INTERNSHIP REPORT
ON
GUI DEVELOPMENT PROJECT

Submitted by:

Ananya Saini

Internship period:

(6-weeks)

Submitted to:

Scientist Sanjeev Kumar

Company:

DRDO - SSPL

Date of submission:

31-07-2024

Acknowledgment

I would like to express my sincere gratitude to everyone who supported me throughout my internship and the completion of this report.

I want to start by expressing my gratitude to SSPL - DRDO for giving me the chance to intern at their prestigious company. During my service, I got important exposure and experience.

I am especially appreciative to Mr. Sanjeev Kumar, my supervisor, for their ongoing direction, encouragement, and support. Their knowledge and perceptions substantially enhanced my educational experience. His essential mentorship and support, sharing his extensive knowledge has helped me navigate through various challenges. His expertise and insights have been instrumental in the successful completion of this project.

Table of Contents

Executive Summary

This report documents my internship experience at Solid State Physics Laboratory, DRDO, where I undertook a GUI Development Project for 8 weeks. The primary objective of the project was to design and develop a user-friendly graphical user interface for an impulsive sound detection system.

The project involved several stages, including requirement gathering, design, implementation, and testing. I utilized various tools such as tkinter(python library)
and various other libraries to develop the GUI.

Throughout the internship, I worked closely with my supervisor, Mr. Sanjeev Kumar who guided me during my entire project. The initial phase focused on understanding the project requirements and what actually is a Graphical User Interface, how it works and then creating wireframes and mockups. During this phase I had several meetings with my mentor so that I could understand what actually was the problem statement and in what direction I needed to work upon.While this phase, I worked on a registration form just to get hold of the various libraries. The next phase involved creating a basic GUI and designing the look according to our needs. During the design phase, I incorporated user feedback to refine the interface. The implementation phase involved coding the GUI components and integrating them with the backend systems. Finally, the GUI was made ready for the testing phase so that it met the desired functionality and usability standards.

Despite encountering several challenges, such as integrating the GUI with existing hardwares and setting up the perfect design, I was able to overcome them with the support of my team and by employing various problem-solving techniques. These experiences have significantly enhanced my technical skills, particularly in user interface design, python programming, front-end development, as well as my ability to work in a collaborative environment.

The completion of this project has resulted in a functional and intuitive GUI that improves user experiences . This report provides a detailed account of the project phases, the challenges faced, and the solutions implemented, along with the learning outcomes and personal growth achieved during the internship.

Introduction

Background

The advancement of sound detection technology has paved the way for numerous applications in various fields such as security, health care, and entertainment. A crucial aspect of these applications is the graphical user interface (GUI) which allows users to interact efficiently with the underlying sound detection systems. During my internship at SSPL – DRDO , I was assigned to develop a GUI for an impulsive sound detection system. This system aims to provide users with a visual representation of the angle and the direction from which the sound is coming from.

Objectives

The primary objective of this project was to design and develop a robust and intuitive GUI for a sound detection system. The specific goals of the project were:

- Understand User Requirements: Conduct research and gather requirements to ensure GUI meets the users needs.
- Design the interface: Create a user-friendly interface that aligns with the user requirements.
- Develop the GUI: Implement the interface using appropriate technologies such as Python and Tkinter and its various libraries to ensure its flawless integration with the sound detection system.
- Visualise Sound Data: Develop features that accurately can display the direction of sound and its direction in real time,
- Test the Interface: Make sure that our developed GUI has no issues thus ensuring the GUI is functional and user-friendly and can be integrated with the hardware.
- Deliver a Usable Product: Ensure that the final product enhances user experience and meets all specified requirements

Company's Profile

Name: Solid State Physics Laboratory (SSPL)

Parent Organisation: Defence Research and Development Organisation (DRDO)

Location: Lucknow Road, Timarpur, Delhi - 110054, India

History:

Established in 1962, the Solid State Physics Laboratory (SSPL) is a premier research institution under the Defence Research and Development Organisation (DRDO), which is part of the Ministry of Defence, Government of India. SSPL was founded with the mission of advancing research and development in the field of solid-state physics and materials science, with a focus on applications that enhance national security and defense capabilities.

Mission and Vision

Mission: To conduct cutting-edge research in solid-state physics and related fields to develop advanced materials and devices for strategic and defense applications.

Vision: To be a leader in the development of innovative technologies that bolster national security and contribute to the scientific and technological advancement of India.

