Report On

Analog Clock

Submitted in partial fulfillment of the requirements of the Course project in
Semester III of Second Year Artificial Intelligence and Data Science

by
Yash Patil(Roll No. 45)
Preeti Prajapati (Roll No. 48)
Dhruv Gharat(Roll No. 11)

Supervisor
Ms Sejal D'mello

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “Title of the project” is a bonafide work of "Dhruv Gharat (Roll No. 11), Yash patil(Roll No. 46), Preeti Prajapati(Roll No.48), submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester III of Second Year Artificial Intelligence and Data Science engineering.

Supervisor

Prof. Name Surname

Dr. Tatwadarshi P. N.
Head of Department

Table of Contents

Pg. No

Abstract

This C program presents a graphical simulation of an analog clock using the Turbo C graphics library. The clock features hour, minute, and second hands and is displayed in a graphical window. The program utilizes fundamental graphics functions to create a clock face with hour markings, and it employs trigonometric calculations to simulate the movement of the clock hands.

The main components of the code include three functions:

1. `dial` Function:
   - Draws the clock face, including hour markings, using the graphics library.
   - Utilizes filled ellipses to represent the clock background and hour markings.
   - Displays the numbers 1 to 12 at appropriate positions on the clock face.

2. `minhand` Function:
   - Draws the minute hand based on the input time parameter.
   - Uses trigonometric calculations to determine the position of the minute hand.
   - Updates the clock face by connecting the previous and current positions of the minute hand.

3. `sechand` Function:
   - Draws the second hand based on the input time parameter.
   - Similar to the minute hand, utilizes trigonometric calculations to determine the position.
   - Updates the clock face to display the movement of the second hand.

The `main` function initializes the graphics window, calls the `dial` function to draw the clock face, and then enters a loop to simulate the passage of time. Inside the loop, the `minhand` and `sechand` functions are called to update the clock hands every second. The loop continues until a key is pressed, allowing the user to exit the program.

It's important to note that the code relies on the Turbo C graphics library, which may pose compatibility issues with modern compilers and operating systems. The usage of outdated header files like `dos.h` and `conio.h` also limits its portability

Problem Statement

In today's digital age, graphical representations and animations are fundamental in various applications, from simple games to complex simulations. Understanding the basics of how these animations are created and rendered is essential for budding software developers and computer enthusiasts. The C programming language, although traditionally used for system-level programming, also offers libraries that allow graphical representations.

Problem:

Develop a C program to simulate an analog clock using the Turbo C graphics library. The program should create a graphical representation of a clock with minute, and second hands. The clock face should be displayed in a graphical window, and the hands should move in a realistic manner to represent the passage of time.

Module Description

Header Files: We have used several header files according to our need

"math.h" provides us declarations and definitions for various mathematical functions. These functions cover a wide range of mathematical operations, including basic arithmetic, trigonometry, logarithmic, exponential, and more.

"dos.h" header file specific to the DOS operating system and is not a standard C library. It provides functions and constants that are used for various system-specific operations, primarily related to disk operations and interrupt handling. The functions in dos.h are specific to the MS-DOS environment and are not portable to modern operating systems.

"graphics.h"

The graphics.h header file is specific to the Turbo C compiler and is used for graphics programming in a DOS environment. It provides functions and constants for creating graphical applications, such as drawing shapes, lines, and text on the screen.

The **dial()** function is responsible for drawing the clock dial with markings for hours.
The **sechand()** and **minhand()** functions are used to draw the second hand and minute hand, respectively. They utilize trigonometric functions to calculate the endpoint of each hand based on the current time.
The main function initializes the graphics mode using **initgraph** and draws the clock dial using the dial function.

The **setlinestyle** function in graphics programming is used to set the style for drawing lines. It allows you to specify the pattern of dashes and dots for the lines that you draw on the screen. This function is part of the graphics.h library in Turbo C.

The **setlinestyle** function in graphics programming is used to set the style for drawing lines. It allows you to specify the pattern of dashes and dots for the lines that you draw on the screen. This function is part of the graphics.h library in Turbo C.

fillellipse() function is used to draw a filled ellipse on the screen with the specified center coordinates and radii.

outtext() function is used to display text on the screen at the current graphics position.

The program uses **delay(1000)** to introduce a one-second delay between each iteration, simulating the passage of time.

