Department of Computer Applications

INVERTIS
UNIVERSITY BAREILLY

A Mini Project Report
On
"CLOCK"

Submitted in Fulfillment of the Requirement for the Degree
of
Bachelor of Computer Applications
Invertis University, Bareilly

Mini Project Coordinator
Faculty Coordinator: Miss Pratha Saxena
Designation
Department of Computer Applications

Submitted By
Student Name: Amit Kumar Verma
Roll No: 2110201025

ACKNOWLEGDEMENT

I extend my thanks to the Faculty **of Computer Applications, Invertis University, Bareilly** who has given me the opportunity to work on this Project.

Today I am feeling a great sense of excitement on my way to successfully complete my mini project on **“Clock”** under the guidance of Dr./Mr./Ms. **“Pratha Saxena”**.

I sincerely thank him/her for responding with great confidence and faith in my work and being with me to encourage and guide me to successful project completion.

I would also like to thank **Dr. Archana Saxena, HOD, Faculty of Computer Applications, Invertis University** for their support and all our friends and colleagues who have created an atmosphere to encourage me from time to time making our work easy.

I would also like to thank **Dr. Manish Gupta, Dean Academics** for his valuable suggestions.

Thank You

Signature of the Candidate

(Amit Kumar Verma)

INDEX

INTRODUCTION

Introduction :

Clock is an essential aspect of many graphical user interface. Building clock by programmatically is very interesting topic. Using C graphics program, creating an analog and digital clock is very fun and challenging project for beginners who are interested in learning graphics programming. Creating a clock may seem like a difficult task but with C or C++ graphics programming, it can be done easily. In this post, I shall show you how to create analog and digital clock which updating its time by C or C++ graphics programming language.

What is clock :

Clocks are an essential part of our modern lives. In our daily life, we can not survive without knowledge of time. We get the exact time by using of clock. A clock is a device used to measure and display the exact time. Clock has been used for thousands of years to help people keep track of time and organize their daily lives.

Types of clock :

There are several types of clocks we see in our daily life. They are Analog Clock, Digital Clock, Mechanical Clock, Quartz Clock, Atomic Clock, Solar Clock, Radio Clock, Smart Clock etc. In this post, I discuss about Analog Clock and Digital Clock.

Analog clock is traditional clocks with a circular dial or face with hour, minute and second hands that point to the numbers or symbols on the dial. Analog Clock use a mechanical mechanism that moves the hands at a constant rate to keep the current time. There are some example of analog clock are wrist watch, wall clock and tower clock etc.

Digital clock is display the time using digits or numbers, typically on an LED or LCD display. They are used in electronic devices like computer, smartphone and alarm clock.

SYSTEM ANALYSIS & FEASIBILITY STUDY

SYSTEM SPECIFICATION

➢ HARDWARE REQUIREMENTS:

- Dual Core Processor
- 2GB RAM
- 40GB Hard disk
- DVD drive
- Mouse and other pointing devices
- Keyboard

➢ SOFTWARE REQUIREMENTS:

- Programming language – C/C++ using OpenGL
- Operating system – Linux operating system
- Compiler – C Compiler
- Graphics library – GL/glut.h OpenGL 2.0

System Architecture:

The clock application will be developed using a client-server architecture. The client-side will be a desktop or mobile application, and the server-side will be a web server running a database management system. The application will communicate with the server using HTTP requests and responses. The server will store user data, such as time zone preferences, alarm settings, and clock customizations.

User Interface Design:

The user interface will be designed using modern design principles, such as minimalism and simplicity. The interface will have the following components:

- Time and date display: This component will show the current time and date in the selected time zone and format.
- Alarm settings: This component will allow users to set, edit, and delete alarms.
- Clock customizations: This component will allow users to customize the clock's appearance, such as color scheme and font style.
- Brightness settings: This component will allow users to adjust the brightness of the clock.
- Pause/stop settings: This component will allow users to pause or stop the clock temporarily.

Testing:

The application will be tested using a variety of techniques, such as unit testing, integration testing, system testing, and acceptance testing. The testing will cover functional and non-functional requirements, such as accuracy, reliability, performance, and usability.

