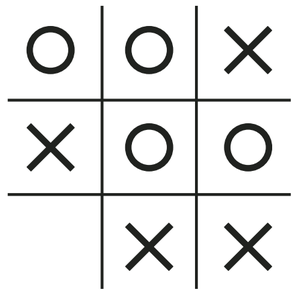
PRESENTATION

IMPLEMENTATION OF TIC TAC TOE USING TKINTER



COURSE TITLE - INTRO TO PYTHON PROGRAMMING (CS213)

NAME- SUBHAV KUMAR
ROLL- CS23B1071
DEPARTMENT- CSE

PROJECT UNDER - DUBACHARLA GYANESHWAR SIR

SESSION 2024-25

Slides By - Subhav Kumar

Introduction

Developed a classic Tic-Tac-Toe game with a graphical user interface (GUI) using Python's Tkinter library.

Key Features

01 Responsive Interface

02 Sound Effects

03 Visual Celebrations

04 Error Handling

Slides By - Subhav Kumar

Aims and Objectives

AIM

To create an engaging and user-friendly Tic-Tac-Toe game that leverages Python's Tkinter for the GUI and Pygame for audio functionalities.

OBJECTIVES

01

GUI Development

Design a responsive and intuitive interface for the Tic-Tac-Toe game.

02

User Experience

Provide visual feedback (e.g., highlighting winning combinations) and error messages for invalid moves.

03

Sound Integration

Implement sound effects for user interactions, wins, and ties using Pygame.

04

Reset Functionality

Enable users to reset the game seamlessly through a menu option.

05

Game Logic

Develop robust game logic to handle player turns, win conditions, and tie scenarios.

Slides By - Subhav Kumar

Role in the Project:

- **Led the development of the game's GUI and integrated audio features.**
- **Implemented the core game logic and ensured smooth user interactions.**

Key Features Developed:

01

Interactive Buttons

Created clickable buttons for each Tic-Tac-Toe cell using Tkinter's Button widget.

02

Sound Effects

Integrated click, win, and tie sounds using Pygame's mixer module to enhance feedback.

03

Win Celebration

Designed a visual celebration using Tkinter's Canvas to display animated ovals upon a win.

04

Menu Options

Developed a reset functionality accessible via the game's menu for restarting the game.

Slides By - Subhav Kumar

Tools And Data Used

Programming Language- Python 

Libraries and Frameworks- i) Tkinter

ii) Pygame

Sound Files : Custom or sourced sound effects for various game events was taken from Freesound.

IDE: Visual Studio Code

Slides By - Subhav Kumar

Results

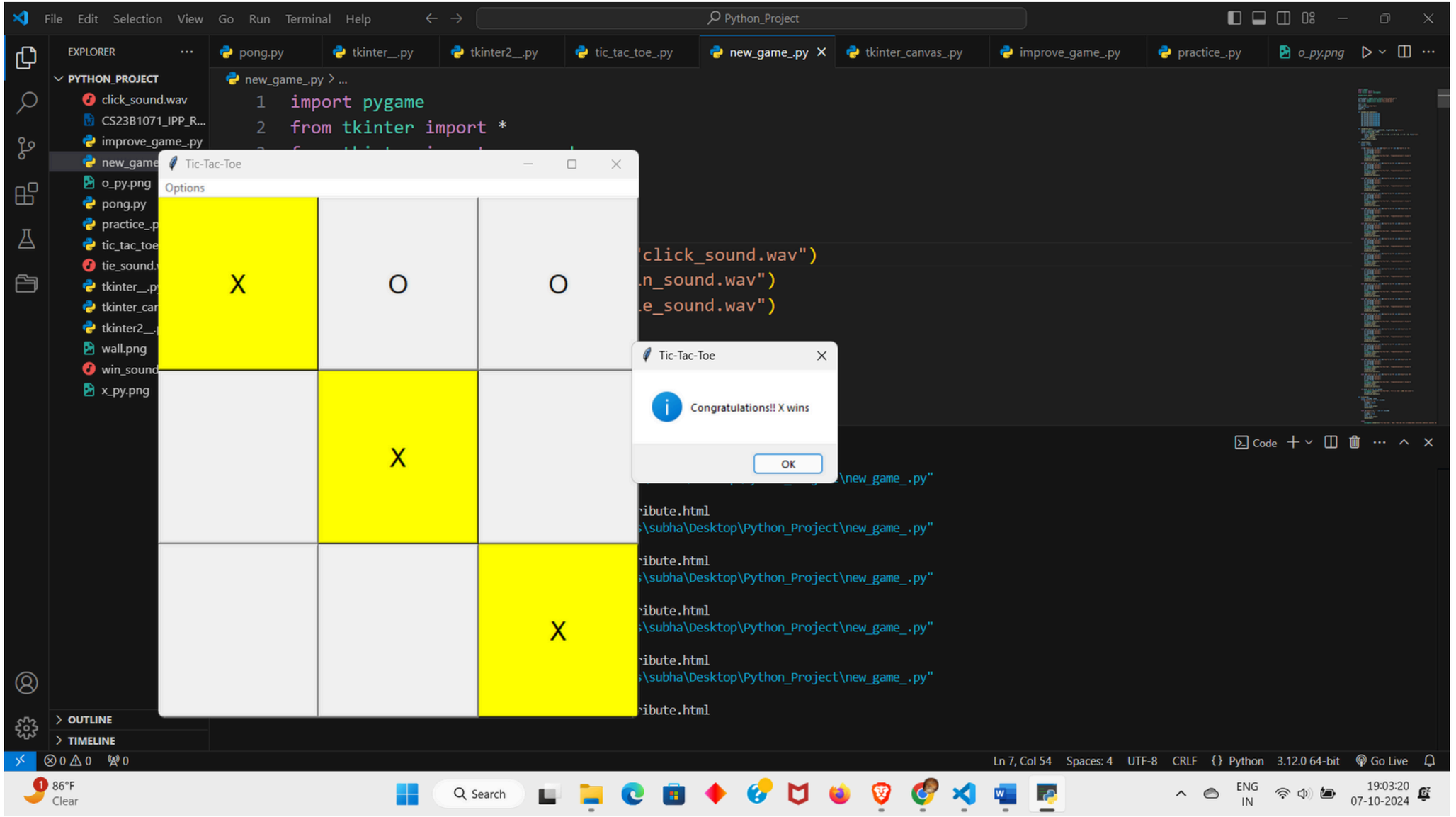
Results:

- Successfully developed a fully functional Tic-Tac-Toe game with a user-friendly interface.
- Integrated sound effects that respond to user actions, enhancing interactivity.
- Implemented visual feedback for winning conditions and tie scenarios.
- Achieved seamless game resets through an intuitive menu option.

In furthur pages results are shown pictorially.

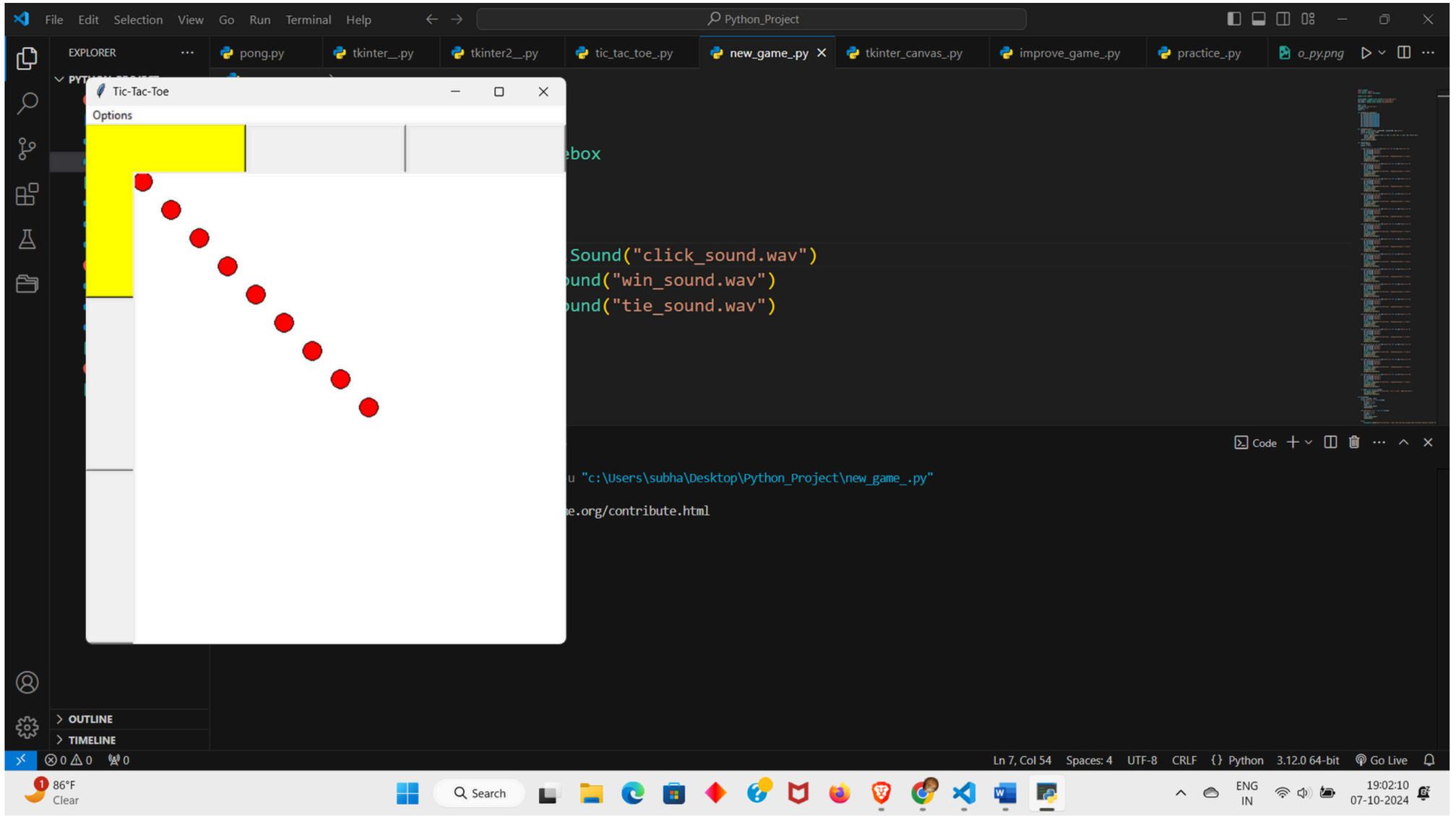
Slides By - Subhav Kumar

Results



Slides By - Subhav Kumar

Results



Slides By - Subhav Kumar

Some Code Snippets

```python
import pygame
from tkinter import *
from tkinter import messagebox

# Initialize pygame mixer
pygame.mixer.init()

# Load sounds
click_sound = pygame.mixer.Sound("click_sound.wav")
win_sound = pygame.mixer.Sound("win_sound.wav")
tie_sound = pygame.mixer.Sound("tie_sound.wav")

root = Tk()
root.title('Tic-Tac-Toe')
clicked = True
count = 0

def disable_all_buttons():
    b1.config(state=DISABLED)
    b2.config(state=DISABLED)
    b3.config(state=DISABLED)
    b4.config(state=DISABLED)
    b5.config(state=DISABLED)
    b6.config(state=DISABLED)
    b7.config(state=DISABLED)
    b8.config(state=DISABLED)
```

Slides By - Subhav Kumar

Some Code Snippets

```
    b9 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma

    b1.grid(row=0, column=0)
    b2.grid(row=0, column=1)
    b3.grid(row=0, column=2)

    b4.grid(row=1, column=0)
    b5.grid(row=1, column=1)
    b6.grid(row=1, column=2)

    b7.grid(row=2, column=0)
    b8.grid(row=2, column=1)
    b9.grid(row=2, column=2)

# Create Menu
my_menu = Menu(root)
root.config(menu=my_menu)

# Create options menu
options_menu = Menu(my_menu, tearoff=False)
my_menu.add_cascade(label="Options", menu=options_menu)
options_menu.add_command(label="Reset Game", command=reset)

reset()
root.mainloop()
```

Ln 14, Col 26 Spaces: 4 UTF-8 CRLF {} Python 3.12.0 64-bit Go Live

Slides By - Subhav Kumar

Some Code Snippets

```
207  def b_click(b):
219          count += 1
220          click_sound.play()
221          checkifwon()
222
223      else:
224          messagebox.showerror("Tic-Tac-Toe", "Hey! That box has already been selected.\nSelect another B
225
226  def reset():
227      global b1, b2, b3, b4, b5, b6, b7, b8, b9
228      global clicked, count
229      clicked = True
230      count = 0
231
232      b1 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
233      b2 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
234      b3 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
235
236      b4 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
237      b5 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
238      b6 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
239
240      b7 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
241      b8 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
242      b9 = Button(root, text=" ", font=("Helvetica", 20), height=3, width=6, bg="SystemButtonFace", comma
243
```

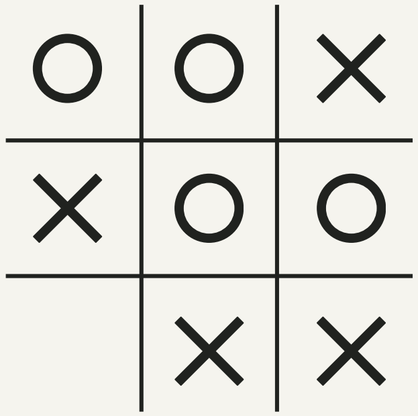
Ln 14, Col 26 Spaces: 4 UTF-8 CRLF {} Python 3.12.0 64-bit Go Live

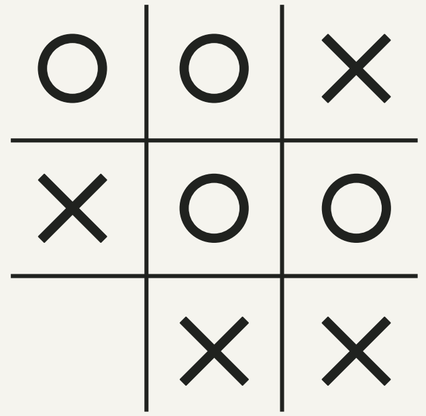
Slides By - Subhav Kumar

Conclusion

- The project effectively demonstrates the capabilities of Python's Tkinter and Pygame libraries in creating interactive applications.
- Received positive feedback on usability and feature integration.

Slides By - Subhav Kumar





Thank you

Slides By - Subhav Kumar