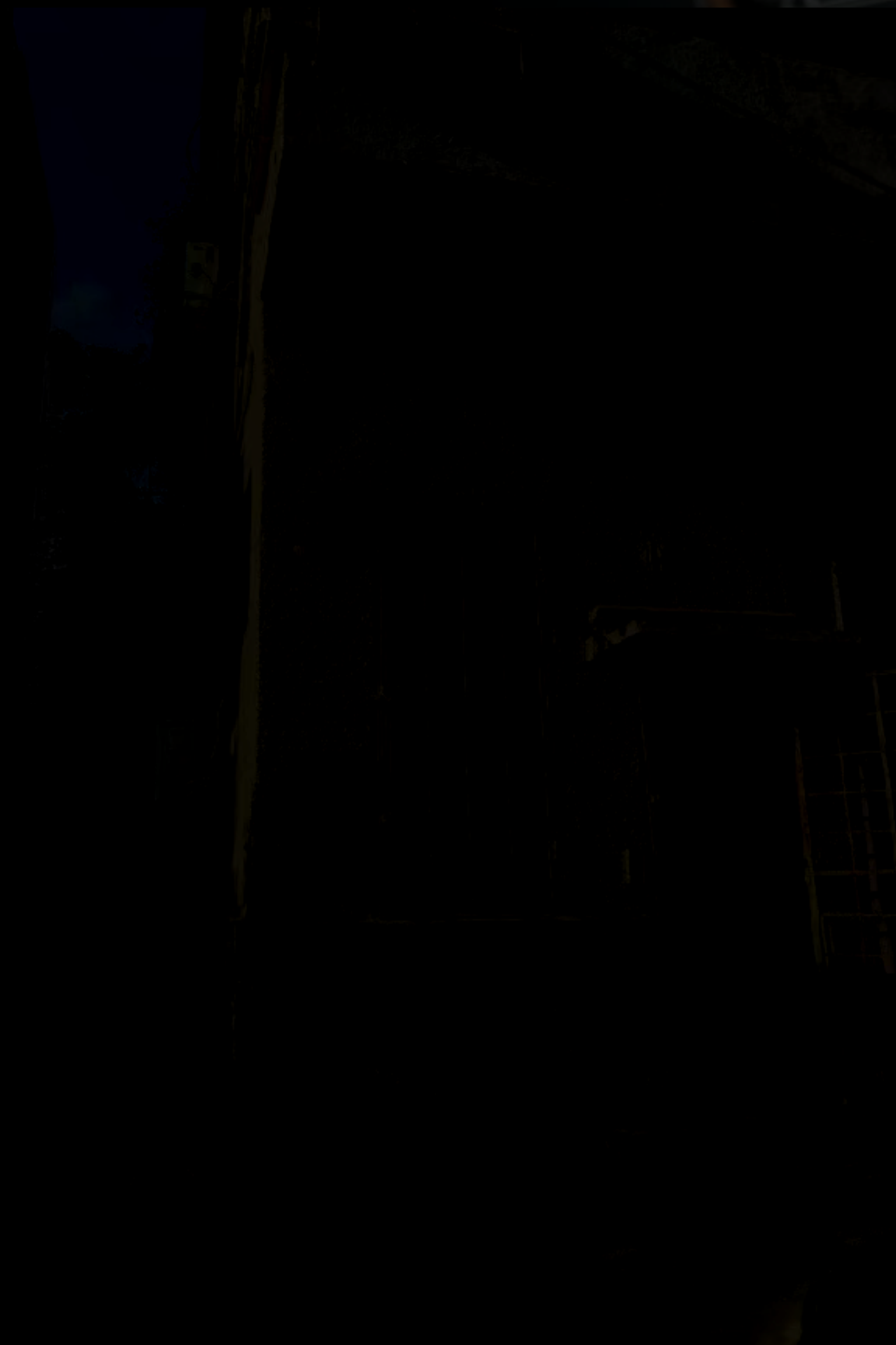




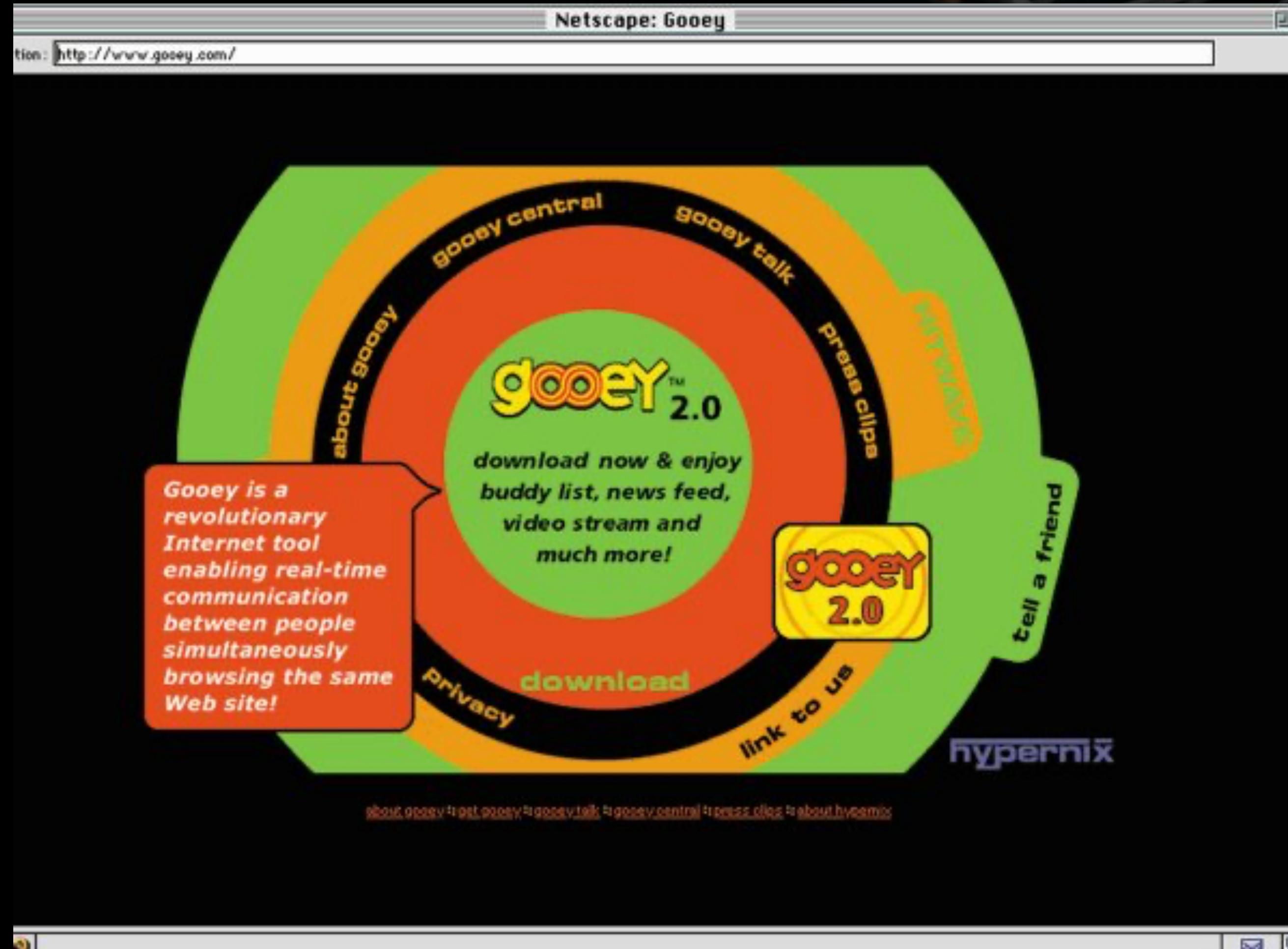
Evolving JavaScript Cultivating Genetic Algorithms for Creative Coding

שלום!

11 Nachmani



From the way way way back machine



FLASH 4

Creative Web Animation



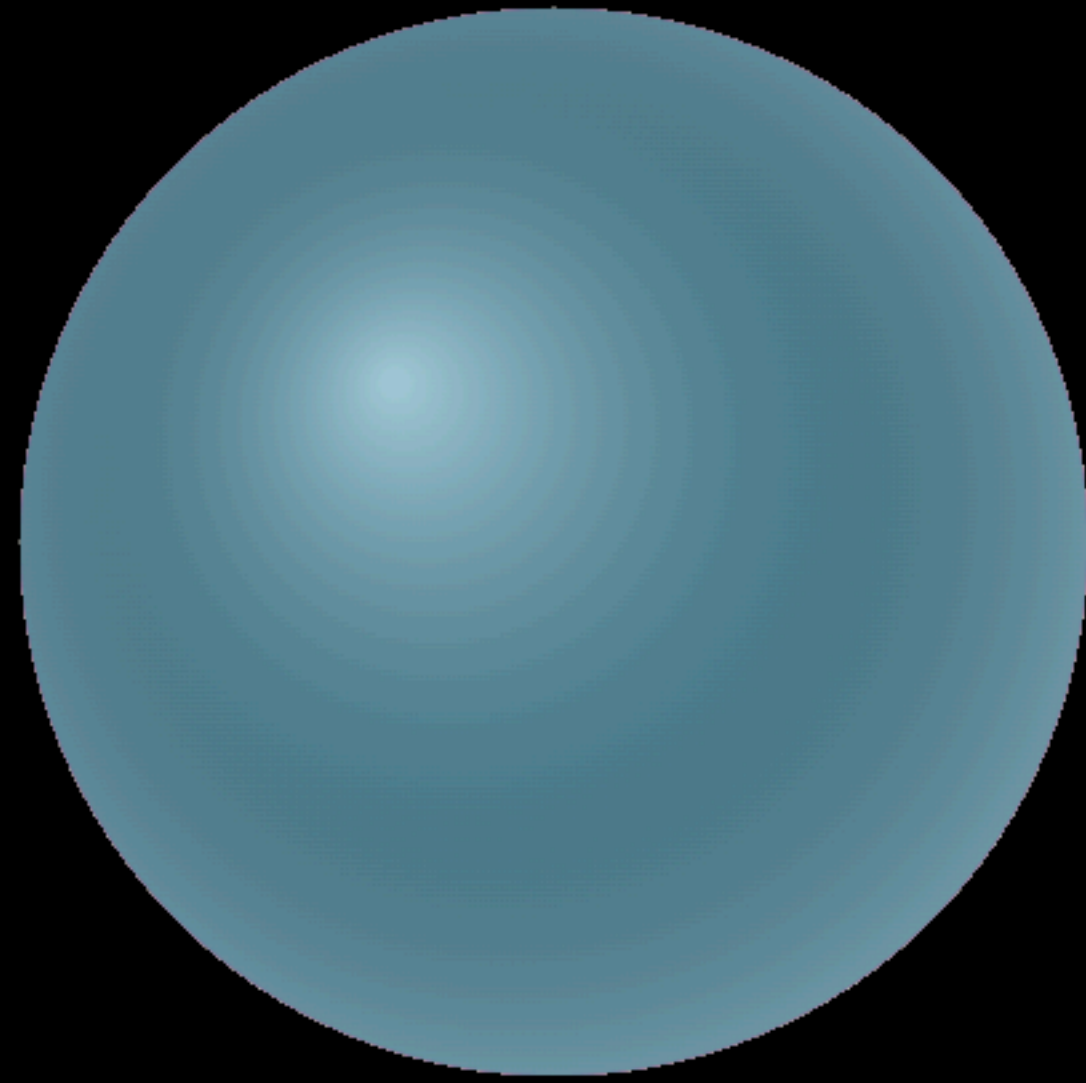
**COVERS
VERSION
4**

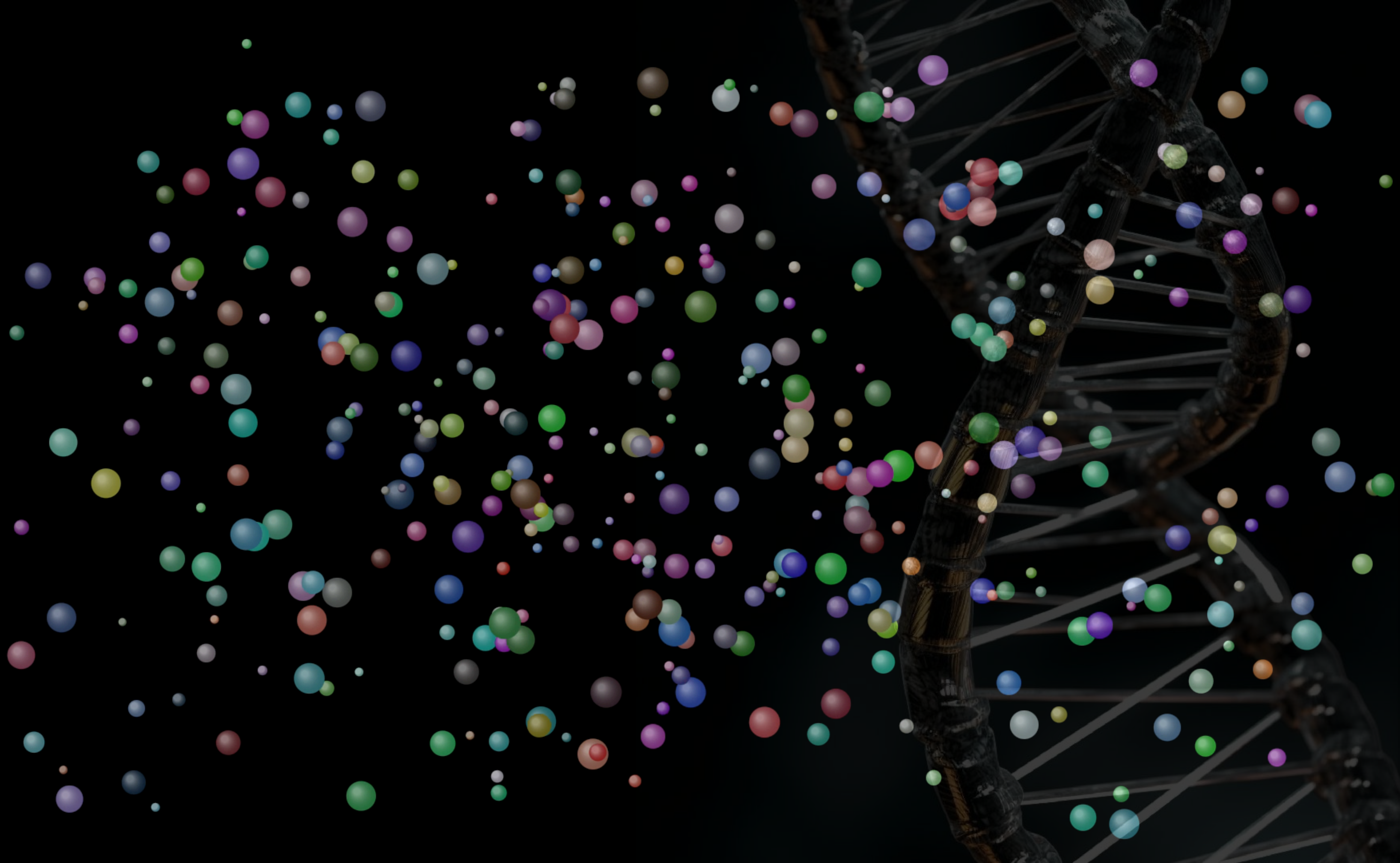
*From simple buttons to interactive movies,
you'll be designing high-impact Web pages in a Flash!*

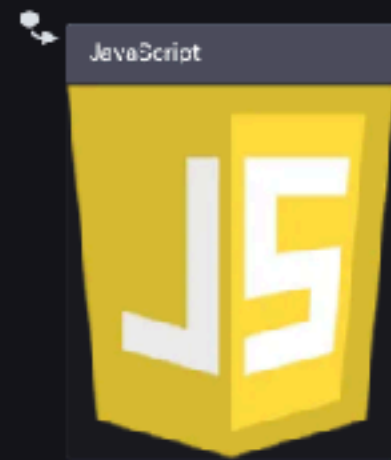
Derek Franklin and Brooks Patton



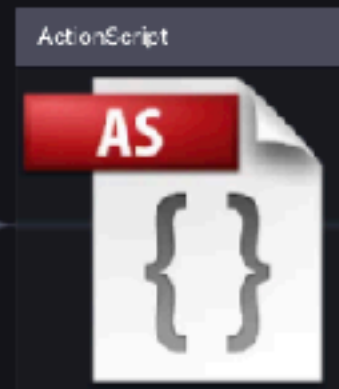
Windows and Macintosh







2000



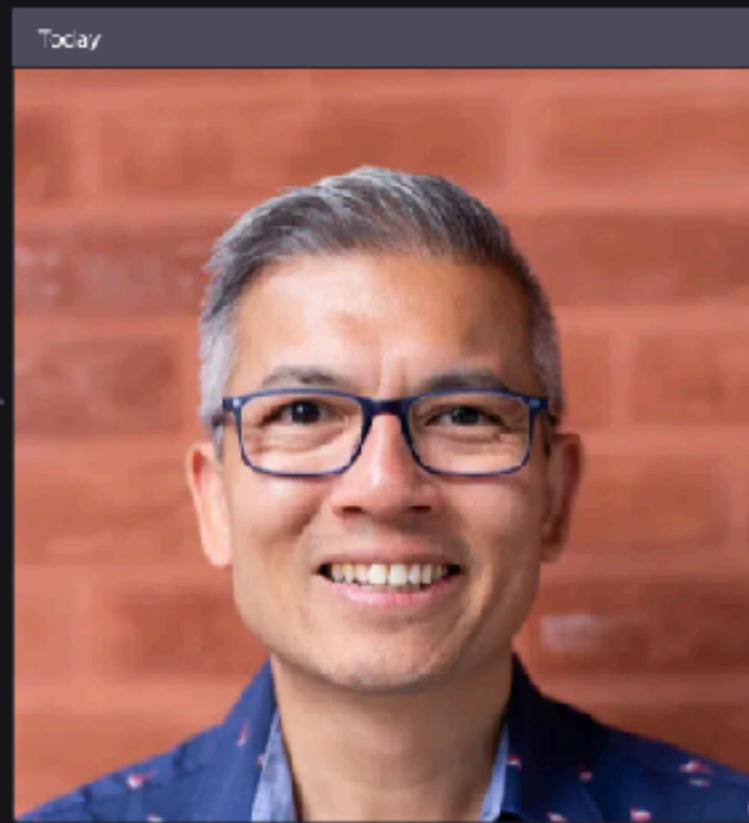
2010



2015

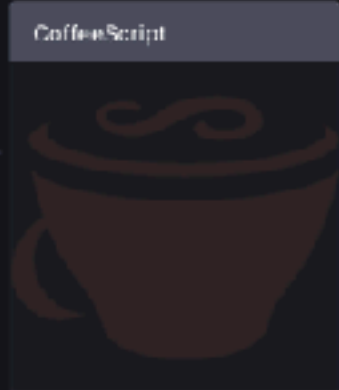


2017



Kevin Maes

2012



2015



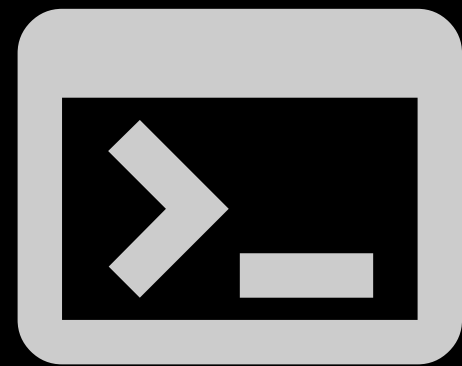
STATELY



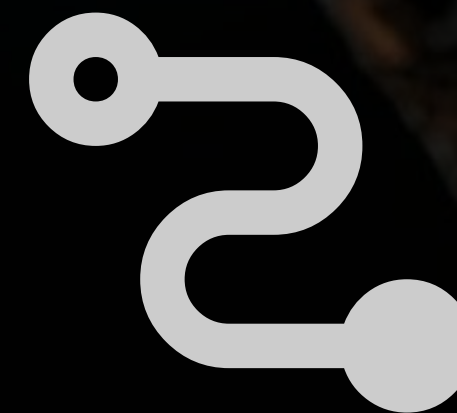
Evolutionary Computation



Lawrence
Fogel



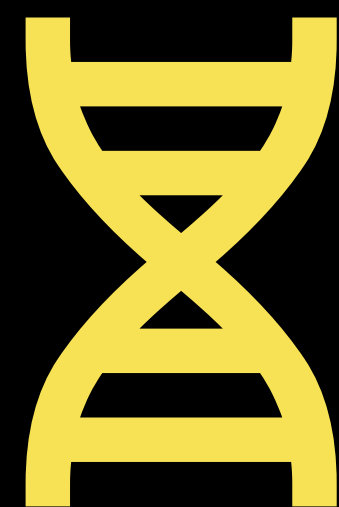
Genetic
Programming (GP)