Key Achievements:

SSPL has made significant contributions to India's defense technology, including:-

• Development of critical components for missile systems, radars, and communication systems.

• Advancements in semiconductor and magnetic materials for various defense applications and many more.

Contact Information: 011 2381 8041

Address:

Solid State Physics Laboratory (SSPL)

Defence Research and Development Organisation (DRDO)

Lucknow Road, Timarpur

Delhi - 110054

India

Project Description

Project Overview

The project involved the design and development of a Graphical User Interface (GUI) for an impulsive sound detection system at the Solid State Physics Laboratory (SSPL), Delhi. The sound detection system aims to analyze and display sound data in real-time,providing users the direction and the angle from which the sound is coming from. The GUI is intended to enhance user interaction, making it easier to understand and manage sound data effectively for various applications in defense and security.

Technologies Used

The development of the GUI utilized a range of technologies and tools, including:

• Programming Language: Python

• GUI Framework: Tkinter

• Tkinter Libraries: PyAudio, Pyserial, math, random, serial , threading, scrolled texts, serial.tools.list_ports

• Development Tools: Pycharm

Key Features

The GUI for the impulsive sound detection system includes several key features:

• Real-Time Sound Visualisation: Displays the direction and the angle of an incoming sound, providing users with immediate feedback.

• User-Friendly Interface: Intuitive design that simplifies interaction and ensures accessibility for users with varying levels of technical expertise.

Detailed Work Experience

Project Overview

During the internship at the Solid State Physics Laboratory (SSPL), DRDO, I worked on a GUI Development Project focusing on an impulsive sound detection system. The project spanned for 8 weeks and involved designing and implementing a user-friendly graphical interface to interact with and visualize real-time sound data.

Tasks and Responsibilities

Required Analysis and Research

Activities: Engaged in meetings with the mentor and the team to understand the project requirements and objectives. Conducted research on existing sound detection systems and GUI best practices. Gathered and analyzed requirements to ensure the GUI met the necessary functionality and user experience standards.

Challenges: Accurately interpreting the system requirements and translating them into functional GUI elements.

Solutions: Referenced online resources to understand common design patterns and user needs.

GUI Design and Prototyping

Activities: Created wireframes and mockups for the GUI layout using design tools and iterative feedback from the mentor. Designed key components of the GUI, including a directional clock for visualizing incoming sound angles.

Challenges: Ensuring the GUI was both visually appealing and functional. Solutions: Thought of various designs and incorporated feedback from team members to refine the design.

Development and Implementation:

Activities: Developed the GUI using Python and Tkinter, focusing on creating an intuitive interface for real-time sound visualization. Implemented real-time data handling from the serial port using the serial library and integrated it with the GUI.

Challenges: Managing real-time updates and ensuring the GUI remained responsive during data processing.

Solutions: Implemented threading to handle serial data in the background, preventing the GUI from freezing.

Documentation and Reporting:

Activities: Documented the development process, including design decisions, challenges faced, and solutions implemented. Prepared and submitted a comprehensive internship report detailing the project, including the GUI design, code, and performance metrics.

Key Learning Outcomes:-

- Proficiency in GUI Development: Gained practical experience in designing and implementing graphical user interfaces using Python and Tkinter.
- Real-Time Data Handling: Developed skills in processing and visualizing real-time data within a GUI environment.
- Multithreading and Asynchronous Programming: Learned to use threading for background tasks to keep the GUI responsive.
- Design and Layout Optimization: Improved abilities in designing user friendly interfaces and managing layout for different screen sizes.
- Problem-Solving: Enhanced problem-solving skills by addressing and resolving development challenges.

Summary:-

The internship provided an invaluable experience in GUI development, offering insights into designing and implementing user interfaces for real time systems. The project not only strengthened my technical skills but also allowed me to apply problem-solving techniques and best practices in a practical setting. The successful completion of the GUI Development Project demonstrated my ability to handle complex technical tasks and contribute effectively to a development team.

