**/\***

**When I wrote the code, only God and i understood, Now Only God understands the code....**

**\*/**

**#include<graphics.h>**

**#include<windows.h>**

**#include<dos.h>**

**#include<conio.h>**

**#include<process.h>**

**#include<iostream>**

**#include<math.h>**

**//line hit box info's**

**int linex1[3] = {250,750,1250};**

**int liney1[3] = {200,200,200};**

**int linex2[3] = {250,750,1250};**

**int liney2[3] = {700,700,700};**

**//-----------------------------------**

**//board part - inner workings**

**//things to change while working**

**//others are just the basic**

**int lTop = -1;**

**int mTop = -1;**

**int rTop = -1;**

**int leftStack[6];**

**int middleStack[6];**

**int rightStack[6];**

**//game count**

**int TotalBoxes;**

**//-------------------------------------**

**//temporary store for the values of the hit box**

**int tempTop;**

**int tempTopValue;**

**int tempLine;**

**//move count**

**int moveCount=0;**

**//graphics coordinates..**

**//left line**

**int leftLeft[5] = {150,150,150,150,150};**

**int topLeft[5] = {620,540,460,380,300};**

**int rightLeft[5] = {350,350,350,350,350};**

**int bottomLeft[5]= {700,620,540,460,380};**

**//middle line**

**int leftMiddle[5] = {650,650,650,650,650};**

**int topMiddle[5] = {620,540,460,380,300};**

**int rightMiddle[5] = {850,850,850,850,850};**

**int bottomMiddle[5] = {700,620,540,460,380};**

**//right line**

**int leftRight[5] = {1150,1150,1150,1150,1150};**

**int topRight[5] = {620,540,460,380,300};**

**int rightRight[5] = {1350,1350,1350,1350,1350};**

**int bottomRight[5] = {700,620,540,460,380};**

**//Boxes class**

**class Box**

**{**

**public:**

**void start();**

**};**

**//tower of hanoi class**

**class towerofhanoi**

**{**

**public:**

**DWORD width,height;**

**//initiate the window**

**towerofhanoi()**

**{**

**width=GetSystemMetrics(SM\_CXSCREEN);**

**height=GetSystemMetrics(SM\_CYSCREEN);**

**initwindow(width,height,"TOWER OF HANOI");**

**}**

**void intro(); //intro of the game**

**void draw(); //main draw logic**

**void drawBoard(); //draw the lines**

**void drawBox(); //draw the boxes**

**void tempStore(int); //store the dragged box temporary**

**void dropOutside(); //restore the dragged box to its previous position**

**int dropInside(int); //put the box inside the line**

**int checkWin(); //win logic**

**void end(); //end - game over**

**void resetGame(int); //setting up the game**

**void putLeftStack(); //setting up the left line with user specified range of boxes**

**//code for collision detections**

**//rectange - rectangle collision**

**int lineHitBox(int x1,int y1,int xw,int yw,int rx,int ry,int rw,int rh,int value) {**

**if (x1 < rx + rw &&**

**x1 + xw > rx &&**

**y1 < ry + ry &&**

**y1 + yw > ry) {**

**// collision detected!**

**return value;**

**}**

**return -1;**

**}**

**//checking for the mouse to hit the boxes**

**int mouseHitBox(int px,int py,int rx,int ry,int rw,int rh,int lineNumber) {**

**if(px>=rx&&px<=rx+rw&&py>=ry&&py<=ry+rh&&lineNumber==0)**

**return 0;**

**if(px>=rx&&px<=rx+rw&&py>=ry&&py<=ry+rh&&lineNumber==1)**

**return 1;**

**if(px>=rx&&px<=rx+rw&&py>=ry&&py<=ry+rh&&lineNumber==2)**

**return 2;**

**return -1;**

**}**

**};**

**//intro of the game after the user has input its box range**

**void towerofhanoi::intro()**

**{**

**int i,j;**

**setcolor(WHITE);**

**for(i=0;i<(width/2)-270;i++)**

**{**

**cleardevice();**

**setcolor(i);**

**rectangle(0,0,639,479);**

**setcolor(WHITE);**

**settextstyle(SANS\_SERIF\_FONT,HORIZ\_DIR,8);**

**outtextxy(i,(height/2)-140,"TOWER OF HANOI");**

**}**

**setcolor(RED);**

**settextstyle(SANS\_SERIF\_FONT,HORIZ\_DIR,3);**

**outtextxy((width/2)-200,height/2,"USE THE LEFT MOUSE TO CLICK"); delay(2000);**

**outtextxy((width/2)-350,(height/2)+40,"ClICK ON THE WINDOW AND PRESS ENTER KEY TO START");**

**getch();**

**}**

**void towerofhanoi::resetGame(int value)**

**{**

**mTop = -1;**

**rTop = -1;**

**TotalBoxes = value;**

**switch(value)**

**{**

**case 1:**

**lTop = 0;**

**leftStack[lTop] = 1;**

**break;**

**case 2:**

**lTop = 1;**

**putLeftStack();**

**break;**

**case 3:**

**lTop = 2;**

**putLeftStack();**

**break;**

**case 4:**

**lTop = 3;**

