Turtle Geometry on a Sphere: a Projected Future for Constructionism

Michael MacFerrin, Brian Hallesy, Garrett Shulman, James Bailey, Neal Robbins, Brandon Shelton, and Michael Eisenberg, *duck@cs.colorado.edu*
Dept. of Computer Science, University of Colorado, Boulder CO USA

Abstract
Historically, the educational philosophy of constructionism has been founded upon the idea that children learn by designing and creating *public, shared* artifacts. Typically, the techniques for making artifacts "public" or "sharable" have involved (e.g.) sending files via email or posting creations (such as graphics or music) on social networks or public websites. Recently, however, another important technique has emerged for sharing certain types of graphical projects– namely, by making use of innovative projection systems to display those projects on large, widely visible, and often unorthodox surfaces. This paper describes a project in which an interactive Logo-style turtle has been implemented on the giant "Science on a Sphere" projected display installed at our University planetarium. Science on a Sphere is a display technology available at numerous museums around the world, and is typically employed to present pre-made animations (e.g., of global weather patterns); but by allowing children to write graphical programs for display on the sphere, programming projects take on an aspect of public performance that is largely absent on smaller computer screens. Moreover, the fascinating mathematical ideas underlying spherical geometry can be naturally represented and explored on this surface. This paper discusses the current implementation of our *Turtle on a Sphere* system and presents a variety of geometric examples making use of the system. We conclude with a somewhat wider-ranging discussion of the potential role of novel projection technologies in constructionist education.

Figure 1. A spherical turtle-drawn flower.

Keywords
Spherical geometry, turtle graphics, projection technology, constructionism.

Introduction: Toward a New Era of Display

In what is likely the most commonly cited description of "constructionism", Papert [9] writes:

> Constructionism–the N word as opposed to the V word–shares constructivism's connotation of learning as "building knowledge structures" irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe. [9, p. 1]

As Papert notes, *public* or *shared* presentation of one's creations is arguably a central element of constructionism's philosophy of learning by design. It enables learning communities of students to observe and build off one another's work; it gives students the opportunity both for hearing critiques of their own work and critiquing the work of others; and it endows student creations with important elements of "social capital", in that creations may be traded, given as gifts, combined into larger collaborative entities, and so forth. (Cf. also the excellent discussion of these themes in the introductory section of Noss and Hoyles [8].)

Figure 1. A spherical turtle-drawn "flower" pattern.

In our own Craft Technology Laboratory at the University of Colorado, this element of public creation is often reflected in projects centered upon physical artifacts (such as polyhedral paper sculptures, popup cards, or programmable clothing, to name just a few [2]). Tangible creations of this sort lend themselves to certain highly natural–often, almost unconscious–types of sharing, performance, and presentation. In contrast, screen-based artifacts–of the sort typical of "traditional" Logo turtle graphics–tend, historically, to be viewed on small, individual computers. As such a traditional turtle project is viewed by at most a few people at a time, and "sharing" tends to take place in more indirect ways. A student might (e.g.) send a file to a friend via email, or a group of students might, over time, observe a running program over the Web; but in general, running a computer program is not seen as an act of *performance*, done for a sizable audience. The public, shared aspect of such projects is thus necessarily muted.

This paper argues that novel technologies for projection enable us to rethink the element of performance in children's programming. As an example, we describe an innovative software project, *Turtle on a Sphere*, in which turtle graphics can be displayed interactively on a giant museum-based spherical display. Turtle on a Sphere illustrates the ways in which recent developments in projection technology can vastly expand the notion of "sharing" and "presenting" computational artifacts. At the same time–and not entirely coincidentally–Turtle on a

Sphere is a natural means for the introduction of important ideas in non-Euclidean (in this case, spherical) geometry. Turtle on a Sphere projects–such as the multicolored flower shown in Figure 1–are both highly visible public decorations, and encounters with challenging mathematics.

