**IMPLEMENTATION OF CODE**

1. In our Virtual Piano project, we’ll be using C++ programming language with the help of Turbo C++ Compiler. To make things easier, we have divided the code into several classes.
2. Each of the classes in our code inherit certain information from each other with the help of inheritance concept in OODP.
3. We also make sure that things go well, by hiding few of the data of a class to another as it may collapse the program working.
4. With the help of Exceptional handling, our code becomes completely user-friendly as it would be able to point out what mistake has the user done while doing console input.
5. Now coming to our code, the function starts with starts by defining required constants and including the required header files. It then defines a class called **piano** which contains some colour codes for the piano keys. The program also defines several functions for drawing the piano keys, handling the mouse, and playing musical notes.

#include<iostream.h>

#include<dos.h>

#include<conio.h>

#include<stdlib.h>

#define SHOW 1

#define HIDE 2

union REGS input,output;

1. In the main function, we check whether the mouse and keyboard are properly connected to the system or not, else the system would raise an error and it will quit. Keys are assigned to keyboard. This function provides the functionality to change the pitch of sound.

|  |
| --- |
|  |
|  | Int main()  { |
|  | int b,x,y,key; |
|  | char ch; |
|  | if(initmouse()==-1) /\* Terminates the program if mouse not connected \*/ |
|  | { |
|  | clrscr(); |
|  | cout<<"\n\nNO MOUSE !\n\n"; |
|  | exit(1); |
|  | } |
|  | pointer(SHOW); |
|  | setupscreen(); |
|  | exitcode=1; |
|  | while(exitcode) |
|  | { |
|  | if(kbhit()) |
|  | { |
|  | ch=getch(); |
|  | if(ch==27) break; |
|  | if(ch==75) a--; |
|  | if(ch==77) a++; |
|  | check\_keys(x,y); |
|  | switch(ch) |
|  | { |
|  | case 'a':case'A':key=0;break; |
|  | case 's':case'S':key=1;break; |
|  | case 'd':case'D':key=2;break; |
|  | case 'f':case'F':key=3;break; |
|  | case 'j':case'J':key=4;break; |
|  | case 'k':case'K':key=5;break; |
|  | case 'l':case'L':key=6;break; |
|  | } |
|  | sound(freq[key]\*a); |
|  | delay(80); |
|  | nosound(); |
|  | } |

1. After the requirement process, the call statement for GUI (setupscreen() is provided), for the piano and it gets the user input from the user.

|  |
| --- |
|  |
|  |
|  |
|  | getmouse(&b,&x,&y); |
|  |  |
|  | while(b==1) |
|  | { |
|  | getmouse(&b,&x,&y); |
|  | key=check\_xy(x,y); |
|  | if(key!=-1 && key<7) |
|  | { |
|  | sound(freq[key]\*a); |
|  | } |
|  | else if(key>6) |
|  | { |
|  | sound(freq[12-key]\*(a/2)); |
|  | } |
|  | } |
|  | nosound(); |
|  | check\_keys(x,y); |
|  | } |
|  | } |
|  | textbackground(0); |
|  | clrscr(); |
|  | \_setcursortype(\_NORMALCURSOR); |
|  | return 0; |
|  | } |
|  |  |

1. The function setupscreen() makes for the GUI for the virtual piano and also they function pianokey() sets an actual piano on the

|  |
| --- |
|  |
|  |
|  |

void setupscreen() /\* Display screen settings \*/

{

\_setcursortype(\_NOCURSOR);

textbackground(backcolor);

clrscr();

drawpiano(9,17);

BOX(4,5,8,5,0);

BOX(5,5,7,5,1);

textcolor(15);

gotoxy(70,5);

cprintf(">Quit<");

gotoxy(4,4);

cprintf("Range");

textbackground(0);

gotoxy(8,5);

cprintf("");

gotoxy(4,5);

cprintf("");

gotoxy(6,5);

cprintf("%d",a);

textcolor(14);

textbackground(backcolor+7);

gotoxy(30,4);

cprintf(" VIRTUAL PIANO ");

}

void drawpiano(int x,int y) /\* Drawing of piano \*/

{

int t=9;