**putLeftStack();**

**break;**

**case 5:**

**lTop = 4;**

**putLeftStack();**

**break;**

**}**

**draw();**

**}**

**//setting up the left line with boxes**

**void towerofhanoi::putLeftStack()**

**{**

**for(int i=0,j=lTop+1;i<=lTop;i++,j--)**

**{**

**leftStack[i] = j;**

**}**

**}**

**//game over - clear everything**

**void towerofhanoi::end()**

**{**

**int play;**

**cleardevice();**

**outtextxy(getmaxx()/2-100,getmaxy()/2-40,"YOU WIN"); getch();**

**cleardevice();**

**closegraph();**

**exit(0);**

**}**

**//checking the winning logic**

**int towerofhanoi::checkWin()**

**{**

**for(int i=TotalBoxes-1,j=1;i>=0;i--,j++)**

**{**

**if(rightStack[i]!=j)**

**return -1;**

**}**

**return 1;**

**}**

**//if the user drops into any of the line but itself**

**//put the box to that collided line**

**int towerofhanoi::dropInside(int lineNumber)**

**{**

**switch(lineNumber)**

**{**

**case 0:**

**if(leftStack[lTop]<tempTopValue&&lTop!=-1)**

**return -1;**

**lTop++;**

**leftStack[lTop] = tempTopValue;**

**break;**

**case 1:**

**if(middleStack[mTop]<tempTopValue&&middleStack[mTop]!=0) return -1;**

**mTop++;**

**middleStack[mTop] = tempTopValue;**

**break;**

**case 2:**

**if(rightStack[rTop]<tempTopValue&&rightStack[rTop]!=0)**

**return -1;**

**rTop++;**

**rightStack[rTop] = tempTopValue;**

**break;**

**}**

**return 1;**

**}**

**//if the user drops the box outside the line or itself //reset the box to its previous position**

**void towerofhanoi::dropOutside()**

**{**

**switch(tempLine)**

**{**

**case 0:**

**lTop++;**

**break;**

**case 1:**

**mTop++;**

**break;**

**case 2:**

**rTop++;**

**break;**

**}**

**}**

**//temporarily store the drag box**

**//remove it from the line where it was held before**

**void towerofhanoi::tempStore(int lineHitPosition) {**

**char t[5];**

**switch(lineHitPosition)**

**{**

**case 0:**

**tempTop = lTop;**

**tempTopValue = leftStack[lTop];**

**tempLine = lineHitPosition;**

**if(lTop!=-1)**

**lTop--;**

**break;**

**case 1:**

**tempTop = mTop;**

**tempTopValue = middleStack[mTop];**

**tempLine = lineHitPosition;**

**if(mTop!=-1)**

**mTop--;**

**break;**

**case 2:**

**tempTop = rTop;**

**tempTopValue = rightStack[rTop];**

**tempLine = lineHitPosition;**

**if(rTop!=-1)**

**rTop--;**

**break;**

**}**

**}**

**//drawing part**

**//----------------------------------------------------------------------------------------------- //----------------------------------------------------------------------------------------------- //draw the lines**

**void towerofhanoi::drawBoard()**

**{**

**rectangle(250,200,250,700);**

**rectangle(750,200,750,700);**

**rectangle(1250,200,1250,700);**

**}**

**//draw Boxes**

**void towerofhanoi::drawBox()**

**{**

**char t[20];**

**//for left line**

**if(lTop>-1)**

**{**

**for(int i=0;i<=lTop;i++)**

**{**

**setcolor(i+1);**

**setfillstyle(SOLID\_FILL,i+1);**

**sprintf(t,"%d",leftStack[i]);**

**rectangle(leftLeft[i],topLeft[i],rightLeft[i],bottomLeft[i]);**

**floodfill(leftLeft[i]+1,topLeft[i]+1,i+1);**

**setcolor(WHITE);**

**settextstyle(DEFAULT\_FONT,HORIZ\_DIR,3);**

**outtextxy(240,topLeft[i]+35,t);**

**}**

**}**

**//for middle line**

**if(mTop>-1)**

**{**

**for(int i=0;i<=mTop;i++)**

**{**

**setcolor(i+1);**

**setfillstyle(SOLID\_FILL,i+1);**

**sprintf(t,"%d",middleStack[i]);**

**rectangle(leftMiddle[i],topMiddle[i],rightMiddle[i],bottomMiddle[i]); floodfill(leftMiddle[i]+1,topMiddle[i]+1,i+1);**

**setcolor(WHITE);**

**settextstyle(DEFAULT\_FONT,HORIZ\_DIR,3);**

**outtextxy(740,topMiddle[i]+35,t);**

**}**

**}**

**//for right line**

**if(rTop>-1)**

**{**

**for(int i=0;i<=rTop;i++)**

**{**

**setcolor(i+1);**

**setfillstyle(SOLID\_FILL,i+1);**

**sprintf(t,"%d",rightStack[i]);**

**rectangle(leftRight[i],topRight[i],rightRight[i],bottomRight[i]); floodfill(leftRight[i]+1,topRight[i]+1,i+1);**

**setcolor(WHITE);**

**settextstyle(DEFAULT\_FONT,HORIZ\_DIR,3);**

**outtextxy(1240,topRight[i]+35,t);**

**}**

**}**

**}**

**//----------------------------------------------------------------------------------------------- //----------------------------------------------------------------------------------------------**

