At office hours, I was advised to use the same mutations as the sample because my images weren't coming out nicely:

- With 1/5 probability, replace a polygon with a random triangle.
- With 1/5 probability, swap the order of two polygons which are in adjacent depth layers.
- With 3/5 probability, mutate an existing polygon:
    - With 1/2 probability, alter the color slightly.
    - With 1/2 probability, alter the shape of the polygon:
        - With 1/3 probability, remove a vertex.
        - With 1/3 probability, add a vertex at a random point.
        - With 1/3 probability, move a vertex slightly.

However, this did not solve the problem. It turned out I hadn't been using the pygame draw functionality correctly. Unfortunately, it would take several days to regenerate all the animations, so I stuck with these mutations.

Coordinates of vertices were moved in the x, y directions using a Gaussian distribution of mean 0 and standard deviation IMG_WIDTH/5 and IMG_HEIGHT/5 pixels, respectively. Components of colors (0-255 scale) were altered using a Gaussian distribution of mean 0 and standard deviation 25.

Random triangles were created by picking a point on the image with uniform distribution, then adding a Gaussian distribution of mean 0 and standard deviation IMG_WIDTH/10 and IMG_HEIGHT/10 to obtain the (x, y) coordinates of vertices. Random colors were picked using uniform distribution over the RGB color space.

Parents were selected proportional to their fitness. This was done with a weighted average where if parent i has f_i fitness, and the sum of the fitness over all parents was f, then parent i has probability f_i/f of being selected.