CALCULATOR

Prepared by:

Ahmed Abd El_ghany (Leader)

Ahmed Adel Mahmoud Kishk

Fady Nashaat Sorial

Marwa Abdullah Mohammed

Salma Salah El_Din Abdullah

Observation:

Dr./ Eman Hesham

3 / 1 / 2022

Contents

Introduction

Idea of Project

Engineers, students and many people need to do complex calculations very fast so we need a calculator with simple GUI and greet features to do these calculations.

We can control our calculator using Keypad or using Mouse and Keyboard of the Laptop, and in this article we will explain how we did it and the Tool Boxes which are used to achieve this product; as

- First Tool box is the Hard ware used.
- Second one is the Software items.
- Idea ware and the programming methodology.

About the Project

Here we will talk about the **Features** of this Calculator as it can work at many modes:

- Scientific Mode
- Graphing Mode
- Programmer Mode
- Converter Mode:
  - Length
  - Temperature

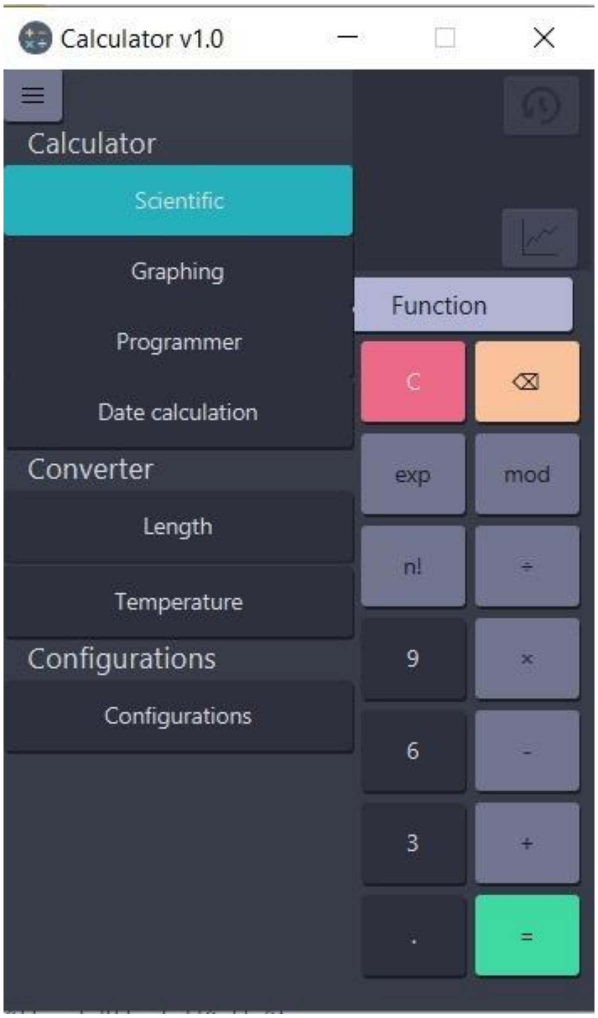


Figure 1.1 Modes of Calculator

And many another Features like Date calculations, and also supports Graphs of Functions, and user can switch between them; and there is a History file to save all the operations the user done in all modes, and all of these will be discussed briefly in the following sections.

For handling all these Features we used many Software Tools Like Java FXML Language, NetBeans IDE and Scene Builder Program and Arduino IDE and also Hard Ware Tools as The Arduino Micro Controller and Key pad.

TOOL BOX

It is mainly subdivided into two types of Tools:

Software Tools

Programming Languages

JavaFX is a Java library used to develop Desktop applications as well as Rich Internet Applications (RIA). The applications built in JavaFX, can run on multiple platforms including Web, Mobile and Desktops.

JavaFX is intended to replace swing in Java applications as a GUI framework. However, it provides more functionalities than swing. Like Swing, JavaFX also provides its own components and doesn't depend upon the operating system. It is lightweight and hardware accelerated. It supports various operating systems including Windows, Linux and Mac OS.

Features of JavaFX:

Java Library: It is a Java library which consists of many classes and interfaces that are written in Java.

FXML: A new language is added to JavaFX, it is The XML based declarative markup language; the coding can be done in FXML to provide the more enhanced GUI to the User.

Built in UI controls: JavaFX contains Built-in components which are not dependent on operating system. The UI component are just enough to develop a full featured application.

CSS like styling: JavaFX code can be embedded with the CSS to improve the style of the application. We can enhance the view of our application with the knowledge of CSS as we will present.

Rich Set of APIs: JavaFX provides a rich set of APIs to develop GUI applications.

High Performance Media Engine: The media pipeline supports the playback of web multimedia on a low latency. It is based on a G Streamer Multimedia framework.

Architecture of JavaFX Application of the project

As the Application of JavaFX consists from:

1. Node: The main actor of the application, and the visible component in our application.
2. Scene: The component that the nodes are displayed on it.
3. Stage: is the base for the scene, and nodes.

The application may has many scenes, but we should present one scene on the stage at a time.

And as the Calculator has many modes and user can switch between them, every mode is a separated scene, and every scene contains its suitable Nodes (labels, buttons and text Areas, ...) to interface with the user.

Important Libraries which are imported

- javafx.event : Provides the classes that deal with events and their handling
- javafx.fxml : Contains the set of classes that are responsible of loading hierarchy from mark - up
- javafx.scene.control: Contains for all javaFX components.

- Javafx.util: provides utilities classes like (Resource Bundle)

JavaFX CSS (Cascading Style Sheets):

In this project we use CSS, so we can define the color, size, font styles, spacing between the paragraph, alignment and many more thing for a web page so that it can look more precise and better. We can also configure the background of the application, layout, design and variety of display for the different size devises.

Adding Style Sheets to the Scene:

However, JavaFX provides us the facility to override the default style sheet and define our own styles for every node of the application. The Style-sheet we create, must have the extension **.css** and it must be located in the directory where the main class of the application resides.

Syntax: `scene.getStylesheet().add("path/Stylesheet.css");`

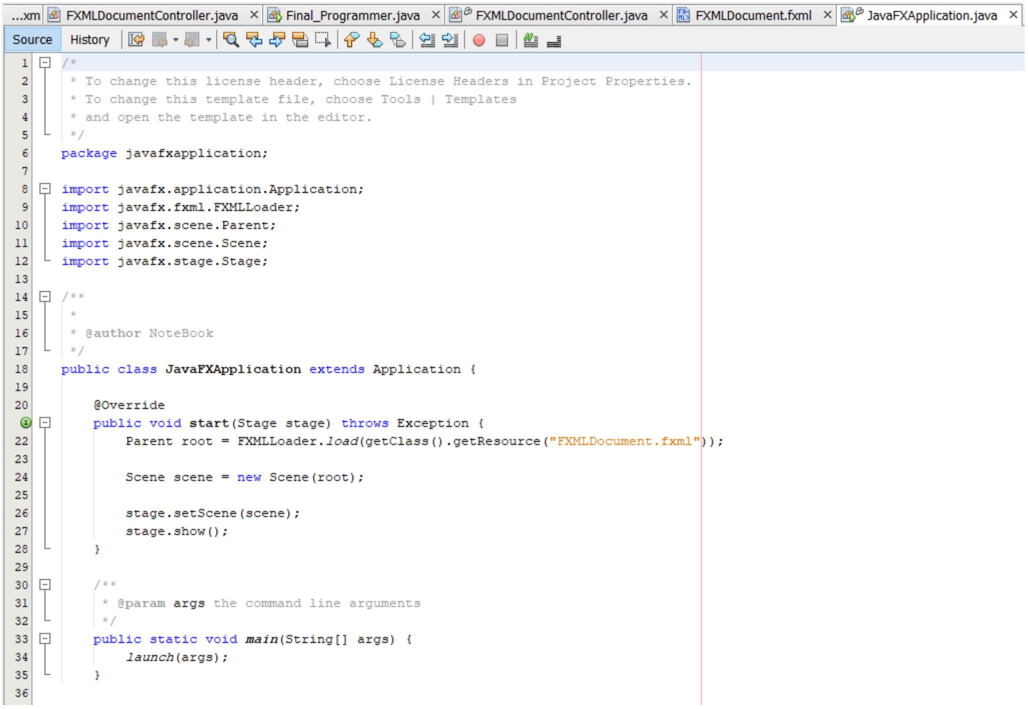
And in our project we have three main style sheets (Base, Button and text)

NetBeans

NetBeans IDE is a free and open source Integrated Development Environment for application development on Windows, Mac, Linux, and Solaris operating systems.

The IDE includes several project templates designed to support different types of development including web applications, general Java applications, and so forth, here we used:

- ***JavaFX Application with FXML*** - An empty JavaFX FXML-enabled application. which is a JavaFX project that is based on an FXML layout, and we can edit the FXML file using Scene Builder.



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javafxapplication;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