The remainder of this paper focuses on the Turtle on a Sphere project as a single instance of a larger technological development in children's programming. In the following (second) section, we discuss the current implementation of the project and its combination of programming language and graphical user interface. The third section presents several sample Turtle on a Sphere projects, and explains the basic concepts of spherical geometry that underlie those projects. In the fourth and final section, we use the Turtle on a Sphere system as a springboard for a more wide-ranging discussion of related and potential future projects integrating novel projection technologies with children's programming and construction.

Turtle on a Sphere: a Brief Description of the System

The Turtle on a Sphere system is, essentially, an implementation of a "classic" Logo-style turtle intended for use with the "Science on a Sphere" (SoS) display system. Before turning to the implementation of our software, it is worth spending some time on the SoS system itself. The display was developed at the National Oceanic and Atmospheric Administration (NOAA) by Dr. Alexander MacDonald as an educational device for viewing computer-generated graphics on a sphere. As the NOAA website [7] explains:

> Science On a Sphere® is a large visualization system that uses computers and video projectors to display animated data onto the outside of a sphere. Said another way, SOS is an animated globe that can show dynamic, animated images of the atmosphere, oceans, and land of a planet. NOAA primarily uses SOS as an education and outreach tool to describe the environmental processes of Earth. [7]

Many museums and institutions around the globe now include the SoS display–including (among others) the Lawrence Hall of Science in Berkeley, the Bishop Museum in Honolulu, the Minnesota Museum of Science in St. Paul, and the Centre National d'Etudes Spatiales (CNES) in Strasbourg. (See [6] for a more extensive list; our project was conducted with the display at our University's Fiske Planetarium in Boulder, Colorado.) The scale of the display–six feet in diameter–and the high resolution of its graphics are both remarkable, as can be seen in the photograph in Figure 1. In practice, the display is implemented with a set of four video projectors positioned around the spherical surface; these projectors are controlled via a software system that translates portions of the spherical display into individual commands for each projector. The net effect, then, is for the four individual projections to combine seamlessly into a single unified spherical animation. In principle, the same sort of technique could be used to project animations onto surfaces such as a cube, cylinder, or cone; we will return to this idea toward the end of the paper.

By searching for photographs or videos on the Web, one can see an extensive variety of Science on a Sphere projects; generally, these projects involve the presentation of "pre-canned" animations or graphics–e.g., an animated weather map, or a display of the surface of the moon. Such projects are unquestionably both gorgeous and educationally worthwhile, but in these cases the students' role is simply to watch and admire the work of (usually unseen) professionals. In contrast, the intent of our Turtle on a Sphere project is to give youngsters a chance to write interactive programs for the sphere, creating aesthetically appealing designs, sharing those designs with viewers, and experimenting directly with the mathematical ideas of spherical geometry.

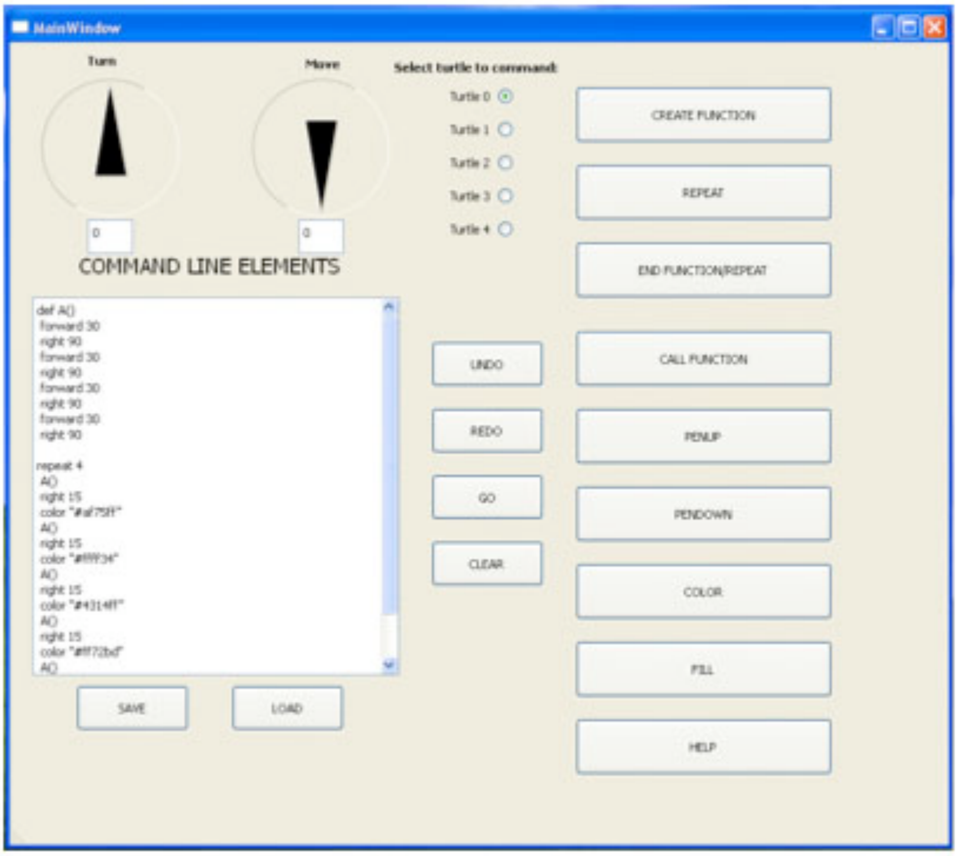

Figure 2. The graphical user interface for the Turtle on a Sphere system. Toward the left, the text window allows for the direct creation of a spherical turtle-walk program. The set of push-buttons at right can be used to insert textual elements into the program. The dials at the top are used to create "forward" and "right" commands directly, by hand. Other elements of the interface are described in the accompanying text. This figure shows a program in the process of being created; the current program is visible in the text window.

The Turtle on a Sphere system is a software-based "front end" to the Science on a Sphere display that allows students to create and observe turtle walks on the giant surface. We will discuss the spherical geometry behind these programs in the following section, but for now it is simply worth noting that the Logo "forward" command, on the sphere, causes the turtle to walk in the arc of a great circle, and that the default measure of the circumference of the sphere is 360. Thus, telling the turtle to move "forward 360" from any starting position and heading whatever on the sphere will cause the turtle to move all the way around the sphere and to finish in the same state that it started. Other turtle commands (right, penup, pendown, and color-changing commands) work in much the same way that they do on the plane.

Figure 2 shows the graphical user interface for the Turtle on a Sphere system, and also provides an overview of the basic functionality of the system. The major elements of the interface consist of a program text window (toward the left), in which turtle commands can be entered either via direct textual insertion, or interactively via selection of buttons and sliders. The set of buttons toward the right of the interface allow the user to input often-used language commands into the text window: the user can create a new function, start a "repeat" command, pick up or put down the turtle's pen, change the color, and so forth. The dials at the top of the screen allow the user to select a parameter for the turtle's forward and right commands, and to insert the appropriate command into the text window.

There are still other features of the Turtle on a Sphere interface worth noting here. The four smaller rectangular buttons toward the center of the screen allow the user to undo or redo an editing command, to run a program on the sphere (the "Go" button), or to reset the sphere to a blank state and clear the program window (the "Clear" button). The five radio buttons toward the upper center of the interface allow the user to associate a given program with any one of five distinct spherical turtles; thus, by selecting a given turtle and then pressing "Go", the same program may be run for distinct turtles in succession. Finally, the "Save" and "Load" buttons toward the bottom left allow the user to save and reload files in the text window.

In sum, then, the graphical interface contains the essential elements for a wide variety of programming projects using the SoS display. The Figure 2 interface runs on a tablet computer; but our current system also has a "pure" language interface through which more complex programs may be entered via keyboard.

Explorations in Spherical Turtle Geometry: a Sampler

The previous section introduced the basic interface for the Turtle on a Sphere system. In this section, we show how the system may be used to explore important ideas in non-Euclidean geometry. (A good mathematical introduction to these concepts may be found in Abelson and diSessa's book *Turtle Geometry* [1, Chapter 5].)

