1.0 Problem Description

Our project aims to convert a painting of the Eiffel tower into a specific art style using only circles and triangles. The art style is defined as follows: 300-350 shapes are used in total, with each shape having a size of 25-50 pixels and spaced at least 10-20 pixels apart. The shapes are allowed to overlap, but no more than 50% of one shape can be covered by another. The outer lines of the shapes are bold, with a width of 2 pixels, and the color scheme should consist of only 2-3 primary colors. The shapes should be evenly distributed throughout the image, without clustering in any particular area.

To accomplish the above, we use genetic algorithms. First, we generate population images through placing circles/ triangles. Each image is evaluated with the fitness function MSE to determine similarity of images with given input. Then, the best performing images undergo mutations and crossovers. Mutation involves moving the location, changing the shape, or changing the color of a shape, while crossover involves combining two candidate images to produce an offspring image. Lastly, we repeat the process till we satisfy our criteria.

After conducting extensive research, we have concluded that a genetic algorithm is the best approach for our project. Genetic algorithms are frequently used to solve optimization problems and are well-suited to explore large search spaces to find an optimal solution. In our case, our objective is to transform a real image of the Eiffel tower into a newly generated one while ensuring that it meets our specific defined art style constraints. Although other techniques such as neural style transfer can produce high-quality images, they may not meet our specific art style properties. In contrast, genetic algorithms are more suitable for generating images that satisfy our art style properties.

2.0 Proposed Method

Initialize Population Generation: We generate 100 population images by placing 300 to 350 circles and triangles on the image space. Each triangle is represented as a class, containing information on its properties and possible mutations. The triangle is represented by 3 points in x-y coordinates and its color opacity on the paint. Circles are represented as a single center point. The painting object, represented as a class, consists of triangle and circle objects, along with necessary functions to draw the shapes on the paint. The drawing is done using the Python Imaging Library (PIL). The main function here is to combine the two best parent objects to create an offspring.

Fitness Evaluation: Once a candidate image is generated, it is passed into a fitness function to evaluate its performance. The fitness function takes in two parameters: the generated image and the input image, and returns a scalar value between 0 and 1. The Mean Squared Error (MSE) between the generated image and the input image is used to calculate the fitness score. The MSE is defined as $MSE = (1/n) * \Sigma (A(i,j) - B(i,j))^2$, where n is the total number of pixels, and $a(i,j)$, $B(i,j)$ is the intensity of the pixel located at position (i,j) in image A and B. A value closer to 1 implies that the candidate image is similar to the input image.

Selection: We use tournament selection to select the best-performing images. The tournament function takes in all generated images, the size of the subset, and the fitness function. First, we select a subset of k images from the 100 population images. From that subset, we select a few images that are close to the highest fitness score and append them into a list.

Mutation: We change the shape properties during mutation. The types of mutation include moving the location of the triangle by shifting its 3 points' coordinates, changing the shape of the triangle by moving each point independently, and changing the color of the triangle. The magnitude of change in these mutations can be controlled either by using a uniform distribution to generate a random value or by assigning a parameter for humans to control. The same process can be repeated for circles, except that circles have a single point with radius adjustment. The mutation process uses the list obtained from selection and a mutation probability. The function randomly selects the types of mutations to occur on two shapes.

Crossover: The undergone mutations are then passed into the crossover function. The function takes in the two best mutated parent images and returns an offspring image. A random crossover point (e.g. half of the image vertically) is set on the image. The upper half of the image is taken from one parent and the lower half from another parent. These two halves are combined to create an offspring image, and the process is repeated.

Previous work: Proost using genetic algorithms to generate minimalist art, represented chromosomes as a list of 150 triangles with specific shape, position, and color, and swapped the positions of half of the triangles from two different parents.

3.0 Necessary Materials

List of Required Resources Physical Resources: The genetic algorithm for image generation requires a Core i7 processor, 16 GB RAM, and a 4GB graphics card. Informational Resources: We require knowledge to guide the project, such as implementing genetic algorithms for AI art and image processing details. Resources can be obtained from IEEE Xplore Digital Library and arXiv. An architecture image is also necessary and can be sourced from Google or Kaggle. Personnel Resources: Dr. Brown and online forums can provide additional assistance.

