Monte Carlo Art Using Scratch

Patrick Honner
Brooklyn Technical High School
29 Ft. Greene Place
Brooklyn, NY, 11217, USA
E-mail: patrick.honner@gmail.com
http://www.MrHonner.com

Abstract

Elementary Monte Carlo simulations offer rich and accessible project opportunities for secondary students that lie at the intersection of mathematics, statistics, science, and computation. Here I outline a class of art projects that leverage the Monte Carlo simulation environment to allow students to explore computational and algorithmic art. These projects build on the key elements of Monte Carlo simulation—parameterization, randomness, approximation—while providing an authentic creative experience for students.

Introduction

In the process of developing a *Computational Modeling* course for advanced high school students, I began investigating and building Monte Carlo simulations in Scratch [1]. Scratch is a visual, drag-and-drop, web-based programming environment developed with elementary school students in mind, but it is gaining popularity in secondary and even higher education as an accessible and robust programming environment [2], [3]. Its sprite-driven approach is well-suited for basic area approximations using probabilistic methods.

While exploring the design of simple Monte Carlo projects in Scratch, I inadvertently produced some images I found aesthetically pleasing. For example, below is the result of a Scratch-based Monte Carlo approximation of the area under part of a polynomial curve.

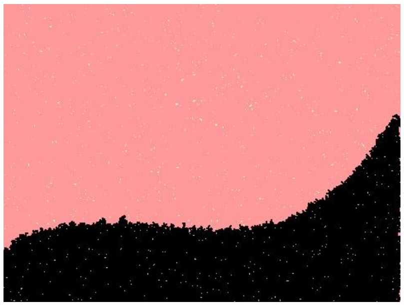

Figure 1: *Twilight on Mars*, $\left[y = \frac{1}{10}x^3 - \frac{1}{2}x^2 + \frac{2}{3}x - \frac{1}{4}; 0 \le x \le 4 \right]$

The stamping feature of sprites in Scratch, together with the various incidental parameter choices (size and shape of sprite, color, etc) contributed to the result image. As my experiences with the Bridges organization have inspired me to look for ways to integrate creative and artistic opportunities for students into their mathematical and computational projects, I immediately recognized the potential to bridge mathematical and artistic experiences for students.

Project Overview – Basic Approximations

The project begins with students performing elementary Monte Carlo area approximations. Students are given solid single-color images of a variety of shapes whose areas are known, and the students then build a simulation and produce approximations of area using probabilistic methods.

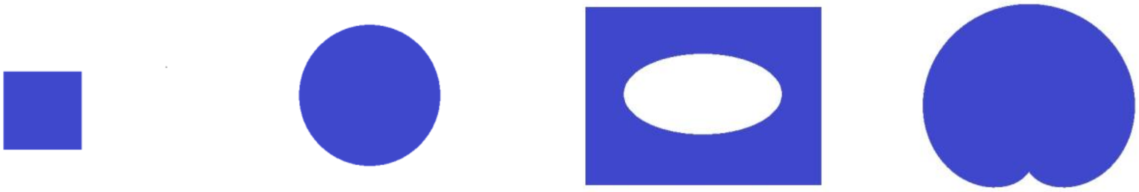

Figure 2: Images for Area Approximation

The simulations are built in Scratch, whose architecture makes designing and implementing these simulations simple and accessible. Sprites—the primary agents in the Scratch computing environment—can *sense* their surroundings. Thus, a sprite's behavior can be conditioned on whether it is touching the known color of the region of interest. When placed on a randomly selected point on the *stage* (the visible universe in Scratch), the outcome is logged as a success if touching the region or a failure if not. The area of the region can then be approximated by multiplying the percentage of successes by the area of the stage.

Once students have built a functioning simulation environment, they are asked to design and execute an area-approximation experiment. Their plan must include specification of a variety of parameters: initial shape(s); number of points; number of trials; size and shape of sprite; etc. Students perform their experiments, collect data, perform statistical analyses, and afterward reflect on the design of their experiment and their simulation environment.