CODE

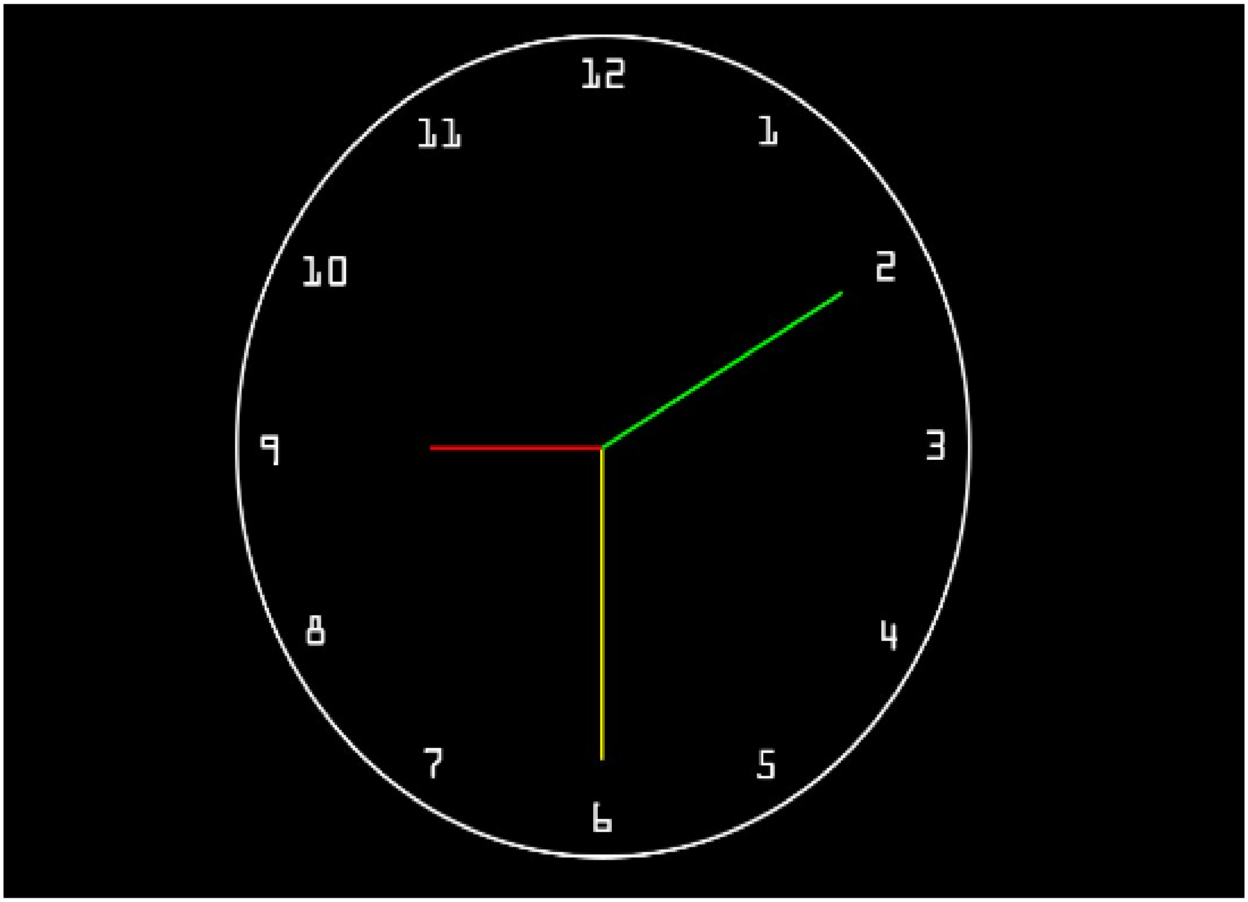
```
/*Developed by Amit Kumar Verma*/
#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <dos.h>
int main(void)
{
    int xmax, ymax, x, y, i = 0, j = 0, k = 0, m = 0, s = 0, h = 0, m1 = 0;
    float l = 0.0;
    int graphic_driver = DETECT, graphic_mode;
    initgraph(&graphic_driver, &graphic_mode, "//turboc3/bgi");
    xmax = getmaxx();
    ymax = getmaxy();
    x = xmax / 2;
    y = ymax / 2;
    setcolor(15);
    setbkcolor(0);
    rectangle(x + 150, y - 200, x + 300, y - 170);
    line(x + 200, y - 200, x + 200, y - 170);
    line(x + 250, y - 200, x + 250, y - 170);
    circle(x, y, 179);
    circle(x, y, 180);
    setfillstyle(0, 0);
    floodfill(x, y, 0);
```

