
Patches as an Expressive Medium for Agent-Based Modelling and Programming

Corey Brady, corey.brady@vanderbilt.edu

Department of Teaching and Learning, Vanderbilt University, Nashville, TN, USA

Lauren Vogelstein, lauren.e.vogelstein@vanderbilt.edu

Department of Teaching and Learning, Vanderbilt University, Nashville, TN, USA

Abstract

Agent-based modelling (ABM) is a powerful approach for simulating complexity and for understanding the emergent phenomena core to multiple disciplines across the physical and social sciences (Wilensky, 2001). ABM is thus often understood as an innovation in STEM education, providing a representational infrastructure for understanding complexity by “growing it” (Epstein & Axtell, 1996; Wilensky & Papert, 2010). While this is certainly true, we argue that expressive and artistic uses of “swarms” of computational agents can *also* provide accessible entry points for learners and can support them in developing a range of intuitions about the kinds of phenomena that they might simulated with ABM. This offers a “STEAM” oriented introduction to modelling, connecting *artistic* perspectives with scientific perspectives in fundamental ways.

In this paper we describe the iterative design and implementation of activities that highlight the expressive potential and social syntonicity (Brady et al, 2016) of one of the fundamental types of agent in the ABM toolkit (the “patches”). We describe a setting in which we have done design-based research over two years, in summer camps (entitled “Code Your Art”) and school-year activities involving rising fifth through eighth grade students (participants aged from 10-15) attending school in a mid-sized urban district in the southeastern USA with a high proportion of traditionally underserved and minoritized youth. Our research questions were:

1. How can we cultivate mappings between *patches* and *pixels* as a provocation for young learners new to programming and ABM to create personally-meaningful visual effects?
2. How can we cultivate mappings between *people* and *patches* as a provocation for youth new to ABM to think ambitiously about designing and creating *dynamic* visual effects?

We present designs two iterations of this camp and in school-year implementations in the classrooms of partner teachers, where investigation of the research questions has continued.

Looking toward future work, we suggest that this approach to making ABM ideas accessible also has a “high ceiling,” closing the paper with currently in-process work in the present school year at the schools of partner teachers, that aims to answer the following question:

3. How might the two above provocations to expressive work with patches combine to produce performance phenomena that could be put in conversation with topics in distributed computing normally considered advanced, including 2D cellular automata, fuzzy logic, and the emergent behaviour of computational systems?

This work contributes to the field as a proof of the feasibility of introducing the powerful infrastructure of ABM to young learners through artistic expression using the full text-based interface of NetLogo. Students drew on personal interests and social dynamics to make sense of and develop understandings of key ideas about agent-based representations. Future directions envision much larger groups of learners (stadium-sized collectives) to construct distributed computing environments that exhibit emergent properties in real dynamic displays.

Keywords

Agent-based modelling; Computational thinking; Group-based design; Digital expressivity

Introduction

Agent-based modeling (ABM) offers a powerful representational infrastructure (Hegedus & Moreno-Armella, 2009) for simulating and understanding emergent phenomena and complex systems (Wilensky, 2001; Wilensky & Rand, 2015). ABM *also* provides a medium that can be used to create computational art of various kinds. We argue that such expressive uses of the ABM toolkit can both be intrinsically valuable for learners and also serve as an entry point for learning about the agent-based approach. We present designs from the first two years of an ongoing project that engages young learners artistically with a grid of programmable *patches*, one of the representational building blocks of ABM.

Research Objectives

In motivating coding and mathematics through artistic expression, our goal is to identify entry points that might make some of the powerful ideas of ABM more accessible. We sought to stabilize these entry points in activities that could be reliably enacted with students in a variety of settings and with different facilitating teachers. We wanted these activities to place key ideas of ABM in a context where learners could build familiarity with them, guided by desires to create compelling visual effects. In taking a patch-centered view on this enterprise, we were guided initially by the following research questions:

1. How can we cultivate mappings between *patches* and *pixels* as a provocation for young learners new to programming and ABM to create personally-meaningful visual effects?
2. How can we cultivate mappings between *people* and *patches* as a provocation for youth new to ABM to think ambitiously about designing and creating *dynamic* visual effects?