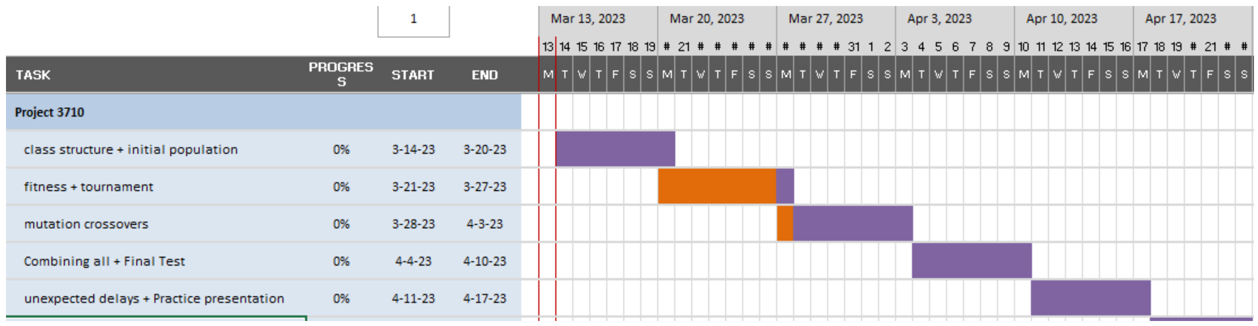
4.0 Evaluation

In order to evaluate the output of our method, we plan to use a combination of objective and subjective measures. For the objective measure, we will use the Mean Squared Error (MSE) between the generated image and the input image as a measure of similarity. This will provide us with a numerical score that indicates how close the generated image is to the input image.

For the subjective measure, we will conduct a user study to gather feedback from participants on the generated images. Participants will be asked to rate the generated images on a scale of 1 to 10 based on various criteria such as overall similarity, shape distribution, color scheme, and boldness of the outer lines. This will give us a more qualitative understanding of the generated images and help us identify areas for improvement.

Our measure of success is that the generated image should have a MSE score close to 1 and a high rating from the participants in the user study. On the other hand, our measure of failure would be if the generated image has a low MSE score and a low rating from the participants in the user study. In either case, we will use the results of the evaluation to make changes to our method and improve the generated images.

5.0 Timeline



TASK	PROGRESS	START	END
Project 3710			
class structure + initial population	0%	3-14-23	3-20-23
fitness + tournament	0%	3-21-23	3-27-23
mutation crossovers	0%	3-28-23	4-3-23
Combining all + Final Test	0%	4-4-23	4-10-23
unexpected delays + Practice presentation	0%	4-11-23	4-17-23

Week 1 (March 14 - March 20): Implement and finalize the class structure for representing the triangle and circle shapes, as well as the painting object. And, implement the initial population generation, which involves placing 300 to 350 circles and triangles on the image space. Conduct initial testing to ensure code correctness.

Week 2 (March 21 - March 27): First, Implement the fitness evaluation function, which takes in the generated image and input image and returns a scalar value between 0 and 1 using the Mean Squared Error (MSE) as a measure of similarity. Second, Implement the tournament selection method, which selects the best-performing images from the population. Conduct initial testing to ensure code correctness.

Week 3 (March 28 - April 3):
Implement the mutation function, which changes the properties of the shapes (location, shape, or color). Implement the crossover function, which combines the two best mutated parent images to produce an offspring image.

Week 4 (April 4 - April 10): Combine all defined methods into one.Conduct final testing and make any necessary improvements to the method. Document the project and prepare a final report.

Final Week (April 11 - April 16):
Practice presentation.

6.0 Risk Disclosure

Lack of Knowledge & Heavy Schedules*:* Our project faces the risk of failure due to limited knowledge of genetic algorithms and heavy school schedules. To mitigate, we'll allocate more time for learning and understanding, and adjust our timeline weekly.

Lack of Resources: The specifications of physical resources such as graphic card and processor may impact the quality of rendering for each candidate image.

Inconsistent Fitness: The fitness function using MSE may produce inconsistent results, leading to inaccurate evaluations of the generated images. This risk will be addressed by thoroughly testing and validating the fitness function.

Limited Mutations: Insufficient mutations can result in a lack of diversity in the generated images. To mitigate this risk, the mutation process will be fine-tuned to ensure sufficient changes and diversity.

Inefficient Crossover: The crossover process may not result in efficient combinations and a suboptimal solution. To address this risk, the crossover process will be optimized for efficient combinations.

Complex Mutations: Complex mutations can lead to low-quality generated images. To avoid this risk, the mutation function will be thoroughly tested and have a wide range of possible mutations to avoid getting stuck in a local optimum.

Error Code: Debugging errors may occur during the development of the project, leading to incorrect results or even failure. To mitigate this risk, we plan to thoroughly test each component and implement debugging tools, such as print statements and breakpoints. We will also regularly back up our code to ensure that any unexpected errors can be easily resolved.

7.0 Reference

[1] S. Proost, "Minimalist Art Using a Genetic Algorithm," Jan. 12, 2020.
[2] S. DiPaola, “Incorporating characteristics of human creativity into an evolutionary art algorithm,” Apr. 2010.