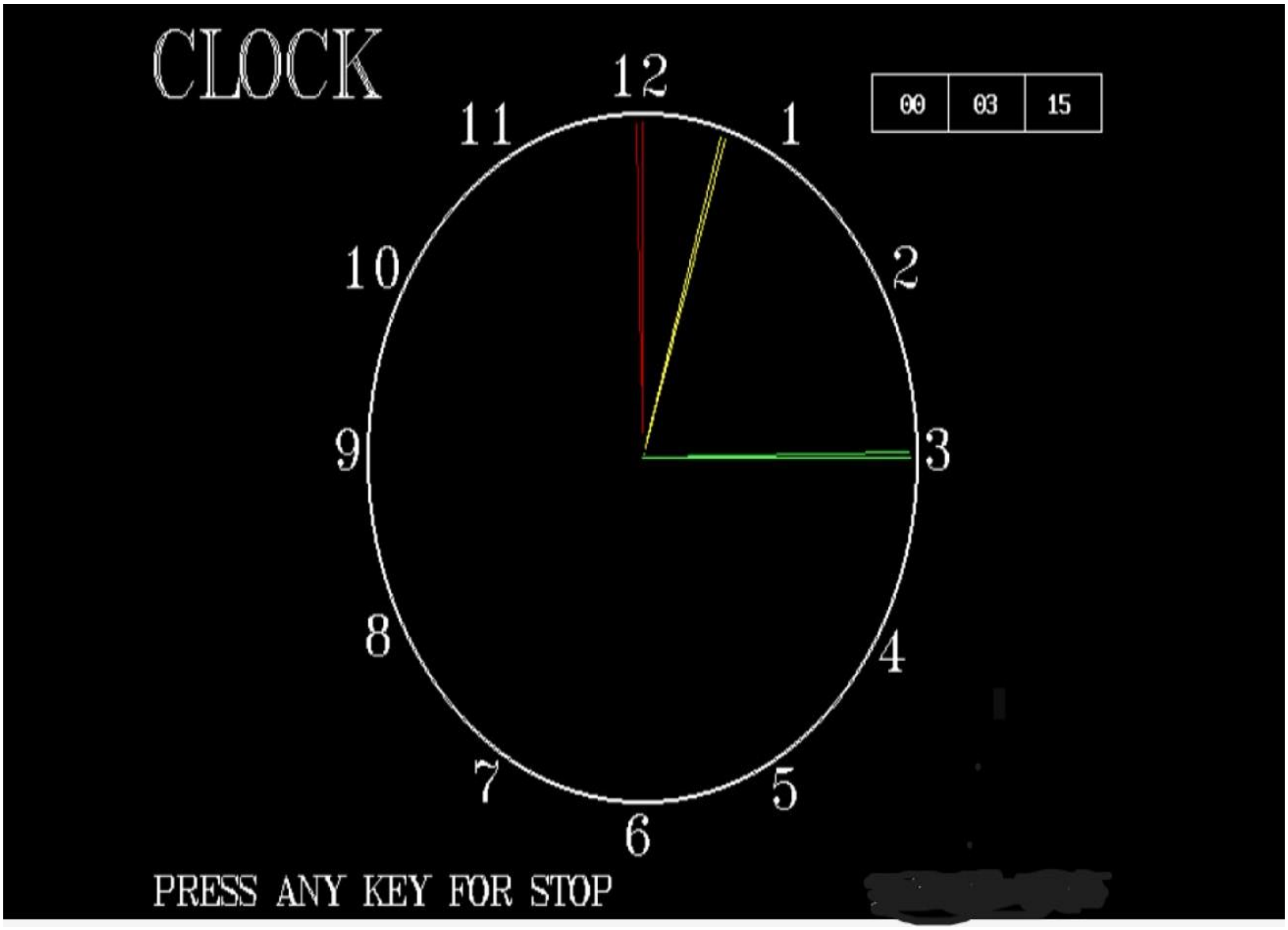
```
settextstyle(1, 0, 4);
outtextxy(x - 200, y - 25, "9");
outtextxy(x + 186, y - 25, "3");
outtextxy(x - 20, y - 220, "12");
outtextxy(x - 10, y + 176, "6");
outtextxy(x + 90, y - 195, "1");
outtextxy(x + 165, y - 120, "2");
outtextxy(x + 155, y + 80, "4");
outtextxy(x + 86, y + 152, "5");
outtextxy(x - 109, y + 148, "7");
outtextxy(x - 180, y + 72, "8");
outtextxy(x - 195, y - 120, "10");
outtextxy(x - 120, y - 195, "11");
settextstyle(1, 0, 3);
outtextxy(0, ymax - 30, "PRESS ANY KEY FOR STOP");
outtextxy(x + 150, y + 210, "AMIT KUMAR VERMA");
settextstyle(1, 0, 6);
outtextxy(0, 0, "CLOCK");
while (!kbhit())
{
    setcolor(14);
    if (m == 60)
    {
        if (k <= 90)
        {
            setcolor(14);
            pieslice(x, y, 91 - k, 90 - k, 175);
            sound(440);
            delay(10);
            nosound();
            setcolor(0);
            pieslice(x, y, 97 - k, 96 - k, 175);
        }
        if (k > 90)
        {
            setcolor(14);
            pieslice(x, y, 451 - k, 450 - k, 175);
            sound(440);
            delay(10);
            nosound();
            setcolor(0);
            pieslice(x, y, 457 - k, 456 - k, 175);
        }
        if (k == 360)
            k = 0;
        k = k + 6;
    }
    else
    {
        if (k <= 90)
        {
            pieslice(x, y, 91 - k, 90 - k, 175);
            setcolor(0);
            pieslice(x, y, 97 - k, 96 - k, 175);
        }
        if (k > 90)
        {
            setcolor(14);
```

```
            pieslice(x, y, 451 - k, 450 - k, 175);
            setcolor(0);
            pieslice(x, y, 457 - k, 456 - k, 175);
        }
    }
    setcolor(4);
    if (j == 60 * 5)
    {
        if (l <= 90)
        {
            setcolor(4);
            pieslice(x, y, 91 - l, 90 - l, 175);
            sound(880);
            delay(20);
            nosound();
            setcolor(0);
            pieslice(x, y, 93.5 - l, 92.5 - l, 175);
        }
        if (l > 90)
        {
            setcolor(4);
            pieslice(x, y, 451 - l, 450 - l, 175);
            sound(880);
            delay(20);
            nosound();
            setcolor(0);
            pieslice(x, y, 453.5 - l, 452.5 - l, 175);
        }
        if (l == 360)
            l = 0;
        l = l + 2.5;
    }
    else
    {
        if (l <= 90)
        {
            setcolor(4);
            pieslice(x, y, 91 - l, 90 - l, 175);
            setcolor(0);
            pieslice(x, y, 93.5 - l, 92.5 - l, 175);
        }
        if (l > 90)
        {
            setcolor(4);
            pieslice(x, y, 451 - l, 450 - l, 175);