Evolutionary
Programming (EP)



John H.
Holland



Genetic
Algorithms (GA)



Evolutionary
Strategies (ES)

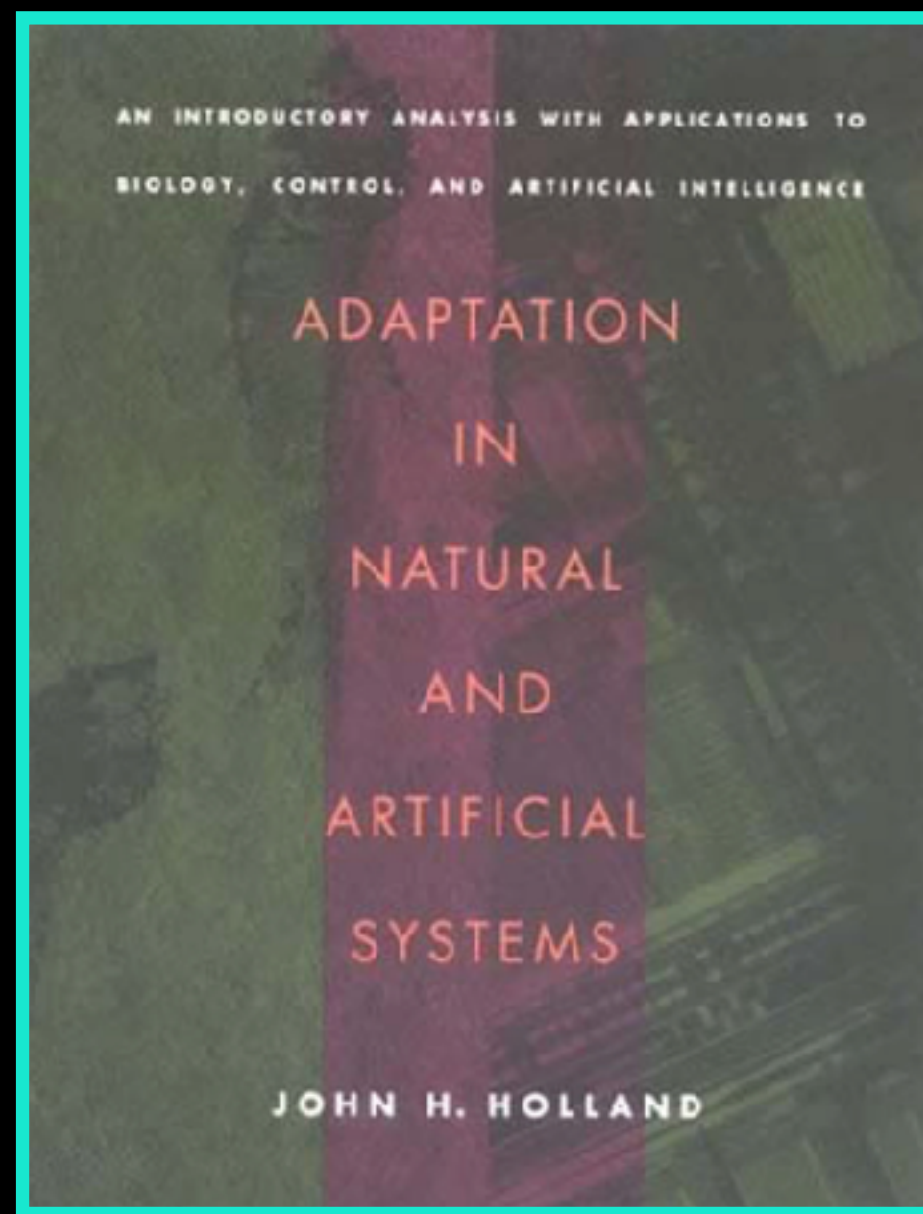
Evolutionary Algorithms

John H. Holland



John H. Holland

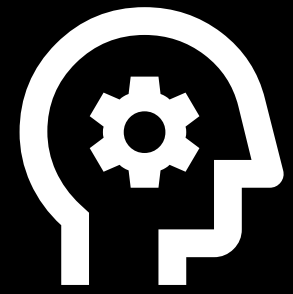
Adaptation in Natural and Artificial Systems



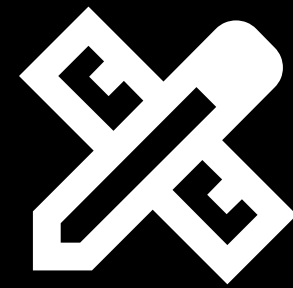
“general theories of adaptive processes apply across biological, cognitive, social, and computational systems”



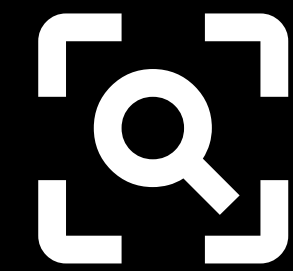
Why genetic algorithms?



Desired behavior



Optimal design



Solve complex search problems

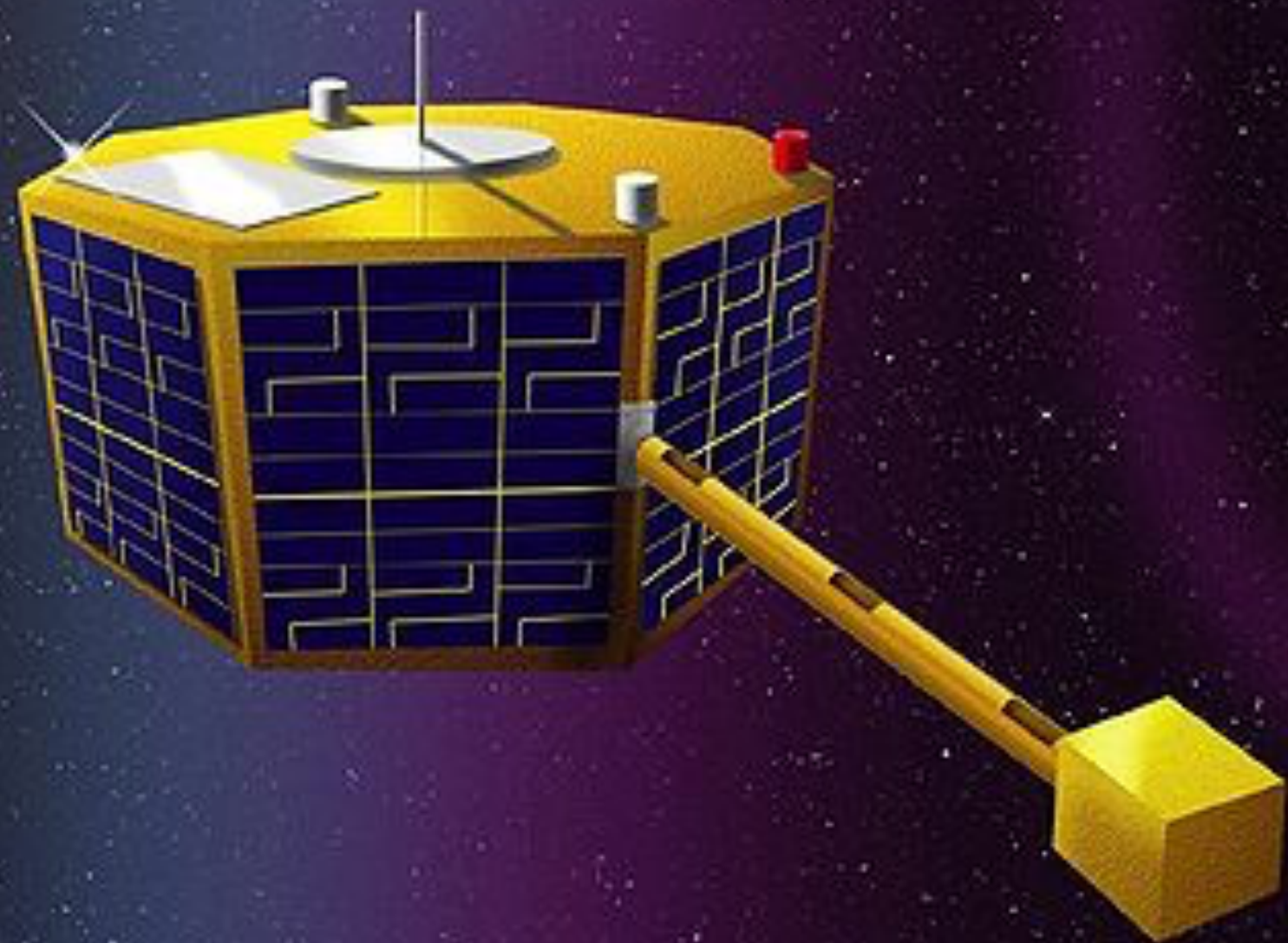


There's a GA for that!

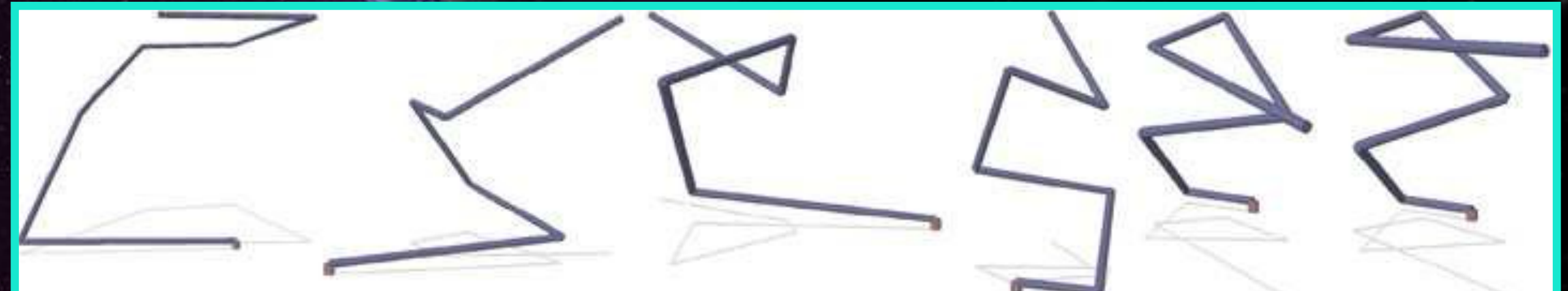
- Designing efficient network topologies
- Automated software testing
- Evolving game strategies
- Scheduling Problems
- Complex hardware design



NASA



Evolved antenna

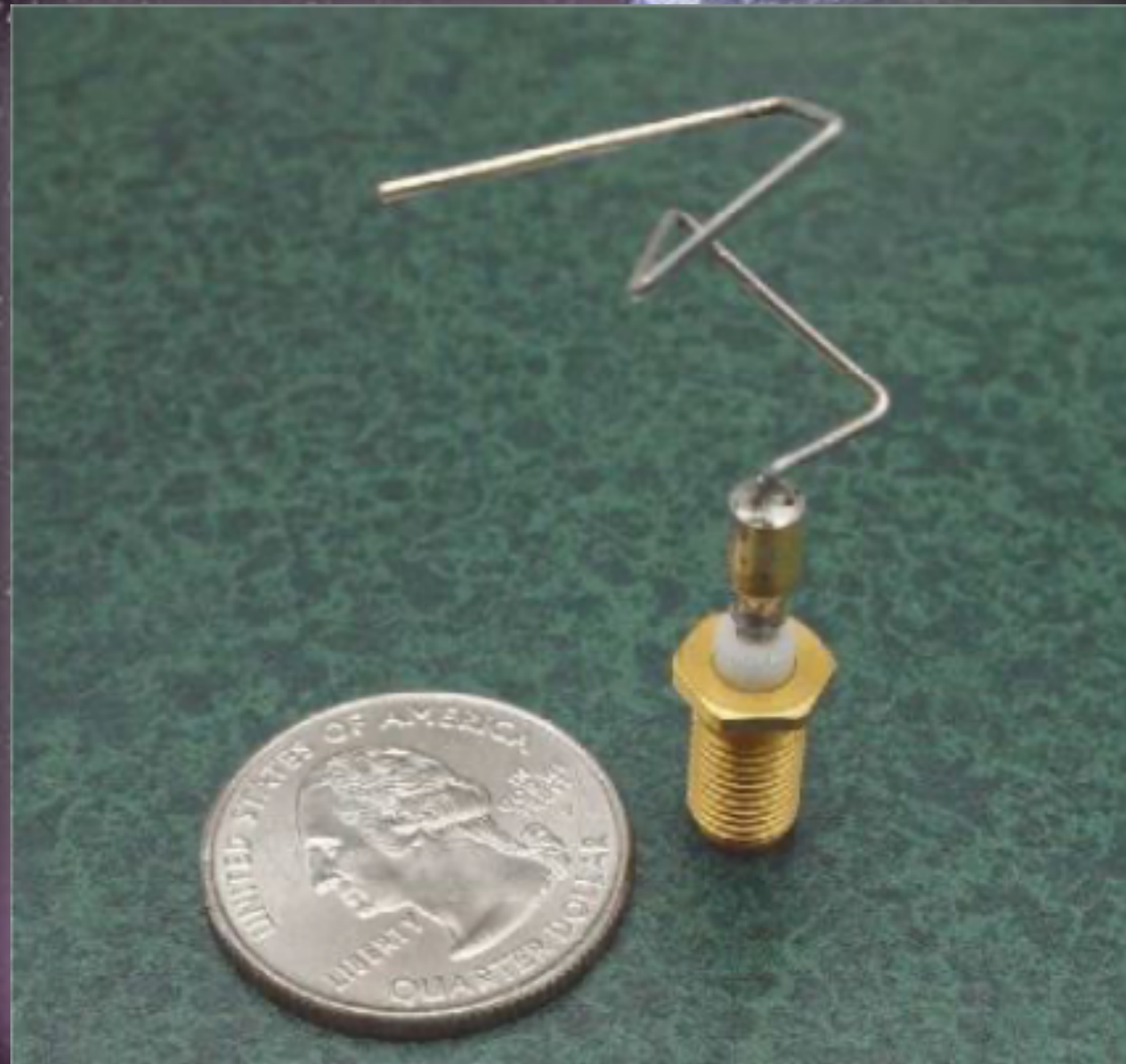
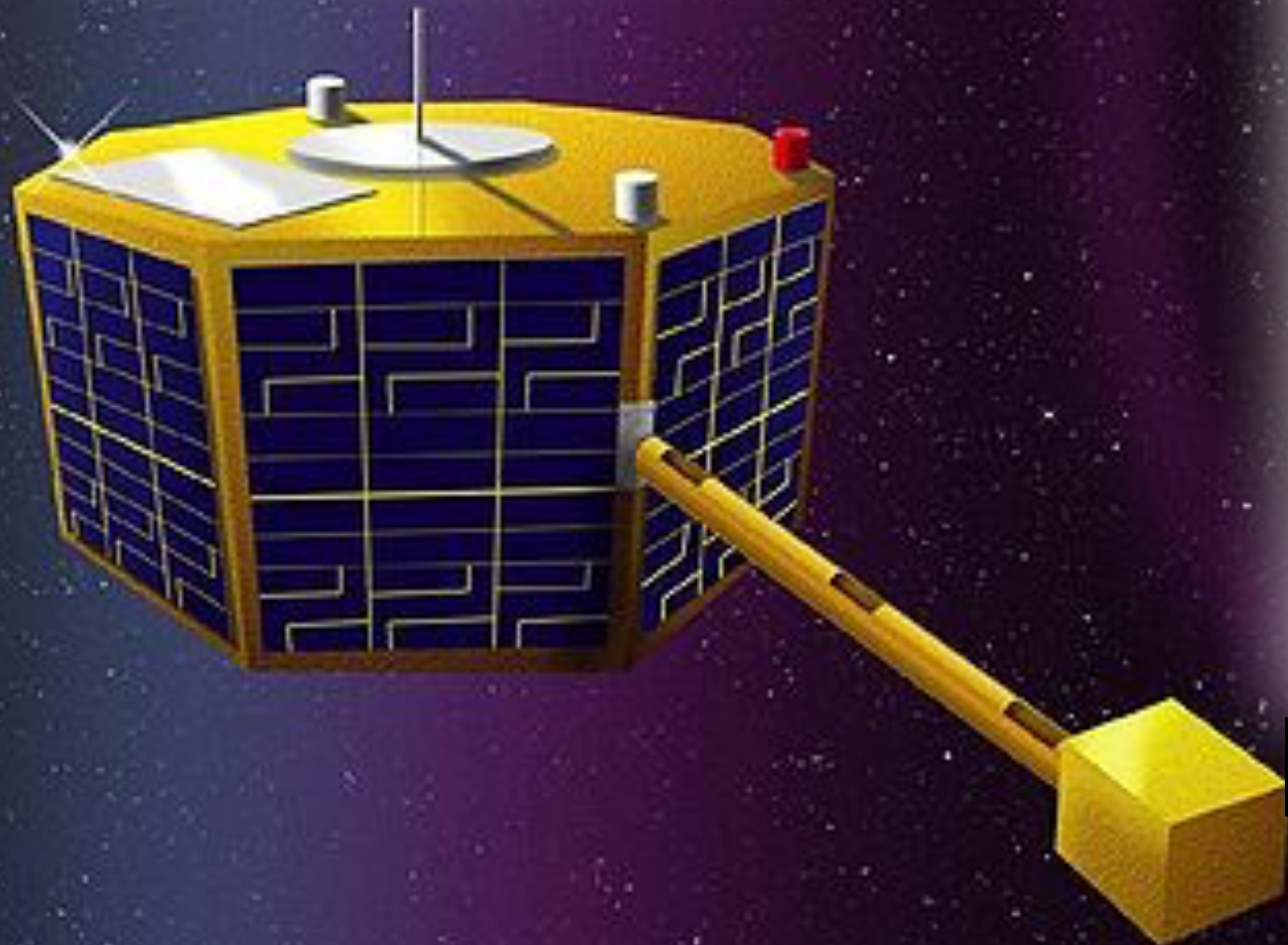


