|  |  |
| --- | --- |
| **Roll No.:** | 20102A0004, 20102A0012,20102A0003,20102A0037 |
| **Name:** | Trisha Shah, Shipra Dagli, Vinay Bhojwani, Snehal Lohar |
| **Subject:** | **Computer Graphics (CG)** |
| **Subject In-charge:** | **Sachin Deshpande** |

**Mini Project**

**Title:** Analog Clock

**Estimated time to complete this experiment:** 4 to 6 hours

**Objective:** To apply various concepts of course computer graphics.

**Possible list of Mini projects:** You can choose amongst Raster algorithms, 2D transformation or 2D Viewing transformation and clipping and create one of the following (or similar mini project):

1. Construct a line graph using line drawing algorithm that show stock market variations.
2. Construct a bar graph using line drawing algorithm that show monthly sales.
3. Construct a pie-chart using line and circle generation algorithms that show distribution of total credits of your program into different categories such as electives, core, basic science, etc..
4. Animate a table fan with blades rotating in clockwise manner and also the fan moving from left to right and vice-versa using 2D transformations.
5. Animate analogue clock using 2D transformations.
6. Animate a freely moving 2D or 3D character/ string (like seen in some screen-savers) using 2D or 3D transformation.
7. You are allowed to use your creativity for performing the good animation apart from the tasks mentioned above.

**Note:**

* you can do it in group of max 4 students
* Do mention your references in the next section.

**Books/ Journals/ Websites referred:** https://www.thecrazyprogrammer.com/2014/10/make-analog-clock-in-c-using-graphics.html

**New Concepts to be learned:** Combining multiple concepts of computer graphics and apply them.

**Theory:**

The Analog Clock project has been designed and developed in the C++ programming language with the help of graphics.h and ctime library.

This program implements an Analog Clock designed with the help of a graphics library in C++.

Following functions have been used to design the layout of the clock:

1) circle(int x, int y , int radius)

2) line(int x1, int y1 ,int x2, inty2 )

3) setlinestyle(int linestyle, unsigned upattern,int thickness);

4)outtextxy(int x ,int y ,"text")

5)setcolor(int color)

The time has been accessed with the help of #include library in the program.

**Program:**

#include<graphics.h>

#include<cmath>

#include <ctime>

#include<iostream>

using namespace std;

#define PI 3.142857

main()

{

initwindow(800,800,"ANALOG CLOCK");

int page=0;

while(1)

{

setactivepage(page);

setvisualpage(1-page);

cleardevice();

setcolor(8);

circle(400,400,209);

circle(400,400,210);

circle(400,400,211);

circle(400,400,212);

circle(400,400,213);

circle(400,400,214);

setcolor(6);

circle(400,400,1);

circle(400,400,2);

circle(400,400,3);

circle(400,400,4);

circle(400,400,5);

setcolor(5);

settextstyle(10, 0, 2);

outtextxy(400+185\*sin(PI/6)-5, 400-185\*cos(PI/6),"1");

outtextxy(400+185\*sin(2\*PI/6)-5, 400-185\*cos(2\*PI/6),"2");

outtextxy(400+185\*sin(3\*PI/6)-5, 400-185\*cos(3\*PI/6),"3");

outtextxy(400+185\*sin(4\*PI/6)-5, 400-185\*cos(4\*PI/6),"4");

outtextxy(400+185\*sin(5\*PI/6)-5, 400-185\*cos(5\*PI/6),"5");

outtextxy(400+185\*sin(6\*PI/6)-5, 400-185\*cos(6\*PI/6),"6");

outtextxy(400+185\*sin(7\*PI/6)-5, 400-185\*cos(7\*PI/6),"7");

outtextxy(400+185\*sin(8\*PI/6)-5, 400-185\*cos(8\*PI/6),"8");

outtextxy(400+185\*sin(9\*PI/6)-10, 400-185\*cos(9\*PI/6),"9");

outtextxy(400+185\*sin(10\*PI/6)-10, 400-185\*cos(10\*PI/6),"10");

outtextxy(400+185\*sin(11\*PI/6)-10, 400-185\*cos(11\*PI/6),"11");

outtextxy(400+185\*sin(12\*PI/6)-5, 400-180\*cos(12\*PI/6),"12");

settextstyle(8,0,3);

setcolor(1);

time\_t now = time(0);

tm \*ltm =localtime(&now);

system("cls");

std::cout<<ltm->tm\_hour <<": "<< ltm->tm\_min;

settextstyle(10,0,7);

setcolor(5);

outtextxy(170,100,"ANALOG CLOCK");

setcolor(CYAN);

setlinestyle(0,0,4);

line(400,400,400+150\*sin(ltm->tm\_hour \*PI/6),400-150\*cos(ltm->tm\_hour\*PI/6));

setcolor(1);

setlinestyle(0,0,3);

line(400,400,400+180\*sin(ltm->tm\_min \*PI/30), 400-180\*cos(ltm->tm\_min \*PI/30));

setcolor(MAGENTA);

setlinestyle(0,0,1);

line(400,400,400+180\*sin(ltm->tm\_sec\*PI/30),400-180\*cos(ltm->tm\_sec\*PI/30));

if(GetAsyncKeyState(VK\_RETURN))

break;

delay(10);

page = 1-page;

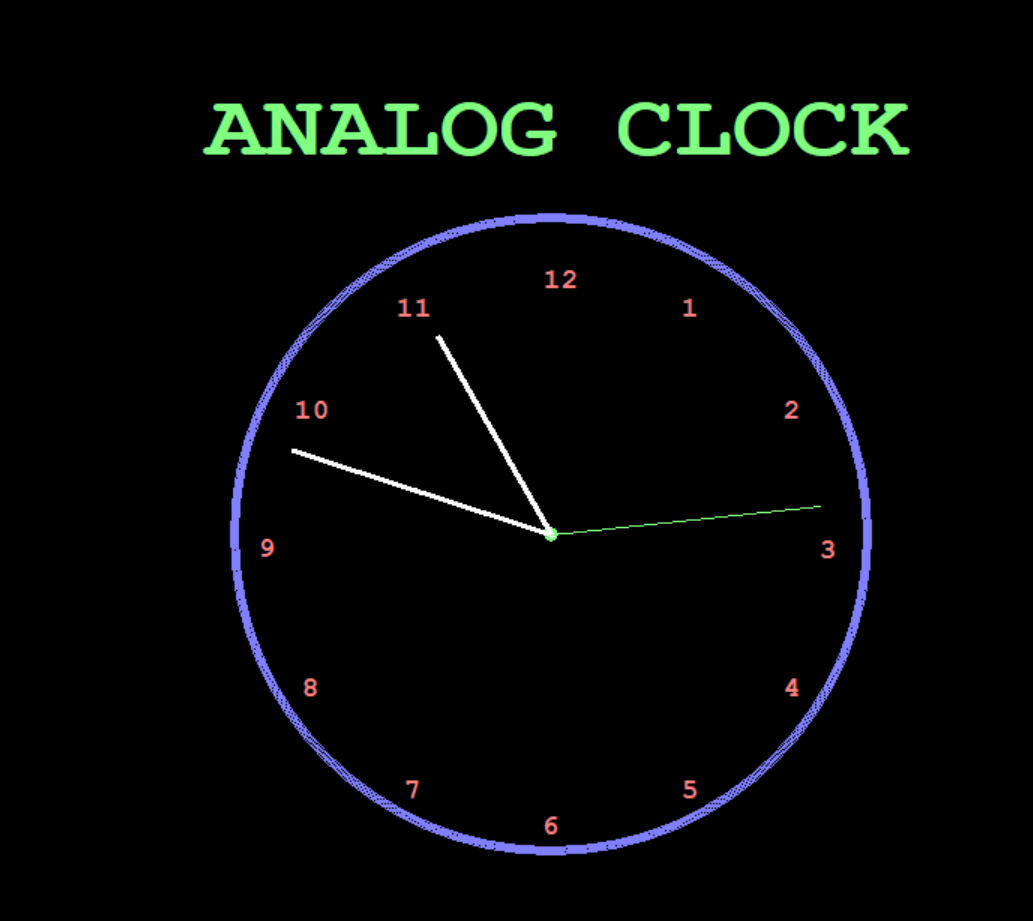
}

closegraph();

return 0;

}

**Output:**

****

**Conclusion:**

The analog clock animation successfully works and shows real time.