/**
 *
 * @author NoteBook
 */
public class JavaFXApplication extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("FXMLDocument.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }

```

Figure 2.1 JavaFX Application

Every JavaFX Application with FXML project has its source package which contain three main files:

- *FXML DocumentController.java*

  This is the controller file for handling Event Actions of UI controls and functions to be done; as it contains The Class that implements "Initializable"

- *FXML Document.fxml*

  FXML source file in which we define the user interface, It is marked up Using Scene Builder as we will explain in the following section.

- *'Project_Name'.java*

  Which contains the class that extends Application, and this class contains two main methods:

  - Start() method: which is over ridded
  - Main() method

But we add any files we need (as it is concluded) in the project in this source package like files .css and any defined classes.

Scene Builder

The Scene Builder project was created using JavaFX by Oracle and it is open source within the Open JavaFX project, it generates FXML markup which can be ported to an IDE.

JavaFX Scene Builder is a visual layout tool that lets users quickly design JavaFX application user interfaces, without coding.

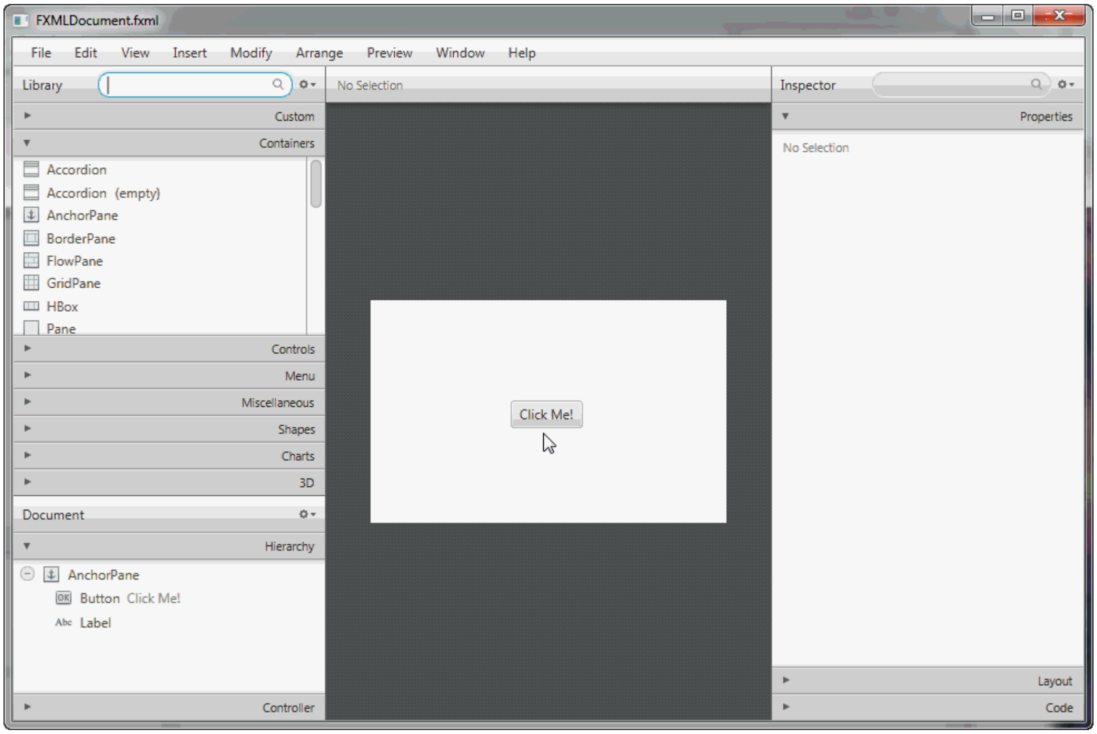


Figure 2.2 Scene Builder IDE

We can drag and drop UI components to a work area, modify their properties, apply style sheets, and the FXML code for the layout that they are creating is automatically generated in the background.

And for every node / Component we define its name “fx:id” and the name of Event handler function” On Action” which will be executed, in the code section

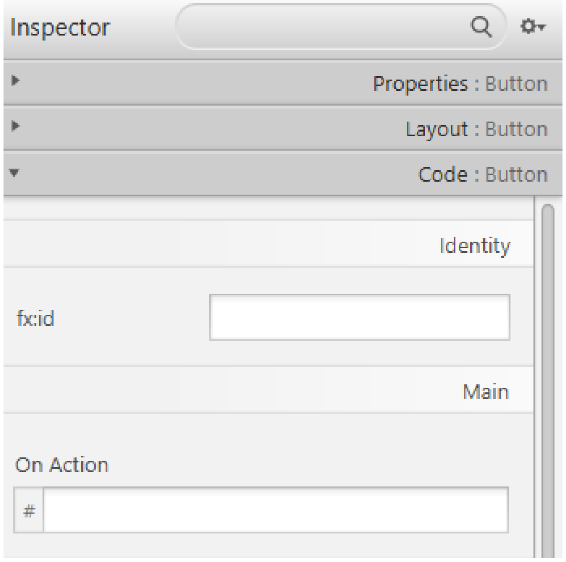


Figure 2.3 Inspector Section of Scene Builder

The result is an FXML file that can then be combined with a Java project by binding the UI to the application's logic.

From a Model View Controller (MVC) perspective:

- ❖ The FXML file, containing the description of the user interface, is the view.
- ❖ The controller is a Java class, optionally implementing the Initializable class, which is declared as the controller for the FXML file.
- ❖ The model consists of domain objects, defined on the Java side, that can be connected to the view through the 'controller'.

Hard Ware Tools

We used some hardware components to interface with the user:

- Two Keypads (4*4) keypad.
- (1*4) keypad.
- Arduino Mega Micro Controller.
- Mouse, to interface with the GUI at PC.

Arduino Micro Controller

The Arduino Mega is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button; and this Micro Supports all needed peripherals to handle our project.

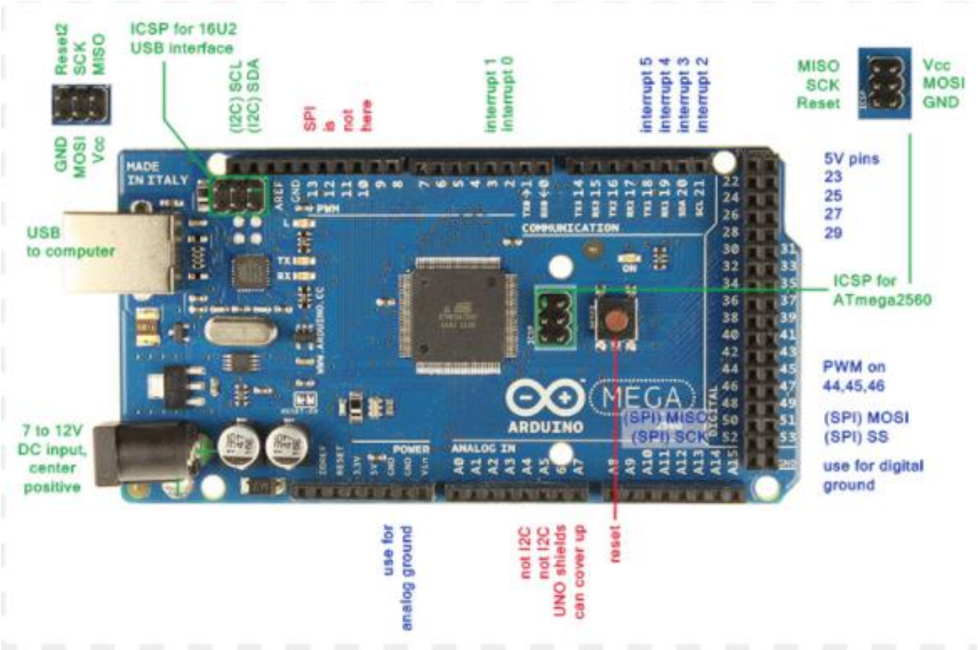


Figure 2.4 Arduino Mega 2560 Pin in/outs

Peripherals Which are used

- **DIO** (Digital Input Output) Peripheral

  The pins on the Arduino can be configured as either inputs or outputs.

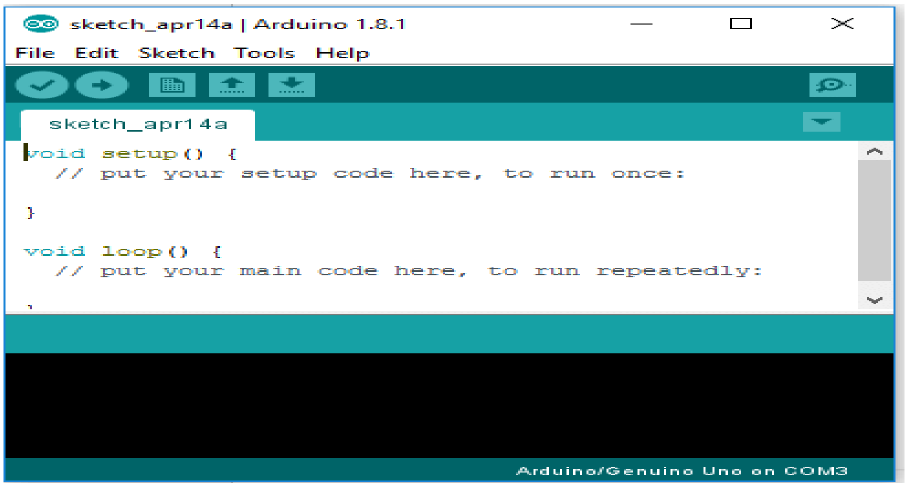
  Here using function PinMode() we can configure it as an input to receive the (High / Low) signals from key pads, as every button from Key pads is connected with a pin of the Arduino

- **UART** (Universal Asynchronous Receiver Transmitter) Serial Communication protocol.

  Used for communication between the Arduino board and a computer, so we can control the GUI of the calculator on PC using Key pads besides the Keyboard of the PC.

Arduino IDE:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board so we can use it for coding at Arduino Mega


sketch_apr14a | Arduino 1.8.1
File Edit Sketch Tools Help
sketch_apr14a
Arduino/Genuino Uno on COM3

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
```

Figure 2.5: Arduino IDE

Key Pad

A **keypad** is a block or pad of buttons set with an arrangement of digits, symbols, or alphabetical letters.

And here we used (4*4) key pad can be controlled using 8 pins of the arduino (Which is configured as input):

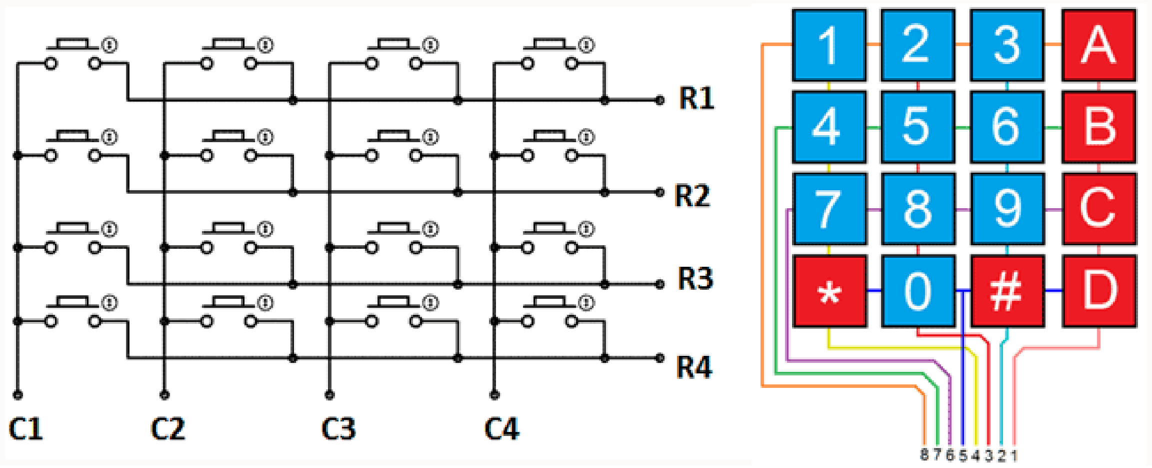


*Figure 2.6: Key Pad (4*4) Pins*

and in our project we used:

1) 2 (4*4) keypads controlled two modes

- The first mode is numbers and operations 0,1,2…...,9 and (+, -, *, /)
- The second mode use (2 ,4 ,6 ,8 ,5) as (UP, LEFT, RIGHT, DOWN, ENTER)