This program also uses **kbhit()** function If a key is pressed during the animation (checked with kbhit()), the loop breaks, and the program exits.

The graphics mode is closed using **closegraph()** before the program exits.

The **setcolor()** function in graphics programming is used to set the current drawing color for subsequent graphics operations. It's part of the graphics.h library in Turbo C. This function allows you to choose a color from a predefined set of colors.

Brief Description of Software and Hardware Used and Its Programming

Software used:

1. Turbo C Compiler: This is an integrated development environment (IDE) and compiler for the C programming language. It's often used for developing DOS applications. The Turbo C compiler provides the graphics.h and dos.h header files, essential for our graphical project.

2. BGI (Borland Graphics Interface): An integral part of Turbo C, BGI provides the necessary functions and routines to draw various shapes, control colors, and manage graphical content. Our project's graphical representation heavily relies on BGI functions such as `initgraph()`, `line()`, `arc()`, etc.

3. DOSBox: An x86 emulator with built-in DOS, useful for running Turbo C on modern operating systems, especially if native support is missing.

Hardware used:

No hardware used in this project

Programming:

1. Graphical Initialization: Using the `initgraph()` function, the BGI mode is initiated, establishing a graphical environment where we can draw and animate objects.

2. Delay Mechanism: The `delay()` function introduces a slight pause in each iteration of the loop. This ensures that the car's movement is smooth and visible to the human eye. Without this delay, the car would move too quickly, making the animation hard to perceive.

CODES

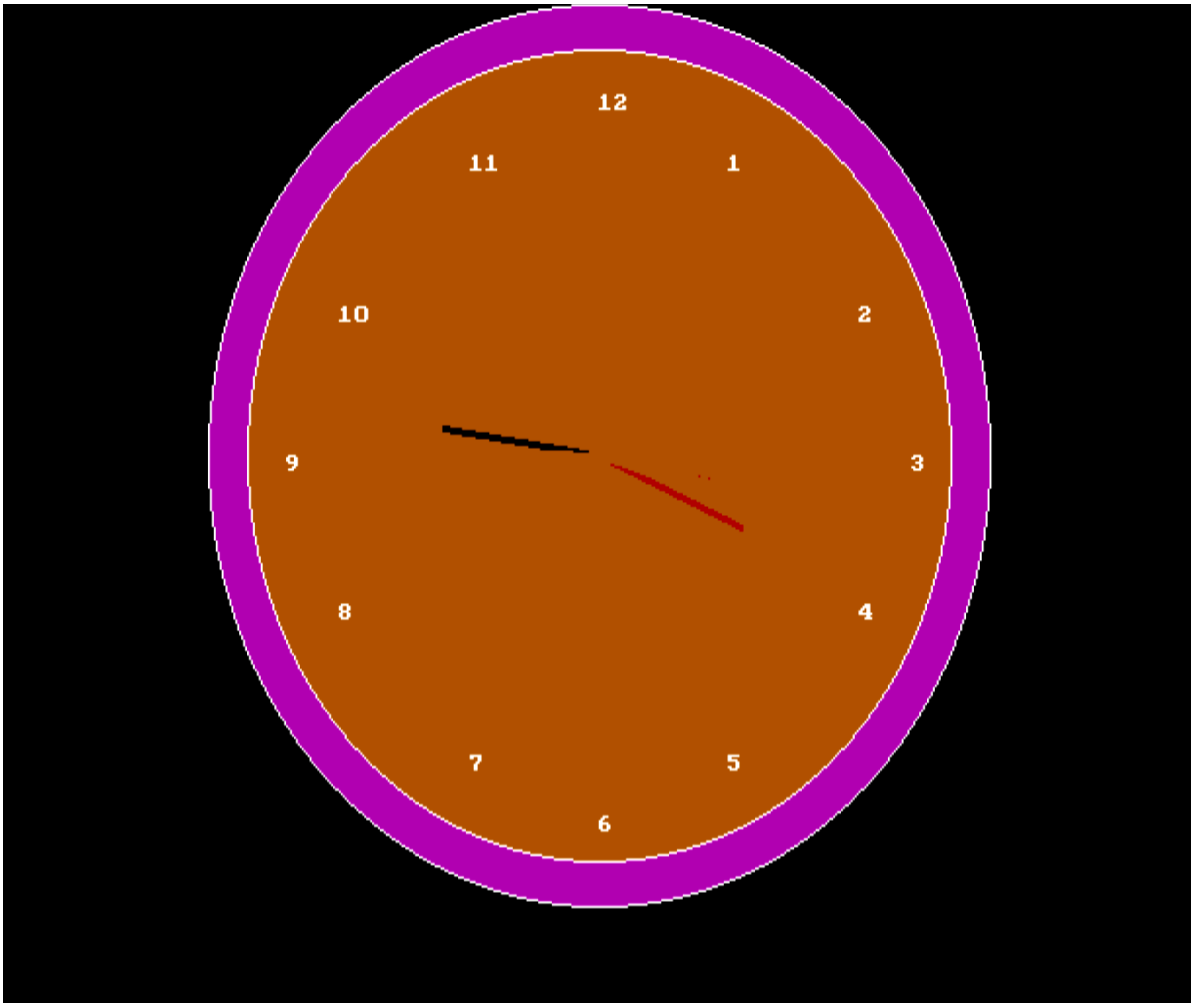
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
#define WBC 5
//^watchbackcolor
#define X 200
#define Y 200
void dial(int x,int y);
void sechand(int timemminute);
void minhand(int t)
{
int x1,y1;
setlinestyle(0,0,3);
x1=X+(80cos(t0.1047));
y1=Y+(80sin(t0.1047));
setcolor(BLACK);
line(X, Y, x1, y1);
setcolor(WBC+1);
line(X,Y,X+80cos((t-1)0.1047),Y+80sin((t-1)0.1047));
circle(X,Y,4);
}
void sechand(int t)
{
int x1,y1;
setlinestyle(0,0,3);
x1=X+(80cos(t0.1047));
y1=Y+(80sin(t0.1047));
setcolor(RED);
line(X, Y, x1, y1);
setcolor(WBC+1);
```

