

Turtle Temari

Michael Eisenberg^a, Antranig Basman^b, Sherry Hsi^c, and Hilarie Nickerson^d

^{a,b,d}Department of Computer Science, University of Colorado, Boulder CO 80309-0430

^cLawrence Hall of Science, Berkeley, CA 94720

^aemail: duck@cs.colorado.edu

Abstract

Temari balls are mathematical craft objects in which patterns of multicolored thread are wound around a spherical surface to create intriguing, sometimes remarkable patterns. In this paper, we demonstrate an interactive programming system, *Math on a Sphere* (MoS), that enables users to create and explore temari-like designs on a spherical surface represented on a computer screen. The programming language of MoS is based on the venerable "turtle graphics" elements characteristic of the traditional Logo language; unlike those traditional systems, however, in MoS the turtle does not draw lines on a plane, but on a representation of a sphere. Thus, MoS provides a medium in which to create striking patterns, and at the same time serves as an introduction to fundamental ideas in non-Euclidean geometry. We step through the creation of several temari designs based on the symmetries of Platonic solids, and show how the reader may access and play with the system on the Web. We conclude with a brief discussion of ongoing and future work related to the MoS system.

Introduction

Temari balls [3, 9] are traditional Japanese mathematical craft objects in which multicolored thread is embroidered onto a spherical base to create visually striking, decorative objects. A perusal of recent craft books on the subject—or, for that matter, an image search on the Web—should quickly convince the reader that temari balls can be both exquisite and mathematically provocative, as their decorations may be thought of as visual representations of spherical symmetry groups.

In this paper we employ a computational tool called *Math on a Sphere* to show how simple programming constructs can be used to create temari-like patterns on a representation of a sphere. Math on a Sphere (which we abbreviate as MoS) is a programming environment created in our lab [4, 6], and is freely available on the World Wide Web [www.mathsphere.org]. By exploring graphical programming on the sphere, users may not only replicate the designs shown in this paper, but may create an infinite variety of their own patterns. Thus, the MoS system is both an artifact for artistic creation on the computer screen, as well as a medium for exploring fundamental ideas of non-Euclidean geometry. Moreover, as we will discuss later in the paper, MoS-created patterns may be used as templates or guidelines for physical creation of temari objects, or may be displayed on a large-scale spherical screen.

By way of preface, it may be motivational to present the patterns that we will illustrate in this paper. Figure 1 shows four temari-like patterns, based on cubical, octahedral, icosahedral, and dodecahedral symmetry; these are representative of the myriad types of decorative strategies that can be explored with the MoS system.

