Edward Goonzalez

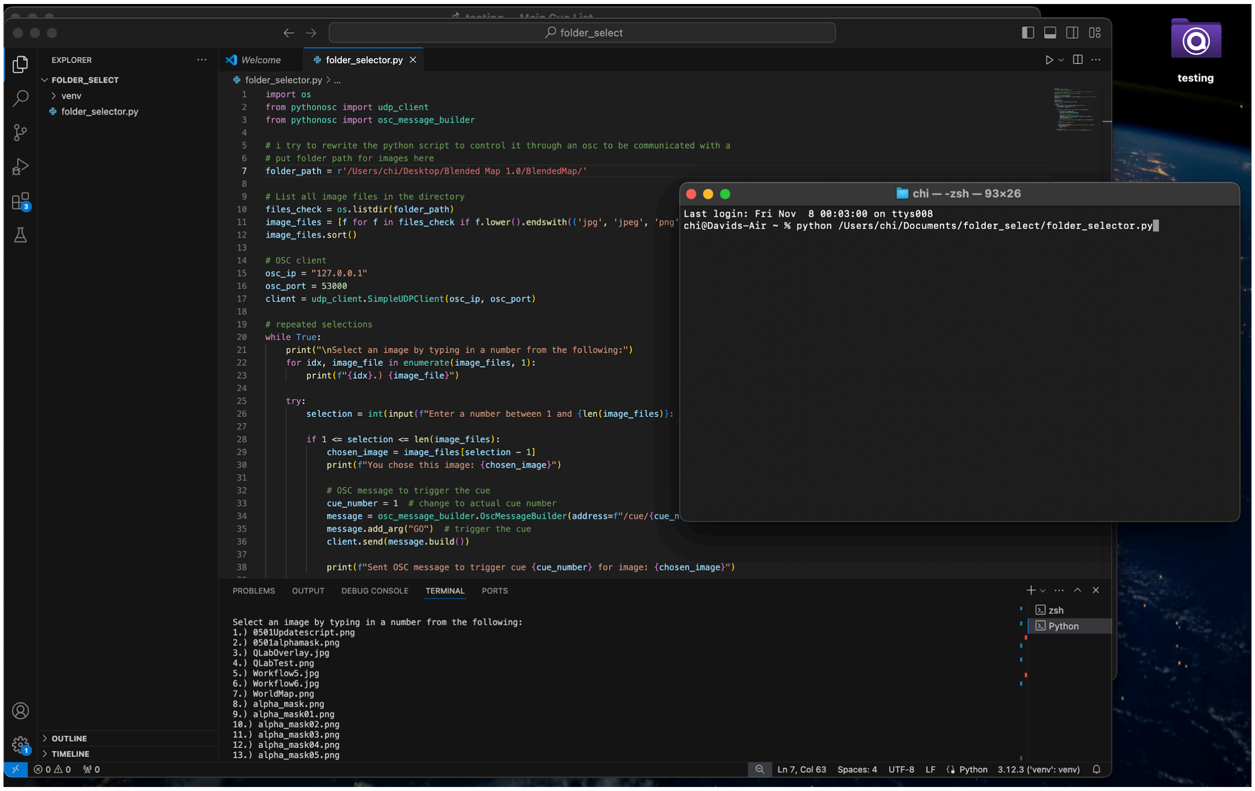
Non-Linear Narrative

December 5th, 2024

Basic Media Controls in QLab

This Python script allows users to select an image from a specified folder while sending an OSC (Open Sound Control) message to trigger an event, such as a "cue", based on the selected image. The selected image will show that it has been selected from the number "cue".

