# Twinkle: Programming with Color

**Jay Silver**
MIT Media Lab
20 Ames St.
Cambridge, MA 02139 USA
silver@media.mit.edu

**Eric Rosenbaum**
MIT Media Lab
20 Ames St.
Cambridge, MA 02139 USA
ericr@media.mit.edu

Fig. 1. A musical score drawn with magic markers and played with a color sensor called "Twinkle"


*TEI 2010*, January, 2010, Cambridge, MA, USA
ACM 978-1-60558-246-7/09/04.

## Abstract
Twinkle allows anyone to program using colors in the real world. Twinkle uses a color sensor to read colors from arrangements of objects, drawings, or collages. Those colors are then mapped to certain outputs, like sounds, graphics, or robotic movements. Color patterns can even be used to control the color sensor itself, closing the loop. The result is that you can program a computer or a robot, or compose a musical score, just by drawing on a piece of paper with crayons. Of course it's not limited to crayons. You could build your program with Lego bricks, arrange your program with the multi colored leaves of early Fall, or think of any collection of objects in the world as a program: from a striped shirt to a handful of M&Ms. In the limit, several interesting new programming concepts emerge from this paradigm: commands are no longer discrete and rigid but mixable and smearable; the program counter becomes visible, handheld, and nondeterministic; and when the color sensor becomes the program counter the application space and the programming space become intertwined.

## Keywords
Tangible Programming, Toolkit, Color, Sensing

## ACM Classification Keywords
D.2.6 Programming Environments: Interactive Environ.

## General Terms
design

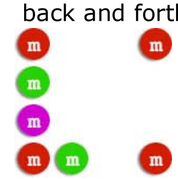
## Introduction
Previous examples of tangible programming typically involve connecting discrete objects, such as wooden railroad blocks, to create sequences of commands [1].

The authors of this paper see these approaches as too literal a translation from virtual programming to real-world tangible programming. Existing examples of programming with color do not leverage the affordances of colors in the real world [4], but show that complex programs can be constructed with color [5]. As in ambient programming [6], we hope to take programming off the screen and computationally enrich the environment, but we place our emphasis on representing programs with real-world objects.

## Twinkle
Twinkle departs markedly from prior systems by appropriating objects already commonly found in the environment. For example, people set out a sequence of colored objects representing commands. The objects are then traversed either by moving the color sensor by hand or by having an automated robot traverse the colors.

## Musical Scores
Imagine creating a musical score by making a colorful painting, and then traversing it non-linearly. The sensor, which doubles as the program counter, could be held by hand, so that each person can play her own variation on the tune.

## "Robot" Programming
Twinkle can also be used to program a creature, implemented either as a physical robot or as a digitally projected "light robot." In either case, the robot senses its color-coded environment, and moves through it, acting as a visible program counter. A sequence of colors represents commands for the robot, such as moves and turns. For example, you could make a rule (using a simple software interface such as Scratch [3]) that the robot should always move forward, but bounce the other way when it senses green. Then you could use grass to make a boundary and use two green leaves as paddles to play pong with the robot as the ball. Here, application space and programming space are intertwined, opening up new possibilities for intuitive use of computational thinking. In another scenario, you could dress the robot up as a dancer, and design a dance sequence with red meaning turn right, green meaning do a spin, and purple meaning wiggle back and forth. In this scenario, you could imagine



spitting 4 red M&Ms onto the page in a square shape to form an iterative loop out of successive right turns. Putting other colors in between the red would indicate the actions to perform while looping.

## Hopes and Dreams
The general hope is two-fold. On the one hand, we hope that people gain a heightened awareness of sequential patterns and material properties such as color in the world around them, while learning to see every day objects as more than the sum of their typical functions. On the other hand, we hope that tangible programming can take a step in a more fluid and intuitive direction, allowing anyone who knows how to swing a crayon to step up and play.

## Acknowledgements
Supported by LifeLong Kindergarten Group at MIT.

## Citations
[1] Horne, M. S. Designing Tangible Programming Languages for Classroom Use. Tangible and Embedded Interaction 2007, ACM Press.

[2] http://drawdio.com

[3] http://scratch.mit.edu

[4] http://en.wikipedia.org/wiki/SimTunes

[5] http://www.dangermouse.net/esoteric/piet.html

[6] Elumeze, N. Toward Ambient Programming for Children. CELDA 2006.