```
line(X,Y,X+100cos((t-1)0.1047),Y+100sin((t-1)0.1047));
circle(X,Y,4);
}
void dial(int x,int y)
{
int const size=200;
setfillstyle(1,WBC);
fillellipse(x,y,size,size);
setfillstyle(1,WBC+1);
fillellipse(x,y,size-20,size-20);

outtextxy(x,y-(size-40),"12");
outtextxy(x,y+(size-40),"6");
outtextxy(x+(size-40),y,"3");
outtextxy(x-(size-40),y,"9");
outtextxy(x+size/3,y-2size/3,"1");
outtextxy(x+2size/3,y-size/3,"2");
outtextxy(x+2size/3,y+size/3,"4");
outtextxy(x+size/3,y+2size/3,"5");
outtextxy(x-size/3,y+2size/3,"7");
outtextxy(x-2size/3,y+size/3,"8");
outtextxy(x-size/3,y-2size/3,"11");
outtextxy(x-2size/3,y-size/3,"10");
circle(x,y,4);
}
void main()
{
int gd=DETECT,gm,i,j,flag=1;
initgraph(&gd,&gm,"C:\\turboc3\\bgi)");
dial(200,200);
do
{
minhand(i);
for(j=0;j<60;j++)
```

```
{
sechand(j);
delay(1000);
if(kbhit())
{
flag=0;
break;
}
}
i++;
}
while(flag);
closegraph();
getch();
}
```

Results And Conclusion



CONCLUSION

This project shows the graphical representation of a working analogue clock.
The objective of this program is to implement simple and basic functions of graphics. Computer graphics plays a major role in today's world where visualization takes the upper hand as compared to textual interaction. This is largely true as we can see user interface becoming more and more attractive all thanks to major leaps in the fields of computer graphics. The project is implemented using various header files provided by C.

The above argument is equally justified in the fields of computer simulation which involve complex graphics being highlighted at its peak. It is becoming more and more popular and the constant dive to improve particular system efficiency by studying its simulated model attracts more people towards it.

References:

1.Computer Graphics Using OpenGL – F.S Hill, Jr.,2nd Edition, Pearson Education ,2001

2.Computer Graphics – James D Foley , Andries Van Dam , Steven K Feiner , John F Hughes , Addison – Wesley 1997

3.Computer Graphics – Donald Hearn and Pauline Baker ,Pearson Education ,2003

4.Smith, J. (2005). Introduction to Computer Graphics.

Online references:

- www.3dcodingtutorial.com
- www.openGL.org
- www.nehe.com