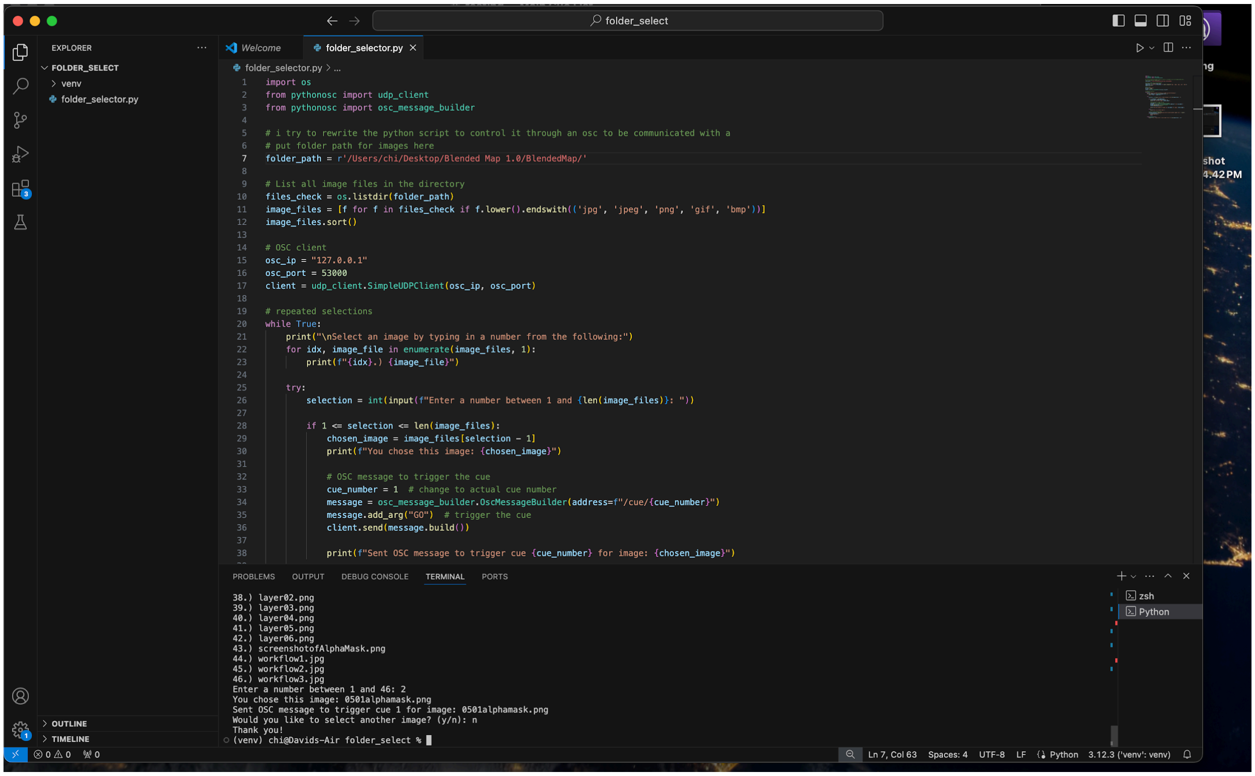
Images:



```python
import os
from pythonosc import udp_client
from pythonosc import osc_message_builder

# i try to rewrite the python script to control it through an osc to be communicated with a
# put folder path for images here
folder_path = r'/Users/chi/Desktop/Blended Map 1.0/BlendedMap/'

# List all image files in the directory
files_check = os.listdir(folder_path)
image_files = [f for f in files_check if f.lower().endswith(('jpg', 'jpeg', 'png'
image_files.sort()

# OSC client
osc_ip = "127.0.0.1"
osc_port = 53000
client = udp_client.SimpleUDPClient(osc_ip, osc_port)

# repeated selections
while True:
    print("\nSelect an image by typing in a number from the following:")
    for idx, image_file in enumerate(image_files, 1):
        print(f"{idx}.) {image_file}")

    try:
        selection = int(input(f"Enter a number between 1 and {len(image_files)}:

        if 1 <= selection <= len(image_files):
            chosen_image = image_files[selection - 1]
            print(f"You chose this image: {chosen_image}")

            # OSC message to trigger the cue
            cue_number = 1  # change to actual cue number
            message = osc_message_builder.OscMessageBuilder(address=f"/cue/{cue_n
            message.add_arg("GO")  # trigger the cue
            client.send(message.build())

            print(f"Sent OSC message to trigger cue {cue_number} for image: {chosen_image}")
```

