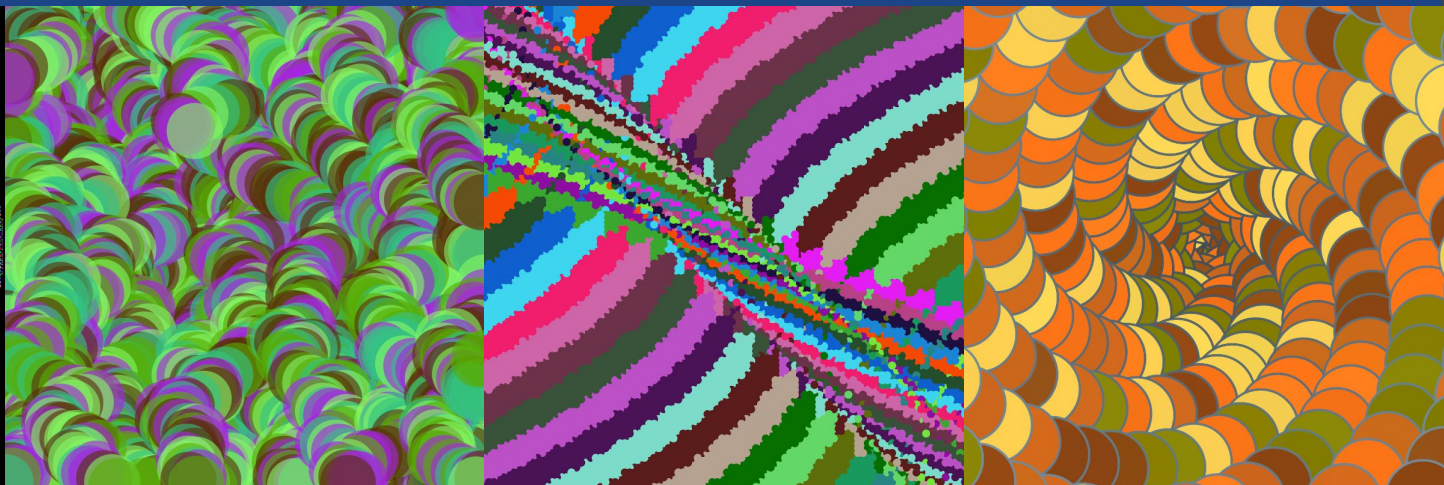
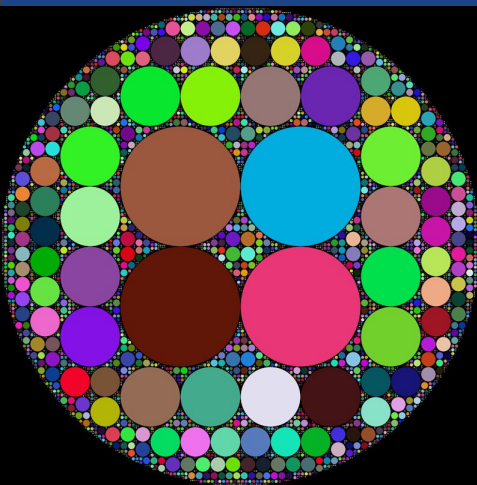


Students Teaching Students Computer Art and Graphics

By **Michael Wehar**

Special thanks to my students and collaborators E. Brickner, X. Dong, J. Gallardo Moreno, O. Khan, X. Li, C. Liu, J. Mancini, M. Newman-Toker, R. Oet, V. Sumano, L. Suresh, P. Tone, and A. Zhang.



Capstone Projects in Computer Graphics

- At my institution, students studying Computer Science are required to complete a capstone project.
- These projects are related to courses or faculty research.
- For the past three years, I have supervised 9 students within 3 cohorts on a research project called AlgoArt (algoart.org).

AlgoArt Platform

- **Goal:** Build bridges between computing and art. Offer a low barrier of entry to create procedurally generated art and graphics.
- Web-based Platform consisting:
 - Open Source Creator Studio (on [GitHub](#))
 - Digital Gallery including user reviews and feedback
- Drawing algorithms are written in JavaScript (JS)
 - Every algorithm follows a framework:
 - Methods for **initialize, start, pause, reset, drawOneStep**
 - Has a params JS file for customization

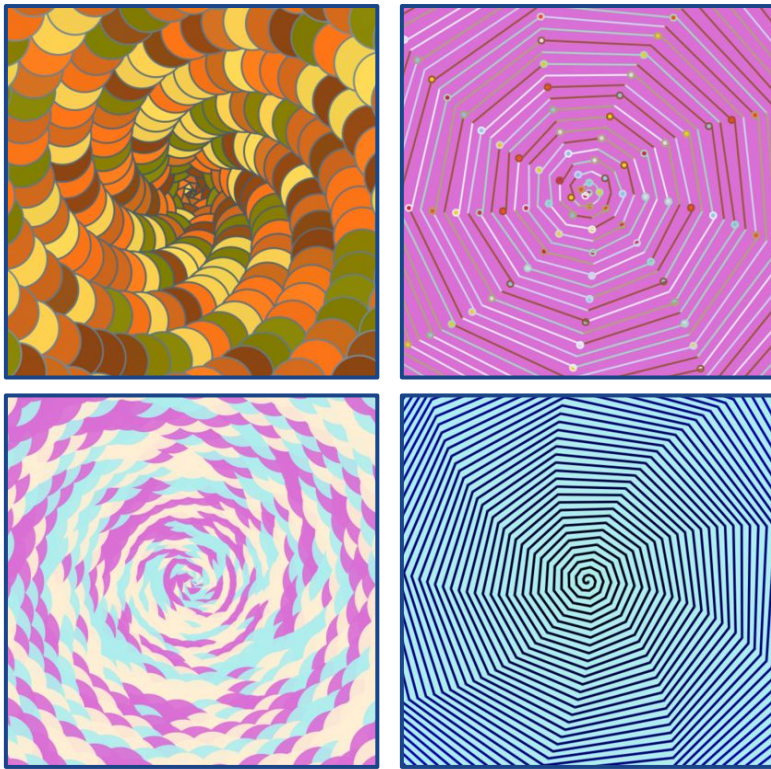
Multi-Generational Capstone (MGC)

- Student capstone projects have been continuing to build out the AlgoArt Platform over the past three years.
 - New students pick up where the previous left off
 - Every student has a well-defined subproject
 - Ex. add a new feature, UI redesign, create a drawing algorithm
 - Each student learns from the prior student work and contributes to the future student learning

Another MGC: [Map-based Educational Tools for Algorithm Learning \(METAL\)](#)

What Did Students Learn?

- Benefits for Students
 - Learn basic skills related to computer graphics
 - Fulfill their capstone requirement
 - Build upon the work of those who came before them
- Examples of Computer Graphics Concepts
 - Coordinate Systems
 - Basic Shapes, Polygons, Curves, and Paths
 - Translation, Rotation, and Scaling
 - Color Models, Palettes, Gradients, and Opacity
 - Animation, Collision Detection, and Z-Order

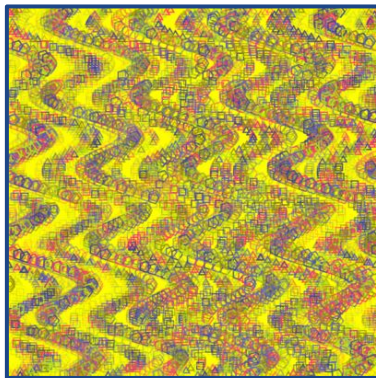
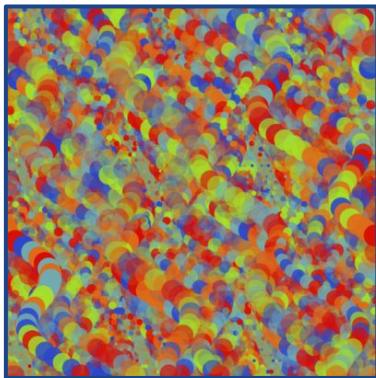
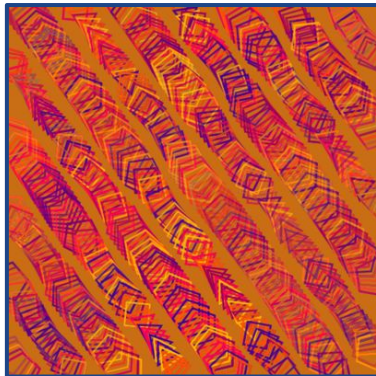
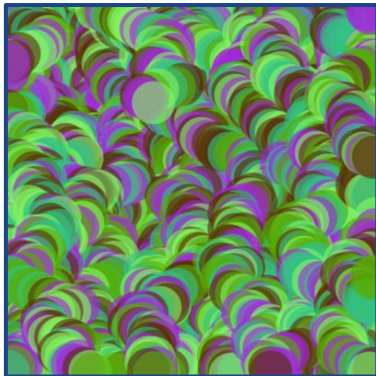


