Sony Computer Entertainment US Research & Development http://www.research.scea.com

Goal Oriented Texture Synthesis

Craig Reynolds

evoStar
Torino, Italy
April 29, 2011

License / copyright notice regarding photos in this presentation: **no derivative works**

Many talented photographers have kindly given me permission to use their images in this presentation. While you are welcome to share the entire presentation, you are prohibited from reusing or republishing any of the photos without explicit permission of the individual photographer. Contact me if you seek such permission.

SONY
COMPUTER ENTERTAINMENT

SCE US R&D

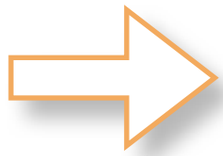
texture synthesis

+

optimization

+

goal



goal-oriented texture synthesis

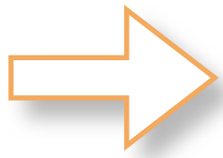
library for procedural texture synthesis

+

genetic programming

+

fitness



goal-oriented texture synthesis

Goal-oriented texture synthesis

- Automatically discover novel images via optimization
- Evolve images based on desired visual properties
- The stochastic nature of evolutionary computation provides an analog of artistic variations on a theme
- Unsupervised or interactive (using human vision)
- Examples:
  - Evolution of textures from high level descriptions
  - Modeling the evolution of camouflage in nature

evolutionary
texture synthesis

unsupervised
procedural fitness

supervised
interactive fitness
based on human vision

Larger perspective of this work

- Related to the concept of evolutionary art
- More specifically aimed at evolving images with objective visual properties
- *Might* be applicable as tool for designers
- Primarily interested in closing the loop between image creation and visual perception

Evolutionary Computation Details

- Genetic programming via Gagné's *Open BEAGLE*
- Uses Montana's strongly-typed genetic programming
- Crossover with "jiggle" mutation of FP constants
- Population 100-200 in 5-10 demes of 20 individuals
- Unsupervised runs:
  - 50 generations, about 5 minutes
  - Textures rendered at 300x300 pixels
  - Fitness examines roughly 10,000 pixels (10%)

Procedural Texture Synthesis

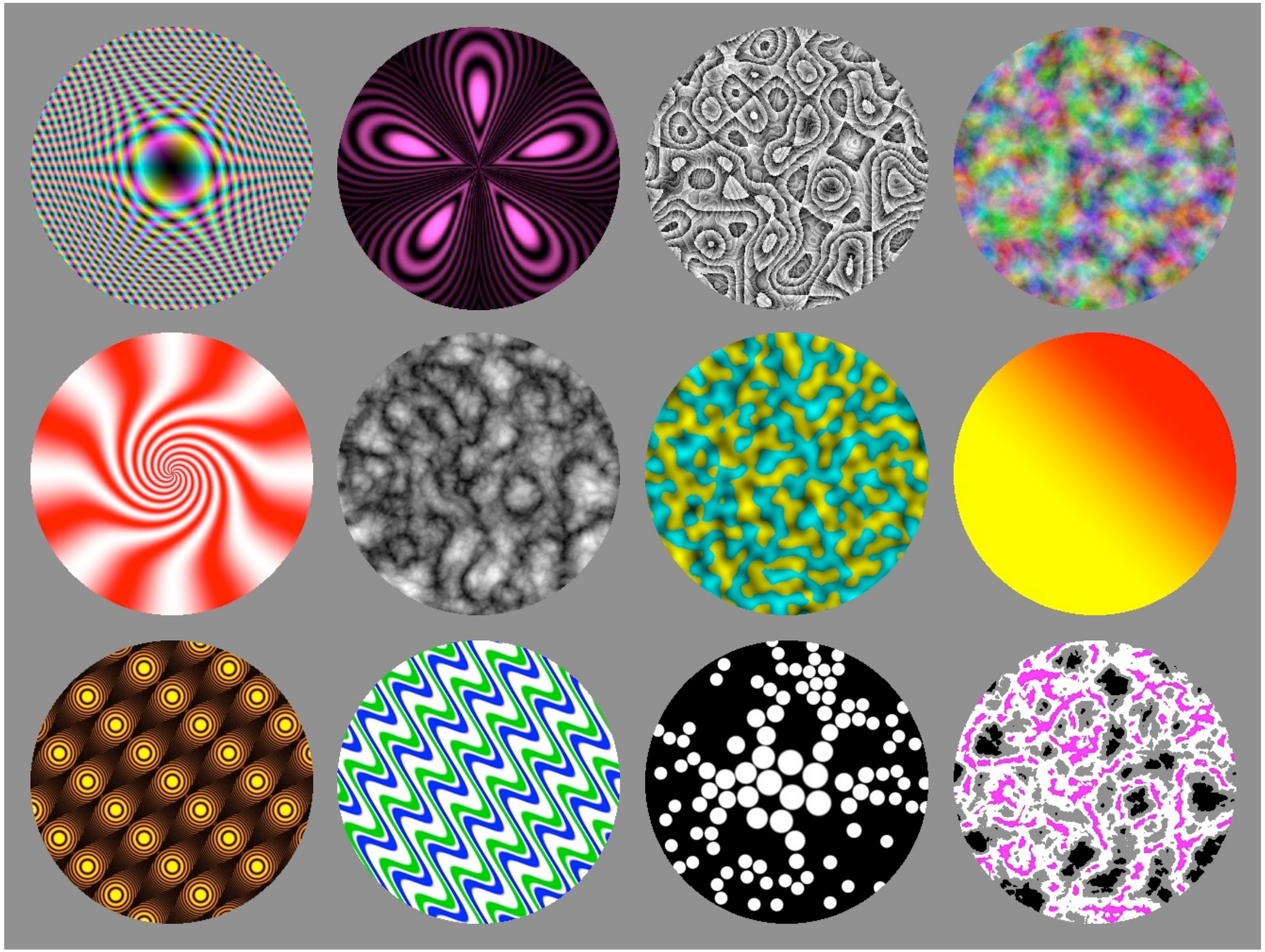
Procedural texture synthesis

- Procedural model: program to make texture
- Library (API) of about 50 texture-valued functions to:
  - generate basic texture patterns
  - operate on one or more textures
- Nested expression of texture functions
- Resolution-independent representation, sample pixel values at continuous (floating point) coordinates

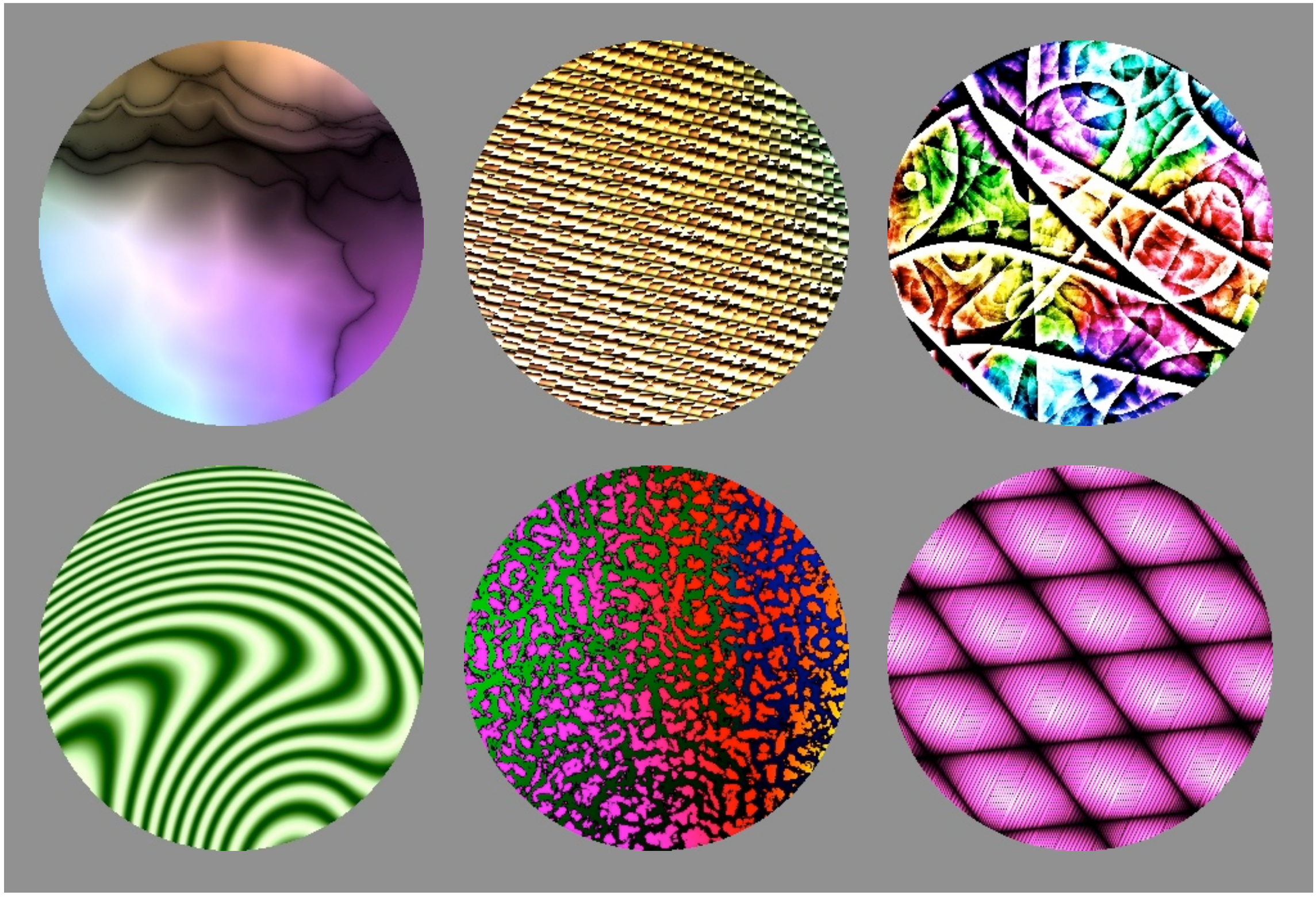
Texture synthesis library

Texture generators: UniformColor, SoftEdgeSpot, Gradation, SineGrating, TriangleWaveGrating, SoftEdgedSquareWaveGrating, RadialGrad, Noise, ColorNoise, Brownian, Turbulence, Furbulence, Wrapulence and NoiseDiffClip.

Texture operators: Scale, Translate, Rotate, Mirror, Add, Subtract, Multiply, Max, Min, SoftMatte, ExpAbsDiff, Row, Array, Invert, Tint, Stretch, StretchSpot, Wrap, Ring, Twist, VortexSpot, Blur, EdgeDetect, EdgeEnhance, SliceGrating, SliceToRadial, SliceShear, Colorize, Gamma, AdjustSaturation, AdjustHue, BrightnessToHue, BrightnessWrap, BrightnessSlice4, HueIfAny, SoftThreshold, SpotsInCircle and ColoredSpotsInCircle



simple hand-written combinations of texture synthesis primitives



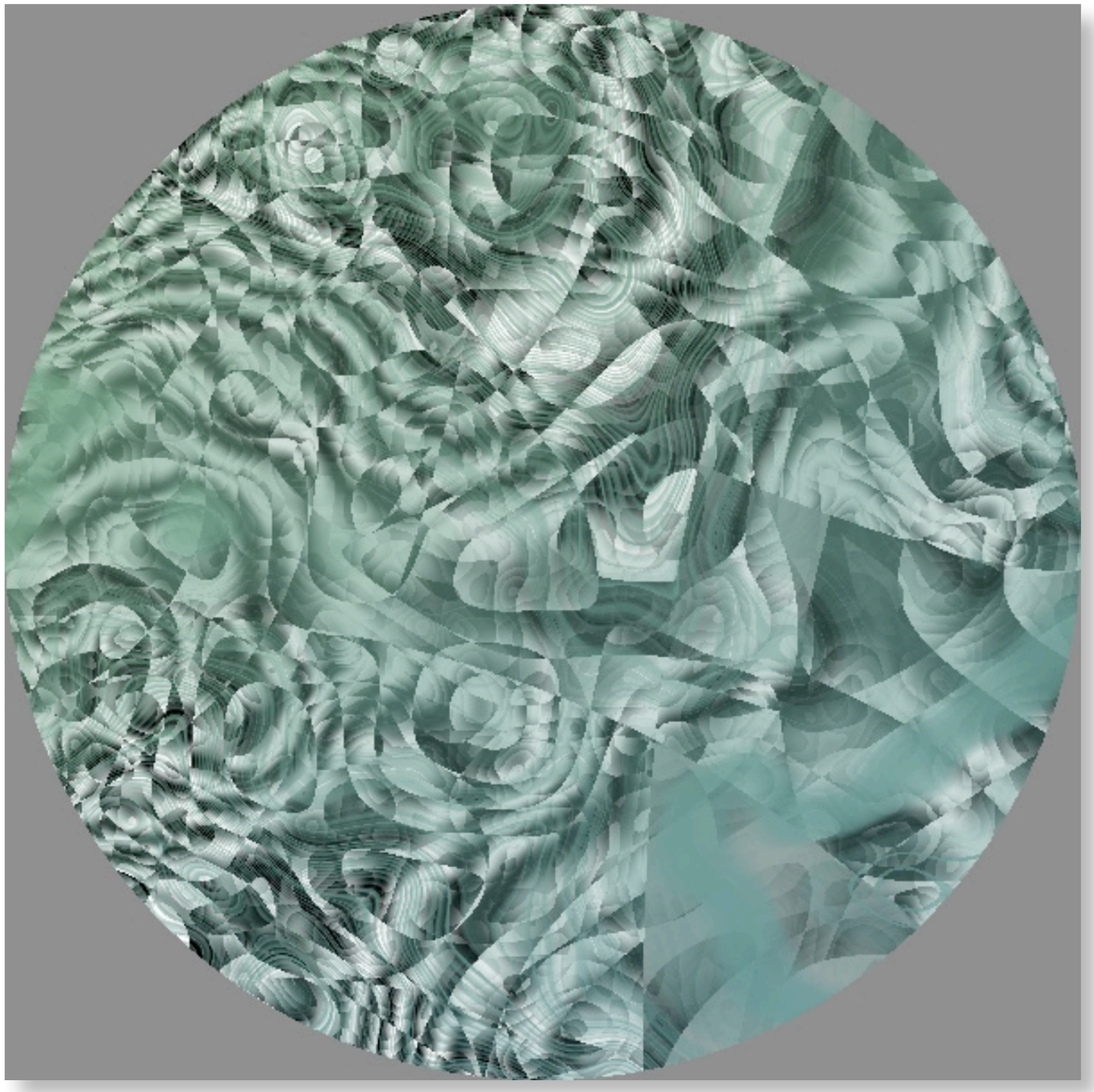
textures evolved with unsupervised GP

Evolved textures with source code

Subtract (EdgeEnhance (0.040705, 4.58566, Add (Wrapulence (2.61481, Vec2 (1.16699, -2.27901)), Furbulence (3.66211, Vec2 (-2.12694, -1.26397)))), Subtract (Subtract (EdgeEnhance (0.0420333, 4.58566, Add (Subtract (Add (Furbulence (0.323467, Vec2 (-2.12694, 1.10331)), Furbulence (3.66211, Vec2 (-2.12694, 1.10331))), Furbulence (3.66211, Vec2 (-2.12694, 1.10331))), EdgeEnhance (0.042568, 4.58566, Add (Wrapulence (2.61537, Vec2 (-2.94796, -2.94796)), Furbulence (3.90532, Vec2 (-2.94796, 0.965091)))))), Furbulence (3.66211, Vec2 (-2.12694, 1.10331))), HueOnly (Subtract (EdgeEnhance (0.042568, 4.58566, Add (Wrapulence (2.84277, Vec2 (-2.94796, 0.965091)), Wrapulence (2.84277, Vec2 (-2.94796, 0.740041)))), Subtract (Add (Wrapulence (3.05225, Vec2 (1.16699, -2.27901)), Furbulence (3.66211, Vec2 (-2.12694, 1.10331))), ColorNoise (0.5612, Vec2 (1.44605, -2.03616)))))))

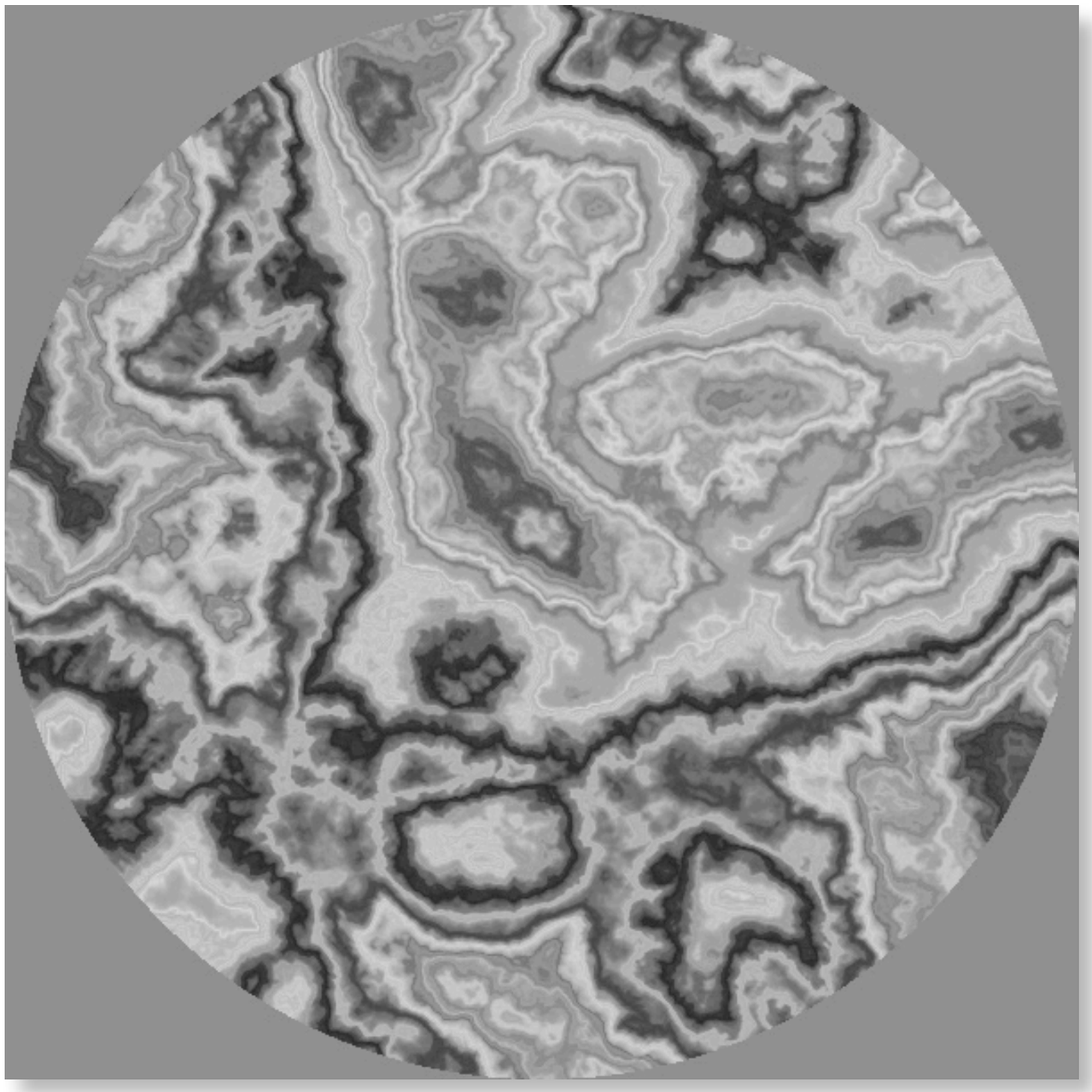
texture evolved from high-level description

Invert (SoftMatte (HuelfAny (Colorize (Twist (-1.76008, Vec2 (-2.90822, -1.26208), Multiply (Brownian (0.880861, Vec2 (2.80615, 1.14405)), Wrap (6.21909, 5.55726, Vec2 (1.88101, -1.10475), Add (VortexSpot (-2.95874, 4.37424, Vec2 (-2.24113, -0.804409), Row (Vec2 (-1.20827, -0.80333), Wrapulence (5.81646, Vec2 (1.46969, 0.464754)))), Multiply (TriangleWaveGrating (15.0552, 0.251605, 4.92253), Wrap (6.21909, 5.25948, Vec2 (-2.90822, -1.26208), Add (ColoredSpotsInCircle (146.485, 0.573184, 0.103147, Stretch (1.92016, 0.932767, Vec2 (0.994563, 1.8778), SineGrating (17.4233, 0.477075)), Translate (Vec2 (1.3634, -3.05406), Colorize (SineGrating (87.1581, 1.2438), SoftEdgedSquareWaveGrating (138.03, 0.0101831, 0.894823, 1.03307))), SliceToRadial (Vec2 (-1.20827, -0.80333), ColorNoise (1.09284, Vec2 (1.24907, -3.11514)))), Brownian (4.15562, Vec2 (-1.20827, -0.80333))))))))), Brownian (0.880861, Vec2 (2.80615, 1.14405)))), SliceToRadial (Vec2 (-1.20827, -0.80333), ColorNoise (1.09284, Vec2 (1.24907, -3.11514))), Colorize (Twist (-1.90423, Vec2 (0.977825, -0.533419), Twist (-1.90423, Vec2 (0.977825, -0.533419), RadialGrad (195.316, Vec2 (1.24907, -3.11514)))), Wrapulence (5.81646, Vec2 (0.0918581, -0.543768)))))



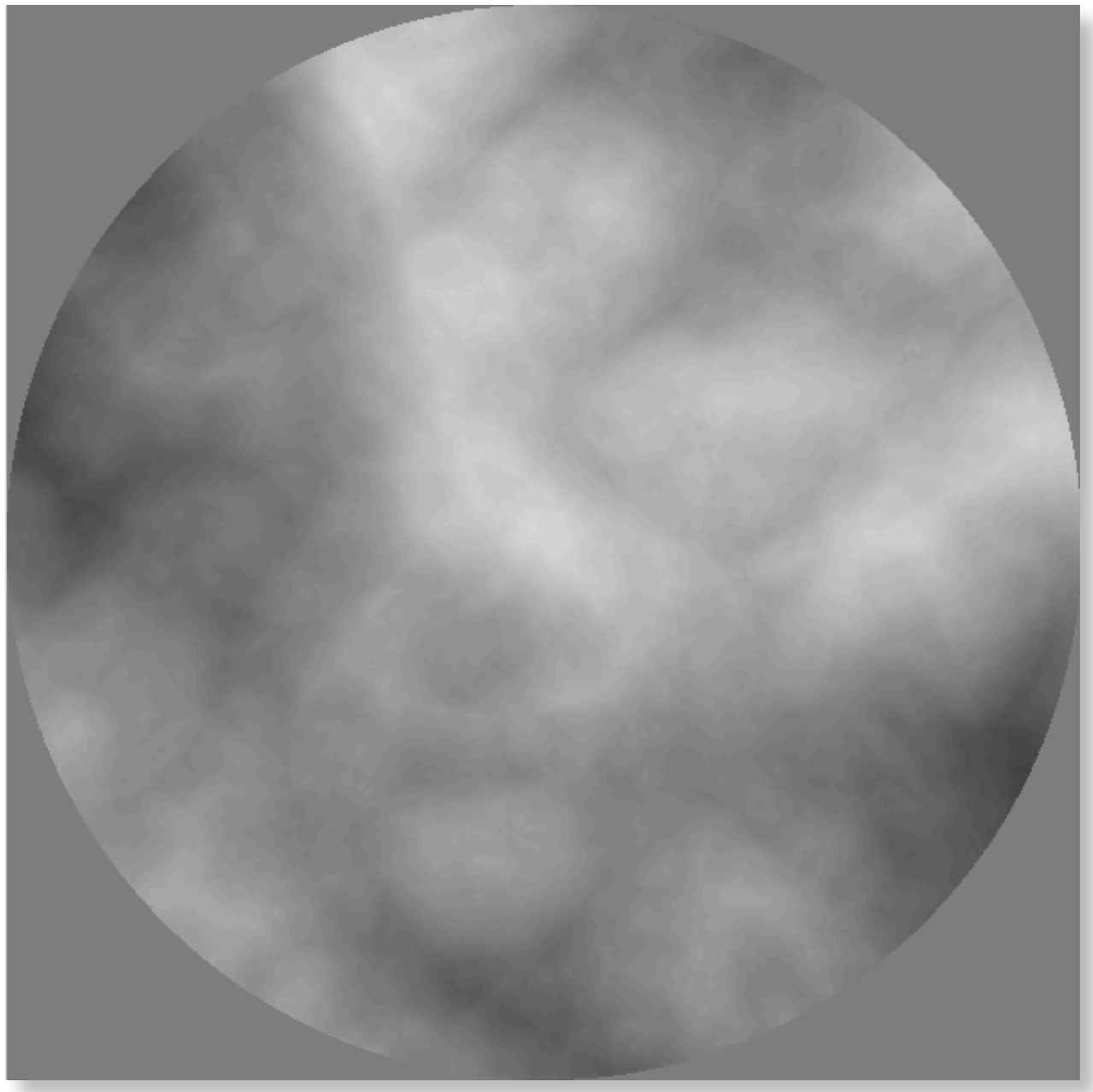
camouflage evolved for *serpentine*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
                Stretch (0.0449509,
                         -1.06448,
                         Vec2 (-1.37922, 0.946741),
                         Furbulence (1.21806,
                                     Vec2 (1.62529,
                                           2.9815)))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```



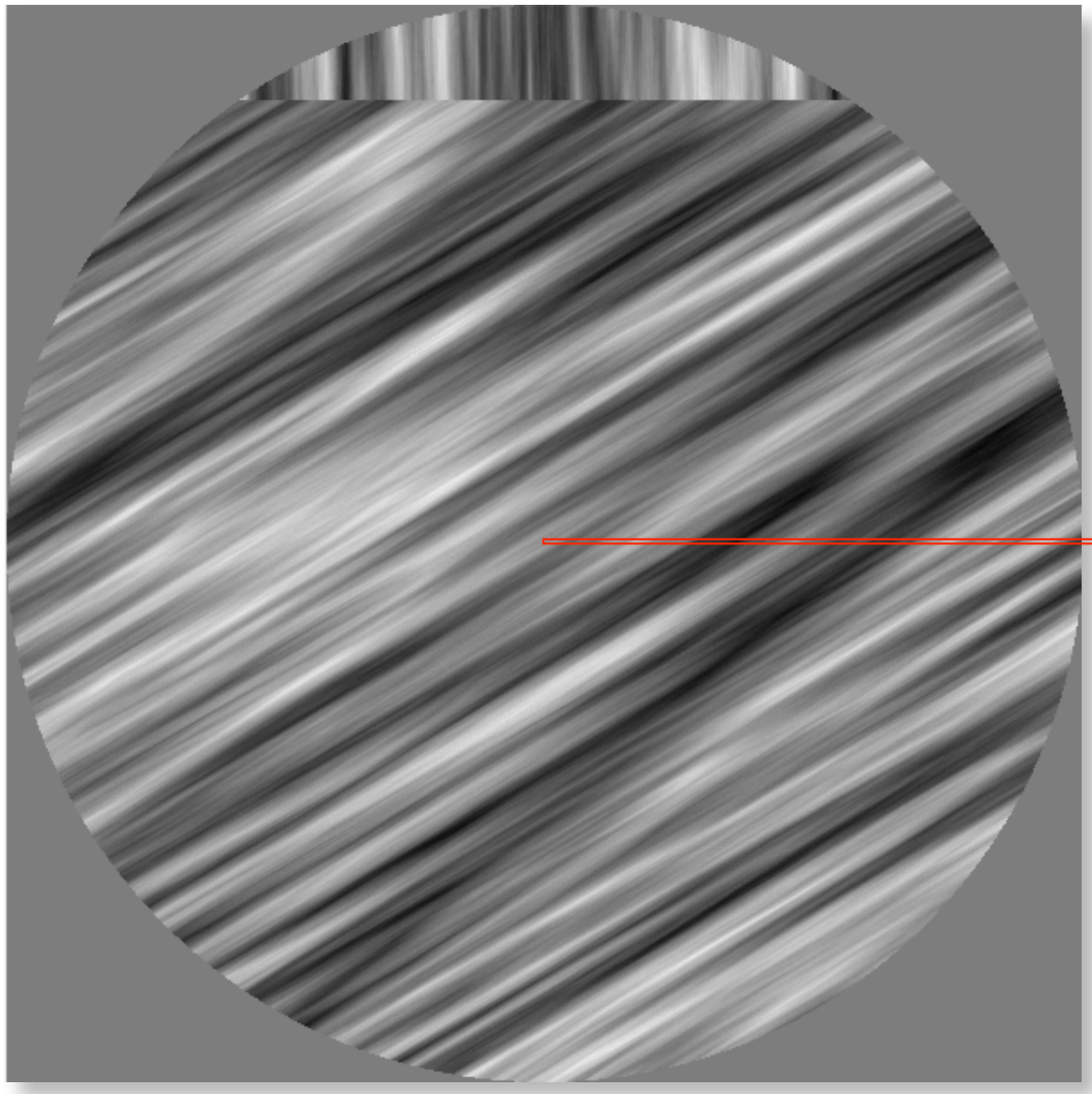
camouflage evolved for *oak bark*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
                Stretch (0.0449509,
                         -1.06448,
                         Vec2 (-1.37922, 0.946741),
                         Furbulence (1.21806,
                                     Vec2 (1.62529,
                                           2.9815)))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416))
```



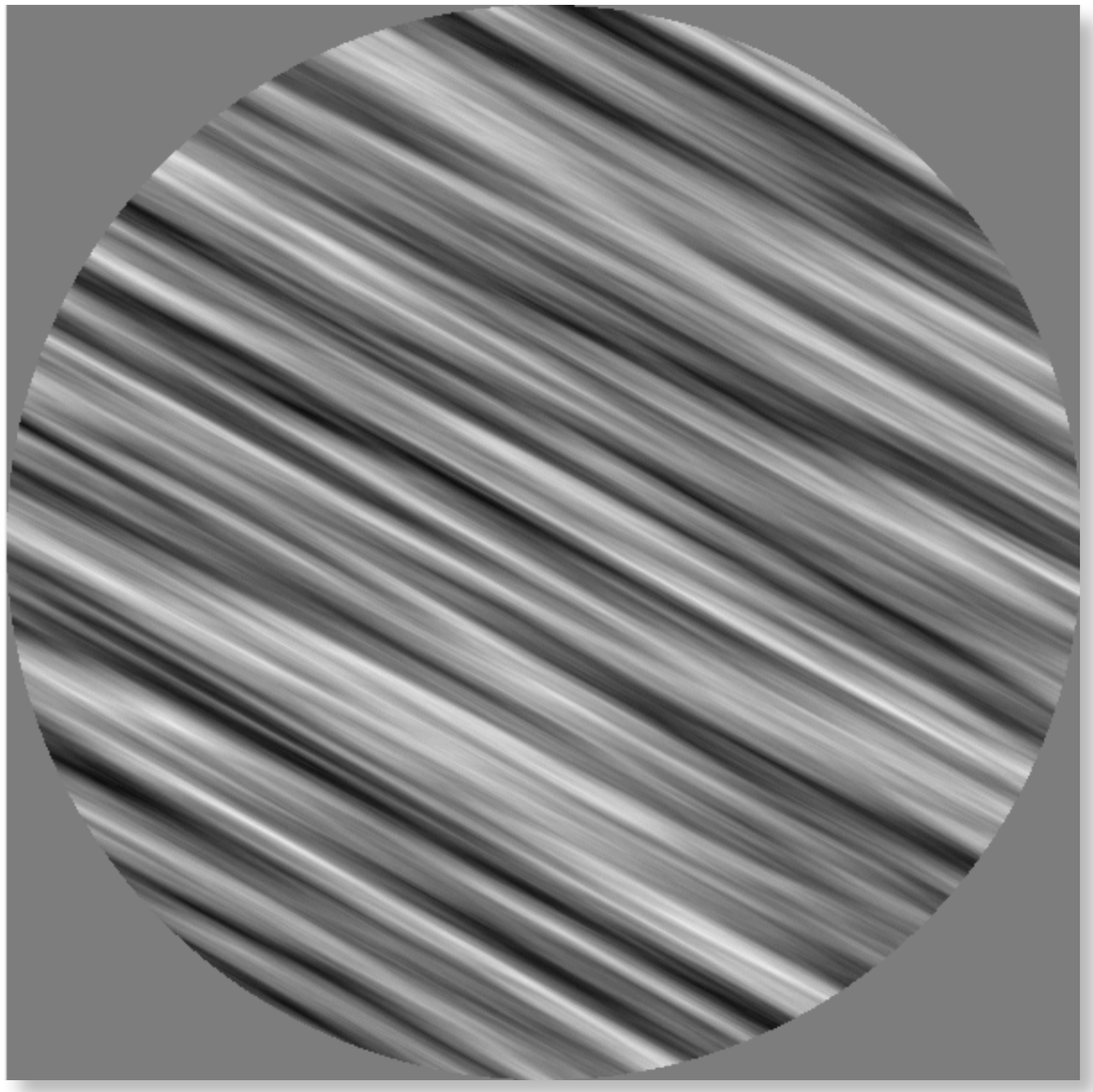
camouflage evolved for *oak bark*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
                Stretch (0.0449509,
                         -1.06448,
                         Vec2 (-1.37922, 0.946741),
                         Furbulence (1.21806,
                                     Vec2 (1.62529,
                                           2.9815)))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```



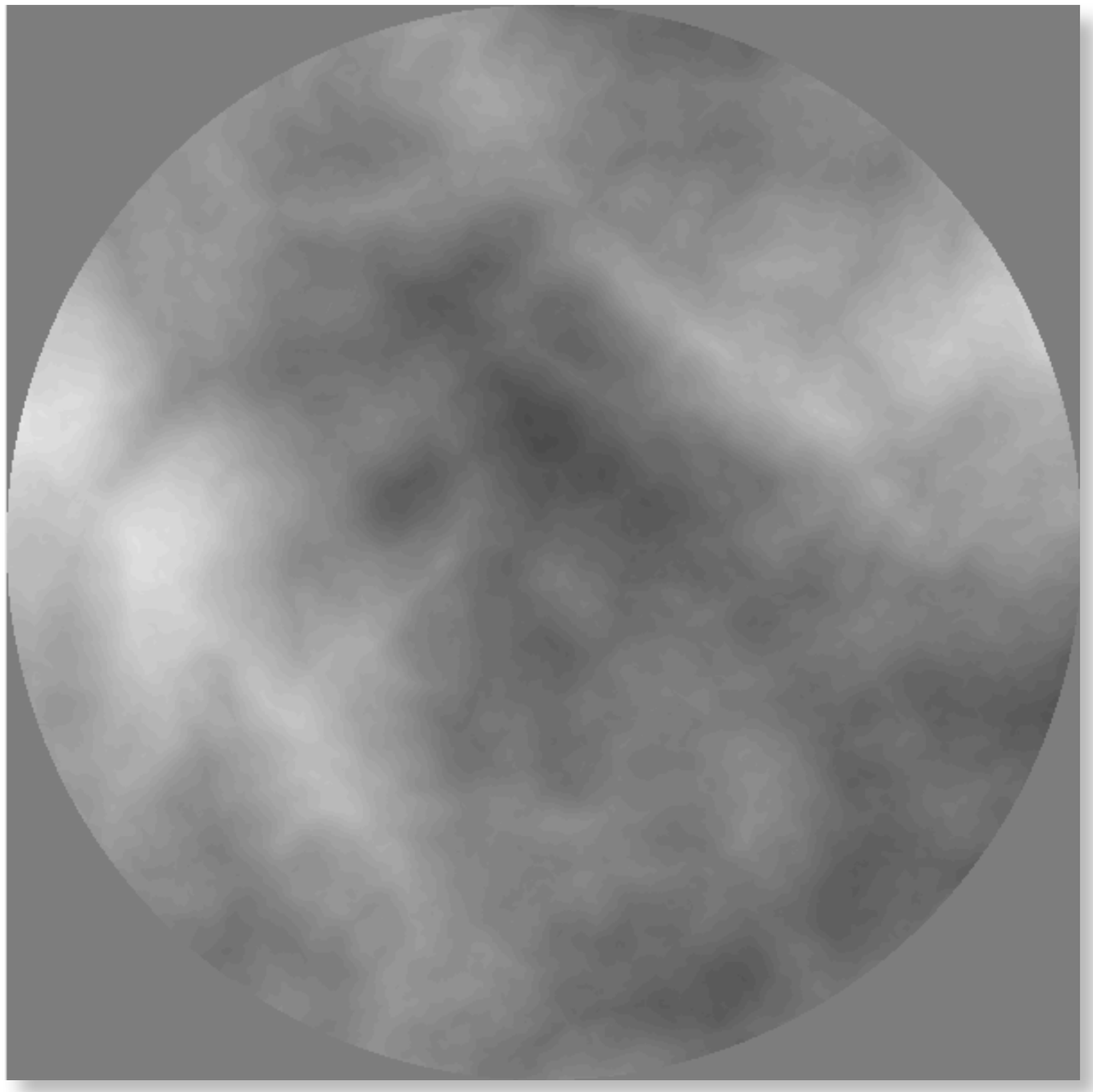
camouflage evolved for *oak bark*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
              Stretch (0.0449509,
                       -1.06448,
                       Vec2 (-1.37922, 0.946741),
                       Furbulence (1.21806,
                                   Vec2 (1.62529,
                                         2.9815))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```



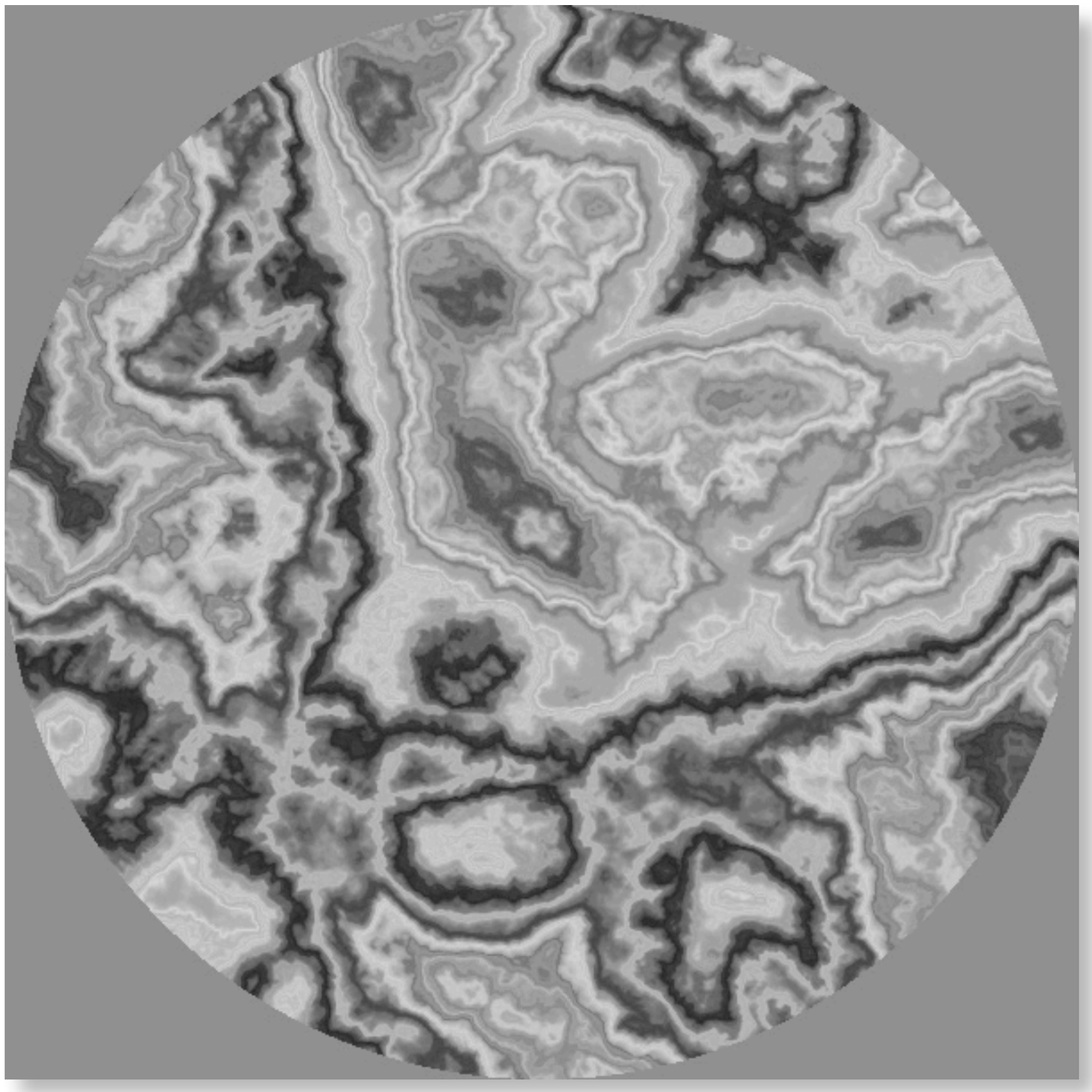
camouflage evolved for *oak bark*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
               Stretch (0.0449509,
                        -1.06448,
                        Vec2 (-1.37922, 0.946741),
                Furbulence (1.21806,
                            Vec2 (1.62529,
                                  2.9815))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```



camouflage evolved for *oak bark*

