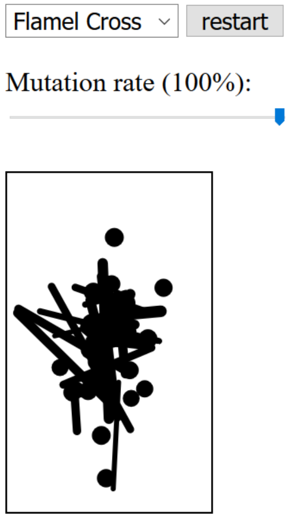
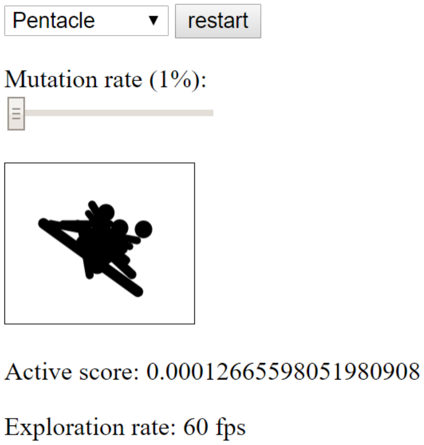
P4 - Flamel Cross, Pentagram, and Idle

https://glitch.com/~p4-evolutions



Flamel Cross ⌄ restart

Mutation rate (100%):

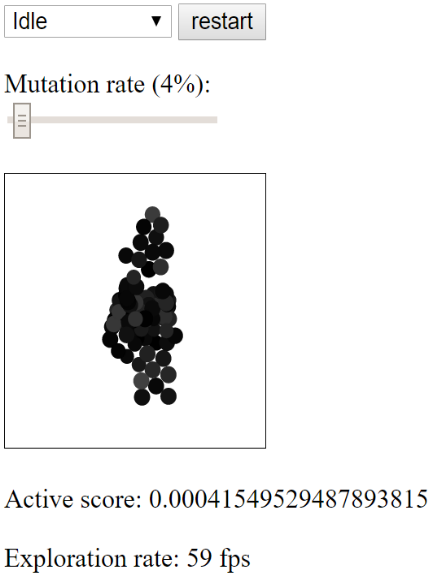


Pentacle ▼ restart

Mutation rate (1%):

Active score: 0.00012665598051980908

Exploration rate: 60 fps



Idle ▼ restart

Mutation rate (4%):

Active score: 0.00041549529487893815

Exploration rate: 59 fps

An Evolutionary Image Rasterizer

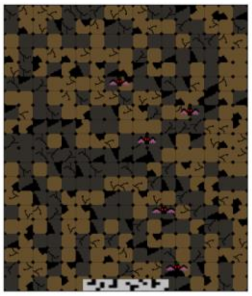
This project evolutionarily rasterizes a single image using a handful of shapes I provided and an inspiring image I also provided. Two of the images – the Flamel Cross and the Pentagram – are simply two shapes that I like that I pulled off Google Images. The third was a creation of my own design.

Each image varies slightly in its number of shapes to draw from and the shapes themselves: the Cross and the Pentagram pull from 50% lines and 50% dots (the Cross has a total of 50 shapes, and the Pentagram 30), while the Idle image has 100 dots with which to draw itself.

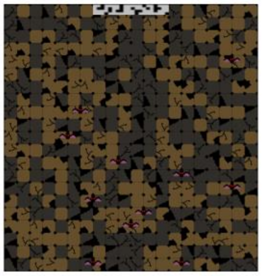
For the Cross and Pentagram, I chose to give them lines since both images are so line-heavy, though I saw no reason to make them comprised of any more than half lines and half dots, as I figured whatever the lines failed to create, the dots could fill in for. The Idle needed only dots, though unlike the other two, it does have to vary in color, if only in a grayscale manner.

P3 – Population 1: Dungeon Rooms

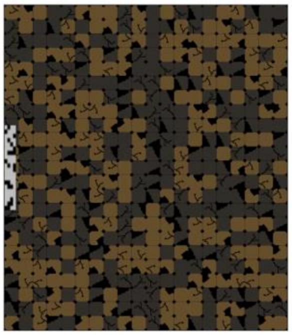
https://glitch.com/~p3-pop1-dungeonrooms



A ominous altar with a southern door and some bats.



