THE MUSIC PLAYER

MADE BY:-

AVNEESH SINGHAL 14103188

ADITYA AMBIKESH 14103199

PRAKHAR SRIVASTAVA 14103209

SHIVANSHU GUPTA 14103212

BATCH:-B6

C Project Report

# MUSIC PLAYER

## Overall approach& Outline

* Our C Project is based on Application Music Player.
* Music Player App outline-
* Our project is to make a replica of music players in C. And to provide the users with the basic options used in these players.
* Displaying the songs by the categorizing them on the basis of movies,artist and normal display.
* Enable them to create their own playlist with the options to add a single song or a full movie song or the songs under the searched artists.
* Enable them to modify the details of the songs.
* Enable them to play the songs of their playlist or a particular movie/album and artists.

# DATA TYPES USED

1. Structures
2. Linked List
3. Queue
4. Multi-lists
5. Searching
6. Pointers
7. File Management
8. Functions
9. Graphics
10. Sounds
11. Keyboard Functions

SYNTAXS USED(New)

* Outtextxy-outtextxy function display text or string at a specified point(x,y) on the screen.
* Getpixel-getpixel function returns the color of pixel present at location(x, y).
* Getmaxy() &getmaxx()- returns the maximum Y& X coordinate for current graphics mode and driver.
* khbit-Checks the availability of keystrokes.
* Putpixel() – puts pixel at defined place in defined color.
* **Technical sections(Algorithm)/Top-Down:**
* Declaring various structures and functions and passing of structures to a functions and arrays to a functions
* KEYBOARD CONTROLS

The keyboard controls have been assigned by various methods for codes of different keys needed for controlling the menus for various purposes. The main working codes used are the getch(); which is a console function and outputs the ASCII code of any keyboard controlled command entered into the console. The working of the code is very simple and easy to debug. The various values of the keyboard were tested and picked up accordingly to assign it to perform control functions of the menu. As visible the menu can be controlled by arrow keys,”UP, DOWN, RIGHT, LEFT ”

The main functions assigned are as follows:

* ‘ ’ arrow are used to move up the selection bar of the menu.

The function is to reduce the y co-ordinate by 40.

The function is passed through an ‘if’ condition to limit the movement till 200 pixel.

* ‘ ’ arrow are used to move down the pointer.

The function is to increase the value of y co-ordinate by 40.

The function is passed through an ‘if’ condition to limit the movement till 400 pixel.

* ‘ENTER’ has been used to proceed to next menu and to use its functions.

The other keyboard input function used in the game is ‘kbhit’ which is running a ‘for’ loop, waiting for keyboard to be hit. As soon as the keyboard is hit, the mainscreen(); function jumps to the main functions screen.

The main game runs in infinite loop broken by an ‘ENTER’ or ‘ESC’ keypress.

* File Management for Storing Music

void write()

{

FILE \*fp;

ff f;

fp=fopen("SONGS.bin","w");

for(temp=top->beg;temp!=NULL;temp=temp->left)

for(temp5=temp->start3;temp5!=NULL;temp5=temp5->down)

for(temp7=temp5->start;temp7!=NULL;temp7=temp7->next)

{

strcpy(f.ge,temp->gen);

for(temp3=temp->start2;temp3!=NULL;temp3=temp3->under)

{

for(temp8=temp3->start1;temp8!=NULL;temp8=temp8->next1)

if(strcmpi(temp8->s\_name,temp7->s\_name)==0)

{

strcpy(f.ar,temp3->artist1);

strcpy(f.ar1,temp3->artist2);

}

}

strcpy(f.na,temp5->name);

strcpy(f.sname,temp7->s\_name);

f.len=temp7->length;

f.n=temp7->nop;

fwrite(&f,sizeof(f),1,fp);

}

fclose(fp);

}

Declared pointer that access the filename, through file management and a structure

* Multi-List

Void cre\_gr(char \*a)

{

temp=(gr \*)malloc(sizeof(gr));

strcpy(temp->gen,a);

temp->start2=NULL;

temp->end2=NULL;

temp->left=NULL;

temp->start3=NULL;

temp->start3=NULL;

if(top->beg==NULL)

{

top->beg=temp;

top->last=temp;

}

else

{

top->last->left=temp;

top->last=temp;

}

if(temp!=NULL)

printf("GENRE NODE CREATED\n");

}

voidcre\_art(char \*a,char \*b,char \*c)

{

for(temp4=top->beg->start2;temp4!=NULL;temp4=temp4->under)

if(strcmpi(b,temp4->artist1)==0&&strcmpi(c,temp4->artist2)==0)

goto y;

temp3=(art \*)malloc(sizeof(art));

strcpy(temp3->artist1,b);

strcpy(temp3->artist2,c);

temp3->start1=NULL;

temp3->end1=NULL;

temp3->under=NULL;

for(temp=top->beg;temp!=NULL;temp=temp->left)

{

if(strcmpi(a,temp->gen)==0)

{

if(temp->start2==NULL)

{

temp->start2=temp3;

temp->end2=temp3;

}

else

{

temp->end2->under=temp3;

temp->end2=temp3;

}

printf("CREATED\n");

break;

}

}

y:

}

voidcre\_d(char \*a,char \*e)

{

for(temp6=top->beg->start3;temp6!=NULL;temp6=temp6->down)

if(strcmpi(e,temp6->name)==0)

goto x;

temp5=(d \*)malloc(sizeof(d));

strcpy(temp5->name,e);

temp5->start=NULL;

temp5->end=NULL;

temp5->down=NULL;

for(temp=top->beg;temp!=NULL;temp=temp->left)

{

if(strcmpi(a,temp->gen)==0)

{

if(temp->start3==NULL)

{

temp->start3=temp5;

temp->end3=temp5;

}

else

{

temp->end3->down=temp5;

temp->end3=temp5;

}

printf("CREATED1\n");

break;

}

}

x:

}

* STRUCTURE

Typedef struct song

{

chars\_name[40];

int length,nop;

struct song \*next;

struct song \*next1;

}s;

typedefstruct details

{

char name[10];

s \*start;

s \*end;

struct details \*down;

}d;

Typedef struct playlist

{

char q;

d \*begin;

d \*final;

}pl;

Typedef struct artist

{

char artist1[10];

char artist2[10];

s \*start1;

s \*end1;

struct artist \*under;

}art;

Typedef struct genre

{

char gen[10];

d \*start3;

d \*end3;

art \*start2;

art \*end2;

struct genre \*left;

}gr;

Typedef structmn

{

char c;

gr \*beg;

gr \*last;

}m;

Typedef structforfile

{

Char sname[10];

Char na[10];

Char ar[10];

Char ar1[10];

Char ge[10];

Int len,n;

}ff;

Typedef struct forfilepl

{

Char sname[10];

char na[10];

int len,n;

}fpl;

* GRAPHICS

Declared a Function void mainscreen() through which outlook of frontpage is made .It consists of simply concentric circlesand concentric arcs with changing position.

for(x=1;x!=600&&!kbhit();x++)

{

int n=95,t=10;

if(kbhit!=0)

{

setcolor(10);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+200);

delay(25);

setcolor(0);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,1+x,t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,2\*n+1+x,2\*n+t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,3\*n+1+x,3\*n+t+x,r+200);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+50);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+150);

arc(getmaxx()/2,getmaxy()/2,4\*n+1+x,4\*n+t+x,r+200);

}

}

}

* Declared a Function void cre\_sn(ff \*g,int l) etc , this function defines the layout of the details to be entered in the asked sequence.

