Documentation

Introduction

Ian's Screen Recorder is a simple graphical user interface (GUI) application built using the Tkinter library in Python. It utilizes the `pyscreenrec` module for screen recording functionalities. This documentation provides an overview of the code structure and explains the purpose and functionality of each component.

Installation

Ensure that you have Python installed on your system. You can install the required modules using the following command:

```
pip install tk pyscreenrec
```

Code Structure

1. Importing Libraries

```
from tkinter import *
import pyscreenrec
```

- The `Tkinter` library is used for building the GUI.
- `pyscreenrec` is a module for screen recording.

2. Creating the Root Window

```
root = Tk()
root.geometry("400x600")
root.title("screen recorder")
```

```
root.config(bg="#fff")
root.resizable(False, False)
```

- `Tk()` creates the main window for the application.
- `geometry()` sets the initial dimensions of the window.
- `title()` sets the title of the window.
- `config()` sets the background color.
- `resizable()` restricts resizing of the window.

3. Functions for Recording Control

```
def start_rec():
    # Function to start recording
    file = Filename.get()
    rec.start_recording(str(file + ".mp4"), 5)

def pause_rec():
    # Function to pause recording
    rec.pause_recording()

def resume_rec():
    # Function to resume recording
    rec.resume_recording()

def stop_rec():
    # Function to stop recording
    rec.stop_recording()
```

- These functions control the screen recording process based on user interactions.

4. Initializing Screen Recorder

```
rec = pyscreenrec.ScreenRecorder()
```

- Creates an instance of the `ScreenRecorder` class from the `pyscreenrec` module.

5. Setting Window Icon

```
image_icon = PhotoImage(file="icon.png")
root.iconphoto(False, image_icon)
```

- Sets the window icon using an image file.

6. Adding Background Images

```
image1 = PhotoImage(file="yelllow.png")
Label(root, image=image1, bg="#fff").place(x=-2, y=35)

image2 = PhotoImage(file="blue.png")
Label(root, image=image2, bg="#fff").place(x=223, y=200)
```

- Places background images on the window.

7. Adding Heading

```
lbl = Label(root, text="Ian's Screen Recorder", bg="#fff", fo
nt="arial 15 bold")
lbl.pack(pady=20)
```

- Displays a heading label on the window.

8. Adding Recording Image

```
image3 = PhotoImage(file="recording.png")
Label(root, image=image3, bd=0).pack(pady=30)
```

- Displays an image indicating recording status.

9. Adding Entry for Filename

```
Filename = StringVar()
entry = Entry(root, textvariable=Filename, width=18, font="ar
ial 15")
entry.place(x=100, y=350)
Filename.set("recording25")
```

- Provides an entry widget for the user to input the filename.

10. Adding Control Buttons

```
start = Button(root, text="Start", font="arial 22", bd=0, com
mand=start_rec)
start.place(x=140, y=250)

pause = Button(root, image=image4, bd=0, bg="#fff", command=p
ause_rec)
pause.place(x=50, y=450)

resume = Button(root, image=image5, bd=0, bg="#fff", command=
stop_rec)
resume.place(x=150, y=450)

stop = Button(root, image=image6, bd=0, bg="#fff")
stop.place(x=250, y=450)
```

- Creates buttons for starting, pausing, resuming, and stopping the recording.

11. Main Event Loop

```
root.mainloop()
```

- Initiates the main event loop, allowing the GUI to interact with the user.

Conclusion

Ian's Screen Recorder provides a simple interface for screen recording, allowing users to control the recording process using the provided buttons. Customize the file paths and image filenames as needed for your application.