BOX(x-5,y-3,75,y+8,color.border); /\*invoking function box \*/

BOX(x-4,y-2,74,y+7,color.back);

pianokey(x,y);

pianokey(x+t,y);

pianokey(x+t\*2,y);

pianokey(x+t\*3,y);

pianokey(x+t\*4,y);

pianokey(x+t\*5,y);

pianokey(x+t\*6,y);

}

|  |
| --- |
|  |
|  |
|  |

void drawpiano(int x,int y) /\* Drawing of piano \*/

{

int t=9;

BOX(x-5,y-3,75,y+8,color.border); /\*invoking function box \*/

BOX(x-4,y-2,74,y+7,color.back);

pianokey(x,y);

pianokey(x+t,y);

pianokey(x+t\*2,y);

pianokey(x+t\*3,y);

pianokey(x+t\*4,y);

pianokey(x+t\*5,y);

pianokey(x+t\*6,y);

}

1. The code then initiates the mouse and the pointer/cursor gets the input from the user and checks the input and then gives the appropriate sounds.

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
| Int initmouse() |
|  | { |
|  | input.x.ax=0; |
|  | int86(0x33,&input,&output); |
|  | return(output.x.ax==0 ? -1 : 0); |
|  | } |
|  | void pointer(int on) |
|  | { |
|  | input.x.ax=on; |
|  | int86(0x33,&input,&output); |
|  | } |
|  | void restrictmouse(int x1,int y1,int x2,int y2) |
|  | { |
|  | input.x.ax=7; |
|  | input.x.cx=x1/8; |
|  | input.x.dx=x2/8; |
|  | int86(0x33,&input,&output); |
|  | input.x.ax=8; |
|  | input.x.cx=y1/8; |
|  | input.x.dx=y2/8; |
|  | int86(0x33,&input,&output); |
|  | } |
|  | void getmouse(int \*button,int \*x,int \*y) |
|  | { |
|  | input.x.ax=3; |
|  | int86(0x33,&input,&output); |
|  | \*button=output.x.bx; |
|  | \*x=output.x.cx/8; |
|  | \*y=output.x.dx/8; |
|  | } |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
|  | |
|  | |
|  | |
| int check\_xy(intx,inty) |
|  | { | |
|  | /\* Mid keys \*/ | |
|  | if(x>=11 && y>=16 && x<=12 && y<=18) | |
|  | return 7; | |
|  | if(x>=20 && y>=16 && x<=21 && y<=18) | |
|  | return 8; | |
|  | if(x>=29 && y>=16 && x<=30 && y<=18) | |
|  | return 9; | |
|  | if(x>=38 && y>=16 && x<=39 && y<=18) | |
|  | return 10; | |
|  | if(x>=47 && y>=16 && x<=48 && y<=18) | |
|  | return 11; | |
|  | if(x>=56 && y>=16 && x<=57 && y<=18) | |
|  | return 0; | |
|  | if(x>=65 && y>=16 && x<=66 && y<=18) | |
|  | return 12; | |
|  |  | |
|  | /\* Big keys \*/ | |
|  | if(x>=8 && y>=16 && x<=15 && y<=22) | |
|  | return 0; | |
|  | if(x>=17 && y>=16 && x<=24 && y<=22) | |
|  | return 1; | |
|  | if(x>=26 && y>=16 && x<=33 && y<=22) | |
|  | return 2; | |
|  | if(x>=35 && y>=16 && x<=42 && y<=22) | |
|  | return 3; | |
|  | if(x>=44 && y>=16 && x<=51 && y<=22) | |
|  | return 4; | |
|  | if(x>=53 && y>=16 && x<=60 && y<=22) | |
|  | return 5; | |
|  | if(x>=62 && y>=16 && x<=69 && y<=22) | |
|  | return 6; | |
|  | return (-1); /\*no key pressed \*/ | |
|  | } | |
|  | void check\_keys(int x,int y) | |
|  | { | |
|  | if(x==7 && y==4) a++; | |
|  | if(x==3 && y==4) a--; | |
|  | if(a<1) a=1;else if(a>30) a=30; | |
|  | textcolor(15);textbackground(1); | |
|  | gotoxy(5,5);cprintf(" "); | |
|  | if(a<10) gotoxy(6,5); | |
|  | else gotoxy(5,5); | |
|  | cprintf("%d",a); | |
|  | if(x>=69 && y>=4 && x<=74 && y<=4) exitcode=0; | |
|  | } | |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |