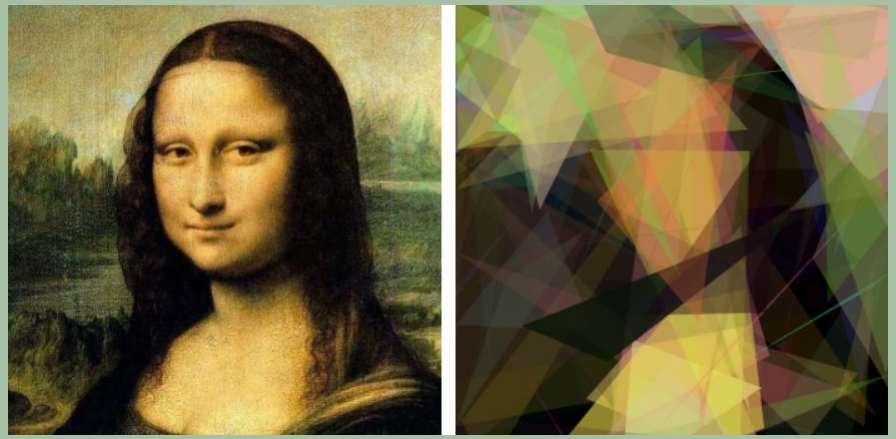
something old
something new

Vivian Dai
Julie Helenek
Connie Tsang

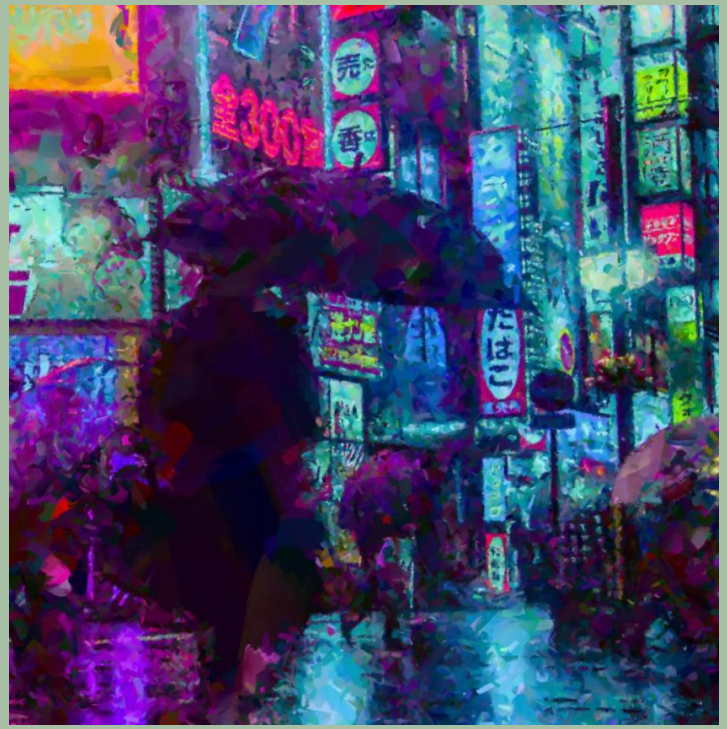
our goal

use a genetic algorithm to create new, abstract pieces of art

our inspiration



Chris Cummins Generative Art



Shahriar Shahrabi Procedural Paintings

step 1: research

What is a genetic algorithm?
How can we use this algorithm to produce art?

key components of a genetic algorithm

population

create a population of randomly generated *individuals*

selection

evaluate the fitness of individuals in the population and select the most fit to join the *mating pool*

reproduction

create a child individual with *crossover* and *mutation*

step 2: mvp

What is our minimal viable product based on our research?

components of mvp

individual

a 2D-matrix the size of the image

population

a list of individuals

gene

a single gene corresponds to a **pixel**

early outputs

attempt 1

attempt 2

attempt 3

tuning parameters

of generations, population size, mutation rate

original

fitness: 65.6%
time: 62.9s
generations: 3500
population size: 75
mutation rate: 0.10

generated

fitness: 80.4%
time: 67.6s
generations: 3500
population size: 75
mutation rate: 0.01

