Planar Symmetry with Turtles

James Dean Palmer
Department of Computer Science
Northern Arizona University
James.Palmer@nau.edu

Abstract

In this paper we describe a pedagogical exercise developed for an introductory computer graphics class which introduces isometry and congruence transformations as a foundation for exploring conformal and affine transformations.

Isometry is a distance-preserving isomorphism between metric spaces. Isometries of the Euclidean plane, E_2, fall into four categories: translation, rotation, reflection and glide-reflection. These operations are called congruence transformations because after applying these operations to the lines or points that make up an object, the size and shape of the object (but not the position or orientation) is preserved.

In the early twentieth century Russian crystallographer E. S. Fedorov enumerated seventeen distinct two dimensional “wallpaper groups” which describe topologically discrete groups of isometries of the plane containing two linearly independent translations.

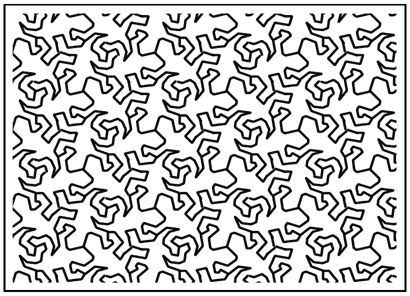

Figure 1: A student’s interpretation of Escher’s *Study of Regular Division of the Plane with Reptiles*.

The wallpaper groups not only provide a mechanism for classifying symmetries that have appeared in human authored designs for thousands of years but create a framework for exploring all the two dimensional symmetries of the plane. The formal study of 2D symmetry has influenced artists such as M.C. Escher, K. Malevich and V. Vasarely [4,5].

More recently, computational methods and techniques have become available to investigate these symmetry groups. One such symmetry exploration tool, Kali, was designed by N. Amenta at the University of Minnesota Geometry Center [3]. Kali allows the user to place lines and curves on an invisible symmetric canvas that can represent all 17 wallpaper groups, 7 Frieze groups and a collection of Rosettes. Akleman, et al. have considered an alternate drawing system that represents the drawing as a graph in an attempt to avoid gaps and lines that may overlap cell boundaries [2].

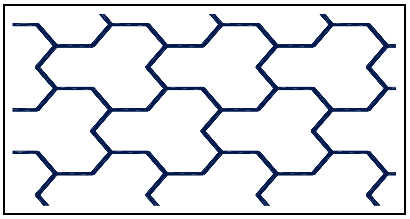

Figure 2: A student’s rendition of group p3lm.

While these applications do attempt to be educational as well as design tools, they don’t allow students to become involved in the actual transformations that make up the underlying symmetries. At the same time, these applications are quite sophisticated and complex. In this paper we suggest using the language Logo as a tool for teaching planar symmetry.

Logo belongs to the Lisp family of languages and was designed as a learning tool at MIT. One feature of the language was the ability to control a robotic creature that crawled across the floor in response to simple commands such as forward, back, left turn and right turn. The physical turtle eventually gave way to a virtual turtle that could trace paths on a computer display. One advantage of using the Logo language is that it has an extremely low learning curve and yet its Lisp underpinnings allow it a sophistication that often is not immediately appreciated. Logo has, for example, been used to explore decentralized systems, emergent phenomena, robotic systems and fractal geometry [1,6].

Logo has also been used to explore symmetry. Only a few lines of Logo are required to generate pentagons, decagons, stars and other shapes that exhibit obvious symmetries. A common task for grade school geometers using Logo is to generate such shapes and identify the inherent symmetries.

The task we posed for our undergraduate geometers, was to "extend" Logo such that the turtle will draw inside wallpaper group cells and the corresponding symmetries will fill the plane (or at least the clipped region of the plane that corresponds to the computer screen). The idea was inspired by Kali which allows students of symmetry access to all of the wallpaper groups and facilitates the construction of complex wallpaper patterns. But our intent is not that students "play" with the symmetry groups or necessarily learn how to build complex designs. Instead we would like students to focus on understanding the implementation mechanics and the underlying transformations. Rather than task the students with reimplementing Kali, we asked them to implement two wallpaper groups each in Logo.

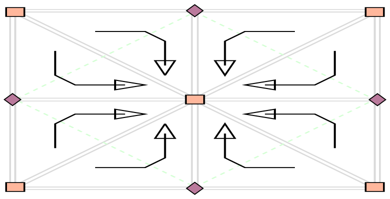

Figure 3: A collection of turtles with different origins responding differently to drawing commands.

Wallpaper groups may be visually described with cell structure diagrams that depict lines of reflection, glide reflection and centers of rotation of orders two, three, four and six (180, 120, 90 and 60 degrees respectively). Figure 3 is an example of such a diagram. Using these cellular descriptions students had to decide how to rotate and translate the turtle after each forward (FD), back (BK) or turn (LT, RT) command. A sample implementation of the p1 group was given. The sample code creates a new *symmetric forward* function, SFD. SFD is a wrapper that applies translation along two axes using an FD call after a pen up (PU) command. It repeats this translation over a user specified number of horizontal and vertical cells.

We were surprised by the diversity and creativity of approaches that students explored. While some students attempted to formulate mathematical equations which based on the turtles current global position finds correct positions for the corresponding symmetrical turtles, other students phrased the problem differently. One student, for example, converted the relative turtle movements into polar coordinates around the rotation points in his symmetry. Another student used a version of Logo that supports multiple turtles and assigned his turtles behaviors. For example, if the turtle implements a reflection, certain turtles will turn left when told to turn right. See Figure 3.

In the context of our computer graphics course, this assignment becomes the basis for the next assignment on conformal transformations where we use Logo to design several line based fractals (a fairly common Logo exercise) [1]. The final assignment in this transformation trilogy introduces affine transformations, where we manipulate turtle paths with affine transformation matrices. This approach for teaching transformations emphasizes the important properties of the transformation. In the case of congruence transformations one becomes intimately familiar with isometry. At the same time, the simplicity and elegance of Logo makes this material extremely accessible. We feel this was an extremely useful exercise for our students and that it could be adapted to fit in both math and computer science curricula where a working understanding of symmetry or congruence mappings are important. *I would like to extend my special thanks to my students, Brian Lipkowitz, Matt Shanker and Murphy McCauley whose work appears here.*

References

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

[2] E. Akleman, J. Chen, and B. Meric. Web-based intuitive and effective design of symmetric tiles. In *Proceedings of the 2000 ACM Workshops on Multimedia*, 2000.

[3] N. Amenta and M. Phillips. Kali, 1996. `http://www.geom.umn.edu/java/Kali/`.

[4] M. Dabrowski. *Contrasts of Form: Geometric Abstract Art 1910-1980*. Museum of Modern Art, 1982.

[5] J. L. Locher. *M. C. Escher: His Life and Complete Graphic Work*. Abrams, 1992.

[6] M. Resnick. *Turtles, Termites, and Traffic Jams*. MIT Press, 1994.