa (thường sẽ ở vị trí (0,0).

**settextjusttify (int Horz, int Vert);**

là hàm xác định vị trí dòng văn bản viết ra ,theo 2 hàm outtext nói ở trên.

-Tham số Horz có thể nhận các giá trị sau:

* LEFT\_TEXT=0 : văn bản xuất hiện ở bên trái con trỏ đồ họa
* CENTER\_TEXT=1: văn bản xuất hiện với tâm là vị trí con trỏ đồ họa
* RIGHT\_TEXT=2: văn bản xuất hiện ở bên phải con trỏ đồ họa

-Còn Vert là tham số có thể nhận các giá trị sau:

* BOTTOM\_TEXT=0: văn bản xuất hiện ở phía trên con trỏ
* CENTER\_TEXT=1: văn bản xuất hiện ở quanh con trỏ
* TOP\_TEXT=2: văn bản xuất hiện ở phía dưới con trỏ

**Hàm  vẽ đường thẳng:**

**line(int x1, int y1, int x2, int y2);**

là hàm vẽ đường thẳng nối liền 2 điểm có tọa độ (x1,y1) và (x2,y2). hàm này không phụ thuộc vào vị trí ban đầu của con trỏ màn hình. Sau khi vẽ xong, con trỏ màn hình nằm ở vị trí điểm (x2,y2). Lưu ý là con trỏ đồ họa chỉ được xác lập vị trí hiện tại trên màn hình đồ họa, chứ không có điểm sáng nhấp nháy vật lý như con trỏ trong màn hình văn bản nha.

Ngoài ra còn 1 số hàm vẽ đường thẳng khác như:

**lineto(int x, int y);**

vẽ đường thẳng từ vị trí con trỏ hiện tại tới điểm (x,y).Sau khi vẽ xong, con trỏ màn hình nằm ở vị trí điểm (x,y)

**linerel(int dx, int dy);**

vẽ đường thẳng từ vị trí con trỏ hiện tại tới điểm (x + dx ,y +dy).Sau khi vẽ xong, con trỏ màn hình nằm ở vị trí điểm (x+dx ,y+dy)

**Hàm vẽ đường tròn:**

**circle(int x, int y, int R);** vẽ đường tròn có tâm tại (x,y) bán kính là R