SOURCE CODE

HOME PAGE

```python
from tkinter import*
import math
import tkinter as tk
import random
import serial
import threading
import serial.tools.list_ports as p


window=tk.Tk()
window.geometry("900x550")
window.title("Sound Detector")

def initialize_ports():
    global ser
    ports = p.comports()
    available_ports = [port.device for port in ports]
    if not available_ports:
        raise Exception("No available ports found. Please check your device
connection.")
    else:
        print(f"Available ports: {available_ports}")
    ser = serial.Serial(available_ports[0], 9600)

initialize_ports()
var_r1=StringVar()

azimuthal = StringVar()
elevationvar = StringVar()


def ChangeDirection():
    global ser
    try:

        line = ser.readline().decode('utf-8').strip()
        angle = int(line)

        azimuthal.set(f"{angle}°")
```

```python
            elevationvar.set(f"{angle}°")
            clock.point_to_number(angle / 30)


    except Exception as e:
        print(f"Error reading data from serial port: {e}")


def read_serial():
    global ser
    while True:
        if ser.in_waiting > 0:
            ChangeDirection()




serial_thread = threading.Thread(target=read_serial, daemon=True)
serial_thread.start()




ChangeDirection()

class DirectionalClock(tk.Canvas):
    def __init__(self, master=None, **kwargs):
        super().__init__(master, **kwargs)
        self.radius = min(self.winfo_reqwidth(), self.winfo_reqheight()) // 2 - 10
        self.center = self.radius + 10
        self.create_clock_face()


    def create_clock_face(self):
        self.create_oval(10, 10, 2 * self.radius + 10, 2 * self.radius + 10)
        for number in range(1, 13):
            angle = math.pi / 6 * (number - 3)
            x = self.center + self.radius * 0.8 * math.cos(angle)
            y = self.center + self.radius * 0.8 * math.sin(angle)
            self.create_text(x, y, text=str(number*30), font=("Arial", 10))


    def point_to_number(self, number):
        self.delete('hand')
        angle = math.pi / 6 * (number - 3)
        x = self.center + self.radius * 0.7 * math.cos(angle)
        y = self.center + self.radius * 0.7 * math.sin(angle)
        self.create_line(self.center, self.center, x, y, width=2, fill='blue',
tags='hand')
```

```python
def on_button_click():
    try:
        number = int(entry.get())
        if 1 <= number <= 360:
            clock.point_to_number(number/30)
        else:
            print("Please enter an angle")
            if number==0:
                clock.point_to_number(360)
    except ValueError:
        print("Please enter a valid number.")


clock = DirectionalClock(window, width=275, height=275)
clock.place(x=300,y=240)


entry = tk.Entry(window)
entry.place(x=690,y=350)


button = tk.Button(window, text="CONFIRM", command=on_button_click)
button.place(x=720,y=380)


label4=Label(window,text="Enter an
angle",width=20,font=("arial",10,"bold")).place(x=665,y=320)



menu=Menu(window)
window.config(menu=menu)
subm1 =Menu(menu)
menu.add_cascade(label="Home")


sub2=Menu(menu)
menu.add_cascade(label="Configuration")



label1=Label(window,text="IMPULSIVE SOUND DETECTION
SYSTEM",fg='white',bg='blue',relief='solid',width=36,font=("arial",26,"bold")).place(x=
125,y=65)
label2=Label(window,text="Azimuthal
Angle",width=20,font=("arial",17,"bold")).place(x=15,y=180)


entry_1=Entry(window,textvariable=azimuthal).place(x=280,y=186)


label3=Label(window,text="Elevation
Angle",width=25,font=("arial",17,"bold")).place(x=400,y=180)
entry_1=Entry(window,textvariable=elevationvar).place(x=700,y=186)
```

```
window.mainloop()
```

