Universal Clock Design Document

Designed by: Pranav Tripathi

Course: ICS4UI

Teacher: Mr. Schattman

Date: June 18, 2015

The Universal Clock is an animation of an analog clock that runs in real time. The clock is able to read the current time and animate it on the hands of a clock.

The Universal Clock supports 4 main time zones: Eastern Standard Time, Pacific Standard Time, Eastern European and Atlantic Time. The User can choose which time zone he wants before launching the program and the clock will automatically adjust its time to correspond. For User’s clarity, the clock will display the time zone that the clock is using on the screen. This way, if someone looks at the clock and is unsure why the time is different than the current time, he/she will be able to check which time zone the clock is displaying.

What the User Inputs:

Before Universal Clock runs, it will follow this procedure

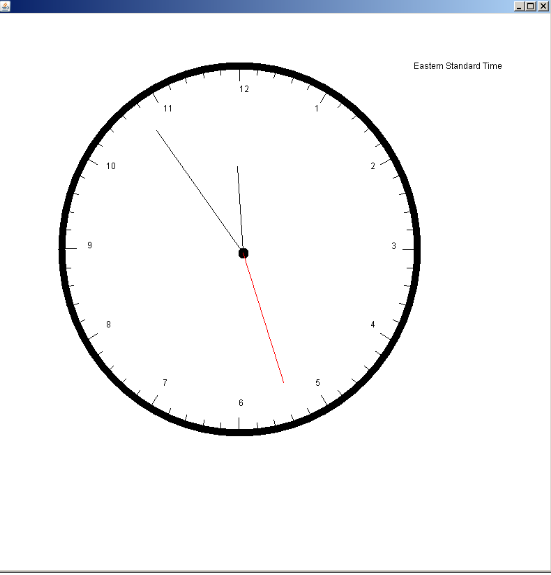
1) Universal Clock will ask the user, “What time zone would you like to use”. The user can respond with “EST” for Eastern Standard Time, “PST” for Pacific Standard Time, “EE” for Eastern European and “IST” for Indian Standard Time.

2) The clock then looks at the current time on the computer and makes the necessary changes for the time zone you selected.  
For Eastern Standard Time, the clock will display the current time on the computer.   
For Pacific Standard Time, it will subtract 3 hours from the current time  
For Eastern European Time, it will add 6 hours to the current time

For Indian Standard Time, it will subtract 3 hours and add 30 minutes.

The Clock’s Output

Here is a visual representation of the Universal Clock:



The clock runs in real time and uses angles to help in its animation

At the top right, it displays the time zone that the clock is displaying

Classes

The program has two classes:

1)ProjectClock- This is the main class. This class calls the function drawClock and sets up the screen

2)Clock- This is the class with the most methods. It deals with all the necessary requirements to get the clock working and also deals with setting up the different time zones.

Further breakdown is given below:

Imports Used:

import java.awt.Color;

import java.awt.Graphics;

import java.util.Scanner;

import javax.swing.JFrame;

import java.util.Calendar;

import java.util.GregorianCalendar;

Main Class: ProjectClock extends Clock

Fields: None

Methods in the Class:

1)Paint- Draws the screen background and calls upon the function drawClock to run the program

2)Main- Asks user to input Time zone and sets up the screen for the animation

2nd Class: Clock extends JFrame

Fields :

(double)- hour, minute, second, count, Mincount, secAngle, MinuteAngle, HourAngle

(String) – timezone

(int)-number=12, GlobalRadius, angle

Calendar time = new GregorianCalendar();

Methods in the Class:

1. **public void sleep(int duration)-** Used in the drawSecondHand method to animate one second delays
2. **public void drawClock(Graphics g, int x, int y, int radius, String Timezone)-** Is the most important method in the class. Calls most of the other methods in the class to animate the clock
3. **public void drawTick(Graphics g, int x, int y, double angle, int length)-** Draws all the ticks on the perimeter of the clock
4. **public void drawTickwithNums(Graphics g, int x, int y, double angle, int length)-**Same as drawTick, except it draws in the numbers(1-12) where they belong on the clock
5. **public void drawSecondHand(Graphics g, int x, int y, double angle, int length)-** Draws the second hand on the clock and makes use of sleep(1000) to animate the clock in real time.
6. **public void drawOtherHand(Graphics g, int x, int y, double angle, int length) –** Draws the hour and minute hands depending on what length is described. These do not use the sleep method and are instead animating by using a count feature inside drawClock.
7. **public void clearOtherHand(Graphics g, int x, int y, double angle, int length)-** Used to erase the previous location of the hour and minute hands.
8. **public void ZoneView(Graphics g, int ZoneX, int ZoneY)-** *It displays the Time zone that the clock is working with on the screen*