Void cre\_sn(ff \*g,int l)

{

int i=0,j=0,k=0;

if(l<0)

return;

for(temp6=top->beg->start3;temp6!=NULL&&k==0;temp6=temp6->down)

if(strcmpi(g->na,temp6->name)==0)

for(temp8=top->beg->start3->start;temp8!=NULL&&k==0;temp8=temp8->next)

if(strcmpi(g->sname,temp8->s\_name)==0)

++k;

for(temp3=top->beg->start2;temp3!=NULL&&j==0;temp3=temp3->under)

if(strcmpi(g->ar,temp4->artist1)==0&&strcmpi(g->ar1,temp4->artist2)==0)

for(temp8=temp3->start1;temp8!=NULL&&j==0;temp8=temp8->next1)

if(strcmpi(g->sname,temp8->s\_name)==0)

++j;

if(k==1&&j==1)

return;

temp7=(s \*)malloc(sizeof(s));

strcpy(temp7->s\_name,g->sname);

temp7->length=g->len;

temp7->nop=g->n;

temp7->next=NULL;

temp7->next1=NULL;

for(temp=top->beg;temp!=NULL&&i==0;temp=temp->left)

{

if(strcmpi(g->ge,temp->gen)==0)

{

for(temp3=temp->start2;temp3!=NULL&&j==0;temp3=temp3->under)

if(strcmpi(g->ar,temp3->artist1)==0&&strcmpi(g->ar1,temp3->artist2)==0)

{

if(temp3->start1==NULL)

{

temp3->start1=temp7;

temp3->end1=temp7;

}

else

{

temp3->end1->next1=temp7;

temp3->end1=temp7;

}

++j;

}

for(temp5=temp->start3;temp5!=NULL&&k==0;temp5=temp5->down)

if(strcmpi(g->na,temp5->name)==0)

{

if(temp5->start==NULL)

{

temp5->start=temp7;

temp5->end=temp7;

}

else

{

temp5->end->next=temp7;

temp5->end=temp7;

}

printf("SONG ADDED\n");

++k;

}

++i;

}

}

if(temp==NULL&&i==0)

{

cre\_gr(g->ge);

cre\_art(g->ge,g->ar,g->ar1);

cre\_d(g->ge,g->na);

cre\_sn(g,--l);

}

else

{

if(temp3==NULL&&j==0)

{

cre\_art(g->ge,g->ar,g->ar1);

}

if(temp5==NULL&&k==0)

{

cre\_d(g->ge,g->na);

}

cre\_sn(g,--l);

}

}

* A variety of display functions have been used, for displaying different data types and structures . Efforts aim towards showing all output data using graphics only. The input ,though remains in dos style for sake of convenience.

Disp\_sn for main song display….

void disp\_sn()

{

int l=1;

cleardevice();

settextstyle(1,0,1);

setcolor(15);

for(temp=top->beg;temp!=NULL;temp=temp->left,l++)

{ char a[5];

outtextxy(20,l\*30,temp->gen);

outtextxy(20,10,"GENRE");

for(temp5=temp->start3;temp5!=NULL;temp5=temp5->down)

{ outtextxy(20+(getmaxx()/4),l\*30,temp5->name);

outtextxy(20+(getmaxx()/4),10,"MOVIE/ALBUM");

outtextxy(20+(2\*getmaxx()/4),10,"SONGNAME\tLENGTH\n");

for(temp7=temp5->start;temp7!=NULL;temp7=temp7->next)

{ outtextxy(20+(2\*getmaxx()/4),l\*30,temp7->s\_name);

sprintf(a,"%d",temp7->length);

outtextxy(20+(3\*getmaxx()/4),l\*30,a);

}

}

}

getch();

}

void disp\_mv()

{

printf("MOVIE WISE LIST\n");

for(temp=top->beg;temp!=NULL;temp=temp->left)

for(temp5=temp->start3;temp5!=NULL;temp5=temp5->down)

printf("%s\n",temp5->name);

}

void disp\_ar()

{ settextstyle(3,0,2);

outtextxy(10,10,"ARTIST'S WISE LIST");

settextstyle(3,0,1);

for(temp=top->beg;temp!=NULL;temp=temp->left,u++)

for(temp3=temp->start2;temp3!=NULL;temp3=temp3->under)

outtextxy(20,30\*u,temp3->artist1);

outtextxy(20+strlen(temp3->artist1),30\*(u+1),temp3->artist2);

}

* Playlist is the main attraction of the program, working on queue system.

