

(Autonomous)

RECORDING USING PYTHON

(Academic Year: 2020-21)

**Project Guide:** **Team Members:**

D.Deepthi, 1)19BQ1A05I5 -P.VANAJA (lead),

Assistant Professor, 2)19BQ1A05E0 –M.AJAY KUMAR,

Dept of CSE. 3)19BQ1A05D1- M.AKASH,

4)20BQ5A0516 – K.SIVAIAH.

**Department of Computer Science and Engineering**

**VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR**

**Table of Content**

1) Problem Statement.

2) Introduction.

3) Software & hardware Requirements.

4) Implementation.

5) Results.

**Problem Statement:**

Recorder is an application to record the screen, audio and web cam and save them to the current working file.

This provides user to watch the recorded videos and audios in offline.​

We have developed this project using python and we have used several modules in python to build this application.

**Introduction:**

Screen recording enables us to create demonstration videos, record gaming achievements and create videos that can be shared online on social media.​

Many industrial software exist out there that can help you do that very easily though.​

In our project we will show you how to make your own simple screen recorder in python that can further extend to your own needs.​

In webcam recording we can actually record that is running on our camera and save that file​

**Software & hardware Requirements**

**Hardware:**

1) **Operating System**: Windows 7 or higher/  
 Mac OS X 10.11 or higher/  
 Linux: RHEL 6/7.  
 2) **RAM**: 4 GB or above.  
 3) **Storage**: 1 GB free disk space.  
**Software:**

1) Python:

**-** Version-3.6 or above.

- Packages- (tkinter, pyautogui, cv2, numpy, win32api, sounddevice)

Download Links:

* Python : <https://www.python.org/downloads/>

**4) Implementation**

**1) Modules**:

**1. Tkinter:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the *Tkinter* module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

Widgets used:

* + - Button
    - Frame
    - Canvas
    - bind
    - PhotoImage

Geometry management tools used:

* + - grid()
    - place()

**2. Pyautogui**:

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be as simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3.

To install with pip, run: *pip install pyautogui.*

 Features:

* Moving the mouse and clicking or typing in the windows of other applications.
* Sending keystrokes to applications (for example, to fill out forms).
* Take screenshots, and given an image (for example, of a button or checkbox), find it on the screen.
* Locate an application’s window, and move, resize, maximize, minimize, or close it (Windows-only, currently)
* Display message boxes for user interaction while your GUI automation script runs.

**3. CV2:**

OpenCV is the huge open-source library for the computer vision, machine learning,

and image processing and now it plays a major role in real-time operation which is

very important in today’s systems. By using it, one can process images and videos

to identify objects, faces, or even handwriting of a human. When it integrated with

various libraries, such as NumPy, python is capable of processing the OpenCV array

structure for analysis. To Identify image pattern and its various features we use

vector space and perform mathematical operations on these features.

OpenCV: To install OpenCV type the below command in the terminal.

                                 pip install open cv-python

Pyautogui in Python can take screenshots, and then use Opencv to synthesize videos to achieve the purpose of screen recording.

1.  **cv2.imshow()**

As the name suggests this method is used for showing the images

Parameters: cv2.imshow(window\_name,image)

The window\_name should be a string value and hence will be in enclose in double or single quotes. The image shall be the image which should be

Example: cv2.imshow(“MrBean”,img)

2.**cv2.cvtColor()**

cvtColor can be interpreted as convert color and this method is used for converting the image from one color space to another color space (Find what is a color space here if you don’t know about it)

Parameters: cv2.cvtColor(src, code[, dst[, dstCn]])

Example: cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)

**3. cv2.waitKey()**

This function is very important, without this function cv2.imshow() won’t work properly.

Parameters: cv2.waitkey(wait time in milliseconds)

Thus if the wait time is entered as 6000, the picture will be displayed for 6s and then get closed (provided you have cv2.destroyAllWindows() in the script). If you use ‘0’ as the parmater then the image will be displayed for infinite time until you press the esc key.

Example: cv2.waitKey(0)

**4. cv2.destroyAllWindows()**

This method destroys (in other words “closes”) all the windows created using the opencv methods. If you want to close a specific window, then you can pass the window name as the argument within this function

Missing to provide the cv2.destroyAllWindows() at the end of the script might make the window opened to crash

**5. cv2.VideoWriter(**filename, fourcc, fps, frameSize**)**

             To save a video in OpenCV cv2.VideoWriter() method is used

6. **SOUNDDEVICE**

•This python module provides bindings for the portaudio library and a few convenience functions to play and record Numpy arrays containing audio signals. The sounddevice module is available for linux, macOS and Windows.

•Sounddevice.rec() : to record audio data from your sound device into a Numpy **array**

**5) Results:**