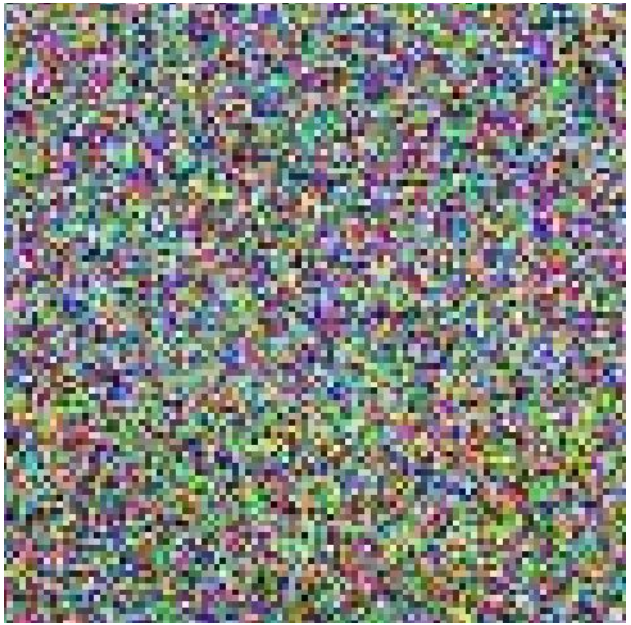
generated

fitness: 71.9%
time: 8.2s
generations: 350
population size: 75
mutation rate: 0.01

generated

challenges

didn’t perform well with larger pictures



step 3a: brushstrokes

Can we use brushstrokes instead of individual pixels?

outline

- Original idea was to use brushstrokes as individuals, similar to polygons' first iteration
  - Made use of the concept of “mutation” in genetic algorithms

- Changed the algorithm to more resemble the pixels and polygons implementation

components of brushstrokes

individual

a list of brushstrokes (i.e., a canvas)

population

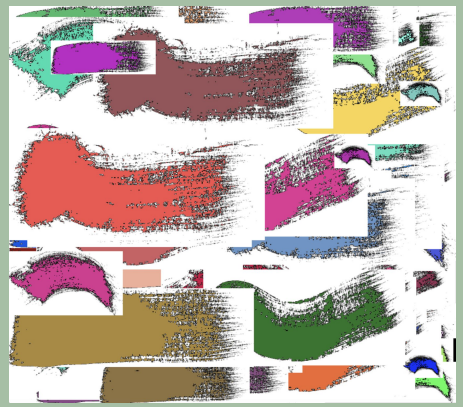
a list of individuals (a list of many canvases)

gene

a single gene corresponds to a **brushstroke.** each brushstroke has its own individual attributes, such as color and type.

early outputs

output



original image

Next steps

- Add transparency
- Optimize runtime because it runs slowly

step 3b: polygons

Can we use shapes instead of individual pixels?

components of polygons

individual

a list containing sets of vertices

↕

a 2D-matrix that has polygons drawn onto it

population

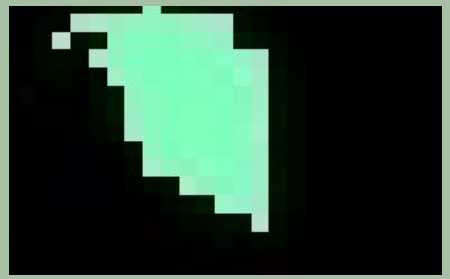
a list of individuals

gene

a single gene corresponds to a **polygon**

early outputs

attempt 1



fitness: 40.2%
time: 142.2s
generations: 300
population size: 75
mutation rate: 0.01

attempt 2

fitness: 84.4%
time: 197.6s
generations: 300
population size: 75
mutation rate: 0.01

attempt 3

fitness: 91.8%
time: 376.8s
generations: 300
population size: 75
mutation rate: 0.01

tuning parameters

of generations, population size, mutation rate, opacity

original

fitness: 95.24%
time: 301.5s
generations: 400
population size: 51
mutation rate: 0.02

generated

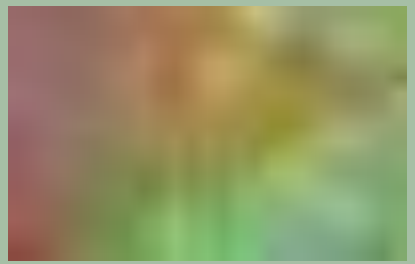
fitness: 95.24

another example

original

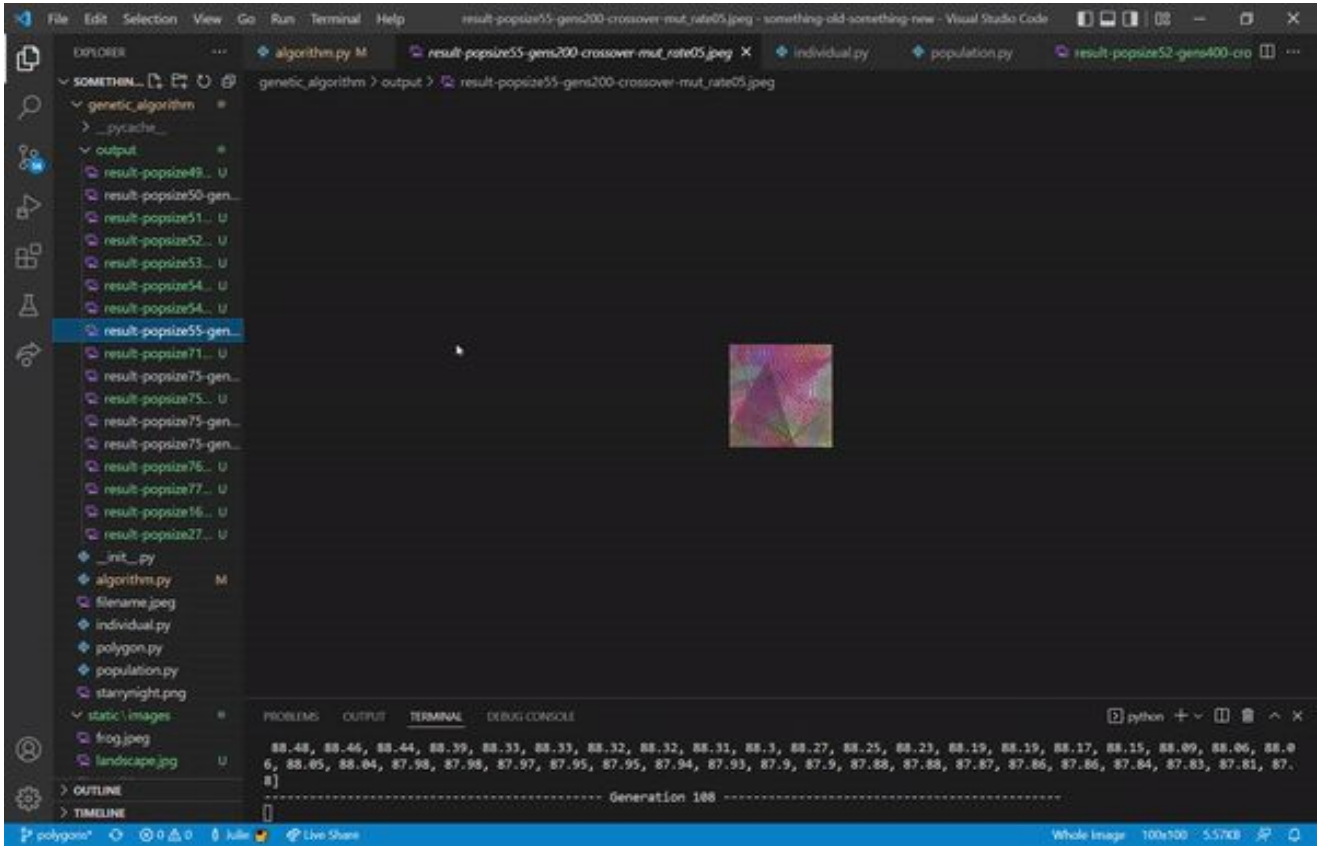
fitness: 93.07%
time: 241s
generations: 300
population size: 75
mutation rate: 0.01
polygons: 125
polygon vertices: 3

generated



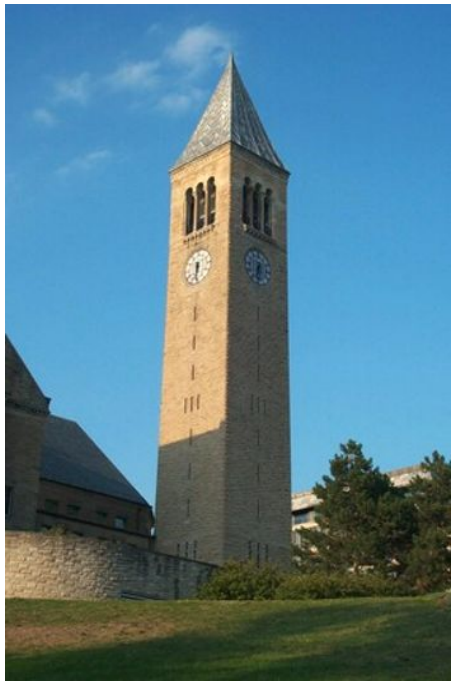
challenges

after a while, the best individual seemed to stop changing



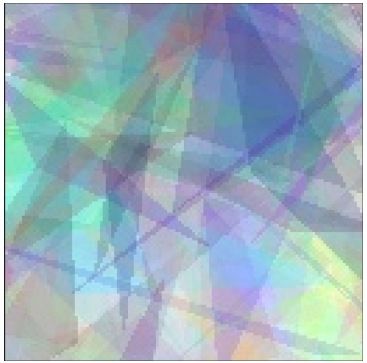
challenges

didn’t perform well with even larger pictures



challenges

struggles with light images



Thank you.