🎬 Clean Screen Recorder

A simple, focused screen recording application with essential features only.

✨ Features

Essential Recording:

- 🎥 High-quality screen capture
- 🎵 Audio recording (microphone)
- ⚙️ Configurable frame rate and quality
- 📁 Easy file output selection

Clean Interface:

- 🎯 Minimal, distraction-free UI
- 🚀 One-click recording start/stop
- 📊 Real-time status updates
- 💾 Smart file naming
- **High-quality MP4 output** using OpenCV H.264 encoding

Audio Recording

- **Microphone recording** with device selection
- **System audio capture** (Windows WASAPI loopback)
- **Automatic audio/video synchronization** via FFmpeg
- **Configurable sample rates and channels**

Video Overlays

- **Mouse cursor highlighting** with customizable color, size, and opacity
- **Webcam picture-in-picture** with position and size controls
- **Real-time overlay processing** with minimal performance impact

Performance Optimizations

- **Modular architecture** for better maintainability and performance
- **Efficient threading** for smooth capture without frame drops
- **Memory management** optimized for extended recording sessions
- **Resource cleanup** and error handling for stability

Project Structure

New Modular Architecture (Recommended)

```
GUIScreenRecorder/
├── app_new.py            # New modular entry point ⭐
```

```
├── src/                  # Modular source code
│   ├── __init__.py
│   ├── core/            # Core recording functionality
│   │   ├── __init__.py
│   │   ├── config.py    # Configuration management
│   │   ├── audio.py     # Audio recording engine
│   │   └── video.py     # Video recording engine
│   ├── ui/              # User interface
│   │   ├── __init__.py
│   │   └── main_window.py  # Main application window
│   └── utils/           # Utility functions
│       ├── __init__.py
│       └── helpers.py   # Helper functions and FFmpeg tools
```

Legacy Version (Still Available)

```
├── app.py                # Legacy monolithic version (working)
├── install_ffmpeg.py     # FFmpeg installation utility
```

Installation

Requirements

- Python 3.12 or higher
- Windows, macOS, or Linux

Core Dependencies

```
pip install opencv-python mss numpy
```

Optional Dependencies

```
# For audio recording
pip install sounddevice soundfile

# For mouse highlighting
pip install pyautogui

# FFmpeg (for audio/video merging)
# Download from https://ffmpeg.org/download.html
# Or use the included installer: python install_ffmpeg.py
```

Quick Start

Using the New Modular Version (Recommended)

```
python app_new.py
```

Using the Legacy Version

```
python app.py
```

Usage

1. **Set Output Path**: Click "Browse" to choose where to save your recording
2. **Configure FPS**: Set desired frame rate (recommended: 20-30 for most uses)
3. **Choose Capture Area**:
    - Monitor 0 captures all screens
    - Check "Capture specific region" for custom areas
4. **Enable Features** (optional):
    - Mouse highlighting with customizable appearance
    - Audio recording (microphone or system audio)
    - Webcam picture-in-picture overlay
5. **Start Recording**: Click "Start Recording"
6. **Stop Recording**: Click "Stop Recording"

Advanced Configuration

Audio Setup

- **FFmpeg Path**: Set the path to FFmpeg executable for audio merging
- **Device Selection**: Choose specific microphone or use system audio loopback
- **System Audio** (Windows only): Capture system sounds and music

Video Options

- **Monitor Selection**: Record specific monitors in multi-monitor setups
- **Region Capture**: Record only a portion of the screen
- **Preview Mode**: See what's being recorded in real-time

Performance Tuning for 10-Hour Recording

- **FPS Settings**: 15-20 FPS for extended sessions (balance quality vs. file size)
- **Region Capture**: Smaller regions reduce processing overhead
- **Memory Management**: The modular architecture provides better memory efficiency
- **Disable Overlays**: Turn off mouse/webcam overlays for maximum performance

Modular Architecture Benefits

Performance Improvements

- **Memory efficiency**: Better garbage collection with isolated modules
- **Threading optimization**: Dedicated threads for capture, processing, and encoding
- **Resource management**: Automatic cleanup prevents memory leaks during long recordings

Better Organization

- **Separation of concerns**: Each module handles specific functionality
- **Easier debugging**: Issues isolated to specific components
- **Code reusability**: Modules can be used independently

Maintainability

- **Simpler updates**: Modify features without affecting the entire codebase
- **Better testing**: Individual components can be tested in isolation
- **Cleaner documentation**: Each module is self-contained and documented

Troubleshooting

Common Issues

FFmpeg not found

- Download from https://ffmpeg.org/download.html
- Set path in UI or add to system PATH
- Use "Test" button to verify installation

Audio recording fails

- Install: `pip install sounddevice soundfile`
- Check device permissions
- Try different audio devices

Poor performance during long recordings

- Lower FPS setting (15-20 for 10+ hour sessions)
- Use region capture instead of full screen
- Disable unnecessary overlays
- Close other applications
- Use the new modular version (app_new.py) for better memory management

Video file corrupted

- Ensure sufficient disk space (estimate ~1GB per hour at 1080p/20fps)
- Check output path permissions
- Try different output location

Performance Tips for Extended Recording

Memory Management

- The modular architecture (app_new.py) provides better memory efficiency
- Use region capture for smaller memory footprint
- Monitor system resources during recording

Storage Planning

- Plan for ~1-2GB per hour depending on resolution and FPS
- Use fast storage (SSD) for best performance
- Consider disk space for temporary audio files during processing

System Resources

- Close unnecessary applications
- Use lower FPS for extended sessions (15-20 FPS)
- Monitor CPU temperature during long recordings

macOS Permissions

On first use, macOS will prompt for Screen Recording permission. Grant it under:
System Settings → Privacy & Security → Screen Recording

Version History

- **v1.0.0**: Modular architecture release with 10-hour recording optimization
- **v0.9.x**: Legacy monolithic version (still available as app.py)

Contributing

The modular structure makes contributions easier:

1. Identify the relevant module (core, ui, utils)
2. Make focused changes to specific functionality
3. Test individual components
4. Submit pull requests for specific modules