```
Colorize (Ring (5.80532,
                Vec2 (-2.12073, 0.411024),
                Stretch (0.0449509,
                         -1.06448,
                         Vec2 (-1.37922, 0.946741),
                         Furbulence (1.21806,
                                     Vec2 (1.62529,
                                           2.9815)))),
          Furbulence (1.21806,
                      Vec2 (-2.94693, -1.86416)))
```



camouflage evolved for *oak bark*

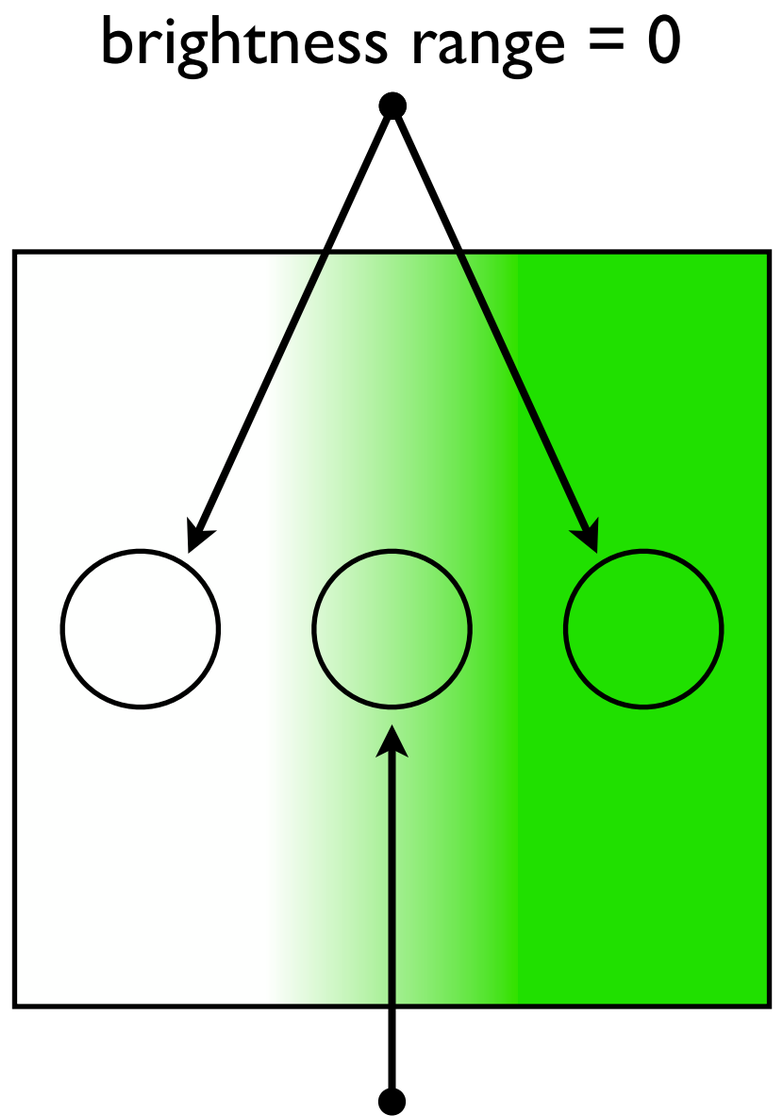
Evolving Textures from High Level Descriptions

Originally a testing procedure

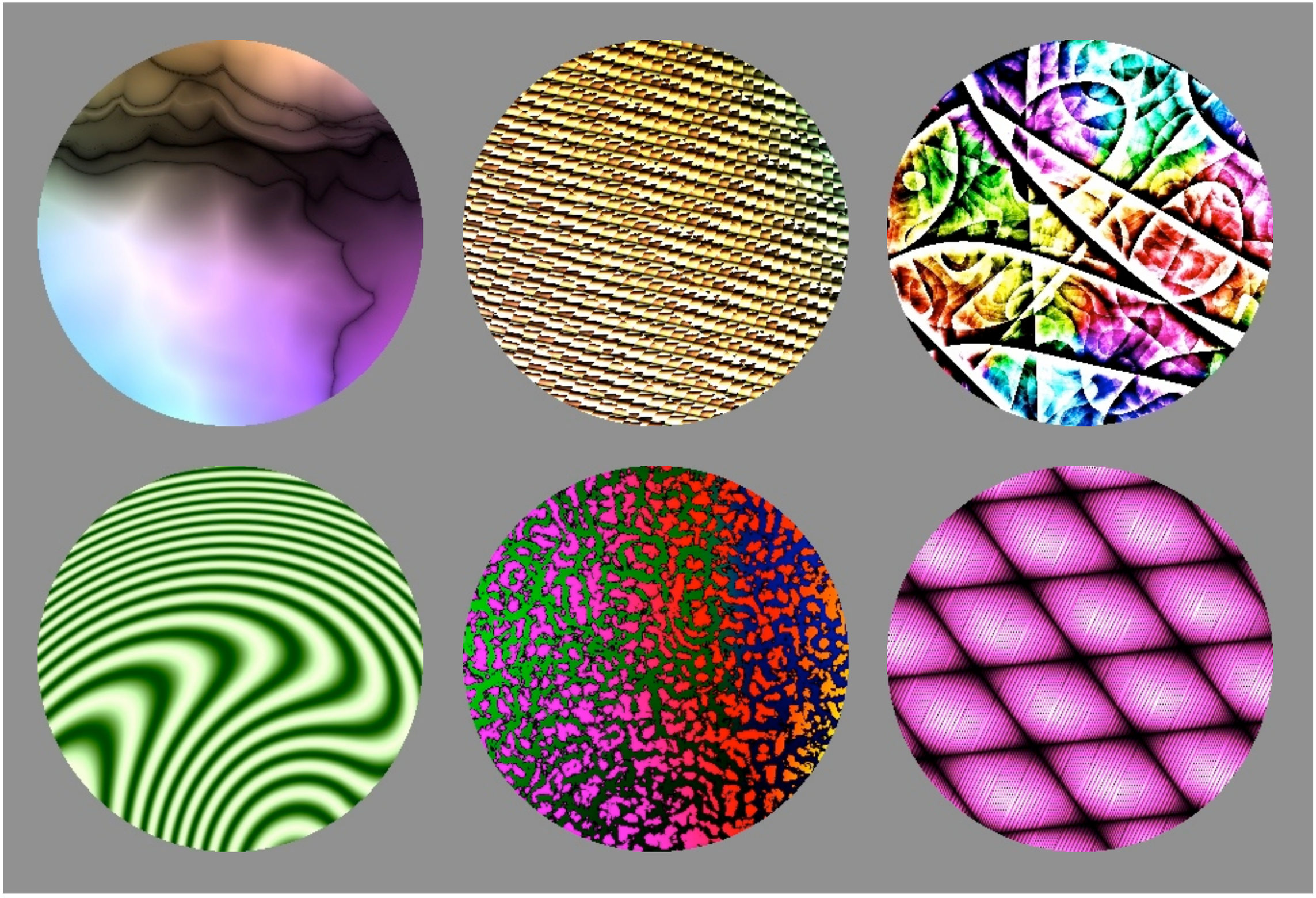
- In preparation for camouflage work:
  - built texture synthesis library
  - connected it to Open BEAGLE
    - needed to test
- Tried some simple fitness functions, which lead to simple and uninteresting results
- Found a set of three metrics that combined to produce interesting results

“colorful, full range, high frequency”

- colorful
  (average pixel saturation above a given threshold)
- well-exposed, full range image
  (semi-uniform brightness histogram)
- textured
  (high average spatial frequency: *variability*)



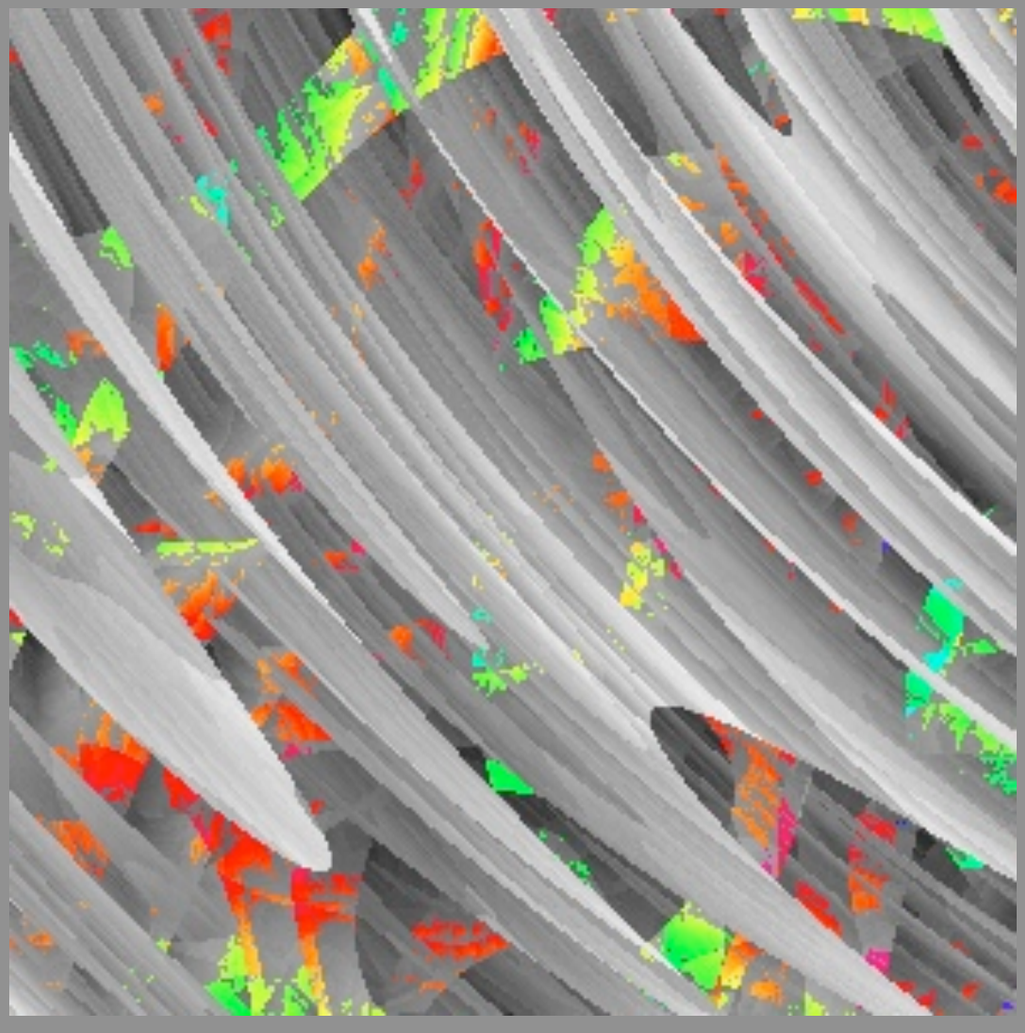
variability: fraction of local neighborhoods whose brightness range (max-min) exceeds a given threshold

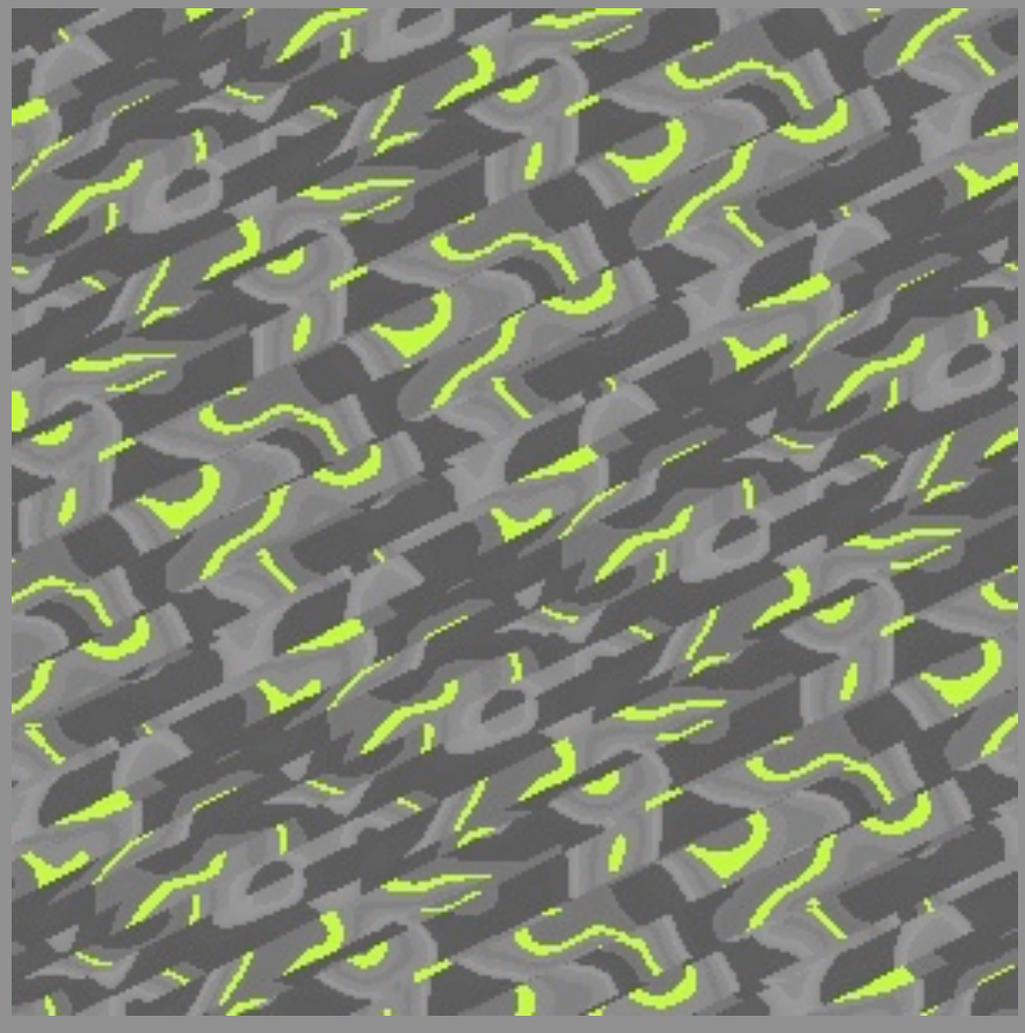


evolved textures: colorful, full range, high frequency

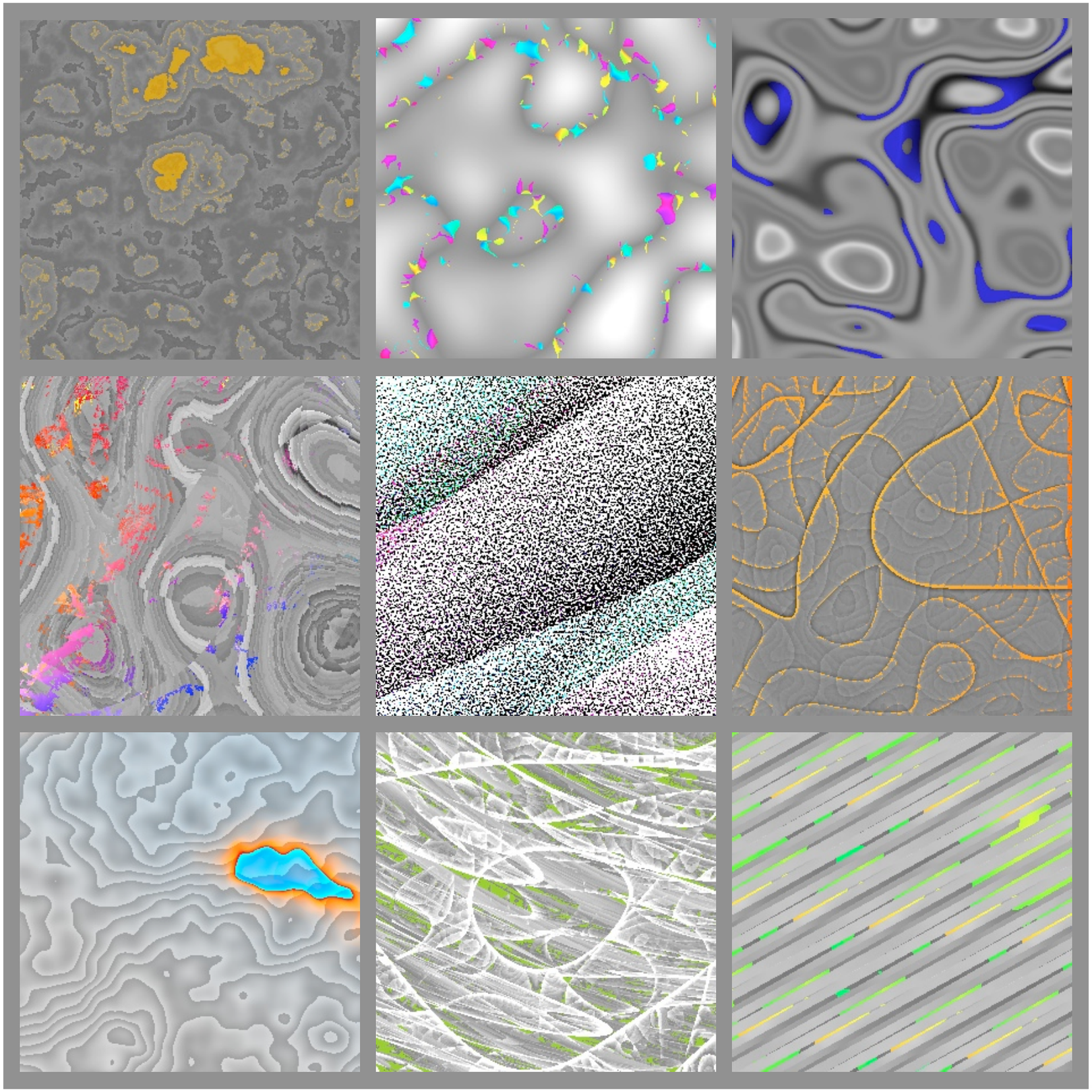
“Gray with accent color”

- fraction of good pixels
  (saturation < gray threshold or > color threshold)
- how close ratio of color/good is to target value
  (5% for a “small amount of accent color”)
- average score for pixel brightness being *midrange*
  (or another target)
- size of bounding box in RGB space for colored pixels
  (promote color variation)
- fraction of variability samples > contrast threshold
  (high frequency)

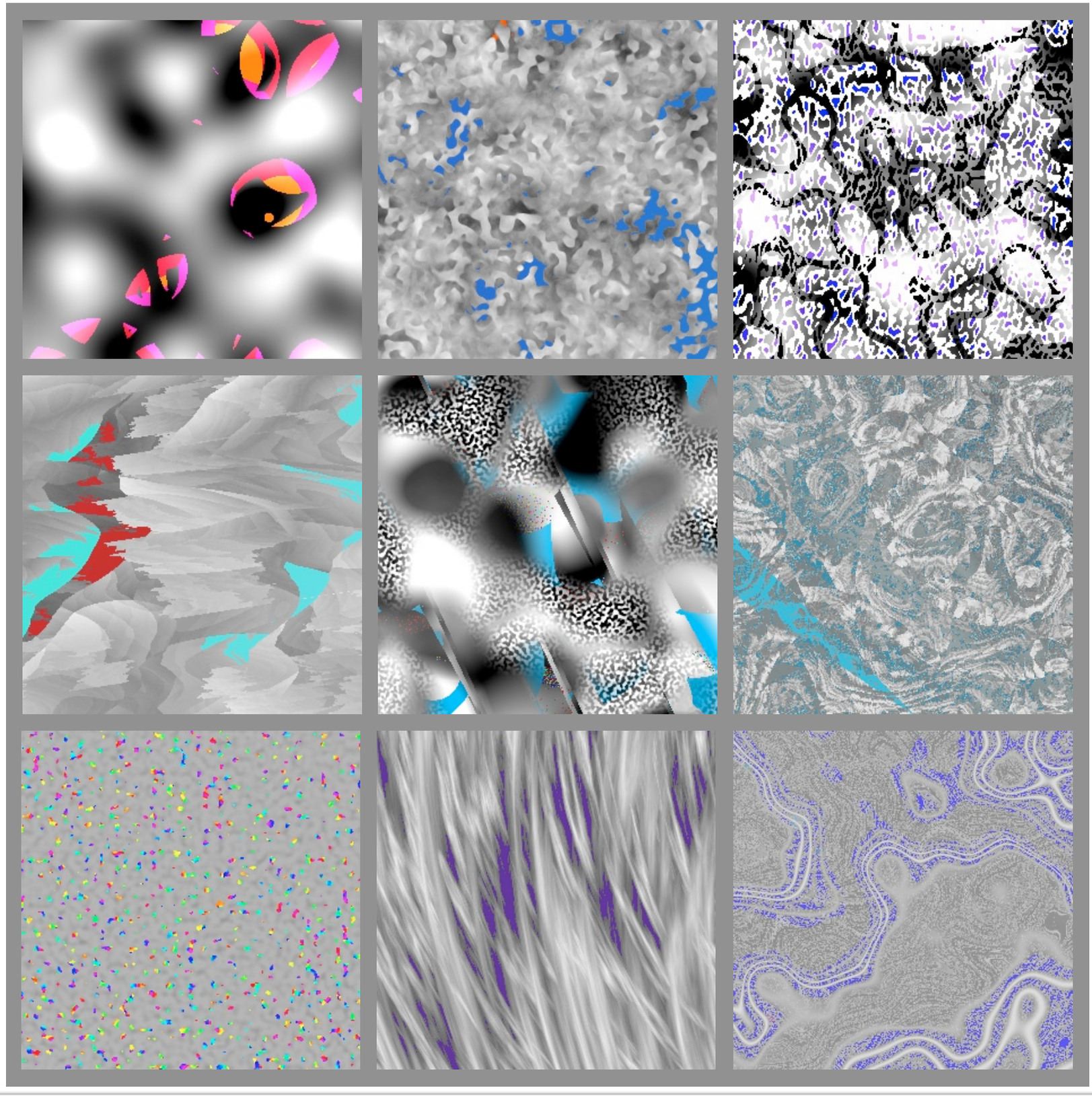




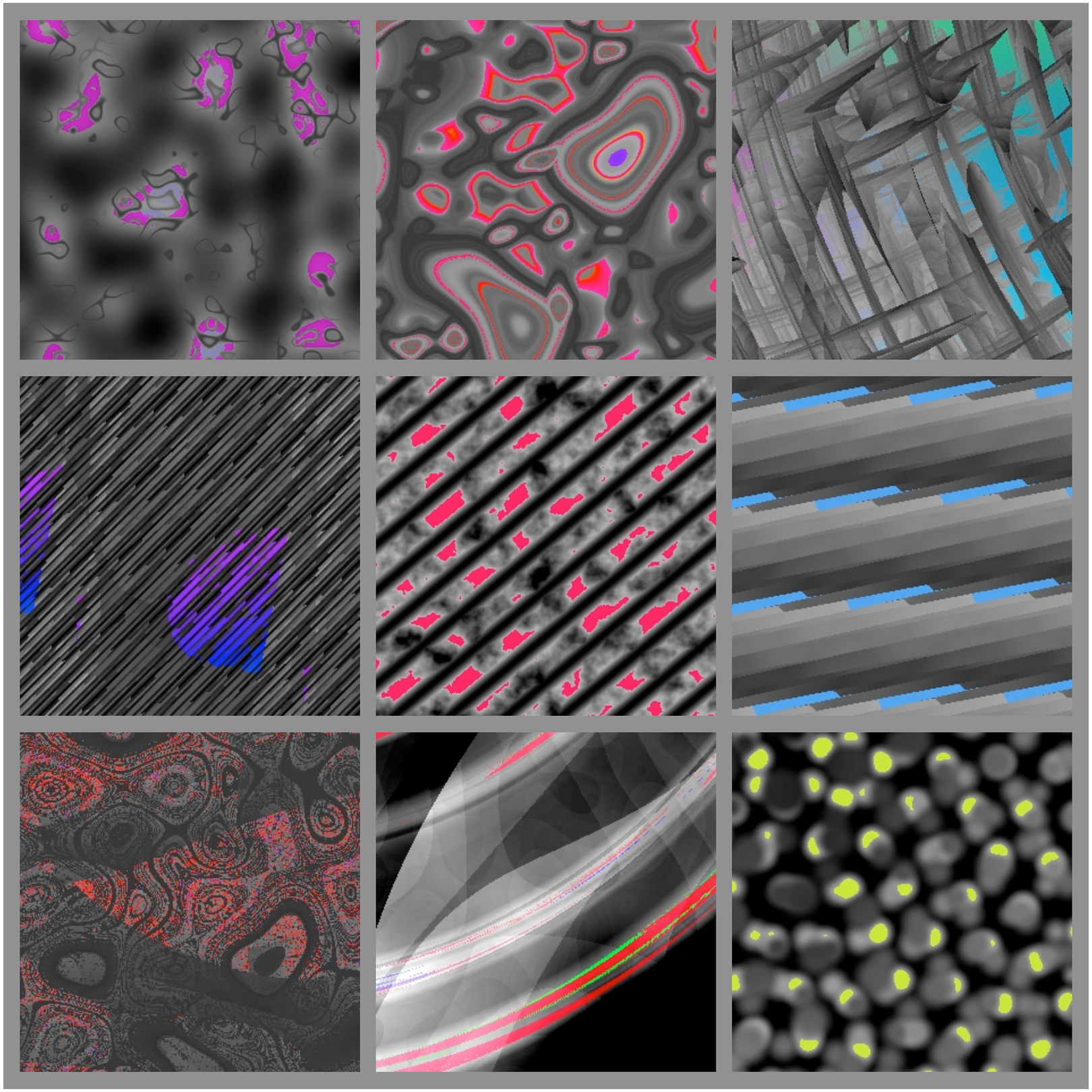
Evolved textures: gray with accent color



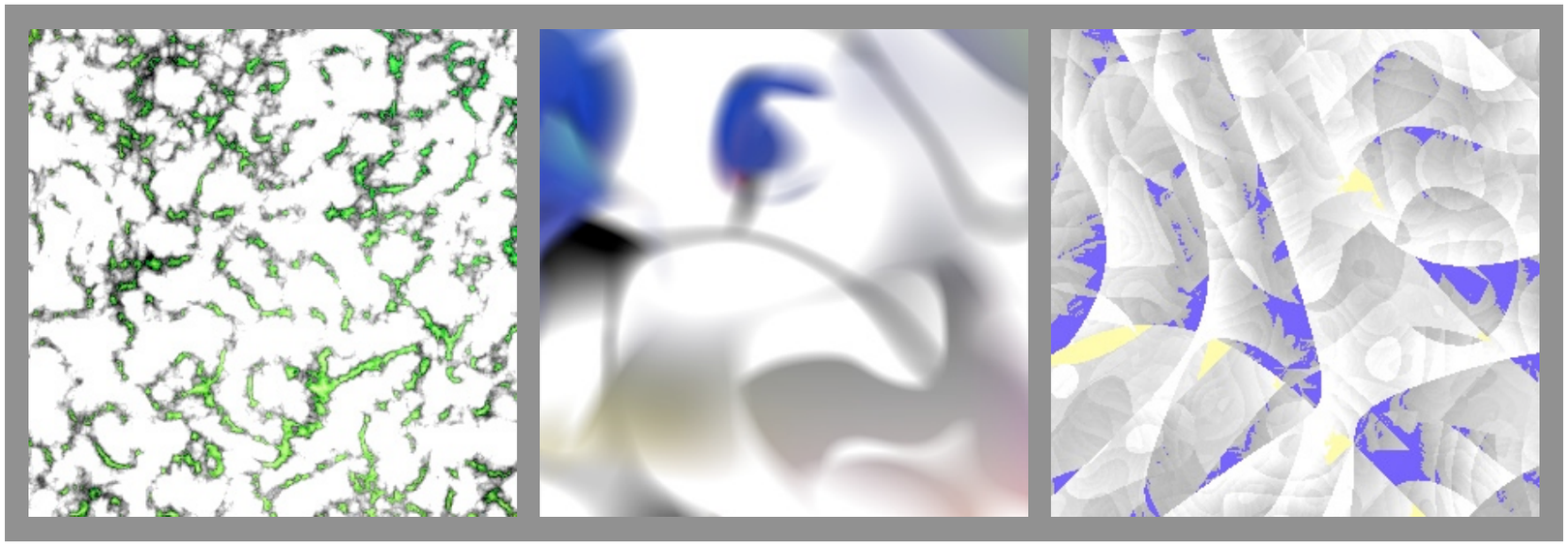
Evolved textures: gray with accent color



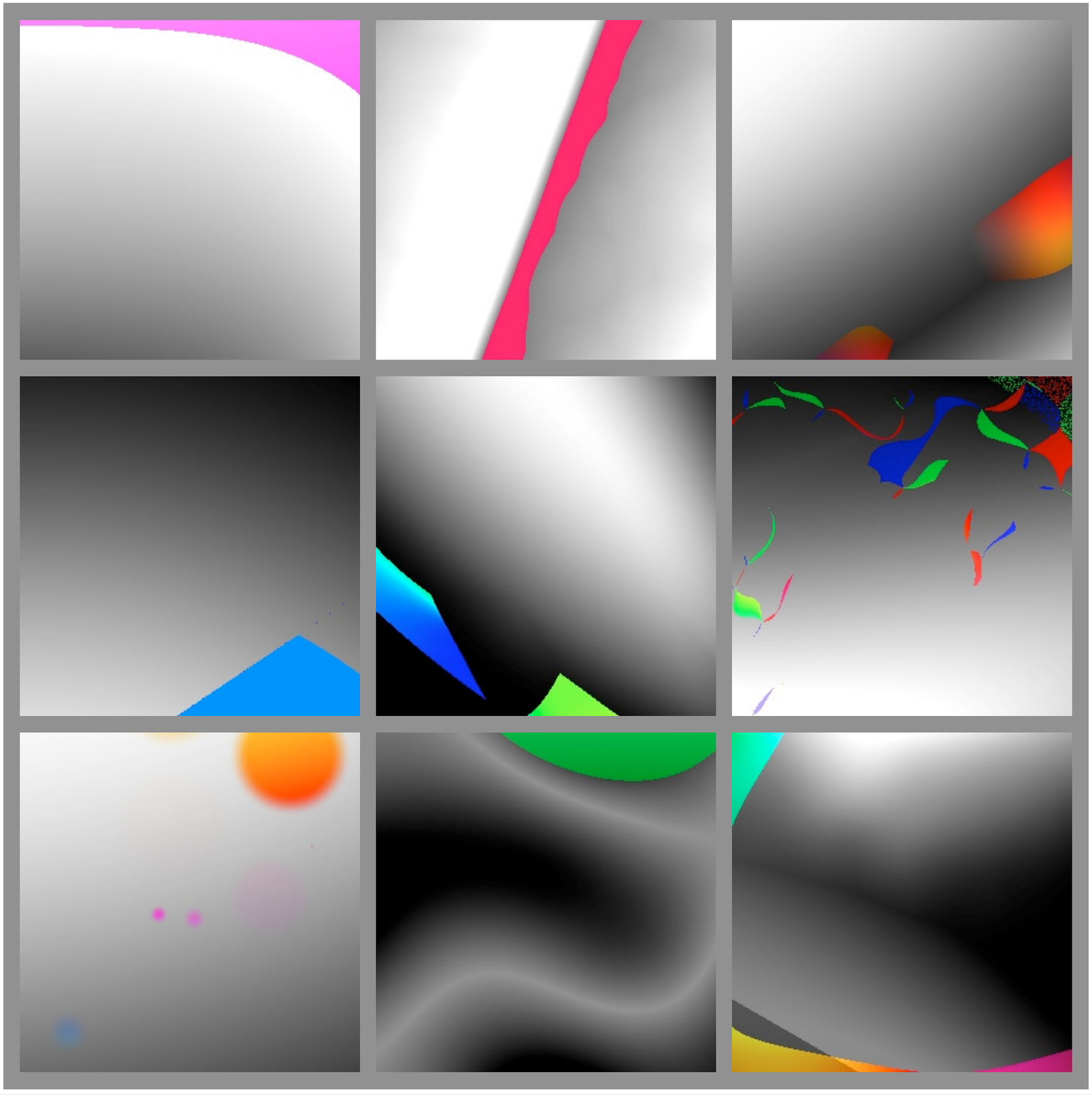
Evolved textures: gray with accent color



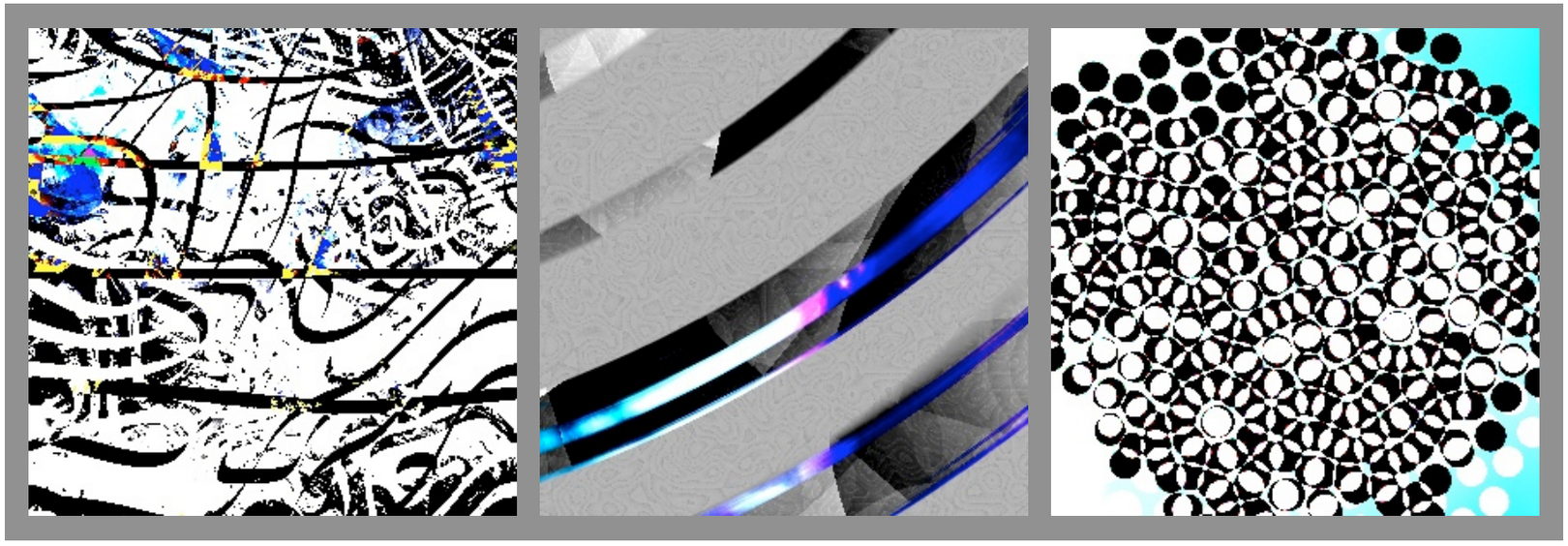
Evolved textures: gray with accent color — dark



Evolved textures: gray with accent color — bright



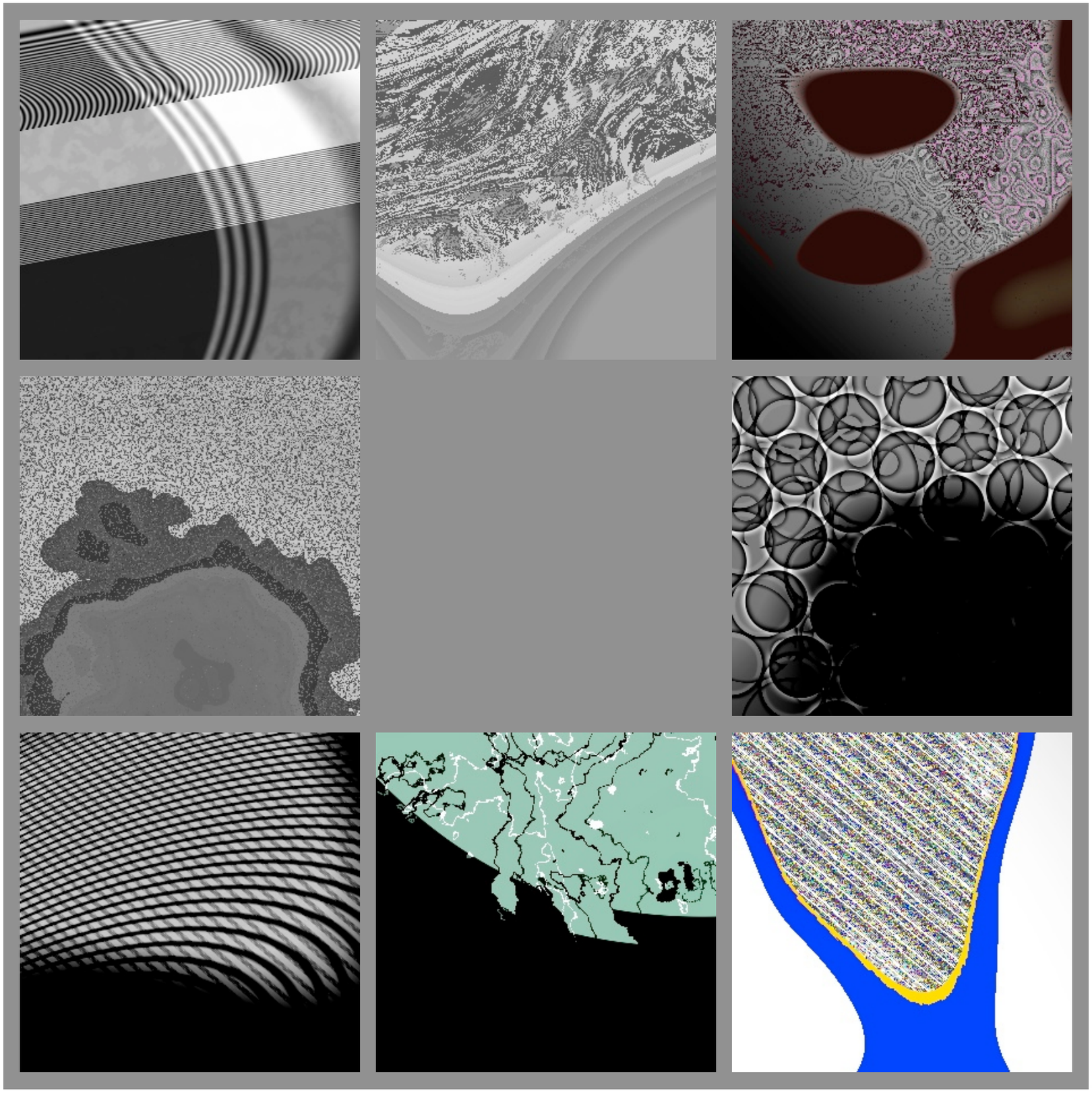
Evolved textures: gray with accent color — “minimalist”



Evolved textures: gray with accent color — no *max/min*

HFTLFB

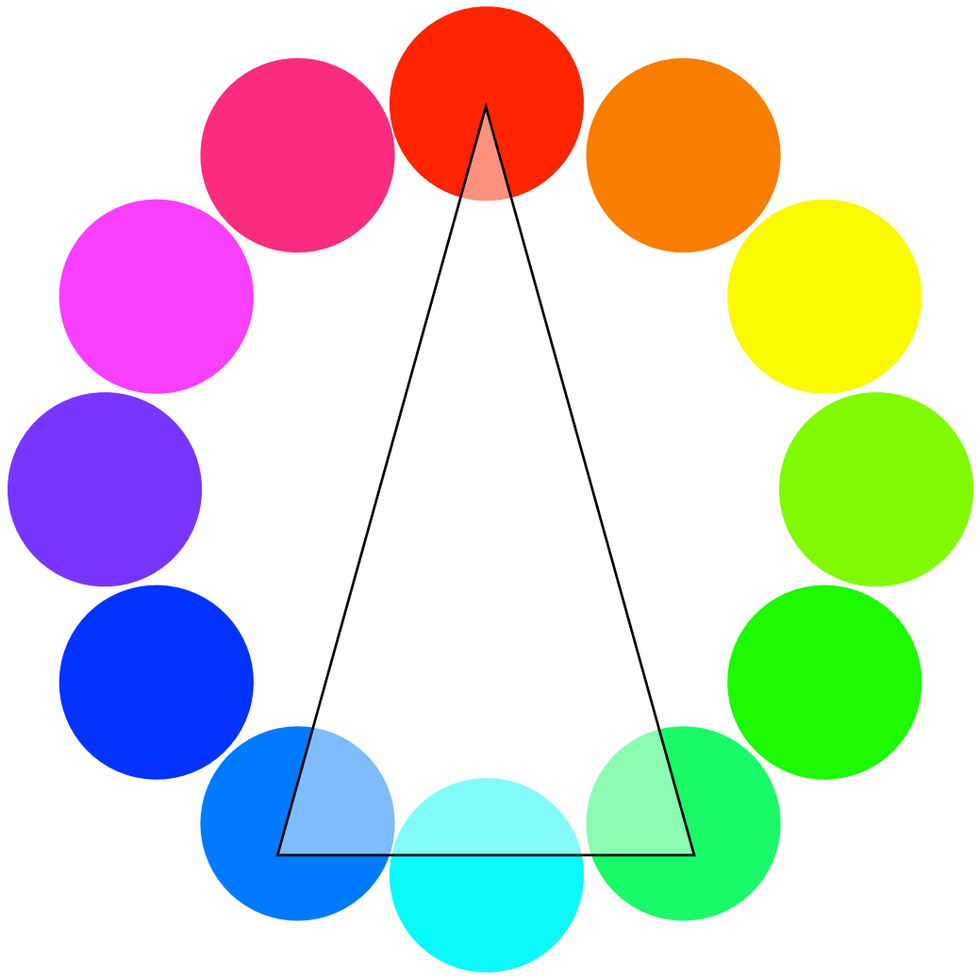
- HFTLFB: “high frequency top, low frequency bottom”
  - high frequency in top 1/4 of image
  - low frequency in bottom 1/4 of image
  - prefer some variation at bottom



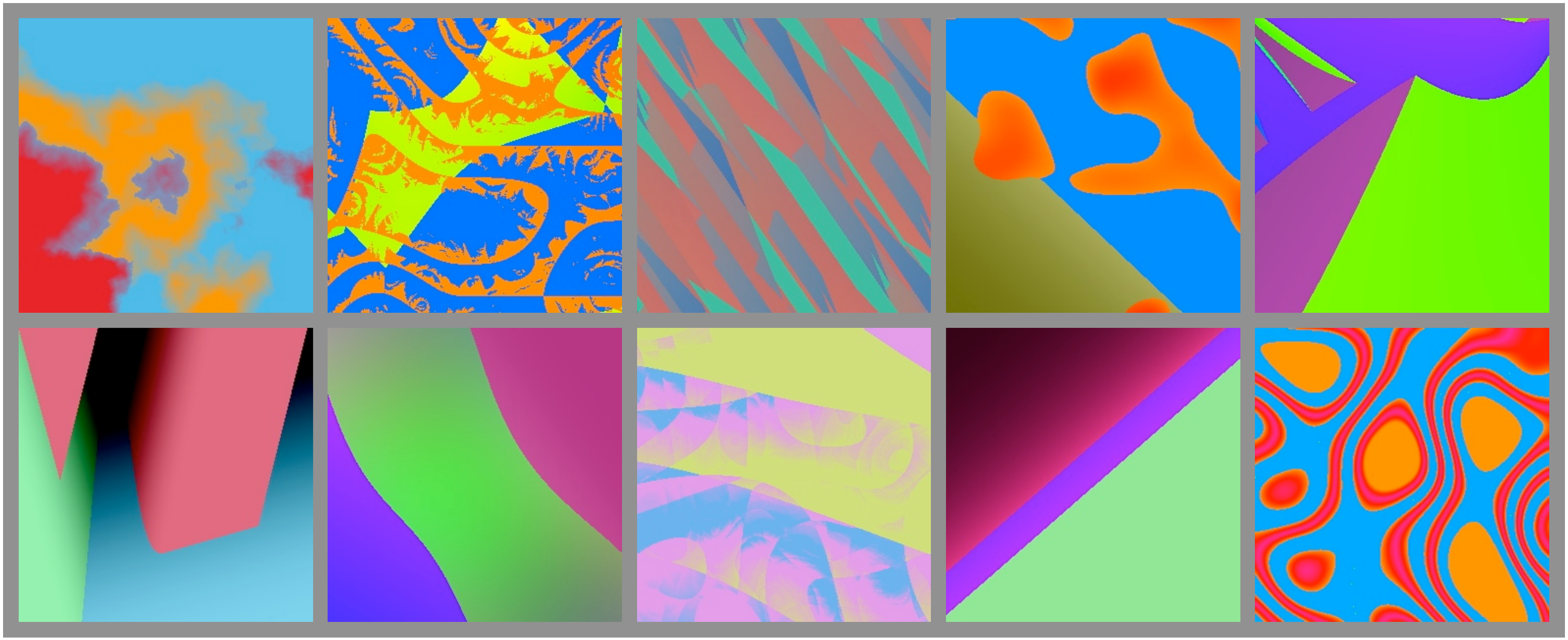
Evolved textures: *high frequency top, low frequency bottom*

Harmonious colors

- Textures with a harmonious color scheme (using a 12 bucket histogram for hue)
  - colorful
  - color mostly in three hue buckets (in ratio of 3:1:1)
  - other nine histogram buckets nearly empty
  - three hues form “harmonious” color triplet



harmonious color triplet



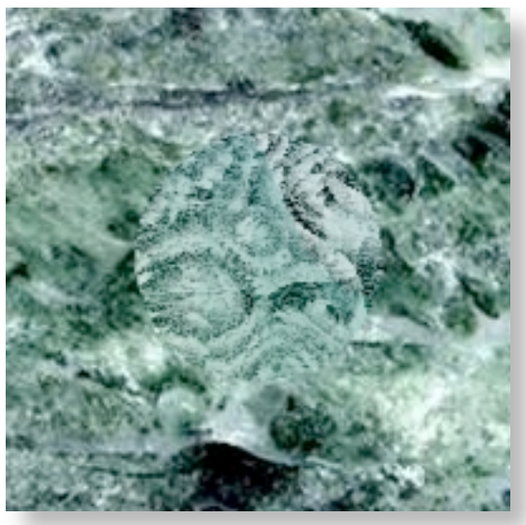
Evolved textures: harmonious colors

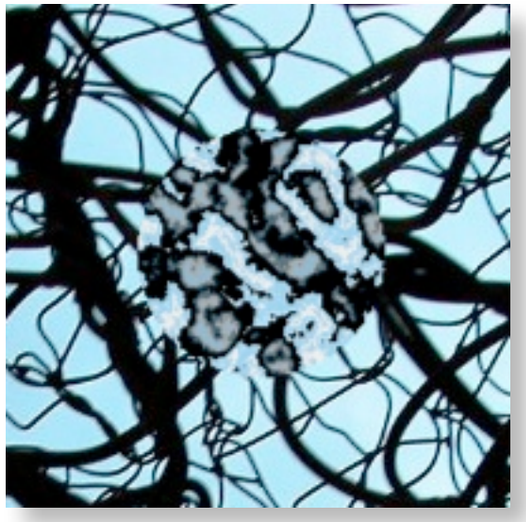
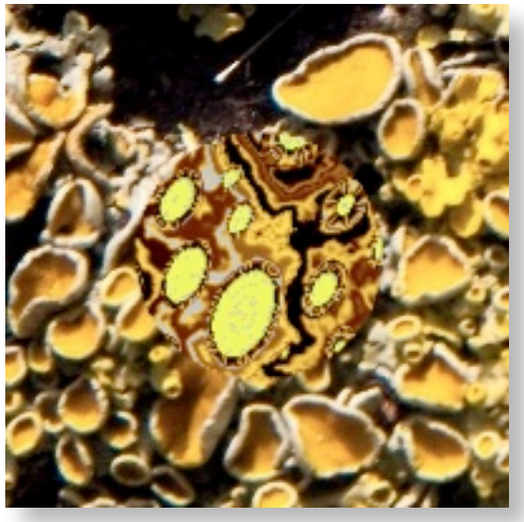
Camouflage

Introduction

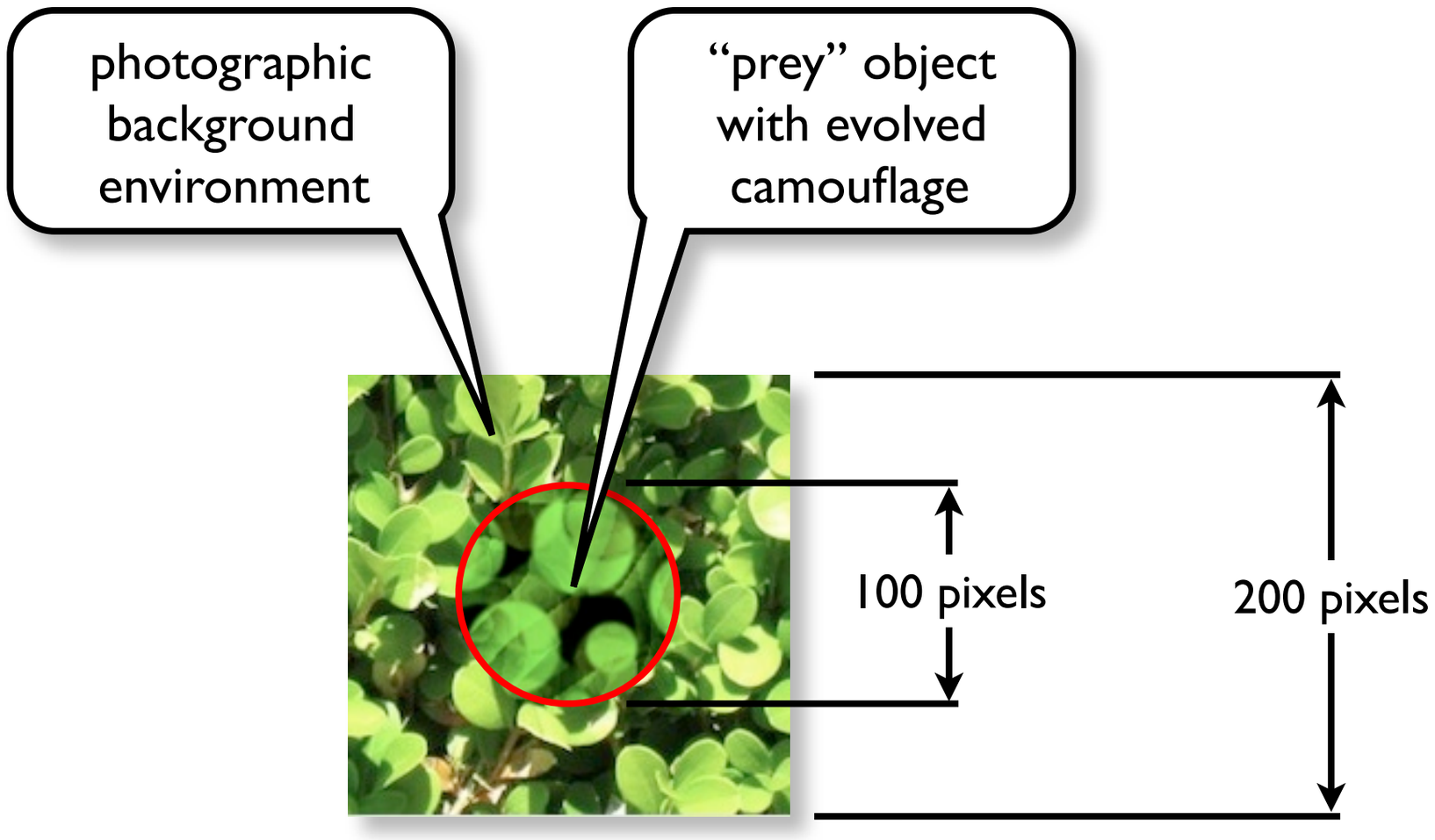
- Abstract model of camouflage evolution in nature.
  - evolutionary computation: GP
  - computer graphics: procedural texture synthesis
  - hybrid computer system: human vision “in the loop”

Evolved camouflage

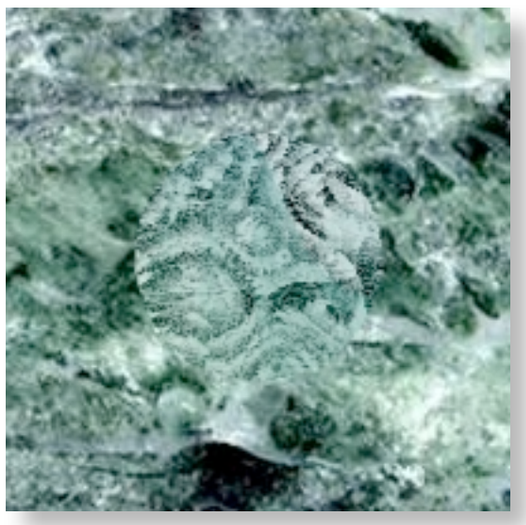



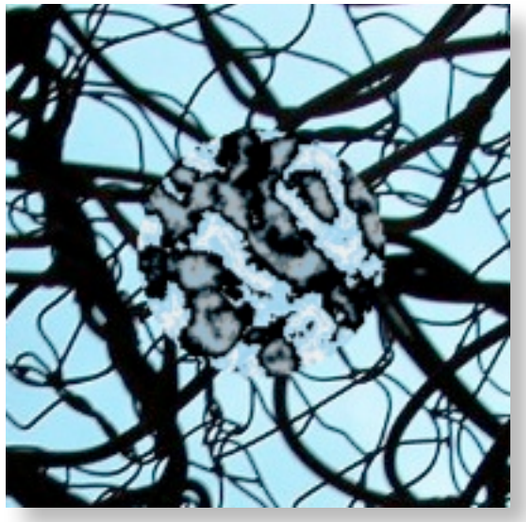
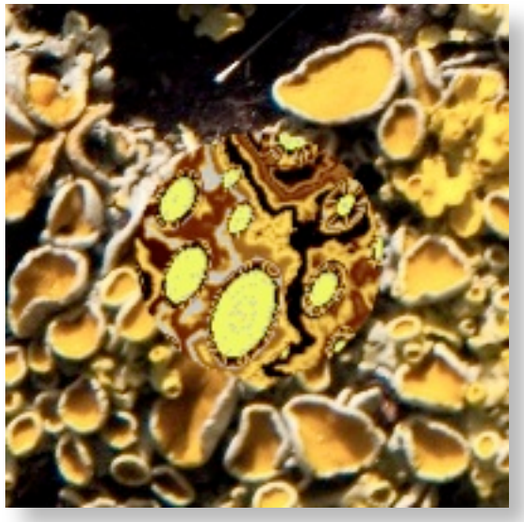




Structure of *thumbnail* images

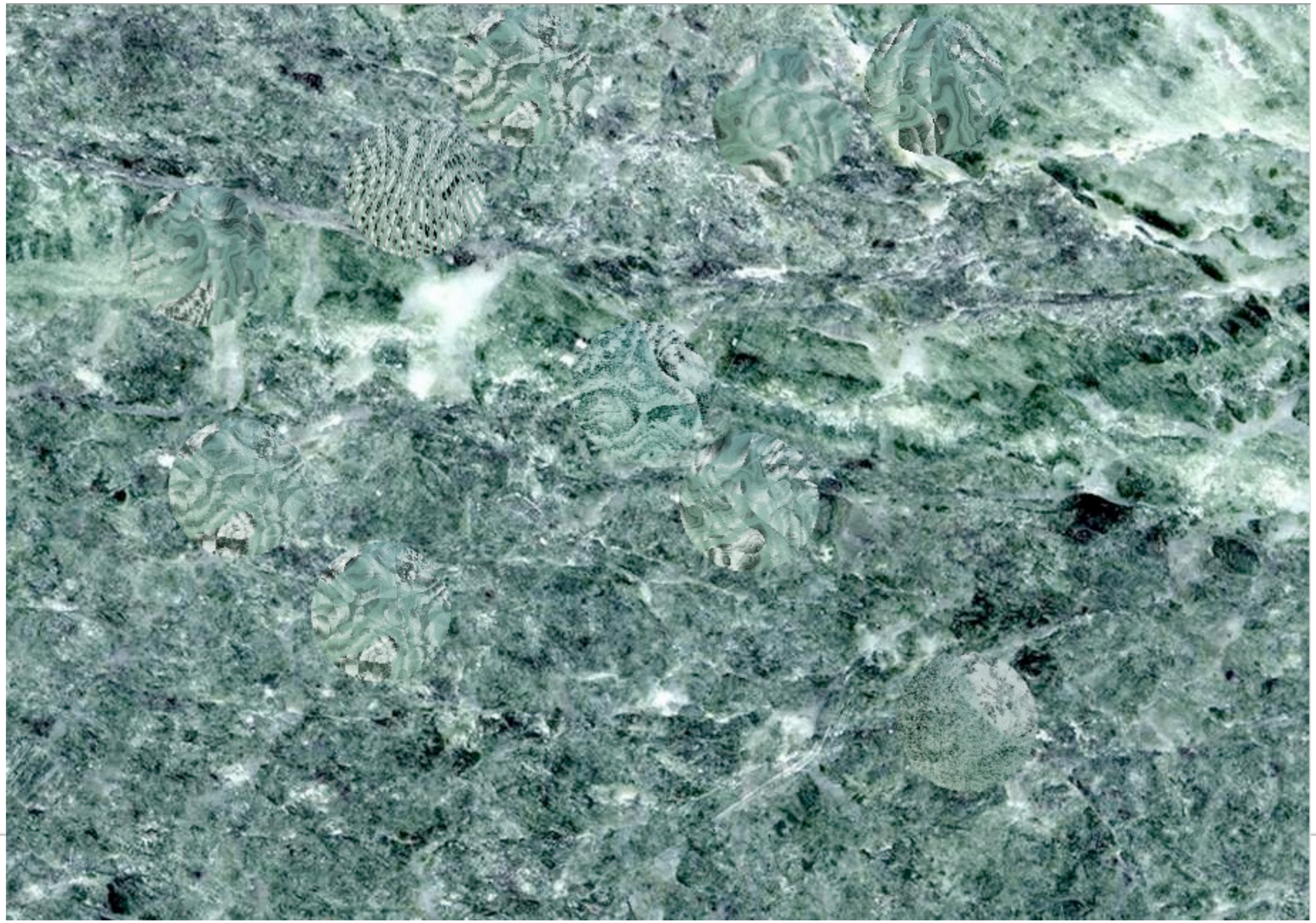


Evolved camouflage

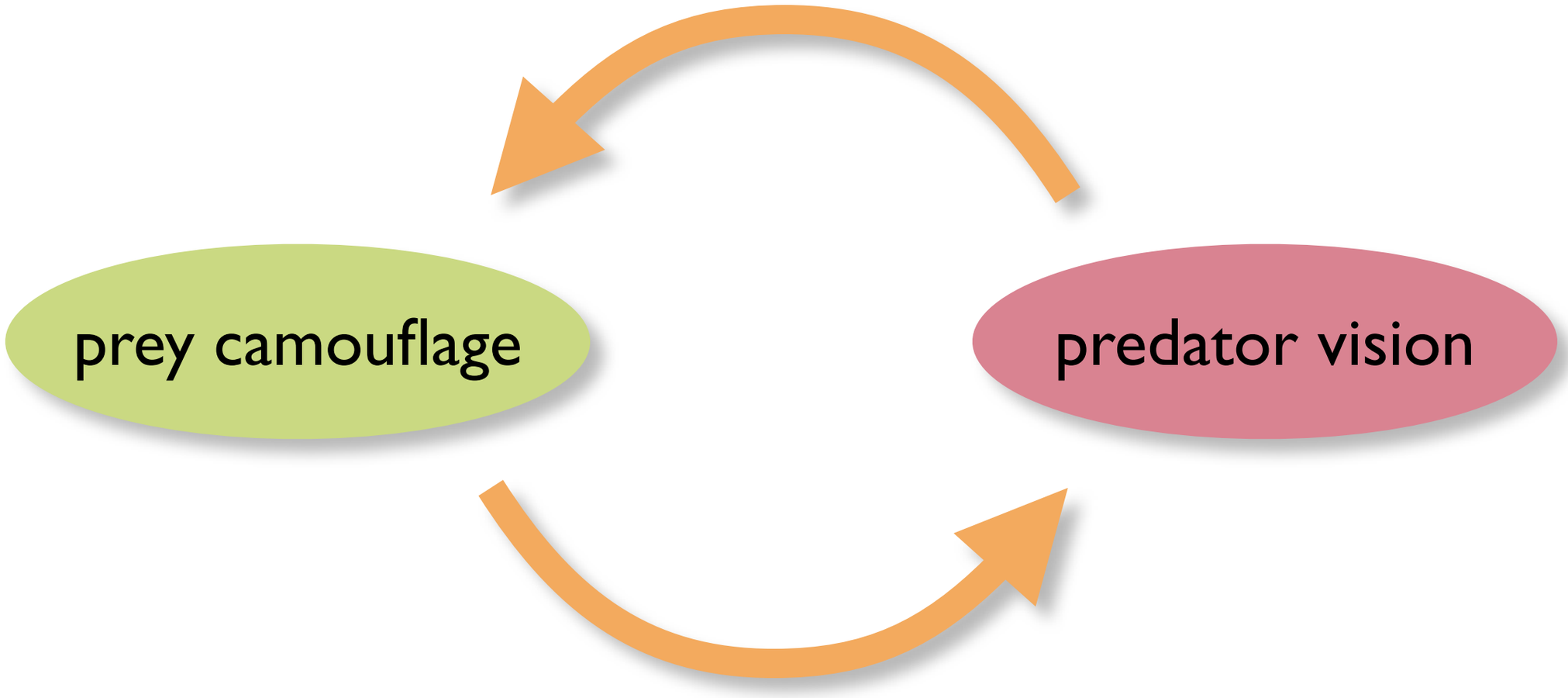




Cohort on environment

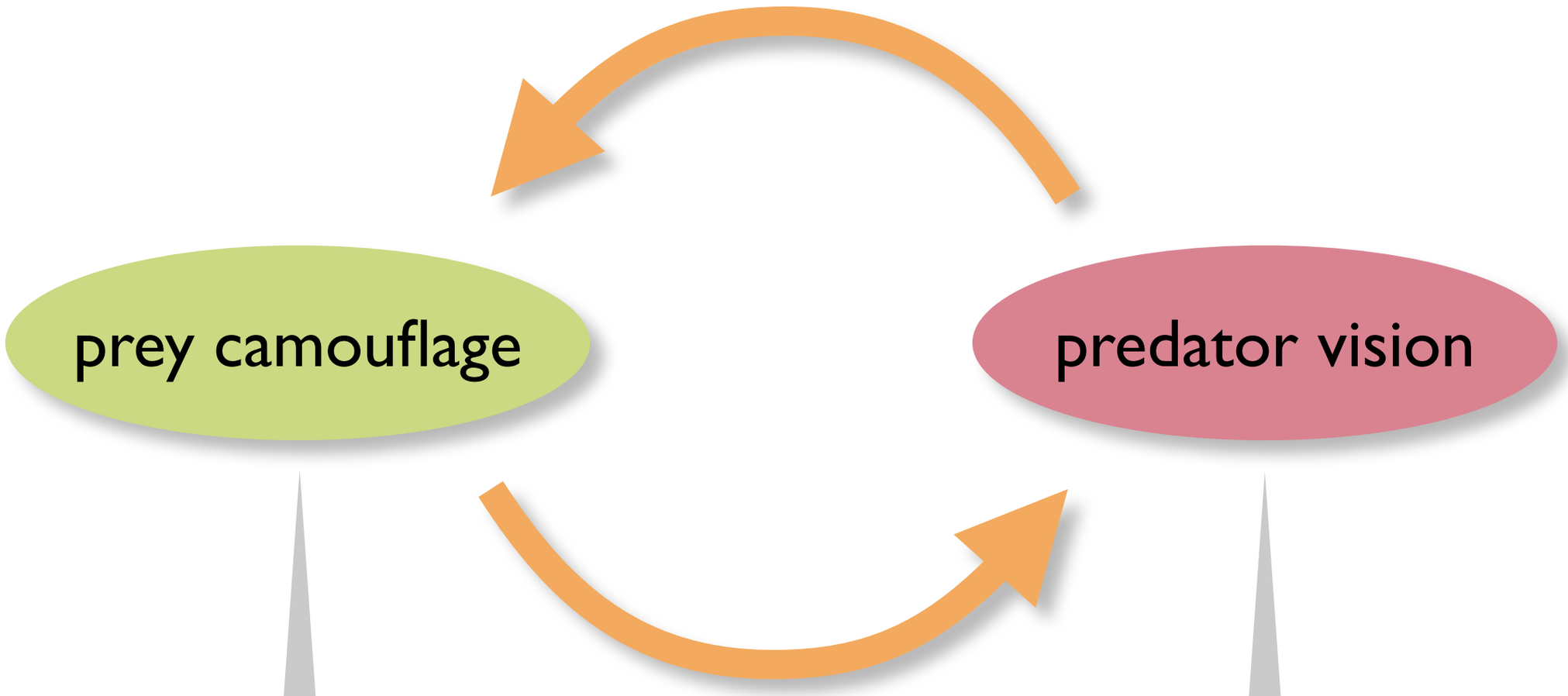


Overview of Model

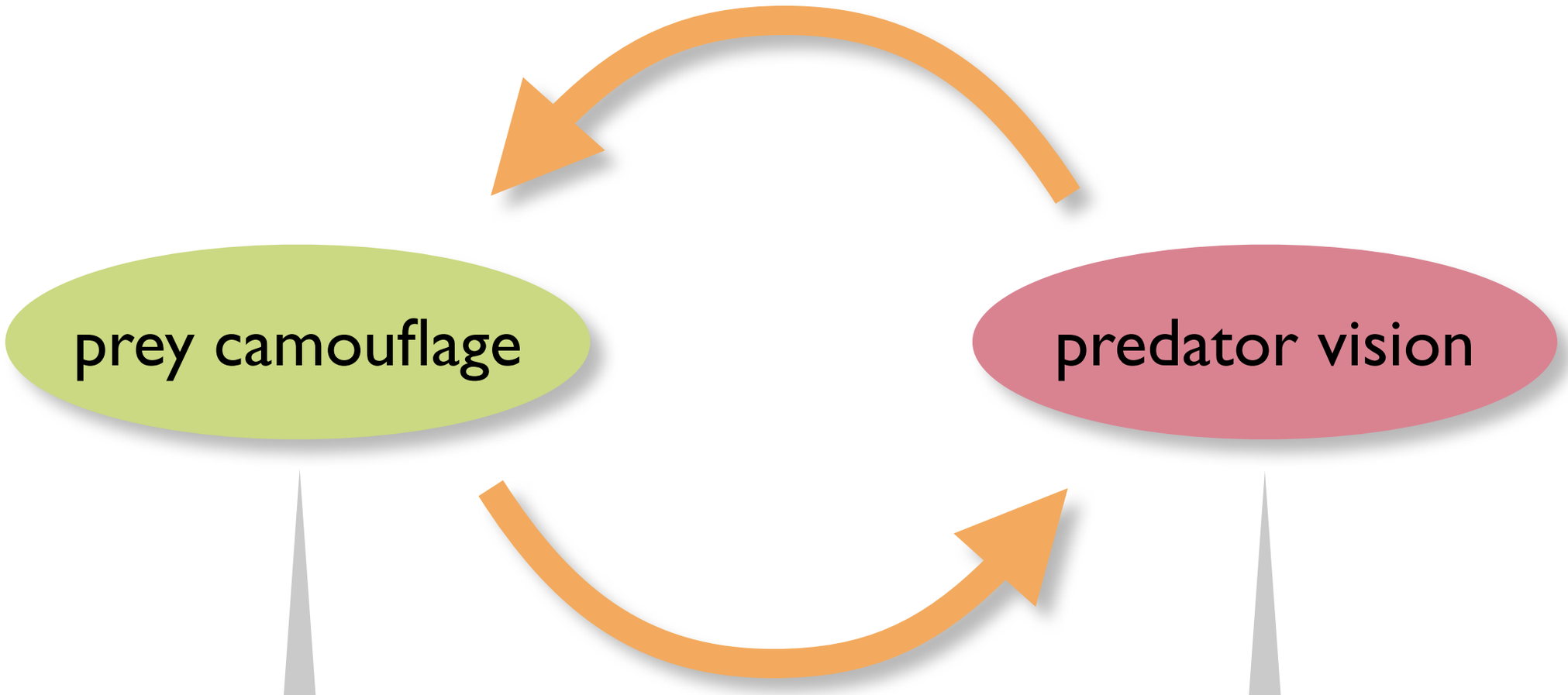
Coevolutionary system



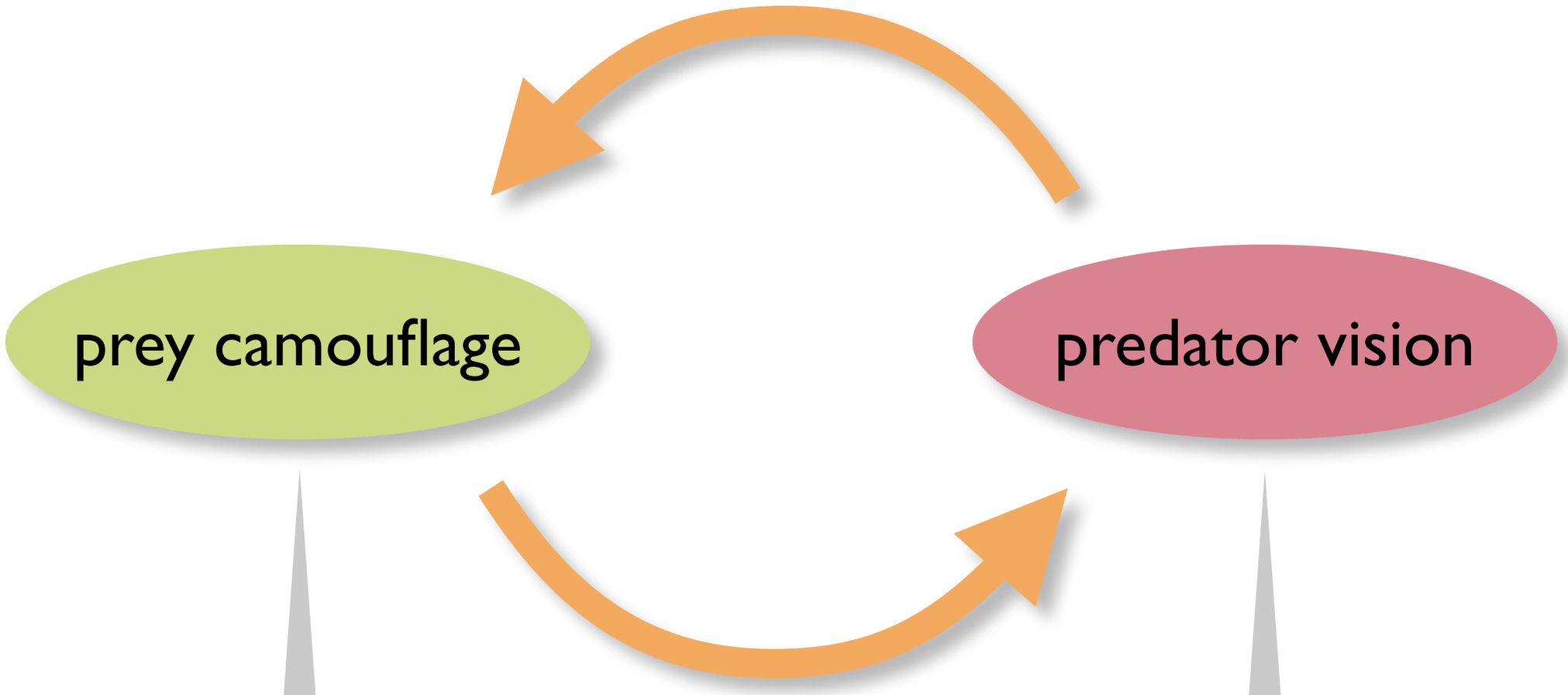
In nature


prey camouflage
predator vision
morphogenesis
evolution
eye / brain
evolution

Current version: hybrid of procedural and interactive



Eventual goal: a fully procedural simulation



Camouflage in Nature

Malagasy Lanternfly, forest canopy, Madagascar

Caterpillar of Common Baron butterfly (*Euthalia aconthea*), Malaysia

Oak Beauty (*Biston strataria*) on bark, England.

Bark bug, Peruvian Amazon

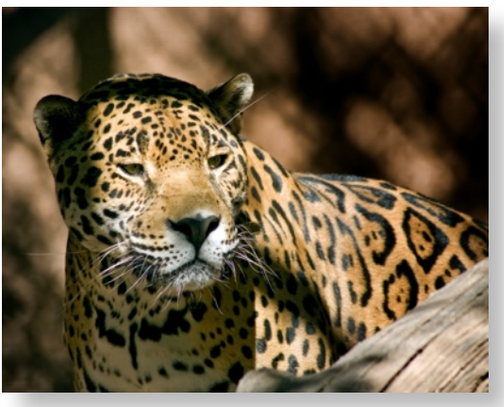
Wolf spider (*Ocyale guttata*)

Crab spider (*Thomisus onustus?*), France

Crab spider (*Thomisus onustus*) with prey, France

Camouflage change speed

permanent | seasonal | weeks | minutes | instantaneous



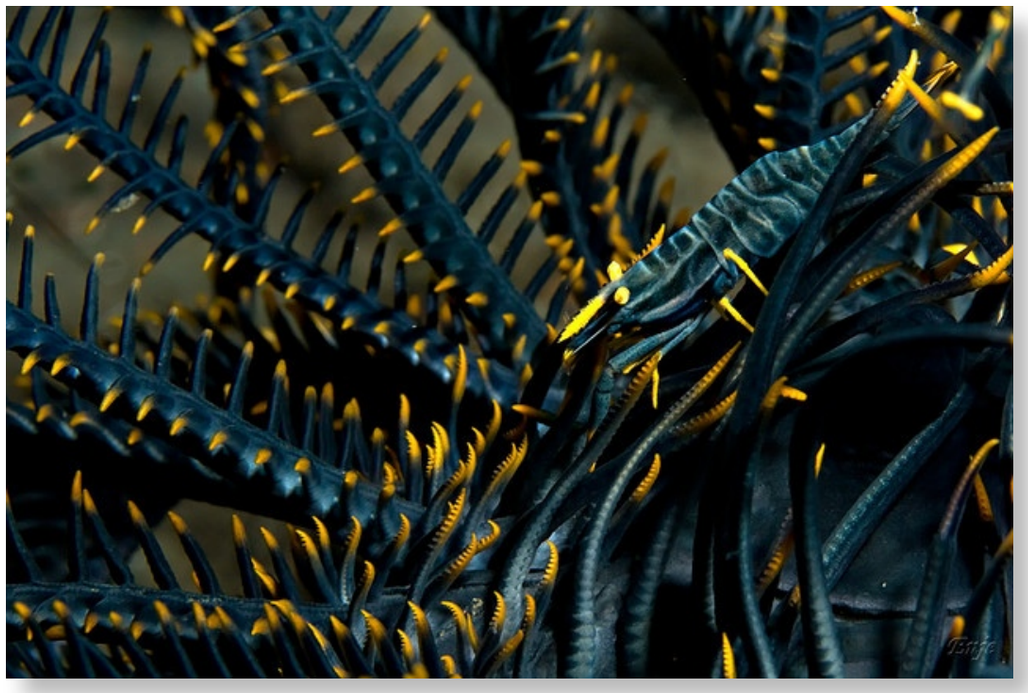


iStockPhoto, Terry L Spivey (Bugwood.org), Ansgar Walk, Hannes Mitchell, Damiano Luchetti, Tennessee Aquarium

Jacky lizard (*Amphibolurus muricatus*), Australia

juvenile C-O sole (*Pleuronichthys coenosus*)

Crinoid with commensal shrimp, Philippines



Crinoids with commensal shrimp

Tawny frogmouth owls (*Podargus strigoides*)

Scops owl (*Otus Scops*) on olive tree

