

**Shri. Tuljabhavani College of Engineering, Tuljapur.**

**Synopsis**

**Title / Topic:** Python GUI based analog clock

**INTRODUCTION**

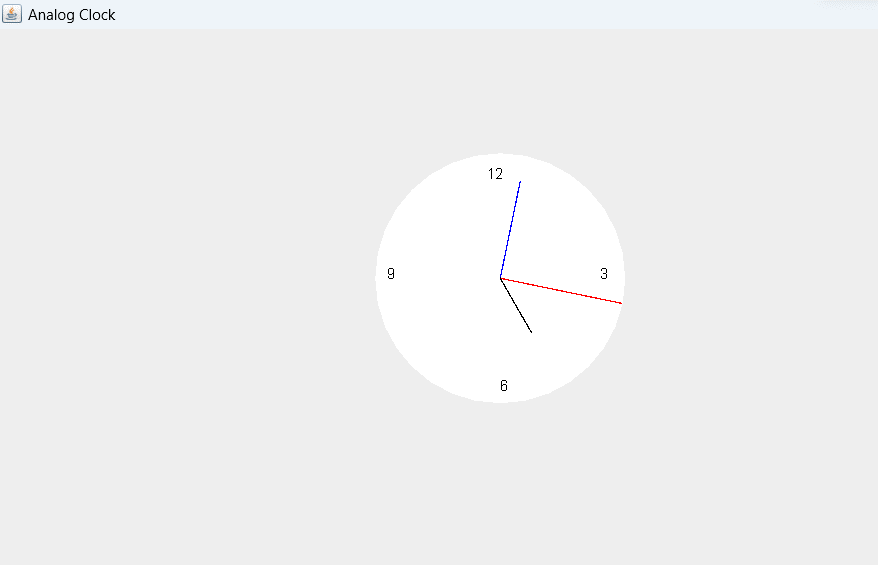
The project's primary objective is to showcase the capabilities of Python and Tkinter in building interactive and engaging applications. The application can be useful for various purposes, such as timekeeping, education, or entertainment. Moreover, the project can serve as a starting point for other GUI-based applications, as it provides an excellent foundation for learning GUI programming in Python.

Python is multi-paradigm programming language. Object-Oriented programming and structured programming fully supported and many of its features support functional programming and aspect-oriented programming.

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python used for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.



import java.awt.Color;

import java.awt.Dimension;

import java.awt.Graphics;

import java.util.Calendar;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class AnalogClock extends JPanel {

public static void main(String[] args) {

JFrame frame = new JFrame("Analog Clock");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(new Dimension(800, 400));

frame.setBackground(new Color(50, 50, 50));

AnalogClock clock = new AnalogClock();

frame.add(clock);

frame.setVisible(true);

new Thread(() -> {

while (true) {

clock.repaint();

clock.delayAnimation();

}

}).start();

}

// Animating the clock

private void delayAnimation() {

try {

// Animation delay is 1000 milliseconds

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

// Paint the clock

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

// Get the system time

Calendar time = Calendar.getInstance();

int hour = time.get(Calendar.HOUR\_OF\_DAY);

int minute = time.get(Calendar.MINUTE);

int second = time.get(Calendar.SECOND);

// 12-hour format

if (hour > 12) {

hour -= 12;

}

// Draw clock body center at (400, 200)

g.setColor(Color.white);

g.fillOval(300, 100, 200, 200);

// Labeling

g.setColor(Color.black);

g.drawString("12", 390, 120);

g.drawString("9", 310, 200);

g.drawString("6", 400, 290);

g.drawString("3", 480, 200);

// Declaring variables to be used

double angle;

int x, y;

// Second hand's angle in Radians

angle = Math.toRadians((15 - second) \* 6);

// Position of the second hand with length 100 units

x = (int) (Math.cos(angle) \* 100);

y = (int) (Math.sin(angle) \* 100);

// Red color second hand

g.setColor(Color.red);

g.drawLine(400, 200, 400 + x, 200 - y);

// Minute hand's angle in Radians

angle = Math.toRadians((15 - minute) \* 6);

// Position of the minute hand with length 80 units

x = (int) (Math.cos(angle) \* 80);

y = (int) (Math.sin(angle) \* 80);

// Blue color minute hand

g.setColor(Color.blue);

g.drawLine(400, 200, 400 + x, 200 - y);

// Hour hand's angle in Radians

angle = Math.toRadians((15 - (hour \* 5)) \* 6);

// Position of the hour hand with length 50 units

x = (int) (Math.cos(angle) \* 50);

y = (int) (Math.sin(angle) \* 50);

// Black color hour hand

g.setColor(Color.black);

g.drawLine(400, 200, 400 + x, 200 - y);

}

}