

Project's Report

Project's members : El Ayddouni Assaad - Zentari Abderrahman

Introduction

Our project is to create a platform that allows users explore digital art created using python. The goal is to combine technology with creativity.

- ➔ When the user enters our platform, he will find a bar on the top to select the implementation he wants.

Project Implementation

1) Generative Art

To create generative artworks, we used loops and conditional statements to generate randomized shapes and patterns. Object-Oriented Programming was applied to structure the code efficiently, allowing easy modification of shapes and styles.

- ➔ If the user chooses generative art, he must select the type and the color of the shapes he wants (could be random), and the way how to draw the shapes. And the last button of the page is to draw a spiral.

Techniques and tools used:

- **Turtle Graphics:** Used for drawing static generative art with simple geometric patterns and recursive design.

2) Data-Driven Visualization

For the data visualization component, the goal was to transform data into artistic visuals rather than conventional charts.

- ➔ At the data visualization, user can select the heatmap or the bar-chart of the dataset.

Techniques and tools used:

- **Pandas:** Used to load, process, and manipulate datasets before visualizing them.
- **Matplotlib:** Used to transform data into artistic visualizations, such as abstract graphs and customized charts.
- **Seaborn:** Used for advanced and aesthetically pleasing data visualizations, helping to make graphs more colorful and engaging.

3) Image and Audio Manipulation

To enhance interactivity, the project included image and audio manipulation features. Users could upload an image or an audio and apply various filters.

- ➔ If the user chooses the image and audio manipulation, he has to select the image or the audio and the filter he wants.

Techniques and tools used:

- **PIL (Pillow):** Used for basic image processing tasks like loading, saving, and simple modifications.
- **PyDub:** Used for manipulating audio files, such as adding effects, changing playback speed, and merging sounds to create unique compositions.

4) Interactivity & Web Integration

The entire project was hosted on a web application, making it accessible through a browser. Users could interact with the gallery, modify generative artworks, and navigate through different visualization pieces.

- ➔ This part of the project is to give build the web and give the users the access to the platform.

Techniques and tools used:

- **Flask:** Used to host the gallery as a web app, making it accessible to users online, and build the backend of the web application.

Challenges and Solutions

While working on this project, we faced many challenges, but the most difficult challenges were:

- There are fewer resources where we can develop our skills in using those libraries, so we waste time searching on youtube some tutorials on how to use the libraries and asking chatgpt to know the functions needed and solve problems.
- We wrote a lot of the code without testing, and when we tested, something goes wrong, and it was too difficult to debug the code and find the location of the error we made.
- At the end of the test, everything works, except the audio manipulation, and after a lot of search on youtube and chatgpt to find where the problem is, we find out that we installed Python 3.13 and this version does not support the audio manipulation library PyDub, so we deleted Python 3.13 and we installed Python 3.11.9 and it finally works.

Conclusion

This project successfully demonstrated the fusion of digital technology and artistic expression. The interactive generative art gallery offers a creative and engaging experience while reinforcing key programming and visualization skills. Future improvements could include AI-powered style transfer and additional interactive features to enhance user engagement.