Multiple overlapping (left), discretized classical (right)

Developed by L. Suresh

Coordinate Systems (Spirals Algorithm)

- Replicates Archimedean Spiral
- Converts polar coordinates to Cartesian coordinates
- Parameters to adjust the spiral style (e.g. discretized)



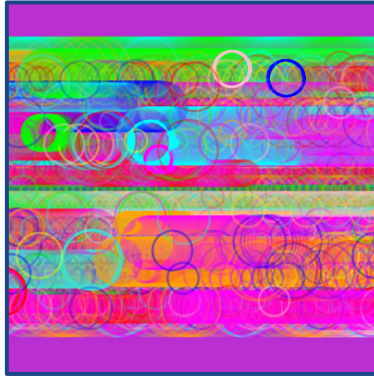
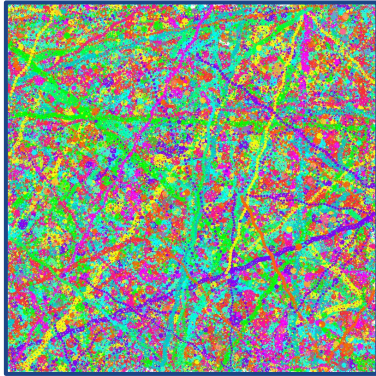
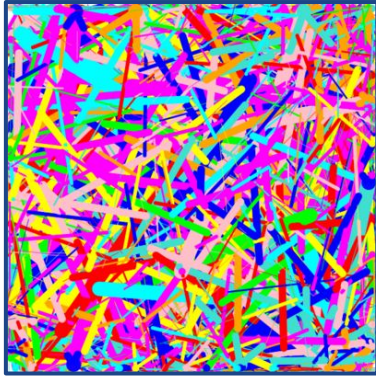
Filled circles (left), variety of polygons (right)

Developed by A. Zhang

Curves and Paths

(Vines Algorithm)

- Visualizes how vines curve and wind around over time
- Implemented several paths:
linear, circular, sinusoidal, hybrid
- Parameters to add randomness



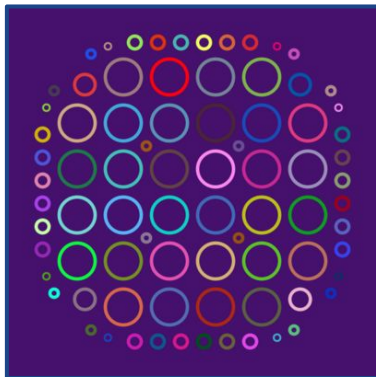
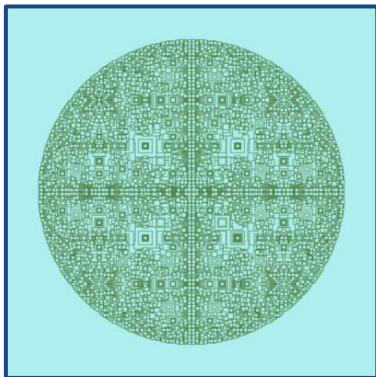
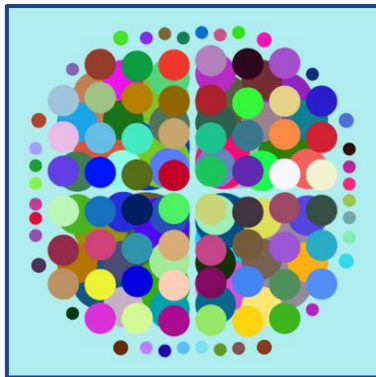
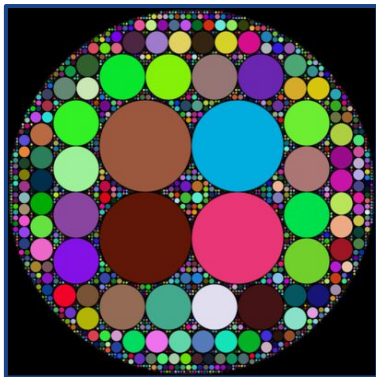
Variety of angles (left), horizontal movement (right)