**//----------------------------------------------------------------------------------------------- //----------------------------------------------------------------------------------------------- //beginning of the draw**

**//every loop starts from here**

**void towerofhanoi::draw()**

**{**

**int x,y;**

**int hit,i;**

**char t[4];**

**cleardevice();**

**drawBoard();**

**drawBox();**

**int minMoves = pow(2,TotalBoxes)-1;**

**//run till a user enters an input**

**while(!kbhit())**

**{**

**//counting the minimum moves possible to solve tower's of hanoi and showing the output**

**sprintf(t,"%d",minMoves);**

**outtextxy(100,30,"Minimum Moves - ");**

**outtextxy(470,30,t);**

**//check for the first hit box**

**if(GetAsyncKeyState(VK\_LBUTTON))**

**{**

**getmouseclick(WM\_LBUTTONDOWN,x,y);**

**for(int i=0;i<3;i++)**

**{**

**hit=-1;**

**//checking for three lines if there is a hit**

**//enable the drag function**

**switch(i)**

**{**

**case 0:**

**if(lTop>-1){**

**hit =**

**mouseHitBox(mousex(),mousey(),leftLeft[lTop],topLeft[lTop],200,80,0); sprintf(t,"%d",leftStack[lTop]);**

**}**

**break;**

**case 1:**

**if(mTop>-1){**

**hit =**

**mouseHitBox(mousex(),mousey(),leftMiddle[mTop],topMiddle[mTop],200,80,1); sprintf(t,"%d",middleStack[mTop]);**

**}**

**break;**

**case 2:**

**if(rTop>-1){**

**hit =**

**mouseHitBox(mousex(),mousey(),leftRight[rTop],topRight[rTop],200,80,2); sprintf(t,"%d",rightStack[rTop]);**

**}**

**break;**

**}**

**if(hit!=-1)**

**{**

**int hitLine=-1;**

**//to store the drag values in tempStore function**

**//to let the box go off from its previous position**

**int count=0;**

**//enable drag option of the box**

**while(!kbhit())**

**{**

**//for dragging of the box**

**if(GetAsyncKeyState(VK\_LBUTTON))**

**{**

**cleardevice();**

**drawBoard();**

**if(count==0)**

**tempStore(hit);**

**drawBox();**

**setcolor(WHITE);**

**outtextxy(mousex()+92,mousey()+38,t);**

**rectangle(mousex(),mousey(),mousex()+200,mousey()+80);**

**count++;**

**}**

**//for droping of the box**

**if(!GetAsyncKeyState(VK\_LBUTTON))**

**{**

**int p;**

**for(int i=0;i<3;i++)**

**{**

**//continue if the line is the same line as the drag one**

**if(hit==i)**

**continue;**

**hitLine =**

**lineHitBox(linex1[i],liney1[i],1,500,mousex(),mousey(),200,80,i);**

**//if there is a hit**

**if(hitLine!=-1)**

**{**

**cleardevice();**

**p = dropInside(i);**

**//if there's a rule break**

**//the higher box cannot be placed above lower box**

**if(p==-1){**

**hitLine=-1;**

**break;**

**}**

**int win = checkWin();**

**//check if there is a win in checkWin() function**

**if(win!=-1)**

**end();**

**//draw the things again**

**moveCount++;**

**drawBoard();**

**drawBox();**

**outtextxy(100,110,"Box Drop Inside");**

**outtextxy(100,70,"Total Move - ");**

**sprintf(t,"%d",moveCount);**

**outtextxy(400,70,t);**

**break;**

**}**

**}**

**//if the box is not dropped in any of the line**

**//put back the box to its original position with dropOutside() function**

**if(hitLine==-1)**

**{**

**cleardevice();**

**dropOutside();**

**drawBoard();**

**drawBox();**

**//if there is a rule break**

**if(p==-1)**

**outtextxy(700,110,"RULE BREAK - LOWER BOX BELOW");**

**outtextxy(100,110,"Box Drop Outside");**

**}**

**break;**

**}**

**delay(40);**

**}//inner while loop**

**}//the drag ending function**

**}//end of for loop**

**}//if there is a hit**

**delay(40);**

**}//Outer while loop**

**}**

**void Box::start()**

**{**

**int tBox;**

**//taking input for the number of boxes**

**std::cout<<"Enter the number of boxes you want to play range from 1 - 5\n"; std::cin>>tBox;**

**//tower of hanoi class**

**towerofhanoi t;**

**//intro of the game**

**t.intro();**

**//setting up the game values**

**t.resetGame(tBox);**

**}**

**int main()**

**{**

**Box b;**

**b.start();**

**getch();**

**return 1;**

**}**