As an initial experiment in spherical turtle geometry, we can make use of several intersecting great circles to create the pattern shown in Figure 3.

Figure 3. A pattern created by three great circles.

The program that generates this pattern can be written directly:

```
forward 360     ; the turtle moves all the way around the globe,
right 90        ; turns right,
forward 360     ; moves all the way around the globe again,
forward 90      ; goes forward 1/4 around the globe,
right 90        ; turns right,
forward 360     ; and makes the third and final great circle
```

Several points are worth noting about the simple pattern of Figure 3. First, we note that the pattern divides the globe into eight identical spherical triangles, one of which is prominent toward the front of the display in the photograph. Each of these triangles is an equilateral triangle (each side is a quarter of a full great circle arc), and each triangle contains three right angles. In the plane, a triangle with three right angles is an impossibility; but on the sphere, the interior angle sum of a triangle depends on its enclosed area. Thus, a tiny equilateral triangle on the sphere has interior angles just a little above the planar value of 60 degrees, for a total just greater than 180; but the much larger triangles formed by the Figure 3 pattern each cover one-eighth of the globe, and consequently (see the discussions in [1, 4]) each of these triangles has interior angles that total 270 degrees.

The pattern in Figure 3 is in fact the projection onto the sphere of an inscribed octahedron (the second shape in the top row of Figure 4). To put it another way: suppose we were to place a large six-foot-diameter octahedron inside the giant sphere so that each of its six vertices just touch the interior of the sphere. A light bulb at the very center of the sphere would then project the octahedron's edges onto the surface in the same pattern shown in the photograph. This observation leads us to try creating patterns on the sphere resulting from the projections of still other inscribed shapes. Figure 4 shows the five so-called "Platonic", or regular solids, whose vertices are all surrounded by the same number of identical regular polygons; and Figure 5 shows the projections of the remaining four Platonic solids (the tetrahedron, cube, icosahedron, and dodecahedron) onto the sphere.

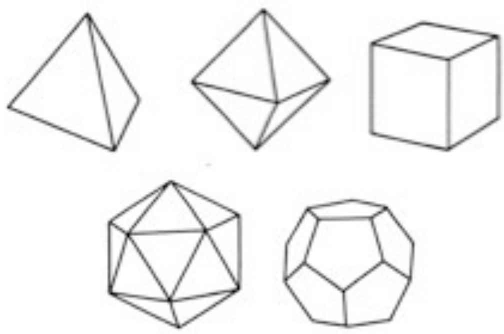

Figure 4. The five Platonic solids. Top row: tetrahedron (composed of four equilateral triangles), octahedron (eight equilateral triangles), cube (six squares). Bottom row: icosahedron (twenty equilateral triangles) and dodecahedron (twelve regular pentagons).

A little reflection on the patterns in Figure 5 will reveal still other interesting deviations from planar (Euclidean) geometry. For example, both the tetrahedral pattern at upper left and icosahedral pattern at lower left contain equilateral triangles (four and twenty, respectively). The large triangles of the tetrahedral projection each have three interior angles of 120 degrees each–that is, they each have an angle sum of 360 degrees (instead of the planar 180). The icosahedral triangles, which are much smaller in area, each have three interior angles of 72 degrees each, for an interior angle sum of 216; inspection of the figure shows that these smaller triangles look a good deal more "planar", or perhaps less "spherical", than those of the octahedron and tetrahedron. For the sake of completeness, it is also perhaps worth noting that the interior angles of the "spherical squares" in the cube are 120 degrees (instead of the planar 90); and those of the pentagons in the dodecahedron are also 120 degrees (instead of the planar 108).

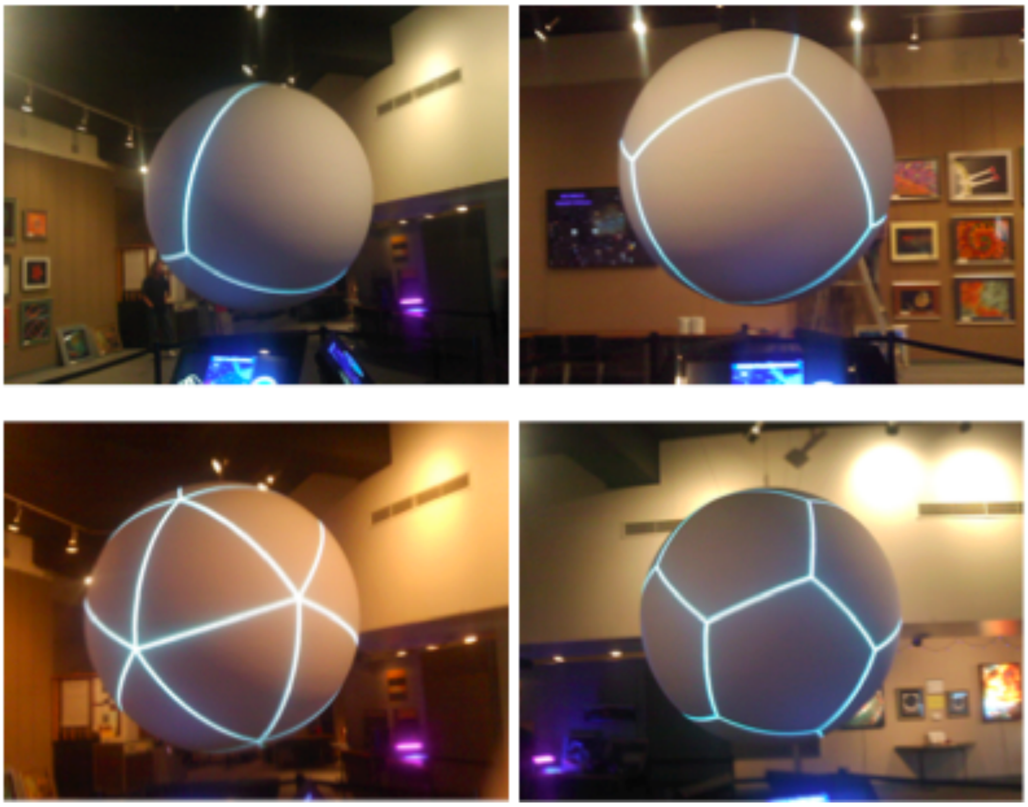

Figure 5. Projections, onto the sphere, of inscribed Platonic solids. Top row: the tetrahedron and cube. Bottom row: the icosahedron and dodecahedron. (The projection of the octahedron is shown in Figure 3 earlier.)

The examples of Figures 3 and 5 are intended to highlight the distinctions between planar and spherical geometry; but it is fun to try other examples of "standard" Logo programs on the sphere to see what happens. The classic turtle-drawn "flower" of Figure 1, consisting of twelve "petal-like" shapes, was drawn in outline by the following program:

```
to petal (size, moves, smallturn, bigturn)
  repeat 2
    repeat moves
      forward size
      right smallturn
    right bigturn

repeat 12
  petal (5, 10, 6, 116)
  right 30
```

A reader familiar with planar Logo patterns will note that the angles used for the repeated "petal" shapes would not result in closed turtle-walks on the plane; but they do on the sphere. The Figure 1 flower also, parenthetically, illustrates the ability of the Turtle on a Sphere program to include "fill" commands (for brevity, the code for this portion of the flower program is not shown).