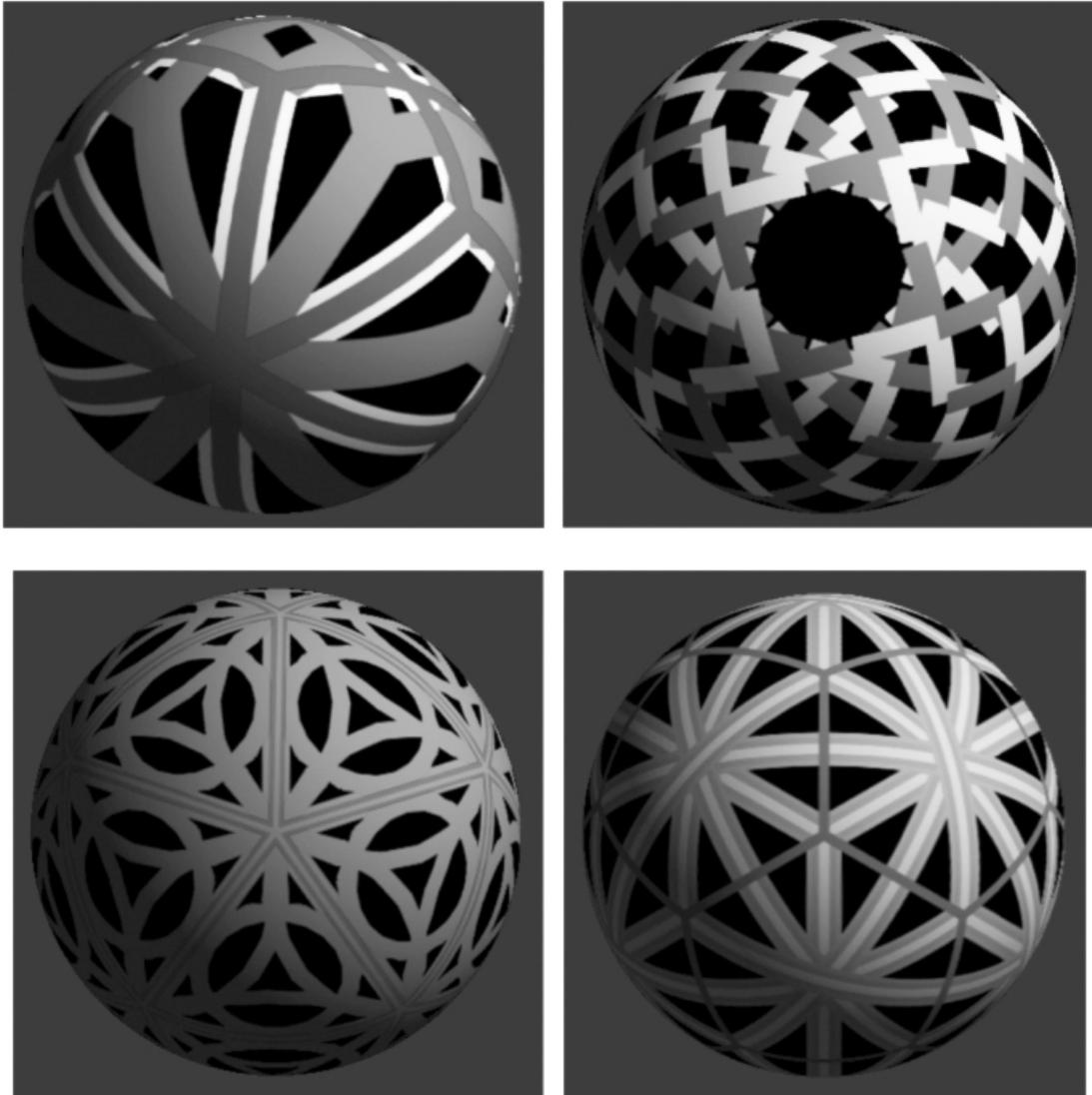


Figure 1. *Temari patterns created by the MoS system. Clockwise from upper left: patterns based on a cubical, octahedral, dodecahedral, and icosahedral "foundation design".*

In the remainder of the paper, we will go into greater detail about how these patterns were created (and how similar patterns might be created by the reader). The following (second) section is a brief introduction to "spherical turtle graphics" as it has been implemented in the MoS system. The third section steps through the central techniques behind the creation of the patterns shown in Figure 1. In the fourth and final section, we briefly outline related work in spherical design, and discuss ongoing and potential future work related to the MoS system itself.

Turtle Graphics on the Sphere

The screen interface of the MoS system is shown in Figure 2. There are several windows visible to the user: the window at the upper left is an editor in which spherical turtle graphics programs may be written, the window at lower left allows the user to type in (and execute) single turtle commands, while the window at right is a depiction of the sphere and turtle (shown as a triangular cursor).



Figure 2. *The interface to the MoS programming system (available through the website www.mathsphere.org). At the upper left, a program editor window; at lower left, an interactive command interpreter; and at right, a representation of the sphere on which patterns are drawn. This sphere may be turned interactively via the mouse so that the user can see any portion of the entire surface.*

The basic idea behind the MoS programming paradigm is that one can treat the turtle as a programmable "marker", rather like a movable pen; it can be given commands to move forward and backward, and turned in different directions, on the spherical surface. As the turtle moves, it may additionally draw a line on the surface shown in Figure 2. The MoS spherical turtle is thus a variant of the standard (planar) Logo turtle, in which commands such as `forward` and `right` are used to draw geometric patterns on the computer screen. The MoS turtle makes use of ideas introduced in Abelson and diSessa's book *Turtle Geometry* [1], which explores the ways in which spherical turtle programming differs from the usual planar variety.