It writes files to save playlist.

void write\_pl()

{

FILE \*fp;

fpl f;

fp=fopen("PLAYLIST.bin","w");

for(temp5=top1->begin;temp5!=NULL;temp5=temp5->down)

{

strcpy(f.na,temp5->name);

for(temp7=temp5->start;temp7!=NULL;temp7=temp7->next)

{

strcpy(f.sname,temp7->s\_name);

f.len=temp7->length;

f.n=temp7->nop;

fwrite(&f,sizeof(f),1,fp);

}

}

fclose(fp);

}

* It can be read too

void read\_pl()

{

FILE \*fp;

ff f;

fp=fopen("PLAYLIST.bin","r");

while(!feof(fp))

{

fread(&f,sizeof(f),1,fp);

cre\_pl2(&f,1);

}

fclose(fp);

}

* Deletion functions

The deletion functions traverse and remove the given string using any traversal type in any type of order.

void del\_s()

{

char c[10];

printf("\nENTER SONG NAME TO BE DELETED");

fflush(stdin);

gets(c);

for(temp=top->beg,temp1=top->beg;temp!=NULL;temp=temp->left)

{

for(temp3=temp4=temp->start2;temp3!=NULL;temp3=temp3->under)

{

for(temp7=temp3->start1,temp8=temp3->start1;temp7!=NULL;temp7=temp7->next1)

{

if(strcmpi(c,temp7->s\_name)==0)

{

if(temp7->next1==NULL)

temp3->start1=NULL;

else

temp8->next1=temp7->next1;

}

temp8=temp7;

}

if(temp3->start1==NULL)

{

if(temp3->under==NULL)

temp->start2=NULL;

else

temp4->under=temp3->under;

}

temp4=temp3;

}

for(temp5=temp6=temp->start3;temp5!=NULL;temp5=temp5->down)

{

for(temp7=temp8=temp5->start;temp7!=NULL;temp7=temp7->next)

{

if(strcmpi(c,temp7->s\_name)==0)

{

if(temp7->next==NULL)

temp5->start=NULL;

else

temp8->next=temp7->next;

}

temp8=temp7;

}

if(temp5->start==NULL)

{

if(temp5->down==NULL)

temp->start3=NULL;

else

temp6->down=temp5->down;

}

temp6=temp5;

}

if(temp->start3==NULL&&temp->start2==NULL)

{

if(temp->left==NULL)

top->beg=temp->left;

else

temp1->left=temp->left;

}

temp1=temp;

}

}

void del\_mv()

{

char c[10];

printf("ENTER MOVIE NAME TO BE DELETED");

fflush(stdin);

gets(c);

for(temp=top->beg,temp1=top->beg;temp!=NULL;temp=temp->left)

{

for(temp5=temp6=temp->start3;temp5!=NULL;temp5=temp5->down)

{

if(strcmpi(c,temp5->name)==0)

{

if(temp5->down==NULL)

temp->start3=NULL;

else

temp8->next=temp7->next;

}

temp6=temp5;

}

if(temp->start3==NULL&&temp->start2==NULL)

{

if(temp->left==NULL)

top->beg=temp->left;

else

temp1->left=temp->left;

}

temp1=temp;

}

}

void del\_ar()

{

char c[10],d[10];

printf("ENTER ARTISTS NAME TO BE DELETED");

fflush(stdin);

gets(c);

fflush(stdin);

gets(d);

for(temp=top->beg,temp1=top->beg;temp!=NULL;temp=temp->left)

{

for(temp3=temp4=temp->start2;temp3!=NULL;temp3=temp3->under)

{

if(strcmpi(c,temp3->artist1)==0&&strcmpi(d,temp3->artist2)==0)

{

if(temp3->under==NULL)

temp->start2=NULL;

else

temp4->under=temp3->under;

}

temp4=temp3;

}

if(temp->start3==NULL&&temp->start2==NULL)

{

if(temp->left==NULL)

top->beg=temp->left;

else

temp1->left=temp->left;

}

temp1=temp;

}

}

* **Learning Outcomes:**Undergoing this C project was nice experience. As we tried to developed a music player in C it involve of various concepts of C language and we learned about the applicative part of various syntaxes in C , various concepts such as Graphics, sounds, pointers, linked list,stack,queue,multilist ,structure to a functions was quite helpful and backbone of the music player.