```
 Mutations used:
    10% chance to mutate color and shape of up to 10% of all polygons in genome.
    10% chance to replace a polygon in the genome with a random triangle.
    20% chance to swap the order of two polygons which are in adjacent in the genome.
    20% chance to mutate an existing polygon's color and shape.
    20% chance to mutate an existing polygon's color.
    20% chance to mutate an existing polygon's shape.

Shape Mutation:
   > With 1/3 probability, remove a vertex.
   > With 1/3 probability, add a vertex close to an edge.
        - If the slope of the edge wass undefined, the y component was chosen at the midpoint,
        and the x component was chosen with a Gaussian distribution w/ mean midpoint_x and stdev
        of 10% of total length of the edge.
        - If the slope of the edge was 0, the x component was chosen at the midpoint, and the y
        component was chosen with a Gaussian distribution w/ mean midpoint_y and stdev of 10% total
        length of the edge.
        - Otherwise, the x component was chosen uniformly between x_min and x_max of the edge while the
        y component was chosen uniformly between y_min and y_max of the edge.
   > With 1/3 probability, move a vertex slightly.
        - A Gaussian distribution w/ mean 0 and stdev 5 pixels was added to the x component and y
        component of the vertex.

Color Mutation:
   > Mutate each of the polygon's R,G,B,A values:
        With 1/2 probability add 'val' chosen with uniform distribution in set {0, 5, 10, 20, 30}
        With 1/2 probability subtract 'val' chosen with uniform distribution in set {0, 5, 10, 20, 30}

Random triangles were created by picking a point on the image with uniform distribution, then adding a
Gaussian distribution of mean 0 and stdev 20% of the image width to the initial point's x coordinate and
adding a Gaussian distribution of mean 0 and stdev 20% image height to the initial point's y coordinate
to obtain the second and third vertices. Random colors were picked using uniform distribution over the
RGB color space.

Parents were selected from an array of individuals sorted in decreasing order of fitness. The first
parent was chosen using the index abs(random.gauss(0, 0.1*N)). Then, the first parent was removed from
possible parent choices, and the second parent was chosen using abs(random.gauss(0, 0.1*N)). This made
the more fit individuals more likely to be chosen for crossover. Once the individuals were chosen, the
number of polygons to take from the first parent was chosen uniformly using random.randint(0, P).
```