As we have begun to see the independent expressive potential for collectives in playing the role of a patch grid, we have formulated an additional question for future work:

3. How might the two above provocations to expressive work with patches combine to produce performance phenomena that could be put in conversation with topics in distributed computing normally considered advanced, including 2D cellular automata, fuzzy logic, and the emergent behaviour of computational systems?

Literature Review

In this section we situate our work on the expressive potential of patches within the constructionist literature on agent-based modeling, in terms of three themes.

Our selection of NetLogo (Wilensky, 1999) was a pivotal design decision. We chose it as our construction environment from out of a wide range of alternatives that might be seen as better oriented to the age (middle school) and experience level (beginners) of our participants. We made this choice in part due to NetLogo's ability to produce images from the computational state of a grid of agents. This aligned with our aims to integrate graphical design with computational thinking and with reasoning about agent-aggregate relations and emergence. We also favored NetLogo in part due to its "high ceiling" nature, which ensured that expertise and agent intuitions developed by learners would provide them enduring connections to rich tools for thinking.

The two³³ principal types of agents in NetLogo are turtles and patches. The *turtle* is the more familiar type among constructionists, introduced as it was in the original LOGO. Our project's patch orientation set us apart from this line of work. We focused on the expressive visual potential of a grid of *immobile* agents that can be imagined in terms of the panels of a quilt or the pixels of a screen. Unlike turtles, these *patches* have *fixed* Cartesian coordinates and uniform shape (squares); but like turtles, they can change colour and hold variables. The turtle as a computational object has been extolled for various kinds of *syntonicity* (Papert, 1980)—attach points for learners

³³ Links are in fact a third type of agent, also treated as "first-class" in NetLogo. But a link can exist only as a connector between two other pre-existing agents.

to identify with the turtle and use its perspective as a lens on problems and phenomena. One of the challenges we set ourselves in this project was to identify ways that a learner or group of learners might identify with and think through the group of patches.

Below, we outline research statements that guide our enterprise of introducing young learners to ABM through expressive uses of NetLogo's patch grid, drawing sources from work on agent-based computing in particular and constructionist thought more broadly.

A characteristic perspective on entry points

Research within the Constructionist tradition (Papert & Harel, 1991), has been interested in engaging learners in computational thinking both (a) in focused, topic-specific explorations, using the design construct of microworlds (Papert, 1980), and (b) in broader, more discipline-general ways with open software environments and construction kits that help learners explore the generativity of systems of powerful ideas (Papert, 1980). Indeed, an important aspect of the power of constructionist environments is that they enable transitions back and forth between constructing *within* microworlds to the construction *of* microworlds, whether designing learning environments for others (Harel & Papert, 1990) or creating as objects-to-think-with (Papert, 1980) for oneself (or both of these at the same time).

This strategy—of establishing entry points that learners do not 'use up' but rather return to with new perspectives and purposes—can be seen as an instance of a general principle of constructionist design. Here, assuming the standing point of agent-based modeling, we outline several pairings of this kind that are relevant to our research questions and to the enterprise of connecting artistic expression with patches to broader adoption of an agent-based perspective.

Playful, artistic expressivity and scientific inquiry

Though the LOGO turtle was a deliberate *restructuration* (Wilensky & Papert, 2010) of Euclidean geometry and thus born as a citizen of MathLand (Papert, 1980), it was also immediately and enthusiastically offered to learners as an expressive partner with which to engineer beautiful and visually compelling creations. Turtles have thus been used to explore forms of artistic expressivity that make use of the capabilities of these agents, in environments such as TurtleArt (Papert, 1980; Bontá, Papert, & Silverman, 2010) and more recently, Scratch (Resnick et al, 2009). A fundamental idea behind this strand of the LOGO tradition is the notion that creating turtle graphics or producing media with turtles can introduce learners to the core principles of turtle geometry and/or programming in playful ways that foreground the construction of personally meaningful artifacts (Papert, 1980; Papert & Harel, 1991).

