Interactive Generative Art Gallery

Subject : Design and implement an interactive art gallery using Python, where users can explore generative art, interact with creative visualizations, and manipulate images or audio in artistic ways.

Work Done by:

Akram EL ASSRI

Mohamed Anas Charkaoui

Major:

GINF

School year: 2024-2025

Concepts and Design of the gallery

Concepts of the gallery:

The gallery comprises a collection of applications, including:

- Drawing App
- Audio Processing App
- Image Manipulation App
- Data visualization App

Users can navigate to each app, each offering unique features that enable the creation of beautiful and creative artworks, which can be saved to the gallery as desired.

Drawing App Features:

The drawing app has features where the user can draw shapes,rotate them,change color and resize them, move them and delete them

the most interesting feature was the drawing functions implemented to generate beautiful mathematical shapes circular patterns rosecurves patterns and spirograph patterns

Data Visualization App Concept:

The concept of the Data visualization app was interesting , a user could upload a data set of his choice, choose cleaning options, choose his desired columns and the app would show what available plot can he generate

- The image manipulation app’s features were applying effects to a photo to generate beautiful effects
- The audio procesing app’s features were also applying effects but to audios the interesting one was mixing audios in my opinion
- lastly the style transfer app using a machine learning model to apply the style of another photo to the desired photo
- For the design of the app, we tought to use a simple yet elegant design that enables ease of use and subsenquently ease of mind while using the app

Image Manipulation App Features:

Apply effects to photos to create stunning visual results.

Audio Processing App Features:

Apply effects to audio files. The app offers an audio mixing feature, which is fun to play around with.

Design of the gallery:

For the design of the app, we tought to use a simple yet elegant design that enables ease of use and subsenquently ease of mind while using the app

example of the home page:

Techniques and tools used:

Backend:

- **Python**

Frontend:

- **HTML, CSS, JavaScript**

Web Framework:

- **Flask:** The web framework for the gallery

Drawing Application:

- **Pygame:** enabling drawing shapes with event handling for dynamic manipulation
- **pygame-gui:** An extension of Pygame that provides a simple GUI framework, enabling the creation of interactive elements like buttons and menus within the drawing app.
- **Shapely:** A library for the manipulation and analysis of planar geometric objects; employed to perform geometric calculations and manage shape properties.

Data Visualization Application:

- **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualizations in Python.
- **Seaborn:** Built on top of Matplotlib, Seaborn provides a high-level interface for drawing attractive and informative statistical graphics.
- **Wordcloud:** A tool for generating word clouds from text data, useful for visualizing the most frequent terms in a dataset.
- **Pandas:** A powerful data manipulation and analysis library; used for data cleaning, transformation, and preparation prior to visualization.

Image Manipulation Application:

- **OpenCV : f**or audio manipulation

Audio Processing Application:

- **pydub:** A simple and easy-to-use library for audio manipulation; used for tasks like applying effects and mixing audio tracks.

Challenges:

The first challenge was:

Initially, we aimed to develop a full-fledged desktop application using **Pygame** and **pygame-gui**, implementing a **state machine pattern** to switch between different app states(pages). This approach required **significant time and effort**, leading to large code (which is still available on GitHub but ultimately unused).

Due to **time constraints**, we pivoted to developing a **web-based application** using **Flask**, while retaining the drawing app portion from the desktop version.

The second challenge was:

The **lack of prior experience** with **HTML, CSS, and JavaScript**.

So We had to **quickly learn and apply** these technologies to create an engaging **user interface (UI)** and **user experience (UX)**.

Fortunately AI tools were useful to implement such features without hours spent in searching the web

Conclusion

This project was an interesting opportunity to delve into the world of pygame and create dynamic apps and also web programming, it was also interesting to think about artistic ways to implement this gallery, thank you professor for this valuable experience .