Project Overview – Art Task

After students have demonstrated proficiency with the basic area approximation, they are tasked with generating a piece of computational art in their simulation environment. The key is the flexibility of sprites and the *stamp* feature in Scratch: sprites can imprint themselves on the stage by stamping, and the user has programmatic control over the size, shape, and color of the sprite.

By thinking of this stamp as drop of paint or a brush stroke, the Monte Carlo simulation process can thought of as painting a canvas.

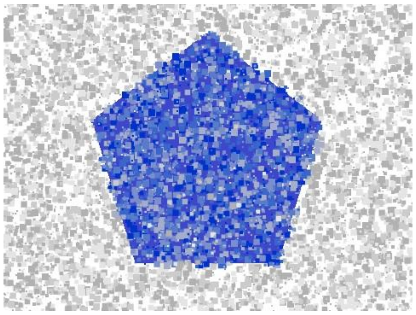

Figure 3: *Multi-hue, large stamp (6-pixel square) pentagon*

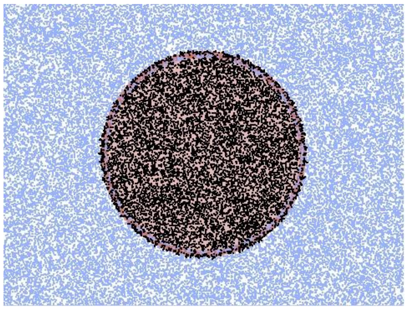

Figure 4: *Single-hue, small stamp (2-pixel square) circle*

Various visual effects can be produced by modifying the nature of the stamp. Sprites can take different shapes, sizes, and colors, and all of these parameters can vary dynamically within a given simulation. In addition, the parameters from the original Monte Carlo area approximations—initial shape, color, number of trails, etc—have various visual consequences.

Students were tasked to design an art project framework using their Monte Carlo area approximation as a starting point. Once the art project environment was in place, students ran a number of simulations using different parameter values, and identified combinations that they found appealing. In their final presentations, students had to show three early variations and explain the impact of the various parameter choices on the resulting images. Students submitted one final piece to represent their project and were asked to explain and defend their choice.

Results

Students produced a variety of imaginative, high quality work as a result of this project. Several students took a suggested approach of using a uniform stamp shape (here, a square) and varying size and color, as shown in Figures 5 and 6.

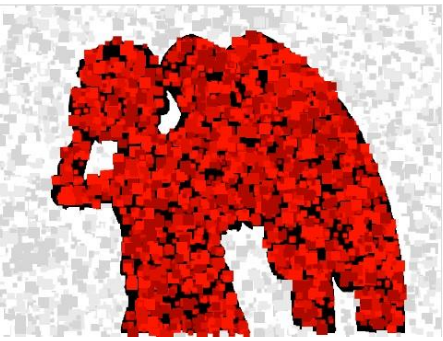

Figure 5: *Celestial Vibrancy*

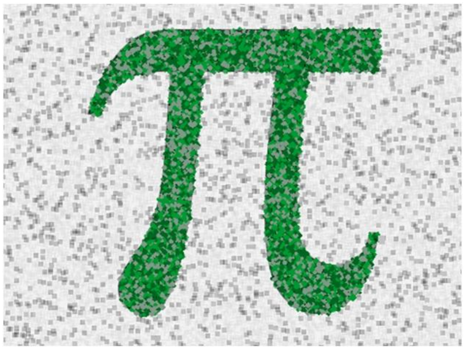

Figure 6: *Pi Mosaic*

Starting with a solid-color image of their choosing, students ran multiple simulations, experimenting with colors, shades, and the number, size, and distribution of stamps. The final works resemble random mosaics.

Others students focused on different parameters to explore the artistic space.