Playfulness is an enduring and generative feature of constructionist inquiry, and powerful ideas (Papert, 1980) are understood in the constructionist aesthetic to overlap strongly with *wonderful* ideas (cf, Duckworth, 2006). Moreover, in the specifically multi-agent setting of NetLogo and ABM, the legitimacy of exploring the expressive range of agents and their aggregate representations is captured in Wilensky & Rand's (2015) assertion of the value of "exploratory modeling" as a complement and partner to "phenomena-based modeling." We thus felt on solid ground in fostering explorations of the range of compelling and visually appealing effects that students might create with the patch grid.

Participatory (intrinsic and shared) and reified computational (externalized and shareable) representations

To help learners make sense of their ideas towards creating or using computational representations, constructionist researchers have often advocated that they enact their thinking and animate agent behaviours and interactions through role-play. In both single-turtle and multi-turtle programming, physical, embodied simulation has thus provided critical entry points for learner. For instance, the practice of *playing turtle* (Papert, 1980) supported LOGO learners in understanding the turtle's turning actions while drawing closed polygons. Imagining an agent's experience of its environment and its behavior in interactions led to *embodied modeling* of systems (Wilensky & Riesman 2006); and collective role-play enabled groups of learners gain conceptual

traction in modeling groups of turtles in activities called *star people* (Resnick & Wilensky, 1998) and *participatory simulations* (Brady et al, 2017; Colella et al 1999; Wilensky & Stroup, 1999).

As with our other dialectical pairs, constructionists do not necessarily think of participatory models as being “used up” on the way to building a reified computational artifact. As important as an external, runnable, and shareable computational representation is, this is not necessarily the single *telos* of modeling work. The concept of *syntoncity*, mentioned earlier, provides an enduring connection between participatory and externalized representations, supporting learners in assigning meaning to computational results and maintaining the connections to agents that help them to reason about models’ behavior under new conditions. And representations at these two extremes support the investigative moves that Edith Ackermann (1996; 1999) called “diving in” and “stepping back.”

Moreover, we have been interested in the enduring value that enactive representations have of their own, supporting learners in inferential discussion, in running new “socially distributed” experiments, and in reasoning about possibilities for change in the represented systems (Hjorth, Brady, & Wilensky, 2018; Reimers & Brady, 2019; Vogelstein, Brady, & Hall, 2017; 2019). In the Code Your Art camp activities, we set ourselves the challenge of giving classroom-sized groups of students shared and meaningful experiences with the NetLogo patch grid. We have reported on the diversity of ways our teachers took up and facilitated such patch participatory simulations (Vogelstein & Brady, 2019), and the aim is to continue to explore this activity space as a realm for experiences that are worth returning to and reasoning about.

These three themes – creating compelling entry points that are rich enough to be worth returning to; seeing artistic expressivity and conceptual depth as compatible and mutually supportive; and making use of groups of learners to create shared experiences of producing phenomena relevant to shared inquiry – drove many of the designs we present in this paper.

Methods and Data Collection

The activities described here occurred within the Computational-Thinking And Mathematics Play Spaces project, or CAMPS, which uses design-based research (Cobb et al, 2003) to identify ways of exploring computational thinking and mathematics with middle school learners in an expressive, artistic environment. Our first design iteration consisted of a one-week (five-day) free summer camp for middle school students, held in a middle school in a southeastern U.S. city.

After building a proposed curriculum, we worked with four teachers from the public schools of the district during an intensive one-week professional development and co-design workshop. The teachers alternated between engaging with activities in the student role, suggesting and testing adaptations, and working through their facilitation plans. During the camp itself, the four teachers worked in two pairs, each with their own group of students. They facilitated the camp activities themselves, with technical support as needed from the research team. During the intervening academic year, two of the four teachers invited the research team into their classrooms, and one started an after-school coding club. In the second summer, all four of the teachers returned and recruited another four. The second year’s professional development workshop thus paired a returning teacher with a newcomer, and tackled both minor and major revisions to the camp curriculum. In the minor-revision camp (“Image Camp”), we expanded on the Year 1 camp with iterative refinements based on student work and school-year findings. In the major-revision camp (“Action Camp”) we introduced a new curriculum, created in partnership with professional dancers, and foregrounding NetLogo *turtles* as opposed to patches. In this paper we confine ourselves to the design trajectory that produced the Image Camp, though aspects of our future work are informed by the design work to place professional artists’ perspectives and practices in conversation with agent-based representation.

