

Playback Control of MP4 Files in a Directory

Participants:

1.SACHIN RAMESH KULKARNI SRN: PES2UG21CS449

2.SAI LITHISH DEGAPUDI SRN: PES2UG21CS456

PROBLEM STATEMENT: create a client-server program using socket programming in Python to fetch and play MP4 files from a particular directory. The server-side program should list all the available MP4 files and send the list to the client. The client-side program should display the available videos and ask the user to input the desired video name to play. After receiving the input from the client, the server should send the requested video to the client in chunks, and the client should play the video while allowing the user to pause or quit the program.

Short description on server:

The server is a Python script that listens for incoming connections on a UDP port and responds to requests from clients. Specifically, it allows clients to request a list of available MP4 video files in a specified directory and then streams the selected video file to the client in real-time using UDP packets. The server sends the video data in chunks, which are then displayed by the client using OpenCV. The server supports basic features such as pausing and quitting the video stream.

Short description of the client:

The client code is responsible for communicating with the server to fetch a list of available video files and then allowing the user to select a video to play. Once a video is selected, the client code receives video data from the server in chunks, decodes the data into frames using OpenCV, and displays the frames in a window. The client also allows the user to pause the video by pressing the 'p' key and to exit the video player by pressing the 'q' key.

Source code:

Server:

```
import socket
import cv2
import numpy as np
import os
import glob

UDP_PORT = 50001
VIDEO_DIR = r'C:\Users\lenovo\Documents\MPCA\videos'
CHUNK_SIZE = 65535

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('', UDP_PORT))

while True:
    print('waiting for instruction')
    data, addr = sock.recvfrom(1024)
    if data.decode() == "CONNECT":
        video_files = glob.glob('./*.mp4')
        video_list = []
        for v in video_files:
            a = v.split('\\')[1]
            video_list.append(a)
        sock.sendto('\n'.join(video_list).encode(), addr)
        print('Video list sent')
        data, addr = sock.recvfrom(1024)
        video_name = data.decode()
        if video_name == 'q':
            continue

        else:
            video_path = os.path.join(VIDEO_DIR, video_name)
```

```

        if os.path.exists(video_path):
            cap = cv2.VideoCapture(video_name)
            while(cap.isOpened()):
                ret, frame = cap.read()
                if ret:
                    frame = cv2.resize(frame, (250, 250))
                    # encode the frame
                    encoded_frame, buffer = cv2.imencode('.jpg', frame,
[cv2.IMWRITE_JPEG_QUALITY, 80])
                    # convert buffer to bytes
                    data = buffer.tobytes()
                    # send the data in chunks
                    for i in range(0, len(data), CHUNK_SIZE):
                        chunk = data[i:i+CHUNK_SIZE]
                        sock.sendto(chunk, addr)
                else:
                    break
            cap.release()
        else:
            print('Video not found:', video_path)
sock.close()

```

client:

```

import socket
import cv2
import numpy as np
import sys

UDP_IP = '192.168.13.1'
UDP_PORT = 50001

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto('CONNECT'.encode(), (UDP_IP, UDP_PORT))

print('Fetching video list...')
data, addr = sock.recvfrom(1024)
video_list = data.decode().split('\n')
print('Available videos:')
for video in video_list:
    print(video)

while True:
    video_name = input('Enter video name (or q to quit): ')
    if video_name == 'q':
        print('Exiting...')

```

```

        sock.sendto(video_name.encode(), (UDP_IP, UDP_PORT))
        sys.exit()
    elif video_name not in video_list:
        print('Video not found:', video_name)
        continue

    sock.sendto(video_name.encode(), (UDP_IP, UDP_PORT))

    while True:
        try:
            data, addr = sock.recvfrom(1<<17) #max limit of bytes received
from call

            if len(data) == 0:
                printf('Empty video')

            nparr = np.frombuffer(data, np.uint8)# convert the bytes to numpy
array

            vid = cv2.imdecode(nparr, cv2.IMREAD_COLOR)# decode the numpy
array to image

            cv2.imshow('Received',vid)# show the image

            if cv2.waitKey(1) & 0xFF == ord('q'):
                sock.close()
                cv2.destroyAllWindows()

            if cv2.waitKey(1) & 0xFF == ord('p'):
                cv2.waitKey(-1) #wait until any key is pressed

        except:
            print('Done playing :', video_name)
            cv2.destroyAllWindows()
            break

```

OUTPUT:

