

# Undergraduate Advanced Project

Design and Implementation of an Array Tool for Classroom Learning

Partner

Elizabeth Simon

MIT Department of Electrical Engineering and Computer Science

MIT Department of Mathematics

May 23, 2013

## 1 Introduction

For my undergraduate advanced project I worked with the Classroom Learning Partner group to add new functionality to their tablet-based classroom interaction system. The system, whose current development is being funded by the *INK-12: Interactive Ink Inscriptions in K-12* research project (<http://ink-12.terc.edu>), enables students to use a tablet pen to create answers to in-class problems and wirelessly send the answers to the teacher. My project was to design and implement a tool that allows students to create arrays to represent and solve multiplication problems, especially two-digit by two-digit problems. The main challenges were to design a tool that was easy to use, seamlessly integrated with the current user interface, and struck a balance between using the tablet pen for freehand drawing vs using it as one would a mouse. The sections that follow describe the Classroom Learning Partner (CLP) software, the design and implementation of the array tool, and testing of the array tool in two fourth grade classrooms.

## 2 Background

CLP employs a model of interaction in which students use a tablet pen to create digital ink inscriptions in an electronic notebook, and the software allows students to share the inscriptions wirelessly with the teacher. Behind-the-scenes CLP attempts, when possible, to interpret the students' work in order to provide the teacher with feedback about student understanding. In addition to supporting drawing with a tablet pen, CLP has a variety of tools to help students create representations of mathematical concepts. The two main CLP tools used in mathematics to date are what we call *stamps* and *tiles*.<sup>[1][2]</sup> Stamps are especially helpful when students create representations that involve multiple identical images, e.g., 3 groups of 5 objects, as shown in Figure 1. Tiles are useful

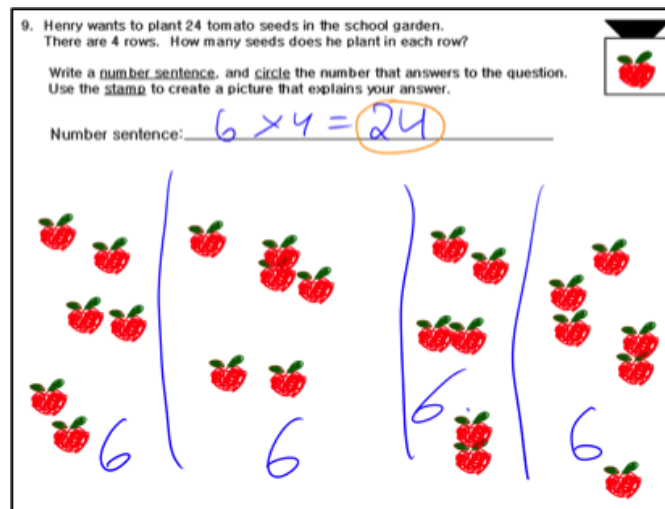


Figure 1: Fourth grade student work for multiplication problem: How many tomato seeds are in 4 rows if there are 6 seeds in each row?

in helping students learn to think abstractly about a problem, as shown in Figure 2. Creating representations with multiple identical images works well for one-digit multiplication. For two-digit multiplication problems, many math curricula teach students to use to represent and solve the problems. Students are taught to divide the side lengths of an array in order to break the problem into smaller problems that the students already

Alex and Henry are washing dirty dishes. What a mess!  
There are 7 stacks of plates. Each stack has 6 plates.  
How many plates are there in all?  $7 \times 6 = 42$   
Use the stamp or tiles or draw a picture to explain your answer.

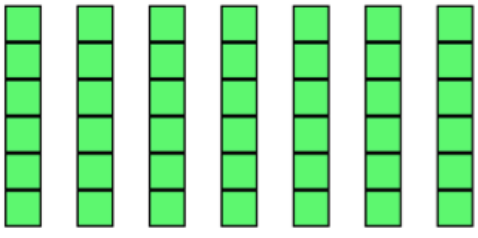


Figure 2: Fourth grade student work to show  $7 \times 6 = 42$

know how to solve.

### 3 Problem Statement

My project was to create a tool that allows students to create arrays while also enabling machine understanding of the arrays. The machine understanding enables the software to group student answers by similarity, which in turn makes it easier for a teacher to review the answers sent to her and choose representative examples to display for class discussion. A major part of the project was to decide how much drawing the students will do on their own versus how much should be aided by the software. If the software did all of the drawing, it would be easier for the system to interpret what was drawn. Research, however, suggests that drawing enables students to be creative and take ownership of their work.[3][4]

## 4 Design Process

Creating this tool involved some important design tradeoffs, and so I spent a lot of time developing a design. There were a variety of possible designs I could choose. On one end of the spectrum, the student could enter the dimensions and locations for divisions into a pop-up menu and the software could draw the array for them. On the other end, the student could hand-draw the array, and the system could attempt to interpret it. Attempting to have the software interpret a hand-drawn array would be very difficult, however; the system would have to identify a rectangle and bounds, as well as interpret a student's handwriting, neither of which is straightforward. (See Figure 3 for an example.) The first step in my project was to decide on a reasonable design somewhere between

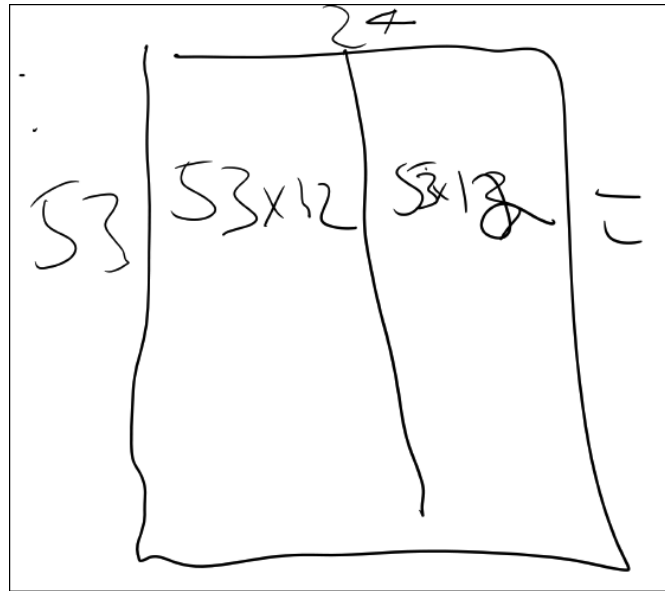


Figure 3: Student hand-drawn array for 53x24

these two extremes.

I worked with principal investigators of the INK-12 research project, Kimberle Koile and Andee Rubin, to come up with a design for my tool. We also met with teachers who

would be using the software in their classrooms to get feedback on the design.

At first we considered having students create the array by pulling out a rectangle to the dimensions they wanted. Since the focus of the lesson is on choosing and dividing appropriate array dimensions, rather than drawing proportional rectangles, we decided against this idea. Instead, we decided to let students specify the dimensions of the array using a pop-up menu. I decided to let the students specify both dimensions in the same pop-up because that was the simplest design in that it enables students to enter the two numbers quickly and easily. The pop-up has two number pads side by side where students can enter the number of rows and columns. We thought students new to the idea of arrays might misidentify rows as columns and vice versa, so I added a preview of the array to the pop-up with labels that update dynamically based on the numbers entered into the menu. We thought this would be a good idea because it would ensure that students don't accidentally switch the dimensions around and it would also help solidify student understanding of the difference between rows and columns.

After the student enters the number of rows and columns, he or she taps on an Enter button, and an array is displayed on the students notebook page. In my original implementation, the array was placed at the top left, but I changed this so that it would be less likely to be placed on top of text or something else already on the page. Now the array is placed further down on the page.

We experimented with several different display sizes for the array. Originally the array took up more than half of the page, but we changed it to be about a quarter of the page. We decided on this size because it gives the students more room to show their work on the page. A student can make an array larger or smaller by means of operators that are available when the student hovers over an array with their tablet pen. (These

operators, accessible via what we call *adorners*, are discussed in the next section.)