First generation

Middle generations

Last generation

NASA

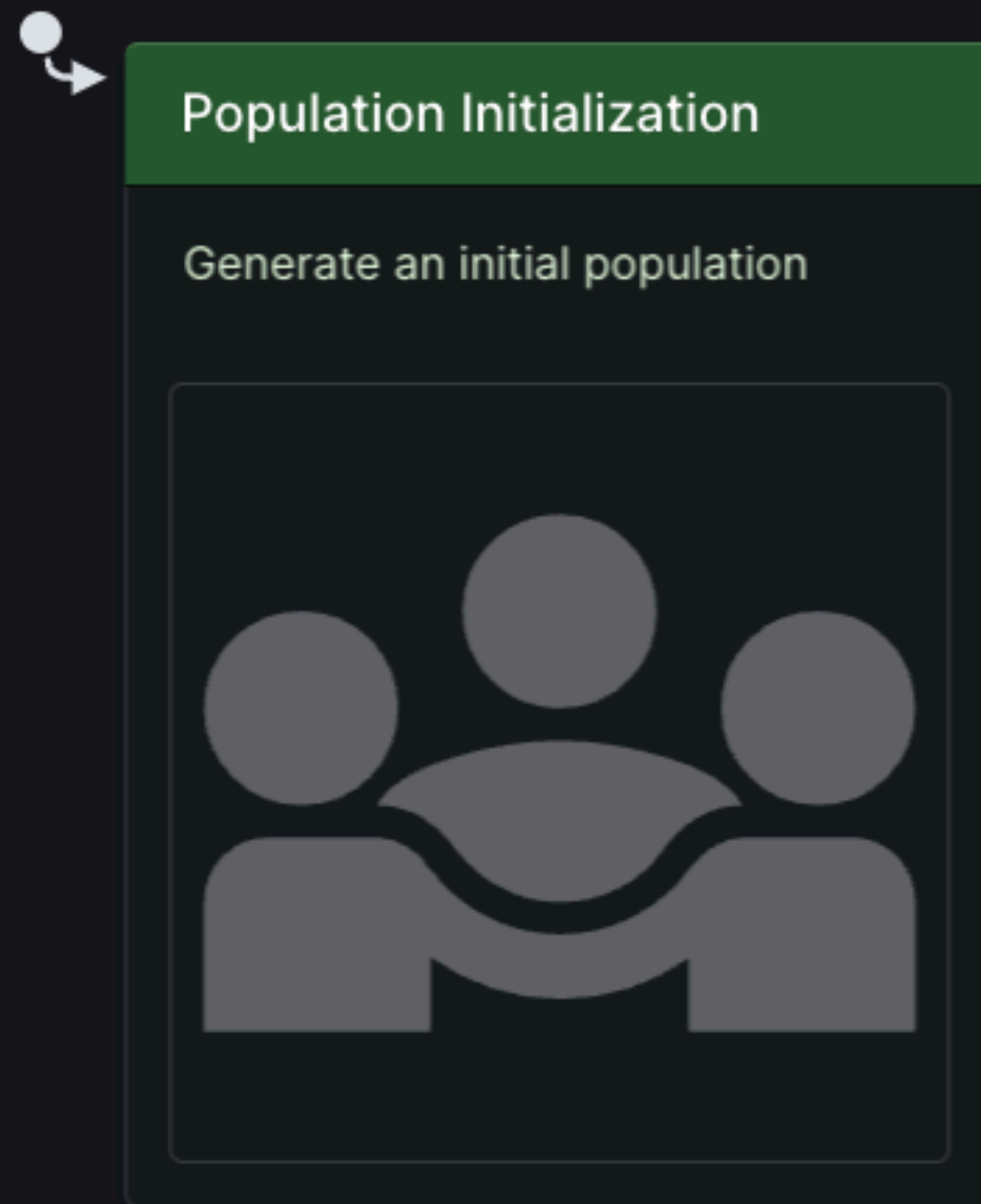


ST5-33-142-7

States of a genetic algorithm



Population Initialization



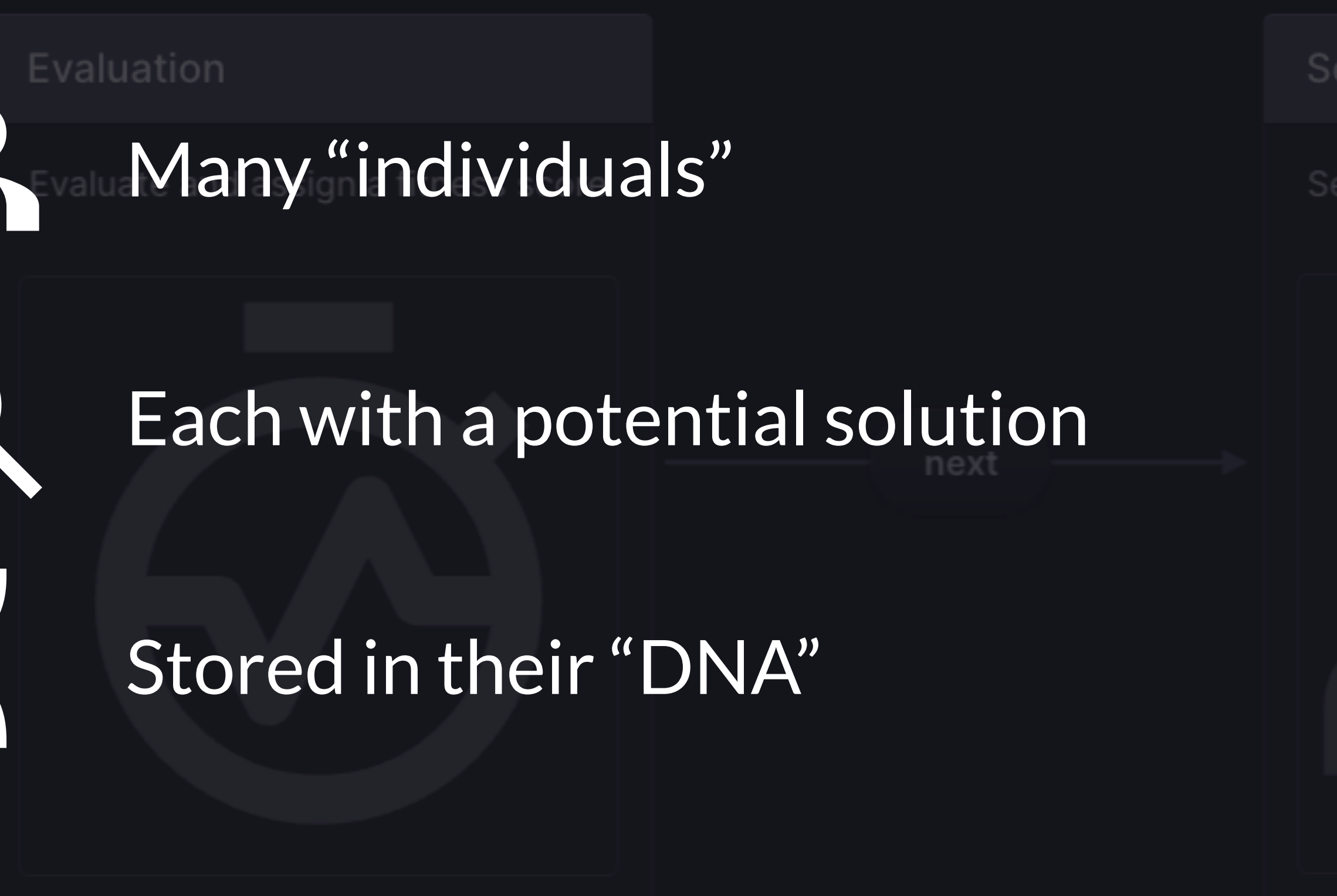
Many "individuals"



Each with a potential solution



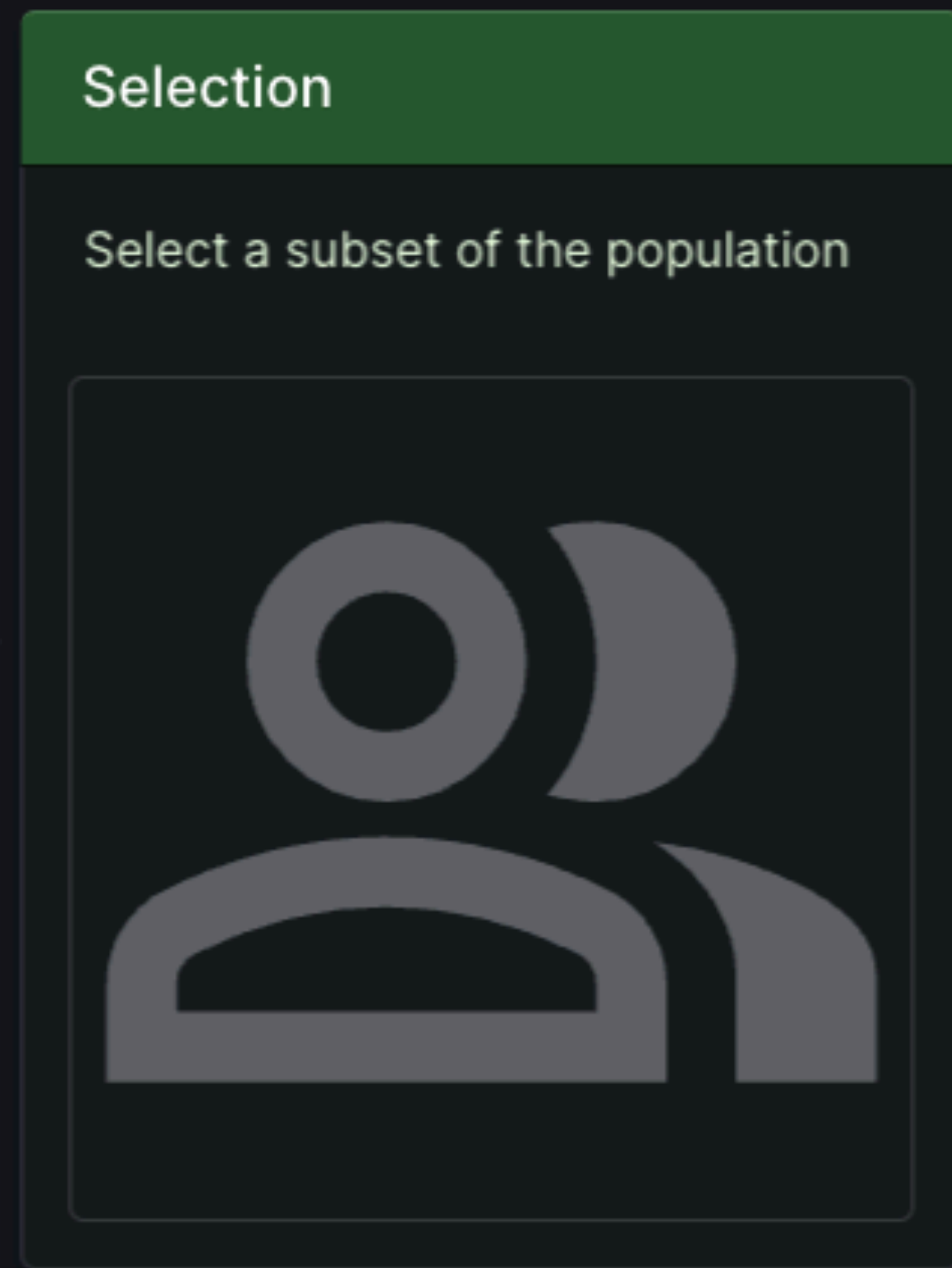
Stored in their "DNA"



Evaluation & Fitness



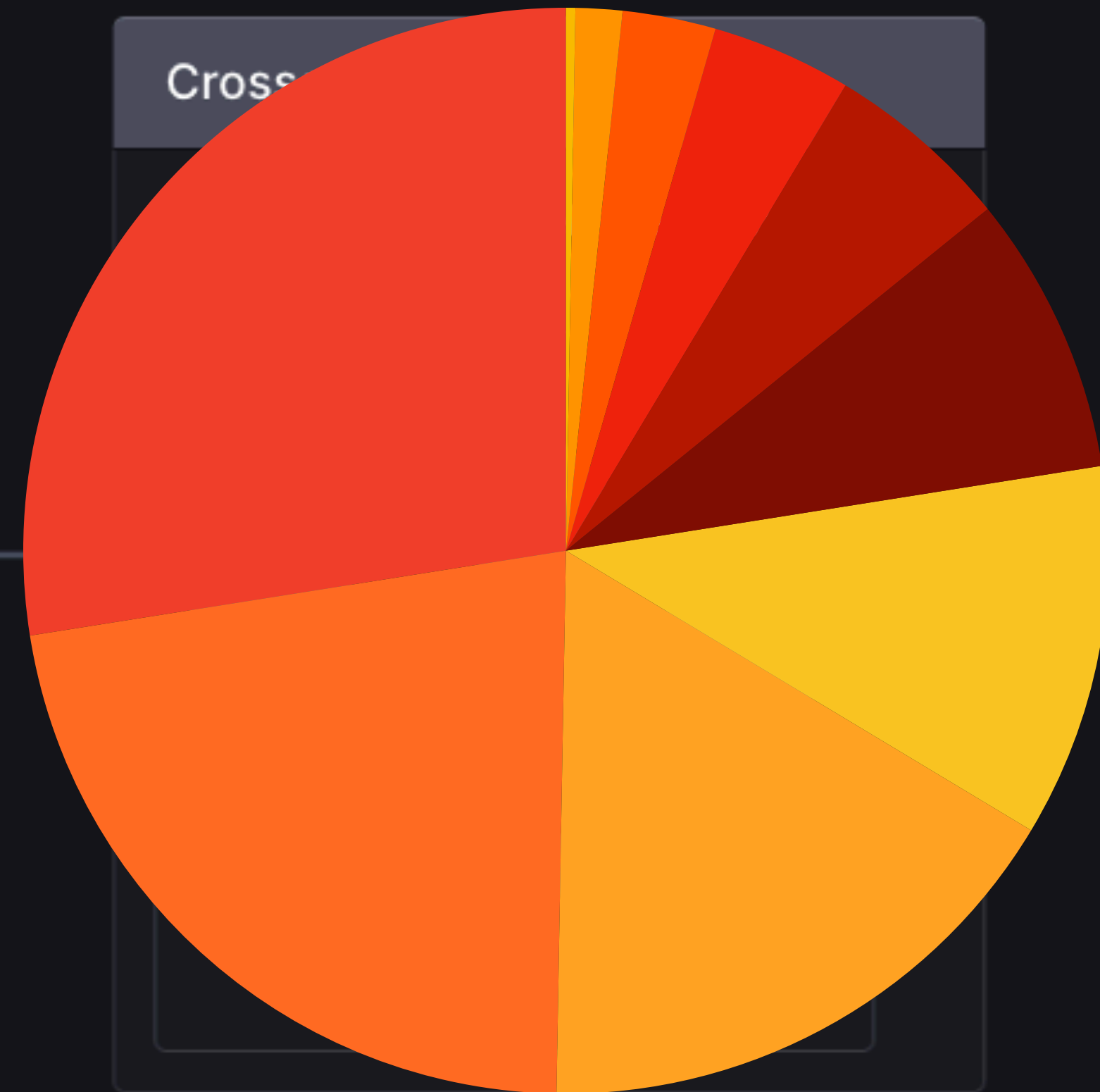
Selection



next

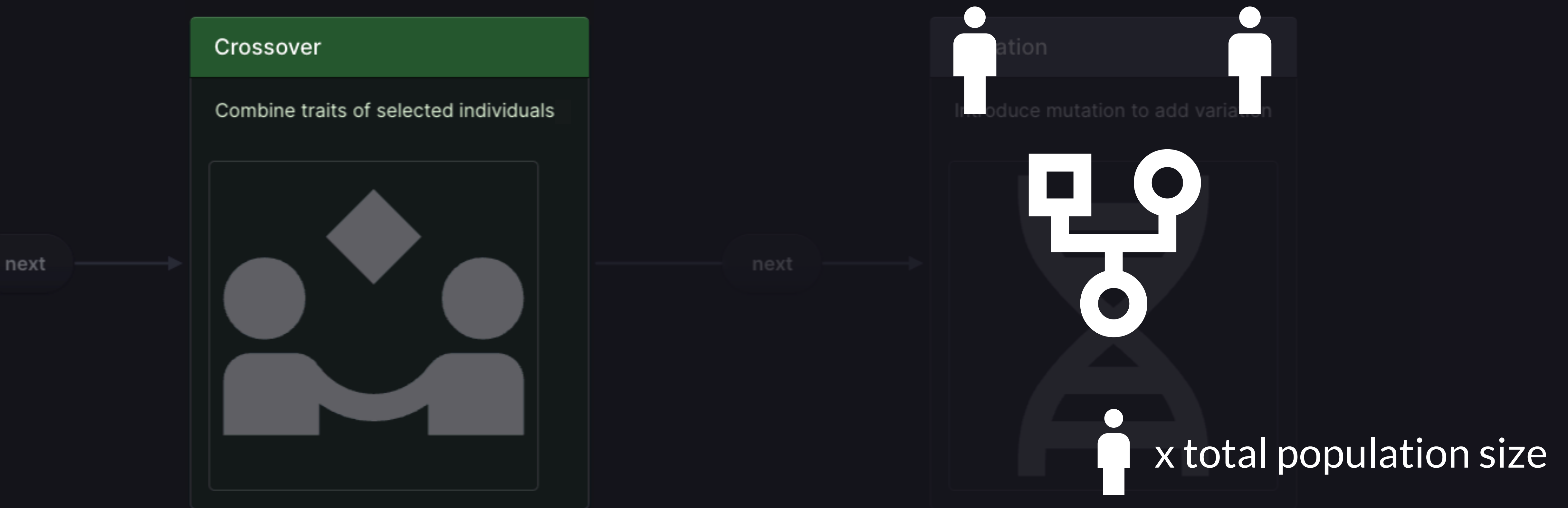
next

Roulette Wheel Selection

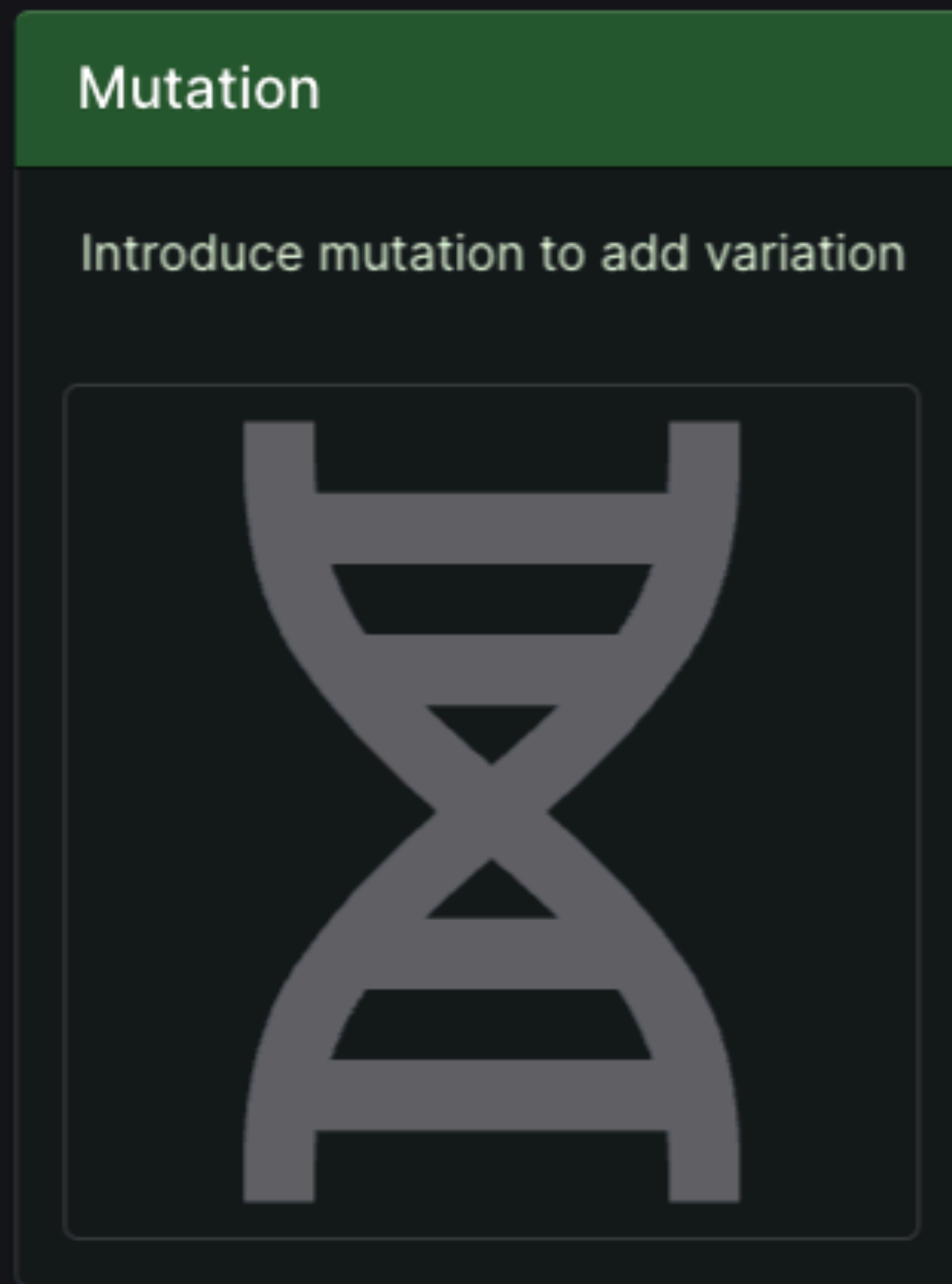


next

Crossover



Mutation



- Need to maintain variation
- Avoid “local optima”
- Strive for the “global optimum”
- Mutation rate
- Mutation amount

Canvas, easy as 1, 2, 3.

1.

```
// index.html  
<canvas id="canvas" width="500" height="500"></canvas>
```
2.