Figure 6 shows two other famous turtle-drawn programs, realized in their spherical versions. At left, the "dragon" curve (as described in [1], p. 93) is shown; on the sphere, the standard turtle-turn for the dragon curve has been changed from its usual value of 90 degrees to a "softer" value of 86.5, to produce a more aesthetically appealing version of the curve. At right in Figure 6, the "C-curve" (see [1], p. 92) has been drawn. The program for drawing this shape is as follows:

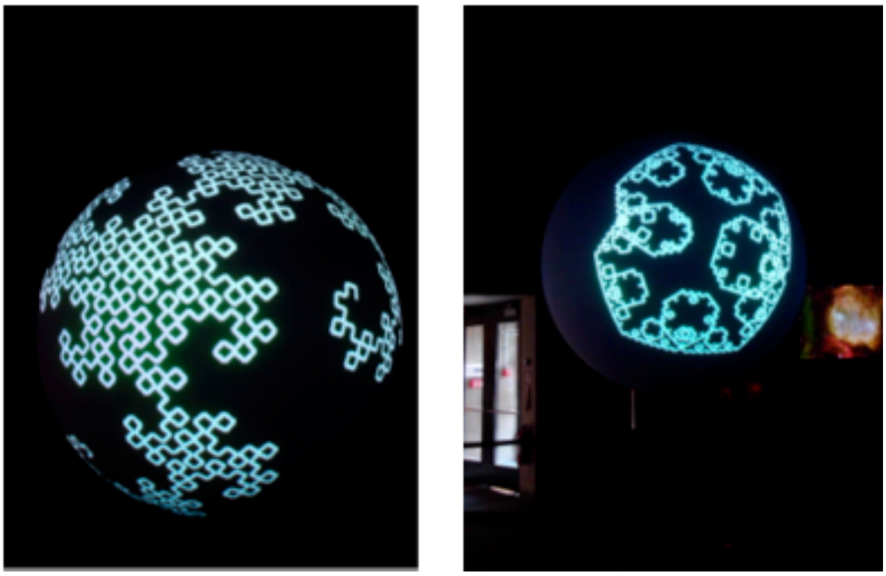

Figure 6. Two recursive patterns drawn on the sphere. At left, a dragon curve; at right, a C-curve.

```
to ccurve (side, level)
   if level == 0:
      forward size
   else:
      ccurve (side, level - 1)
      right 90
      ccurve (side, level-1)
      left 90

ccurve (2, 10) ; create a 10th-level c-curve with a side of 2
```

In the plane, the C-curve would not "close in" on itself, but instead looks like a rather elaborate "C" shape; on the sphere, as seen toward the right of Figure 6, this large C-curve makes a (highly attractive) closed shape.

Projection and Display Technology, Unorthodox Turtle Geometries, and Programming as Performance: New Directions for Constructionism

The "Turtle on a Sphere" system is still in a relatively early stage of development, and has only been tested very informally with local students. Our goal is to use this as a springboard for future work, culminating in a system that can be appropriated and extended by interested parties–

researchers, teachers, hobbyists, and students. Before its public release, however, any system along these lines would need additional work: extensions to the language (such as analogues to the Logo "setpos" and "getpos" commands, among others) and refinements of the current interface.

The larger subject of this paper, though, goes beyond one particular software system. Turtle on a Sphere is a single instance of what could well be a much more pervasive technological idea– namely, the use of customized surfaces and projectors to produce "screens" of many shapes, sizes, and environmental settings. While the implementation of NOAA's Science on a Sphere display is highly sophisticated, the basic idea–using multiple projectors to create a unified picture–is potentially applicable to a variety of other surfaces. One might imagine using similar techniques to create graphics on (e.g.) a cylindrical surface; a conical surface; a pseudosphere model (to explore ideas in hyperbolic geometry; cf. [10] for a general discussion of computational approaches to this subject); or even the interior (as opposed to exterior) of a hemisphere. Each new surface for display does, admittedly, require significant effort: an underlying software translator must be developed to control the output of multiple finely-situated projectors. Nonetheless, once such a driver has been implemented for a particular surface (say, a cylinder), it can be ported to a wide number of sites, just as the Science on a Sphere display has now been installed in numerous public settings.