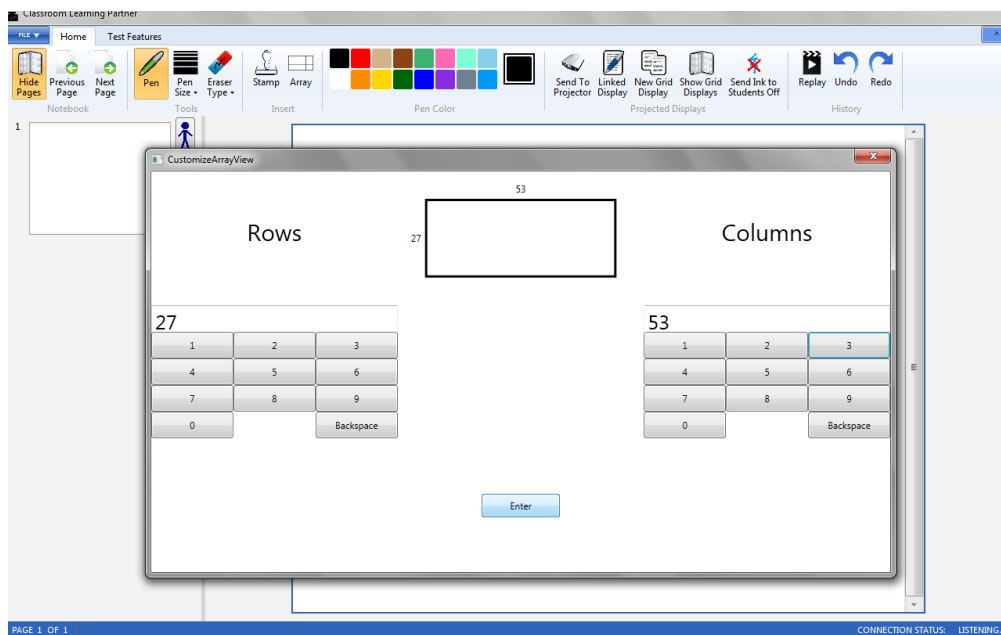


Figure 4: The pop-up menu for creating an array

#### 4.1 Interacting with the Array

In CLP software, students can interact with objects on the page using operators accessible via what we call *adorners*. If a student hovers over the object with the tablet pen, a dashed rectangle along with three icons pops up. These icons allow them to delete, move, or resize the page object. It is possible to adjust how quickly these adorners pop up by changing a variable in the code that specifies the timeout. I tried a few different time outs to make sure that the adorners don't pop up too quickly, but students don't have to wait too long for them to pop up.

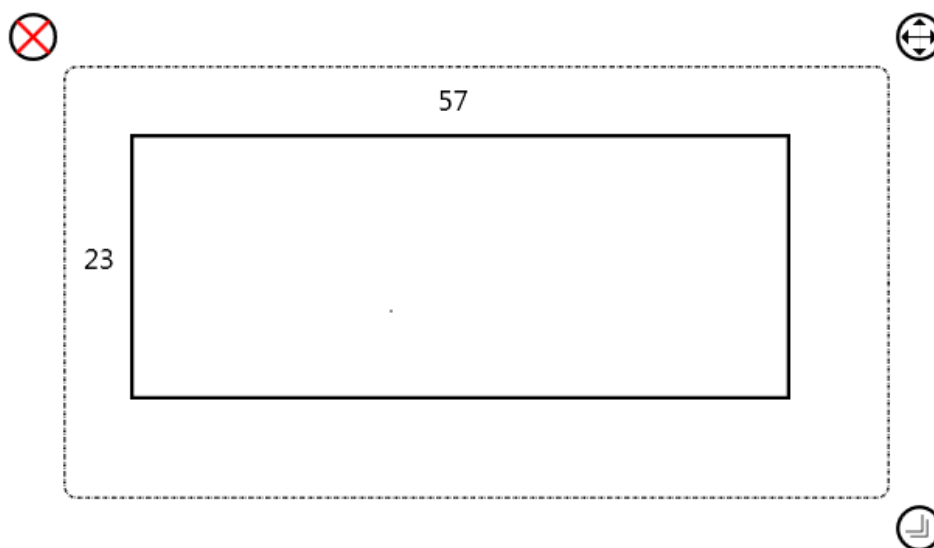


Figure 5: The icons outside the array can be used to delete, move, or resize the array

## 4.2 Creating Divisions

We considered a variety of different options for how to let students create divisions in the array. These possibilities included:

1. Students hand-draw the division
2. Students draw the division and the software adds a line where it thinks the division is
3. Students draw the division and the software replaces it with a line
4. Students tap where they want a division and it is drawn by the software

We decided that having the software figure out where the student intended to create a division from a hand-drawn line would be difficult and overly complicated: if the students drew something that was not intended to be a division, such as a label or calculation, the software would have no way of differentiating that writing from a division.

We thought that having the software be able to know definitively where the divisions were outweighed the benefit of letting students draw their divisions.

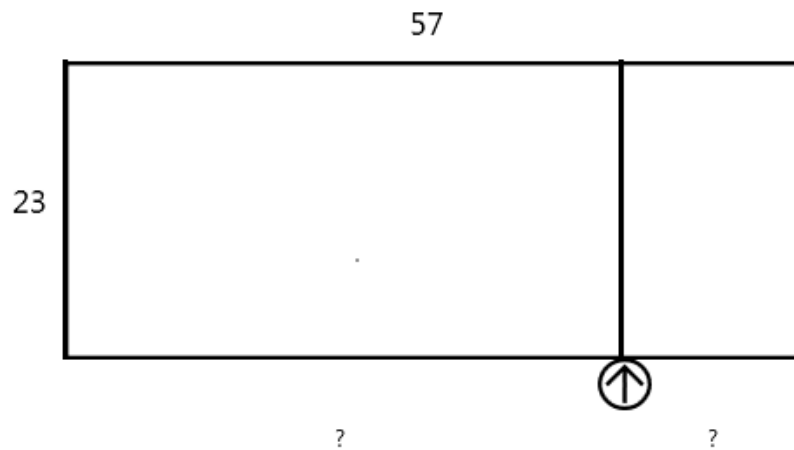


Figure 6: Arrow icon is used to create a division

In the design I implemented, when a student wants to create a division, he or she hovers near the bottom or right side of the array that has been placed on the page until an arrow pops up. The student then can use the pen to move the arrow to the desired location, tap the pen, and a division will be created. Students can delete divisions by using the top of their pen as they would use an eraser on a pencil.

An arrow icon was chosen because it provides feedback to the student about where a line would be drawn. I originally considered a scissors icon, but decided that scissors might indicate that the array would be split into two parts, each of which could then be moved independently, which is not the case. (This cutting functionality will be implemented as future work.)



### 4.3 Labels

Once the student creates an array division, we want them to be able to label the dimension of each portion of the side they divided. CLP has a built-in handwriting interpretation tool, which relies on Microsoft's handwriting recognition software, and we considered having the students write labels that would then be automatically interpreted by the software. Testing with the handwriting recognition routines, however, revealed that the recognition was not accurate enough on elementary students handwriting to be used reliably. Instead, we opted to have students use a number pad to specify the dimensions. In my implementation, when a division is created two question marks appear on the screen where the labels would go, and the students click on the question mark to pop up a number pad for entering the value. The chosen number then replaces the question mark on the screen. If they decide to change the label they can click on the number, which pops up the number pad again.