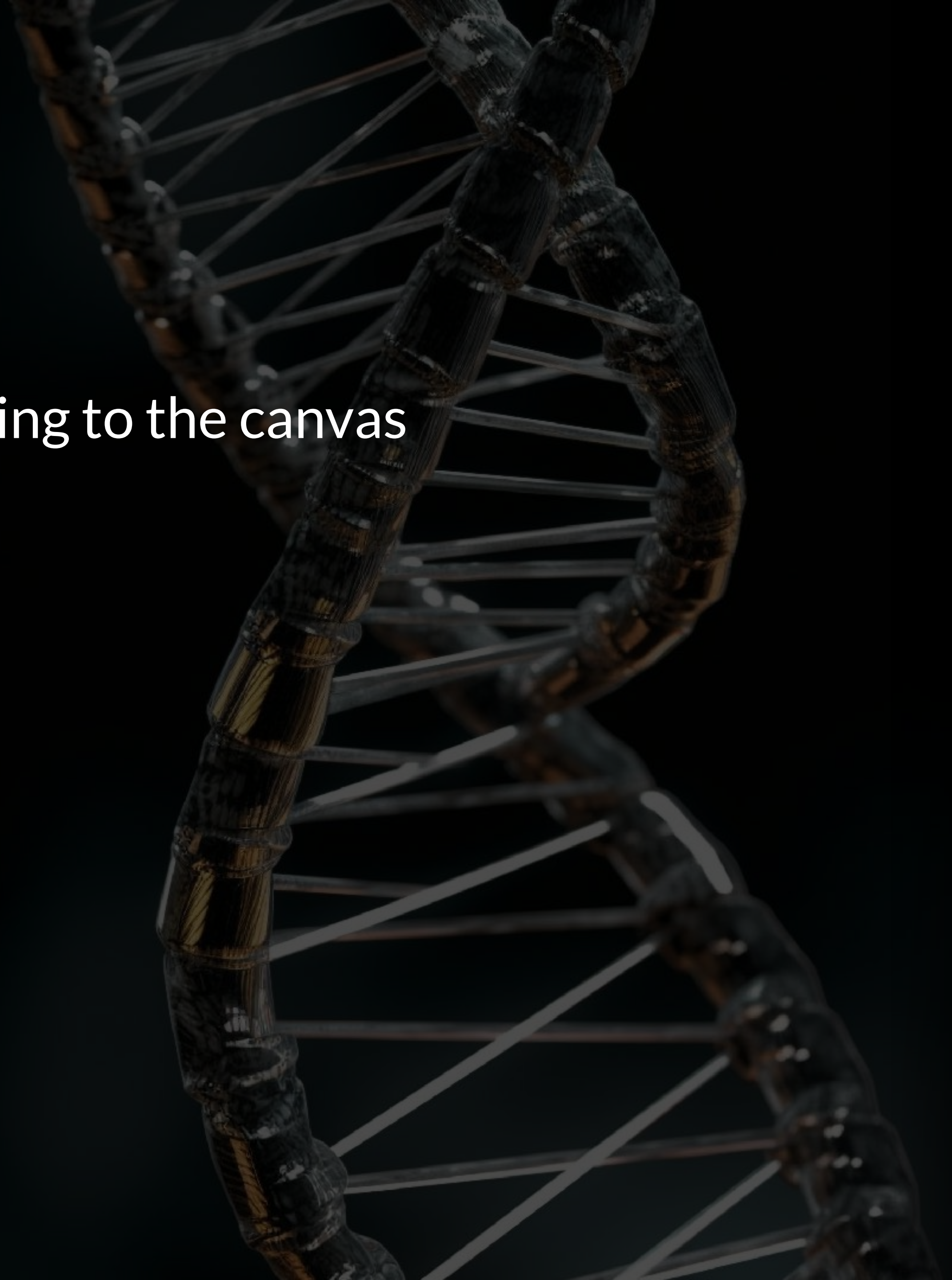
```
// Access the canvas element  
const canvas = document.getElementById('canvas');
```
3.

```
// Get the 2d canvas context  
const ctx = canvas.getContext('2d');
```

See [Canvas API docs](#) for how to draw shapes, text, styles, etc.

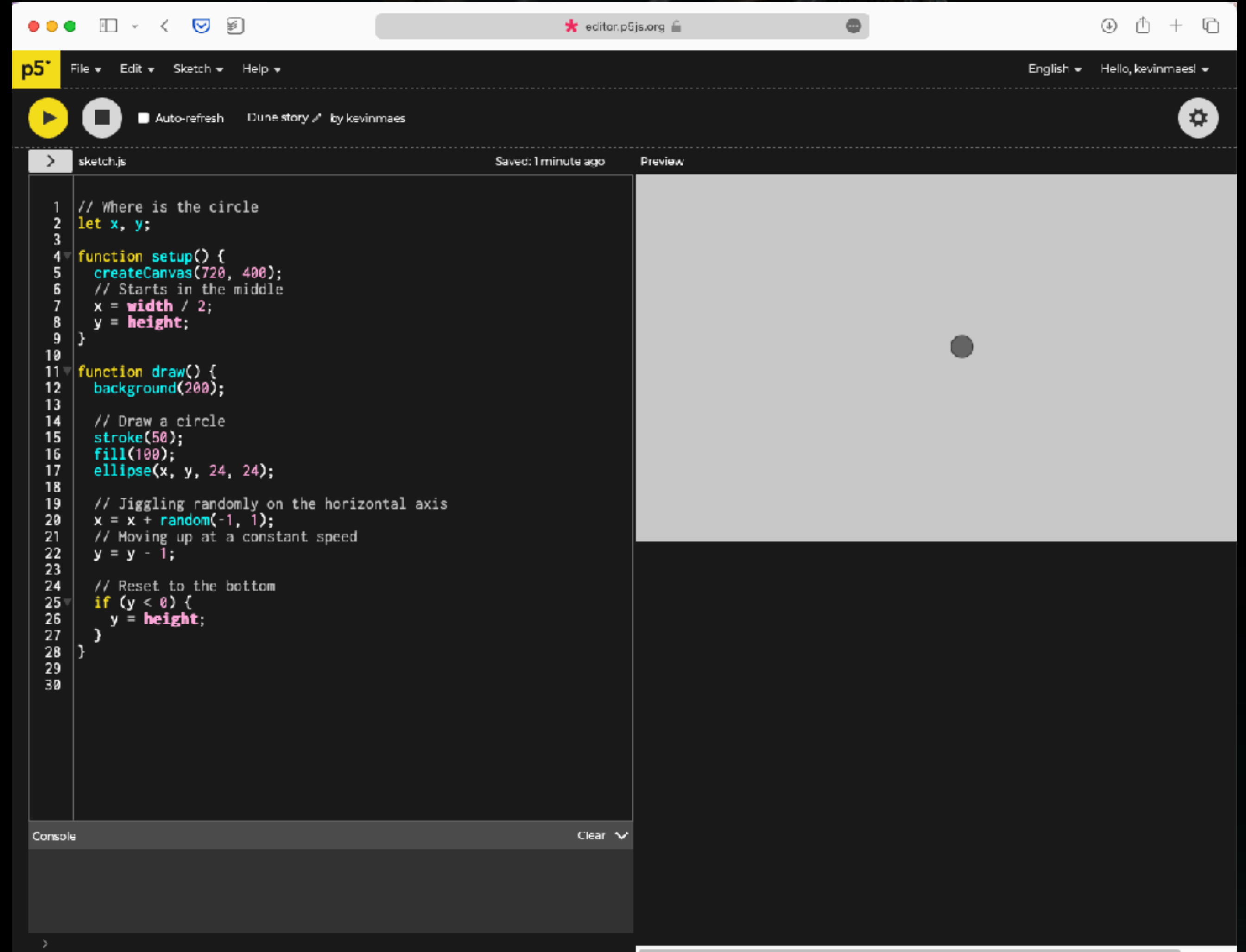
Creative Coding Tips

- Don't forget to clear the canvas before drawing to the canvas
- Always use `requestAnimationFrame`
- Crank up the frame rate and be impressed
- Consider OOP for particle systems



p5.js

- Simplified syntax
- Built-in animation loop
- Event handling
- Extensive library
- Community resources
- Playground environment



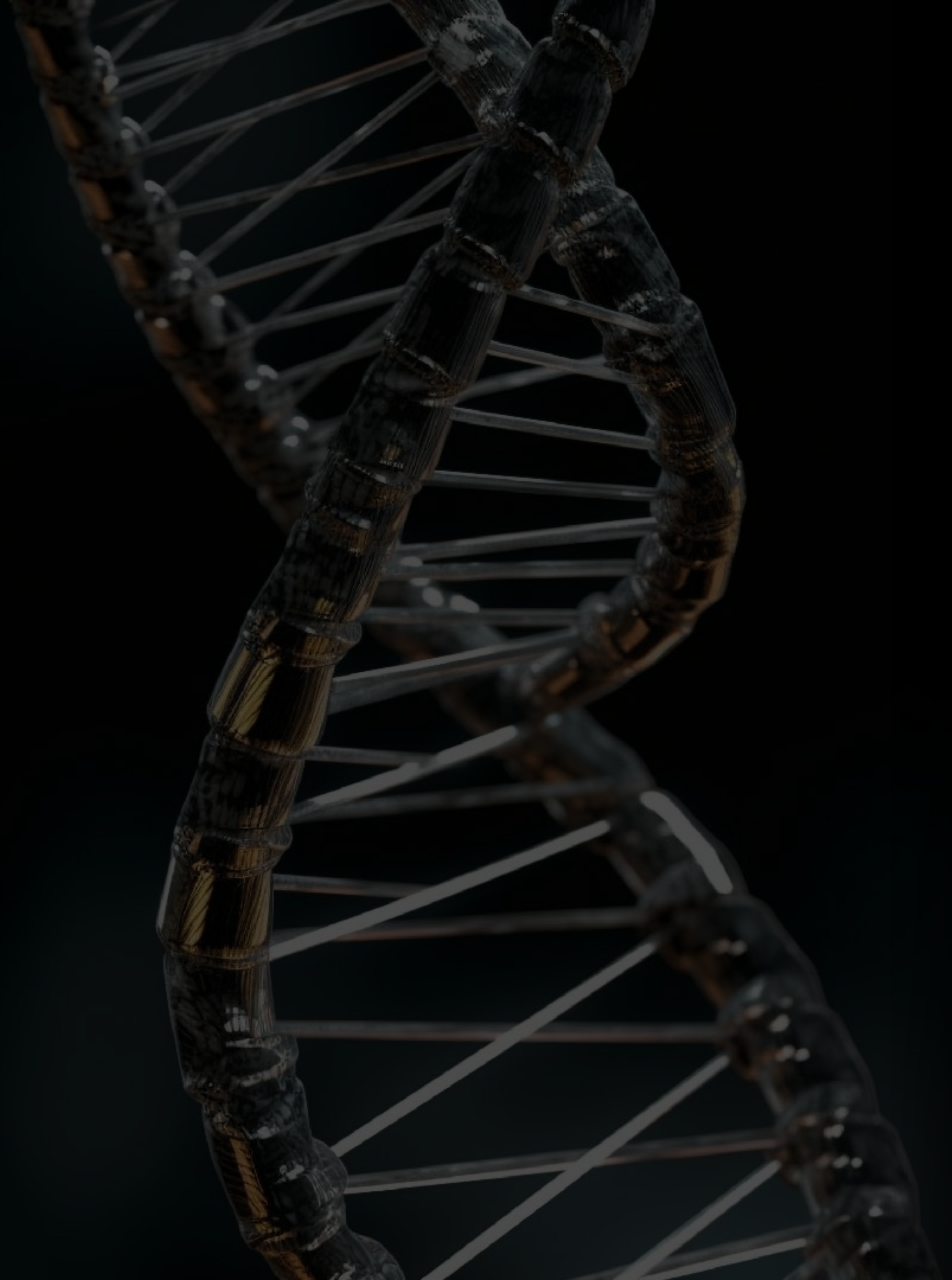
The screenshot shows the p5.js editor interface. The top navigation bar includes the p5.js logo, a menu (File, Edit, Sketch, Help), and user information (English, Hello, kevinmaes). Below the navigation bar, there are icons for play, stop, and auto-refresh, along with the text "Dune story by kevinmaes". The main area is split into two panes: a code editor on the left and a preview window on the right. The code editor shows the following JavaScript code:

```
1 // Where is the circle
2 let x, y;
3
4 function setup() {
5   createCanvas(720, 400);
6   // Starts in the middle
7   x = width / 2;
8   y = height;
9 }
10
11 function draw() {
12   background(200);
13
14   // Draw a circle
15   stroke(50);
16   fill(100);
17   ellipse(x, y, 24, 24);
18
19   // Jiggling randomly on the horizontal axis
20   x = x + random(-1, 1);
21   // Moving up at a constant speed
22   y = y - 1;
23
24   // Reset to the bottom
25   if (y < 0) {
26     y = height;
27   }
28 }
29
30
```