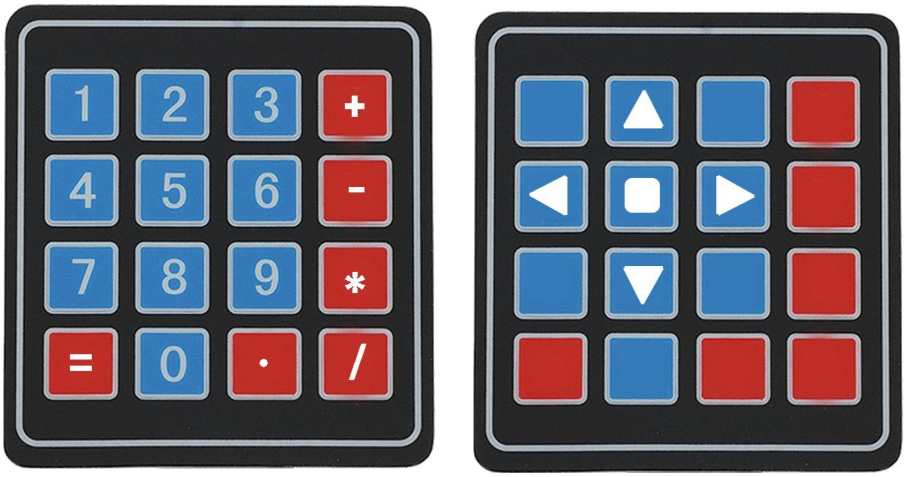
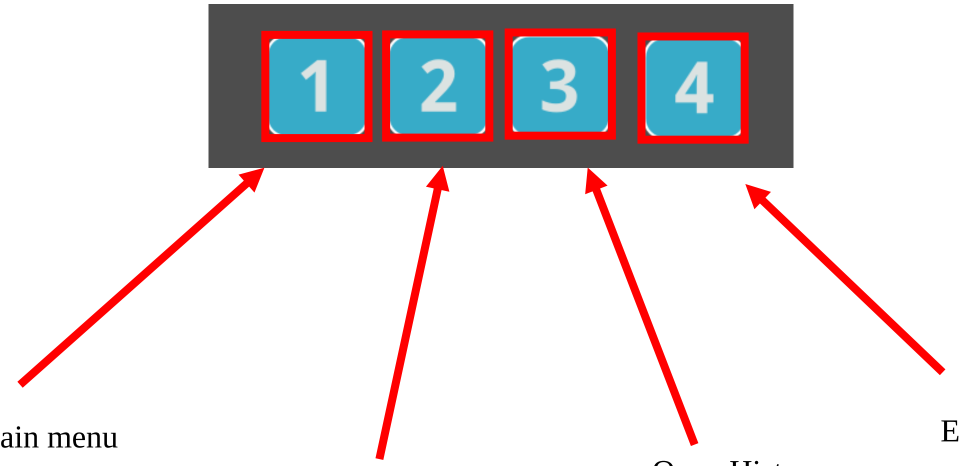


Figure 2.7 Key Pad Modes

2) 1*4 Keypad control different functions :



Open main menu

Activate or deactivate
Arrows mode

Open History

Exit the program

Figure 2.8 Key Pad for Functions of Calculator

Hardware Connections

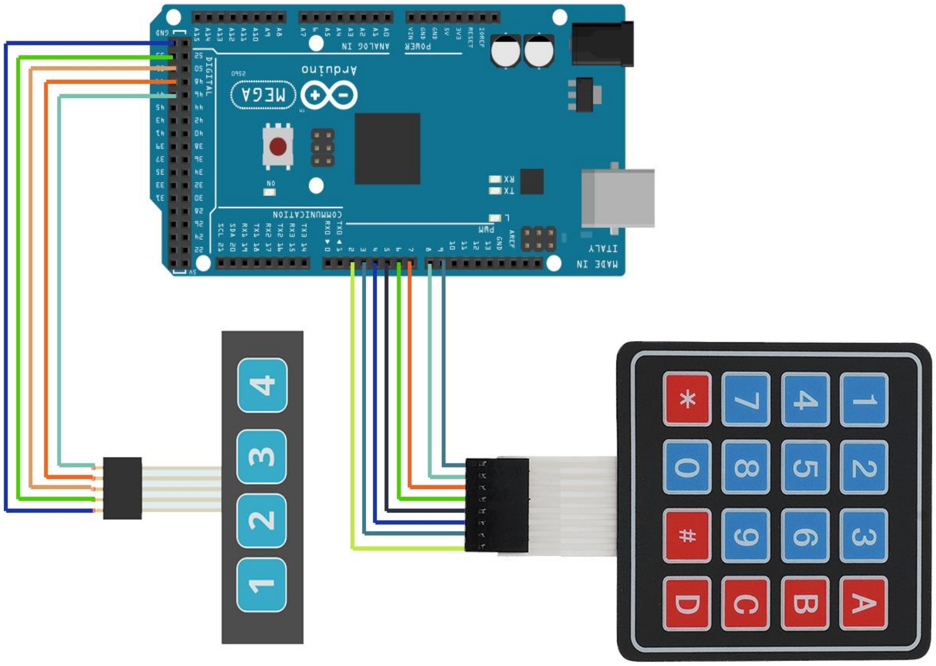


Figure 2.9: Connection of Hard ware of the calculator

Pins of Arduino	Key Pad1
p2	**1**
p3	**2**
p4	**3**
p5	**4**
p6	**5**
p7	**6**
p8	**7**
p9	**8**

Modes of Calculator

As we mentioned before the calculator can work at many modes to facilitate and arrange the calculations of user according to the purpose of him.

Scientific Mode

Features of this Mode

Here we provide the User with accessibility of:

- ✓ Many operations such as simple operations like: addition, subtraction, multiplication, division and also more complicated operations like square root, log, ln and Trigonometry functions like: sin, cos, sinh, cosh, sec and csc.

GUI OF the Scene

We have a variety of options which can be shown by accessing menu buttons and 2nd buttons provided in the GUI and illustrated in the following figures.

- Pressing 2nd button gives more operation options.

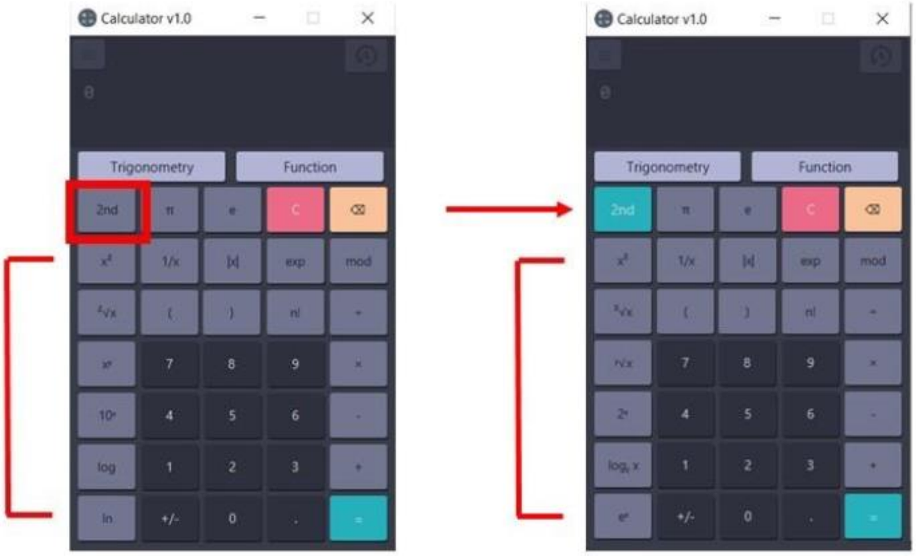
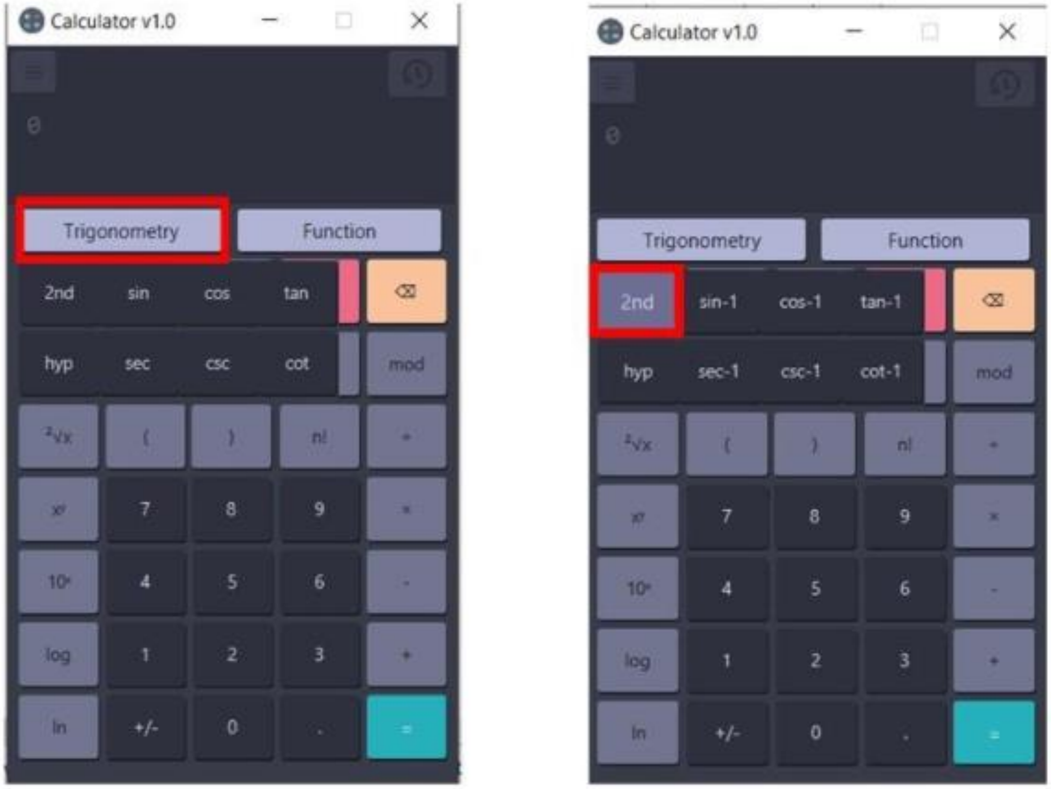