The camp was titled “Code Your Art,” and advertised as involving computer programming and art for rising 6th, 7th, and 8th grade students. In camp sessions, multiple forms of data were collected, which support our design reflections here. Consenting students were interviewed at the start and end of the week and completed pre- and post-questionnaires. During activities, consenting

students were captured in screen-recordings, and multiple cameras (fixed cameras and body-mounted GoPros) documented 'offline' activities in each classroom. Finally, students' projects were captured as digital files and through the shared practice of "publishing" one's work to a classroom-specific online "gallery." These data sources have provided invaluable records of the experience of our activities, informing ongoing design and analysis.

Results

In this section, we describe designs related to our first two research questions: first, emphasizing connections for learners between patches and pixels; and second, between patches and people.

Patches and pixels.

Designs that encouraged learners to connect patches and pixels focused on image construction and manipulation. In both of our camps, we began with constructions using Perler™ beads (heat-fusible coloured plastic beads that students can lay out on a grid and form into a solid, unified construction by applying a hot iron). Alongside these physical constructions, students also could work with a virtual Perler bead design NetLogo model, with the same grid size (29x29) as the Perler bead frame. This permitted students to create paired constructions, as well as to begin their coding-based image manipulation work from an initial design shared with their physical construction.



Figure 1. Sample student Perler Bead Design constructions from year 1, as posted to the class galleries

These activities gave students the basic notion of images as comprised of pixels, each in a particular location and, for a fixed image, each having a particular colour.

Over the week, students engaged with, adapted, and modified effects—many using patches alone; others using turtles and patches in coordination. Effects using patches alone included logic that changed patches' colours, based on colour their current colour. This category included, code such as:

```
ask patches [ set pcolor pcolor + 10 ]
```

or

```
ask patches with [ pcolor = blue ] [ set pcolor red ]
```

Students made their own adaptations to such code, motivated by the challenges of new images or desired visual effects. For instance, they changed constants (e.g., set pcolor pcolor + 2), or, later, added logic to select patches whose pcolors were in a range (e.g., ask patches with [pcolor > 100 and pcolor < 110] [set pcolor pcolor - 90]). Such changes were particularly important as participants shifted to images that they collected from the internet.

A second category of effect involved changing patches' colours based on their coordinate locations. And a third involved using patch-owned variables to store multiple colours in each patch, allowing them to participate as pixels in multiple images. This enabled students to create patch performances that involved stop-motion animation or composite images in which different regions showed corresponding pixels from different images (see Figures 2, 3).



Figure 2. Five stills from a Year 1 final project, beginning with a pair of images with similar shape, and using patch memory and turtles to create a performance that composed and decomposed the images.

Turtles entered the picture in part as messengers or transmitters of colour from one part of a composition to another. These ideas supported many students in creating dynamic effects and transitions.



Figure 3. Still from a Year 1 project, using a high-resolution image with turtle and patch effects to “dematerialize” part of the image.

As described below, we focused in the school year on foregrounding and exploring the power of mathematical operations in the patch context. Our intention was to support students in creating “interactive image performances” and making these interactions open-ended and contingent on user interaction or randomness. We also aimed supporting a trend we noted of taking “actions on images” again in a way that foregrounded mathematical transformations.