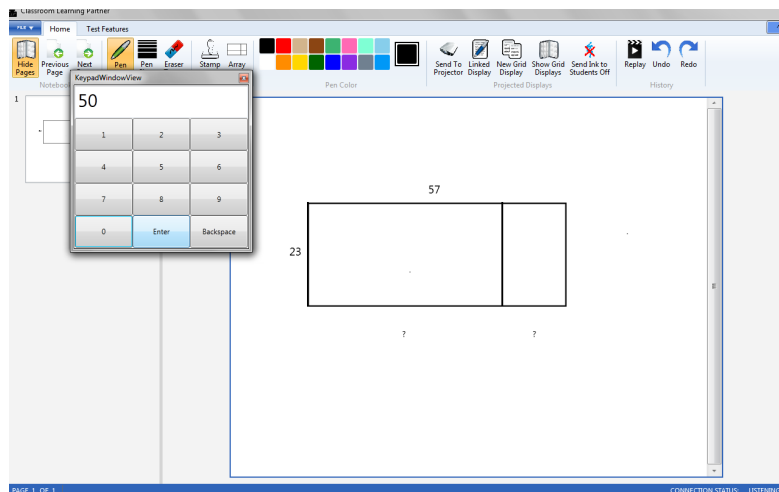


Figure 7: A number pad is used to label new segments created by the divisions

I was originally storing the location of the array divisions in two lists, one for vertical and one for horizontal, but this representation proved unnecessarily complicated

with the introduction of labels. I switched to a new representation in which each division creates two new objects called `CLPArrayDivisions`. Each `CLPArrayDivision` is the part of the array on either side of the division. The `CLPArrayDivision` object stores the location, size, orientation and value of the label. This made it much simpler to display the value on the side of the array.

## **4.4 Different Versions**

Because different teachers had different requests for the tool, we created different versions. I added a flag in the code to turn off the division creating behavior so that there would be a version where students do not create divisions. This version is useful for students who are not comfortable yet with the concept of dividing up arrays, or who are still getting used to using the tool.

### **4.4.1 Grid**

Some teachers wanted the arrays to have a grid in order to help their students understand the concept of rows and columns. The grid is especially important for students who haven't developed a complete understanding of arrays as an abstract concept. We considered a few different display options, including tic marks on all sides, tic marks on two sides, or a full grid within the array boundary. We decided to use the full grid, both because it was easier to implement and because it more clearly showed rows and columns, but with an option for turning the grid off in the code. (In future work, this grid on/off feature will be accessible to teachers and students.) We also needed to decide on an appropriate color for the grid. I decided to make the grid a pale blue similar to graph paper so it would not be too distracting but would still be visible.

8

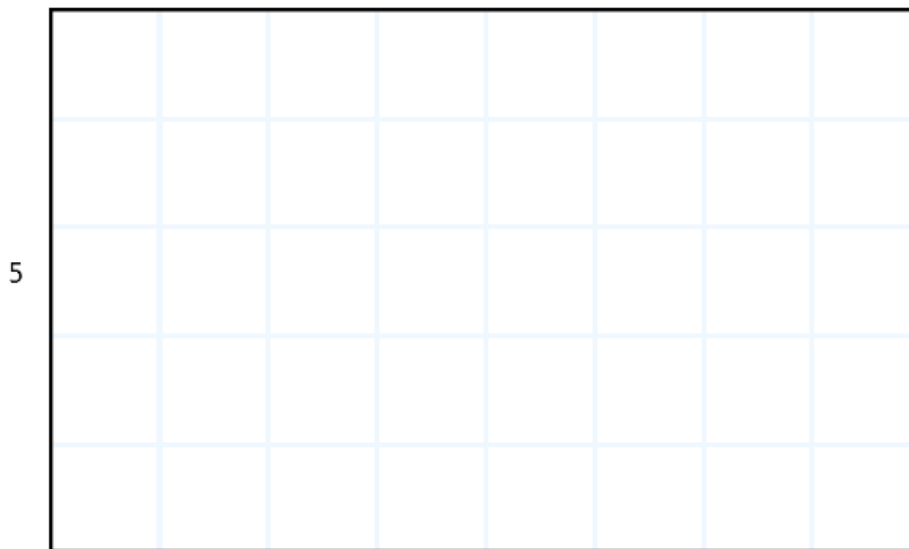


Figure 8: An array with the grid turned on

## 5 Testing

After I built the array tool, we tested it in two classrooms at Northeast Elementary School in Waltham, MA. One of the classes had learned about arrays earlier in the year and just needed a short reminder of the concept before using the tool. This class was not familiar with the technology so they also needed to be taught how to use the tablets. The other class, a small class of special needs students, was experienced with using the tablets but they were not familiar with arrays. Before they were able to use the array tool they needed to learn about the concept of arrays and how they work as a model for multiplication. Testing the tool in these two different classrooms allowed us to get a diverse set of feedback from students with different needs.