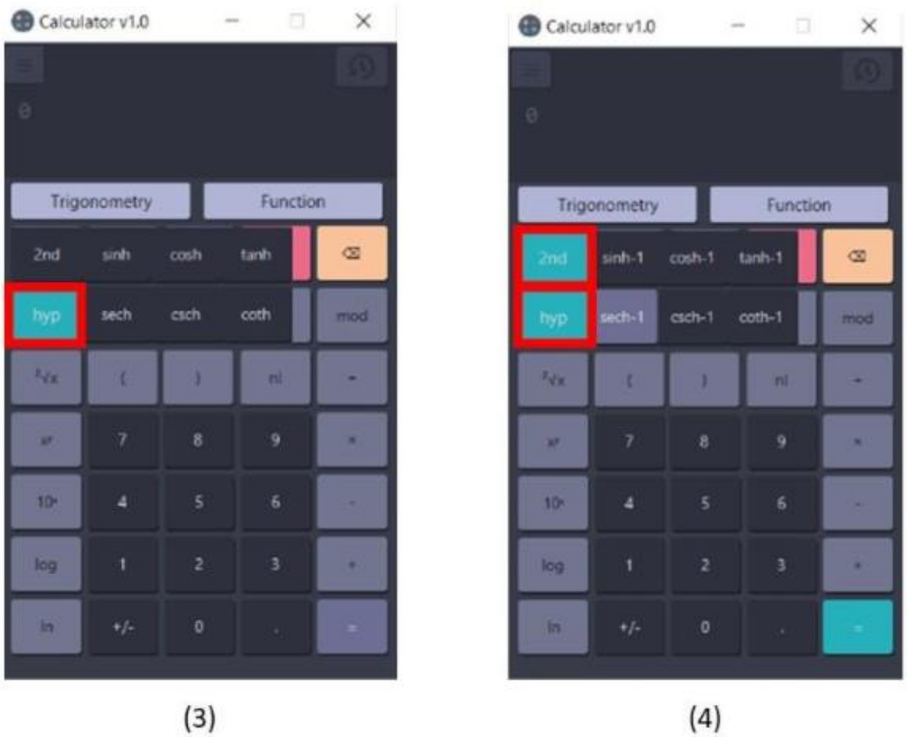
Figure 3.1.1: Scientific Mode GUI

- “Trigonometry” button gives us 6 options, and when we press 2nd button they transfer to the inverse functionality then “hyp” button gives us more option in addition to “hyp” button pressed with “2nd” button, so in total we have 24 options as specified in the following figures.



(1) (2)

Figure 3.1.2 GUI of Scientific Mode



(3)

(4)

Figure 3.1.3: GUI of Scientific Mode

- “Function” button gives us 4 more options, Absolute, ceil, floor and random numbers between 0 and 1.

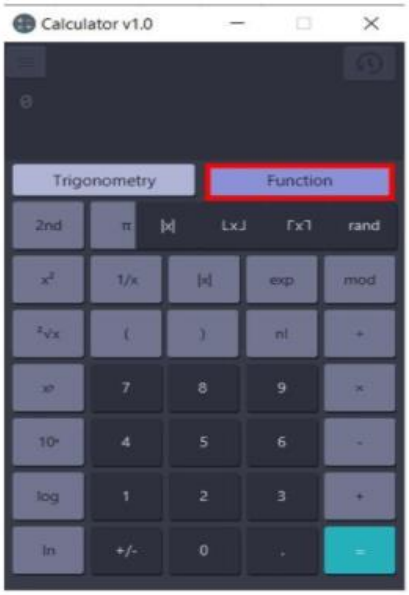
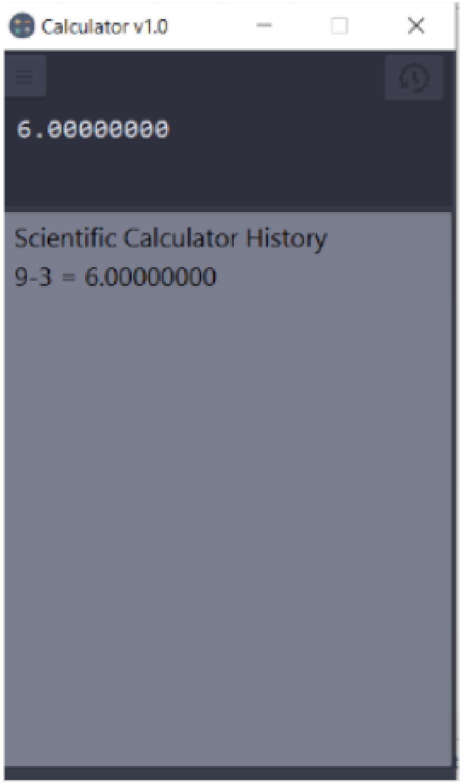


Figure 3.1.4: “Function Button” of Scientific Mode



6.00000000

Scientific Calculator History
9-3 = 6.00000000

Figure 3.1.5: “History” Of Calculator

- History Button appends each operation we calculate whither using server where we can have our history in txt file if the history server is on or only check it by clicking the button and it will be deleted after closing our application if server is off.

Algorithm of Scientific Mode

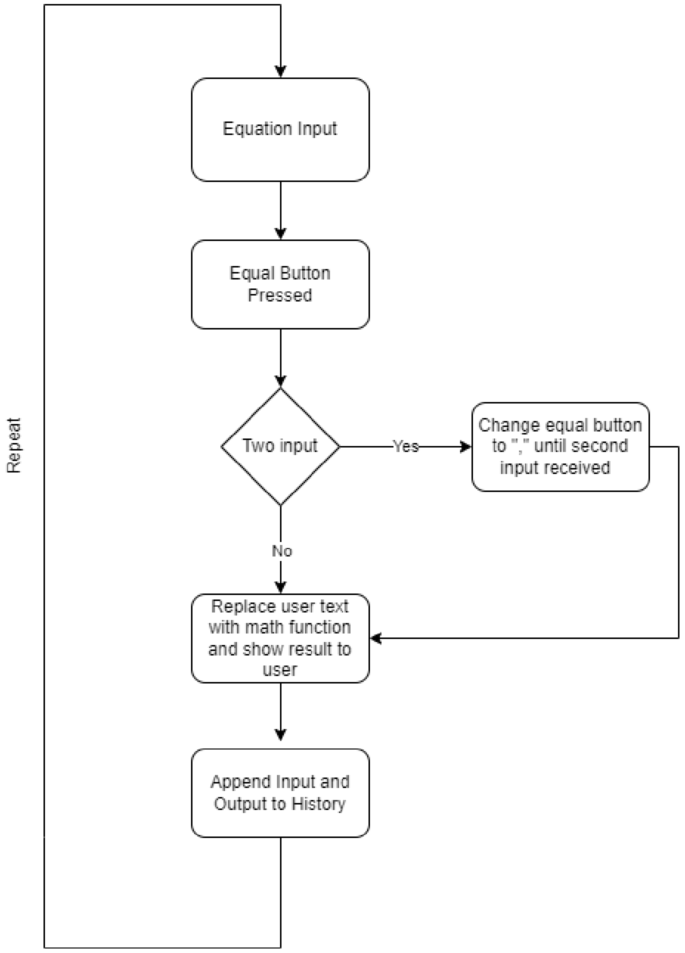

Equation Input
Equal Button Pressed
Two input
Yes
Change equal button to "," until second input received
No
Replace user text with math function and show result to user
Append Input and Output to History
Repeat

Figure 3.1.6: Flow Chart of Scientific Mode

Graphing Mode

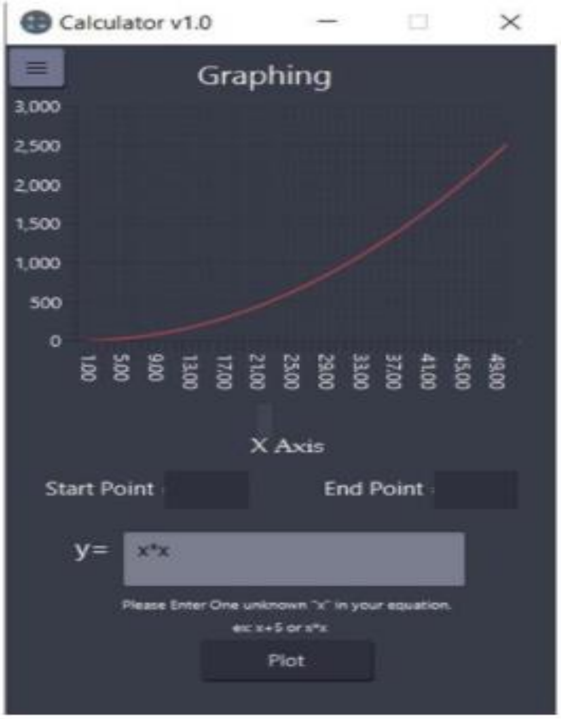
Features of this Mode

Here we provide the User with accessibility of:

- ✓ Graphing the equation entered by the user with one limitation that it contains only one unknown and it must be called x.
- ✓ Graphing on default range which starts from x=0 to x=50 with step=1.
- ✓ Graphing on different ranges where user can enter start point and end point where step=1.

GUI OF the Scene

- ➢ When user does not provide range, the default range for x is from 0 to 50 with step = 1.

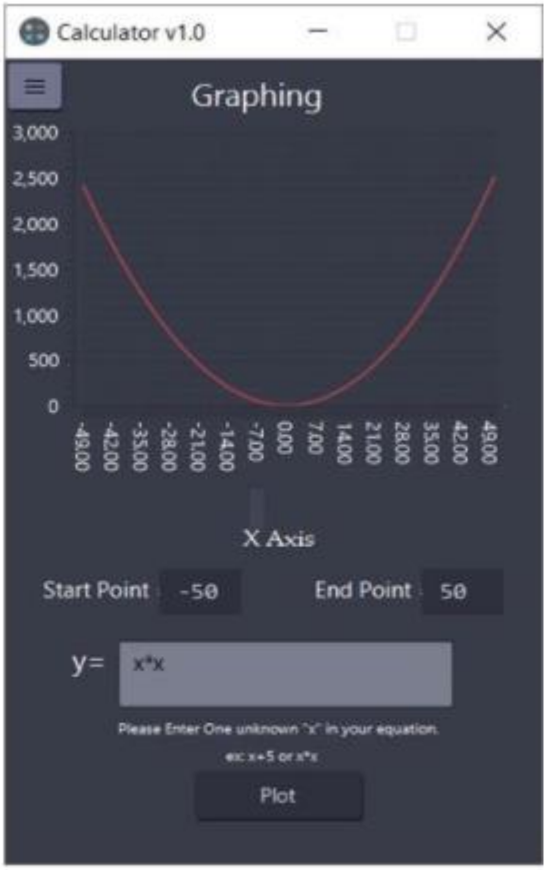


y= x*x

Please Enter One unknown "x" in your equation.
ex: x+5 or x*x

Figure 3.2.1: GUI of Graphing mode

- When user provide range.