Owl on fir(?) tree.

(photographer unknown)

Northern leopard frog (*Rana pipiens*), Michigan USA

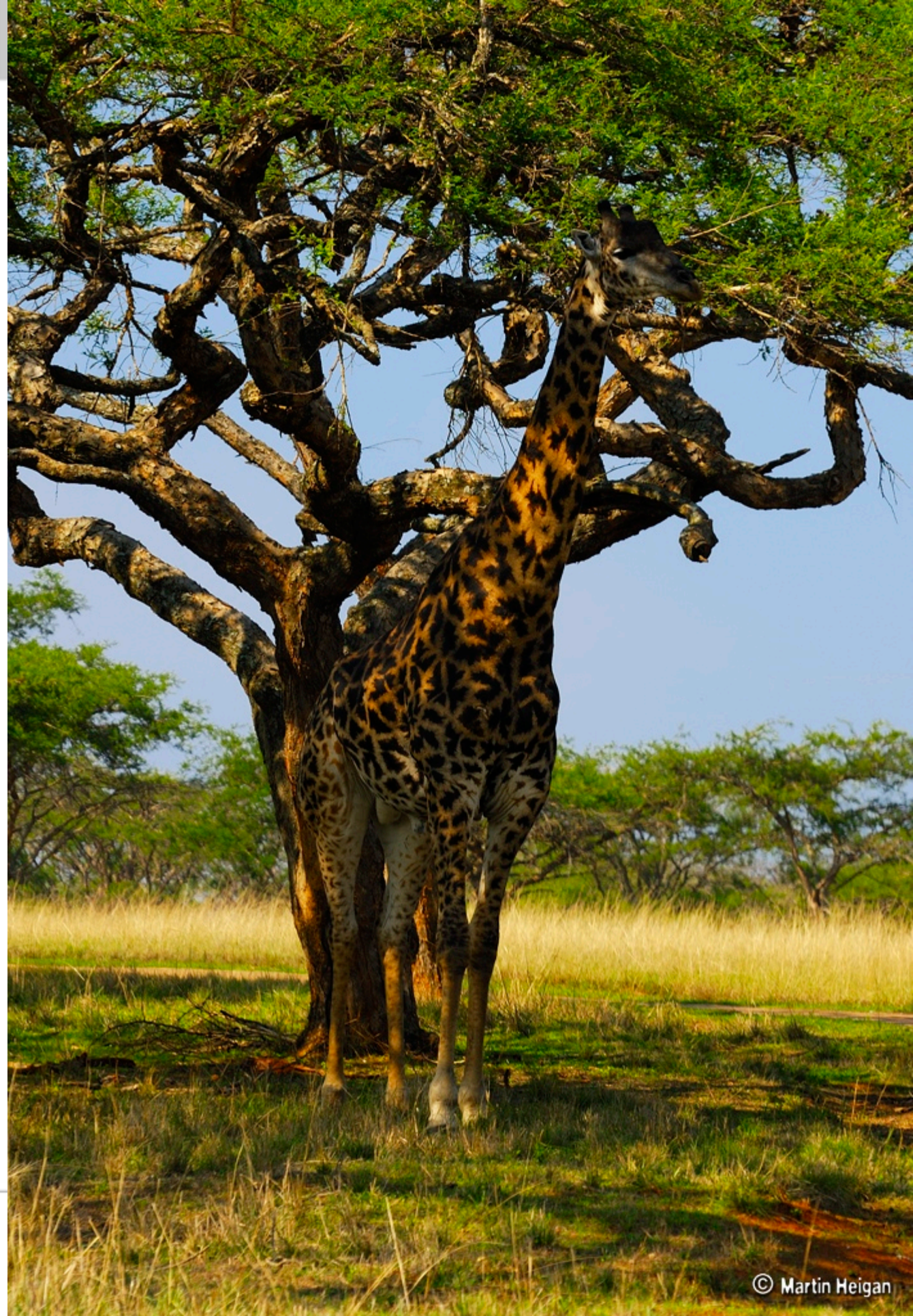
Southern leopard frog (*Rana sphenocephala*), Florida USA

European green toad (*Bufo viridis*) on pebbled concrete, Czech Republic.

Gray Tree Frog (*Hyla versicolor*) on concrete, Indiana, USA

Leaf-tailed Gecko (*Uroplatus fimbriatus*), Madagascar

Leaf-tailed Gecko (*Uroplatus henkeli*), Madagascar

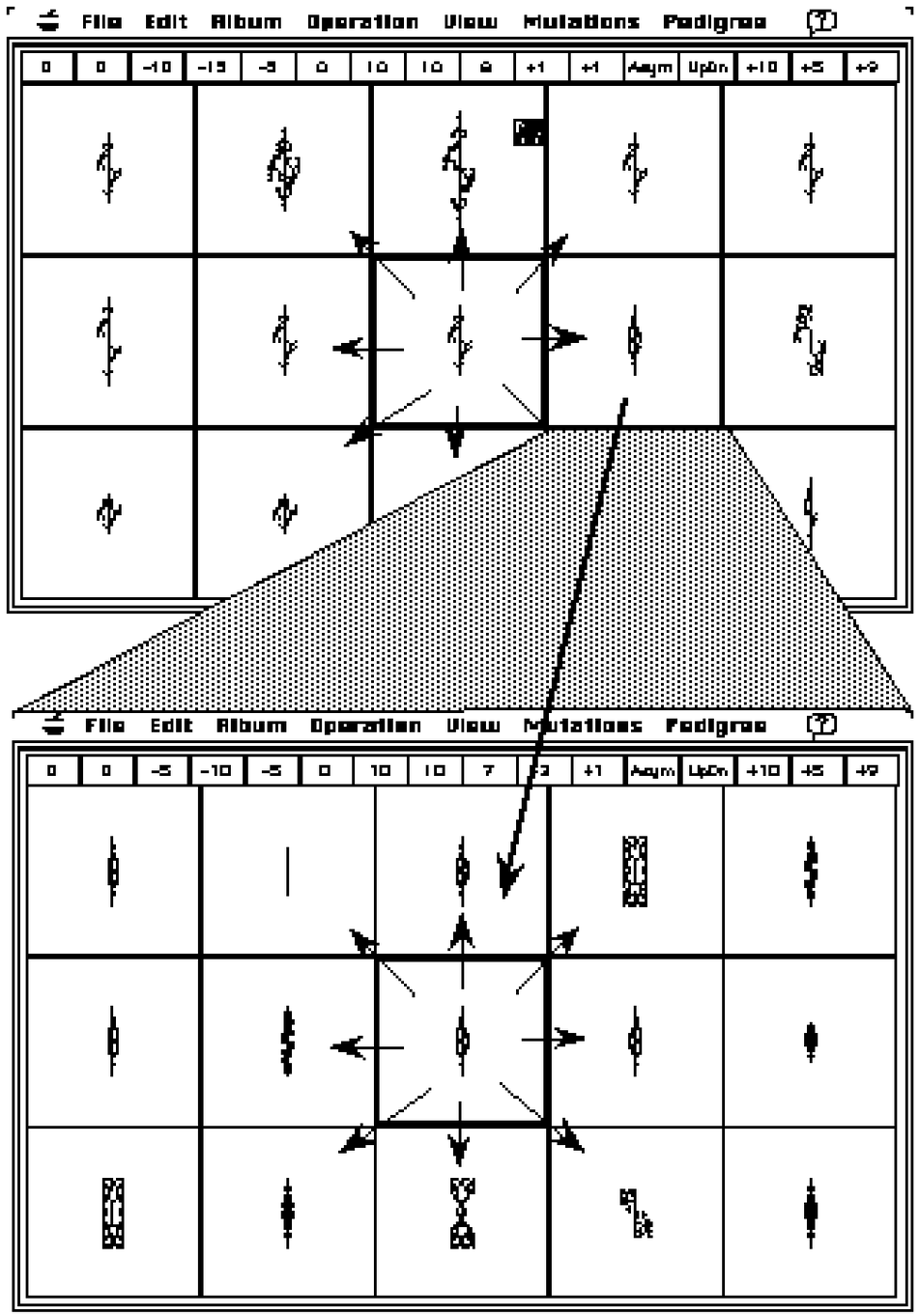


Acacia tree and giraffe (*Giraffa camelopardalis*)

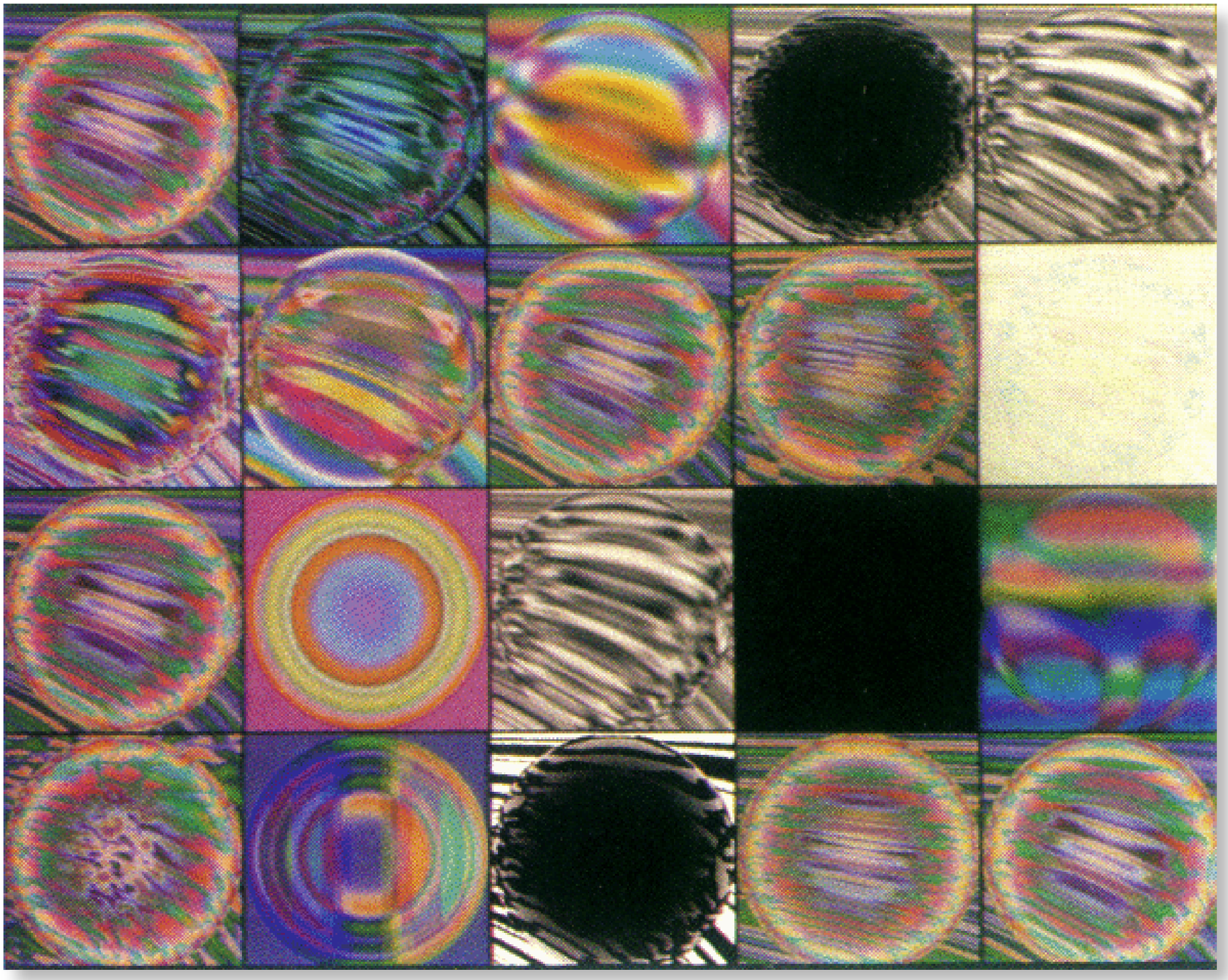
Background

Influences

- Coevolution: Hillis 1990, Angeline 1993, Funes 1998
- "Pure" procedural texture synthesis: Perlin 1985, ...
- Interactive evolution: Dawkins 1986, Sims 1991, Stanley 2008
- Reaction-diffusion: Turing 1952, Murrey 1988, Witkin & Kass 1991, Turk 1991
- Camouflage: Beddard 1895, Thayer 1909, Cott 1940, Bond & Kamil 2002, Merilaita 2003, Cuthill 2005, Schaefer & Stobbe 2006, Sherratt 2007

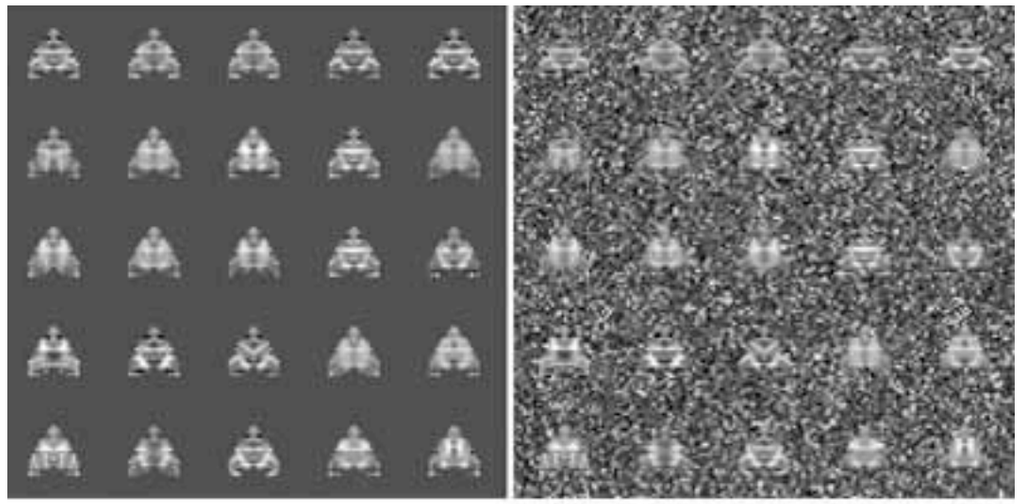


Screen shots of Dawkins' *Blind Watchmaker* software

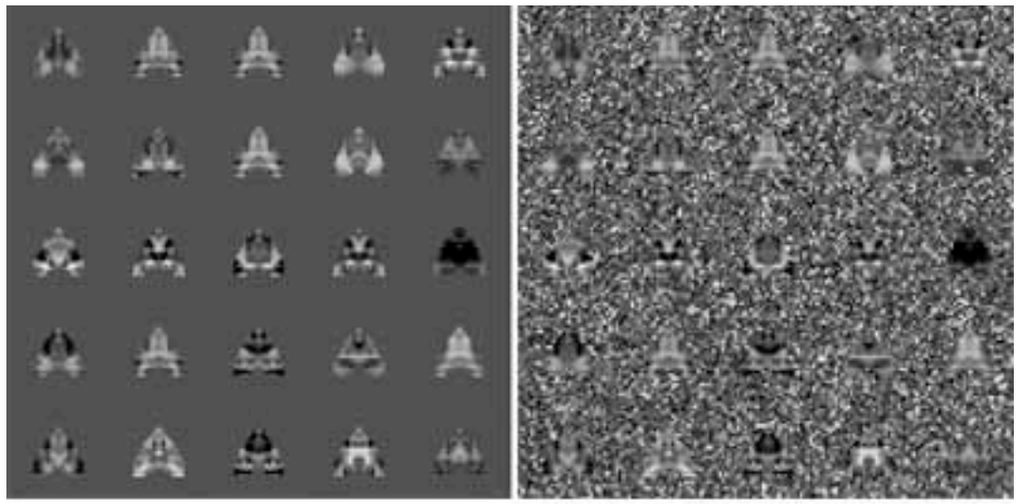


User interface for Sims' interactive evolution of color texture patterns

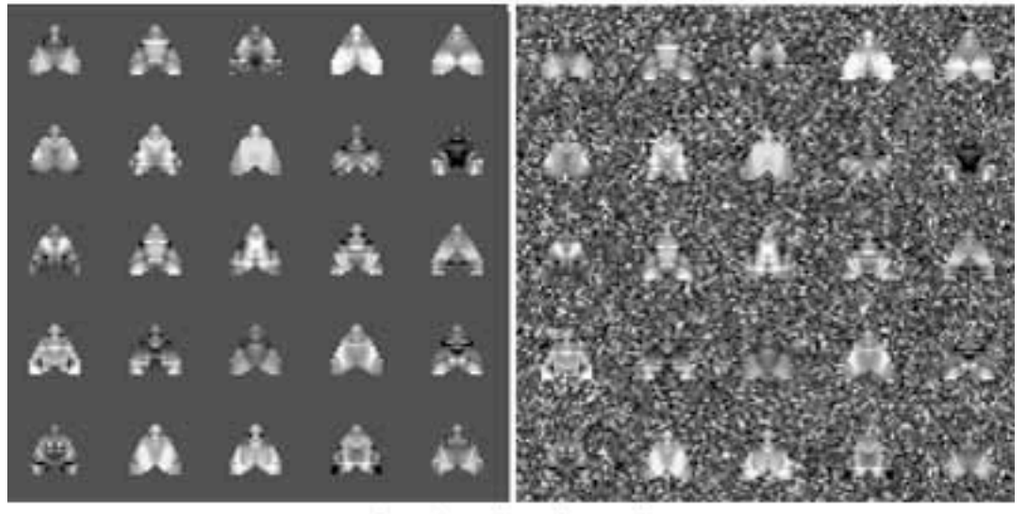
blue jay (*Cyanocitta cristata*) and display screen — Alan Bond and Alan Kamil (1998-2007)



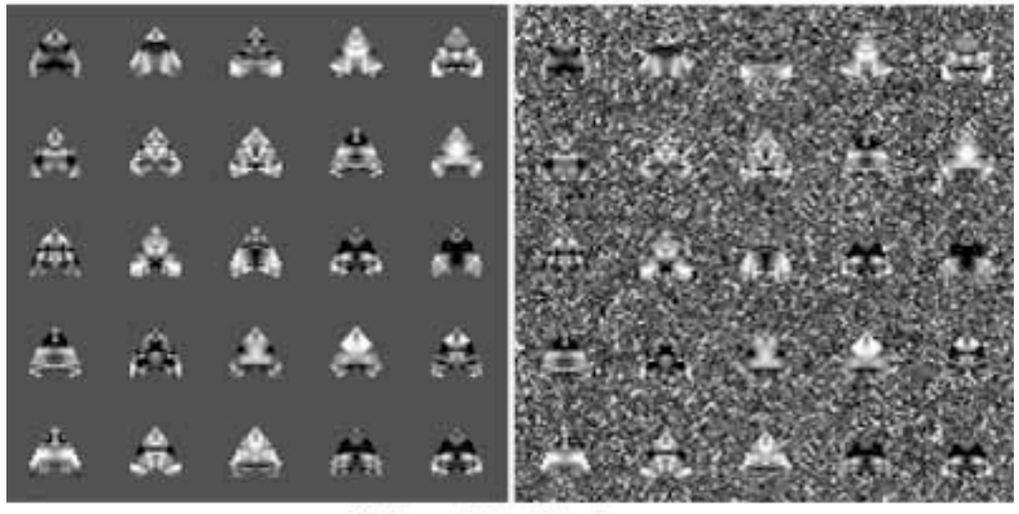
Original population



Selected by virtual predator



Randomly selected

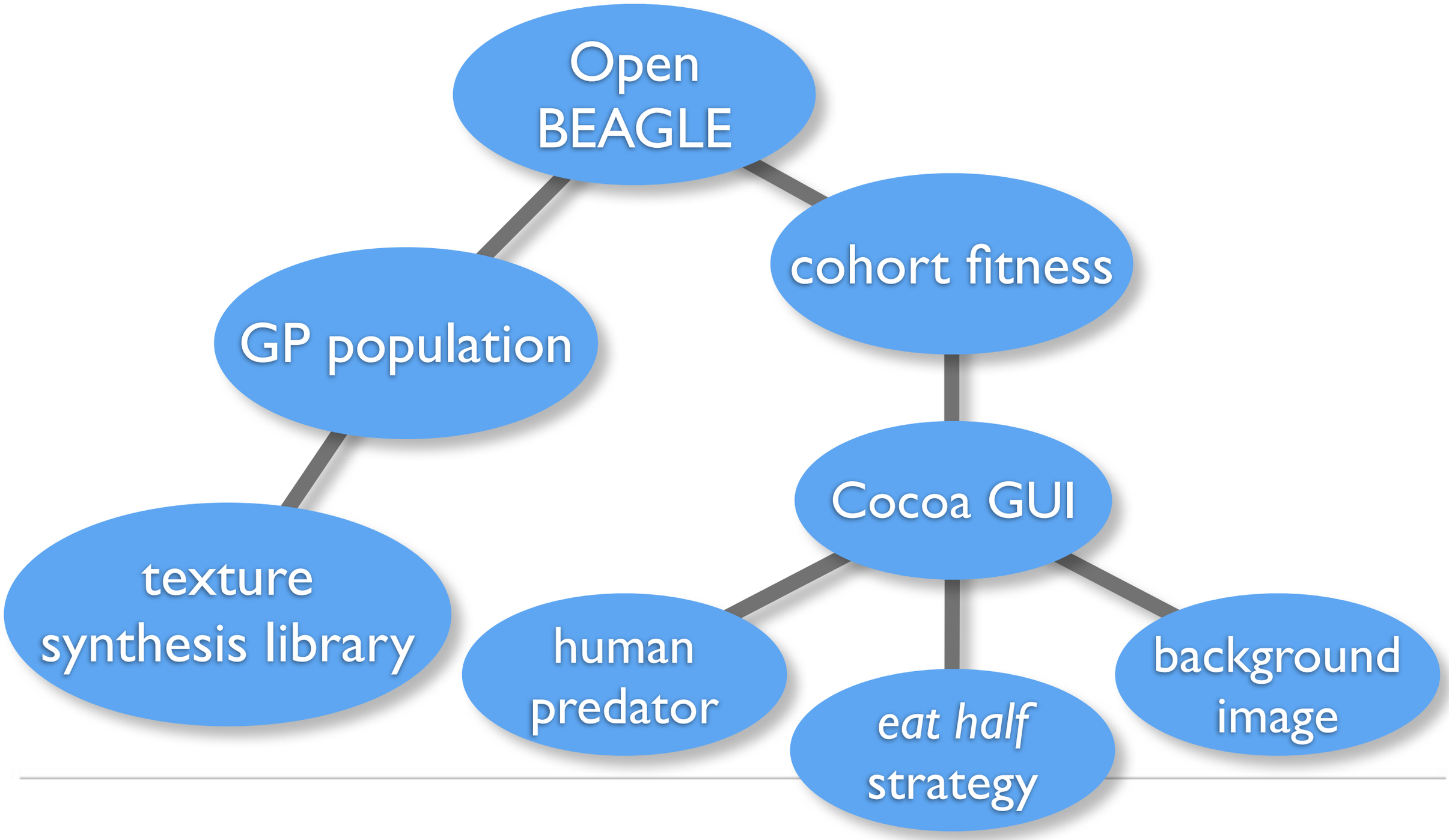


Selected by blue jays

Alan B. Bond and Alan C. Kamil (1998-2007) — evolved virtual prey

Implementation

System components

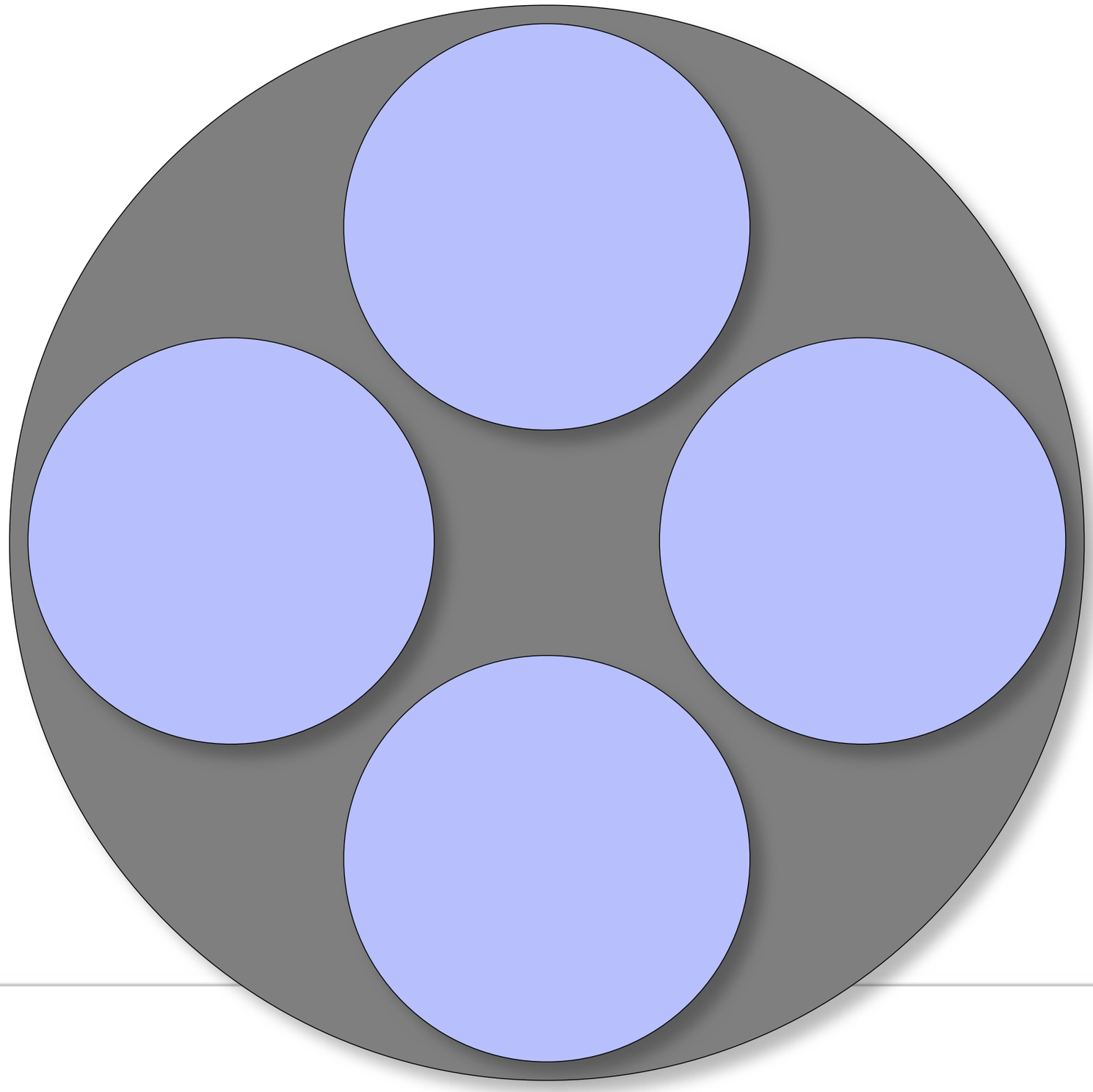


Evolutionary Computation

