Generative Vector Abstraction project

Brief outline of what it does

The code combines image-based patterns, artistic shapes, and interactive bouncing balls to create an engaging and visually captivating display. It offers a blend of artistic expression and dynamic motion, making it an intriguing visual experience.

❑ Imaginative Visual Display:

- The code creates a captivating visual display using the Processing programming language.
- It generates a mesmerizing pattern of colored circles based on an image.
- The circles are scattered randomly across the screen, creating an abstract representation of the image.

❑ Artistic Shapes:

- Alongside the circles, the code draws artistic shapes, including a triangle, square, circle, and lines.
- Each shape is filled with vibrant colors, adding visual interest to the display.
- The combination of shapes and circles creates an artistic and visually appealing composition.

❑ Bouncing Balls:

- The code introduces a set of bouncing balls that move around the screen.
- Each ball has a unique color and size, adding dynamic elements to the display.
- The balls exhibit gravity-like behavior, gradually accelerating their downward motion.
- When a ball reaches the screen's boundaries, it bounces back, creating an interactive effect.

Brief outline of the code

This project is written in the Processing programming language and creates a visual display with various elements.

❏ It declares and initializes variables and objects, including an image (`PImage` object) and an `ArrayList` of `Ball` objects.

❏ In the `setup()` function:

- It sets the size of the display window and sets the background color.
- Loads an image called "gerchard_richter.jpg."
- Configures the drawing settings (no stroke, fill color).
- Creates several `Ball` objects and adds them to the `Balls` list.

❏ In the `draw()` function:

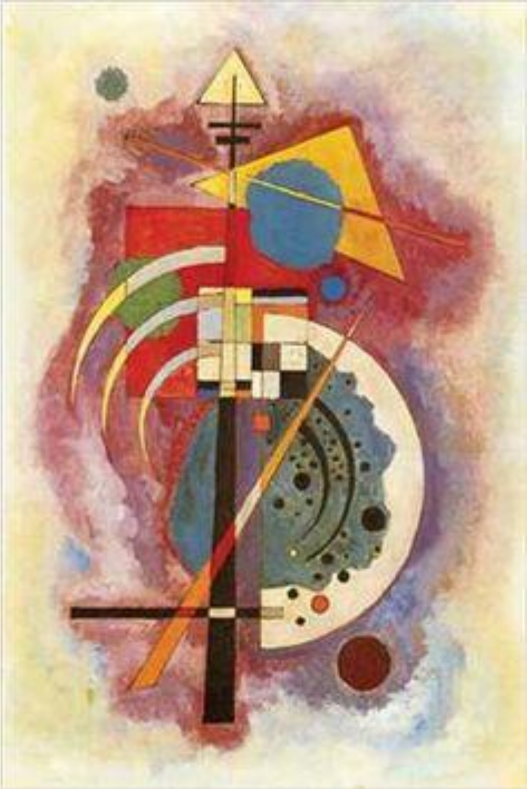
- Saves the current transformation matrix state.
- Repeats a loop 1500 times, generating random coordinates within the image dimensions.
- Retrieves the color values (RGB) from the corresponding pixel in the loaded image.
- Sets the fill color based on the retrieved RGB values with a reduced alpha (transparency).
- Draws an ellipse (circle) at the generated coordinates with the chosen fill color.
- Restores the transformation matrix state.
- Draws various shapes (triangle, square, circle, lines, and ellipses) with predefined colors.
- Updates the position of each `Ball` object, changes its color, and applies gravitational and bouncing effects.
- Finally, it displays each `Ball` object on the screen.