Here, we give only a brief introduction to the key ideas of MoS spherical turtle geometry; more elementary detail on the MoS language and system is provided in [4], and a fuller mathematical discussion of the spherical turtle is given in Abelson and diSessa's book. (And of course, the reader is encouraged to go to the website and experiment on his or her own!) The major elements of the language are as follows:

- The turtle is assumed to have a position and heading at the start of any given program. Typically, the MoS turtle begins at a point on the equator of the sphere, with a heading due north (pointing toward the pole).
- A `forward` (or `fd`) command moves the turtle a given number of steps along a great circle in the direction specified by its heading. We use the convention that `forward 360` will move the turtle through a complete great circle from any starting position; thus, by executing the command `forward 360`, the turtle will draw a great circle and finish in its very same starting position.
- A `right` (or `rt`) command turns the turtle by a given number of degrees. Thus, for instance, a `right 90` command will cause the turtle to execute a "right face" in its current

position; a `right 360` command turns the turtle through 360 degrees (leaving the original heading unaltered).

- A `penup (pu)` command changes the state of the turtle ("picking up the pen") so that it will not draw lines as it moves; `pendown (pd)` changes the state of the turtle so that it will draw.
- There is additional syntax for iterating sequences of commands (`repeat`); for conditional expressions; and for defining functions.
- Finally, there are commands for changing the width and color of drawn lines.

For those readers experienced in standard Logo turtle graphics, all these elements will be familiar; indeed, the only major change from the planar Logo turtle is in the meaning of the forward command. Nonetheless, the effect of moving the turtle in great circle "straight lines" atop a spherical surface leads to novel ideas for those whose prior experience has been restricted to the plane. For example, a typical introductory spherical turtle program ([1, 4]) can be executed by the single command:

```
repeat 3 {forward 90 right 90}
```

If this command is executed with the turtle in its standard starting position, then the turtle first draws a longitude line to the North Pole; it then turns 90 degrees to the right and draws a second longitude line back to the equator; and finally, it turns 90 degrees to the right and moves back to its starting position along the equator. The overall result is an equilateral triangle with three right angles (an impossibility on the plane), as shown in Figure 3.



Figure 3. *An equilateral right triangle drawn by the turtle on the spherical surface.*

Turtle-Drawn Temari Patterns: a Sampler of Techniques

Having introduced the basic building blocks of the MoS turtle system, we now take a larger (and we hope, not too ambitious) step toward the creation of temari-like designs. The four designs that we describe here are based on structural patterns derived from four of the five Platonic solids (the cube, octahedron, icosahedron, and dodecahedron—only the tetrahedron is not included here). That is, in each case we begin by creating a turtle program that draws the projection of an inscribed Platonic solid on the sphere; then we use that program as the foundation for a decorative pattern.

Cubical temari. Our first example is based on a turtle program that draws the projection of an inscribed cube on the spherical surface. To make a single ("spherical square") face of the inscribed cube, we use the following expression:

```
repeat 4 {forward 70.5 right 60}
```

This expression might seem a bit unexpected to the "planar Logo" aficionado. In standard turtle geometry, a square is made with four `right 90` turns; here we are using four `right 60` turns. Space limitations preclude a thorough discussion of the mathematical issues here (again, see [1, 5] for more detail), but the essence of the issue is that the interior angles of a "spherical square" depend, interestingly, on the area of the square. What we are seeking, then, is the particular size of a forward move that will permit us to create a square with interior angles of 120 degrees; and we can find (through either linear algebra or patient numerical experimentation in the MoS system) that the desired size of the square edge is 70.5. In Figure 4, we illustrate the idea by first showing a single square; then three of the squares surrounding a single point; and then six of the squares arranged about the sphere to make an inscribed cube.

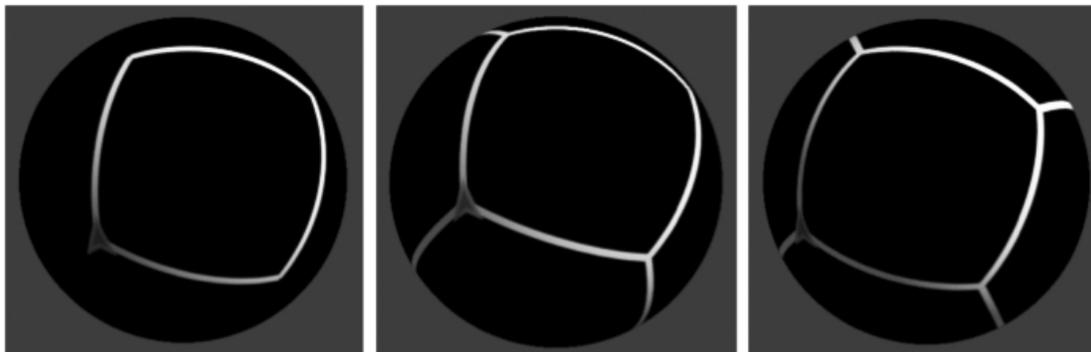


Figure 4. *Left: a spherical square with interior angles of 120 degrees. Center: three squares surround a point (unlike the plane, where four are needed!). Right: the projection of the complete inscribed cube.*