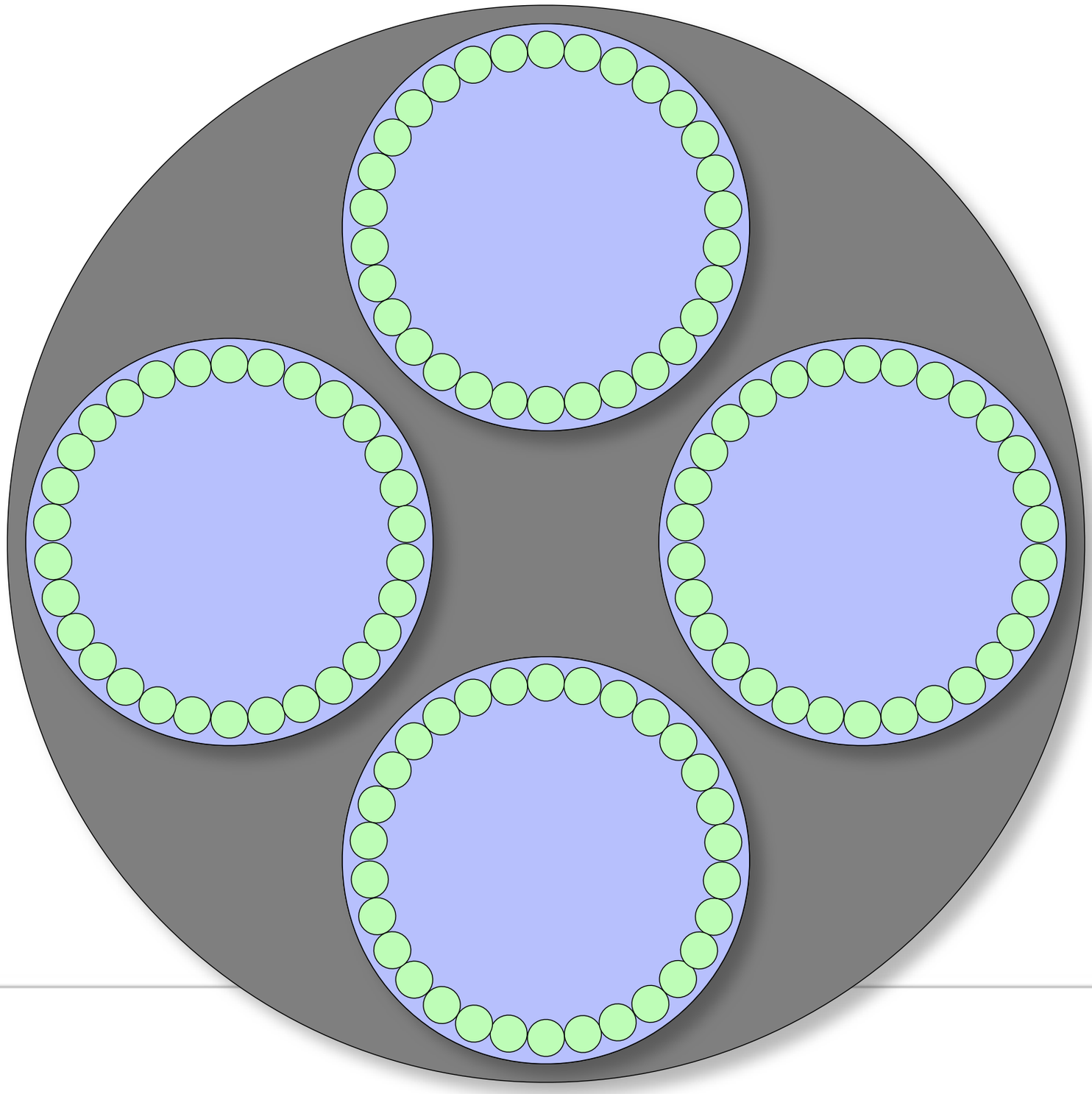
Evolutionary computation

- Genetic Programming
  - texture synthesis library
  - Open BEAGLE
- Steady-state population
  - high elitism
  - less generational / more ecosystem simulation
  - remain in breeding population until “eaten” by predator
- Interactive cohort-based fitness

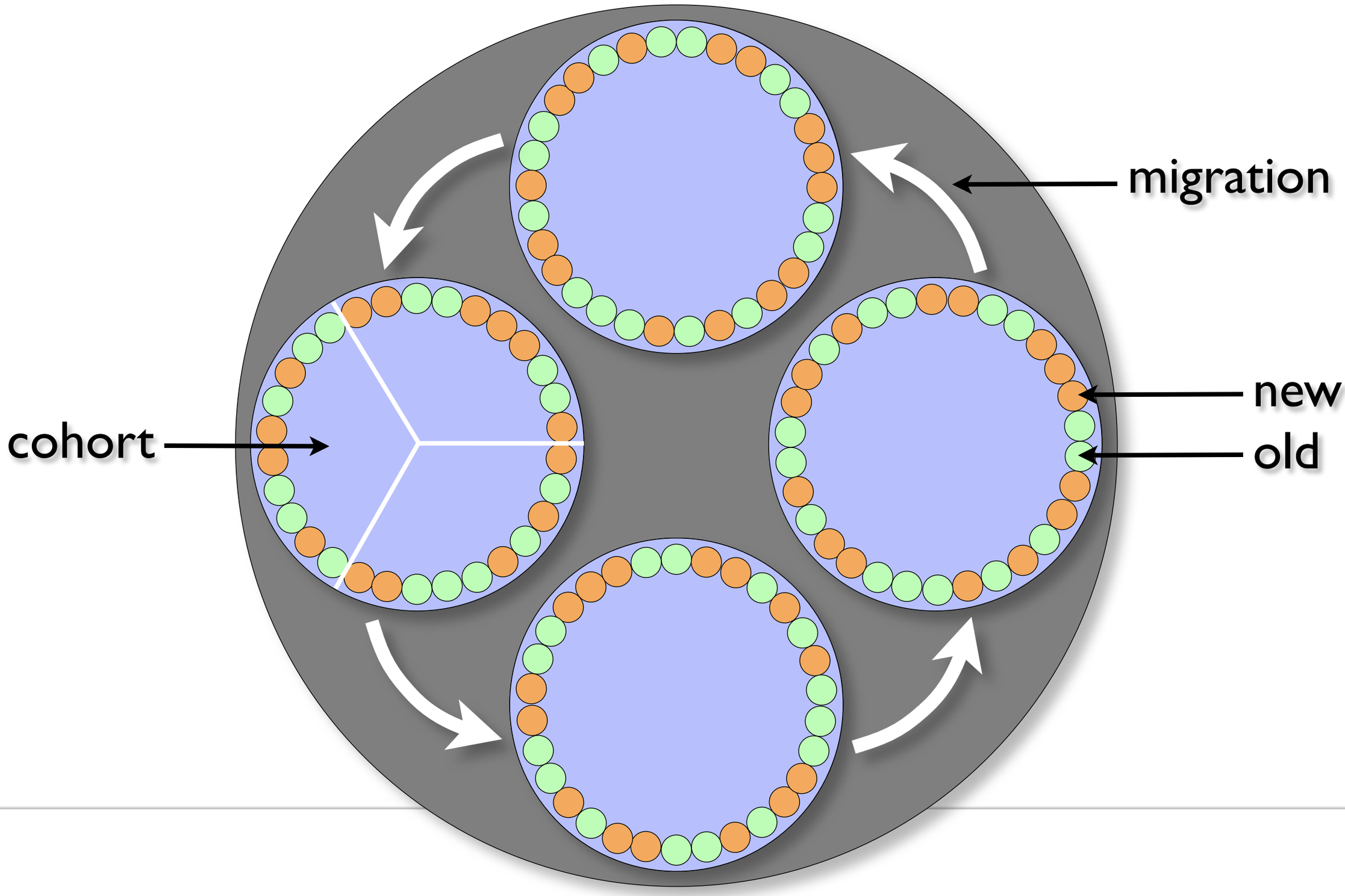
Population divided into 4 demes



4 demes of 30 individuals



cohorts, migration, elitism

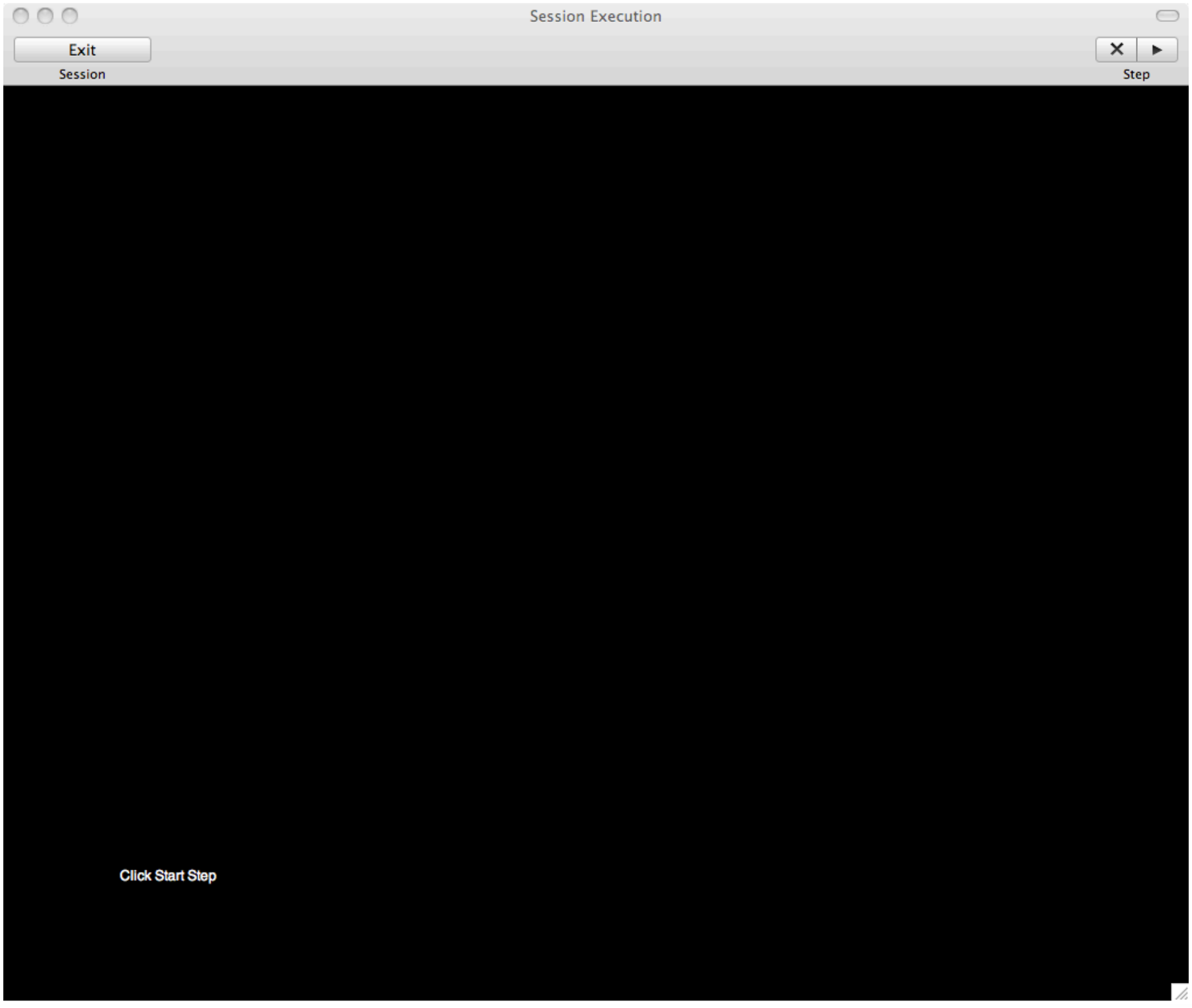


User Interaction and GUI

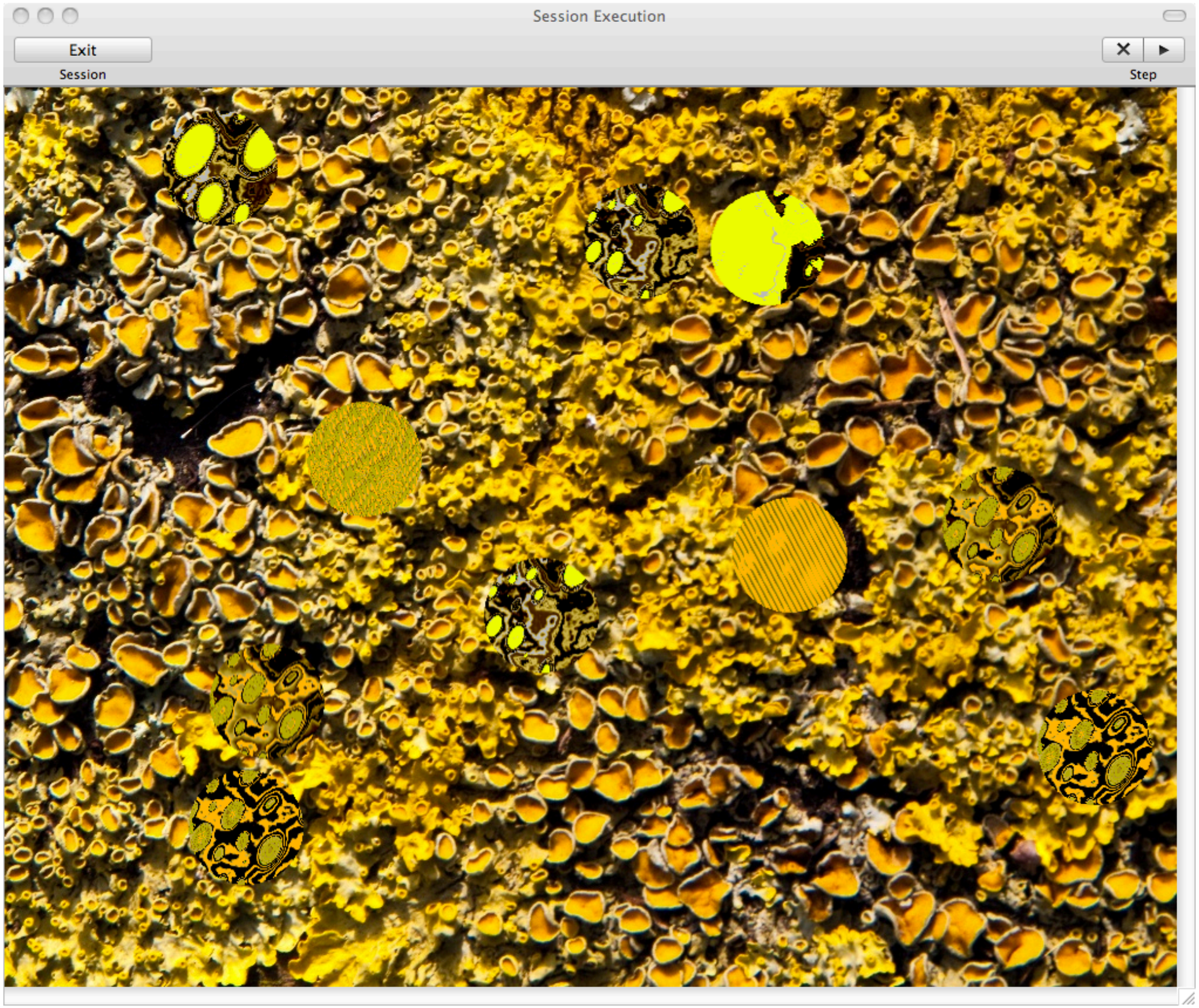
Ladybirds (10-spot (*Adalia decempunctata*), 2-spot (*Adalia bipunctata*) and cream-spot (*Calvia 14-guttata*)) on weeping silver birch tree, England.

Each round of camouflage game

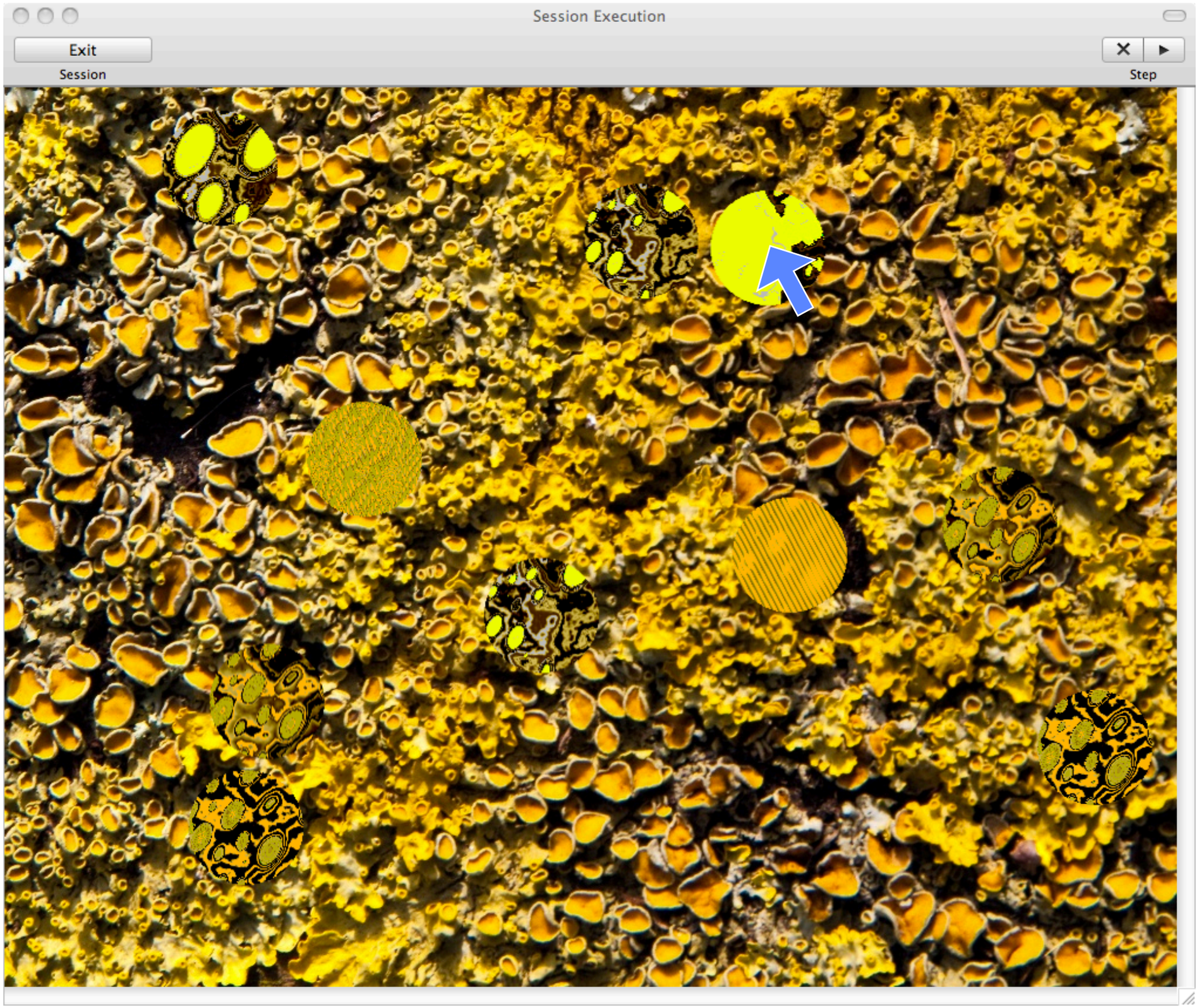
- Start with blank window, click to begin
- Background image displayed with cohort of ten prey
- Repeat five times:
  - Player/predator clicks on most conspicuous prey
  - Prey is *eaten*: removed from population and display
- End of round, blank window



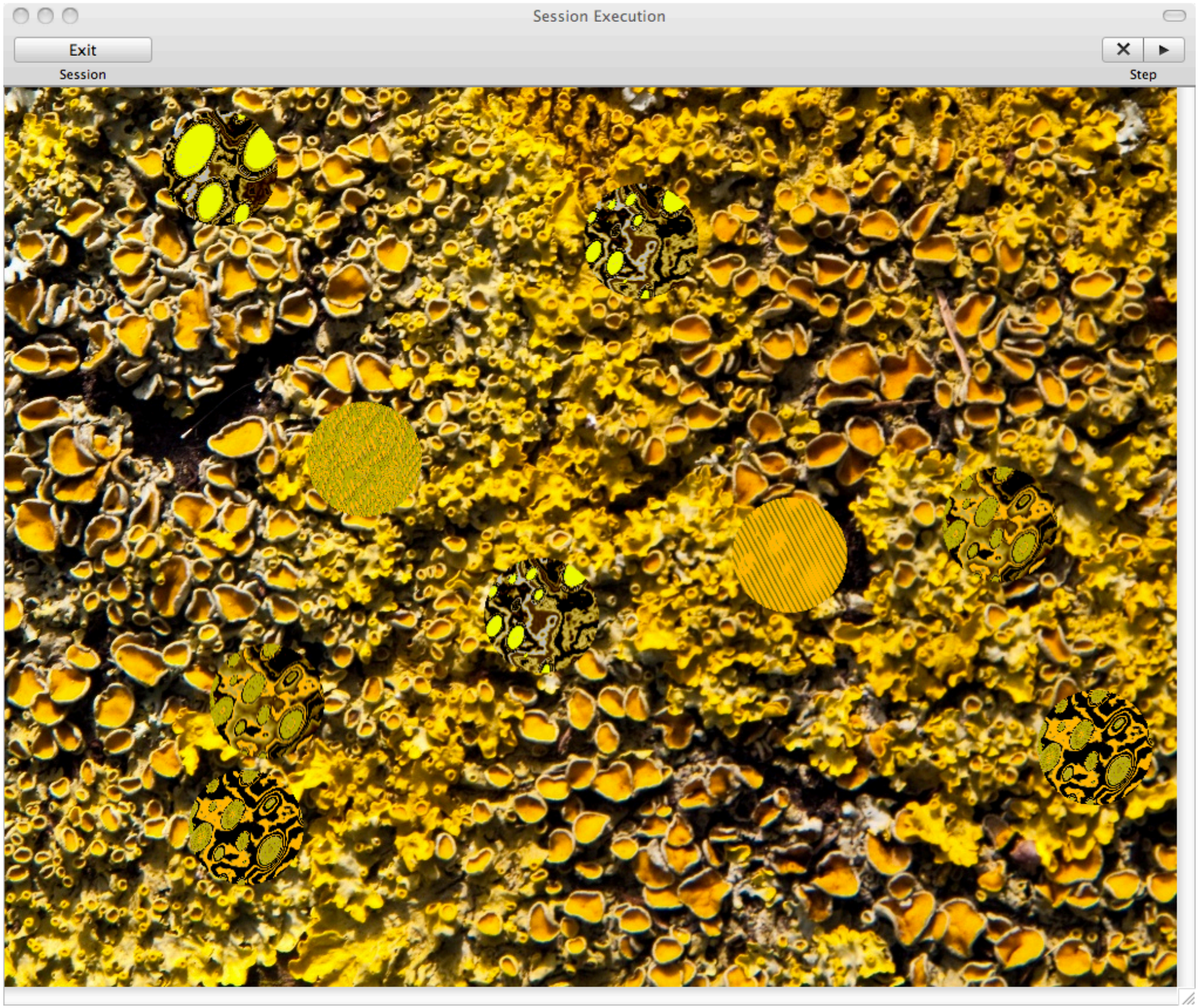
beginning of one “round” of camouflage game



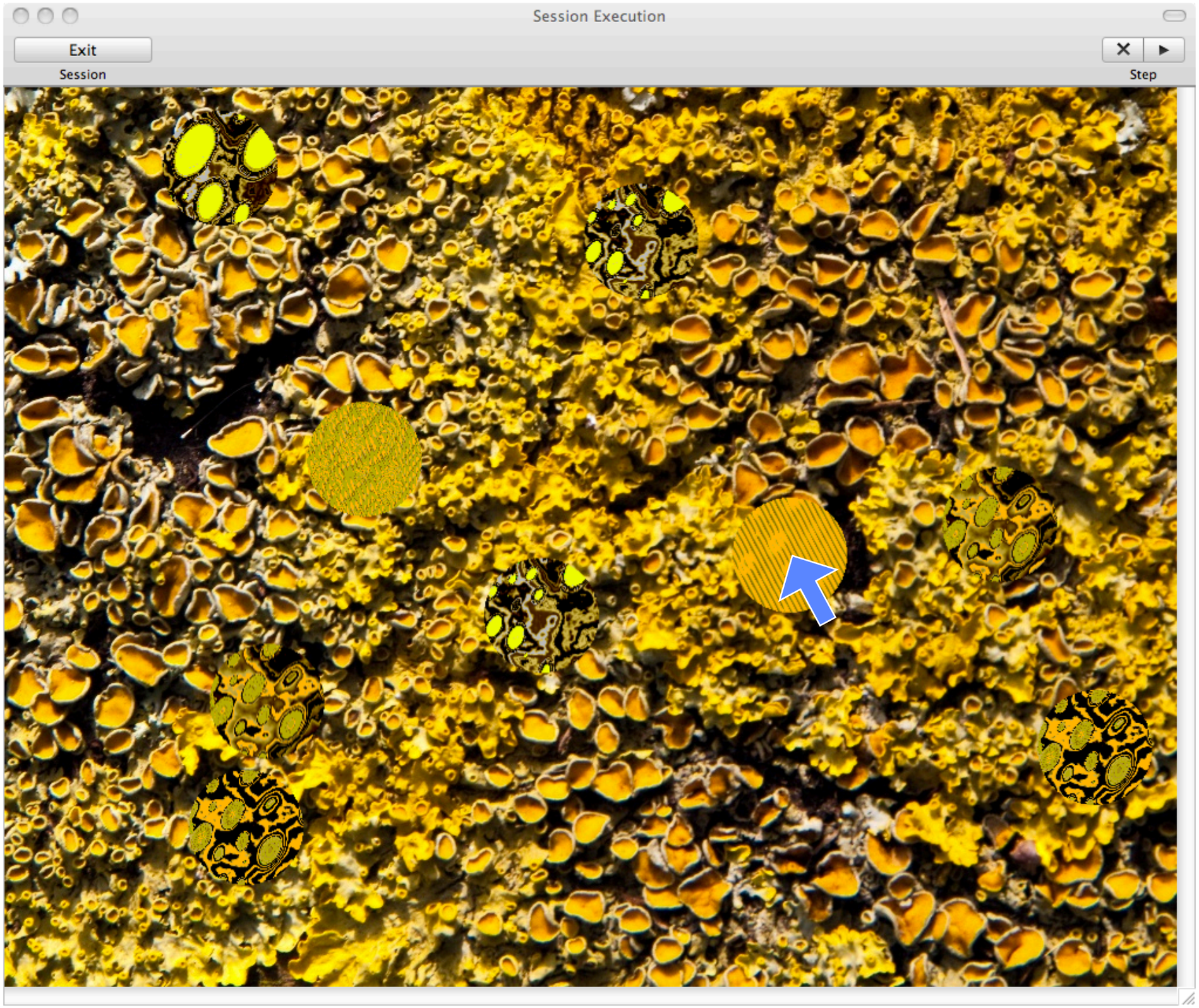
10 prey



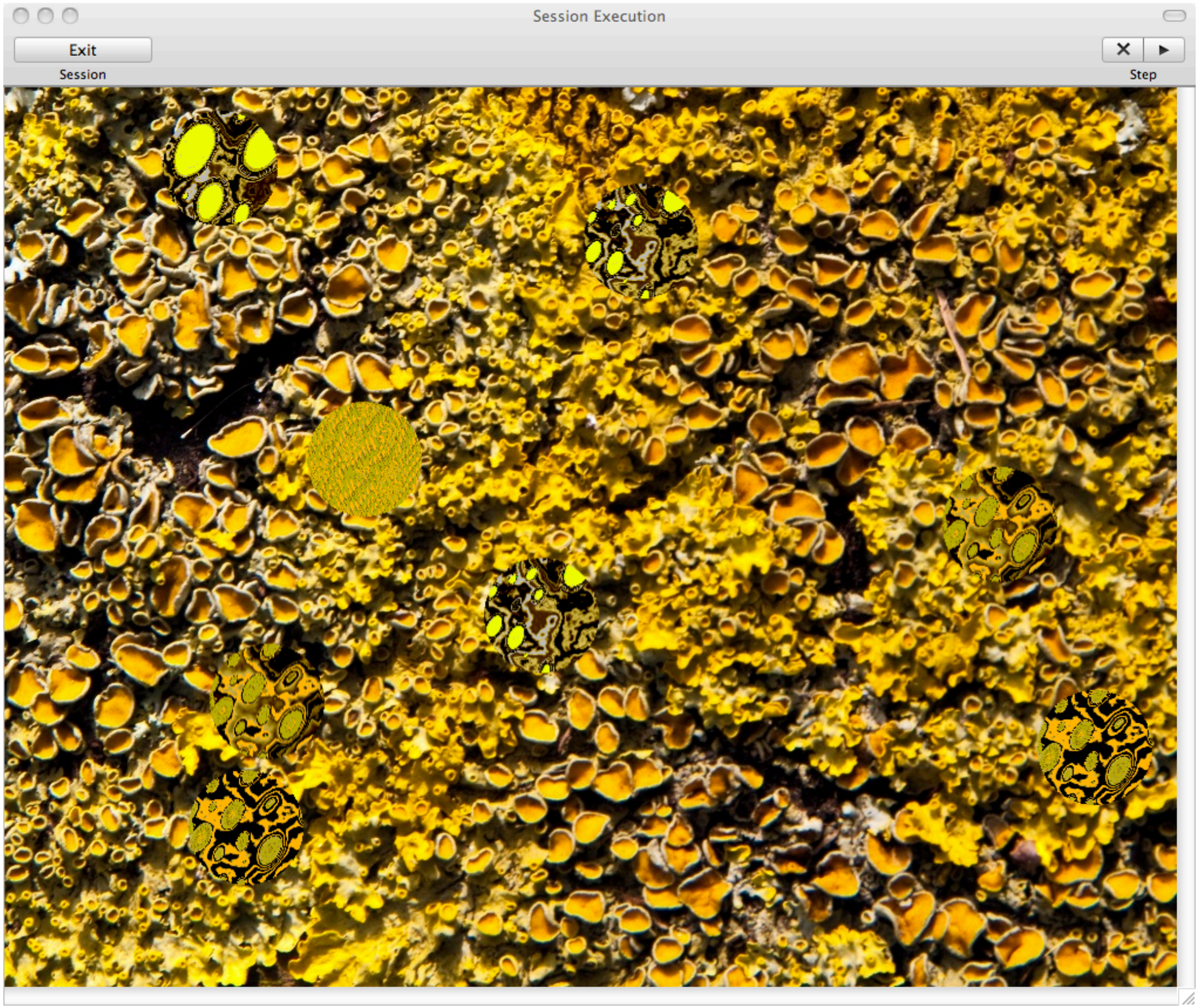
predator selects prey 1



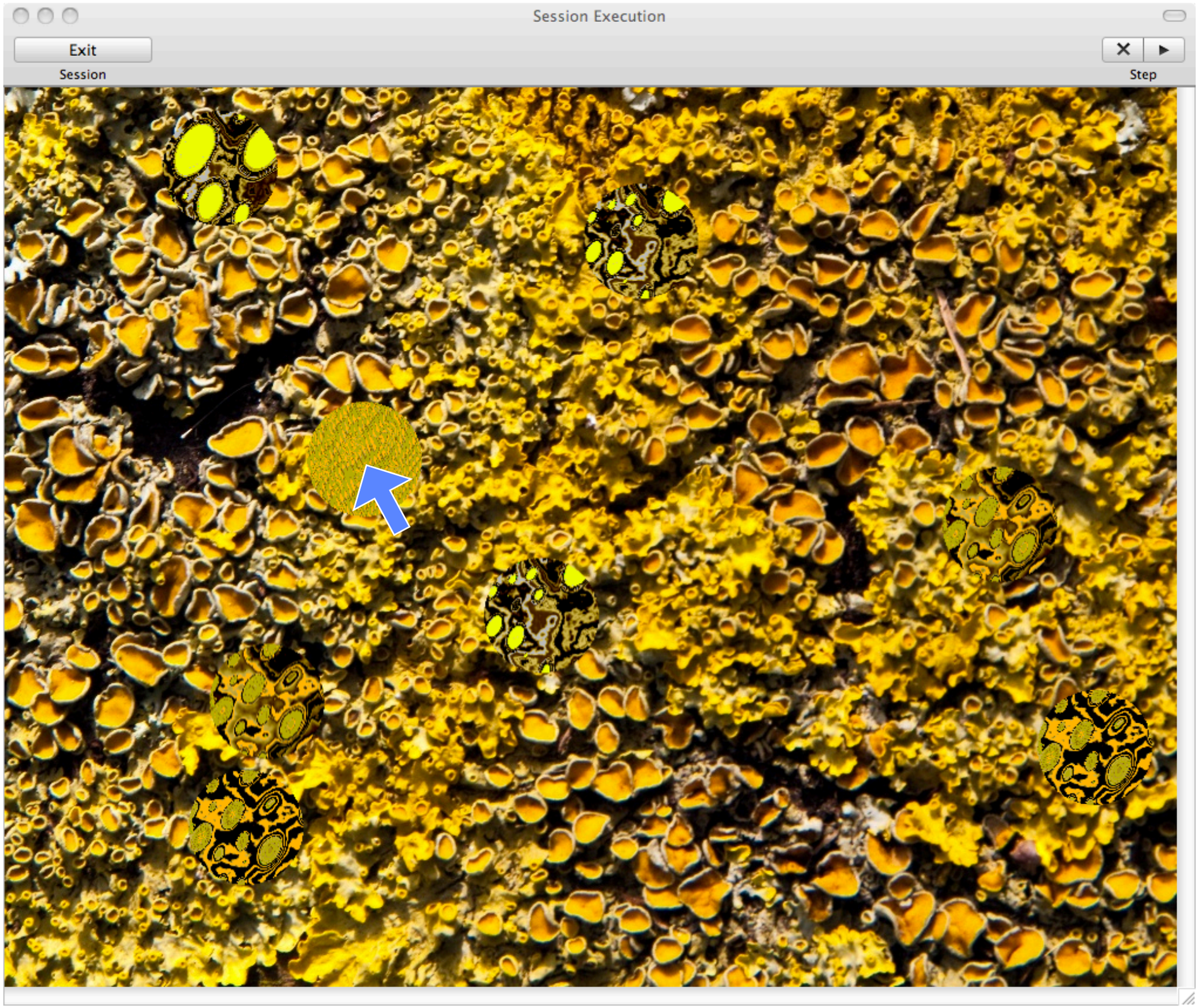
9 prey remaining



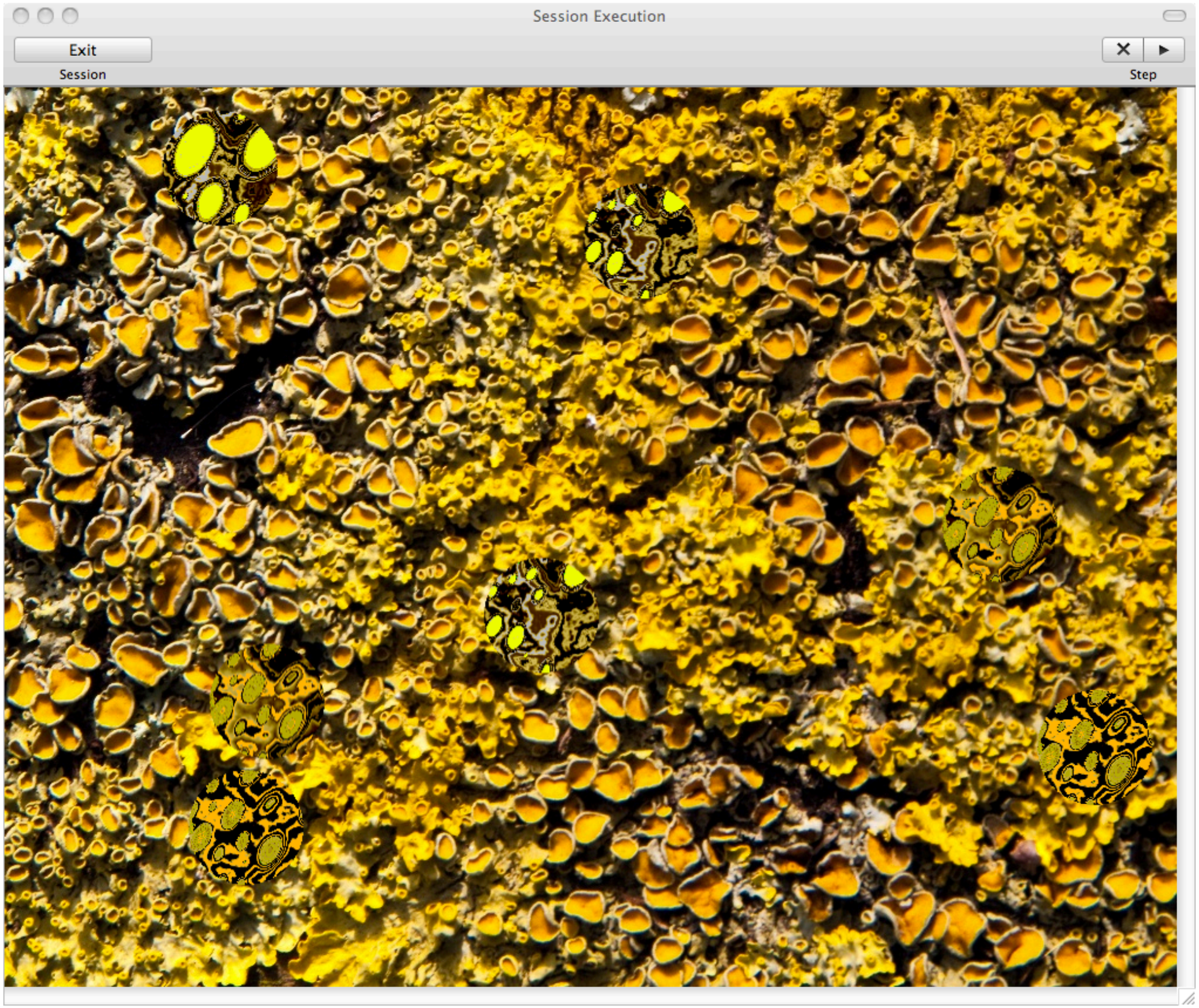
predator selects prey 2



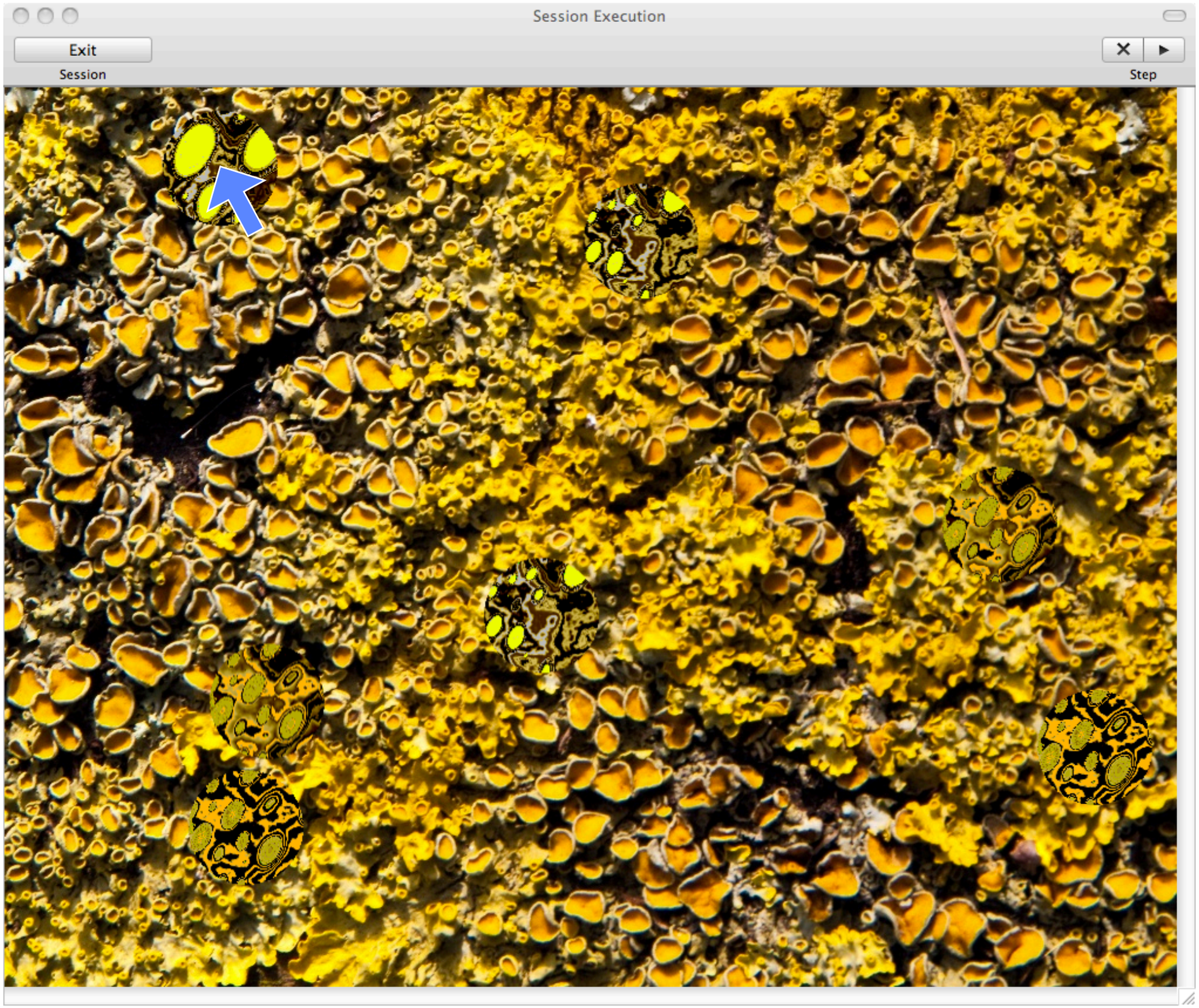
8 prey remaining



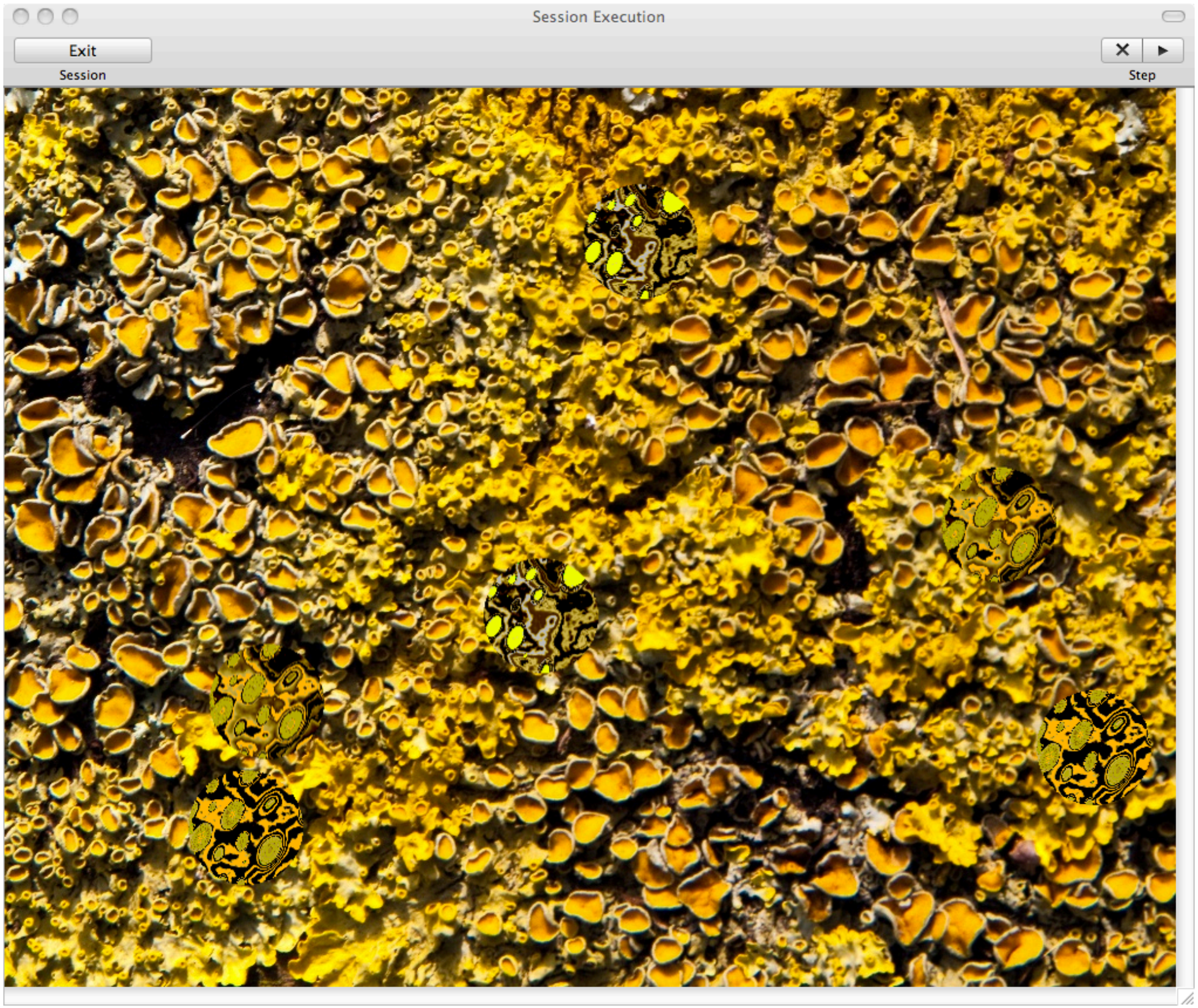
predator selects prey 3



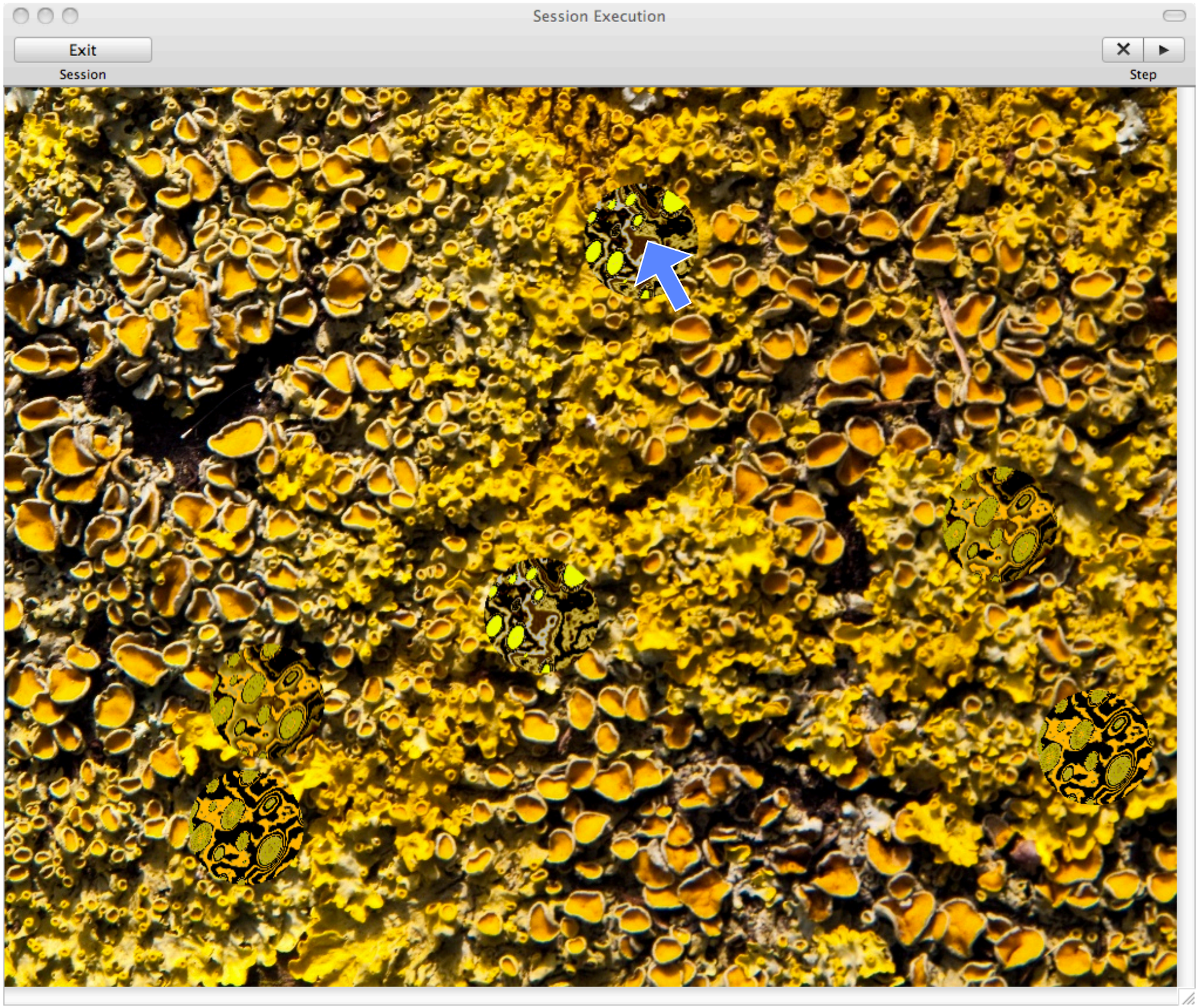
7 prey remaining



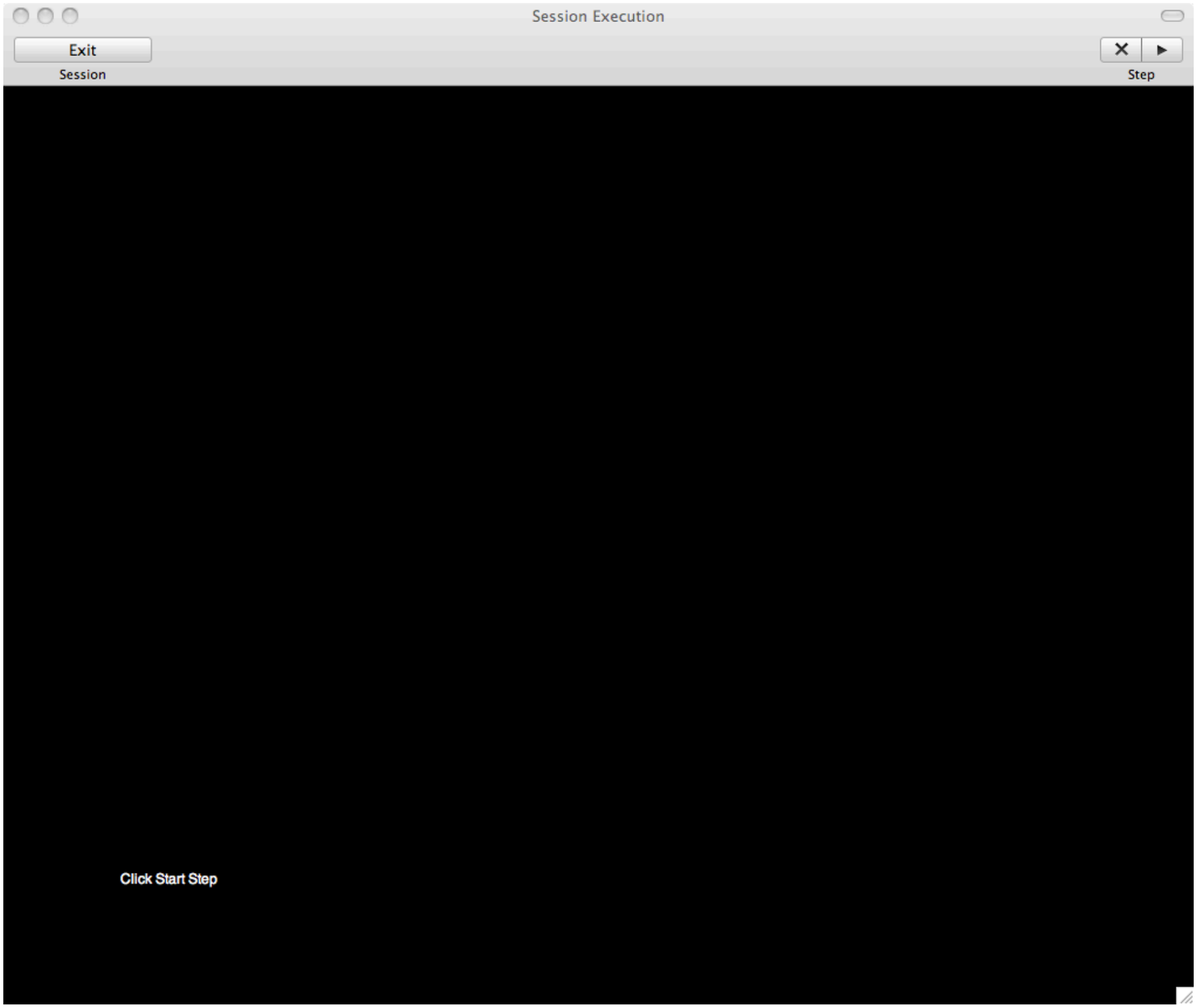
predator selects prey 4



6 prey remaining



predator selects prey 5