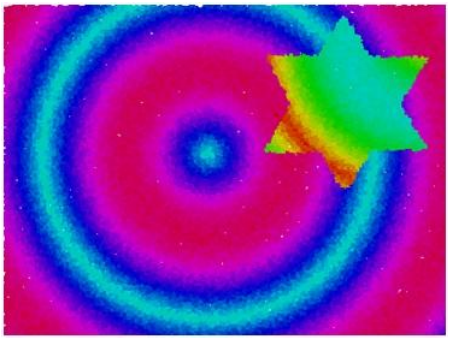

Figure 7: *Star in Transition*

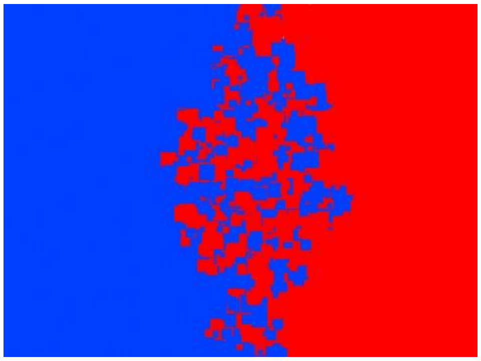

Figure 8: *Takeover Through Chaos*

In Figures 7 and 8, students began by designing their own custom starting image. They then ran simulations with varying size, color, and shape of the stamp, using location, distance, and adjacency conditions to control stamping. Students used elements of randomness to add visual variation.

Conclusions and Extensions

Area approximation via Monte Carlo simulation is an accessible topic to secondary students that meaningfully combines elements of mathematics, computer science, and statistics. In addition to promoting real mathematical and statistical work in the context of computing, it directly addresses several of the Common Core State Standards [4], including the practice standards CCS.MP4 *Model with Mathematics* and CCS.MP5 *Use Appropriate Tools Strategically*.

The art project, inspired by previous Bridges experiences, was successful in getting my students, who are comfortable thinking like mathematicians, thinking like artists and designers as well. Reflecting on their work, some students talked about creating a "digitized look" by varying sprite size, and others mentioned the role negative space played in their final aesthetic decisions. Students even took their final Monte Carlo-generated piece and added some finishing manual touches, disrupting the uniformity of the computer-generated work with a human artist's touch.

Extensions to future iterations of this project could include adding smoothing conditions to the image or introducing constraints on inherent quantities, such as total area or total color(s). In addition, elements of domino tiling [5], duotone tilings [6], and Truchet tilings [7] could be integrated into the Monte Carlo simulation process.

The Scratch programming environment worked well in this project in that students could quickly implement elementary Monte Carlo area approximations. However, there are significant restrictions with respect to color processing in Scratch. For example, sprites can not identify the color value of their location; they can only tell if they are touching a particular color. This limits the extent to which dynamic image processing can occur. Programming languages such as Processing or Python would provide more suitable environments for these more advanced explorations.

References

[1] http://scratch.mit.edu (as of 3/15/15)

[2] Malan, D. J. and Leitner, H. H. 2007. Scratch for budding computer scientists. SIGCSE Bull. 39, 1, 223-227.

[3] http://scratched.gse.harvard.edu/ (as of 3/15/15)

[4] National Governors Association Center for Best Practices, Council of Chief State School Offices, *Common Core State Standards Mathematics*, National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C., 2010.

[5] R.A. Bosch, “Constructing domino portraits,” in Tribute to a Mathemagician, ed. B. Cipra et al., A.K. Peters, 2004, 251-256.

[6] G.R. Greenfield, "Composite Digital Mosaics using Duotone Tiles", *Bridges 2009: Mathematics, Art, Architecture, Culture*, 2009, Kaplan C. and Sarhangi, R. (Eds.) 155-162

[7] R.A. Bosch, "Opt Art: Special Cases", *Bridges 2011: Mathematics, Music Art, Architecture,* 2011, Sarhangi, R., and Sequin, C. (Eds.) 249-1256