Figure 5. *Left: two inscribed cubes, offset by 60 degrees. Center: Six pair of broken red lines are added (a similar set of six pairs is drawn opposite on the sphere). Right: the original cube structures are redrawn in blue.*

To make the pattern shown at the upper left of Figure 1, we use this spherical cube pattern as the "core" of our design. The first step is to draw two cubes, starting from the same turtle position, but offset by 30 degrees (at left in Figure 5). We then draw, from the starting point, two sets of thick red lines; each set is composed of six "bent" lines offset by 60 degrees. A similar set of red lines is drawn at the point directly opposite the starting point on the sphere; and the resulting intermediate design is seen at the center of Figure 5. Finally, we re-draw the original pair of cubes (i.e., the same pair shown at the left of Figure 5), but now in a thinner blue line. The final design is seen at the right of Figure 5; this is the same design as in the upper left of Figure 1, but (in Figure 5) we have left the turtle visible to emphasize the progression of steps used to create the pattern.

Octahedral temari. The pattern at the upper right in Figure 1 is created from an octahedral "core", shown at the left in Figure 6 (and composed of eight "right equilateral" triangles of the sort that we created in Figure 3 earlier). The key idea behind our example is that, as the turtle draws each line of the octahedral core, it includes extra "slashes" drawn in thick yellow and red. The central portion of Figure 6 shows the result for a standard octahedron; we then repeat that same pattern, six times from around the starting point with a gap of 60 degrees between each separate pattern. The result is shown at the right of Figure 6; our pattern in Figure 1 is identical, except in the earlier drawing we have chosen not to include the initial octahedral core lines.



Figure 6. *Left: Projection of an inscribed octahedron on the sphere. Center: additional red and yellow "slashes" added to every drawn line. Right: six copies of the pattern shown at the center, rotated around a central point with gaps of 60 degrees.*

Icosahedral temari. The projection of the inscribed icosahedron onto the sphere has twenty equilateral triangles—but unlike those of the octahedron, we need to adjust the size of the triangles so that five of them surround a point (unlike four, as in the octahedral case). Thus, we need triangles whose interior angles are 72 degrees; and the appropriate side length of the triangle is 63.5.

The decorative idea for the icosahedron makes use of yet another technique, for drawing "spherical circles"—sets of points equidistant from a chosen center. As with so much else, the planar technique for drawing a Logo circle (a typical strategy is to repeat, 360 times, a short move followed by a 1 degree turn) will not work on the sphere since it does not take into account the curvature of the surface on which the turtle moves. Our strategy (see also [4]) is to begin with the turtle at the center of the desired circle; then repeatedly move the turtle to the periphery of the circle, draw a short approximate chunk of the desired circle, and move back to the center. By repeating this process approximately 20 times, the result is (to the eye) a reasonable spherical circle.

Figure 7 shows progressive stages of the icosahedral temari pattern. At left, we have a single icosahedral triangle, along with two circles drawn from the center of the triangle: one "large" (with a radius enough to reach the triangle vertices) and one "small" (with a radius just long enough to reach the edges). At the center, we see five of these triangles surrounding a single point. The full pattern is created by repeating this pattern for the entire icosahedron, and then redrawing the original icosahedron core with two pen widths (first thick, then thin) in two distinct colors.



Figure 7. *Left: an icosahedral triangle, drawn with two red circles. Center: A fivefold copy of the pattern at left. Right: The pattern at center extended to the entire sphere; the icosahedral core is now redrawn.*

Dodecahedral temari. The dodecahedral temari, like the icosahedral example, explicitly shows the core structure of the underlying Platonic solid. In the case of the dodecahedron, each face is a regular spherical pentagon with a side length of 41.81 and interior angles of 120 degrees, as seen at the left of Figure 8. As a decorative strategy, we begin with this pentagon and, at every vertex, we have the turtle draw three thick lines (using steadily diminishing pen widths, so that each color appears on top of the previous one) from that vertex toward the center of the opposite edge (center of Figure 8). By repeating this motif, we finish with the complete temari pattern (shown at the right of Figure 8; this is the same as the design shown in Figure 1, but here we have left the turtle visible).