A foreboding altar with a northern door and some bats.



A dark temple with a western door

An Infinite Dungeon Room Creator

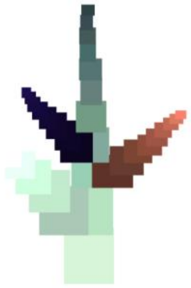
For my first P3 population generator, I decided to create a dungeon room of a "random" size and with some of its features – such as floor tile layout, inhabitants, and door placement – randomized as well. The text description assigns an adjective and name to the room, as well as giving the location of the door and saying whether or not it has bats, which ended up being, unfortunately, the only kind of occupant. Given more time to dedicate to this project, I would have thoroughly enjoyed creating some more possible occupants for the room and placing them in randomly as well – in the future I may do just that as a creative and artistic exercise.

All of the assets in this generator were created by me, with the colors of the floor tiles pulled from my inspiring image. When placing the floor tiles, I used nested for loops – one that iterates along the y axis and the other on the x, with the iterator incrementing by the width of the tiles. With some help from Professor Smith, I was also able to rotate each tile randomly, giving even more variance to the appearance of the room in a way that I found satisfying given the scope of the project and the time I had to work on it.

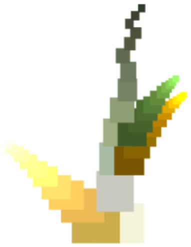
When placing the bats, there is essentially just a 1/3 chance of no bats being placed at all, a 1/3 chance of there being a 3% chance of a bat being placed on any given tile, and a 1/3 chance of there being a 1.5% chance of a bat being placed on any given tile. One interesting visual perception in the bats is their size: the width and height of each bat varies from 90 – 125% of its original value. As a sign of life, each bat has a different interval over which it blinks – essentially, each bat blinks once every few seconds, but each bat has its own timer on which it does so.

P3 – Population 3: Monoliths

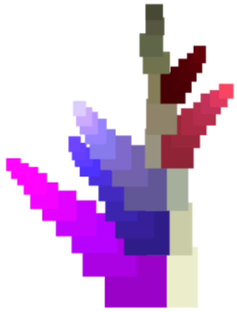
https://glitch.com/~p3-pop--monolith-maker



This monolith is 11 rocks tall with 3 branches.



This monolith is 13 rocks tall with 3 branches.



This monolith is 10 rocks tall with 5 branches.

An Infinite Monolith Maker

My third and final population generator creates an infinite series of colorful monoliths that are each between 10 and 15 “rocks” (which are just rects) tall that branch (supposedly) anywhere between 3 and 5 times. The text tells how tall the monolith is in “rocks” and how many branches it has, with about 90-ish% accuracy – this is due to the fact that with the way I told the program to draw the branches, there is a small chance it won’t draw all the branches, causing the number of branches mentioned by the text and the number of branches drawn to sometimes differ a bit, though I did manage to minimize the odds of such an event happening. For the positioning of each rock, I moved the next one up by 80% of its own height, then shifted it on the x axis by anywhere between 97 – 103% of its own location – essentially moving it left or right by up to 3%, which made for some of the tops of the monoliths to vary interestingly.

For the color of the base monolith, I picked 3 random values between 200 and 255 and put those into a fill command, then with every drawn rock I set those values to be anywhere between 80-95% of themselves (e.g. r *= rand(.8, .95)), ensuring that the color of each rock would both evolve and become darker in a way that heuristically seems to make sense.

For the positioning, colors, and lengths of the branches I pulled what I felt were a few interesting tricks. For their positions, I picked randomly between left and right, then moved the first branching rock 50% to the side of the base rock on the x axis without modifying the y; for every branch rock after that, I modified the x and y by the size of the current rock/2, then multiplied that by a random value between .5 and .9, ensuring some variation in the positioning of the rocks and giving some of them some very interesting curves. For their colors, I chose random values between 0 and the current r, g, and b values of the starting base rock, then iterated those values by 5 – 20% with every rock; this ensured my rocks would be colorful and, unlike the base, be getting lighter instead of darker, both of which I felt added some interesting perceptual variation, even if they didn’t necessarily “make sense”. For the length of the branches, each branch has just as many rocks as remain to be drawn in the monolith – so if a branch is drawn at the second rock of a 12 rock tall monolith, that branch will be 10 rocks long.