            setcolor(0);
            pieslice(x, y, 453.5 - l, 452.5 - l, 175);
        }
    }
    if (i <= 90)
    {
        setcolor(10);
        pieslice(x, y, 91 - i, 90 - i, 175);
        delay(1000);
        sound(220);
        delay(5);
        nosound();
        setcolor(0);
        pieslice(x, y, 91 - i, 90 - i, 175);
    }
    if (i > 90)
    {
        setcolor(10);
        pieslice(x, y, 451 - i, 450 - i, 175);
        delay(1000);
        sound(220);
        delay(5);
        nosound();
        setcolor(0);
        pieslice(x, y, 451 - i, 450 - i, 175);
    }
    if (i == 360)
        i = 0;
    i = i + 6;
    s = i / 6;
    if (j == 60 * 5)
        j = 0;
    j++;
    if (m == 60)
        m = 0;
    m++;
    if (s == 60)
    {
        m1++;
        s = 0;
    }
    if (h == 24)
```

```
            h = 0;
        if (m1 == 60)
        {
            h++;
            m1 = 0;
        }
        gotoxy(62, 4);
        printf("%02d", h);
        gotoxy(68, 4);
        printf("%02d", m1);
        gotoxy(74, 4);
        printf("%02d", s);
        s++;
    }
    getch();
    closegraph();
    return 0;
}
```

OUTPUTES





```
CLOCK

00 03 15

PRESS ANY KEY FOR STOP
```

[redacted]

CONCLUSION AND ENHACEMENTS

The project was started with modest aim with no prior experience in any programming projects as this, but ended up in learning many things, fine tuning the programming skills and getting into the real world of software development with an exposure to corporate environment. During the development of any software of significant utility, we are forced with the tradeoff between speed of execution and amount of memory consumed. This is simple interactive application. It is extremely user friendly and has the features, which makes simple graphics project. It has an open source and no security features has been included.

The user is free to alter the code for feature enhancement. Checking and verification of all possible types of the functions are taken care. Care was taken to avoid bugs. Bugs may be reported to creator as the need.

Further this project can be enhanced by adding few more options i,e menus in game. Using this we can design a 3D game which contains cube instead of single window and multiple number of balls which are randomly moving and all faces of cube is considered as wall.

BIBLIOGRAPHY

Reference Books and E-books

- **EDWARD ANGEL**

  Interactive Computer Graphics, 5th edition, universities of New Mexico.

- **Websites for Reference**
  - www.opengl.org/resources/code/samples/redbook
  - www.wikipedia.org

Reported By:

AMIT KUMAR VERMA

BC2021183