// Revealing the Simple Complexity

// Revealing the Simple Complexity

By Armaghan Behzad Behbahani

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Architecture
In
Architecture

Paola Zellner Bassett
.....Committee Chair

Dane Webster
.....Committee Member

Dave Dugas
.....Committee Member

July/7/2017
Blacksburg, VA

Keywords: Persian Patterns, Generative Design, Interactive Design

Abstract {

{ In a world were original ideas are hard to find, it is important to speculate the excising methods and seek inspiration. }

// “Revealing the simple complexity” is a series of explorations which investigates the wonders of Persian patterns and develops its principles through generative and interactive design.

//The exploration consists of four main stages of studies and manifestations.

[1] Fundamental studies, delves into the mathematical detail and principle of traditional Persian patterns, looking at the underlaying rules of the construction and composition of the patterns.

[2] Generative coding and modeling, investigates ways in which traditional pattern algorithms can be simplified and reconstructed. This stage declares the main body of my thesis and includes producing work which explores techniques for form-finding using algorithm and code.

[3] Interactive studies, are another important component of the process. The interaction between designer/user, code/design has been investigated though motion and light sensors in this phase.

[4] A conclusion of all the previous stages have been gathered in the form of architectural manifestations and future application in this chapter.

//My explorations have tried to uncover simple ways in which we could rethink complicated form finding strategies, and suggests a new direction for future explorations in interactive and generative architecture.

}

Acknowledgment {

// I would like to dedicated my thesis to my loving parents, Abbas and Afsaneh, my wonderful sister Arezoo and brother Ahmadreza, for their endless love, support and sacrifice. I could not have done this without you.

// Thank you to my caring and true friend Ahmadreza for all the love, patience and motivation and for standing beside me through my toughest times.

// I would also like to thank my committee:
Paola, for her effort and guidance in pushing me to go beyond the boundaries and breaking the trivial paths.
Dane, for his encouragements and believing in me, to code my way through.
And Dave for his valuable guidance and support.

// And thank you to all my friends and extended family here in the United States and back home for their kindness and understanding.

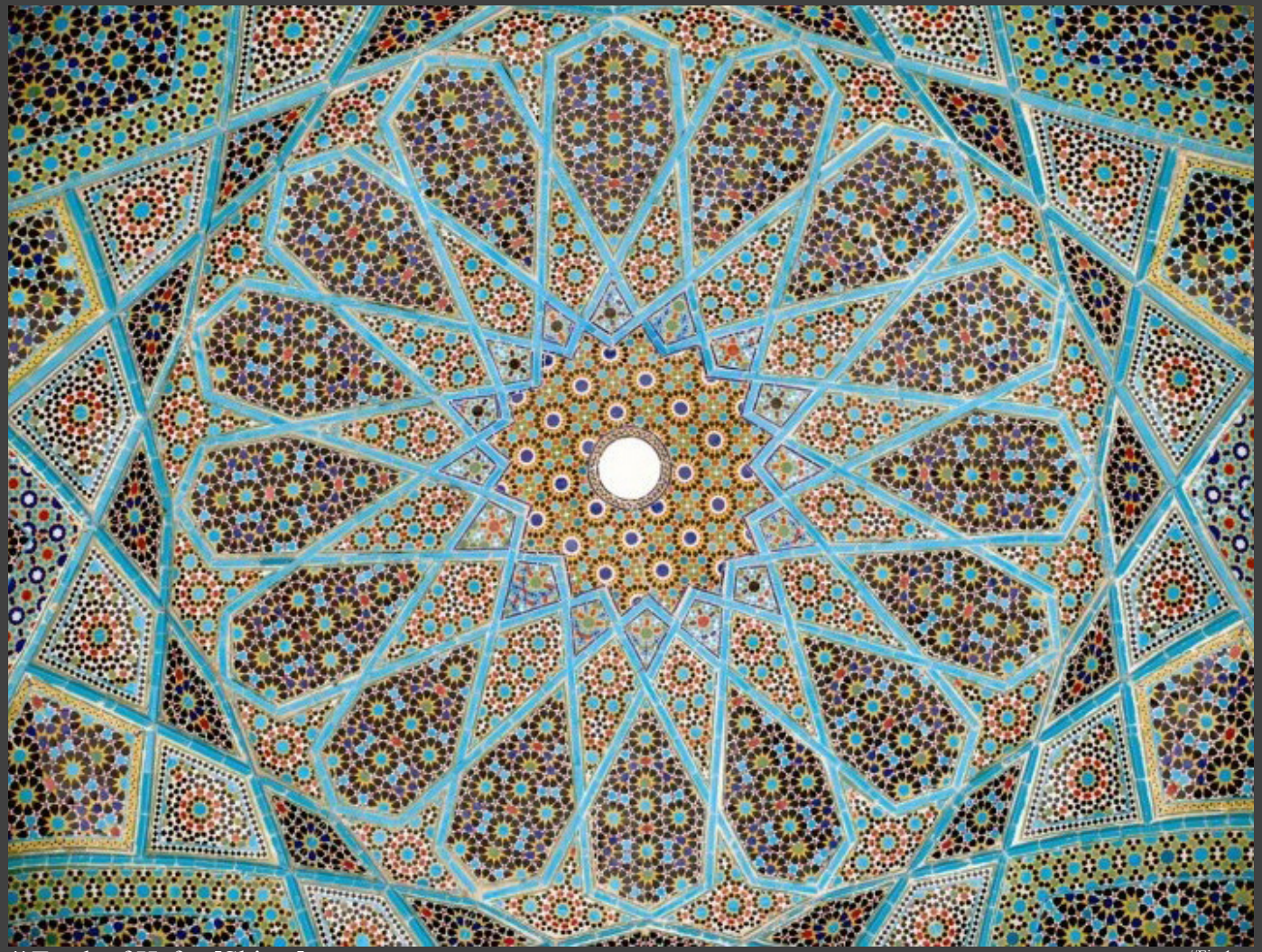
}

Contents {

}

1 // Fundamental Studies

Persian Patterns as Precedents



// Tomb of Hafez [Shiraz]

//Fig 1.

// Shah Mosque [Isfahan]

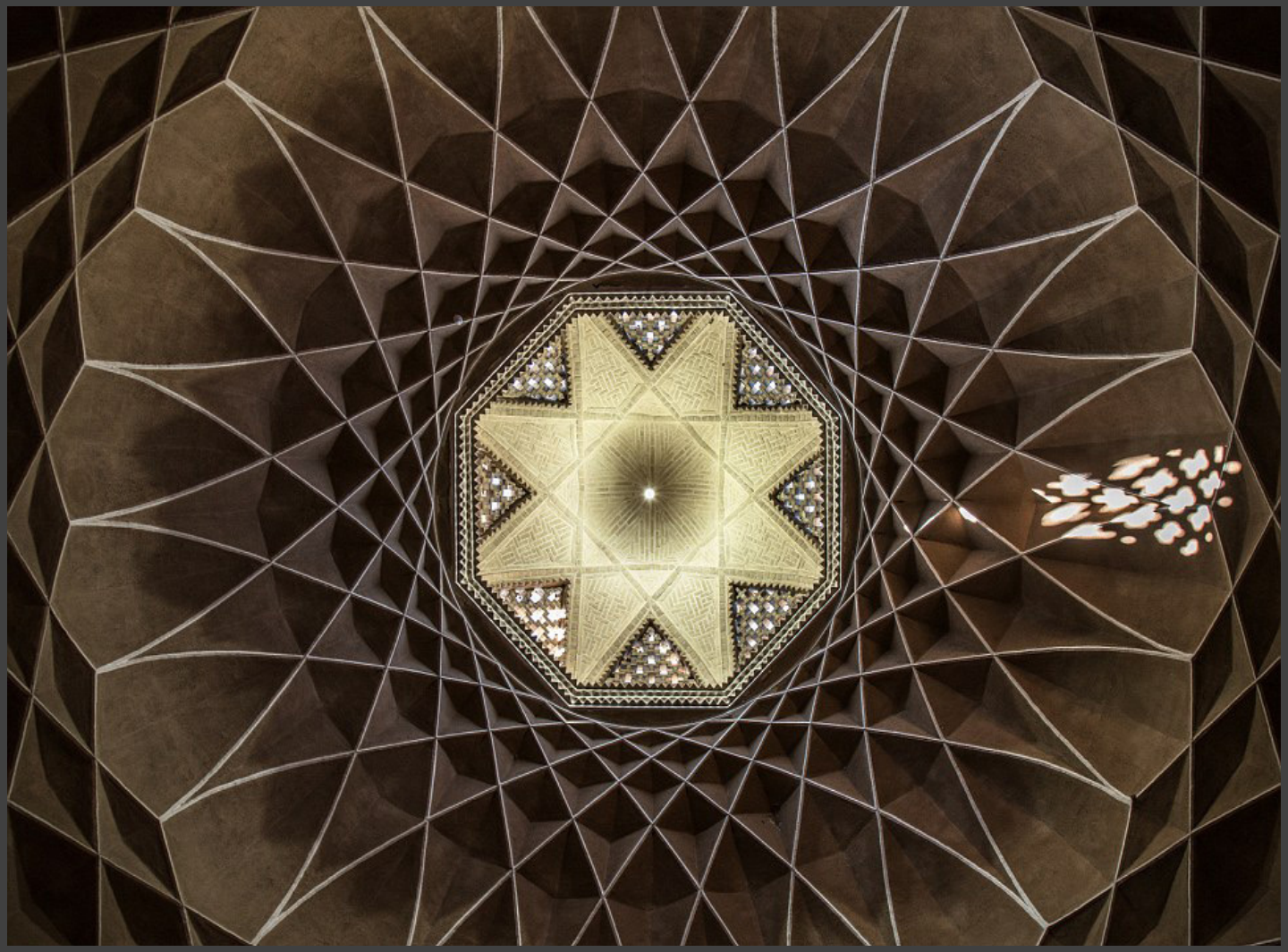
//Fig 2.

// Nasir Al Molk Mosque [Shiraz]

//Fig 3.

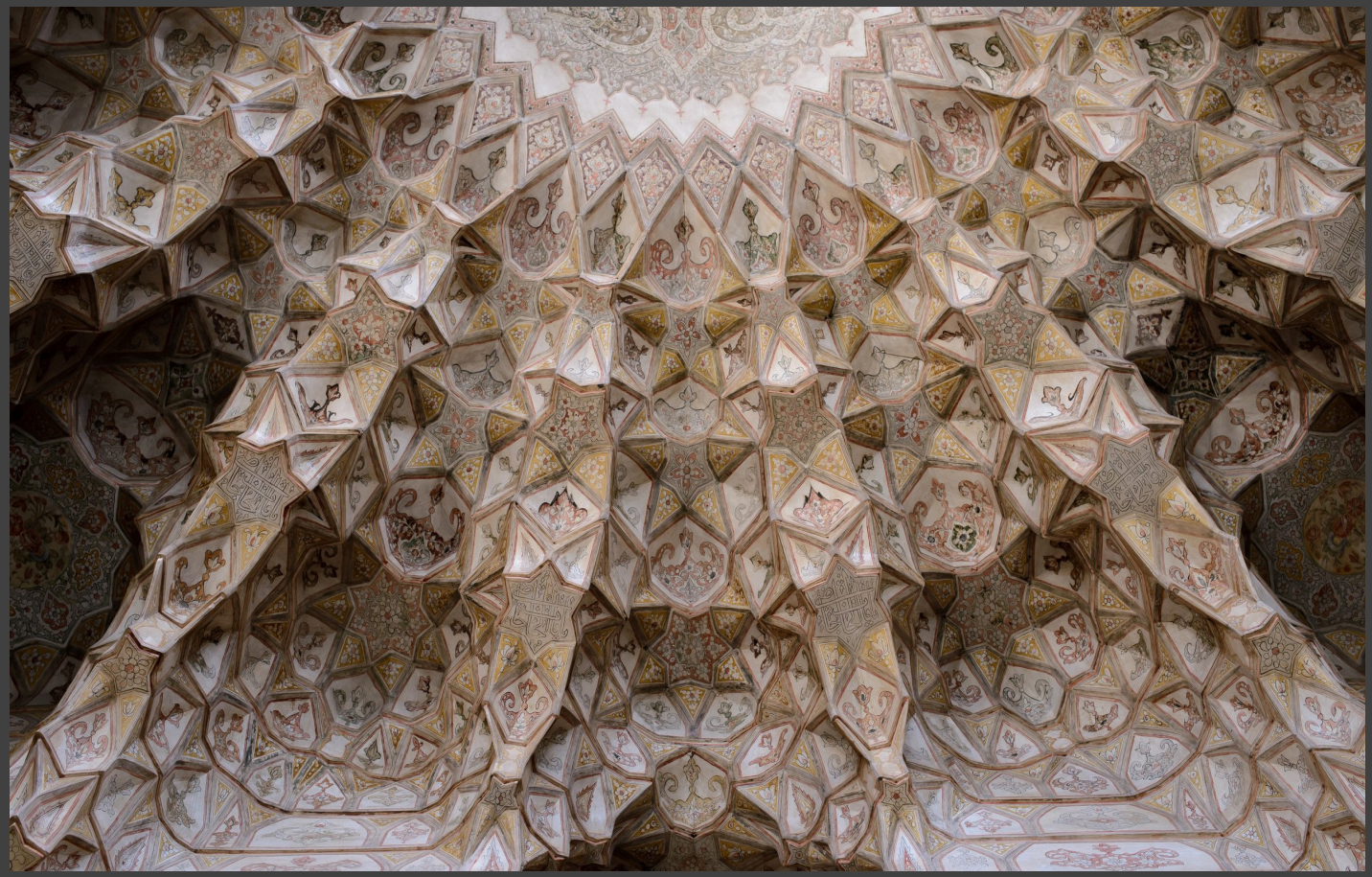
// Jameh Mosque [Isfahan]

//Fig 4.



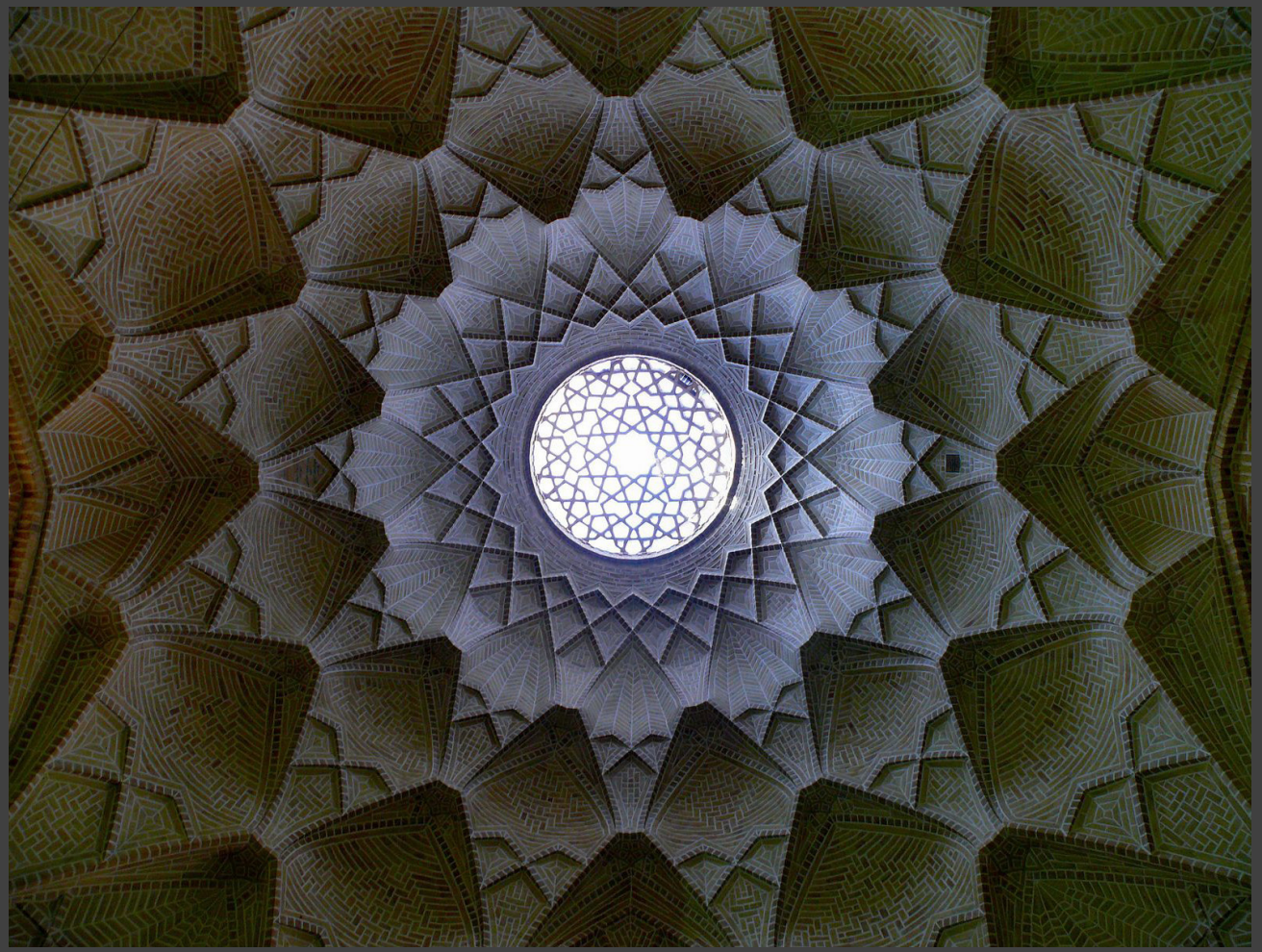
// Dolatabad [Yazd]

//Fig 5.



// Agha Bozarg Mosque [Kashan]

//Fig 6.



// Yazd Bazzar [Yazd]

//Fig 7.

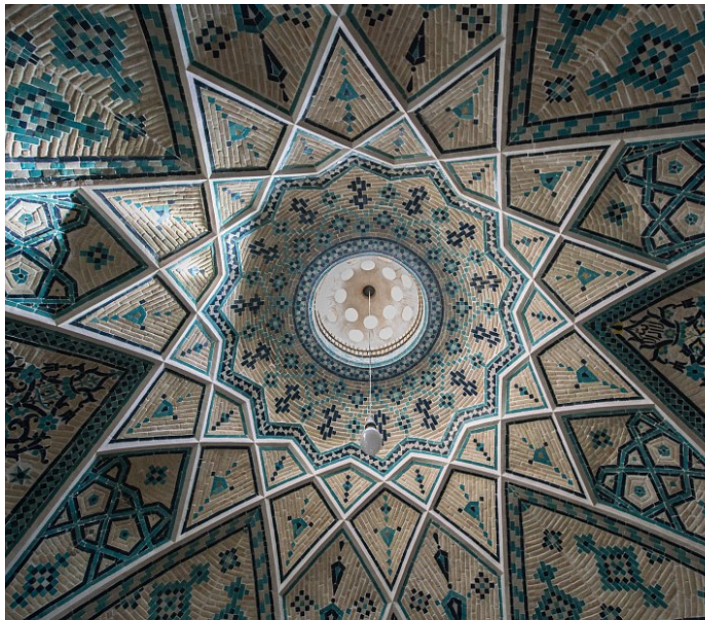
{ How can we use these precedents to influence the ways we think about form? }

{ Underlaying Rules of Persian Patterns |

// Many research have proven the existence of a profound relationship between mathematics and Persian patterns.
Patterns from both Pre-Islamic and Islamic periods have followed a set of geometrical rules and algorithms to construct the most complicated compositions found in architecture and cultural heritage[1].

// Today, we are using computers as tools to provide us with simple algorithms to create complex and unimaginable shapes and forms.

//In this research, the first step toward bridging the gap between the traditional and the contemporary way of thinking about form and space, is understanding the underlaying principles of Persian patterns.