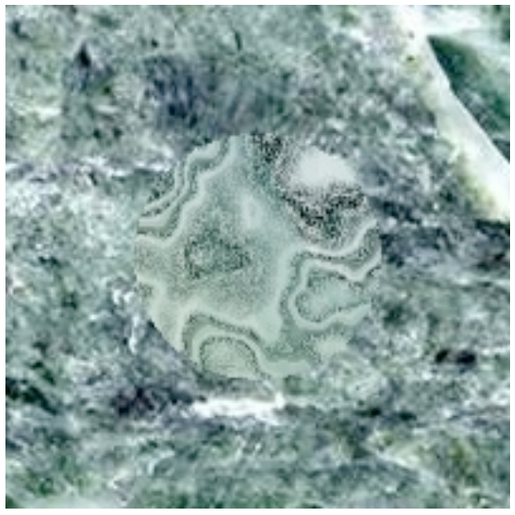
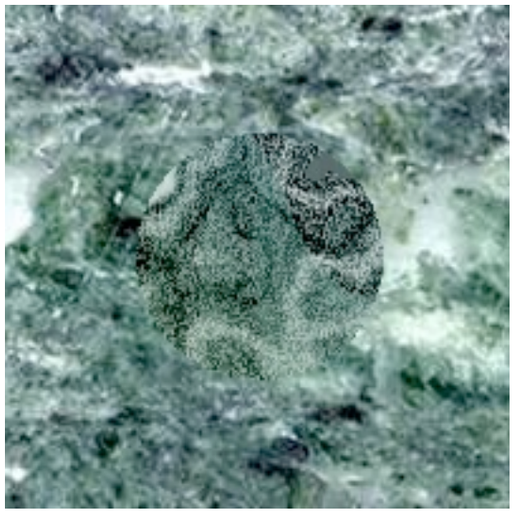
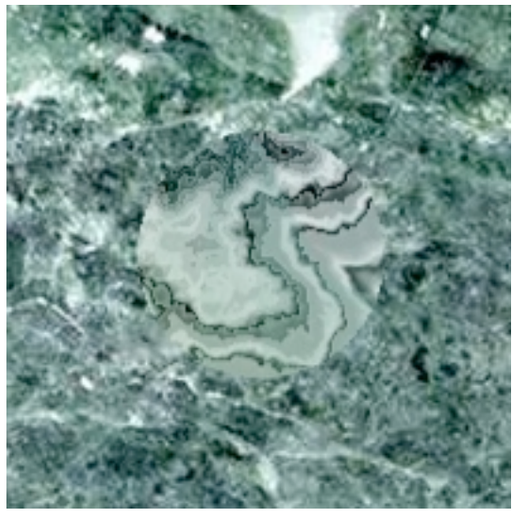
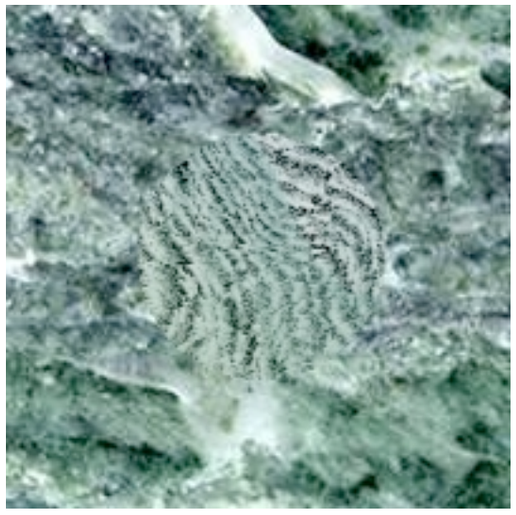
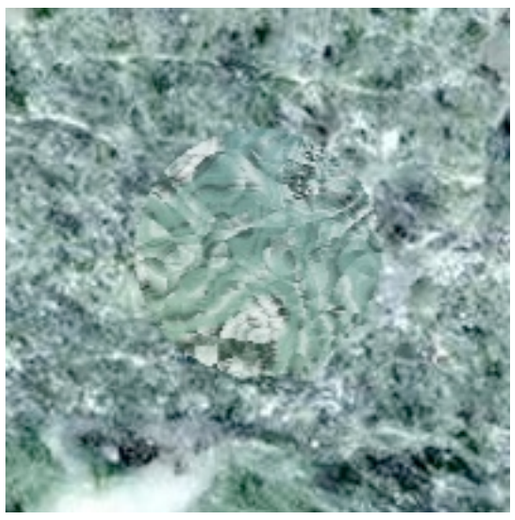
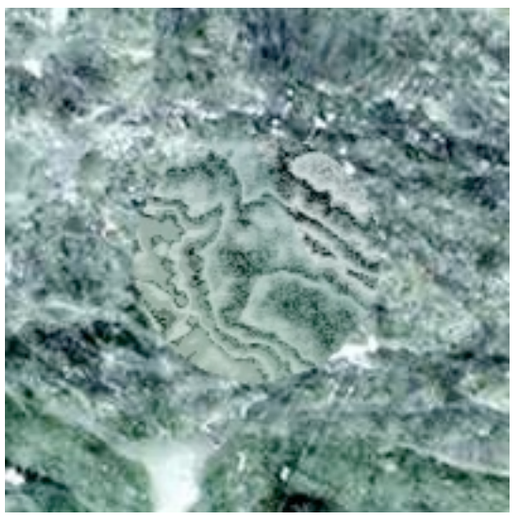
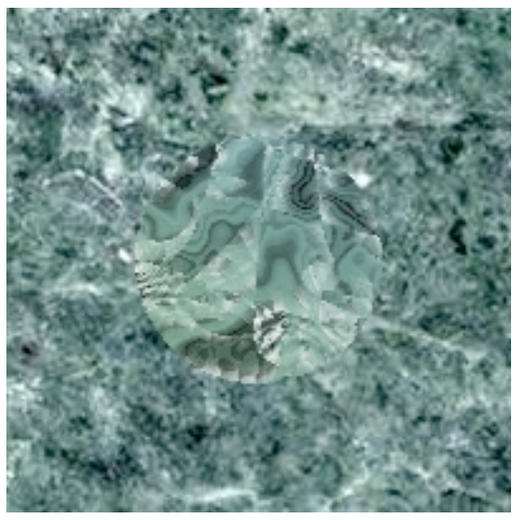
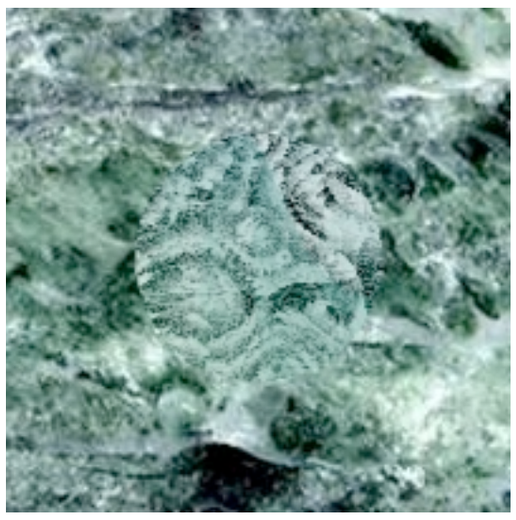
end of one “round” of camouflage game

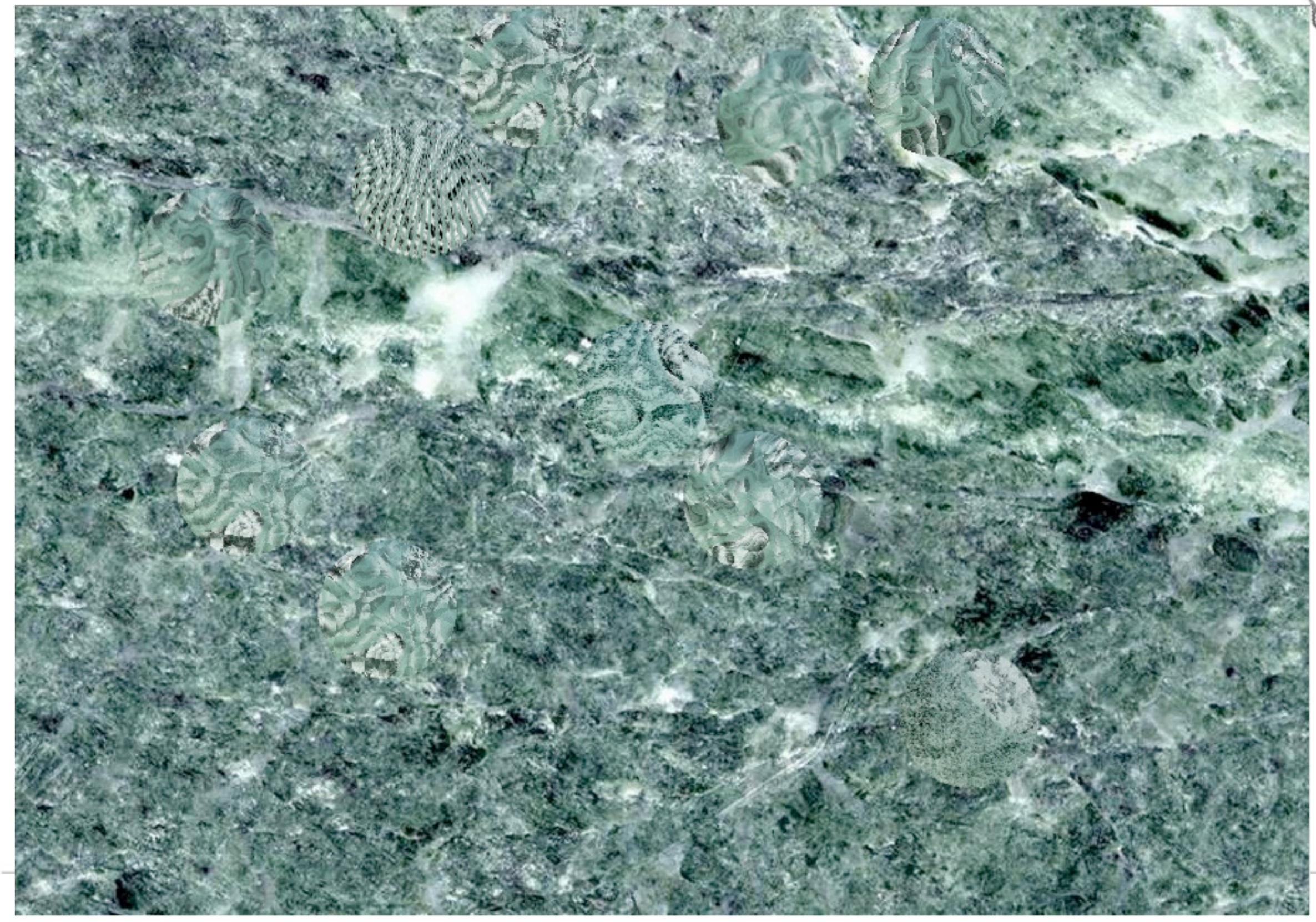
Typical run

- 1000 cohorts — sometimes 2000 or more
- 10,000 individuals fitness tested
- 83 “generations” in traditional GA/GP (p=120)
- **5000** mouse clicks by human predator
- 3 hours of steady work — usually spread over weeks

Results

Serpentine (polished stone)

Flowers and leaves

(*lantana montevidensis* in my backyard)

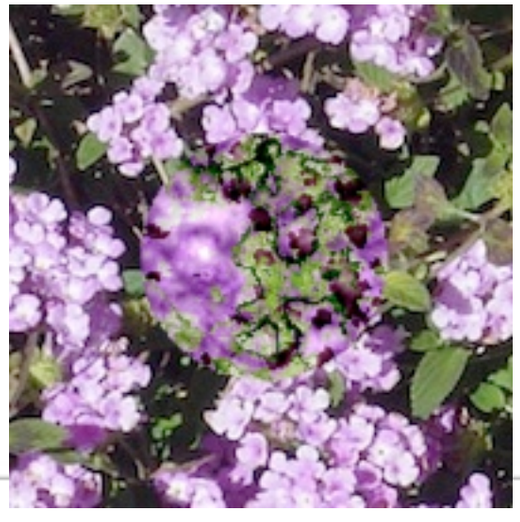




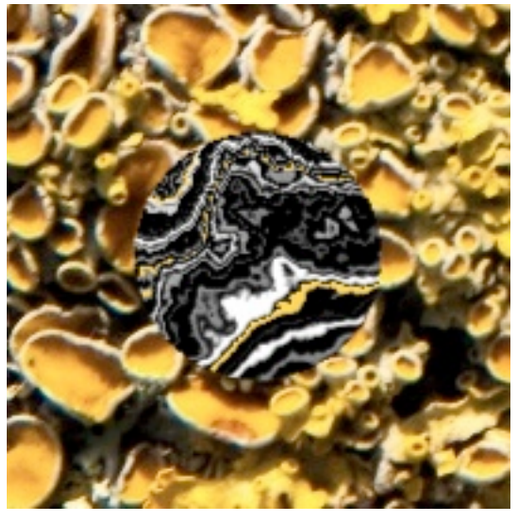
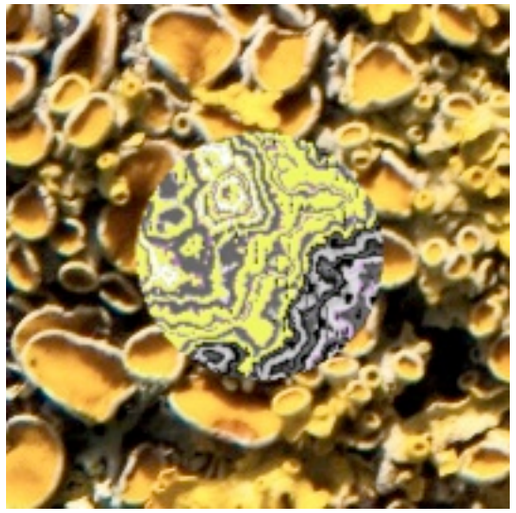
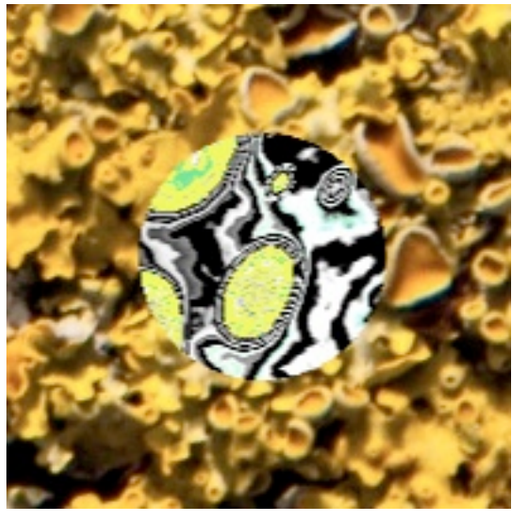
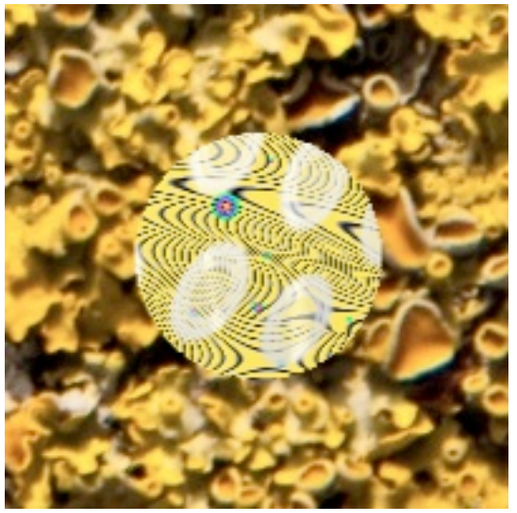
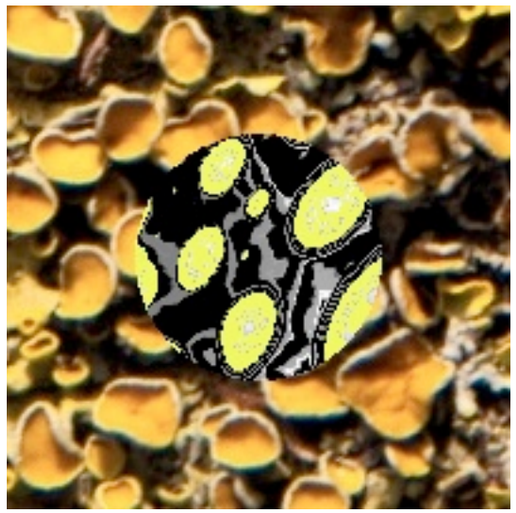
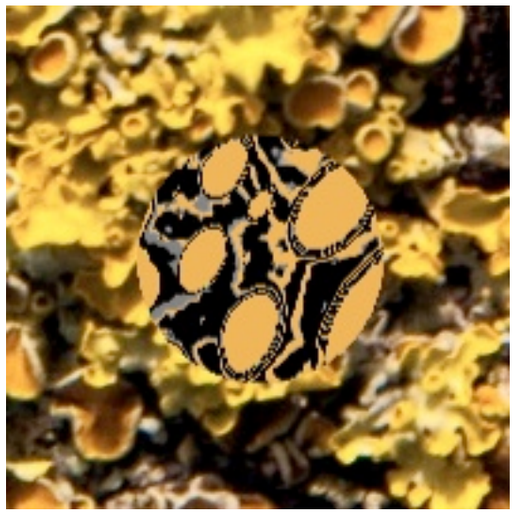
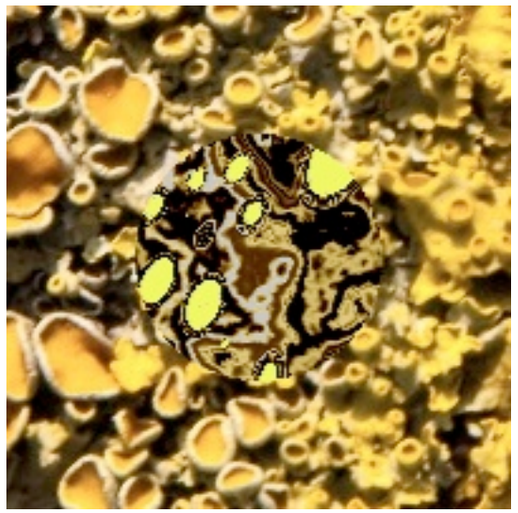
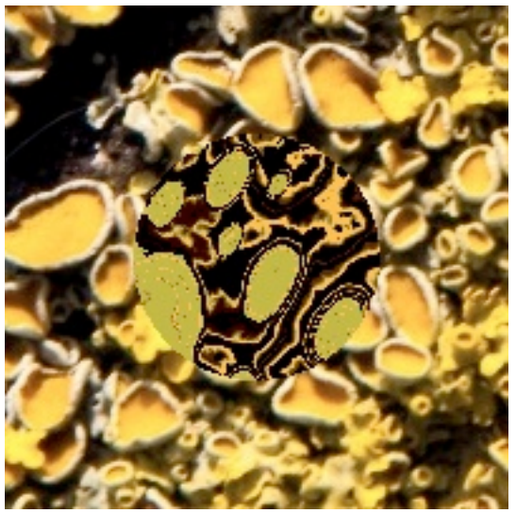
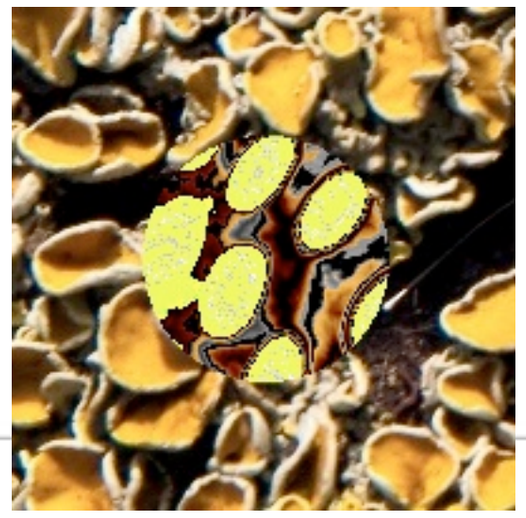
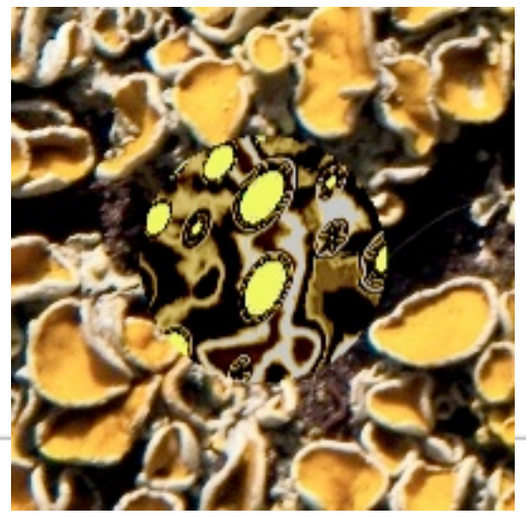
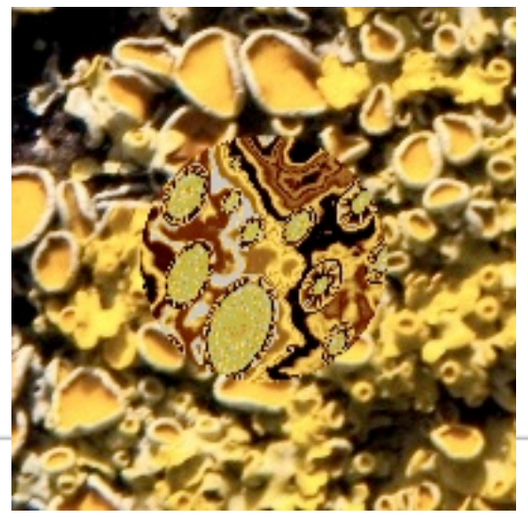
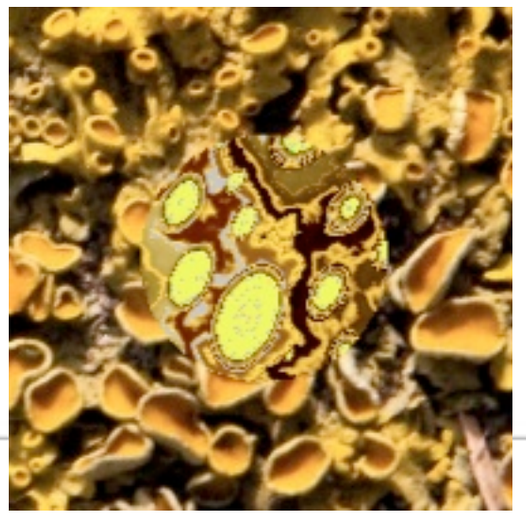




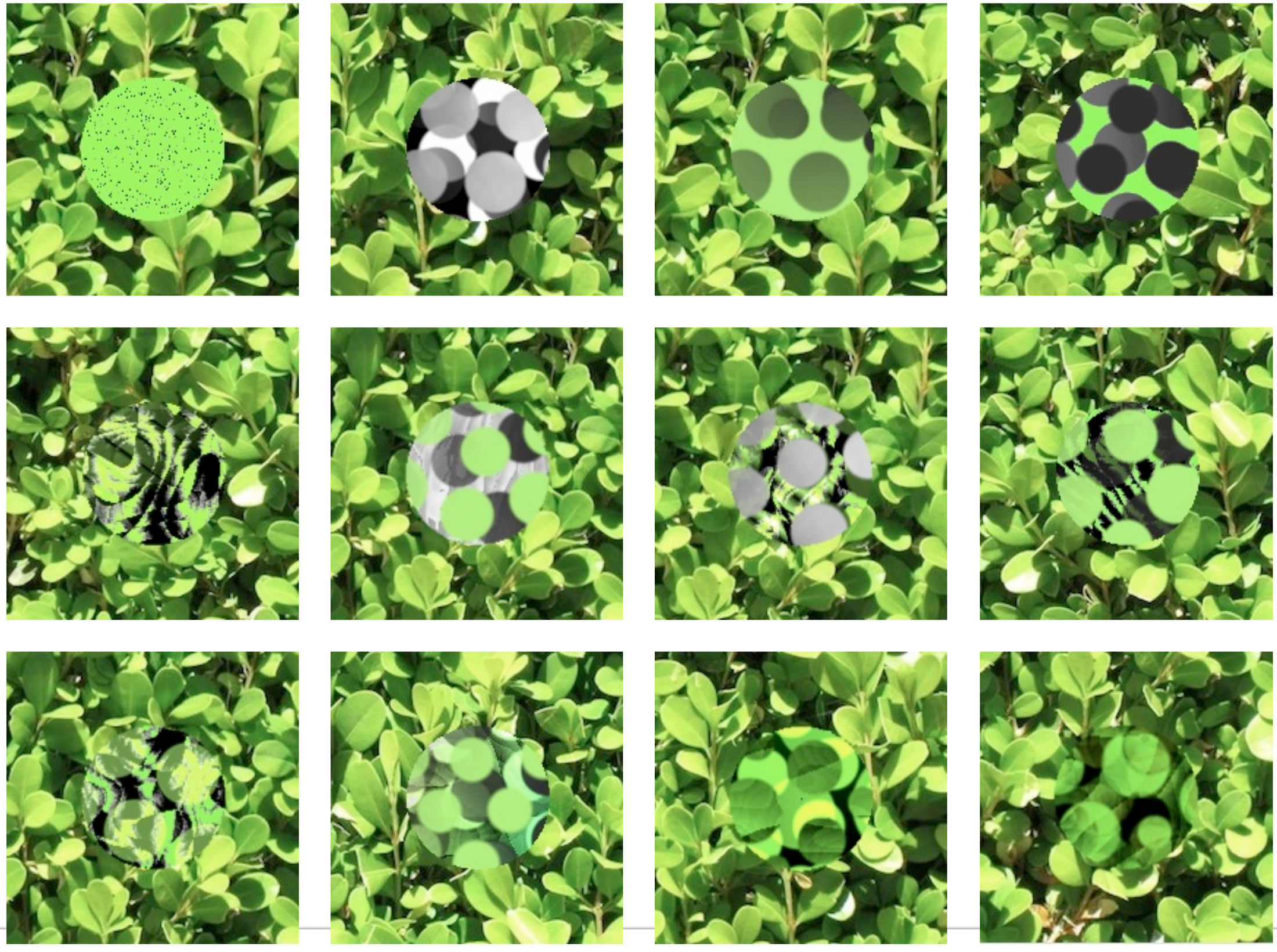


Lichen

Hedge



(shadows?!)

Peppers
(unsuccessful run)

Acknowledgements

- Research sponsor:
  - Sony Computer Entertainment
    - Dominic Mallinson (VP US R&D)
- GUI implementation: Bjoern Knafla
- Advice: Christian Gagné, Daniel Weinreb, Iztok Lebar Bajec, James O'Brien, Lance Williams, Ken Perlin.

Thank you!

craig_reynolds@playstation.sony.com
http://www.red3d.com/cwr/gots/