We would argue, further, that there is a strong potential for accessible home or classroom implementation of such displays in the future. A "do-it-yourself" Science on the Sphere could plausibly be implemented using a smaller sphere (of perhaps 1 foot in diameter and composed, say, of white acrylic) and a collection of four affordable nanoprojectors of the type currently emerging in the commercial market. Going just a bit further, it should be possible to employ affordable 3D printers to create a variety of customized geometric surfaces (cones, cylinders, etc.) in white plastic; and these surfaces could be the basis of a wide range of classroom or home displays. In other words: the hardware components behind the Science on the Sphere display could be re-implemented in smaller form, with cheaper materials; while the sophisticated software driving the display could be downloaded to homes and classrooms, just as it is now in museum settings. The result, within a decade or so, could well be a pervasive presence of beautiful new display surfaces, displaying homemade or student-created graphical effects tailored for particular geometries.

Such systems are themselves part of a still larger, growing ecology of novel display technologies–including numerous competing technologies for three-dimensional and volumetric displays. [3, 5] In effect, by considering the roles of these emerging technologies for constructionist education, we can re-imagine children's programs as *performances* in the making–as things to run in numerous public settings and in specialized or customized environments. Programs could run on surfaces arranged in playgrounds, theme parks, classrooms, or private homes; graphical effects could combine elements of projected displays and three-dimensional screens; small personalized projectors could be used to superpose graphics on top of still other displays (for example, children might use their own personalized projectors to produce handheld animated effects combined with those produced by the SoS display).

Constructionism, as an educational philosophy, involves a close attention to the intellectual role of design and personalized creation; but at the same time, as we noted at the outset, children's artifacts attain particular educational potency when they are embedded within social practices– for instance, when they can be presented in public, shared with an audience, and linked to meaningful physical settings and environments. Novel display technologies–of which Science on a Sphere is merely an early example–offer an avenue through which to rethink these aspects of constructionist education. Such displays offer cognitive and intellectual challenges–"powerful ideas" (in Papert's phrase) of three-dimensional and non-Euclidean geometry and visualization. At the same time–and equally important–they offer beautiful and motivating opportunities for children to realize, share, and express their ideas.

Acknowledgments

This research was made possible through a grant from the University of Colorado Outreach Office, and is funded in part by NSF grant IIS0856003. Thanks are due to Francisco (Tito) Salas, Jim Love, and Mike Levy of the Fiske Planetarium in Boulder, Colorado, whose help and collaboration made it possible to develop the spherical-turtle language and interface. This project began as a result of wonderful conversations with Uri Wilensky; thanks also to Ann Eisenberg, Gerhard Fischer, Clayton Lewis, and Antranig Basman for their contributions and discussions.

References

[1] Abelson, H. and diSessa, A. [1980] *Turtle Geometry.* Cambridge, MA: MIT Press.

[2] Craft Technology Lab website: www.cs.colorado.edu/~ctg

[3] Dodgson, N. [2005] Autostereoscopic 3D displays. *IEEE Computer*, 38:8, pp. 31-36.

[4] Eisenberg, M. [2010] Computational Diversions: Turtle Really and Truly Escapes the Plane. *International Journal of Computers for Mathematical Learning*, 15:73-79.

[5] Favalora, G. [2005] Volumetric 3D Displays and Application Infrastructure. In *IEEE Computer*, 38:8, pp. 37-44.

[6] NASA website (list of "Science on a Sphere" locations): www.nasa.gov/centers/goddard/multimedia/frozen/soas_locations.html

[7] NOAA website for "Science on a Sphere": http://sos.noaa.gov/index.html

[8] Noss, R. and Hoyles, C. [2006] Exploring Mathematics through Construction and Collaboration. In Sawyer, K. (ed.) *Cambridge Handbook of the Learning Sciences*. New York: Cambridge University Press, pp. 389-405.

[9] Papert, S. [1991] Situating Constructionism. In Harel, I. and Papert, S. (eds.) *Constructionism*. Norwood, NJ: Ablex.

[10] Stevenson, I. [2000] Modelling Hyperbolic Space: Designing a Computational Context for Learning Non-Euclidean Geometry. *International Journal of Computers for Mathematical Learning*, 5:2, 143-167.