Start Point -50 End Point 50

y= x*x

Please Enter One unknown "x" in your equation.

ex: x+5 or x*x

Figure 3.2.2: GUI of Graphing Mode

Algorithm of Graphing Mode

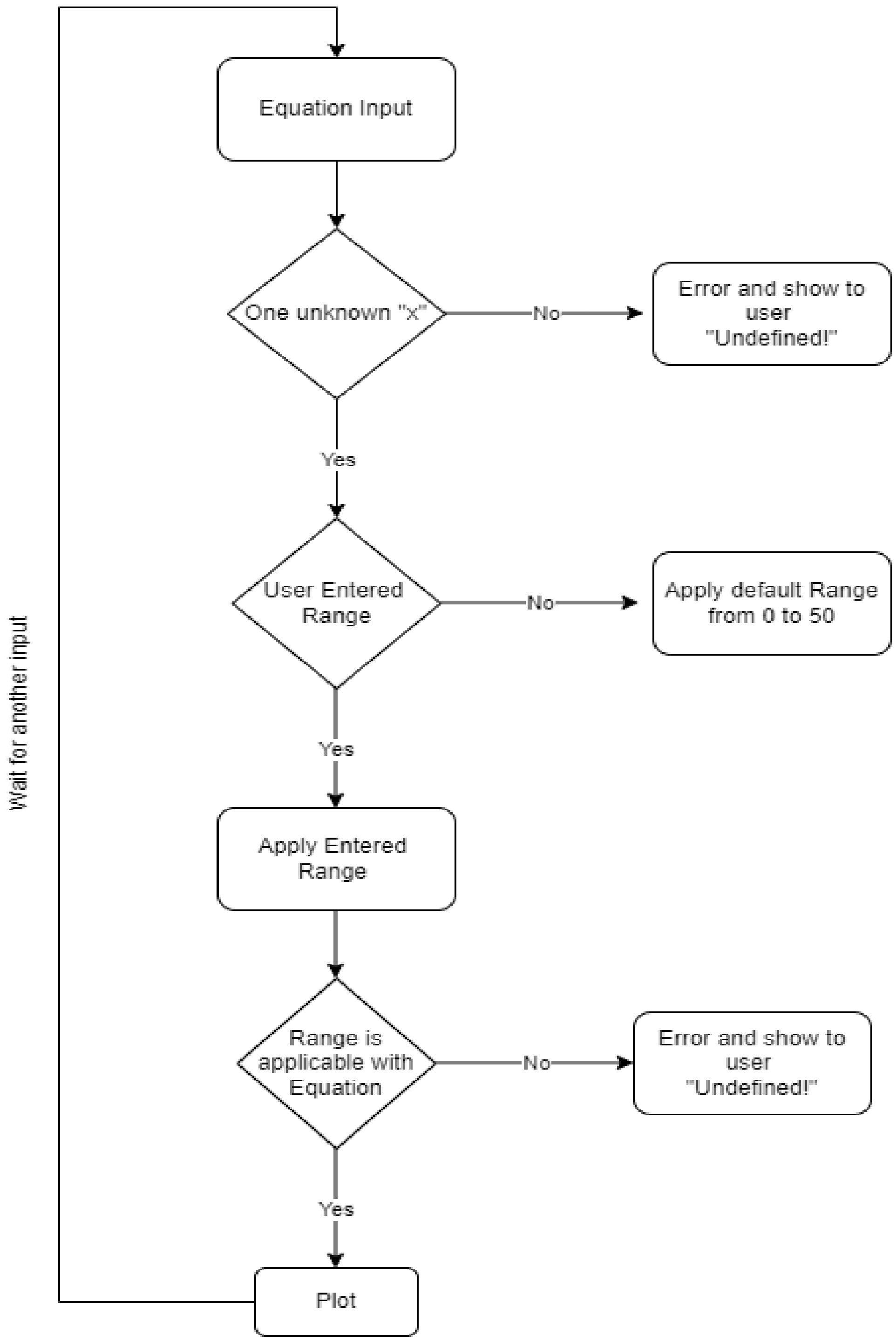

Equation Input
One unknown "x"
No
Error and show to user "Undefined!"
Yes
User Entered Range
No
Apply default Range from 0 to 50
Yes
Apply Entered Range
Range is applicable with Equation
No
Error and show to user "Undefined!"
Yes
Plot
Wait for another input

Figure 3.2.3: Flow Chart of Graphing Mode

Programmer Mode

Features of this Mode

Here we provide the User with accessibility of:

- ✓ Dealing with Binary, Hexadecimal, Octal and decimal Numbers and converting between them.
- ✓ Doing some Logic operations like: Anding (AND), Oring (OR), Inverting (NOT), and XOR between two numbers (Hexa, Binary, Octal or Decimal).

GUI OF the Scene

As the user choose Programmer Mode from the menu, this figure appears:

There are

1) Two Text Areas for Numbers input
2) 4 Buttons to define the Numbering System
3) 4 Buttons for the Logical operations (OR / NOT / AND / XOR)
4) Buttons from 0 to 9 are enabled as it works at decimal numbering system by default
5) Clear(C) button and Equal (=) Button
6) Button For showing history

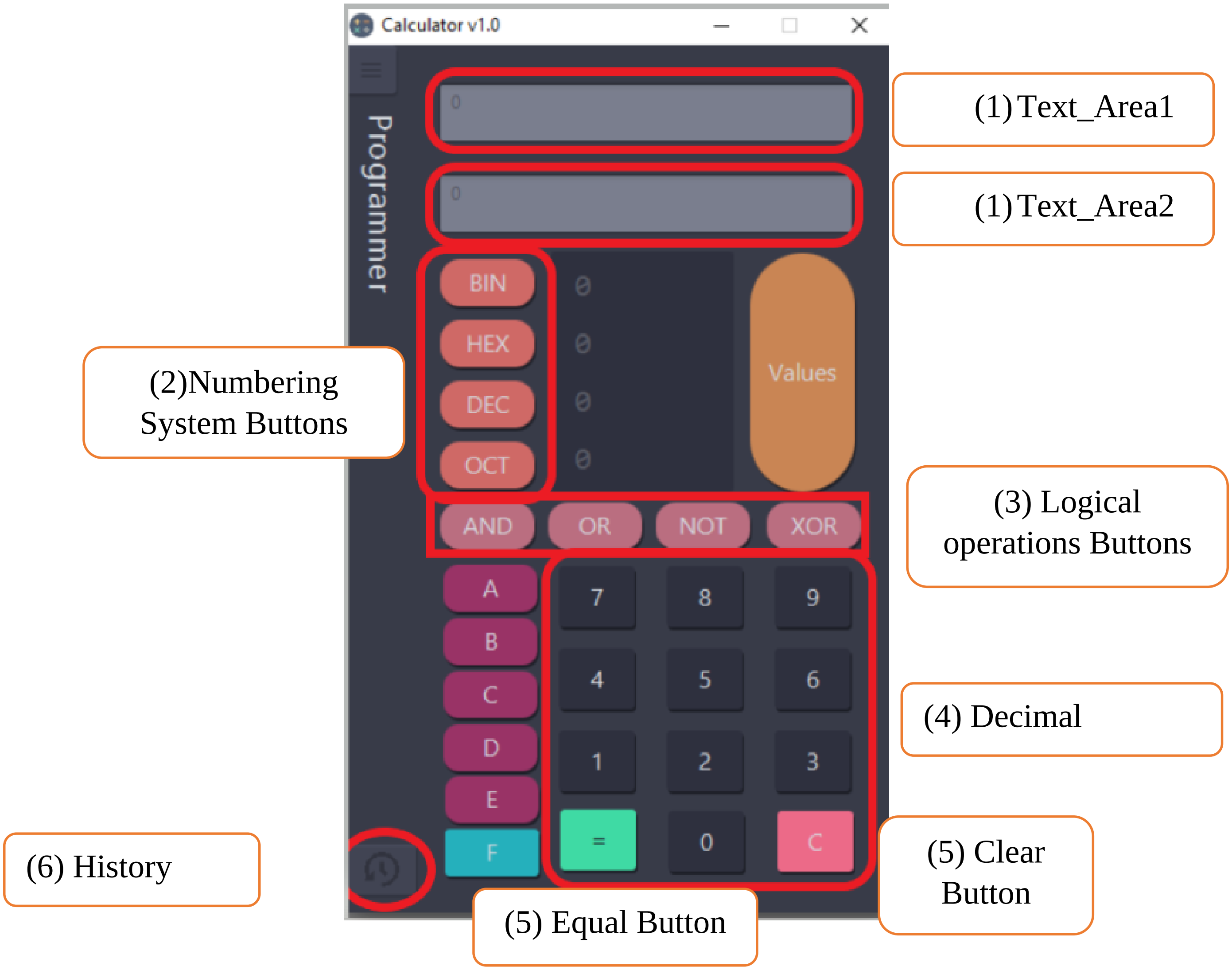
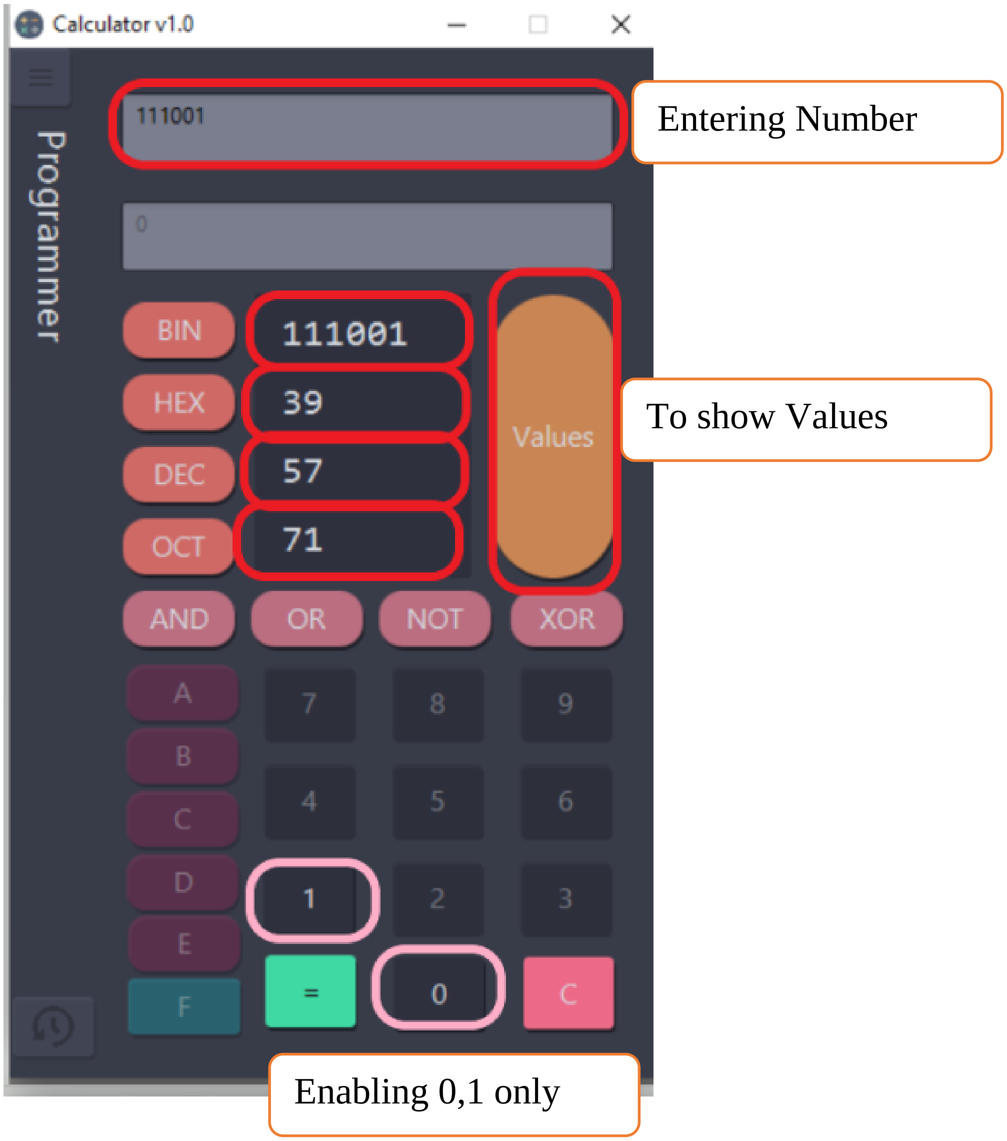