Developed by O. Khan

Collision Detection

(Collisions Algorithm)

- Simulates balls moving around and colliding in 2D
- Balls can bounce, combine, or break apart based on parameters
- Conservation of momentum



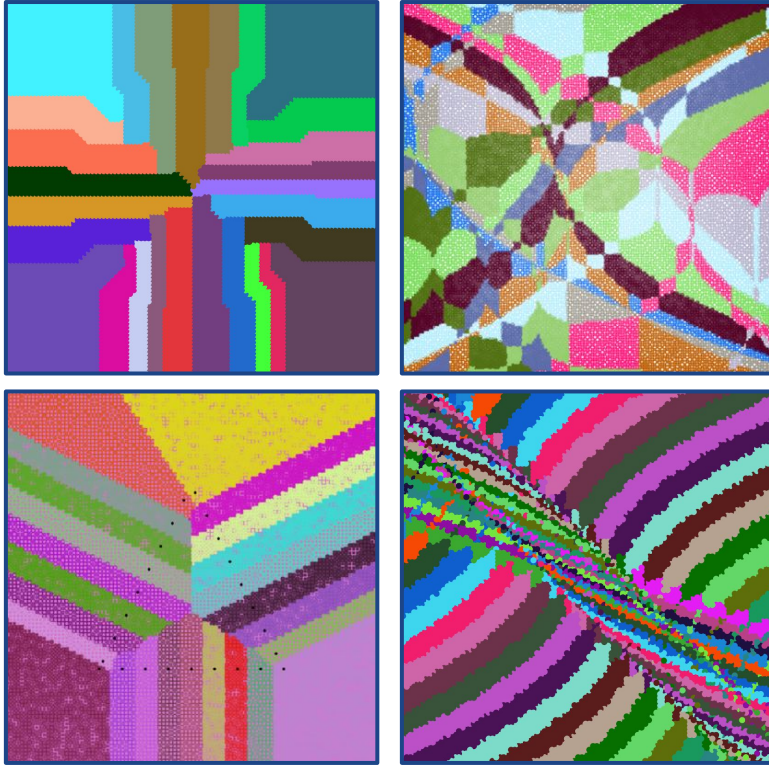
Densely packed (left), sparsely packed (right)

Developed by M. Newman-Toker

Fractals

(Fractals Algorithm)

- Iteratively draws shapes into regions avoiding overlaps
- Leads to drawing smaller shapes into smaller regions
- Inspired by Sierpiński Triangle



Straight line boundaries (left), curved boundaries (right)

Developed by J. Gallardo Moreno

Voronoi Diagrams (Voronoi Algorithm)

- Select seed points
- Create regions based on which seed point is closest
- Euclidean distance as well as custom distance functions are used to measure closeness

Creator Studio Demo

- **How does it work?**
 - Select algorithm
 - Customize parameters
 - Run algorithm
 - Watch image generation
- **Real-time animation provides:**
 - Visual explanation of how an algorithm works
 - Visual feedback for the student algorithm developer

Future Directions

- Students appear to be learning more with each subsequent cohort.
 - As the platform continues to evolve, will its educational value increase for each future student cohort?
- Generated artworks demonstrate student learning.
 - Could artworks act as visual proof of a student mastering a technical concept?
- Viewing the AlgoArt Platform as an educational tool.
 - Could usage of the platform be beneficial in introduction to programming or computer graphics curriculums?

Thank you!

Visit our [GitHub repo](#)

Also, see [AlgoArt.org](#) (work in progress!)

We acknowledge support for this work from the Swarthmore
College Research and Academic Division Funds.