Figure 8. *Left: a dodecahedral pentagon, drawn in blue. Center: Interior lines added to the pentagon. Right: the pattern at the center repeated over the entire sphere.*

Continuing, Future, and Related Work

Besides the craft literature on temari, there are several interesting recent works illustrating beautiful symmetric patterns based on spherical geometry [2, 7, 8]. The purpose of the MoS system is to provide a freely available medium that allows students and artists to experiment with such patterns via programming. By playing with a medium of this sort, users may also gain a working knowledge—an intuitive grasp—of spherical (and more broadly, non-Euclidean) geometry. Though it has not been the focus of this paper, the examples illustrated here introduce notions such as geodesics (great circles on the sphere), intrinsic curvature, and the dependence of many staples of high school geometry (such as the rule that triangles have interior angles that total 180 degrees) upon the assumptions of Euclidean space.

The MoS language system, while powerful, is still a work in progress: most importantly, we are working on expanding the language to accommodate compound data structures, which will in turn enable a wide variety of decorative programming strategies beyond those shown in this paper. There are still other somewhat longer-term planned additions to the language (a "region-fill" function, multiple interacting turtles) that will likewise extend both the mathematical and artistic functionality of the MoS system.

One important element of the current language is that it is designed to be compatible with the giant "Science on a Sphere" spherical display surface available at numerous museums and planetariums [http://sos.noaa.gov/What_is_SOS/index.html]. That is, once the user has created a working spherical turtle program, that program can potentially be run on an actual sphere (not merely on a screen representation of a sphere); we have in fact conducted workshops for children at the Lawrence Hall of Science in Berkeley in which the youngsters do just that. This ability in turn suggests that temari artistry need not be limited to embroidery as a medium of expression, but could be enacted (and expanded into realms such as animation) via projection on a spherical surface. As of this writing, we are conducting a somewhat more extended (four-part) set of MoS lessons for middle school students in Boulder; the fourth and final session is planned to take place at NOAA headquarters so that the students can project their programmed creations on the spherical screen there.

All this is not to say, of course, that we have any objection to the classical craft of temari! (Quite the opposite, in fact.) Our own view is that a system like MoS can complement traditional temari; conceivably, for example, a temari crafter might experiment graphically with varying decorative motifs and color patterns as a preliminary step before constructing a woven temari ball. A bit (but only a bit) more futuristically, we could imagine combining MoS programming with a 3D printing component so that students or hobbyists could print (and perhaps paint) physical models of their designs. In spherical art, as in so much else, traditional crafts and computational technology can serve to inform and strengthen each other.

Acknowledgments. The MoS system was implemented in our lab with the invaluable contributions of Michelle B. Redick and Michael MacFerrin. Many thanks for their assistance to Michael Biere, Hilary Peddicord and their colleagues at NOAA. The work described in this paper was supported in part by a grant from the National Science Foundation, DRL1114388.

References

- [1] Abelson, H. and diSessa, A. [1980] *Turtle Geometry*. Cambridge, MA: MIT Press.
- [2] Conway, J.; Burgiel, H.; and Goodman-Strauss, C. [2008] *The Symmetries of Things*. Wellesley: A.K. Peters.
- [3] Diamond, A. [1999] *The Temari Book*. Asheville, NC: Lark Books.
- [4] Eisenberg, M. [2012] "Computational Diversions: The Return of the Spherical Turtle. In *Technology, Knowledge, and Learning*, 17:3, pp. 115-122.
- [5] Eisenberg, M. [2010] "Computational Diversions: Turtle Really and Truly Escapes the Plane". *International Journal of Computers in Mathematical Learning*, 15: 73-79.
- [6] Hsi, S. and Eisenberg, M. [2012] "Math on a Sphere: Using Public Displays to Support Children's Creativity and Computational Thinking on 3D Surfaces." In *Proceedings of Interaction Design and Children (IDC 2012)*, 248-251.
- [7] Popko, E. [2012] *Divided Spheres*. Boca Raton, FL: CRC Press.
- [8] Van Brummelen, G. [2013] *Heavenly Mathematics*. Princeton, NJ: Princeton University Press.
- [9] Yackel, C. [2011] "Teaching Temari: Geometrically Embroidered Spheres in the Classroom". *Bridges 2011*, pp. 563-566.