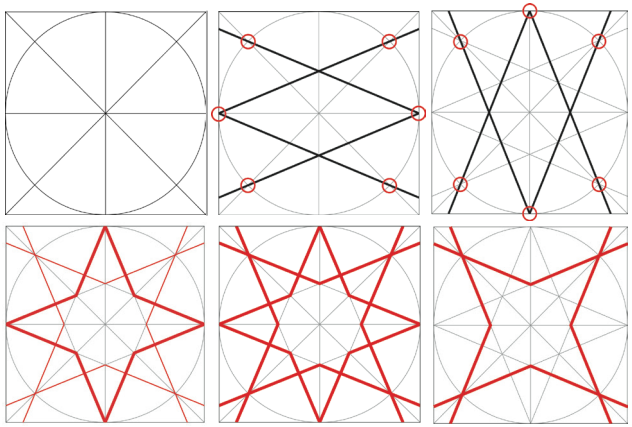


//Fig 8. Hammam House [Kashan] |

//The underlaying rules:

[1] Initial shapes
[2] Pattern construction principles
[3] Construction process
[4] Interplay between planar space and spatial depth

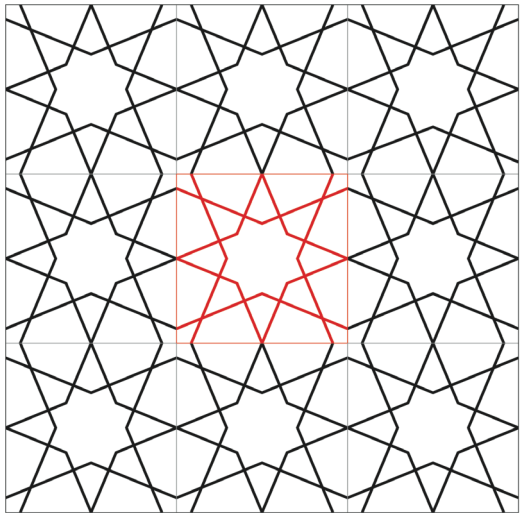
[1] [2]



[3]

//Fig 9. Pattern construction steps |

[4]

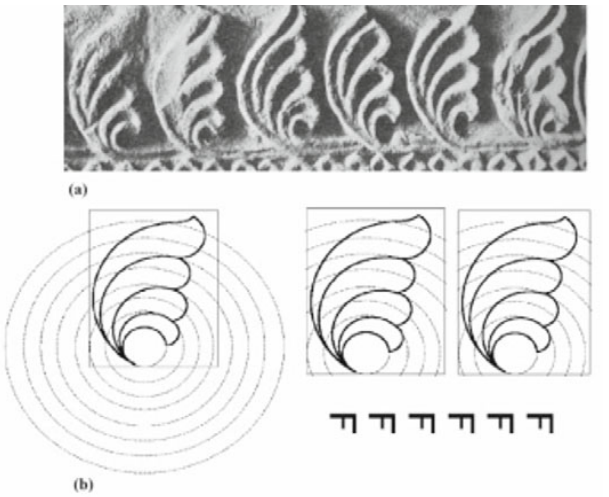


//Fig 10. Grid composition |

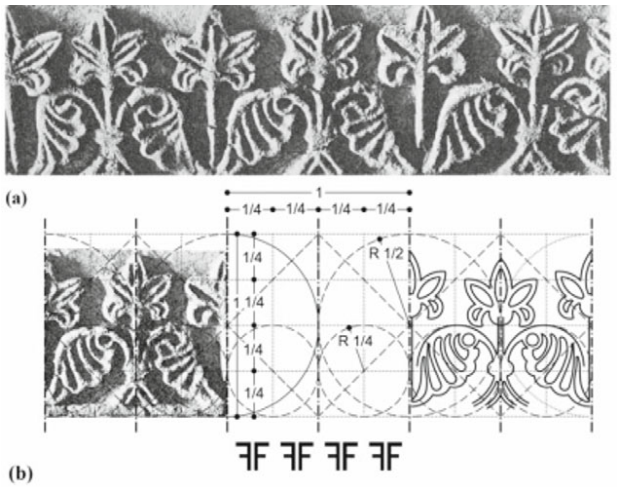
{ Underlaying Rules of Persian Patterns |

// In addition to the basic construction process, geometric symmetries such as scale, rotation, translation and reflection have also been extracted.[2]

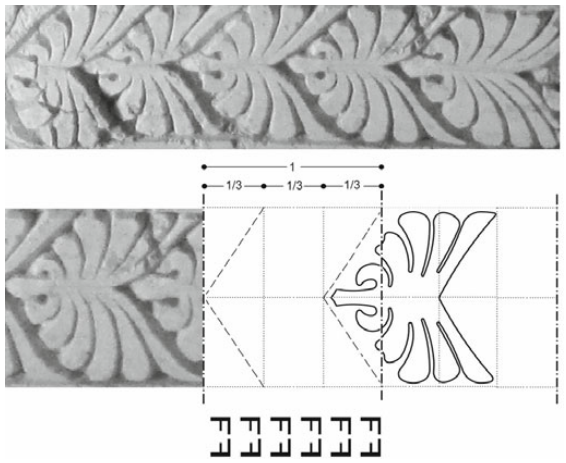
// Translation



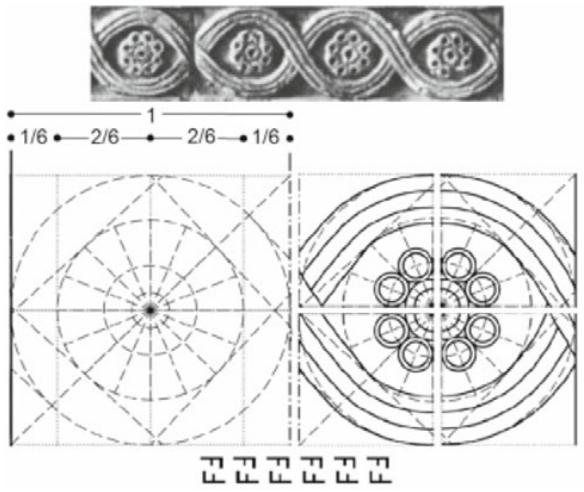
// Vertical Reflection



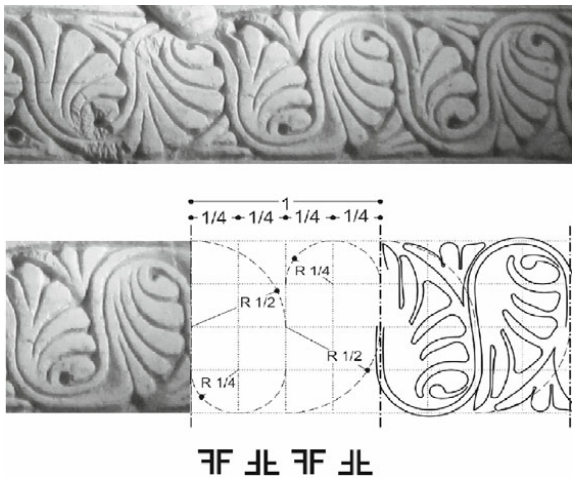
// Translation + Vertical Reflection



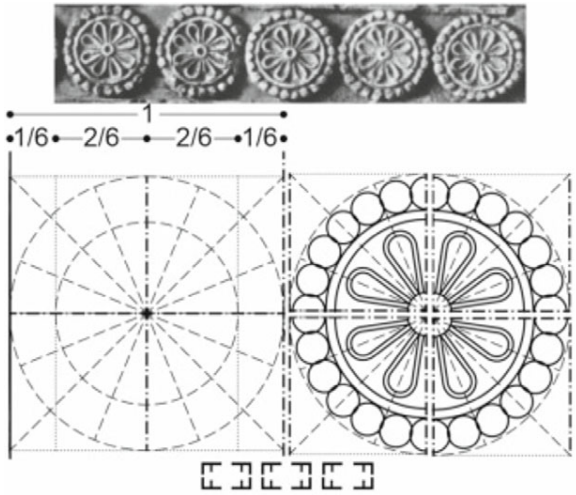
// Rotation



// Rotation + Vertical Reflection

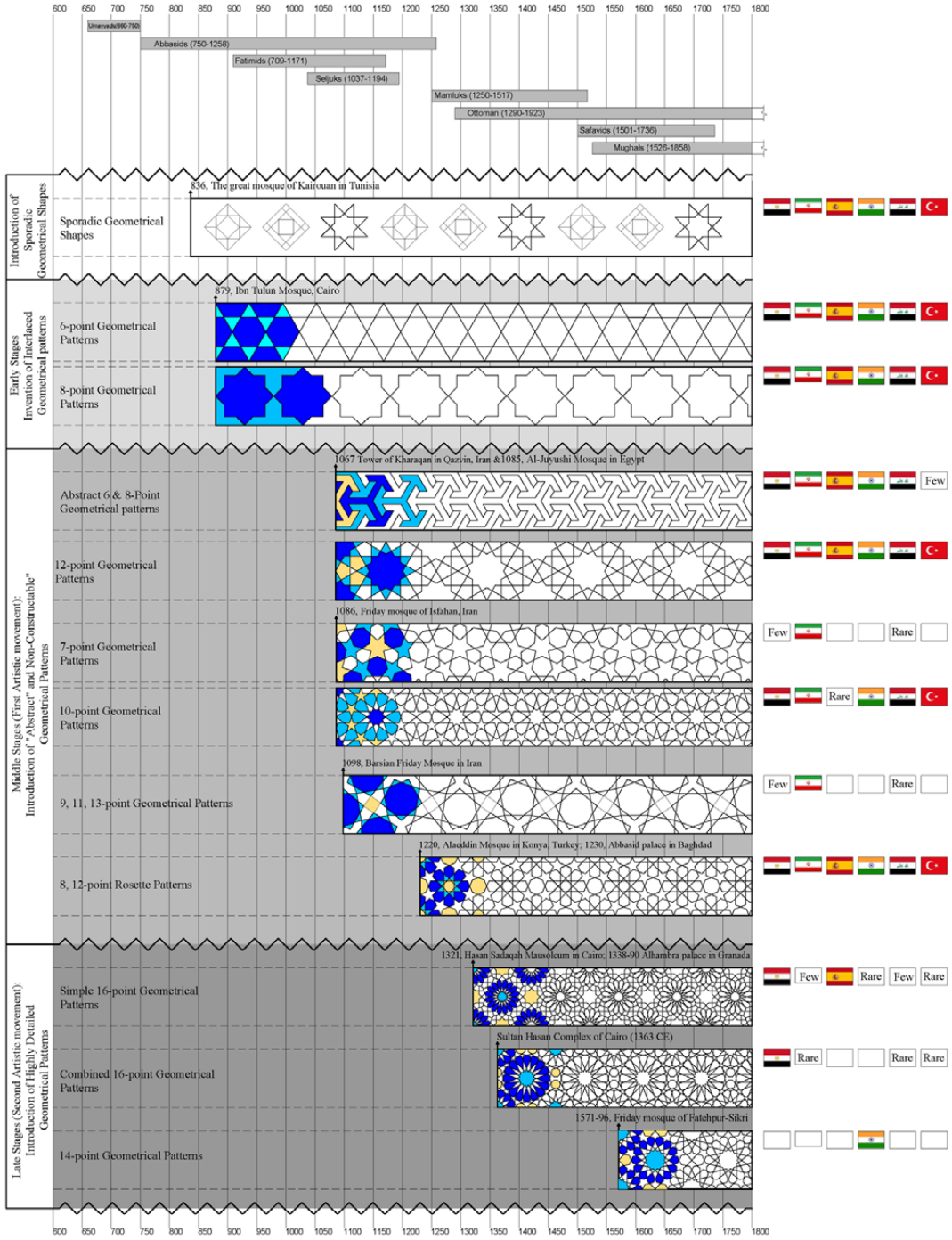


// Rotation + Horizontal Reflection



//Fig 11. Pattern construction |

{ Patterns Through History |



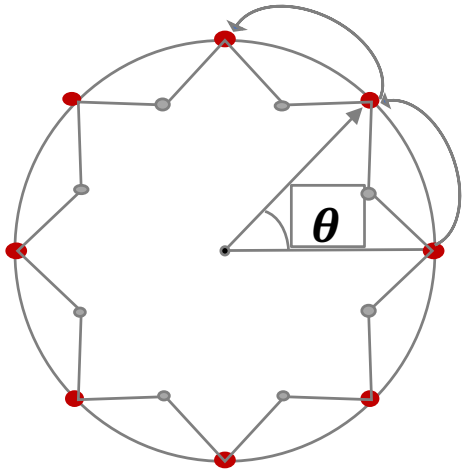
// Through time, pattern variations show a significant development in complexity of the overall shapes and compositions.[3]

//Fig 12.Time chart of evolution of Islamic Geometric Patterns throughout history |

{ Calculations |

// The following matrices show the process for drawing a simple shape using traditional method of calculation.

{ Rotation : $\begin{pmatrix} \text{Cos } \theta & -\text{Sin } \theta \\ \text{Sin } \theta & \text{Cos } \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$



// Construction of an 8 pointed star using rotation |

1. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix}$

2. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

3. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix}$

4. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

5. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\sqrt{2}/2 \\ -\sqrt{2}/2 \end{pmatrix}$

6. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} -\sqrt{2}/2 \\ -\sqrt{2}/2 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

7. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} 0 \\ -1 \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{pmatrix}$

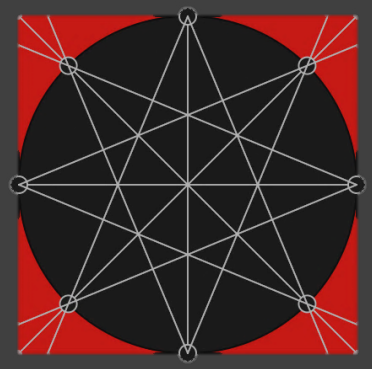
8. $\begin{pmatrix} Cos\ 45 & -Sin\ 45 \\ Sin\ 45 & Cos\ 45 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

{ Pen and Paper Algorithm |

// In order to create a simple shape, the following "pen and paper" algorithm had to be written in code. *"Processing"* is considered as the programming language and software, since it offers a great potential for designers to code within the context of the visual arts.[4]

// The algorithm and code are written using a similar logic to the pattern construction, where coordination of lines and points had to be defined and constructed. Many calculations had to be done to write the code which suggests, having the strategy and using similar algorithms would not be efficient for operating a generative approach.

```
{
void setup() {
  size ( 800, 800, FX2D);
  background (50);
}
void draw() {
  float radius=100;
  float center_x=width/2;
  float center_y=height/2;
  rectMode(CENTER);
  fill(110, 20, 20);
  stroke(10);
  rect(center_x, center_y, 200, 200);
  fill(20, 20, 20);
  stroke(10);
  ellipse(center_x, center_y, 200, 200);
  strokeWeight(1);
  stroke(100);
  line (center_x, center_y-radius, center_x,
center_y+radius);
  line (center_x-radius, center_y, center_x+radius,
center_y);
  line (center_x-radius, center_y-radius,
center_x+radius, center_y+radius);
  line (center_x+radius, center_y-radius, center_x-
radius, center_y+radius);

  float a=sqrt(2)/2;

  PVector p1= new PVector (center_x+a*radius,
center_y-a*radius);
  PVector p2= new PVector (center_x-a*radius,
center_y-a*radius);
  PVector p3= new PVector (center_x-a*radius,
center_y+a*radius);
  PVector p4= new PVector (center_x+a*radius,
center_y+a*radius);
  ellipse(p1.x, p1.y, 10, 10);
  ellipse(p2.x, p2.y, 10, 10);
  ellipse(p3.x, p3.y, 10, 10);
  ellipse(p4.x, p4.y, 10, 10);

  PVector p5= new PVector (center_x, center_y-radius);
  ellipse(p5.x, p5.y, 10, 10);

  PVector p6= new PVector (center_x, center_y+radius);
  ellipse(p6.x, p6.y, 10, 10);

  PVector p7= new PVector (center_x-radius, center_y);
  ellipse(p7.x, p7.y, 10, 10);

  PVector p8= new PVector (center_x+radius, center_y);
  ellipse(p8.x, p8.y, 10, 10);

  float m53=(p3.y-p5.y)/(p3.x-p5.x);
  //line (p5.x, p5.y, p3.x, p3.y);
  line (p5.x, p5.y, ((center_x+radius)-p5.y)/m53 + p5.x, center_x+radius);

  float m54=(p4.y-p5.y)/(p4.x-p5.x);
  //line (p5.x, p5.y, p4.x, p4.y);
  line (p5.x, p5.y, ((center_x+radius)-p5.y)/m54 + p5.x,
center_x+radius);

  float m61=(p1.y-p6.y)/(p1.x-p6.x);
  //line (p6.x, p6.y, p1.x, p1.y);
  line (p6.x, p6.y, ((center_x-radius)-p6.y)/m61 + p6.x, center_x-radius);

  float m62=(p2.y-p6.y)/(p2.x-p6.x);
  //line (p6.x, p6.y, p2.x, p2.y);
  line (p6.x, p6.y, ((center_x-radius)-p6.y)/m62 + p6.x, center_x-radius);

  float m83=(p3.y-p8.y)/(p3.x-p8.x);
  line (p8.x, p8.y, (center_y-radius), m83*((center_y-radius)-p8.x)+p8.y);
  //line (p8.x, p8.y, p3.x, p3.y);

  float m82=(p2.y-p8.y)/(p2.x-p8.x);
  line (p8.x, p8.y, (center_y-radius), m82*((center_y-radius)-p8.x)+p8.y);
  //line (p8.x, p8.y, p2.x, p2.y);

  float m71=(p1.y-p7.y)/(p1.x-p7.x);
  line (p7.x, p7.y, (center_y+radius), m71*((center_y+radius)-p7.x)+p7.y);
  //line (p7.x, p7.y, p1.x, p1.y);

  float m74=(p4.y-p7.y)/(p4.x-p7.x);
  line (p7.x, p7.y, (center_y+radius), m74*((center_y+radius)-p4.x)+p4.y);
  //line (p7.x, p7.y, p4.x, p4.y);
}
```
}



// Star shape using pen and paper algorithm in processing |

2 // Generative Explorations

Cultivating the Imagination

{ Re-thinking the Algorithm |

// In the generative method, it is important to find an efficient set of algorithms to write as code and produce a wide range of variations. Although there are many principles such as rotation and translation which must be borrowed, the ways we think about shape and pattern using lines and points, must be revised.

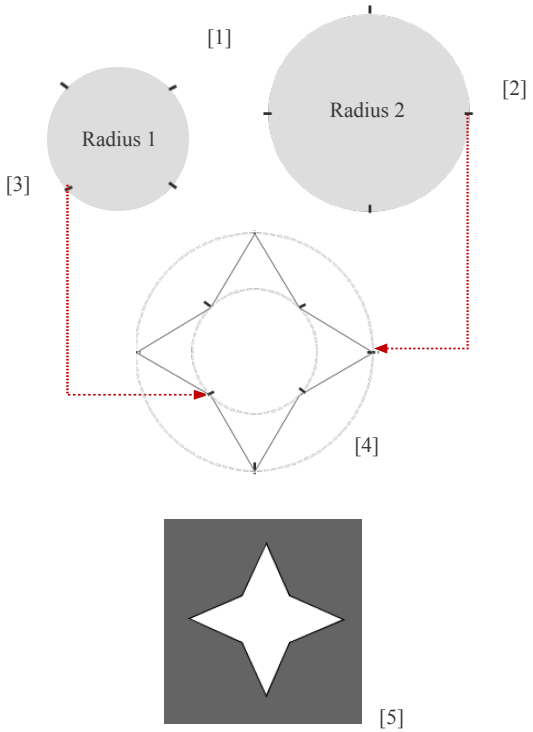
// As a result, re-thinking the algorithms play an important role in the process of this exploration.

// For example for producing a 4 point star shape, we can use the following set of rules:

[1] Two circles in different ratios
[2] Assigning 4 points on one circle in an equidistant manner
[3] Assigning 4 points on the other circle with a 45 phase difference alternatively to the 4 points on the first circle
[4] Over lapping the two sets of points
[5] Connecting the points

```
void setup() {
  size(640, 360);
}
void draw() {
  background(102);
  star(100,100,30,70,4);
}
void star(float x, float y, float radius1, float radius2, int
npoints) {
  float angle = radians(360)/ npoints;
  float halfangle = angle/2.0;
  beginShape();
  for (float a = 0; a < radians(360); a += angle) {
    float sx = x + cos(a) * radius2;
    float sy = y + sin(a) * radius2;
    vertex(sx, sy);
    sx = x + cos(a+halfangle) * radius1;
    sy = y + sin(a+halfangle) * radius1;
    vertex(sx, sy);
  }
  endShape(CLOSE);
}
```

// **BeginShape**



// **EndShape**

// Re-thinking the algorithm means:

[1] Taking advantage of the built-in functions which generative tools (such as Processing) offer.

[2] Finding the alternative algorithms which would produce a similar output but in a more efficient manner.