Notes inspiration about artworks

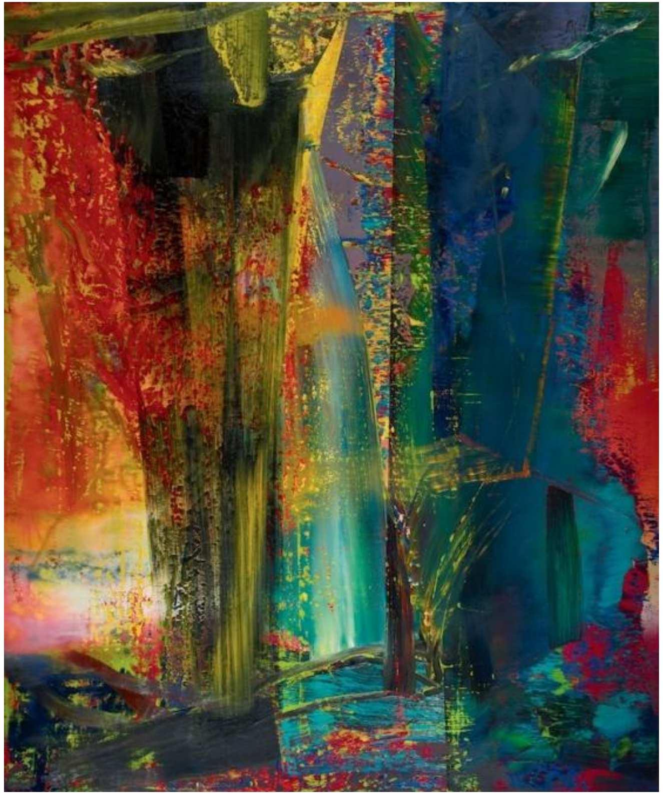
I have created a sketch that produces a colourful image resembling a specific 20th-century abstract artists.

As I embarked on writing the code, I decided to incorporate the styles of two artists into my work: Wassily Kandinsky's "Homage to Grohmann" and Gerhard Richter's "Abstract Painting 599" from 1986. Kandinsky's work captivates me due to its distinct shapes and their arrangements.

While coding, I carefully replicated the shapes from "Homage to Grohmann," such as triangles, circles, squares, and two lines. I meticulously applied colours to them and positioned them correctly. To introduce dynamic elements into the composition, I experimented with a moving ellipse, creating captivating variations. Furthermore, I employed my understanding of pixel structures to incorporate the colours from Richter's "Abstract Painting 599" into my sketch, utilizing ellipses and image manipulation techniques.



Wassily Kandinsky "Homage to Grohmann"



Gerhard Richter "Abstract Painting 599", 1986

Notes on the design decisions

Originally, I had planned to incorporate only Wassily Kandinsky's "Homage to Grohmann" into my code. However, I was truly inspired by the colour palette in Gerhard Richter's "Abstract Painting 599." It left such a profound impression on me that I couldn't resist merging the two artworks in my project. I envisioned using Gerhard Richter's "Abstract Painting 599" as the background, allowing the geometric shapes to boldly emerge against the vibrant canvas and creating an intriguing exotic composition. Imagine the interplay of colours and forms, producing mesmerising visual effects that transport us into the realm of abstract art.

My goal was to capture the essence and style of both artists, merging their works into a unique and awe-inspiring piece of art that will captivate and delight viewers. Additionally, I aimed to infuse the final creation with a sense of depth and emotion, inviting observers to explore the depths of artistic expression.

Explanation to the viewers on how to run the art project

In this mesmerising display, I have created a captivating visual experience using a combination of abstract forms, vibrant colors, and dynamic movements.

The centerpiece of this project is a beautiful image, "gerchard_richter.jpg", which serves as the foundation for the visual elements. By analysing the colors within the image, I have transformed them into an array of enchanting points that scatter across the canvas.

You will notice a variety of shapes elegantly dancing across the screen. Triangles, squares, and circles gracefully blend together, each representing a unique aspect of the artistic expression. The colors used for these shapes were carefully selected to evoke emotions and create a harmonious composition.

This project is a fusion of art and technology, showcasing the power of creativity and programming. It invites you to immerse yourself in a world of visual delight and explore the intricate interplay between colors, shapes, and motion.

Explanation on how to run a code

In my code I have 2 tabs:

❑ Abstract_art_final_main_java
❑ Ball

So to run a code you need to open 2 tabs and add each code into different tab.