Exploring the Arts Online with the Wolfram Language

Christopher Carlson
Wolfram Research, Inc
100 Trade Centre Drive
Champaign, IL, 61820, USA
E-mail: carlson@wolfram.com

Abstract

The Wolfram Language is one of the quickest paths from ideas to explorations, whether you're working with graphics, sound, 3D structures, text, images, or pure math. A significant barrier to the use of the Wolfram Language has fallen recently with the introduction of the Wolfram Open Cloud, which lets anyone with a browser do Wolfram Language programming online. I'll show you how to get started exploring with a free account, and we'll explore together, sharing discoveries as we go.

Introduction

The Wolfram Language is familiar to many as *Mathematica*, where it was born and incubated for 25 years until it recently escaped the confines of the desktop with the rollout of the Wolfram Open Cloud [1]. Anyone with a browser can now do Wolfram Language programming online—in a free account for casual use or on a subscription basis for more resource-intensive work. The language has been popular among Bridges attendees for exploring and implementing ideas. I use it nearly every day for quick, 10-minute explorations of graphical phenomena I discover online.

This workshop is for anyone curious about where programming with the Wolfram Language could take them. Artists and mathematicians will discover a new way of making abstract ideas concrete. Teachers of the arts and programming will find a language that is readily assimilated by middle and high school students and that gives quick and satisfying results that motivate learning.

Workshop participants will need a laptop with Internet connectivity. Some familiarity with the basic notions of programming will be helpful, but knowledge of the Wolfram Language is not necessary.

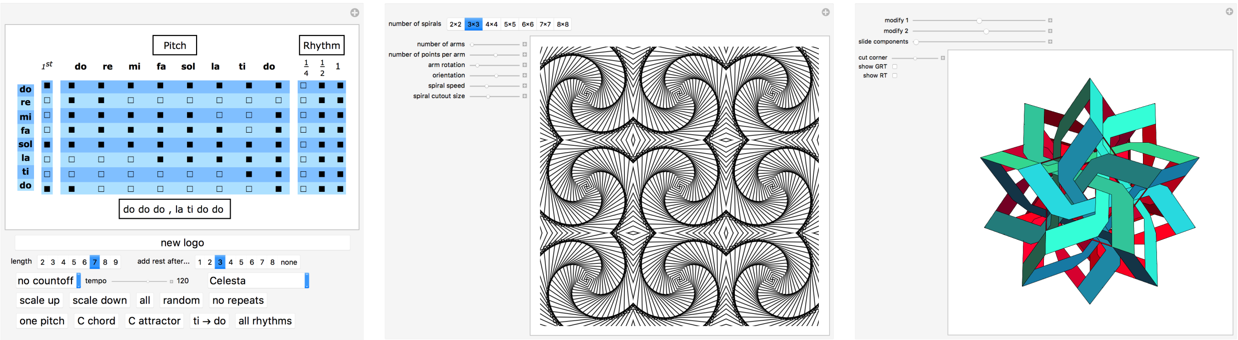

Figure 1: *Some Wolfram Language explorations [2][3][4].*

The Wolfram Language

The Wolfram Language differs from conventional programming languages in several ways that make it particularly suited for exploring the arts.

First, the language is interpreted, meaning that you can write code and immediately execute it. There are no intervening compile or link steps to slow you down. That makes it particularly easy to explore ideas incrementally, adding or changing bits as you go and immediately evaluating to see the effects. With the language's notebook interface, you can organize code, outputs, textual notes, and interfaces into a complete, documented record of your exploration.

Second, the Wolfram Language is ultra high-level, meaning that you can cover a lot of territory with just a few keystrokes. (It's perhaps the only language where it's actually possible to write interesting programs in a single 128-character Tweet. You can see examples at the Tweet-a-Program site [5].) In contrast to many languages, where the goal is minimalism, the Wolfram Language endeavors to include a broad swath of high-level, sophisticated algorithms in the language itself. Programming with the Wolfram Language is often more a matter of assembling algorithmic components than writing code yourself from scratch. "Superfunctions" like `Classify` and `Solve` encapsulate entire domains into ultra high-level functions that automate making the often bewildering choices of algorithms and parameters so that you can concentrate on the task rather than the technology.

The Wolfram Language's algorithmic functionality is wide as well as deep. Its 5000+ functions cover graphics (2D and 3D), sound (both MIDI and arbitrary waveforms), image processing, interface construction, geometric regions and meshes, and linguistic analysis—to name a tiny cross-section of functionality related to the arts. And although its functionality is vast, it is exceptionally well integrated. You won't run into roadblocks because one corner of the language doesn't fit well with another. That means that you can follow your explorations wherever they lead.

The single most powerful feature for exploration is the `Manipulate` function, which makes it trivially easy to attach an interactive interface to any concept. `Manipulate` has proved so useful that there is an entire website dedicated to it, the Wolfram Demonstrations Project [6], with over 10,000 interactive explorations, some examples of which are shown in Figure 1.

A Sample Exploration

The workshop explorations will all be `Manipulate` interfaces. It's what I use every day to explore ideas that I run across online. Here's an example of how one of those explorations works, and a taste of what we'll be doing in the workshop.

While I was browsing a collection of historic logos, I ran across the Mercedes Benz logo (Figure 2, left side) and wanted to explore variations of it such as changing the relative weights of the star and circle or the geometry of the star.

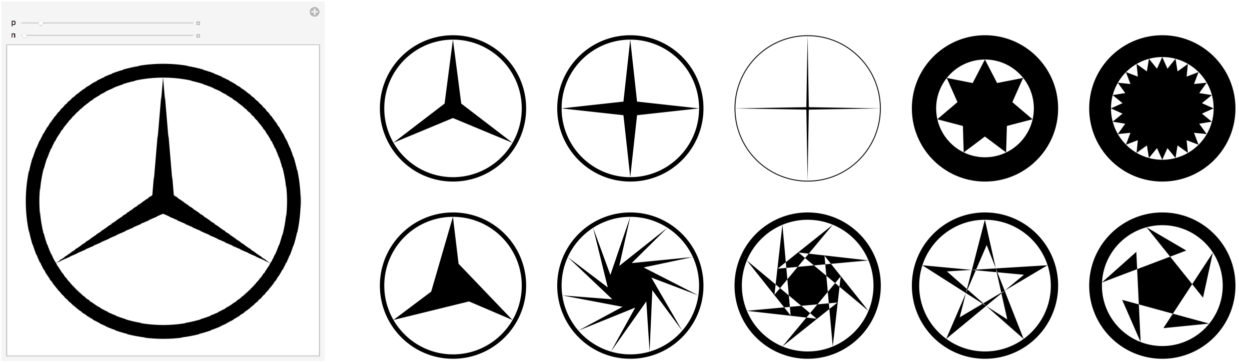

Figure 2: Mercedes Benz logo interface and explorations.

I started by writing code to draw the star. The first `CirclePoints` expression below gives three points evenly spaced around a unit circle starting at 90°. The second gives three more points with a radius of .1 and starting at 150° so that they fall angularly between the first points. `Riffle` shuffles the two lists together, and `Polygon` draws a black polygon with the points as vertices:

```
Graphics[
    Polygon[Riffle[
        CirclePoints[{1, 90°}, 3],
        CirclePoints[{.1, 150°}, 3]
    ]]]
```

Next, I added the black rim around the star by drawing a radius 1 black disk, and on top of that a radius .9 white disk:

```
Graphics[{
    Disk[{0, 0}, 1],
    {White, Disk[{0, 0}, .9]},
    Polygon[Riffle[
        CirclePoints[{.9, 90°}, 3],
        CirclePoints[{.1, 150°}, 3]
    ]]}]
```

I made that basic, static graphic interactive with `Manipulate` so that I could vary the number of points on the star and the sharpness of its points. The sharpness is controlled by the variable p, which varies from 0 to 1 with an initial value of .1. Changing `p` changes the radius `.9 p` in the second `CirclePoints` expression, which governs how deep the star's indentations are:

```
Manipulate[
    Graphics[{
        Disk[{0, 0}, 1],
        {White, Disk[{0, 0}, .9]},
        Polygon[Riffle[
            CirclePoints[{.9, 90°}, n],
            CirclePoints[{.9 p, (90 + 180/n)°}, n]
        ]]}],
    {{p, .1}, 0, 1},
    {{n, 3}, 3, 50, 1}
]
```

The resulting `Manipulate` interface is shown on the left side of Figure 2. By dragging the sliders, I could explore variations in the geometry of the star. It took perhaps 5 minutes to go from the urge to explore the logo to an interface I could explore it with. It is similarly straightforward to add controls for the thickness of the rim or to twist the outer vertices of the star relative to the inner ones. Some results of my explorations with those additional controls are shown on the right of Figure 2.

The Workshop

In the workshop, I'll show you how to get started with a Wolfram Open Cloud account and what resources you can take advantage of to come up to speed with the Wolfram Language [7]. I'll briefly review some of the work that artists and Bridges attendees have done with the Wolfram Language, and then we'll explore.

Participants will start with examples from a collection of explorations in various domains and modify code from there to move into new territory. Each participant will work independently, but share interesting discoveries with the group along the way. By the end of the workshop, participants should have gained enough familiarity with the Wolfram Open Cloud to continue exploring at home.

References

[1] Wolfram Open Cloud, https://develop.open.wolframcloud.com/
[2] M. Trott, "Left-Right-Rotating Spiral Array" from the Wolfram Demonstrations Project http://demonstrations.wolfram.com/LeftRightRotatingSpiralArray/
[3] J. Kiehl, "Musical Logo Widget" from the Wolfram Demonstrations Project http://demonstrations.wolfram.com/MusicalLogoWidget/
[4] S. Kabai, "Great Rhombic Triacontahedron Sculpture" from the Wolfram Demonstrations Project http://demonstrations.wolfram.com/GreatRhombicTriacontahedronSculpture/
[5] Tweet-A-Program, https://twitter.com/wolframtap
[6] The Wolfram Demonstrations Project, http://demonstrations.wolfram.com/
[7] Wolfram Language, http://www.wolfram.com/language/