// Diagram of the 4 point star algorithm |

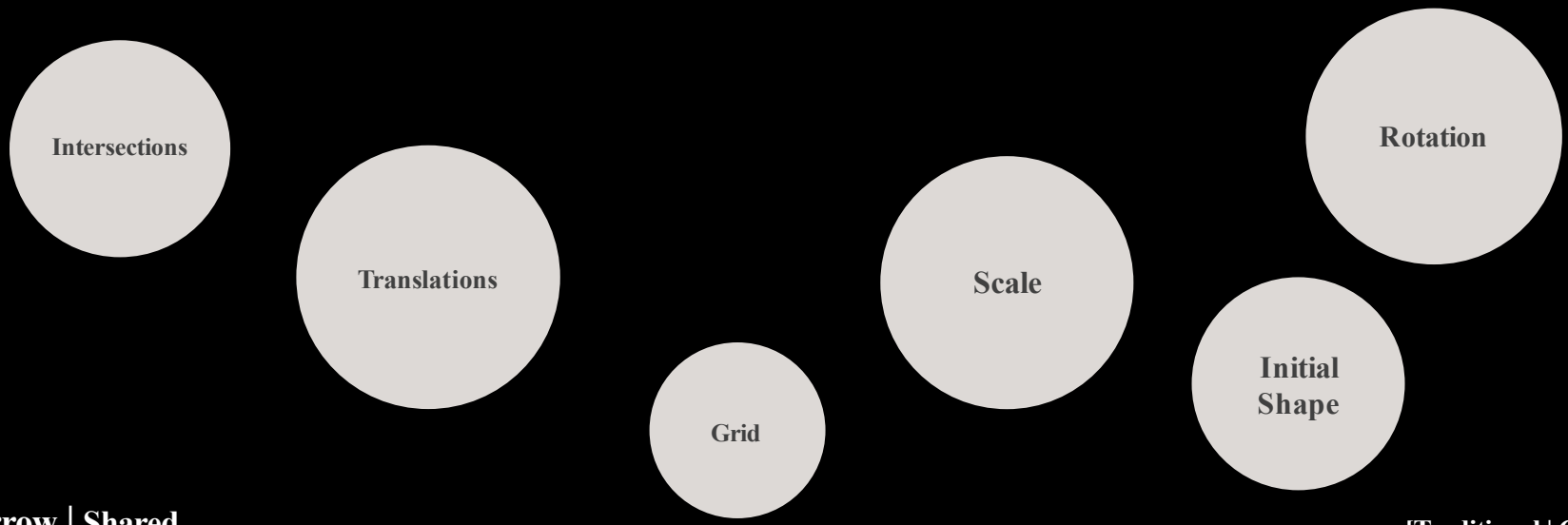
{ Re-thinking the Algorithm

// Borrowing the initiative

// Questioning the default

// Looking for a better option }

{ Borrow and Add |

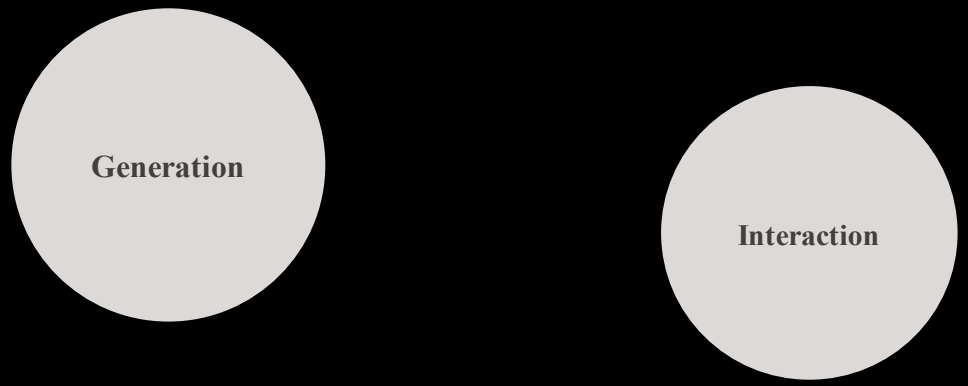


Borrow | Shared

[Traditional | Computational]

Add | Differed

[Computational]



{ Explorations Through Code |

// All the explorations are based on 3 or 4 main algorithms. Also, all the explorations include a shared initial shape (polygon) which produces a wide set of forms, compositions and graphics. The reason behind this is to create a consistent series of forms which can be manipulated and altered easily and coherently.

// The following images are part of the series of explorations. The codes have been written and ran in Processing.

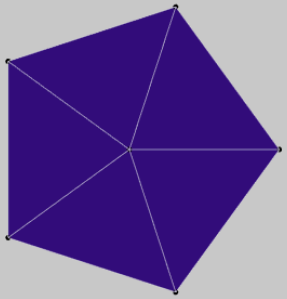
// In producing all images line weight, color, opacity and movement have been considered and coded.

Java ▾

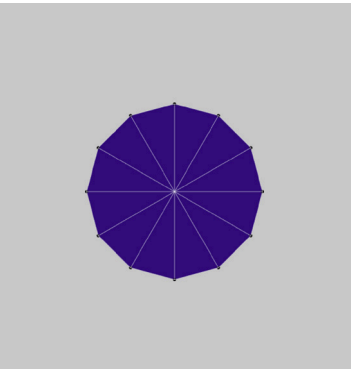
Atempt_2

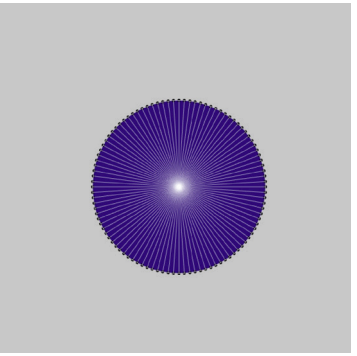
```
int total=5;
float z;
PVector [] p= new PVector [total];
PVector orig= new PVector (0, 0, 0);

void setup () {
  size(800, 800, P3D);
  smooth(8);
  cam= new PeasyCam (this, 500);
  float angle= 360.0/total;
  for (int i=0; i<total; i++) {
    float x= cos (radians(angle*i)) *100;
    float y= sin (radians(angle*i))*100;
    p[i]= new PVector (x, y, z);
  }
}

void draw() {
  background (200);
  fill(50, 20, 120);
  strokeWeight (5);
  stroke(#000000);
  point (orig.x, orig.y, orig.z);
  for (int i=0; i<total; i++) {
    point (p[i].x, p[i].y, p[i].z);
  }
  strokeWeight (1);
  stroke(#FFFFFF, 80);
  beginShape (TRIANGLE_FAN);
  vertex(orig.x, orig.y, orig.z);
  for (int i=0; i<total; i++) {
    vertex(p[i].x, p[i].y, p[i].z);
  }
  vertex (p[0].x, p[0].y, p[0].z);
  endShape();
```

Console Errors



// The side count for the polygons can be changed in the code.

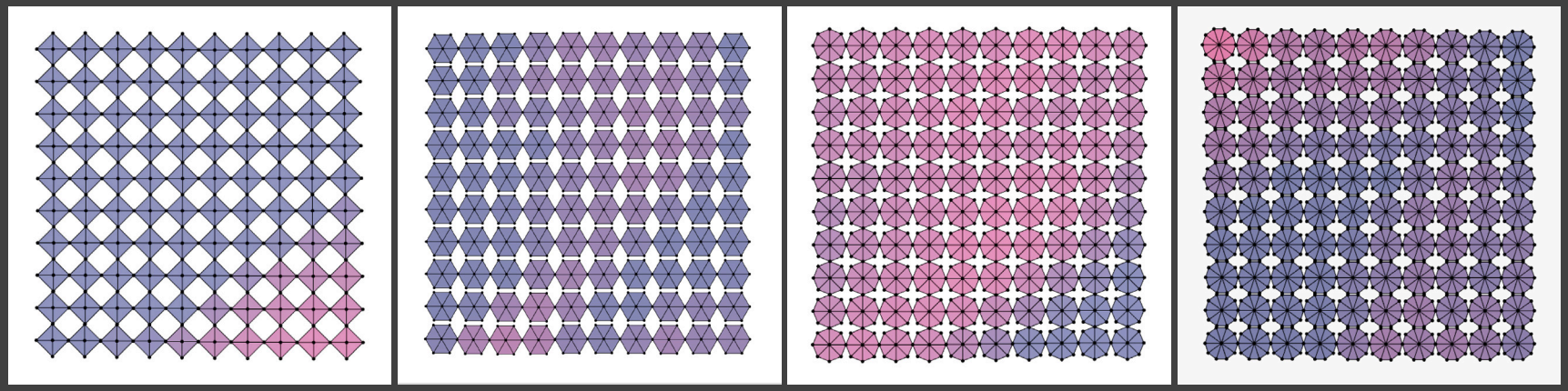


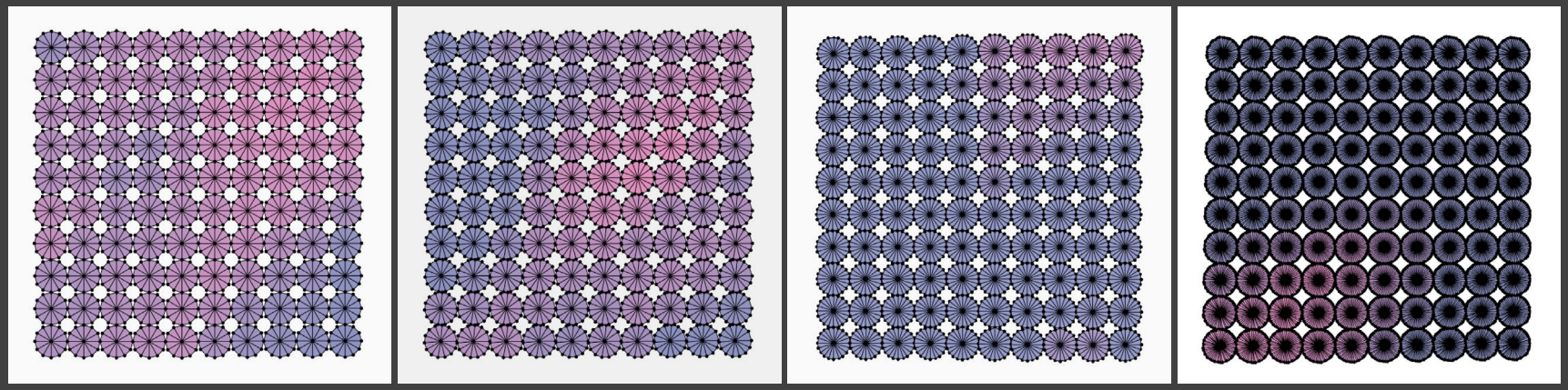


// Coding the initial shape [Polygon] |

{ Grid Studies |

// In the grid study process, the intention was to replicate the grid composition seen in the precedent.

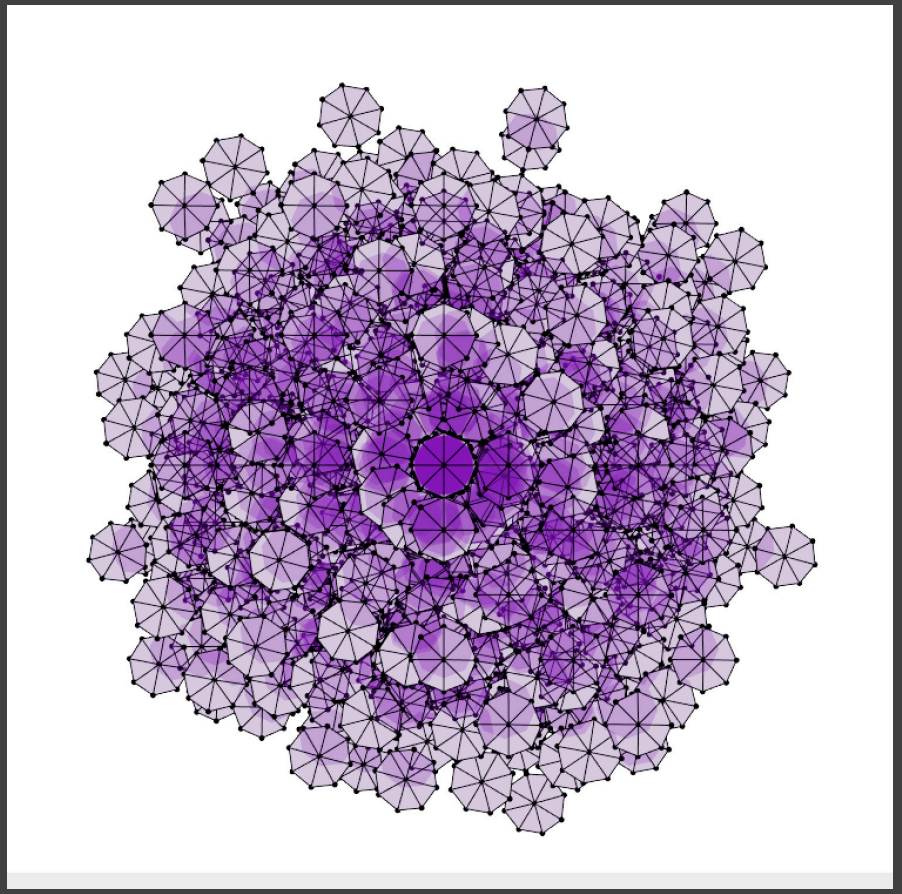


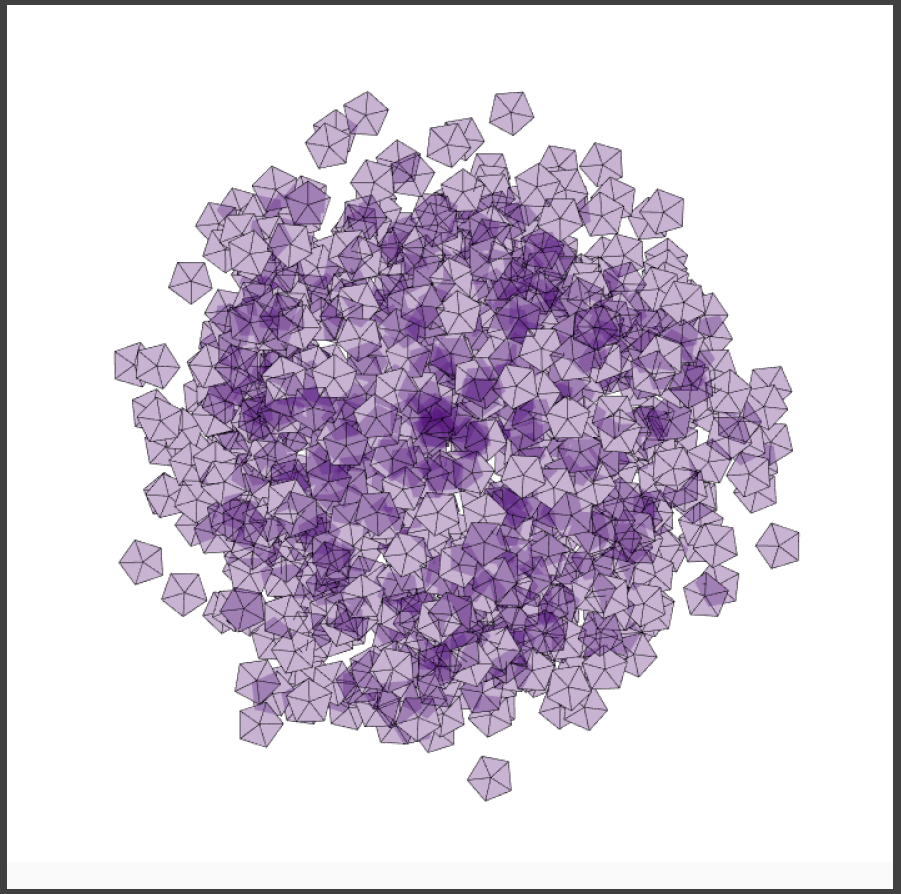


// As the number of sides in the polygon increases, the space between them becomes tighter and more circular.
Perception is challenged to recognize the white spaces as well as the colored spaces.

{ Layering |

// Another characteristic seen in the Persian patterns is layering, which has also been investigated through code.



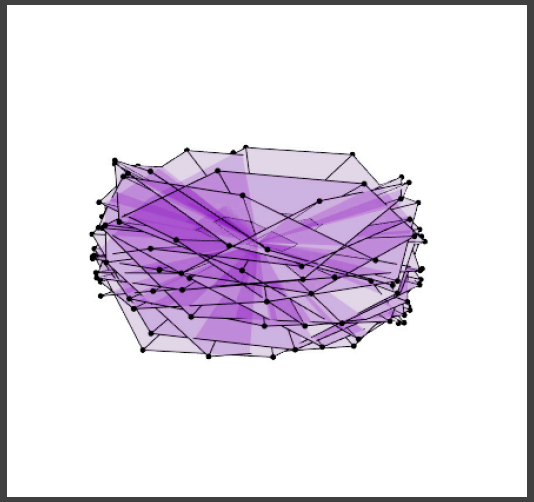


// By layering the polygons, without any intersection, three dimensional cluster looking forms appear. Different tones of the dominant color are created due to the opacity of each polygon.

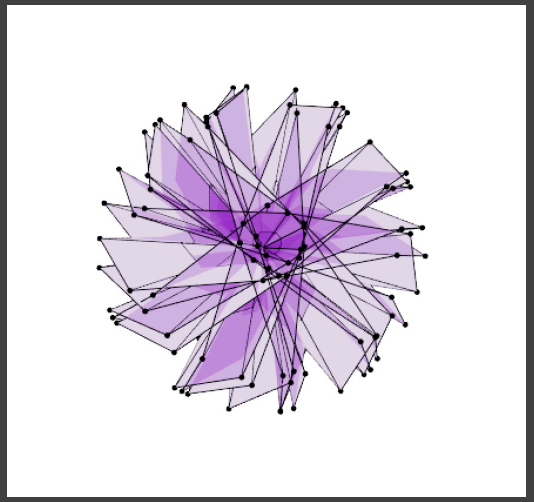
{ Kinetic Studies |

// Another interesting aspect of generative design is that it allows the integration of other attributes, such as movement, into the during the design development of architectural concepts and form.

// Side View

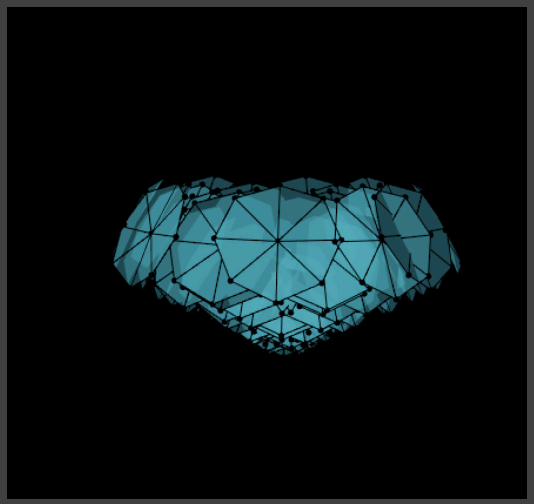


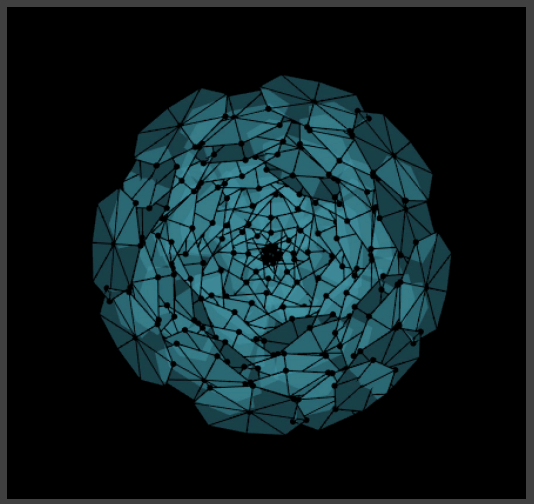
// Plan View



// Two dimensional movements

// These explorations, rather than being two-dimensional and decorative, borrow the two-dimensional principles from Persian patterns to produce three-dimensional constructs that suggest spatial form.

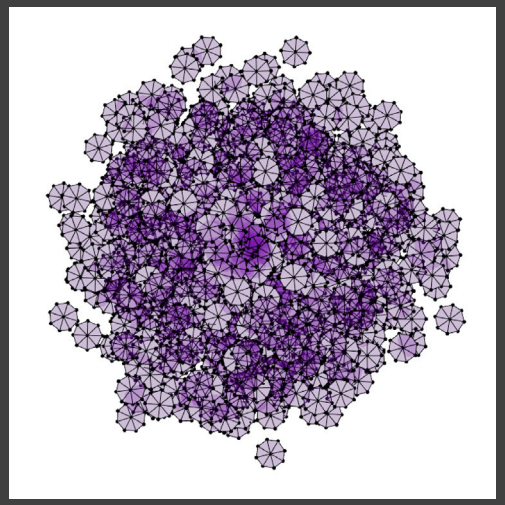


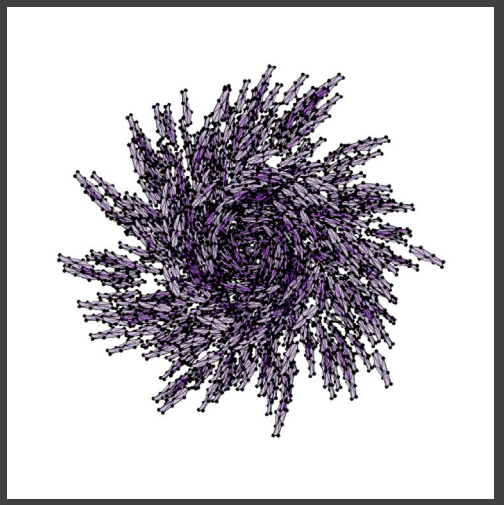


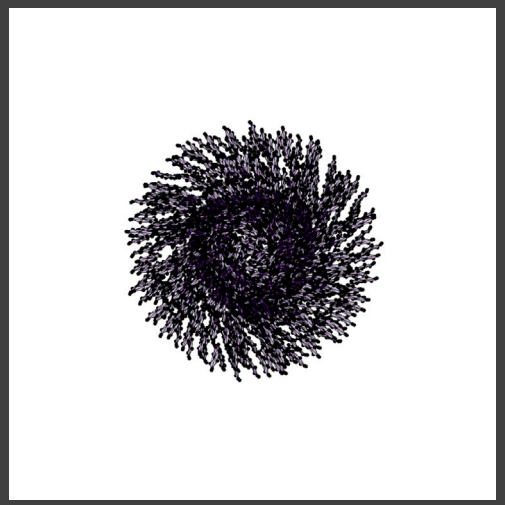
// Three dimensional movements

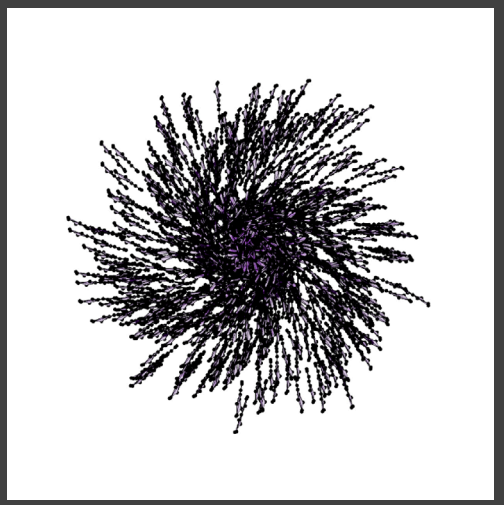
{ Kinetic Studies |

// Plan Views







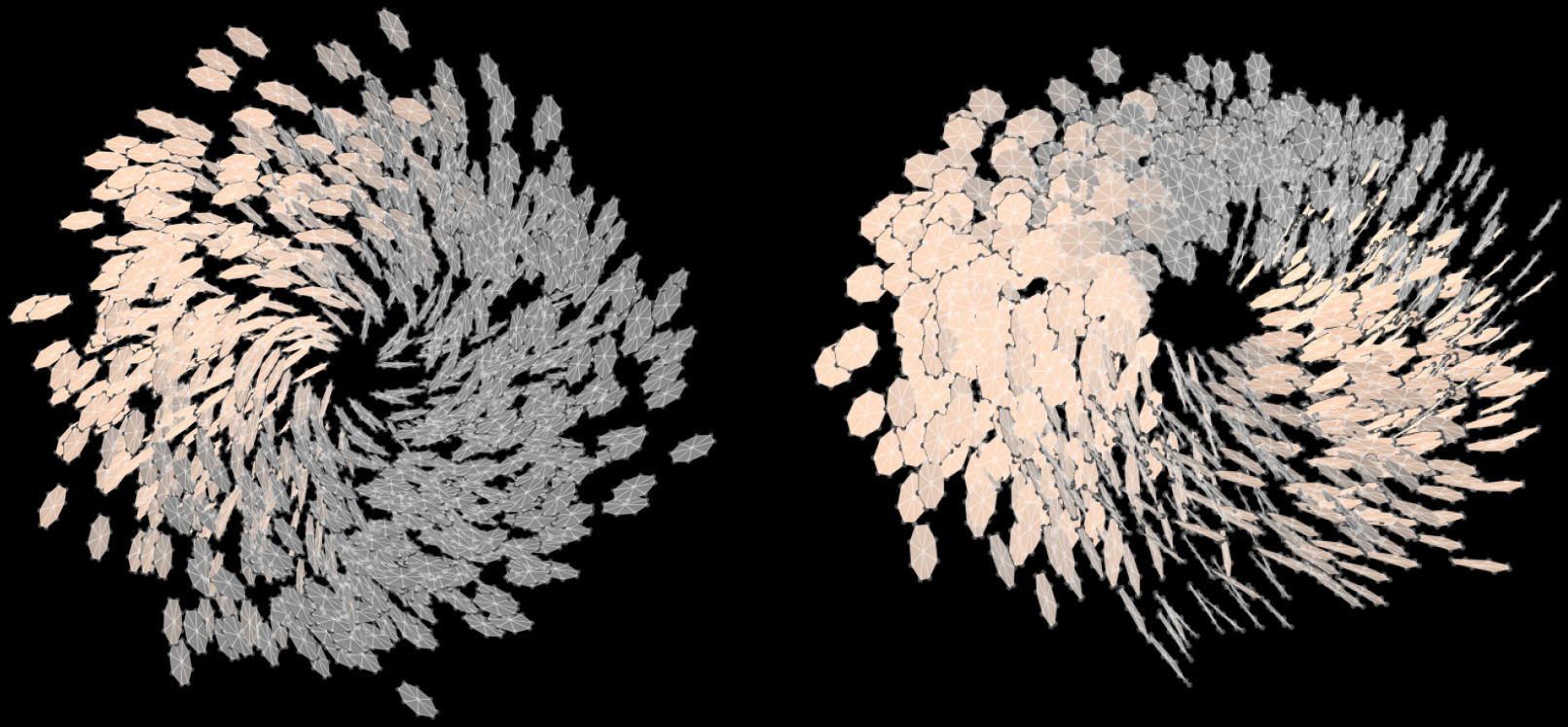


// The three dimensional movement has also been investigated in a more complex composition .

{ Kinetic Studies | Ambient Light Effects

// Plan View

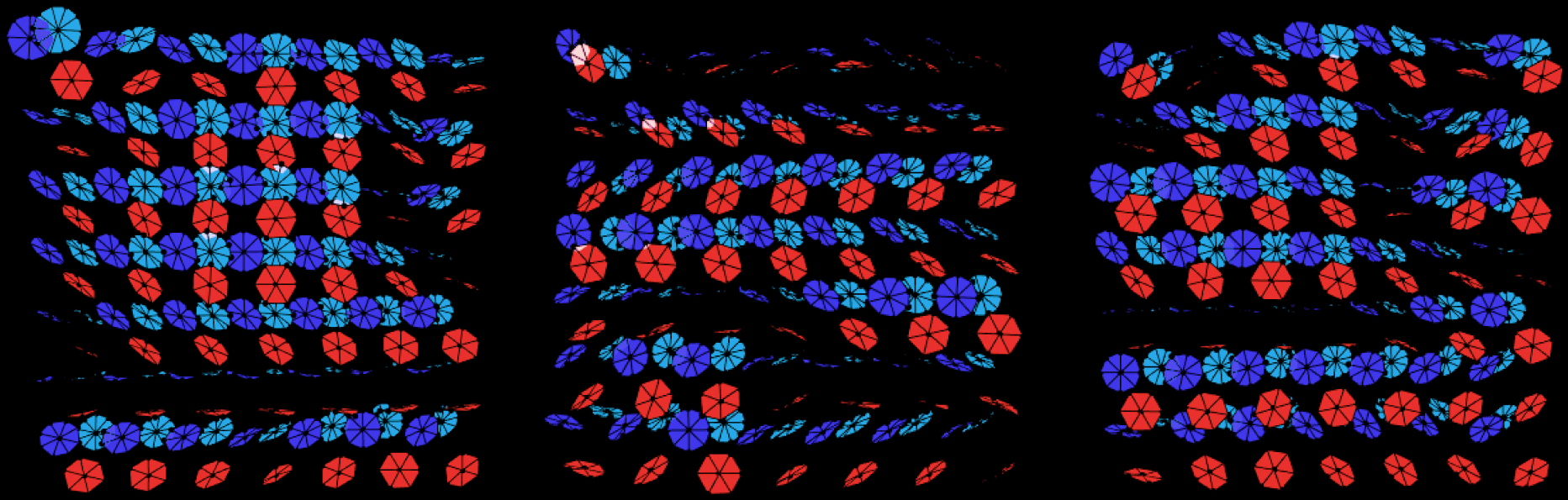
// Side View



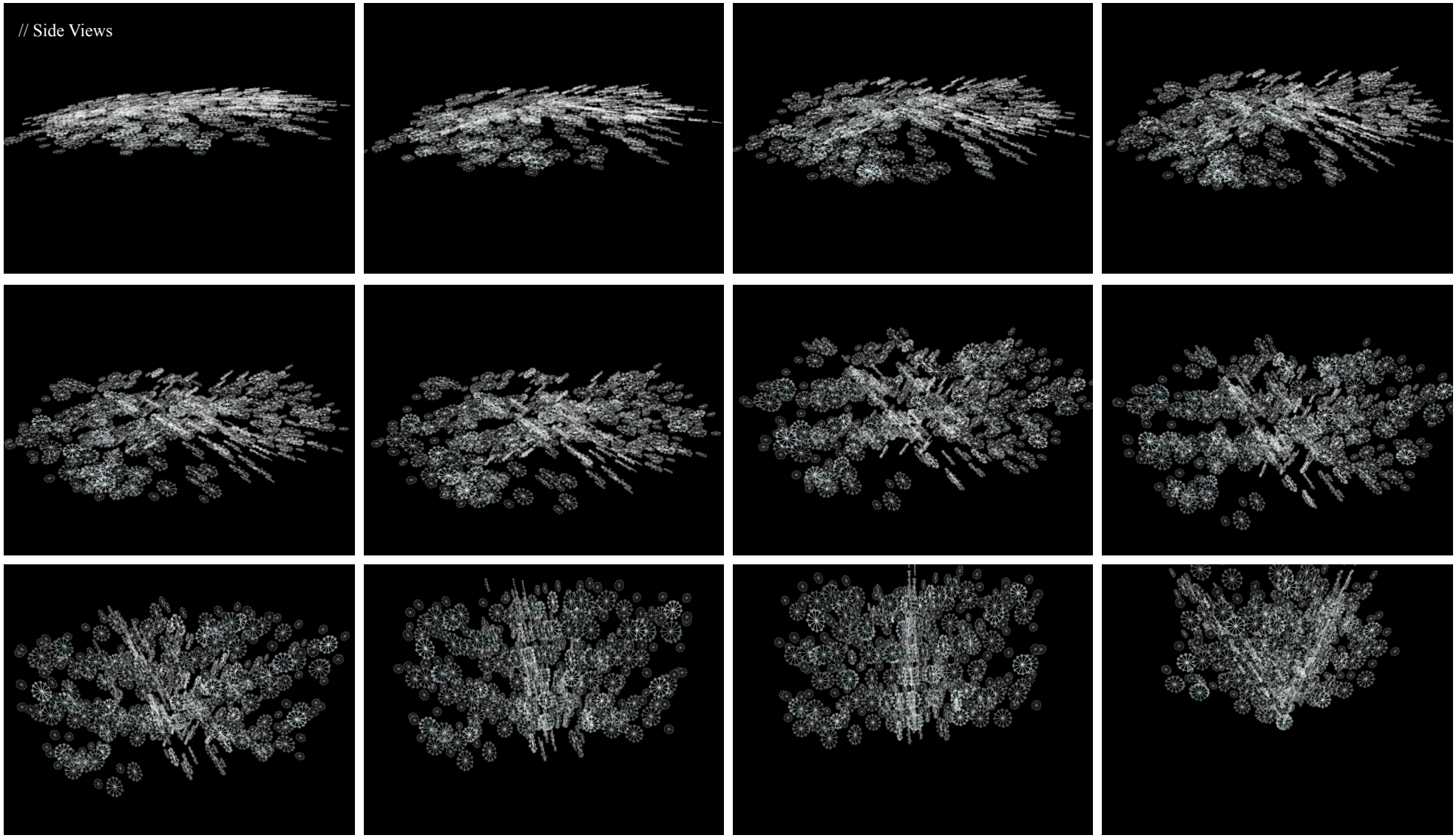
// Integrating ambient light into the design environment changes the perception of the form as it moves, appearing to switch color from one side to the other.

{ Form Follows Movement |

// Frozen frames of movements would allow interesting forms and compositions to be observed and selected for further development.

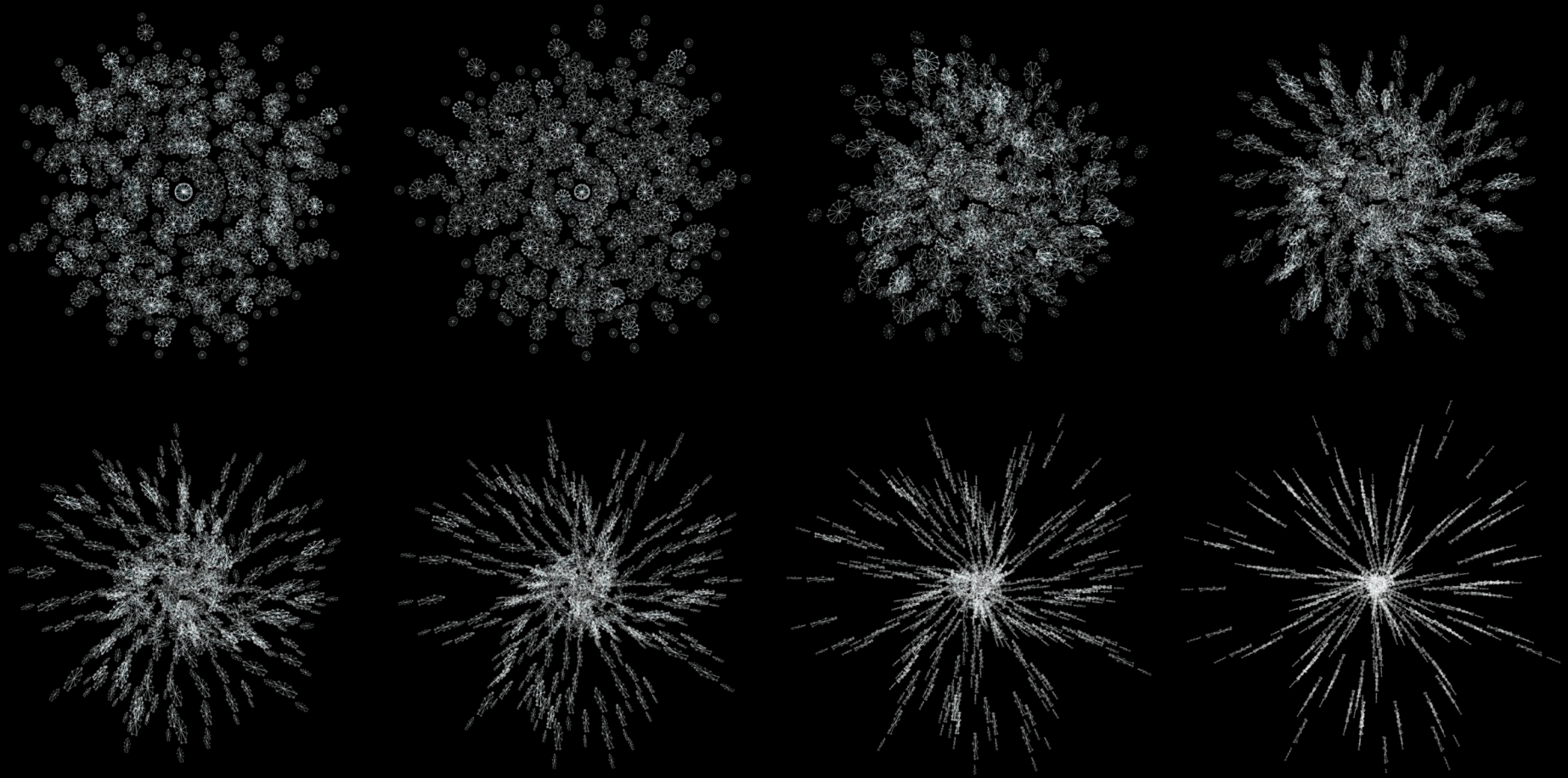


{ Form Follows Movement |



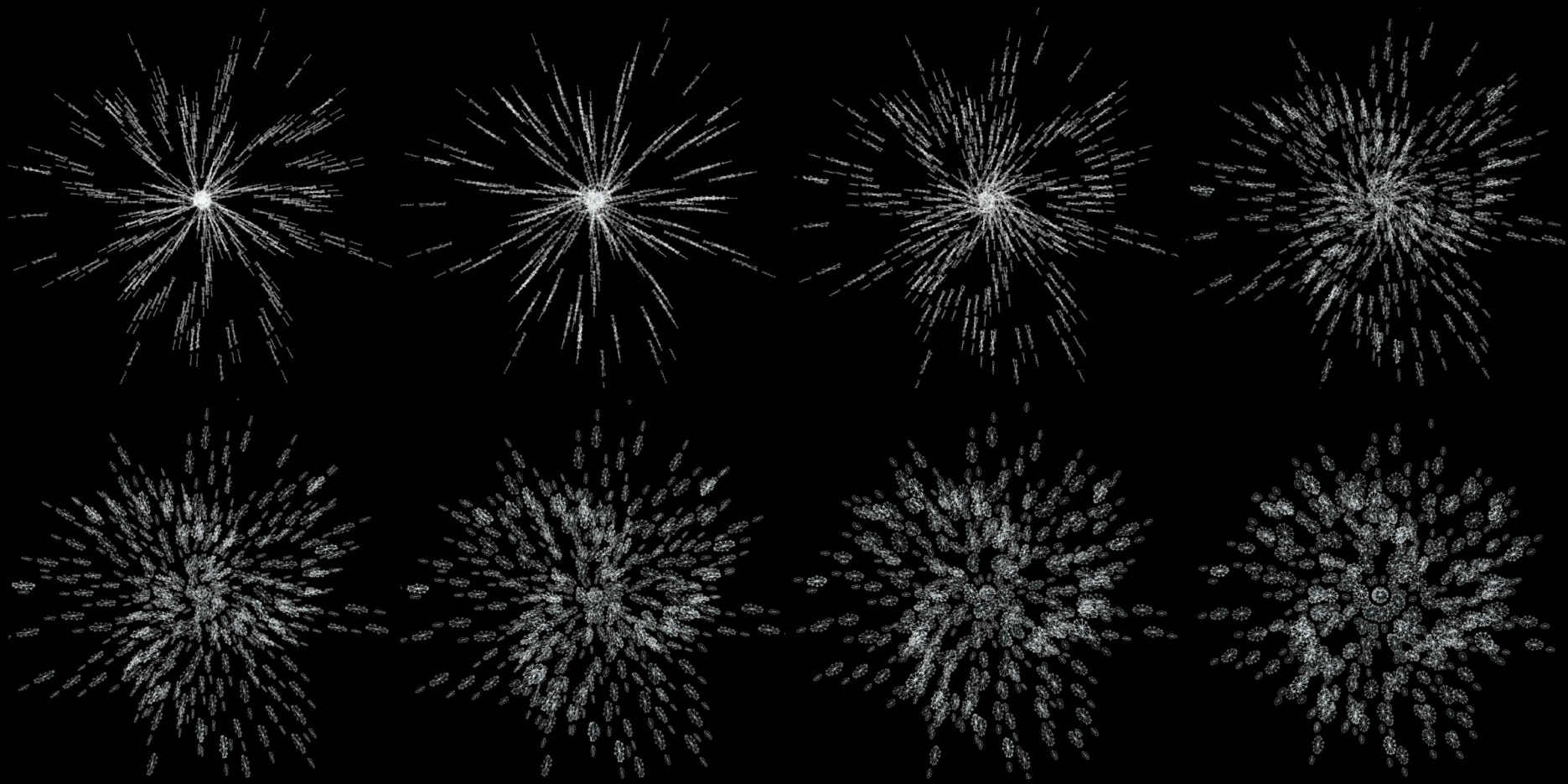
// This example presents different manners and suggests different forms as it is moving in more than two dimensions in space.

{ Form Follows Movement | Part 1



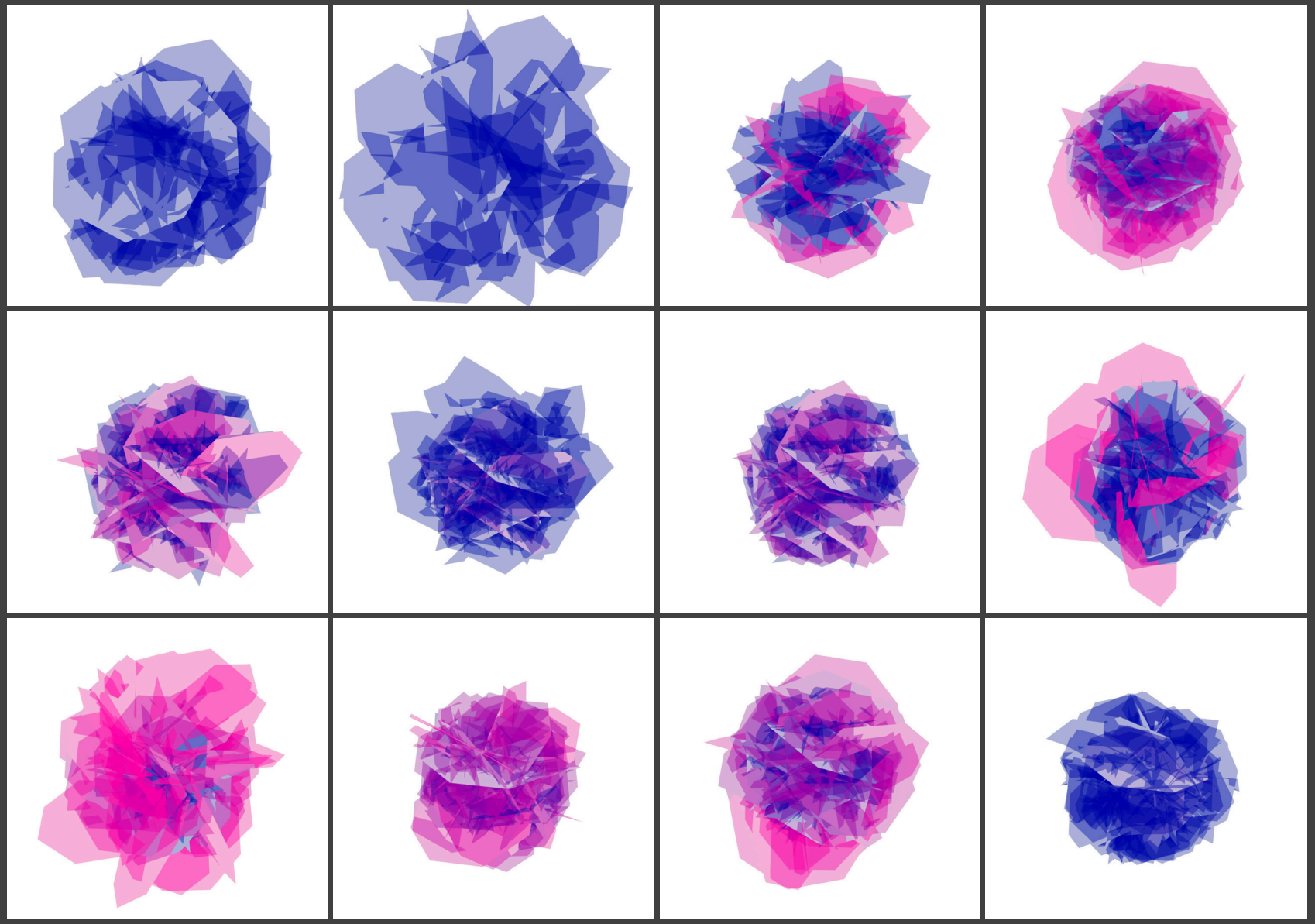
// Plan Views

{ Form Follows Movement | Part 2



// Plan Views

{ Form Follows Movement |

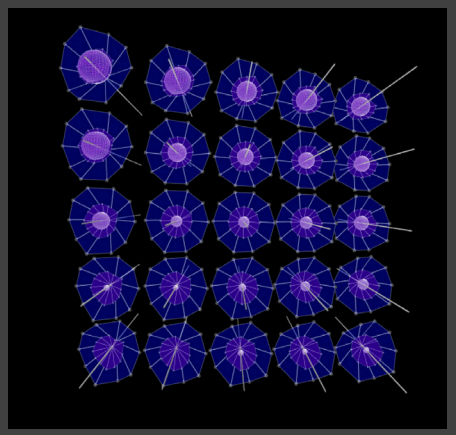
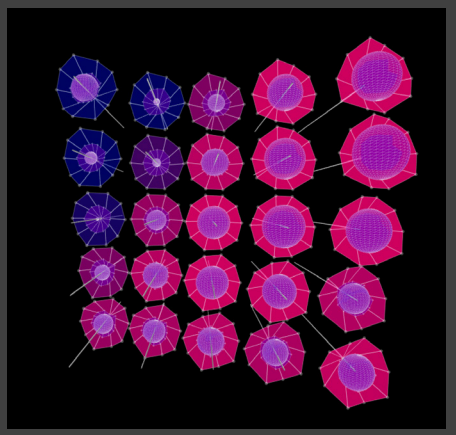
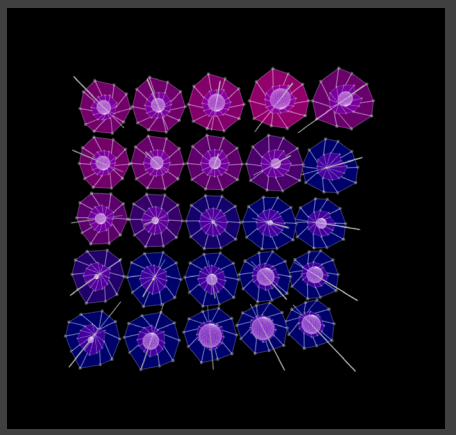
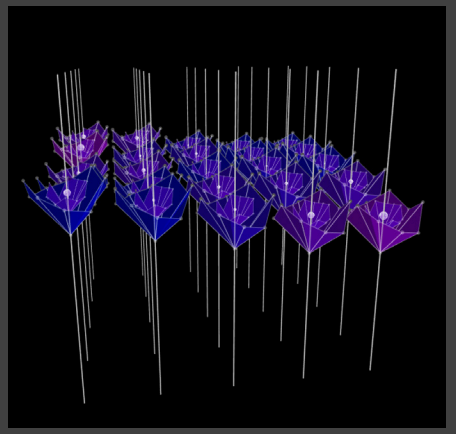
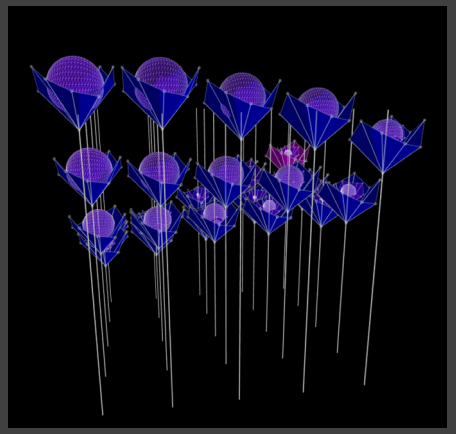
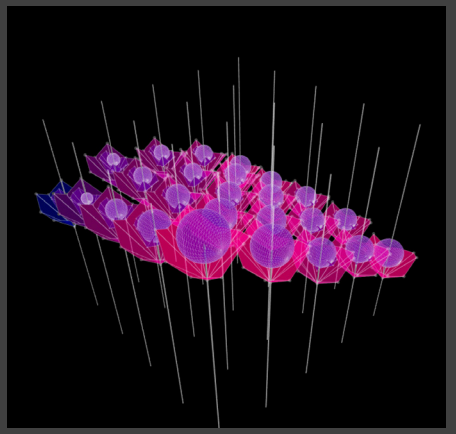
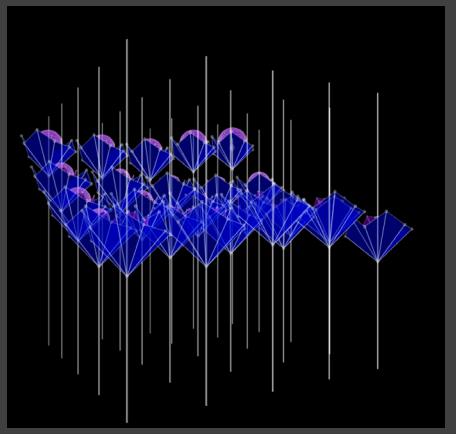


// To explore variety in the movement and to challenge the outputs, in some cases like this example, in some cases like this example, random numbers were used in parts of the code. In examples where a particular form was desired, a specific numbers had to be

{ Form Follows Movement |

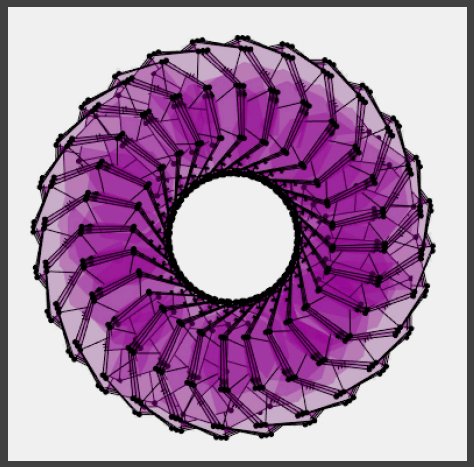
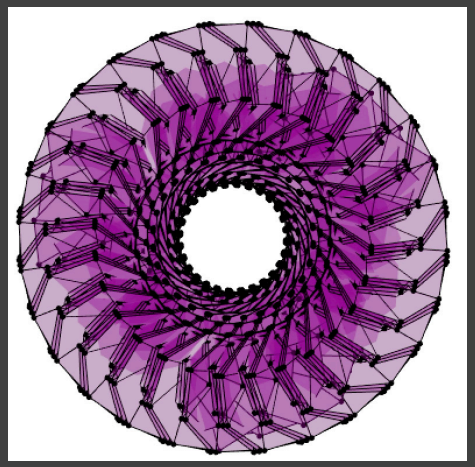
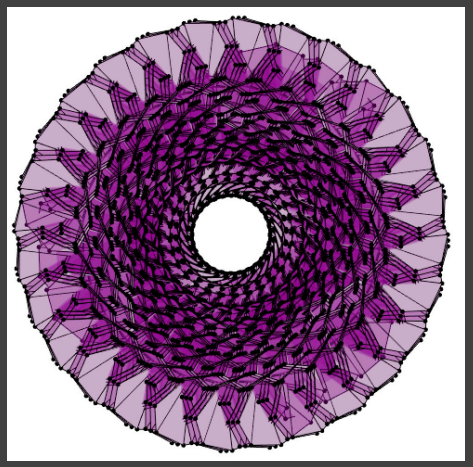
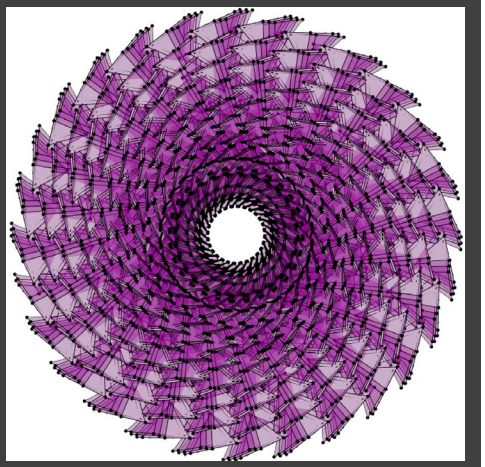
// This example uses specific numbers to create the overall structure which is a combination of a fixed rod grid and a dynamic polygon grid.
The movement, however, uses randomized numbers.

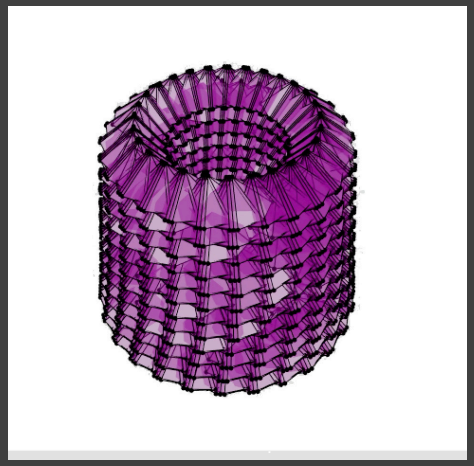
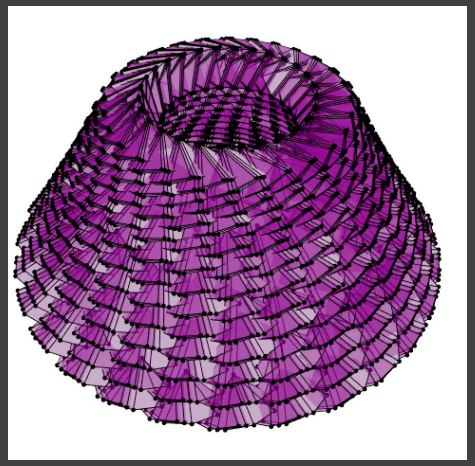
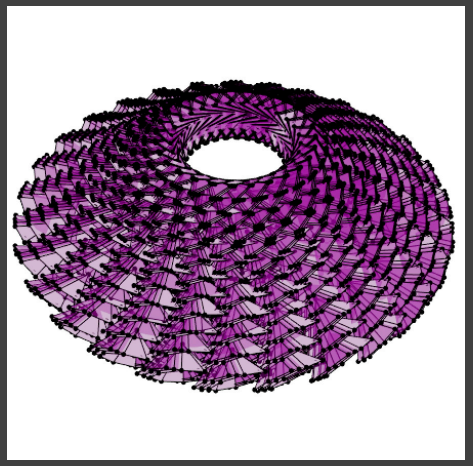
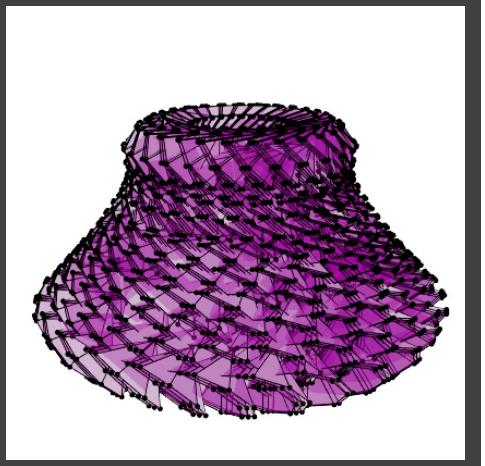
// Plan Views

{ Form Follows Movement |

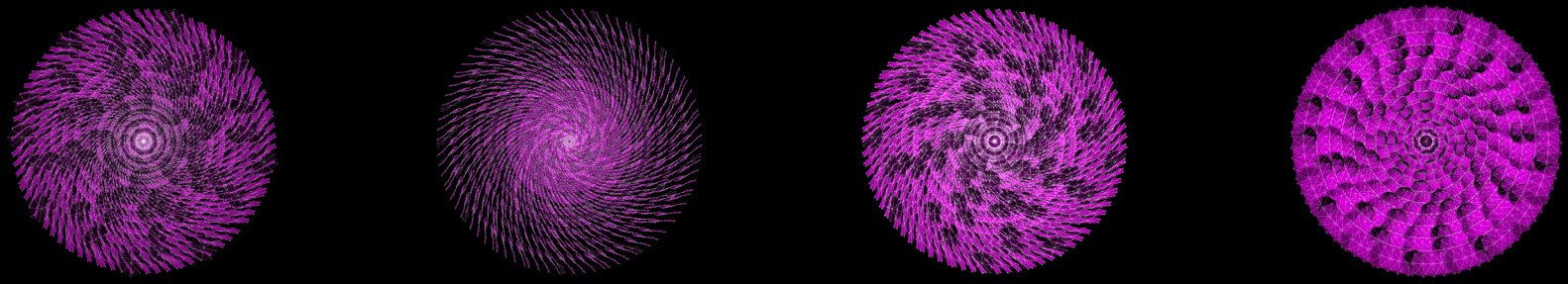
// Plan Views

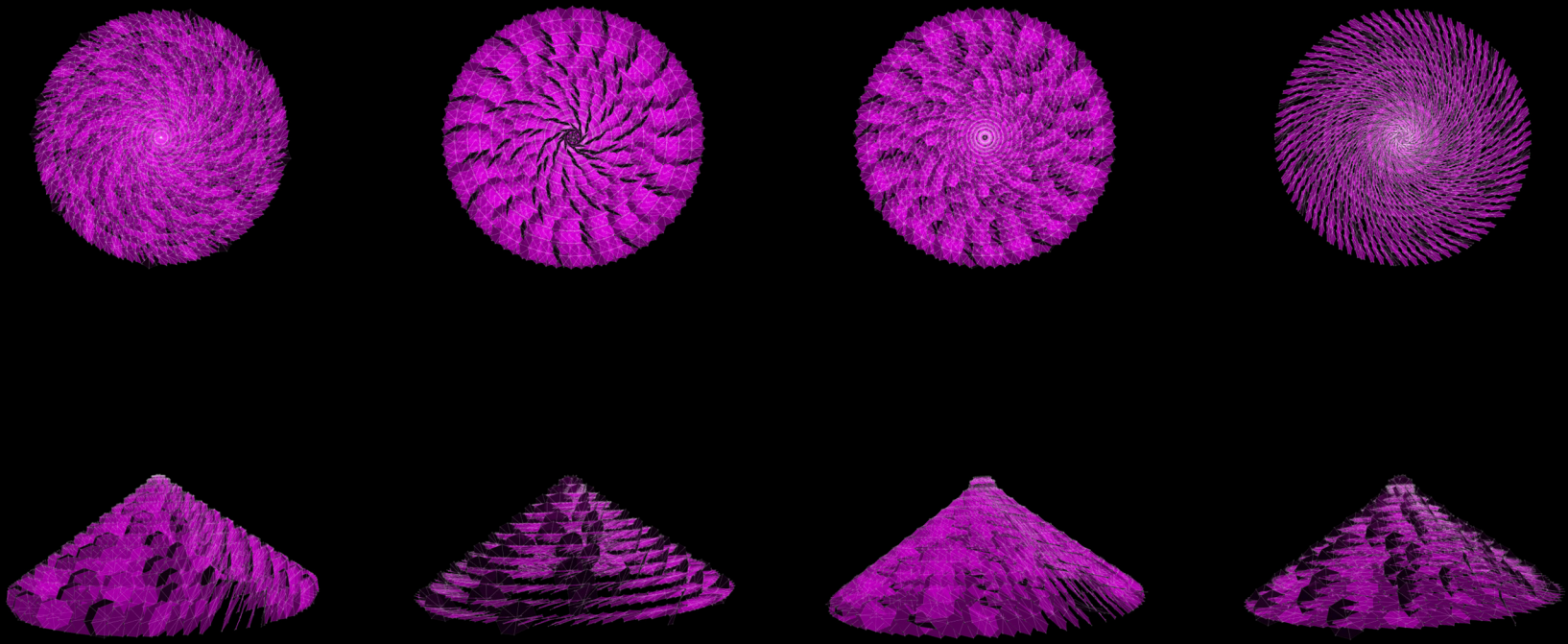





// Side Views

{ Form Follows Movement |

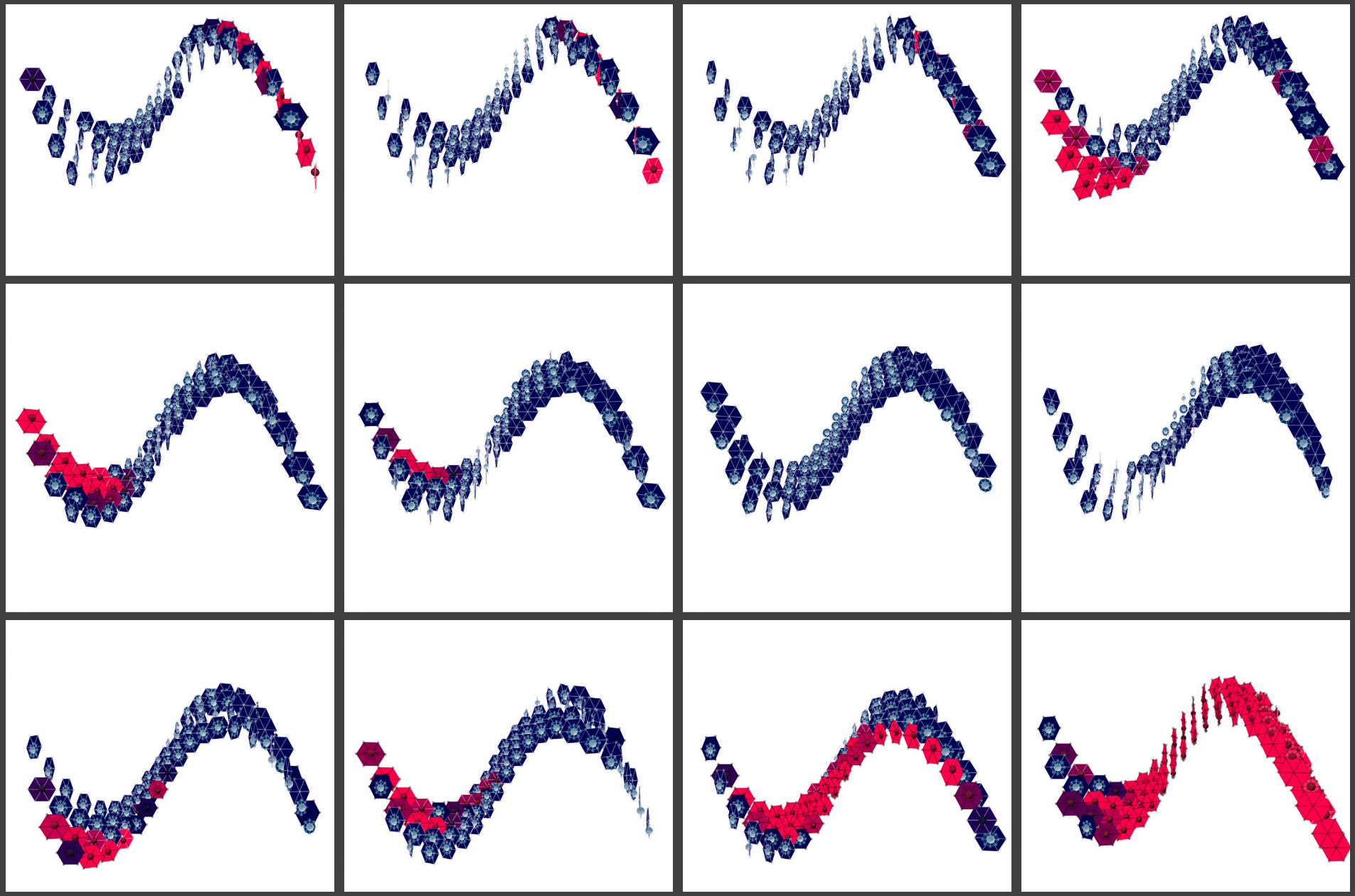


// Plan Views

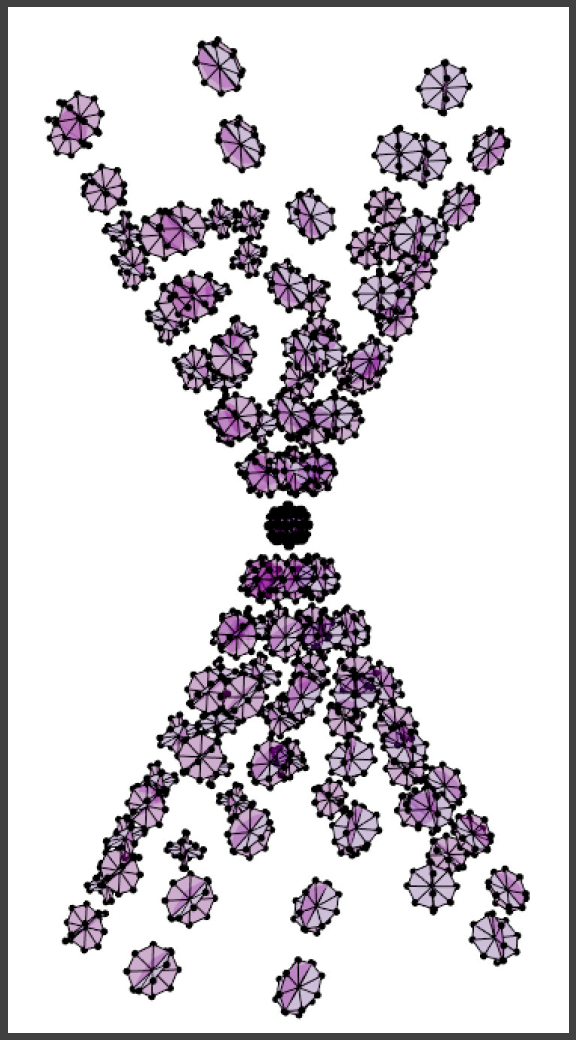


// Side Views

{ Form Follows Movement |

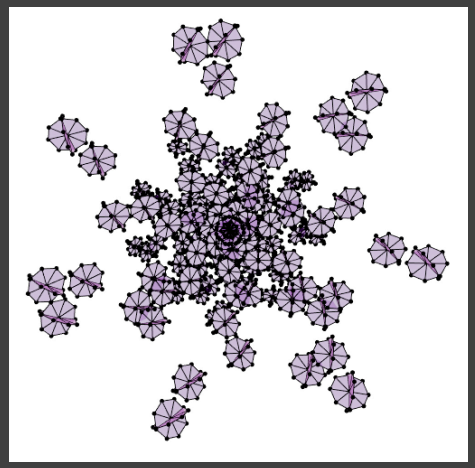


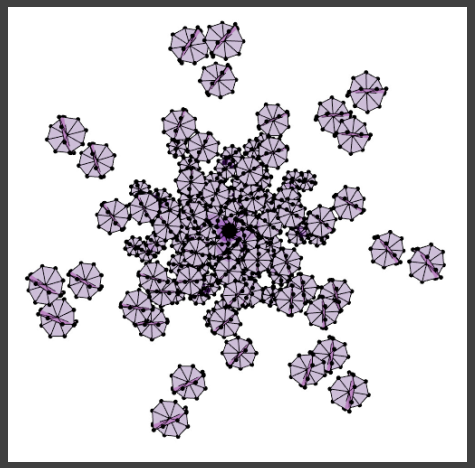
{ Form Follows Movement |

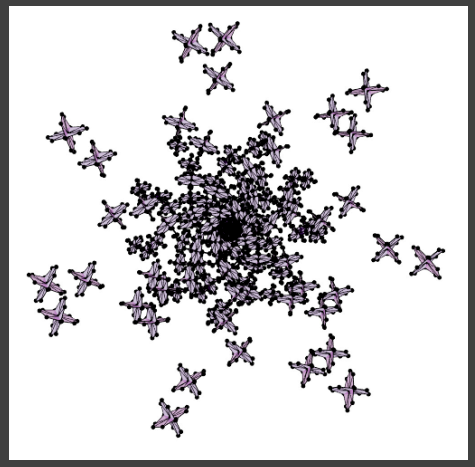


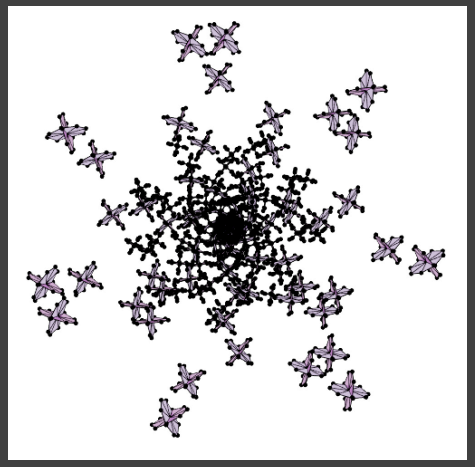
// Side View

// Plan Views

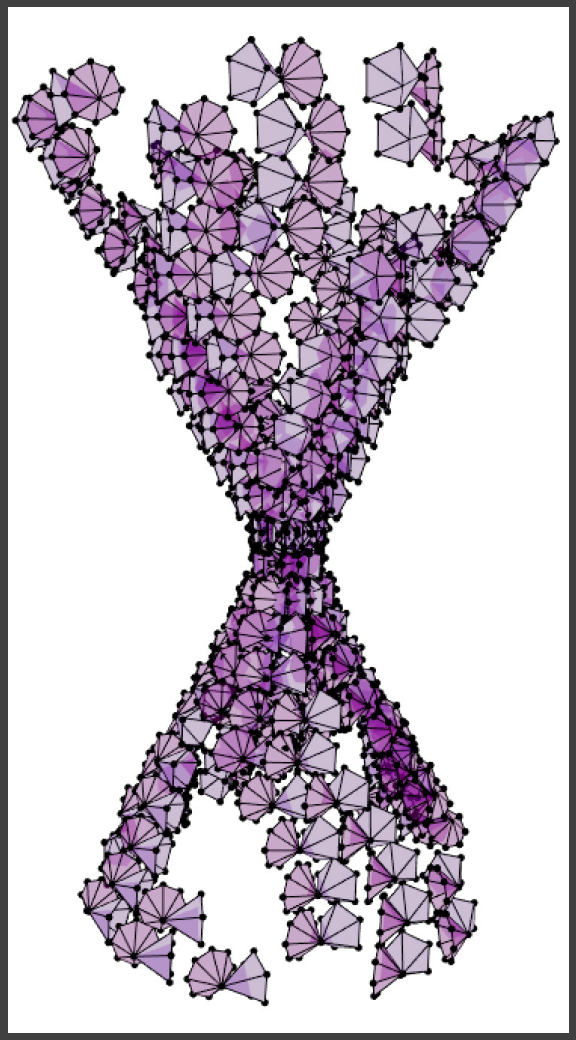






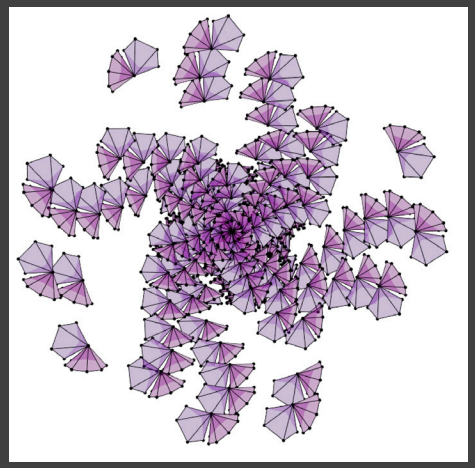


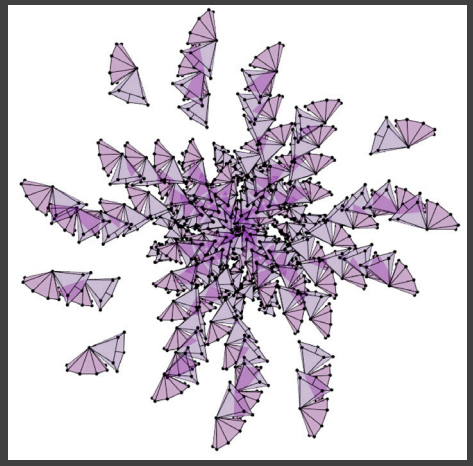
{ Form Follows Movement |

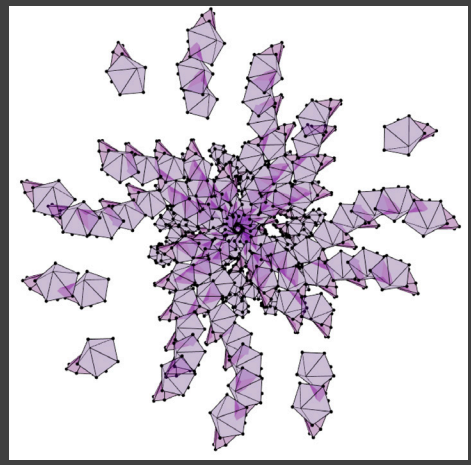


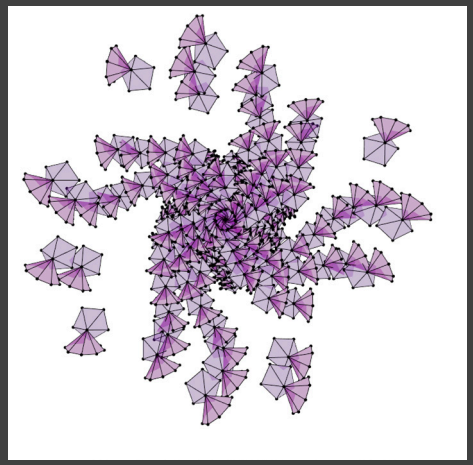
// Side View

// Plan Views

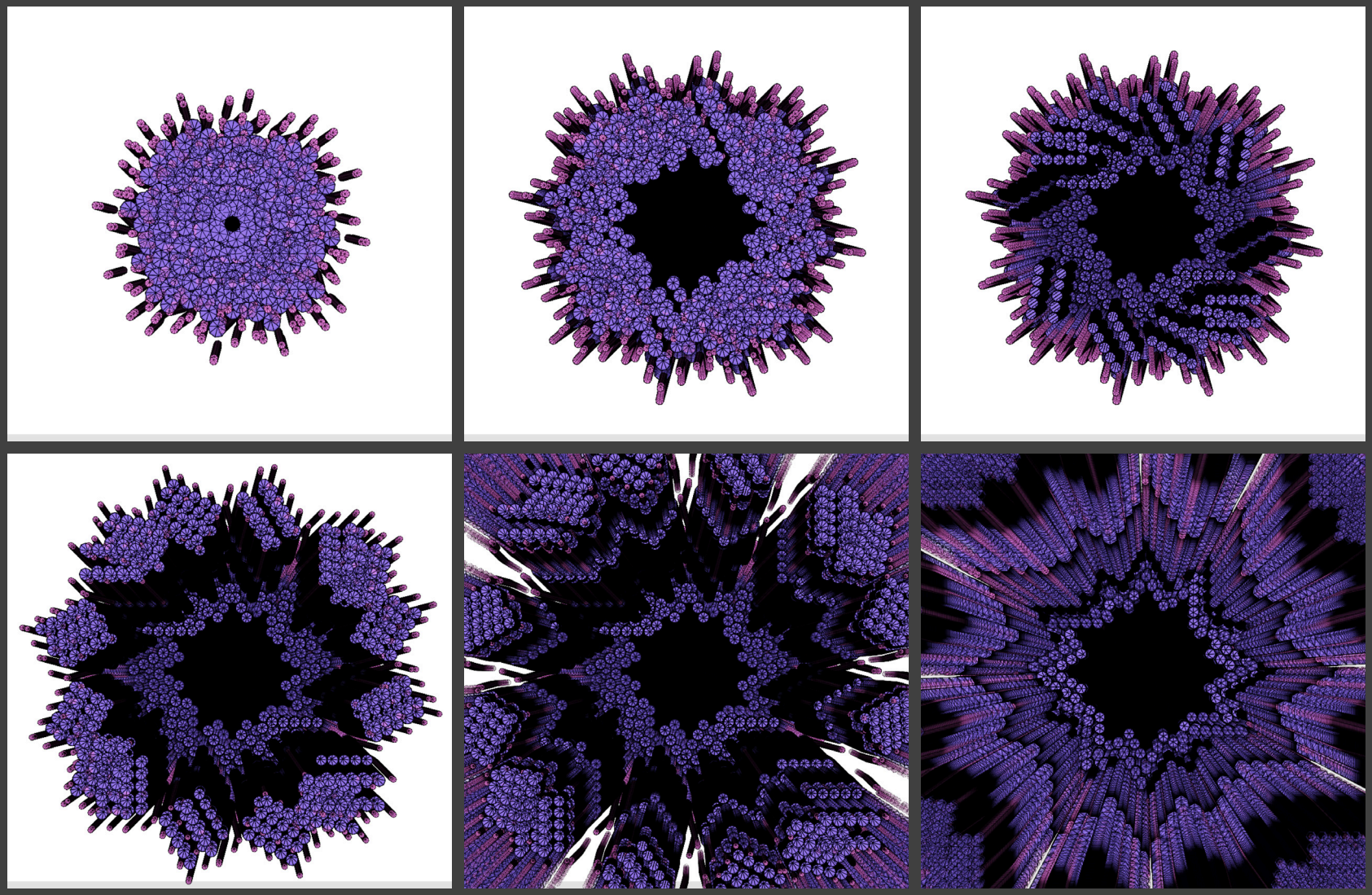








{ Preserving Memories | The Foot Print Effect

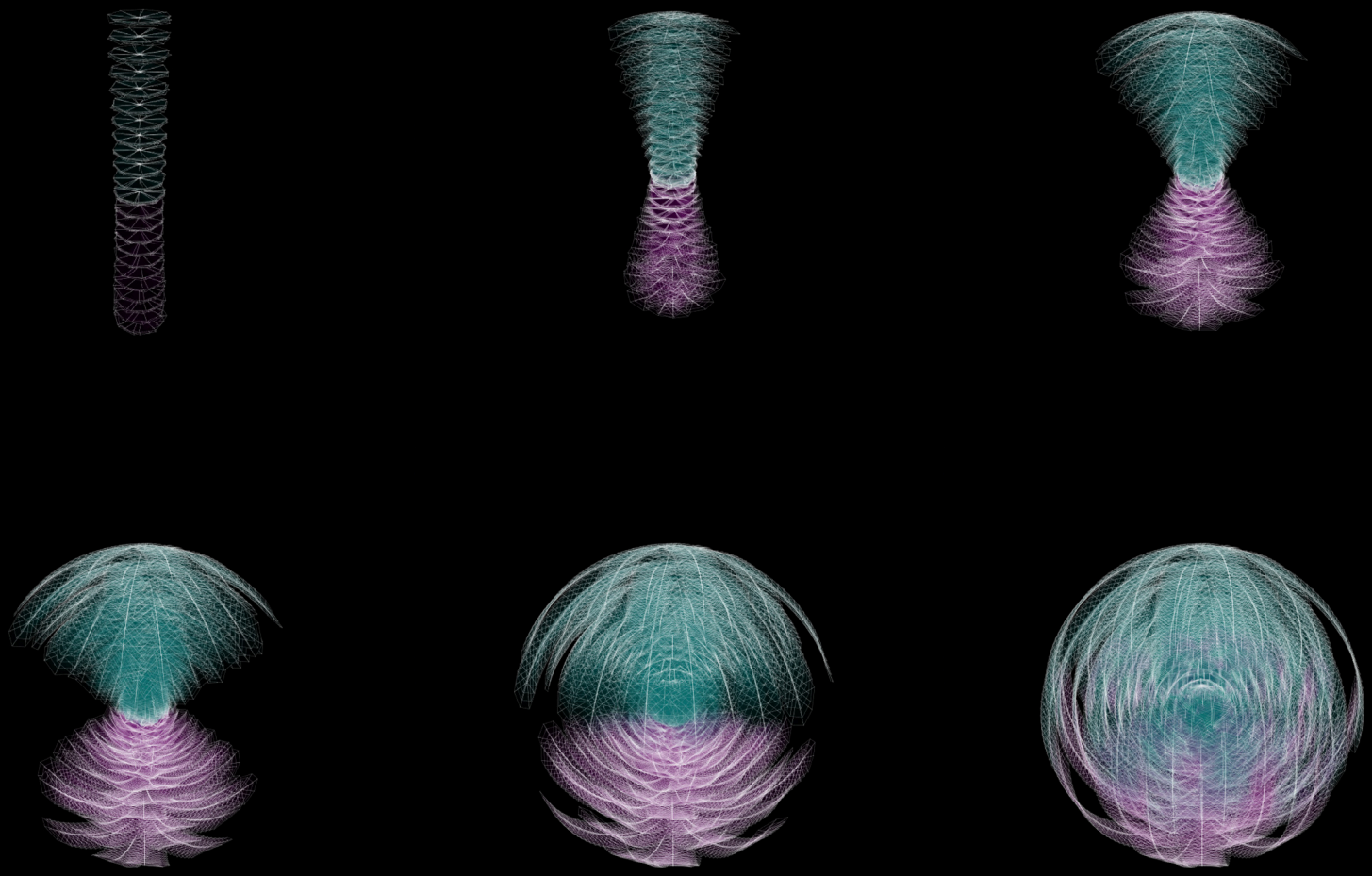


// Plan Views

// Interesting graphics appear when the code allows for movement to be tracked, leaving a foot print behind.

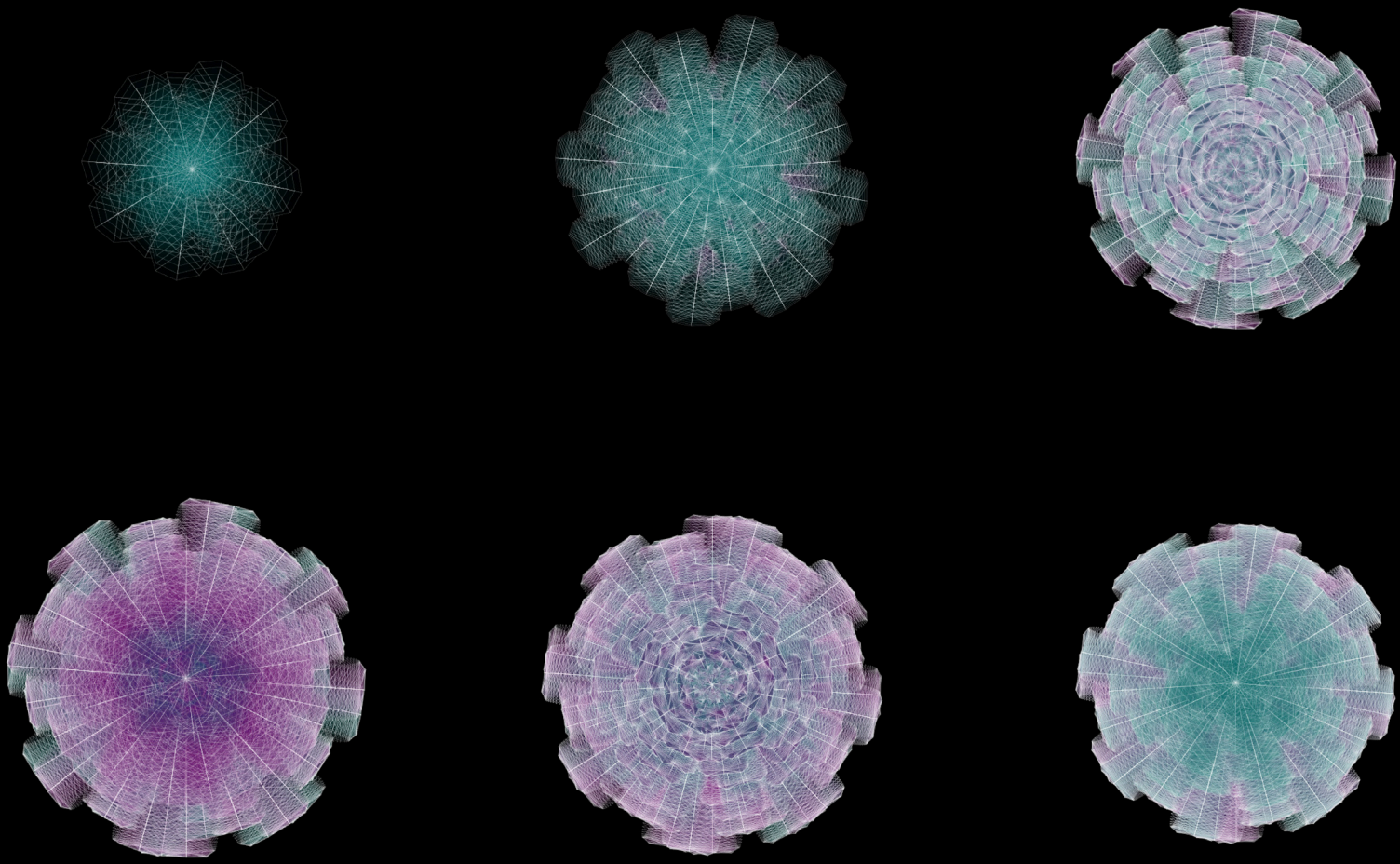
This method of creating form, suggests a new drawing technique for conceptual illustrations.

{ Preserving Memories | Part 1



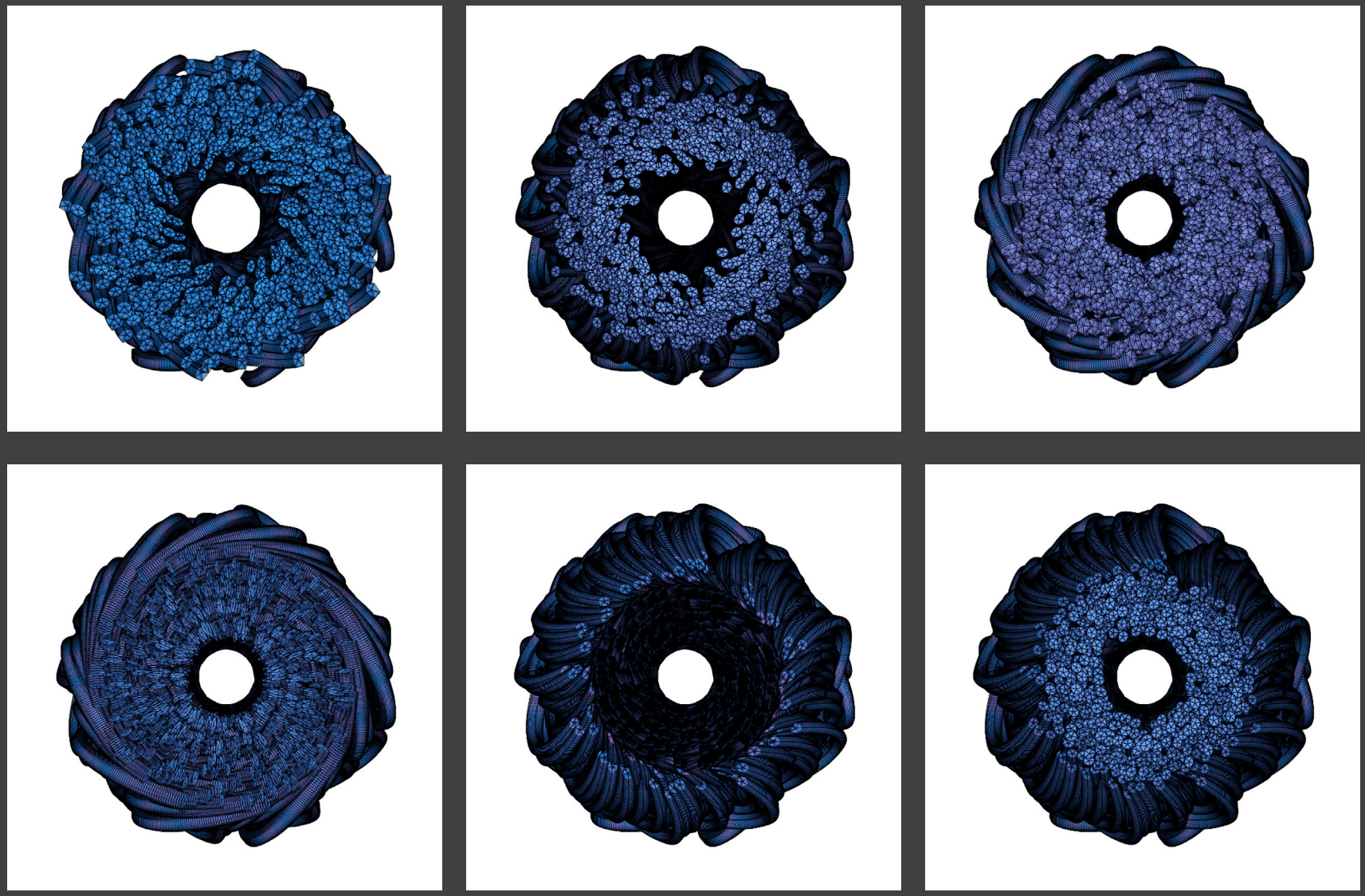
// Side Views

{ Preserving Memories | Part 2

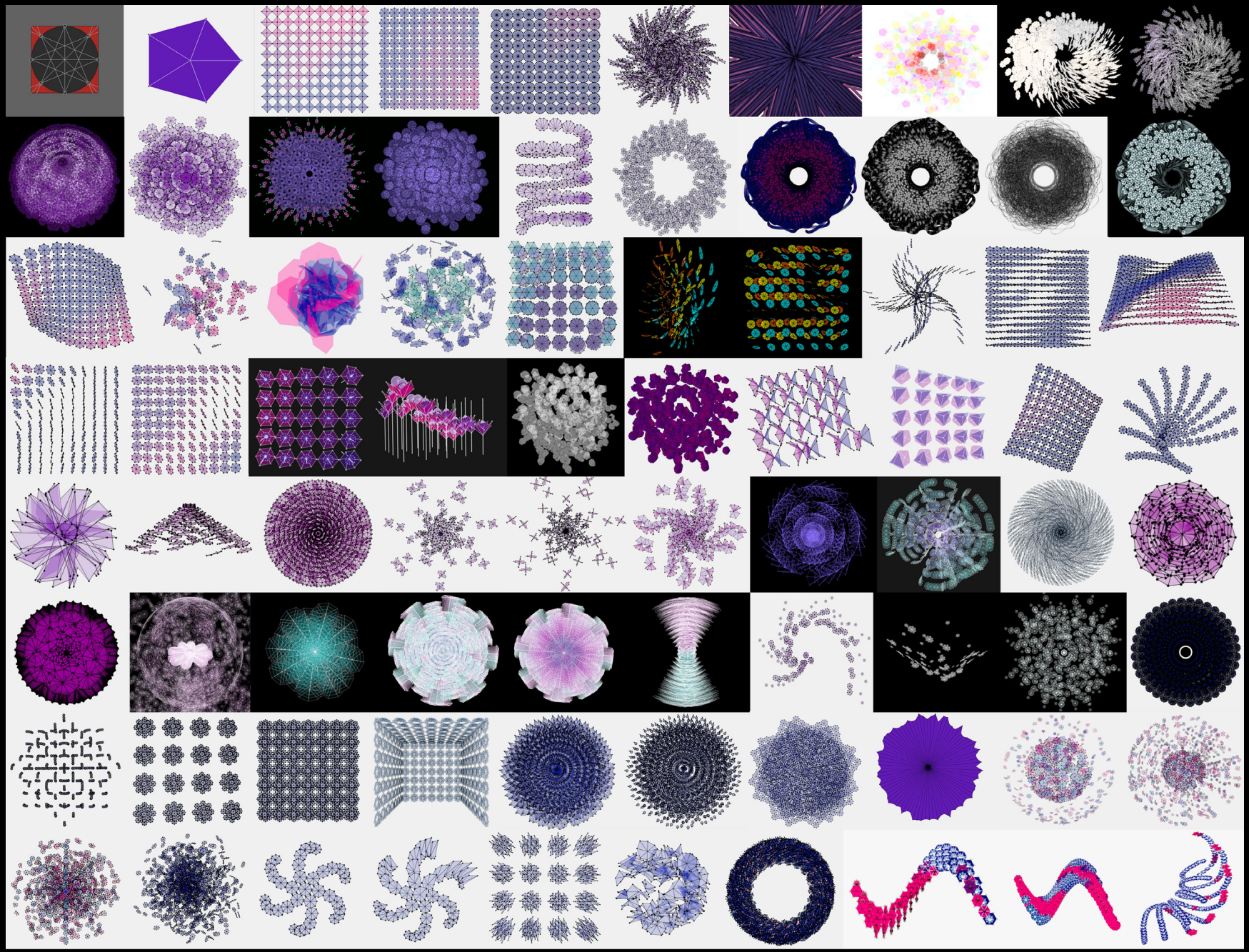


// Plan Views

{ Preserving Memories |



// Plan Views



// A collection of the generative coding outputs

{ The explorations suggest

// increase in the accessibility of new ideas and possibilities in the design process }

3 // Interactive Explorations

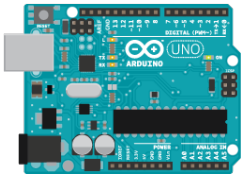
Integrating code with micro controllers and sensors

{ Arduino | Processing

// The interactive phase connects micro controllers and sensors to the designs generated in Processing.

// Arduino is an open-source electronics platform based on easy-to-use hardware and software and is intended for interactive projects. Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.[5]

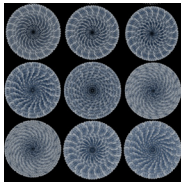
[Arduino]



// Interactive Sensors

// Interactive Code

[Processing]



// Generative Design

// Design Code

```
Atempt-2-4-3-breathing-Arduino1
 This example code is part of the public domain
 */
//#include <APsync.h> // Include the library

AP_Sync streamer(Serial); // Create an Object named streamer of type APsync
const int greenLEDPin = 9;    // LED connected to digital pin 9
const int redLEDPin = 10;     // LED connected to digital pin 10
const int blueLEDPin = 11;    // LED connected to digital pin 11

const int redSensorPin = A0;  // pin with the photoresistor with the red gel
const int greenSensorPin = A1;   // pin with the photoresistor with the green gel
const int blueSensorPin = A2;   // pin with the photoresistor with the blue gel

int redValue = 0; // value to write to the red LED
int greenValue = 0; // value to write to the green LED
int blueValue = 0; // value to write to the blue LED

int redSensorValue = 0; // variable to hold the value from the red sensor
int greenSensorValue = 0; // variable to hold the value from the green sensor
int blueSensorValue = 0; // variable to hold the value from the blue sensor

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);

  // set the digital pins as outputs
  pinMode(greenLEDPin, OUTPUT);
  pinMode(redLEDPin, OUTPUT);
  pinMode(blueLEDPin, OUTPUT);
}

void loop() {
  // Read the sensors first:

  // read the value from the red-filtered photoresistor:
  redSensorValue = analogRead(redSensorPin);
  // give the ADC a moment to settle
```

Arduino/Genuino Uno on /dev/cu.usbmodem1421

//Arduino interface

```
Atempt_2_4_3_breathing_Arduino1
import peasy.*;
PeasyCam cam;
int total=8;
float t=10;
PVector [] p= new PVector [total];
PVector orig= new PVector (0, 0, 0);
int[] nums = {255, 255,255};

import processing.serial.*;
Serial myPort;  // Create object from Serial class
String val;     // Data received from the serial port

void setup () {
  size(800, 800, P3D);
  smooth(8);
  cam= new PeasyCam (this, 4000);
  String portName = Serial.list()[1];
  myPort = new Serial(this, portName, 9600);
  float angle= 360.0/total;
  for (int i=0; i<total; i++) {
    float x= cos (radians(angle*i)) *100;
    float y= sin (radians(angle*i))*100;
    p[i]= new PVector (x, y, 20);
  }
}

void draw() {
  background (255);
int factor=3;
int bias=0;

  for (int a=0; a<=width; a+=200) {
```

Java

Console Errors

//Processing interface

// Processing is compatible with Arduino, allowing them to connect and exchange information and data.

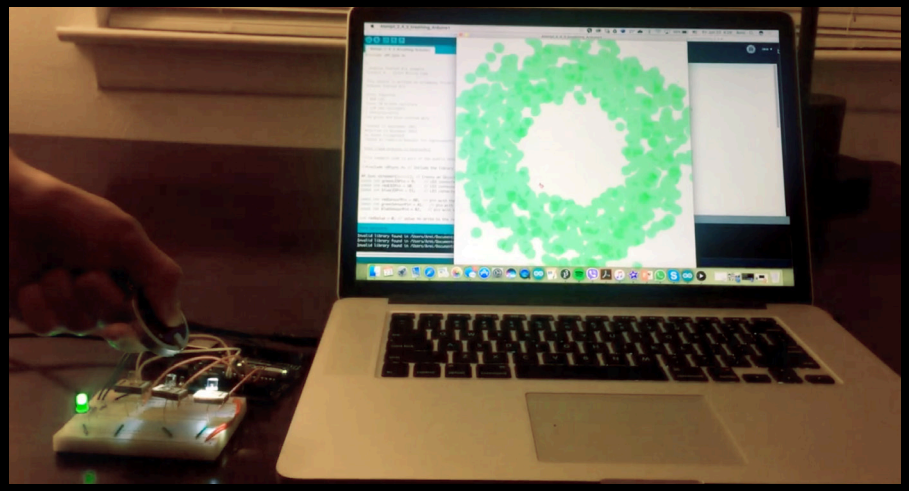
//In these explorations, Arduino was considered as the interactive aspect and Processing as the generative component.

{ Arduino | Processing

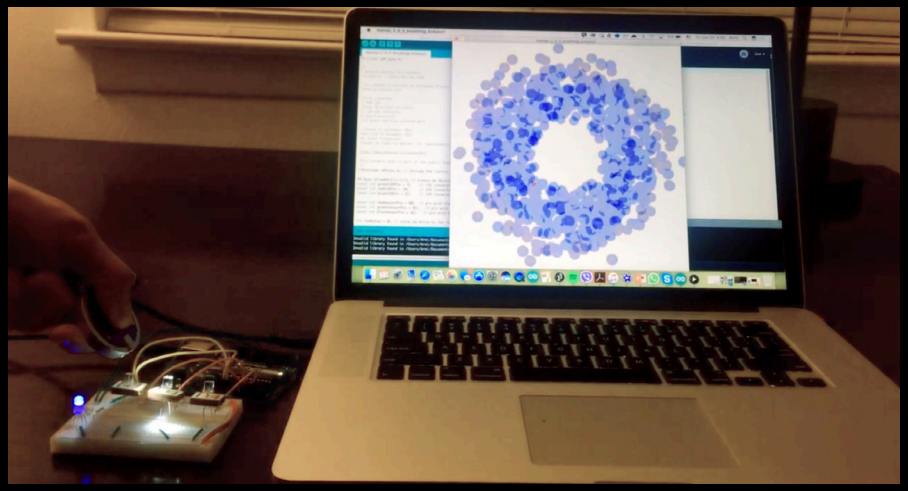
// Interactive studies consist of 4 components:
[1] Processing [2] Arduino [3]Arduino Uno micro controller
[4] Light sensors

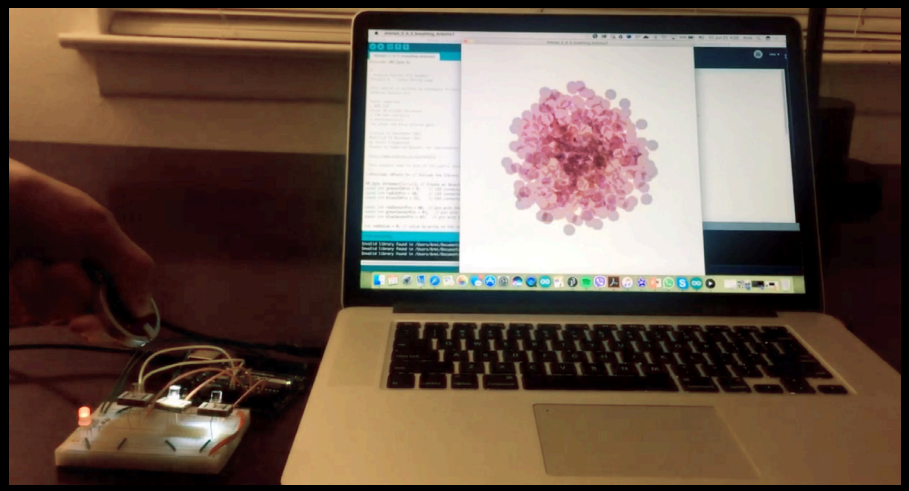
// The light sensors were placed in a circuit and connected to the Arduino Uno which then had to be plugged into the computer to transfer the brightness information which each Red, Blue and/or Green light sensor collected. The Arduino code allowed the data to show on the computer, as it operated the processing code to control the movement and color of the design.