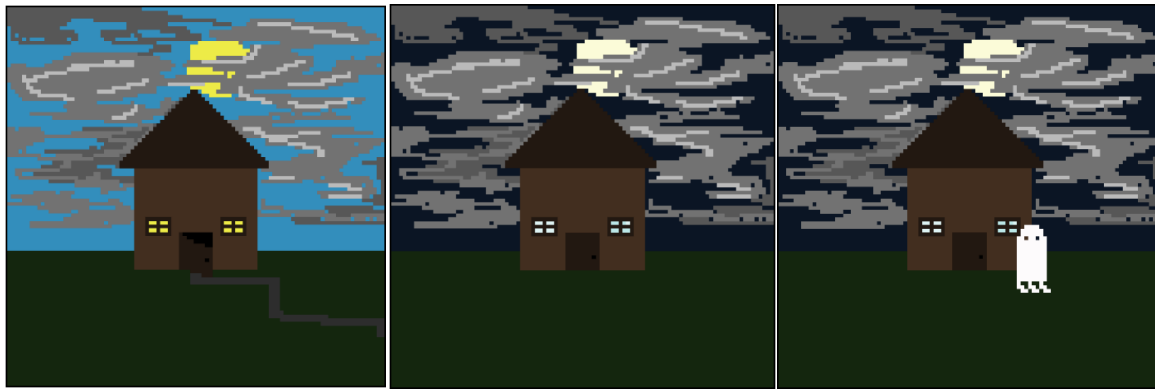


Figure 4. Three stills from a Year 2 final project, that began with a drawn image and used a variety of transformations to achieve effects such as the change from day to night, and haunting.

Patches and people.

The connections we encouraged between patches and *pixels* aimed to foster a view of the NetLogo environment as a ‘canvas’ for dynamic graphic design, visual effects, and interactive art. The connections we encouraged between patches and *people* aimed to support strategies for conceptualizing the patch agentset and orchestrating their collective action.

Year one experiences

One of the most dramatic examples of connecting patches with people was our use of “stadium cards” activities. Based on the “card stunt” events done in large sporting arenas (see Figure 5), stadium cards activities asked a classroom group of students to populate the patch grid. In the course of the five-day camp, we did stadium cards activities on the first three days. Through these activities, we introduced patches as agents, highlighting (a) that they could be spoken to as a whole group (agentset) or in sub-groups using a “with” selector; (b) that they had changeable *pcolor*, where the colour possibilities were indexed by a number; (c) that they had fixed location, indexed by *pxcor* and *pycor* coordinates similar to a Cartesian plane; (d) that while they could not get a view of the whole grid, they could see and communicate with their *neighbors*; and (e) that they could *sprout* turtles, who could “take on” their colour if asked.



Figure 5. Card stunts, shown at increasing scales.

Our two classrooms from the Year 1 camp implemented stadium cards activities in very different ways. Although both approaches were consistent with the goals expressed in the professional development sessions, each highlighted different aspects of the “patch experience.” As described in Vogelstein & Brady (2019), one approach to the activities foregrounded the limited knowledge of the patches, placing students inside the grid, holding their colour-cards (Figure 6).



Figure 6. Students as patches in a Stadium Card activity. Left: presenting colours. Right: changing colours.

An alternative approach placed students on the “sidelines” of the patch grid, laying their colour-cards on the floor. This arrangement focused on the limited *agency* of the patches, while affording them a view of the whole construction (Figure 7).



Figure 7. Stadium Card patch grid responding to: `ask patches [if pcolor = black [ set pcolor red ]]`.

Year two experiments

In our school-year work and in the Year 2 camp, we aimed to solidify promising directions that had emerged in our first-year explorations. Both our successes and our failures in this regard were worthy of mention.

Foregrounding mathematics and generative syntax

In an effort to foreground mathematical manipulations and their power as tools for patch effects, we designed and tested an approach to introducing new NetLogo syntax that foregrounded what we referred to as “combinatorial play” and that facilitated the diffusion of ideas in the class. To do this, we used a collaborative version of NetLogo called GbCC (Brady et al, 2018), which offered a public interactive gallery for sharing and remixing work.

For example, early on in the first day of the school-year activities, we introduced the command to the agentset of patches:

```
patches> if pxcor = 2 [ set pcolor red ]
```

Students typed this code in to their version of the activity interface, to confirm that their screen looked like the instructor's. They were asked to make a change to one part of that line of code, and to investigate the difference in the effect, and repeat. Whenever they created an interesting effect, they were told to Share their results, publishing their screen and the associated code. As students' results appeared in the gallery, they “bootstrapped” each other in exploring variations such as: