Paper 102: Programming & Problem solving through C

Lecture-31:Graphics

Animation

- An image can be stored in the memory
- Such an image can be written back to the screen in the same or different location
- In case of animation, you may want to redraw the image quickly at different location
- Drawing the image from scratch at each position may be too time consuming
- The next example shows a ball bouncing in a rectangle box

Functions used for animation

- The image of the ball is created with circle()
- This is stored in memory with getimage() function

```
void far getimage(int left, int top,
                  int right, int bottom,
                  void far *addrBuff)
```

- where

  left, top, right, bottom coordinates of screen area to be saved

  addrBuff address of memory buffer for image

Functions used for animation

- How large should the array be to store the image?
  - Assume 2 bytes per pixel plus overhead of several hundred bytes
  - imagesize() function can be used for the same.

```
size=imagesize(int left, int top,int right,int bottom)
printf(" Size=%u", size);
```

- A better way is to find the size and allocate memory dynamically using malloc() function

Functions used for animation

- Image is later restored back at different place in memory
- This is done with putimage() function

```
void far putimage(int left, int top,
int right, int bottom, void far *addrBuff, int putop)
```

- where

  left, top, right, bottom coordinates of screen area to be saved

  addrBuff address of memory buffer for image

  putop interaction between old and new pixels

Putop constants

- The values of putop (operator) are

Value	Constant	Comment
0	COPY_PUT	Replaces the old image with new
1	XOR_PUT	XOR old and new images
2	OR_PUT	OR old and new images
3	AND_PUT	AND old and new images
4	NOT_PUT	Replaces with inverse of new image

- A new image is drawn with operator set to COPY_PUT. The effect of XOR_PUTing one image with the same image is to erase it.

Animation example

```
#include<graphics.h>
#include<conio.h>
#include<alloc.h>

void main(void)
{
    int driver=DETECT,
    gm,area,ch,x=25,y=25,xdirn=1,ydirn=1,maxx,m
    axy;
    char *buff;

    initgraph(&driver, &gm,"c:\\tc\\bgi");

    setcolor(WHITE);
    setfillstyle(SOLID_FILL, RED);

    circle(50,50,25);
    floodfill(50,50, WHITE);

    area=imagesize(25,25,75,75);
    buff=malloc(area);
    getimage(25,25,75,75,buff);
```

```
          maxx=getmaxx();
          maxy=getmaxy();
          rectangle(0,20,maxx,maxy);
          outtextxy(250,10,"Animation");

          while(1)
          {
                    if(kbhit())
                    {
                    ch=getch();
          /*if ENTER is hit reverse the
direction*/
                    if(ch=='\r')
                    {         xdirn
          *=-1;
                              ydirn
          *=-1;
                    }
                    else
                    {         if(ch==27)
                              break;
                    }
                    }


          putimage(x,y,buff, XOR_PUT);
          delay(0);
          x=x+(xdirn*1);
          y=y+(ydirn*1);
          putimage(x,y,buff,XOR_PUT);
```

```
/*check if ball touches horizontal
boundaries*/
        if(x>maxx-50||x<0)
        {
                sound(50);
                delay(10);
                nosound();
                xdirn *=-1;
        }

/*check if ball touches vetical
boundaries*/
        if(y>maxy-50||y<20)
        {
                sound(50);
                delay(10);
                nosound();
                ydirn *=-1;
        }
}
        getch();
        closegraph();
        restorecrtmode();
}
```