// Green sensor activated

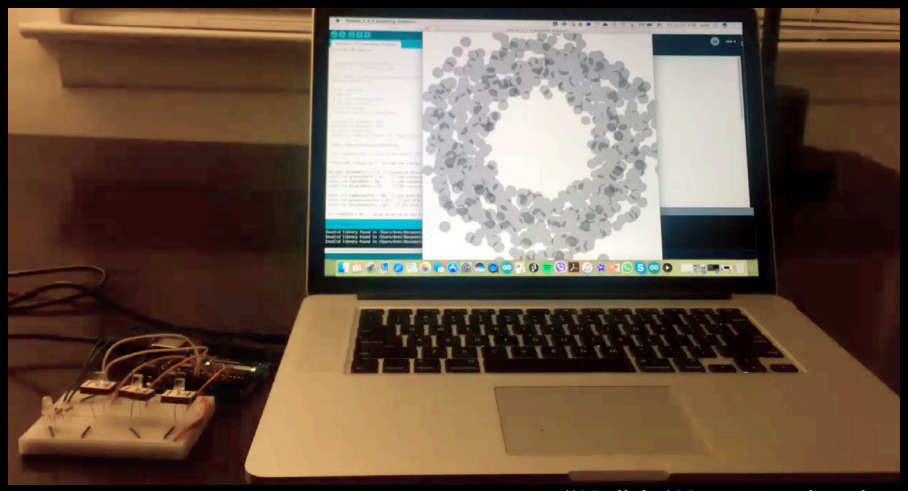


// Blue sensor activated





// Red sensor activated

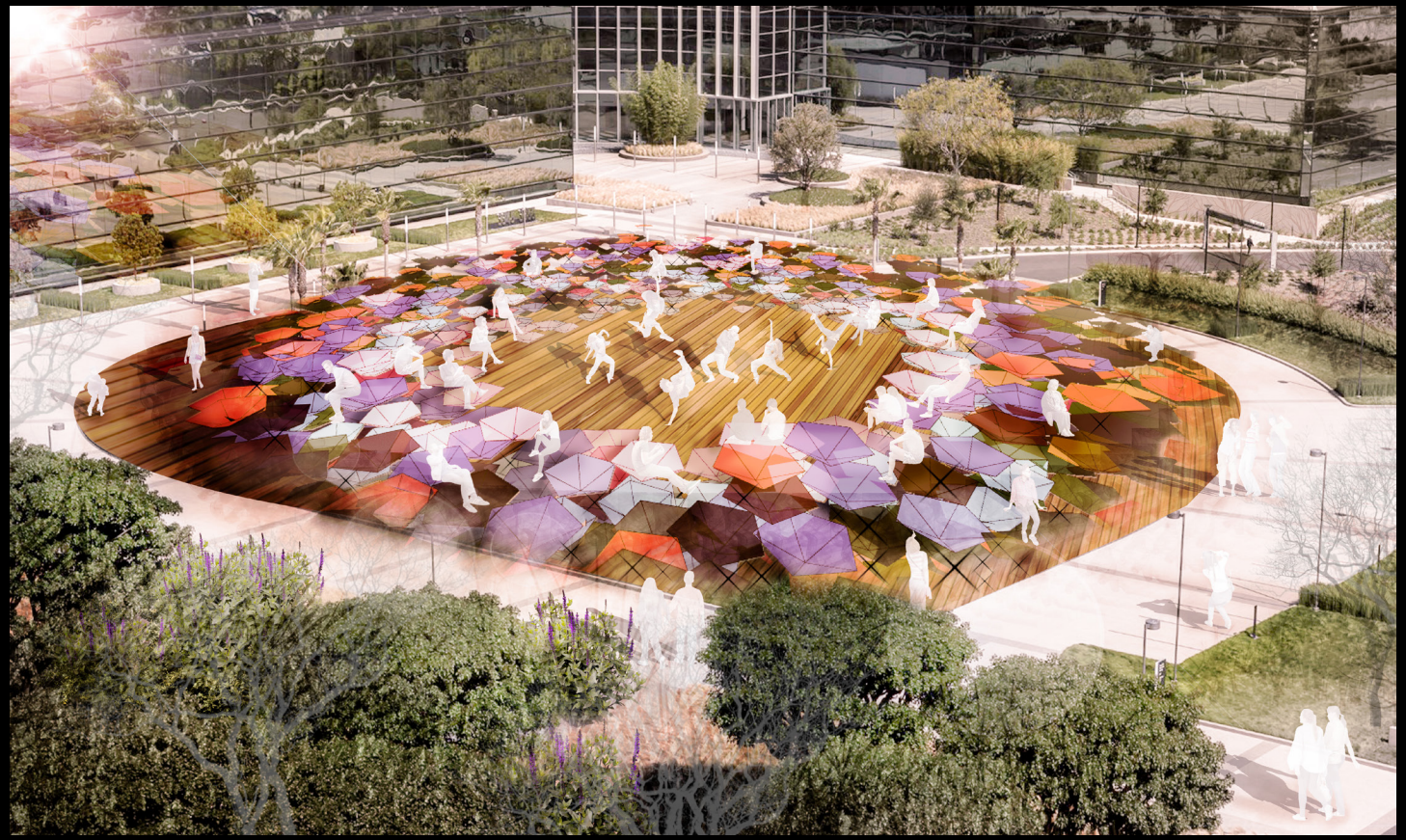


// No light | No sensor activated

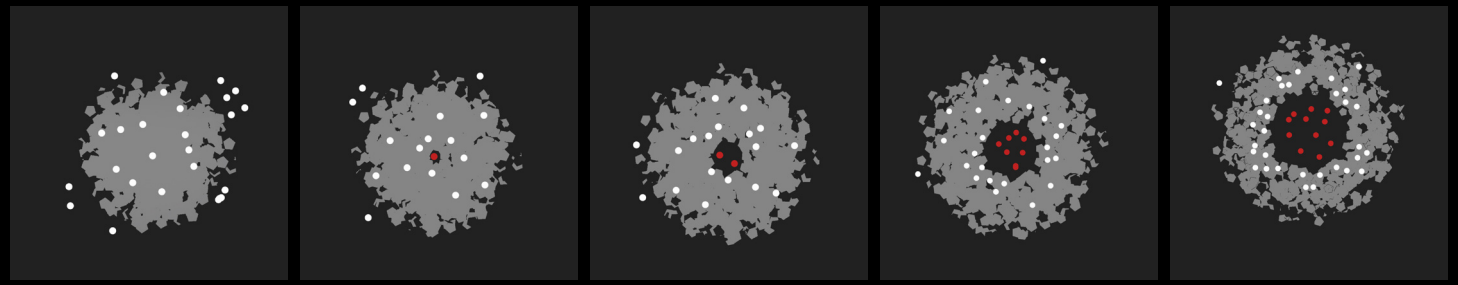
4 // Manifestations

Visualization and Development

{ Interactive Arena | Human Interaction



// This visualization suggests possibilities for the design development of a public arena. The spaces can be adjusted based on the activity type and the crowd population.



- Play | Study | Gather
- No Events

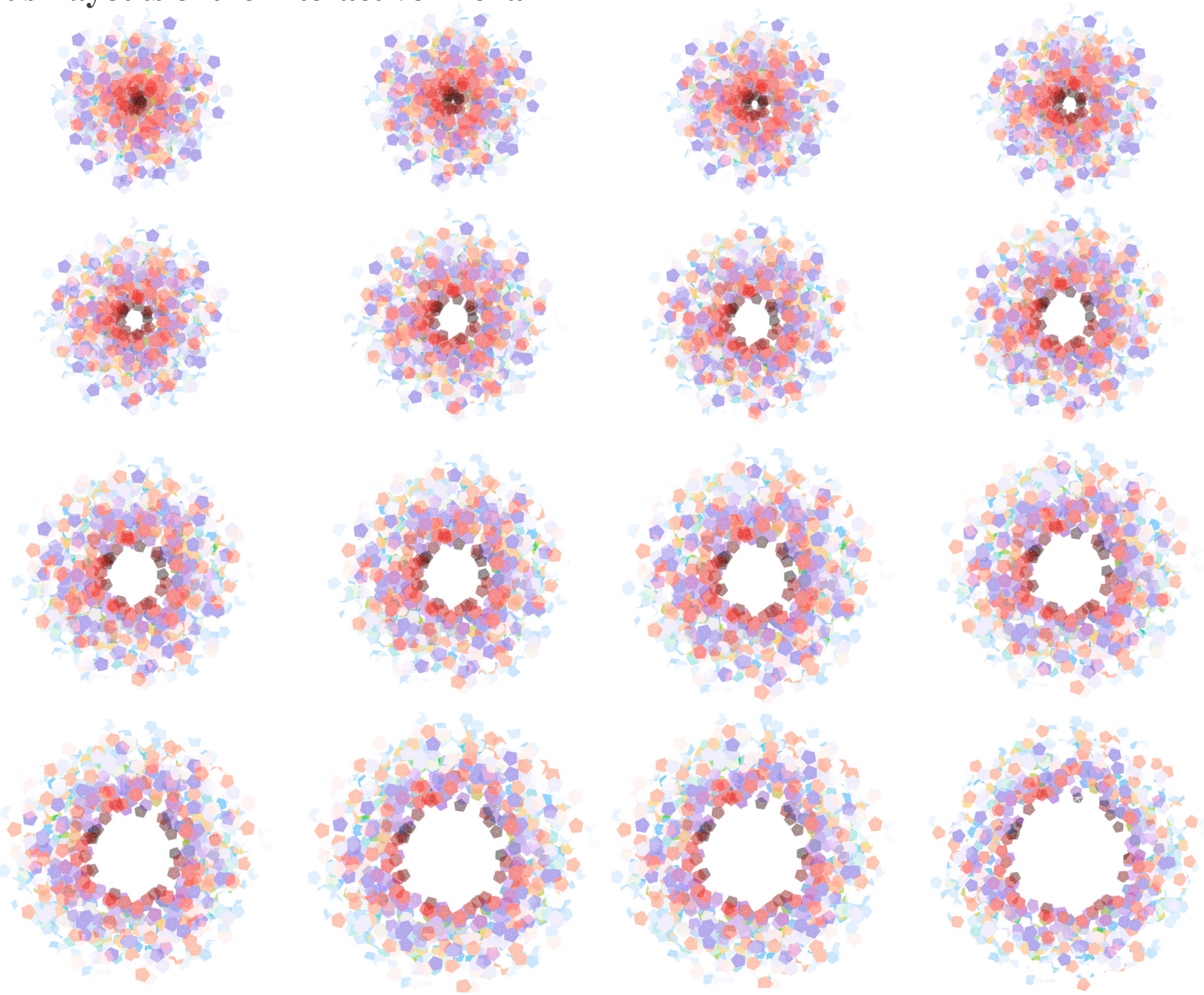
- Play | Study | Gather
- Solo Performance

- Play | Gather
- Couples Performance

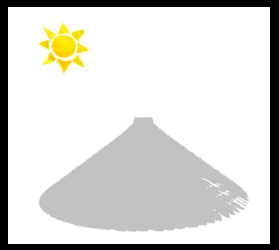
- Gather | Social Interaction
- Small Group Performances

- Social Interaction
- Large Group Performances

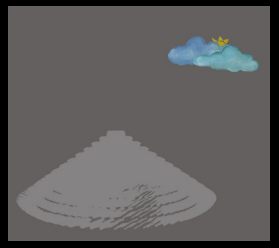
{ Various Layouts of the Interactive Arena



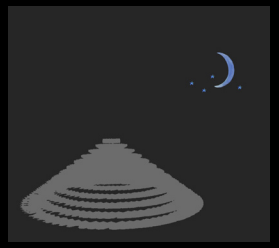
{ Interactive Off-Road Shelter | Environmental Interaction



// Clear sky | Shut



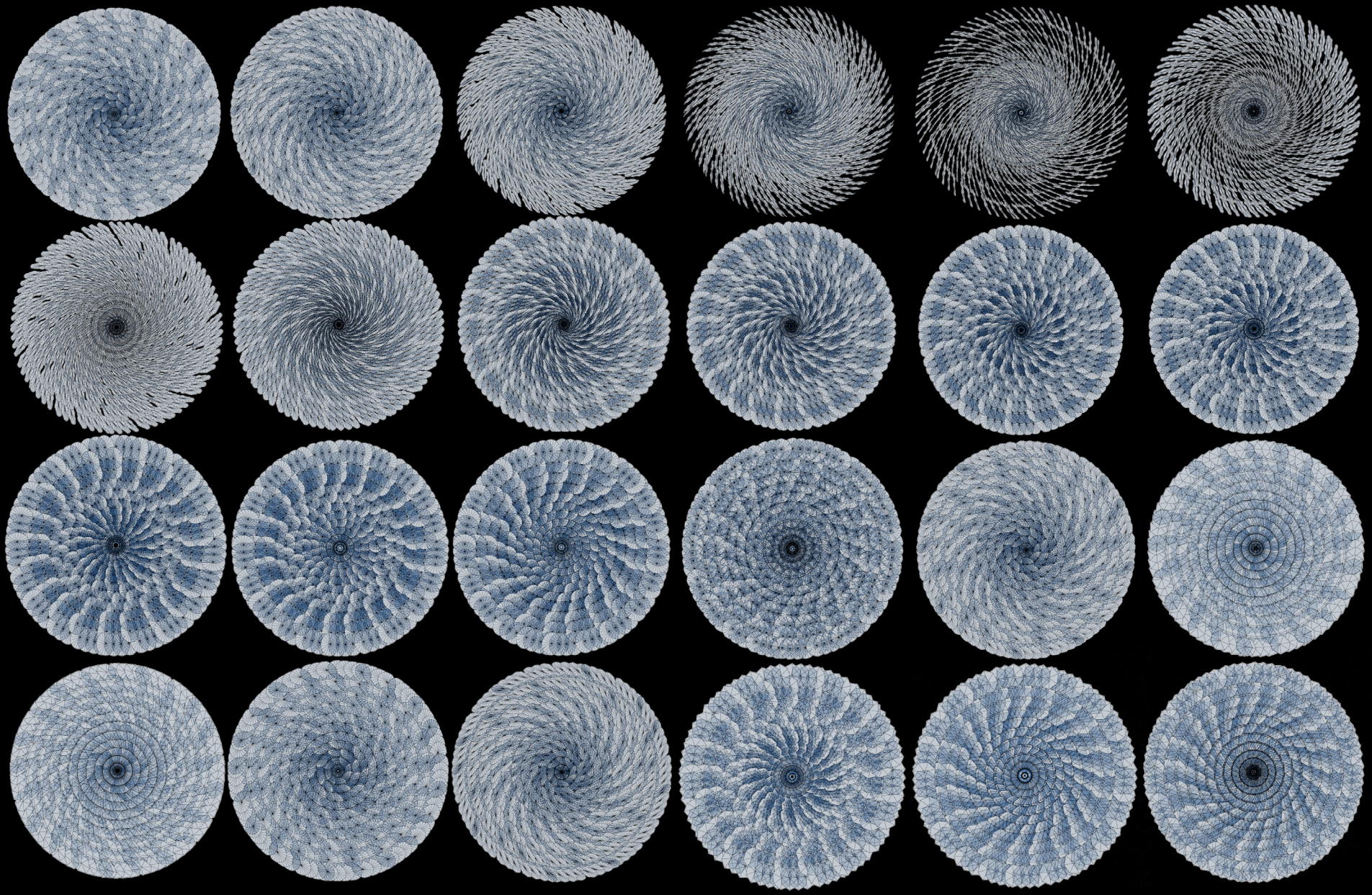
// Cloudy sky | Slightly open



// Night time | Fully open

// This visualization hints the possibility for an environmental interaction with the cone shaped huts. The shelters are sensitive to temperature and light with the louver system adjusting to changes in the environment.

{ Generation



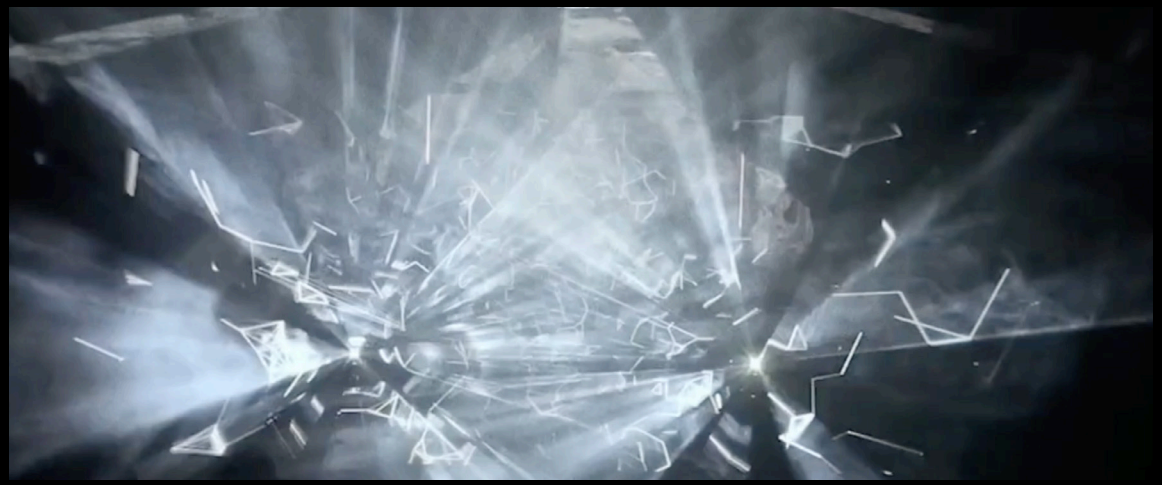
// Plan Views of the interactive hut system shows the generative aspect of the design |

{ Virtual Reality | Augmented Reality

// Virtual Reality is becoming more tangible and interactive design is finding a place in this area.
Virtual and augmented reality frees architects and designers from structural and physical constraints and allows them to create spaces with a new perspective.

// Stepping in the world of virtual reality, heightens possibilities for developing generative designs and enhancing the experience of space for users and audience.

{ Audio-Visual Performance | Projection | Projection Mapping



// In the entertainment industry, concerts and performing arts, Audio-visual systems and projection mappings are becoming more common and approachable.

// Architecture could also be created with this approach. With the use of generative and interactive design, the quality of spatial characteristics could be enhanced and a more pleasant user experience may become possible.

5 // Future Explorations

For the Years Ahead

Left to Explore {

// For architects and designers, taking advantage of virtual and augmented reality, audio-visual systems, projection and projection mapping, would mean reaching towards a wider range of possibilities to integrate design principles with technology.
The benefits of interactive and generative design in these areas are that with fewer physical constraints, ultimately uncountable design choices would be possible to imagine, develop and produce.
The hope is to answer to as many various user needs as possible and improve spatial experience for a wider range of people and activities.

// However, some important questions remain for further exploration and research.

// Many questions could be taken to further investigation.
Architecture will always carry a profound physicality and materiality and it is important to recognize the physical constrains.

[How could a physical impulse become part of the written code and how designers become aware of it during the coding phase?]

[How material could manifest its characteristics in the sense of virtual reality?]

[How does architecture become a virtual space which could provide physical characteristics?]

[In what ways can architects rethink compositions and forms in the context of code and algorithms to develop a broader range of concept and design possibilities?]

[How could interactive design enhance spatial experience and create opportunities for a wider range of purposes?]

}

// Interactive Design is { The Future of Architecture

// Architecture is { Revealing the Simple Complexity

References {

[1] Nexus Network Journal, Persian Architecture and Mathematics,Vol. 14, No.2 ,2012.

[2] Juhel, Alain. Touring Persia with a Guide Named… Hermann Weyl, Nexus Network Journal, Persian Architecture and Mathematics,Vol. 14, No.2 ,2012, 203-226

[3]Abdullahi, Yahya. Evolution of Islamic geometric patterns, Frontiers of Architectural Research, 2013, No. 2, Higher Education Press, 243–251.

[4] https://processing.org

[5] https://www.arduino.cc

{All images are by the author unless noted otherwise}

//Fig 1. | Source Unknown
//Fig 2. | http://www.ivodaskalov.com/iran | Photo credit: Ivo Daskalov
//Fig 3. | http://www.massimorumi.com | Photo credit: Massimo Rumi
//Fig 4. | Source Unknown
//Fig 5. | ©Eric Lafforgue
//Fig 6. | http://www.ivodaskalov.com/iran | Photo credit: Ivo Daskalov
//Fig 7. | Source Unknown
//Fig 8. | http://www.massimorumi.com | Photo credit: Massimo Rumi
//Fig 9. | School of Islamic and Geometric Design | http://www.sigd.org/resources/
//Fig 10. | School of Islamic and Geometric Design | http://www.sigd.org/resources/
//Fig 11. | Nexus Network Journal, Persian Architecture and Mathematics,Vol. 14, No.2 ,2012. 227-250 | Photo credit: unknown
//Fig 12. | Frontiers of Architectural Research, 2013, No. 2, 243–251 | Photo credit: unknown