CONFIGURATION PAGE

```
import serial
import threading
from tkinter import *
from tkinter import scrolledtext
import serial.tools.list_ports as p

def ports():
    ports = p.comports()
    port_list = [i.device for i in ports]
    return port_list

def connect():
    global ser
    port = pn.get()
    baud = cn.get()
    try:
        ser = serial.Serial(port, baud)
        monitor_text.insert(END, f"Connected to {port} at {baud} baud.\n")
        reading()
    except Exception as e:
        monitor_text.insert(END, f"Failed to connect: {e}\n")

def read_from_port():
    while ser.is_open:
        try:
            value = ser.readline()
            valueIntString = value.decode('UTF-8')
            monitor_text.insert(END, f"Received: {valueIntString}\n")
        except Exception as e:
            monitor_text.insert(END, f"Error: {e}\n")
            break

def refresh():
    if ser.is_open:
        ser.close()
        monitor_text.insert(END, "Disconnected.\n")
    connect()
```

```python
def reading():
    thread = threading.Thread(target=read_from_port)
    thread.daemon = True
    thread.start()


root = Tk()
root.geometry("900x550")
root.title("COMPORT SETTING")


cn = StringVar()
pn = StringVar()


Heading = Label(root, text="COMPORT SETTINGS", fg='White', bg='Blue', width=32,
font=("arial", 28, "bold")).place(x=100, y=20)


port_label = Label(root, text="Ports:", width=20, font=("arial", 10,
"bold")).place(x=100, y=100)
ports_available = ports()
droplist1 = OptionMenu(root, pn, *ports_available)
pn.set("SELECT THE COMPORT")
droplist1.config(width=20, font=("arial", 8), bg='white')
droplist1.place(x=280, y=100)


baud_label = Label(root, text="Baud Rate:", width=20, font=("arial", 10,
"bold")).place(x=110, y=160)
baud_rates = ['4800', '9600', '19200', '38400', '57600', '115200', '230400', '460800',
'921600']
droplist2 = OptionMenu(root, cn, *baud_rates)
cn.set("SELECT THE BAUD RATE")
droplist2.config(width=20, font=("arial", 8), bg='white')
droplist2.place(x=280, y=160)


button = Button(root, text="Connect",
command=connect,font=("arial",8,"bold"),fg="blue",bg="white").place(x=350, y=250)
refresh_button = Button(root, text="Refresh",
command=refresh,font=("arial",8,"bold"),fg="blue",bg="white").place(x=500, y=250)


monitor_text = scrolledtext.ScrolledText(root, width=90, height=15)
monitor_text.place(x=85, y=290)


root.mainloop()
```
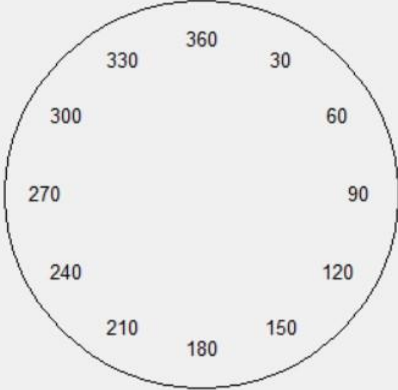
COMPORT SETTINGS

Ports:　　SELECT THE COMPORT

Baud Rate:　　SELECT THE BAUD RATE

18

Connect　　Refresh

COMPORT SETTINGS

Ports:　　COM4

Baud Rate:　　19200
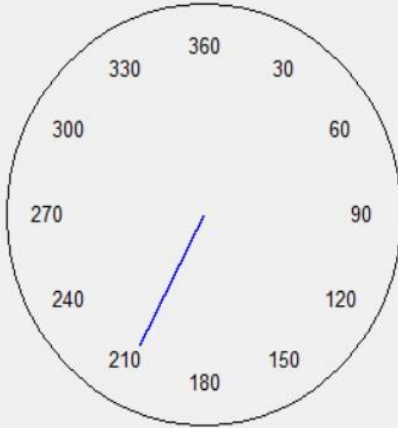
Connect　　Refresh

Connected to COM4 at 19200 baud.

REFERENCES

https://www.geeksforgeeks.org/what-is-graphical-user-interface/

https://www.geeksforgeeks.org/python-gui-tkinter/

https://youtu.be/O_AFXmkwpK0?si=MkNQ9CKp2ZS-zyue

https://youtu.be/-GhzpvvIXlM?si=9eBGyy0xDNuYyWHP

https://youtu.be/OXG6fauVVpY?si=55OlSzv7FO37awvQ

https://youtu.be/ls3BAhPV06M?si=NuMi9AQDKQ4bk6li

https://youtu.be/rpgqdR-KHhE?si=ozNBqTqnF-cYGi-n

https://youtu.be/OK_ntOcU28g?si=EBus4Wm7C7g3YROK

https://youtu.be/n2FKsPt83_A?si=dhSe1bA1WnBXkjoL

https://youtu.be/jngL0pSzh44?si=uYb1BF_bVlokKypf

https://youtu.be/jngL0pSzh44?si=Avu08rFkW5aFLksZ

https://youtu.be/qw_XHRJP-vc?si=oYs31psVn5C5nCGB

https://youtu.be/5g23mXx9r7M?si=H68Xe2xe5EEko0iS

https://youtu.be/89a-onbToN8?si=LSlQSjoZCKSGj9Fh