Figure 3.3.1 GUI of Programmer Mode

1) For Converting from one Numbering system to the others the user defines the System (Binary as an example), Only 0 and 1 can be enabled and then enter the number in text_Area1 the press on "Values", and the result appears at text Fields of the another Numbering systems.



111001

0

BIN 111001

HEX 39

DEC 57

OCT 71

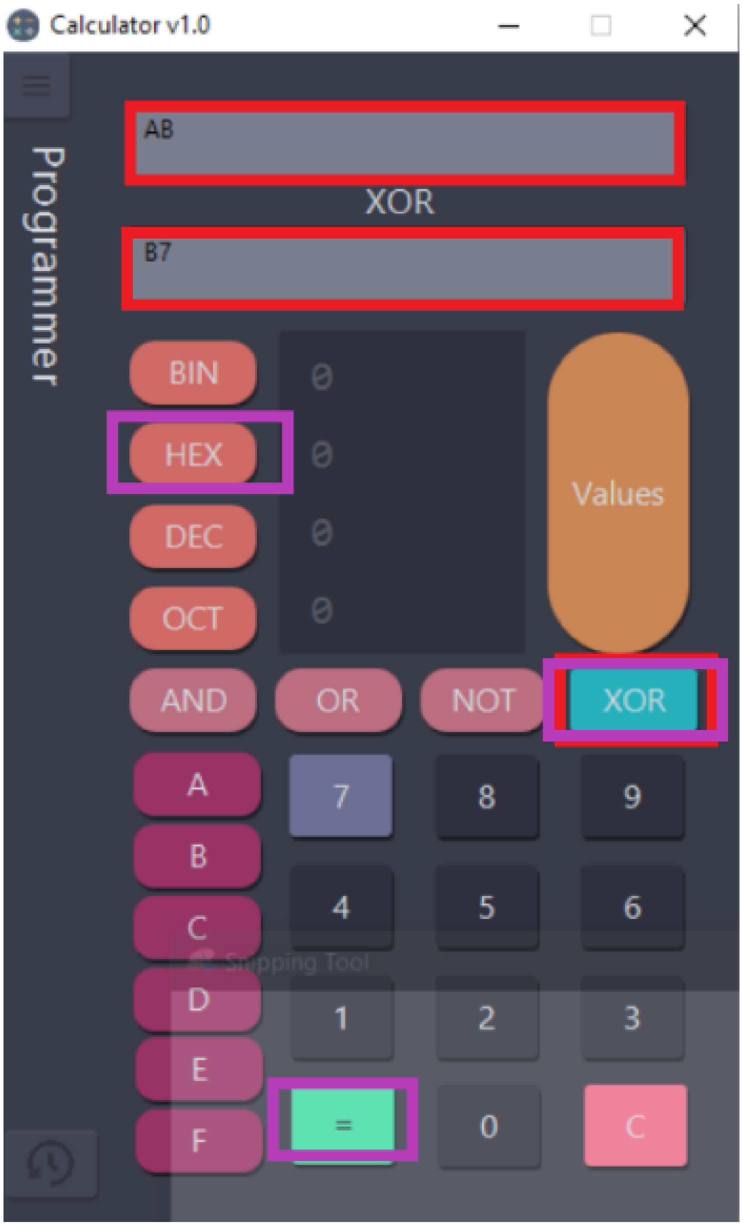
Entering Number

To show Values

Enabling 0,1 only

Figure 3.3.2 First Feature For Programmer Mode

2) For Doing a Logical operation between two Numbers the user chooses the numbering system (Hex as an Example) and enter first number at (text_Area1) and then chooses the operation from (Logical operations Buttons) (XOR as an example) and hence enter the second Number in the second text Area and press Equal Button (=).



AB

XOR

B7

Figure 3.3.3: Second Feature of Programmer Mode

Algorithm of Programmer Mode

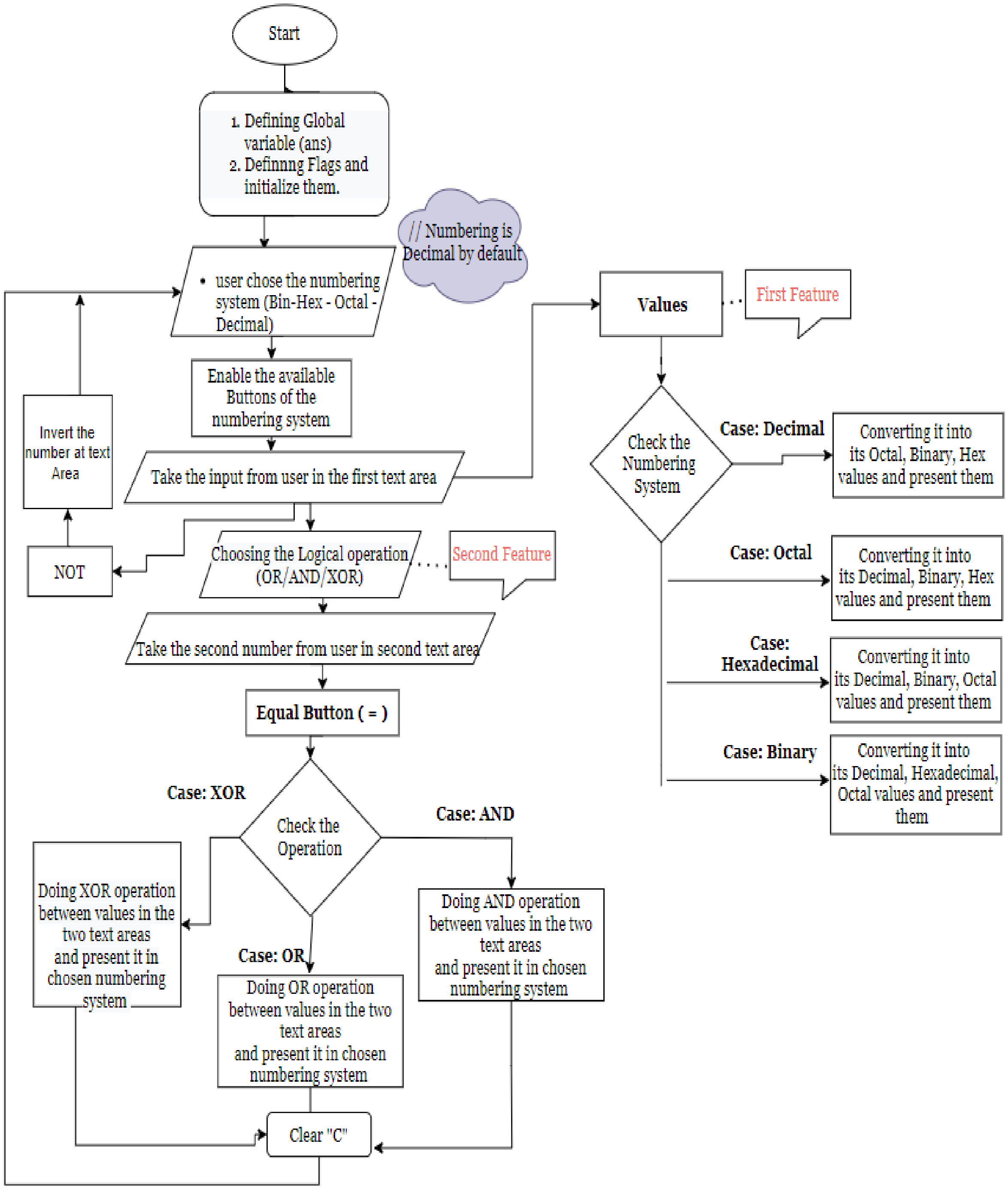

Start
1. Defining Global variable (ans)
2. Definnng Flags and initialize them.
// Numbering is Decimal by default
• user chose the numbering system (Bin-Hex - Octal - Decimal)
Enable the available Buttons of the numbering system
Take the input from user in the first text area
Invert the number at text Area
NOT
Choosing the Logical operation (OR/AND/XOR)
Second Feature
Take the second number from user in second text area
Equal Button (=)
Case: XOR
Check the Operation
Case: AND
Doing XOR operation between values in the two text areas and present it in chosen numbering system
Case: OR
Doing OR operation between values in the two text areas and present it in chosen numbering system
Doing AND operation between values in the two text areas and present it in chosen numbering system
Clear "C"
Values
First Feature
Check the Numbering System
Case: Decimal
Converting it into its Octal, Binary, Hex values and present them
Case: Octal
Converting it into its Decimal, Binary, Hex values and present them
Case: Hexadecimal
Converting it into its Decimal, Binary, Octal values and present them
Case: Binary
Converting it into its Decimal, Hexadecimal, Octal values and present them

Figure 3.3.4 Flow Chart of Programmer Mode

Date Calculation Mode

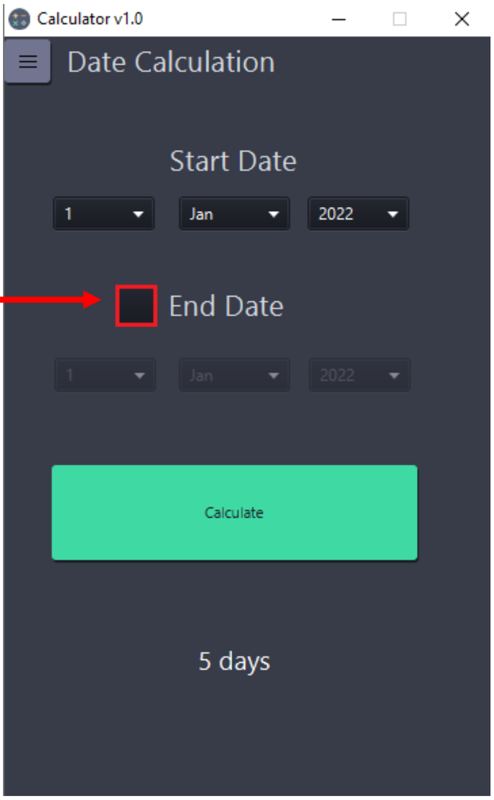
Features of this Mode

Here we offer the user the possibility to calculate the time from two modes:

- first one with taking the current date as an end date.