The first class started out using the arrays without divisions. Since this class was doing two-digit-by two-digit multiplication, we had them use a version without a grid. The

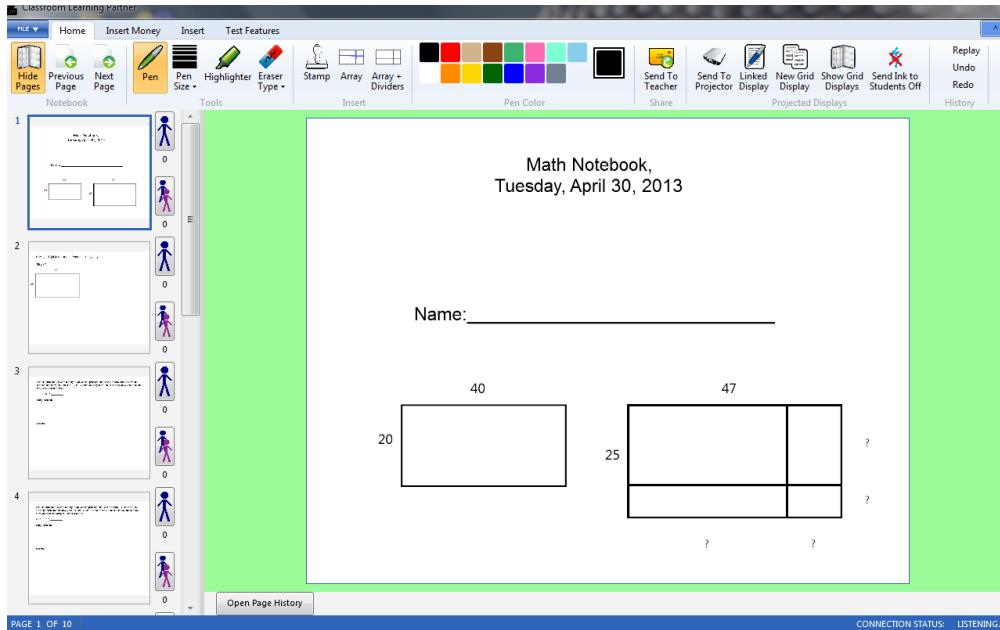


Figure 9: The CLP interface as it would appear in use by a student: students can click on the *Array* or *Array+Dividers* buttons to use the array tool

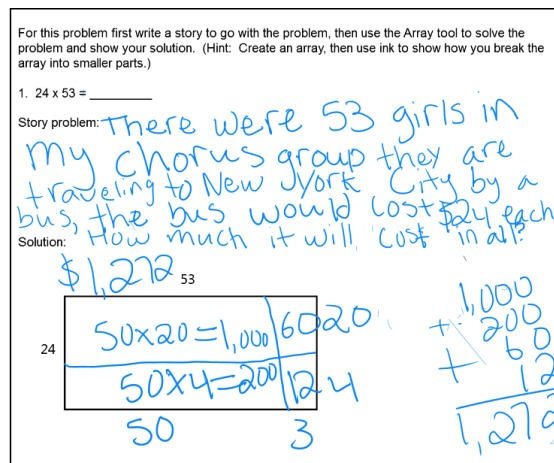


Figure 10: A student using the array tool without the divisions

tool seemed to work well for the students. We later had them use the tool with the divisions, which also seemed to work well. The major piece of feedback we got from many of the students was that the adorners popped up too quickly and made it difficult to work with the array. To fix this issue we adjusted the timeout so that the adorners