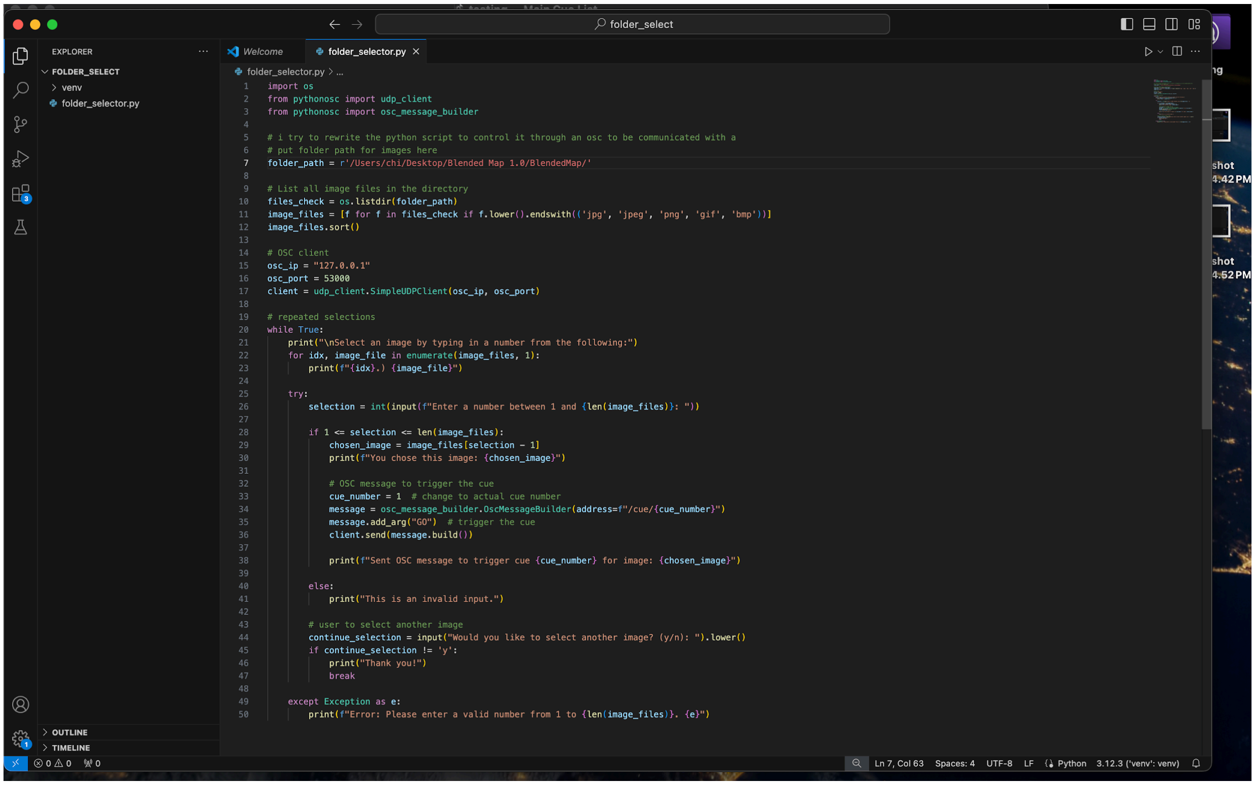
```
Select an image by typing in a number from the following:
1.) 0501Updatescript.png
2.) 0501alphamask.png
3.) QLabOverlay.jpg
4.) QLabTest.png
5.) Workflow5.jpg
6.) Workflow6.jpg
7.) WorldMap.png
8.) alpha_mask.png
9.) alpha_mask01.png
10.) alpha_mask02.png
11.) alpha_mask03.png
12.) alpha_mask04.png
13.) alpha_mask05.png
```

```
Last login: Fri Nov  8 00:03:00 on ttys008
chi@Davids-Air ~ % python /Users/chi/Documents/folder_select/folder_selector.py
```



```python
import os
from pythonosc import udp_client
from pythonosc import osc_message_builder

# i try to rewrite the python script to control it through an osc to be communicated with a
# put folder path for images here
folder_path = r'/Users/chi/Desktop/Blended Map 1.0/BlendedMap/'

# List all image files in the directory
files_check = os.listdir(folder_path)
image_files = [f for f in files_check if f.lower().endswith(('jpg', 'jpeg', 'png', 'gif', 'bmp'))]
image_files.sort()

# OSC client
osc_ip = "127.0.0.1"
osc_port = 53000
client = udp_client.SimpleUDPClient(osc_ip, osc_port)

# repeated selections
while True:
    print("\nSelect an image by typing in a number from the following:")
    for idx, image_file in enumerate(image_files, 1):
        print(f"{idx}.) {image_file}")

    try:
        selection = int(input(f"Enter a number between 1 and {len(image_files)}: "))

        if 1 <= selection <= len(image_files):
            chosen_image = image_files[selection - 1]
            print(f"You chose this image: {chosen_image}")

            # OSC message to trigger the cue
            cue_number = 1  # change to actual cue number
            message = osc_message_builder.OscMessageBuilder(address=f"/cue/{cue_number}")
            message.add_arg("GO")  # trigger the cue
            client.send(message.build())

            print(f"Sent OSC message to trigger cue {cue_number} for image: {chosen_image}")
```

```
38.) layer02.png
39.) layer03.png
40.) layer04.png
41.) layer05.png
42.) layer06.png
43.) screenshotofAlphaMask.png
44.) workflow1.jpg
45.) workflow2.jpg
46.) workflow3.jpg
Enter a number between 1 and 46: 2
You chose this image: 0501alphamask.png
Sent OSC message to trigger cue 1 for image: 0501alphamask.png
Would you like to select another image? (y/n): n
Thank you!
(venv) chi@Davids-Air folder_select % 
```



```python
import os
from pythonosc import udp_client
from pythonosc import osc_message_builder

# i try to rewrite the python script to control it through an osc to be communicated with a
# put folder path for images here
folder_path = r'/Users/chi/Desktop/Blended Map 1.0/BlendedMap/'

# List all image files in the directory
files_check = os.listdir(folder_path)
image_files = [f for f in files_check if f.lower().endswith(('jpg', 'jpeg', 'png', 'gif', 'bmp'))]
image_files.sort()

# OSC client
osc_ip = "127.0.0.1"
osc_port = 53000
client = udp_client.SimpleUDPClient(osc_ip, osc_port)

# repeated selections
while True:
    print("\nSelect an image by typing in a number from the following:")
    for idx, image_file in enumerate(image_files, 1):
        print(f"{idx}.) {image_file}")

    try:
        selection = int(input(f"Enter a number between 1 and {len(image_files)}: "))

        if 1 <= selection <= len(image_files):
            chosen_image = image_files[selection - 1]
            print(f"You chose this image: {chosen_image}")

            # OSC message to trigger the cue
            cue_number = 1  # change to actual cue number
            message = osc_message_builder.OscMessageBuilder(address=f"/cue/{cue_number}")
            message.add_arg("GO")  # trigger the cue
            client.send(message.build())

            print(f"Sent OSC message to trigger cue {cue_number} for image: {chosen_image}")
```

```python

        else:
            print("This is an invalid input.")

        # user to select another image
        continue_selection = input("Would you like to select another image? (y/n): ").lower()
        if continue_selection != 'y':
            print("Thank you!")
            break

    except Exception as e:
        print(f"Error: Please enter a valid number from 1 to {len(image_files)}. {e}")
```