- second mode with taking the end date from the user.

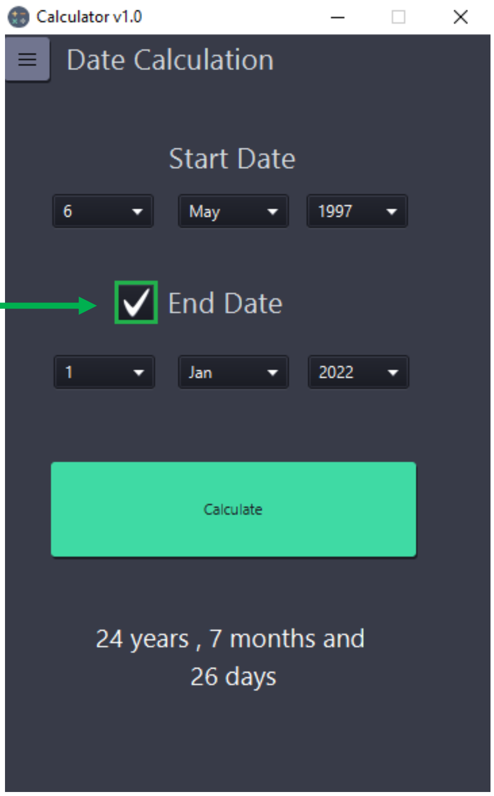
GUI OF the Scene



If checkbox **Not selected** it will get end date reference from current date

Figure 3.4.1: First Mode of Date Calculation

If checkbox **selected** it will get the end date reference from the second input.



Start Date

6 May 1997

☑ End Date

1 Jan 2022

24 years , 7 months and 26 days

Figure 3.4.2: GUI of Second mode of Date Calculation

<u>Note:</u>

- If User want to use custom end date he should make sure that the end date is larger than the Start date.

Algorithm of Date Calculation:

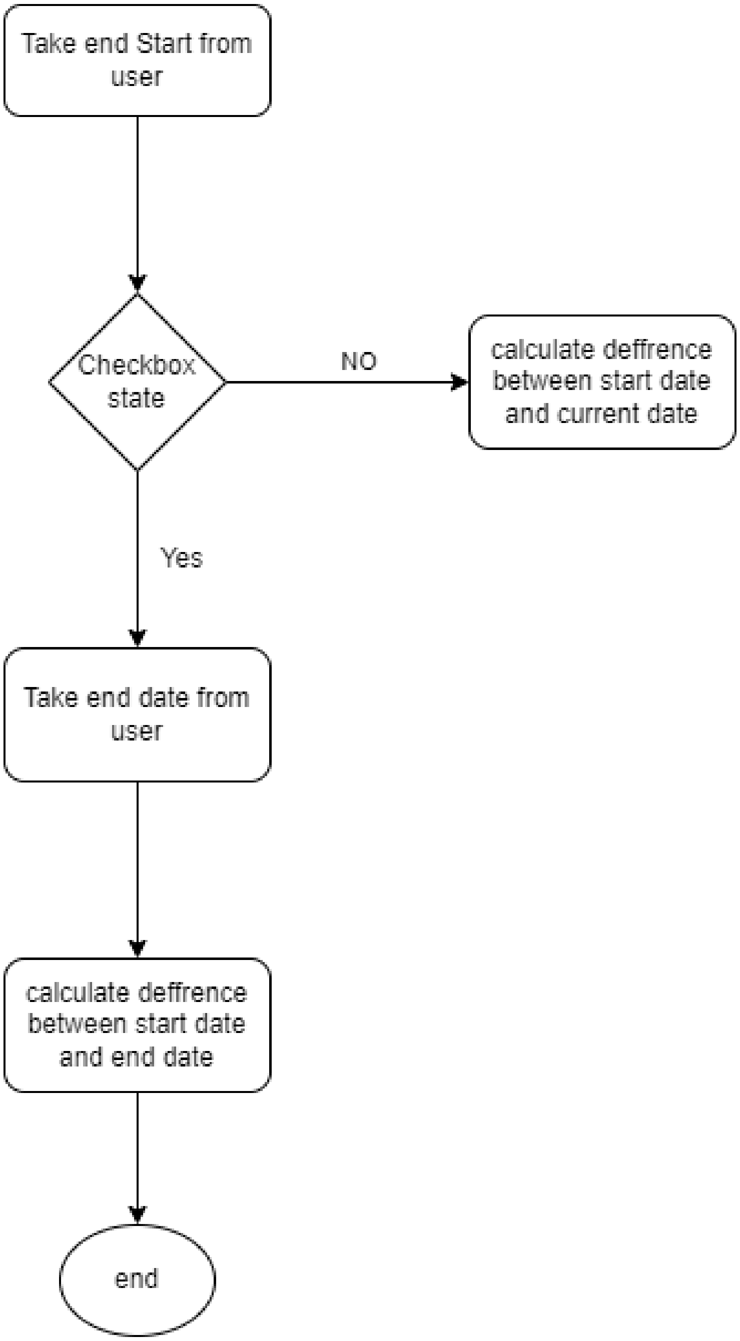

Take end Start from user
Checkbox state
NO
calculate deffrence between start date and current date
Yes
Take end date from user
calculate deffrence between start date and end date
end

Figure 3.4.3: Flow Chart of Date Calculation

Converter Mode

Features of this Mode

In this section we will talk about the Converter of the calculator, we have 2 types of converters as shown.

1)Length Converter

2)Temperature Converter

GUI OF the Scene

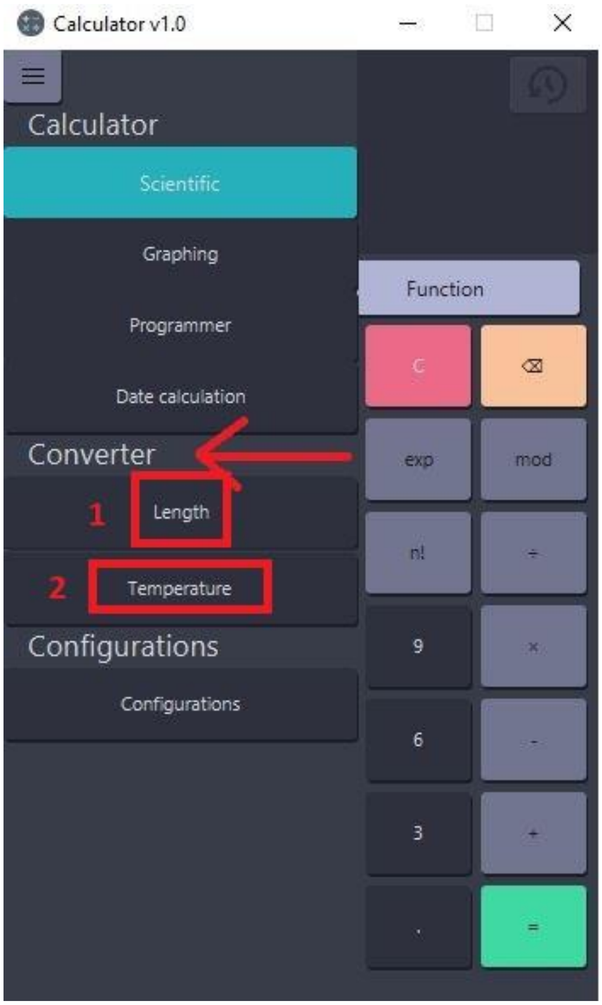


Figure 3.5.1 GUI of Converter Mode

We can access any type of them by arrows from keypad or by click on it by mouse.

1) Length

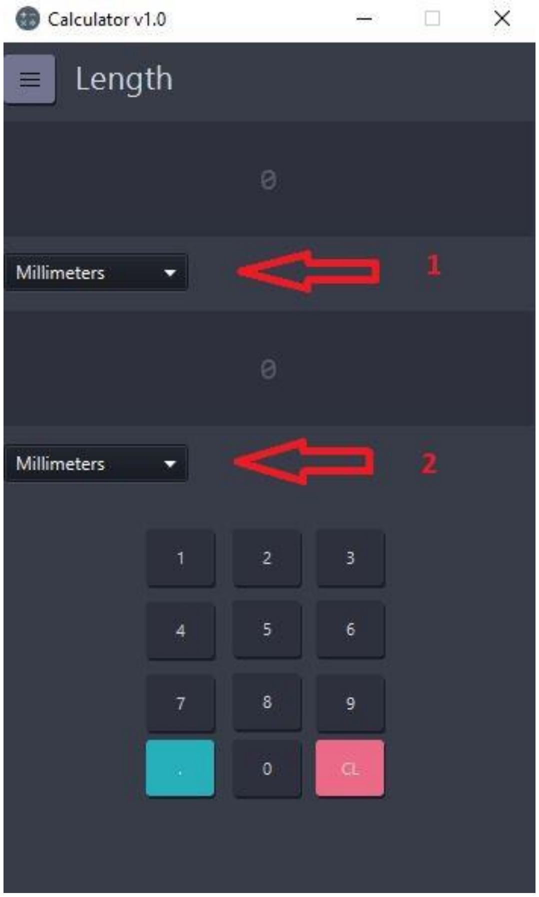


Figure 3.5.2 (Length Feature) of Converter Mode

It is a very friendly for user, in the picture we have 2 text fields, and we have 2 choice boxes that we can choose from it which unit we want to convert to which unit.