The preview window shows a dark gray background with a small, light gray circle in the center. The bottom of the interface features a console area with a "Clear" button.

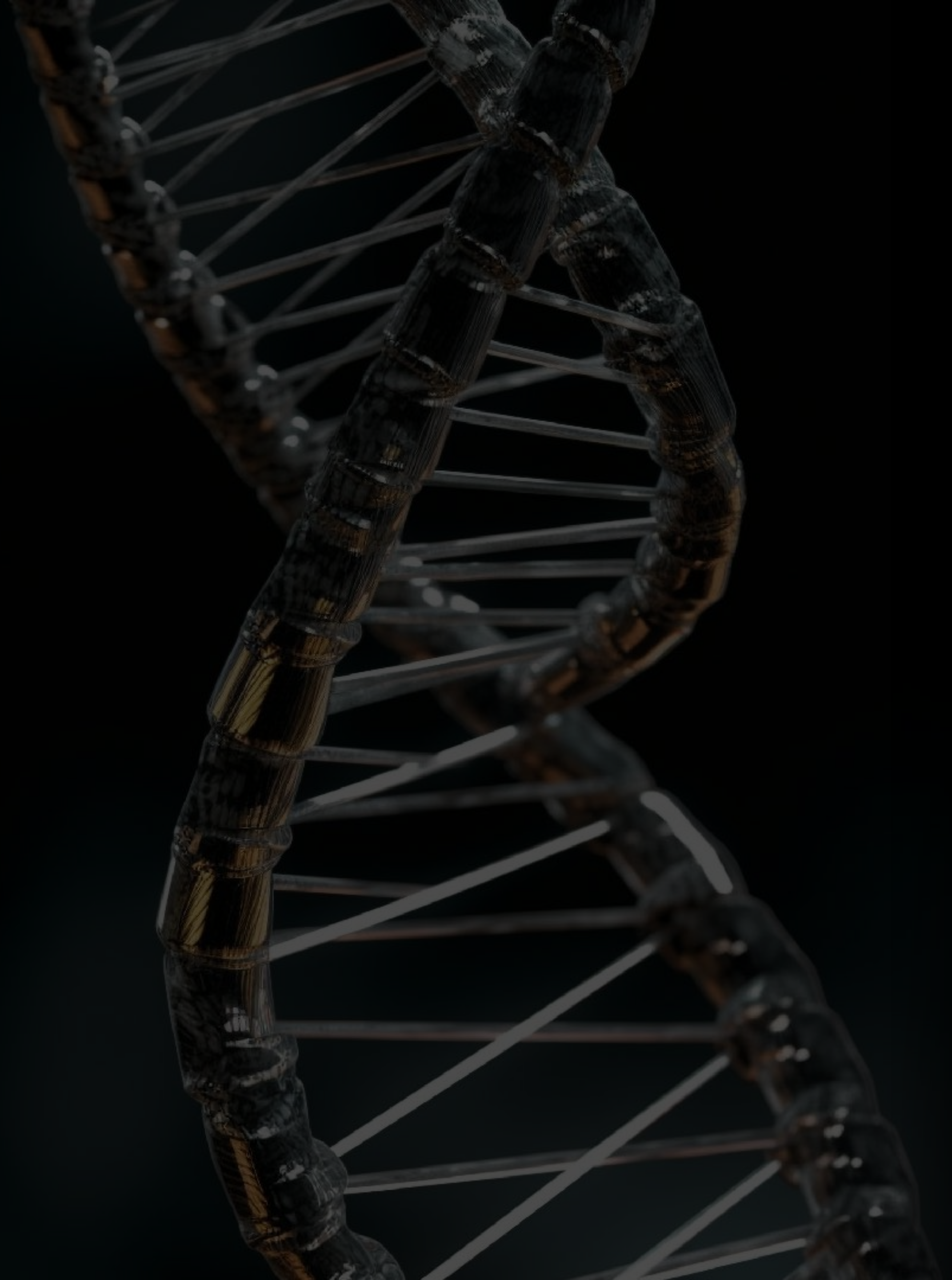
Things I've learned

- Population size
- Tweak the fitness function
- Try other selection methods
- Adjust mutation rates and amounts



Can GAs help us?

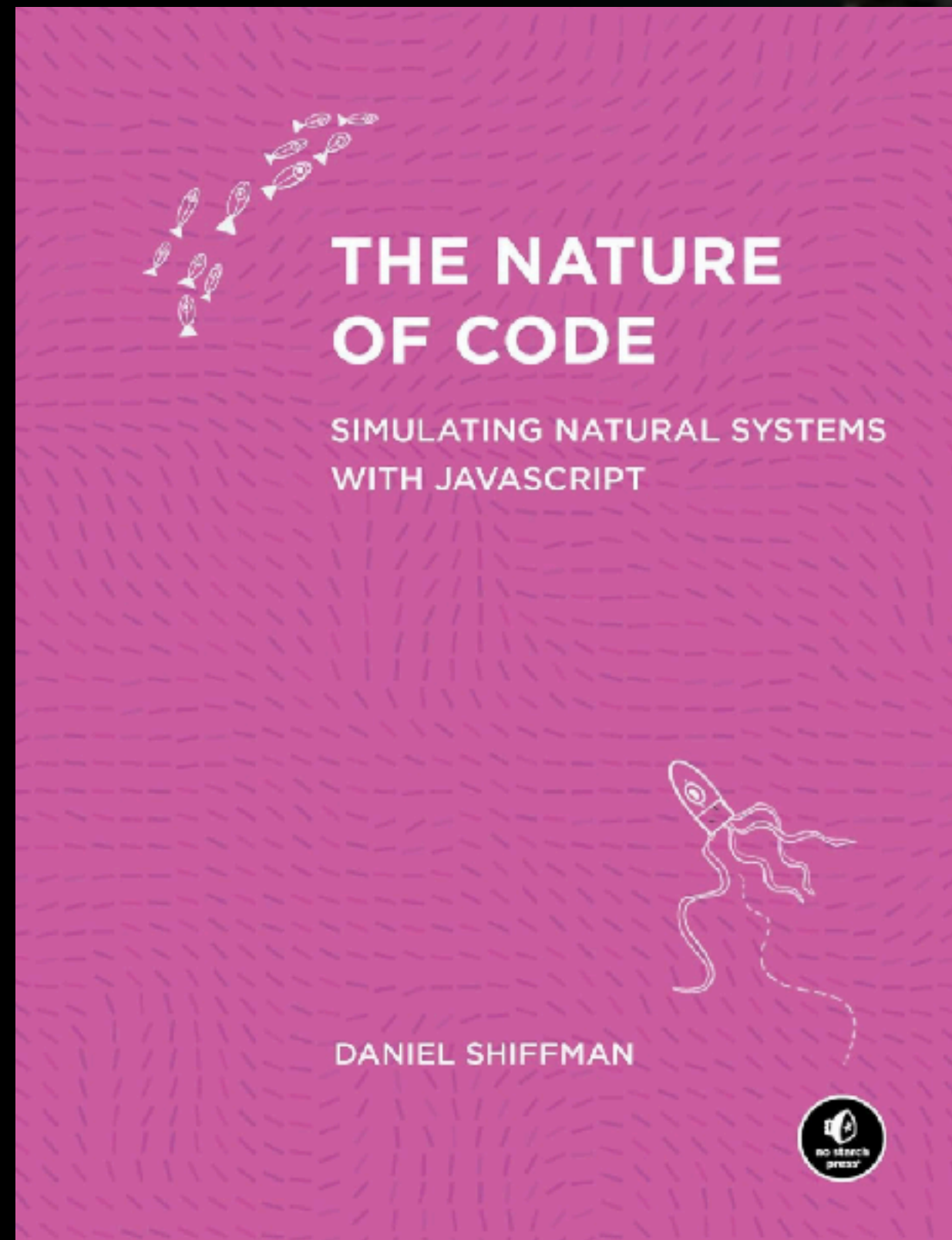
- Optimizing UI Layouts
- Automating A/B Testing
- Personalized content recommendations
- Code optimization or readability
- Dynamic pricing models
- Improve website accessibility



Resources

Daniel Shiffman

thecodingtrain.com/



New JavaScript version
coming September 2024!

**EDGE
OF
TOMORROW**



LIVE

DIE

REPEAT

AND

REPEAT



Thank you!