Key Components:

- Image File Path Folder: The script lists all image files (JPG, JPEG, PNG, GIF, BMP) from a specified folder (folder_path).
- OSC Communication: The script uses the python-osc library to send OSC messages by getting a specified IP and port based on the QLab setup.
- User Interaction: The script prompts the user to choose an image by entering a number "cue" which corresponds to the list of images in the folder. Each image has an assigned number attached to them to be called back.
- OSC Trigger: Upon selecting an image, the OSC message is sent to a specific address to trigger an event (event = cue).

Description:

- Directory Listing: The folder path (folder_path) is set to a specific directory on the local system. All image files within the folder are listed, and only files with supported image extensions (jpg, jpeg, png, gif, bmp) are considered.
- User Image Selection: The user is shown a list of images; with numbers assigned to each of them. The user enters a number that corresponds to their choice of image.
- OSC Message: Once an image is selected, an OSC message is sent to trigger an event (e.g., a cue) with a predefined cue number (default is 1).

Repetition & Exit: After triggering the cue. The user is given a choice to either select another image or exit the program. This process repeats itself until the user opts to exit by entering 'n'.

Error Handling:

- If the user enters an invalid number (outside the list range), the script catches the error and prompts for a valid selection. Any other exceptions are caught and displayed as an error message.

Dependencies:

- python-osc: Library for OSC communication.
- os: For directory operations (Folder path).

Instructions to Run:

- Set the path to your image folder in folder_path.
- The OSC message is sent to the specified IP and port (osc_ip, osc_port).
- Run the script, and choose an image until you exit the program. Enjoy!