First we choose the first unit, then we choose the second unit and then type the number we need to convert and answer will appear in the text field automatic.

We can choose the unit like next picture.

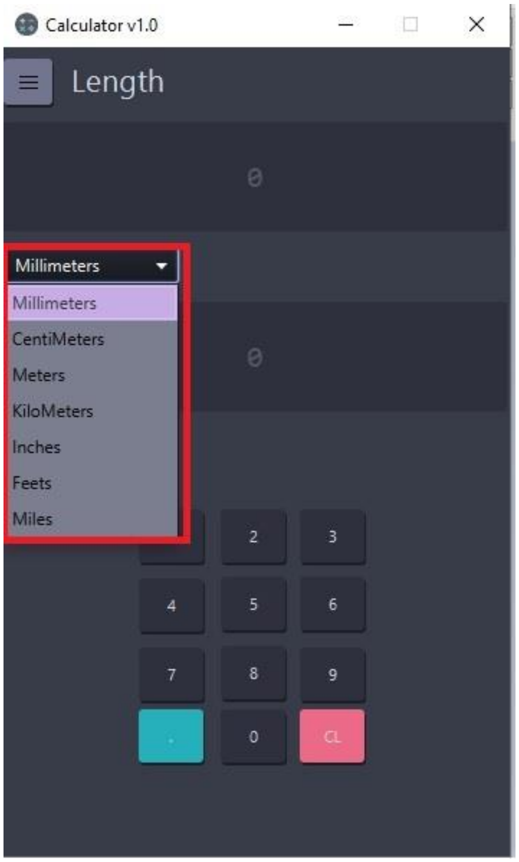


Figure 3.5.3: Converter Mode (Units of Length)

The units we have is:-

- Millimeters
- Centimeters
- Meters
- Kilometers
- Inches
- Feet
- Miles

After we choose the unit, type the number then enjoy.

2) Temperature

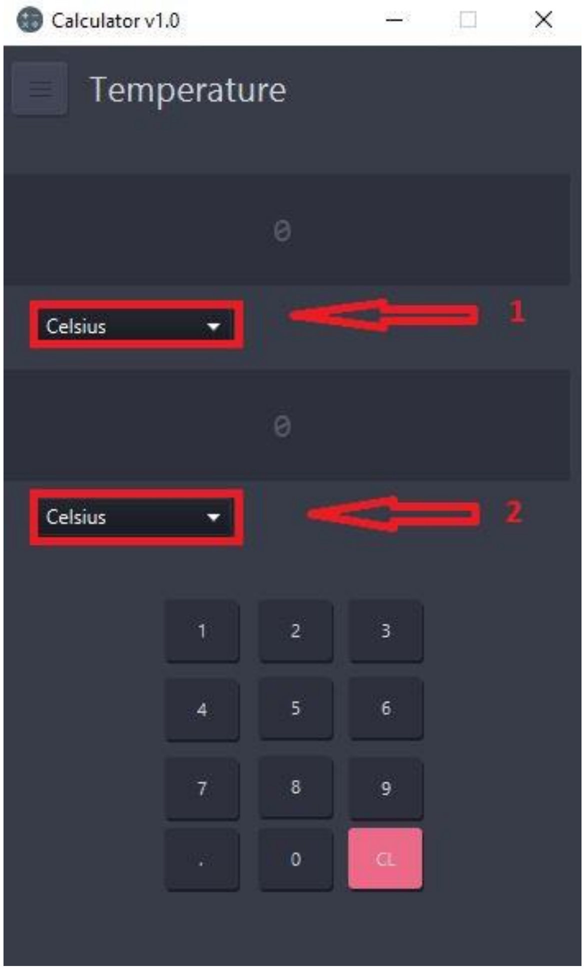


Figure 3.5.4: (Temperature) of Converter Mode

In Temperature mode, it is the same like the Length mode, we have 2 text fields and 2 choice boxes as shown.

First we choose the unit we want to convert from in choice box number 1

Second we choose the unit we want to convert to in choice box number 2

At last we type the temperature that we need to convert.

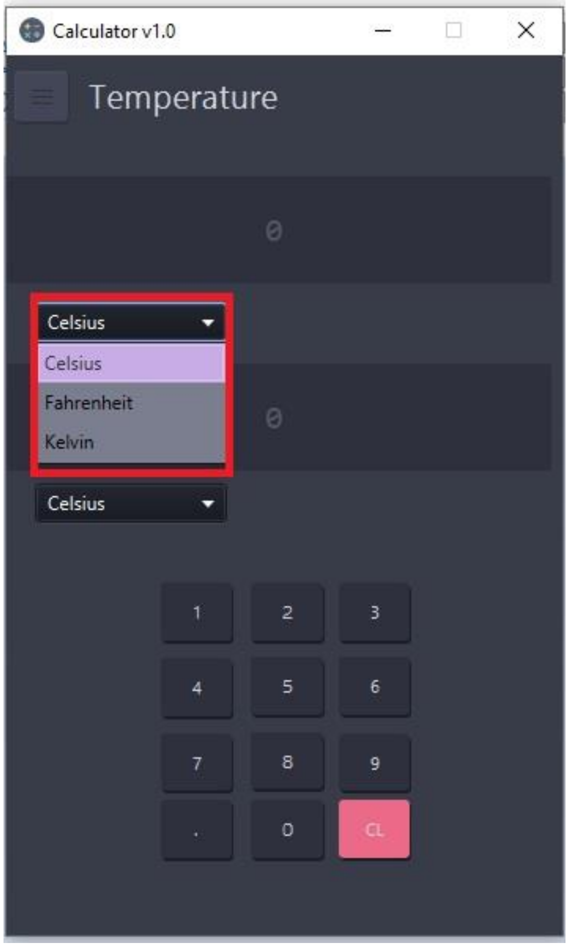


Figure 3.5.5: (Temperature - units) of Converter Mode

We have 3 units as shown:

- Celsius
- Fahrenheit
- Kelvin

We can access them by the arrows from keypad or by the mouse from laptop

Also the numbers by keypad or from the laptop

Type the temperature and see the result.

Algorithm of Converter Mode

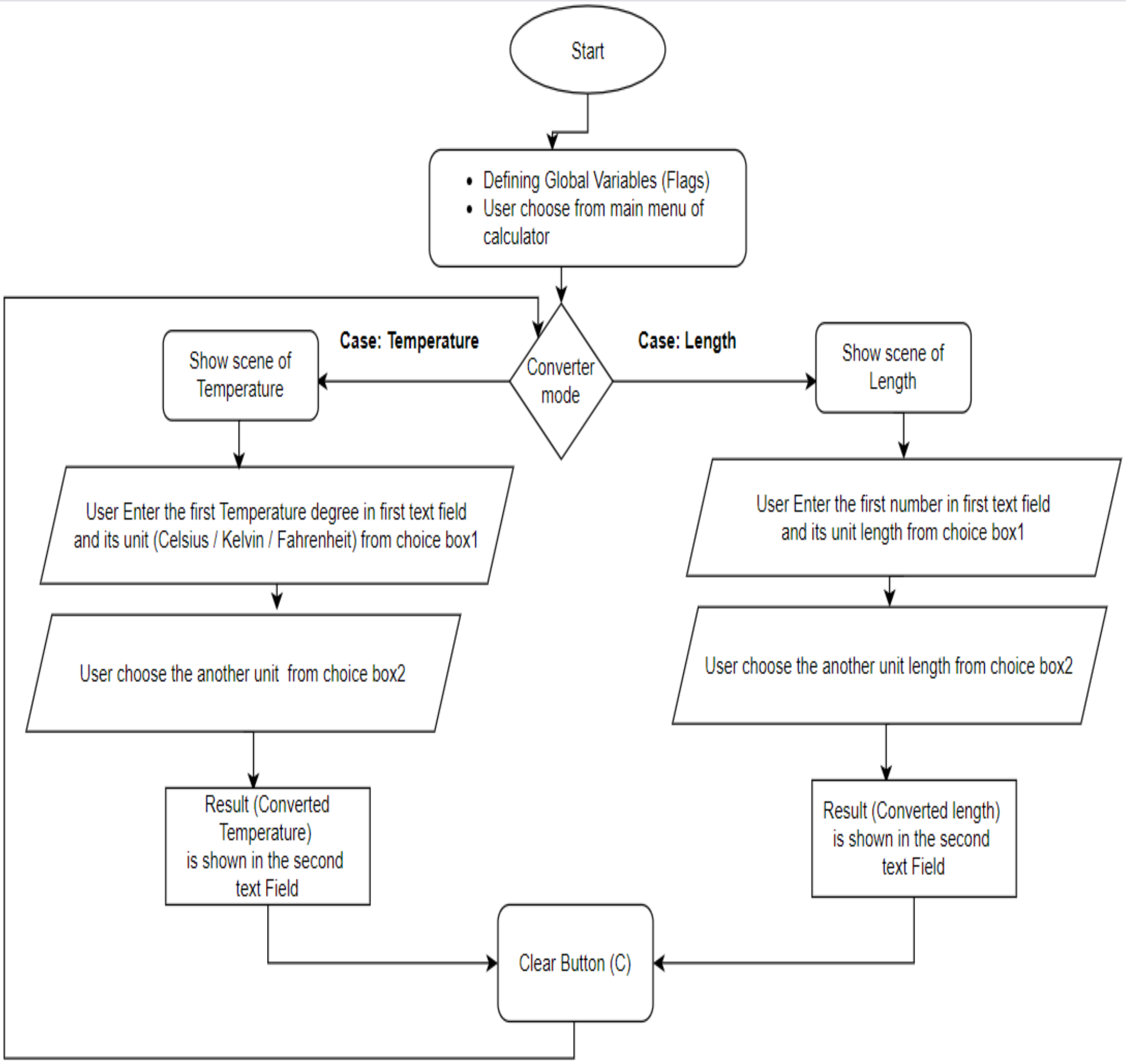

Start
Defining Global Variables (Flags)
User choose from main menu of calculator
Case: Temperature
Converter mode
Case: Length
Show scene of Temperature
Show scene of Length
User Enter the first Temperature degree in first text field and its unit (Celsius / Kelvin / Fahrenheit) from choice box1
User Enter the first number in first text field and its unit length from choice box1
User choose the another unit from choice box2
User choose the another unit length from choice box2
Result (Converted Temperature) is shown in the second text Field
Result (Converted length) is shown in the second text Field
Clear Button (C)

Figure 3.5.6: Flow Chart of Converter Mode

References

[1] URL: https://www.tutorialspoint.com/javafx/index.htm

[2] Lectures of Dr./ Eman Hesham.

[3] URL: https://www.javatpoint.com/javafx-css