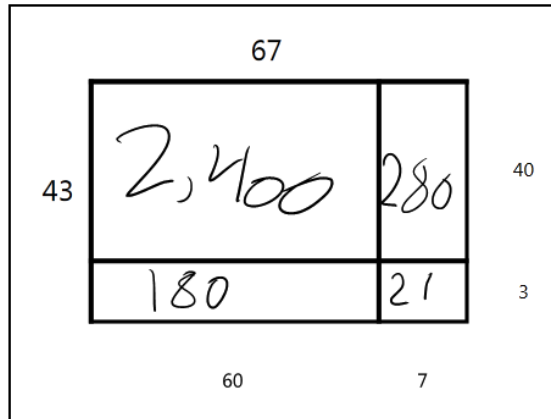


Figure 11: A student using the array tool with the divisions

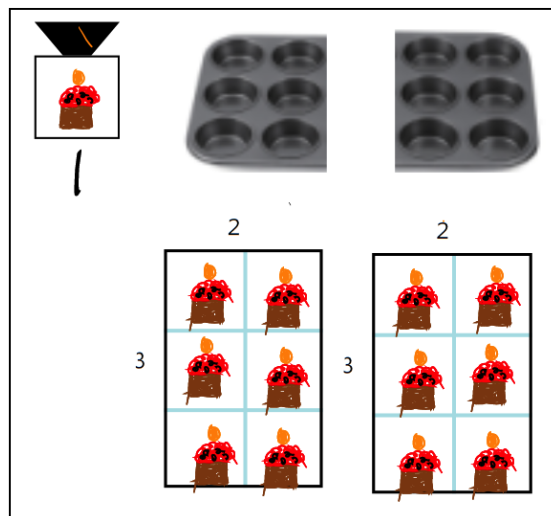


Figure 12: A student using the array tool with the grid in combination with stamps

wouldn't pop up so quickly.

For the special needs class, we started them out with a version with a grid and the division behavior turned off. The array tool worked well for the students. They used the stamp tool, which makes copies of a drawing, to fill in the squares of the grid with drawings of plants to represent a garden. The major piece of feedback we received from this class was that the grid lines were too light and were hard to see for some of the

students. To fix this, we changed the color to a darker shade of blue which seemed to work better.

## 6 Future work

There are several improvements that I'd like to add to the array tool. We have been considering a better way to provide access to operators on the arrays and other page objects. Instead of having adorners pop up when a pen hovers over the object, for example, we might instead have users select an object then explicitly choose to pop up operators. We would also like to add an adorer to the array that can turn the grid on or off so that students could decide if and/or when a grid might help them. We also could modify the user interface to give the teacher an option about whether to make the grid on/off feature available to students for a particular lesson. We also might want to enable arrays to be snapped together, as sometimes students find it helpful to think about multiplication or division by combining arrays instead of dividing them. Other possible features we could add would be the ability to move divisions around using the pen or the ability to split an array into smaller arrays that could then be manipulated independently.

Building this array tool has been a useful step in learning how to design and implement educational tools that work well for students, and the tool itself has provided students with the opportunity to improve their multiplication skills. Future versions of the tool will provide more features to enable mathematics students to further improve their skills.

## References

- [1] Koile, K., Reider, D., and Rubin, A. INK-12: a pen-based wireless classroom interaction system for K-12. In *The Impact of Tablet PCs and Pen-based Technology on Education: Evidence and Outcomes*. Reed, R. and Berque, D. (editors). Purdue University Press, 2010.
- [2] Koile, K. and Rubin, A. Machine interpretation of students hand-drawn mathematical representations. In *Proceedings of WIPTTE, 2013*.
- [3] Carruthers, E. and Worthington, M. *Childrens Mathematics: Making Marks, Making Meaning*. Sage Publications, London. 2006.
- [4] Woleck, K. R. Listen to their pictures: an investigation of childrens mathematical drawings. In A. Cuoco (Ed.), *The Roles of Representation in School Mathematics*. Reston, VA: National Council of Teachers of Mathematics, 2001, 215-227.