

# Birth of a Virtual Manipulative

S. Asli Özgün-Koca and Michael Meagher

**Abstract:** Teachers use hands-on manipulatives to help students understand mathematical concepts more concretely. Unfortunately, the use of hands-on materials alone may be problematic as students explore higher-level mathematical topics. In this paper, the authors explore the use of virtual manipulatives - online java applets - as a means to enhance the impact of hands-on materials. Through the exploration of the classic “Spaghetti Sine Curve” activity, the authors highlight intellectual synergy provided to students by the combination of virtual and concrete materials.

**Keywords:** GeoGebra, manipulatives, virtual environments

## 1. INTRODUCTION

Teachers use many instructional tools in mathematics education in order to help students understand mathematical concepts. Moyer (2001) and Moyer & Jones (2004) among others have researched the unique power of manipulatives, both virtual and physical, in supporting learner understanding (see also Cass, Cates, Smith, & Jackson, 2003; Cavanaugh, Gillan, Bosnick, Hess & Scott, 2008; Moch, 2001). In particular, Namukasa, Stanley, and Tuchtie (2009) highlight the visual and kinesthetic senses that are activated when manipulatives are being used. While physical manipulatives are very common in elementary mathematics classes, many secondary mathematics teachers also want to provide kinesthetic experiences for their students, even for higher mathematical topics such as trigonometry. One classic example (see Cookie (2006), NCTM (n. d.), and Sturdivant (2002)) is the use of a unit circle and spaghetti strands to make the connection between the sides of a right-angled triangle and a sine curve. In order to do this students create a series of right-angled triangles making angles of 10, 20, 30 degrees etc., and for each triangle they measure the opposite side (representing the sine) and break a strand of spaghetti with that length and transfer the strand to the axes. See, for example, the image in Figure 1, top, where the values of sine for 10 and 20 degrees have been measured in the unit circle and transferred to the Cartesian axes. The second image, shown in Figure 1, bottom, and third (Figure 2) show the spaghetti strands after transfer for values, every 10 degrees, up to 180 degrees and 360 degrees, respectively. As the strands are transferred the ends of the strands describe a sine curve on the domain of real numbers.

Manuscript received March 21, 2012; revised May 9, 2012; accepted May 9, 2012.

S. Asli Özgün-Koca is an associate professor of mathematics education at Wayne State University, Detroit MI (e-mail: ay3141@wayne.edu). Michael Meagher is an assistant professor of mathematics education at Brooklyn College, Brooklyn NY (email: mmeagher@brooklyn.cuny.edu)

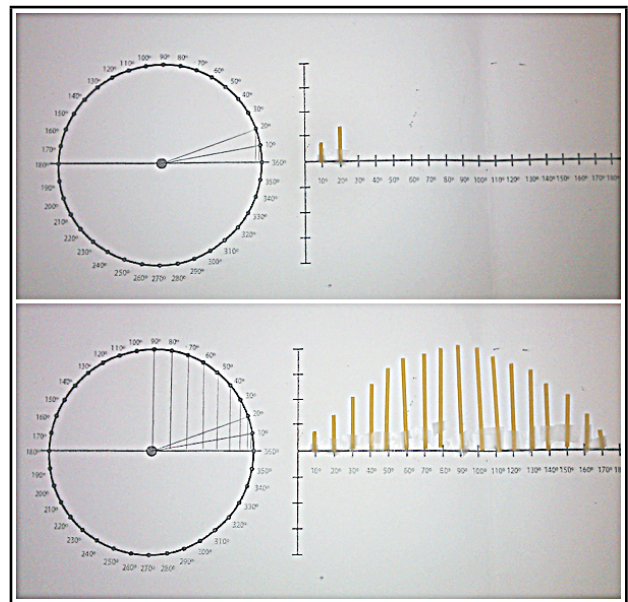


Fig 1: Partial sinusoidal graphs constructed with spaghetti

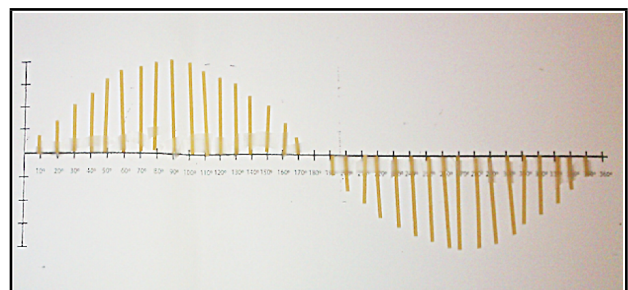


Fig 2: Sinusoidal graph for one complete period constructed with spaghetti

## 2. THEORETICAL PERSPECTIVE AND RELATED LITERATURE

Our goal in this article is to give an example of how a bridge can be made from a physical manipulative to a

virtual manipulative using advanced digital technologies. By describing in detail each step that goes into making the sketch, we reflect on the pedagogical decisions that inform the design to make the manipulative a useful learning tool for students trying to understand the mathematical connection between the unit circle and the sine curve. Thus we aim to provide a particular example of a virtual manipulative, to provide readers with some techniques for achieving certain functionalities in the GeoGebra environment, and, most importantly, to explain the pedagogical thinking to support student learning behind the design of a particular virtual manipulative, which can then be applied to the design of other manipulatives. Starting from physical manipulatives we believe that a carefully designed experience with a virtual manipulative can help students in thinking more abstractly and can help them make connections to the symbolic realm of mathematics. Bruner (1966) discusses the journey that students take from using these concrete materials to abstract higher-level thinking. He provides a detailed theory of representations and classifies them into three categories: enactive, iconic, and symbolic modes of representations. The enactive stage of representation is action-oriented, and representations or manipulatives are physically manipulated. The second level of representation is the iconic form, in which events and ideas are summarized and manipulated by perceptions and images. At the highest level of representation, learners begin to manipulate symbols abstractly and Bruner argues that “certain features of symbolic systems . . . only now [become] understood . . . [features] beyond what is possible through actions or images” (Bruner, 1966, p. 11). Moyer, Bolyard, and Spikell (2001) define virtual manipulatives as “interactive, Web-based, visual representation[s] of a dynamic object that provides opportunities for constructing mathematical knowledge” (p. 3) although, as Özgün-Koca & Edwards (2011) have shown, virtual manipulatives can be authored by teachers in environments such as GeoGebra and used offline. Our interpretation of Bruner’s modes of representations is that virtual manipulatives provide an environment which lies between enactive and iconic modes of representations. While virtual manipulatives are clearly action-oriented they are based on pictures like iconic mode of representation. Additionally, they have the capability of linking different representations (Namukasa et al., 2009). One common thread in the research comparing the use of virtual manipulatives with their physical counterparts is that there are unique beneficial characteristics of both types of manipulatives (Brown, 2007; Olkun, 2003; Özgün-Koca & Edwards, 2011; Rosen & Hoffman, 2009; Suh & Moyer, 2007; Zacharia, Olympiou, & Papaevripidou, 2008). In the following paragraphs, we will share the design of a virtual manipulative that aims to bring the physical spaghetti activity explained at the beginning of this paper to new levels. Our aim is to share pedagogical decisions that we made when

we created the virtual manipulative and how we accomplished them technologically (other than basic geometric constructions) with GeoGebra.

### 3. VIRTUAL SPAGHETTI

Our first decision was to use only the Graphics view to have a clean and simple environment by going to the View tab on the upper menu and unselecting Algebra and Spreadsheet windows (see Figure 3). One can change the unit of the axis by selecting Options from the upper menu, then choosing Settings and Graphics. We created the unit circle on the left side of the window by choosing Circle with Center and Radius. Having the unit circle on the coordinate axis informed our constructions and calculations for later purposes, which we will discuss in detail below. After creating the radius by using Segment between Two Points, we constructed and measured the angle,  $\alpha$  (shown as 47 degrees in Figure 3). Our next construction was the line segment  $BE$  by using Segment between Two Points to complete the right triangle  $ABE$ . Following that, we measured the length of the vertical leg opposite of the angle,  $\alpha$ . The Point  $B$  on the circumference of the unit circle can be dragged along the circumference of the circle. Even at this point, an environment is created for students to interact with the virtual manipulative and think about how the length of the vertical leg of the right triangle changes as you drag the Point  $B$  and go around the circle. Another initial decision that we made was about how many decimal places to show. In order to have a better representation, we decided to have 4 decimal places by choosing Options from the upper menu and then Rounding. However, that decision made the screen become too crowded with information. Therefore, in order to round the angle representation to a whole number, we opened Object Properties for the angle  $\alpha$  and under the Basic tab, we set Show Label as Caption. Moreover, under the Scripting tab and under the On Update tab, we entered `SetCaption[  $\alpha$ , “”+round( $\alpha$ )]`.

After completing the initial constructions, we needed to decide how to place the virtual spaghetti strands on the coordinate axes. Our first thought was to use the Segment command `Segment[point A, point B]` which creates a line segment through points  $A$  and  $B$ . Using the Segment command, we could ask students to make an entry for every 10 degrees. We provide the following text on the screen to guide the students:

For every  $10^\circ$ , observe the length of  $BE$  and create a line segment on the coordinate plane. In order to create your line segments, first drag  $B$  and then type `Segment[(x1, y1), (x2, y2)]` in your input line with the coordinates of beginning and end points for your segment and hit Enter. Example: `Segment[(10°, 0), (10°, 0.17)]`

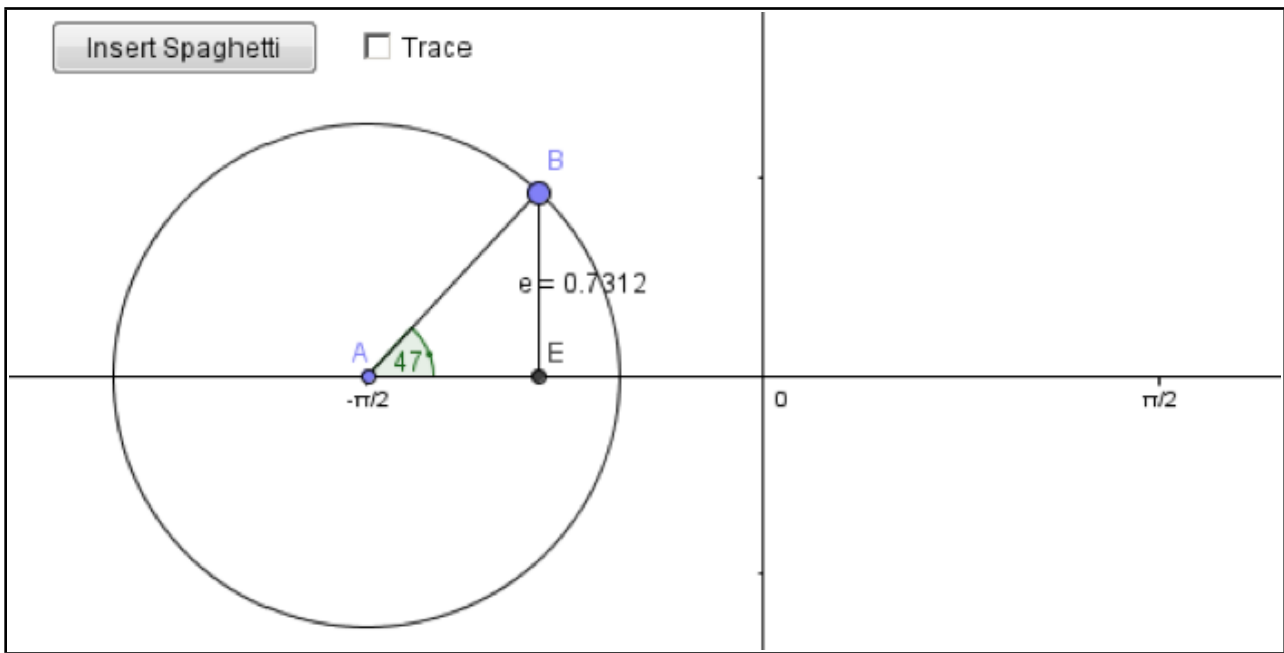


Fig 3: Initial construction

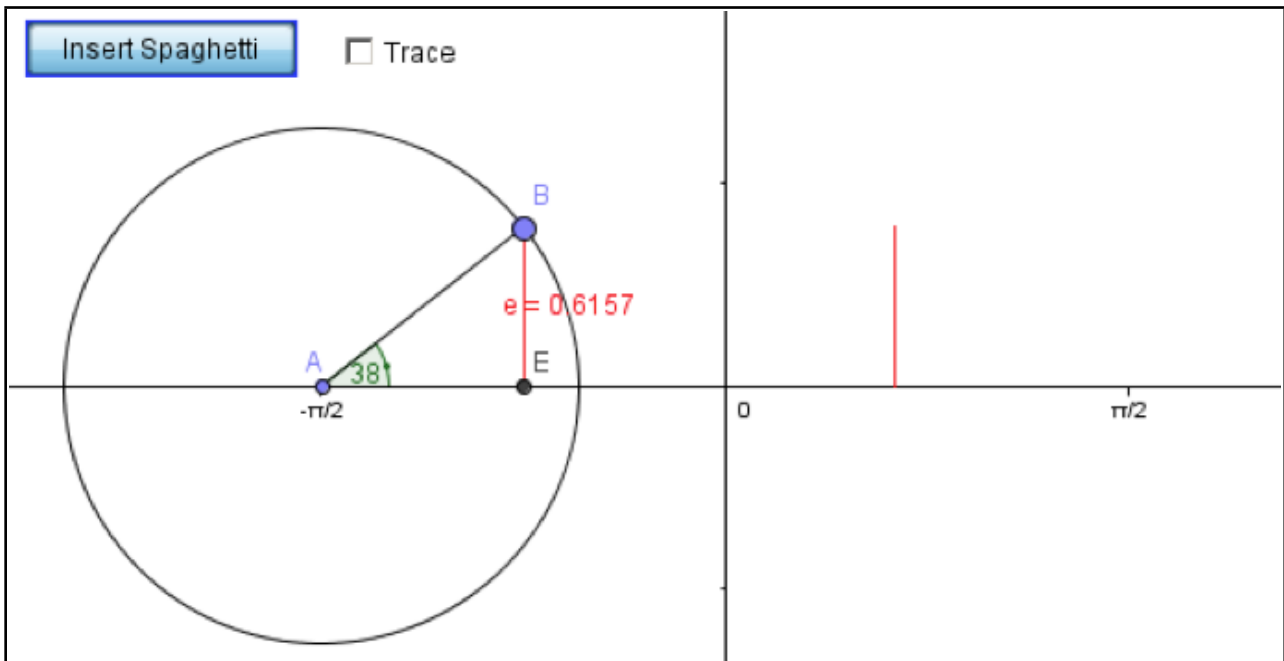


Fig 4: Inserting the Virtual Spaghetti Strand

We thought this process was nearly as cumbersome as putting physical spaghetti strands on the vertical leg of the right triangle, breaking them and pasting them to the coordinate axes. Also students could enter incorrect values which would create an unintended visual. However, at that time we did not know how to place virtual spaghetti strands on the coordinate axes more smoothly. In the release notes for GeoGebra 4.0, we realized that there was a new feature called `Insert Button`. `Insert Button` “is used to execute a series of GeoGebra commands with a single click on a button.” The `Insert Button` tool allows the insertion of a button in the `Graphics View`. When the button is created, the caption is provided as GeoGebra script to be executed. Therefore, we thought that this would work perfectly. Our next step was to decide how to capture the instances of the angle  $\alpha$  and the length of the vertical leg of the right triangle. `CopyFreeObject` command “creates a free copy of the object and preserves all basic Object Properties” ([http://wiki.geogebra.org/en/CopyFreeObject\\_Command](http://wiki.geogebra.org/en/CopyFreeObject_Command)). Using the `Copy Free Object` command, we wrote a script to insert virtual spaghetti strands to the specific point on the coordinate axes. Essentially, the script allows one to capture the current angle,  $\alpha$ , and length of the vertical leg of the right triangle each time after one drags the Point  $B$ . Therefore, we decided to use the following script:

```
Segment[(CopyFreeObject[\alpha],0), (CopyFreeObject[\alpha], CopyFreeObject[y(B)])]
```

Even though this looks like a complex script, it is fairly simple. We can dissect this script as follows:

- A segment is defined on the basis of two points and placed between those two points. We used the command `Segment[point A, point B]`.
- The coordinates of the first point are defined as `(CopyFreeObject[\alpha],0)`. It captures the current degrees of the angle  $\alpha$  for the  $x$ -coordinate and pairs it up with 0 for the  $y$ -coordinate. This point decides the place of the virtual spaghetti strand on the  $x$ -axis.
- The coordinates of the second point are defined as `(CopyFreeObject[\alpha], CopyFreeObject[y(B)])`. This time both coordinates of the ordered pair are captured. The measurement of the angle  $\alpha$  is captured for the  $x$ -coordinate and the  $y$ -coordinate of the Point  $B$  is captured for the  $y$ -coordinate. Using these coordinates, a segment is placed on the  $x$ -axis capturing the current value of  $\alpha$  and the  $y$ -coordinate of the Point  $B(\alpha, 0)$  and  $(\alpha, y(B))$ . Constructing the unit circle on the coordinate axes helped us create the negative part of the sine curve easily. The length of  $BE$  is always positive and it is not used in the background scripting and calculations at all. However, it provides a powerful visual feedback for students.

In order to highlight the relationship with the length of  $BE$  and the length of the virtual spaghetti, we decided to

use color to provide a better visual. Using color will allow students to see the specific connection between the length on the side of the triangle and the strand placed on the axis. In order to highlight this connection we wanted to change the color of the segment  $BE$  at the moment a student clicks the “Insert Spaghetti” button and at the same time have the segment appear in place on the axis in the same color. Having highlighted this connection at the moment the virtual transfer of the virtual spaghetti strand is made, as one drags the Point  $B$  again, both segments’ color revert to black. Another reason for using color in this way is that after inserting a couple of virtual spaghetti strands, it may be difficult for students to keep track of which one they inserted last. In order to achieve this use of color, we took the following steps:

- First we created segment  $t$  using the `Segment` command: `t=Segment[(0,0),(0,0)]`. The reason that we defined  $t$  as being at the origin is to hide it when one first starts the GeoGebra file. Every time one drags the Point  $B$ , the segment  $t$  will be redefined. That is why we clicked on `Show Trace` under `Object Properties` to keep the virtual spaghetti strands on the screen.
- In the script of the `Insert Button` we redefined  $t$  again as `t=Segment[(CopyFreeObject[\alpha],0), (CopyFreeObject[\alpha], CopyFreeObject[y(B)])]`
- We changed its color to red using the `SetColor` command: `SetColor(t,‘red’)`.
- We repeated the same step for the line segment  $BE$  which was called  $e$  in the GeoGebra file: `SetColor(e, ‘red’)`.

Based on these observations, the final script for our “Insert Spaghetti” button was:

```
t=Segment[(CopyFreeObject[\alpha],0), (CopyFreeObject[\alpha], CopyFreeObject[y(B)])]
SetColor(t,‘red’)
SetColor(e, ‘red’)
```

We need to change the colors back to black as one starts dragging the Point  $B$ . That is why we needed a script for the Point  $B$ . Under `Object Properties`, under the `Scripting` tab, and under the `OnUpdate` tab of the Point  $B$ , we entered the following commands:

```
SetColor(t,‘black’)
SetColor(e, ‘black’)
```

This way, we were able to change the colors back to black for the two line segments. Now, we can ask students to drag  $B$  to obtain specific angles and `Insert Spaghetti` with one click (see Figure 4).

Figure 5 shows a screenshot with 3 virtual spaghetti strands inserted. As you can observe, the spaghetti strand inserted last is red, which sets it apart from the previous ones. Moreover, the vertical right leg of the triangle is also red, providing students with a visual connection.

This pedagogical decision of using color to help students to make connections more effectively was not straightforward to execute technologically. However, it uses the capabilities of the technological environment fully and sets it apart from the physical version of the activity. Figure 6 displays a completed figure with virtual spaghetti strands inserted every 10 degrees from 0 to 180. Compared to the hands-on version, this graph is neater and provides students with a powerful visual that complements the physical manipulative.

We argue that this virtual environment has aspects to offer, such as the precision shown in Figure 6, which are not available with physical manipulatives. We further exploited the affordances of advanced digital technologies, creating a function which traces the sine function as a final confirmation after students have inserted their virtual spaghetti strands. In order to accomplish that we took the following steps:

- We first created a point  $F$  defined as  $(\alpha, y(B))$  in a similar manner to the use of angle  $\alpha$  and  $y$ -coordinate of  $B$  as previously discussed.
- We clicked on the Show Trace option under Object Properties.
- We did not want to make  $F$  visible at the beginning of the activity, hence we wanted to use the Check Box to Show/Hide Object tool.
- When one clicks to add a Check Box to Show/Hide Object on the upper menu and an empty space on the screen, a dialog box is opened and a caption can be provided such as Trace.
- Then it asks you to choose a defined object to link it - in our case this was the Point  $F$ .
- The checkbox needs to be checked in order to be able to see the Point  $F$ .

Now when the GeoGebra file is started, the Point  $F$  is hidden. When the Trace checkbox is checked, dragging the Point  $B$  would generate a series of points as in Figure 7.

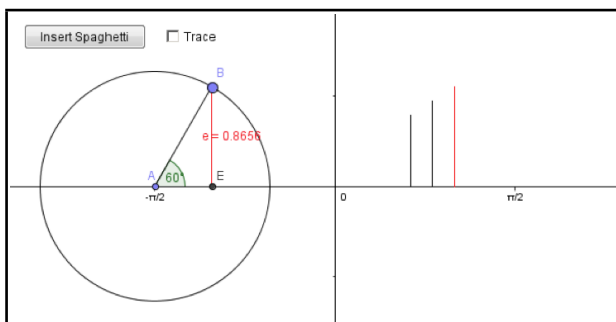


Fig 5: Three Virtual Spaghetti Strands Inserted

#### 4. CONCLUDING THOUGHTS

There are many websites offering a variety of virtual manipulatives for mathematics teachers to choose from.

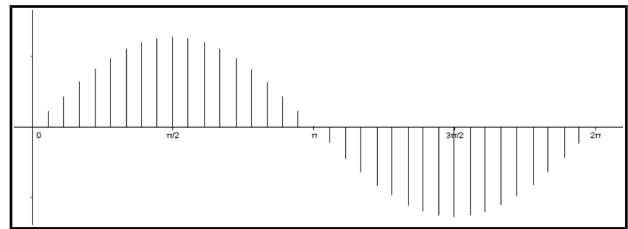


Fig 6: Completed figure

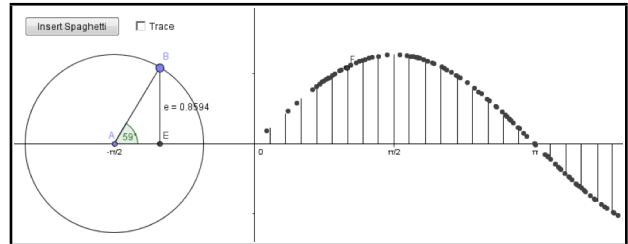


Fig 7: Final tracing

However, teachers often have very specific goals, and it can be quite difficult to locate a virtual manipulative which can fulfill expectations completely. When we were looking for a virtual manipulative which would transform the classical hands-on “spaghetti sine” activity into a dynamic interactive environment, we could not find one so we created our own virtual manipulative. Creating our own manipulative forced to think very carefully about the pedagogical issues outlined above and gave us total control in terms of our goal for the activity. The virtual environment is much more powerful visually when compared to an environment using only physical manipulatives. Students are acting on mathematical objects, getting immediate feedback, making observations, and testing conjectures in the virtual environment. When scaffolded with a careful series of questions, students can easily make powerful connections among different representations and between their physical actions and the symbolic world.

Not only teachers, but also students, can become designers of virtual manipulatives. For example, we could ask students to create a similar virtual manipulative for cosine as an extension of this activity. When designing such environments, teachers’ and students’ mathematical and technological knowledge come into play and interact. Moreover, they need to reflect on the presentation and design aspects through which they can discuss what makes an elegant mathematical construction, presentation, and communication.

#### REFERENCES

Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge, MA: The Belknap Press of Harvard University Press.

- Brown, S. E. (2007). *Counting blocks or keyboards? A comparative analysis of concrete versus virtual manipulatives in elementary school mathematics concepts*. (ERIC Documentation Reproduction Service No. ED 499 231).
- Cass, M., Cates, D., Smith, M., & Jackson, C. (2003). Effects of manipulative instruction on the solving of area and perimeter problems by students with learning disabilities. *Learning Disabilities Research & Practice, 18*, 112-120.
- Cavanaugh, C. Gillan, K., Bosnick, J., Hess, M., & Scott, H. (2008). Effectiveness of Interactive online algebra tools. [Electronic version]. *Journal of Educational Computing Research, 38*(1), 67-95
- Cookie, M. (2006). *Sine Curves and Spaghetti*. Retrieved from <http://octm2009spaghetti.pbworks.com/f/spagteac.pdf> [March 2012]
- Moch, P. (2001). Manipulatives work. *The Educational Forum, 66*, 81-87.
- Moyer, P. S. (2001). Are we having fun yet? How teachers use manipulatives to teach mathematics. *Educational Studies in Mathematics, 47*, 175-197.
- Moyer, P. S., & Jones, M. G. (2004). Controlling choice: Teachers, students, and manipulatives in mathematics classrooms. *School Science and Mathematics, 104*(1), 16-31.
- Moyer, P. S., Bolyard, J. J., & Spikell, M. A. (2001). *Virtual manipulatives in the K-12 classroom*. (ERIC Documentation Reproduction Service No. ED 468 203).
- Namukasa, I. K., Stanley, D., & Tuchtie, M. (2009). Virtual manipulative materials in secondary mathematics: A theoretical discussion. *Journal of Computers in Mathematics and Science Teaching, 28*(3), 277-307.
- NCTM (n.d.). *Graphs from the Unit Circle*. Retrieved from <http://illuminations.nctm.org/LessonDetail.aspx?id=L785> [March 2012]
- Olkun, S. (2003). Comparing computer versus concrete manipulatives in learning 2D geometry. *Journal of Computers in Mathematics and Science Teaching, 22*(1), 43-56.
- Özgün-Koca, S. A., & Edwards, T. (2011). Hands-on, Minds-on or Both? A discussion of the development of a mathematics activity by using virtual and physical manipulatives. *Journal of Computers in Mathematics and Science Teaching, 30*(4), 389-402.
- Rosen, D., & Hoffman, J. (2009). Integrating concrete and virtual manipulatives in early childhood mathematics. *Young Children, 64*(3), 26-33.
- Sturdivant, M. (2002). *Sine and Cosine Graphs and Laws of Sines Using Spaghetti*. Retrieved from <http://octm2009spaghetti.pbworks.com/f/Trig+Graphs+and+Laws.pdf> [March 2012]
- Suh, J., & Moyer, P. S. (2007). Developing students' representational fluency using virtual and physical algebra balances. *Journal of Computers in Mathematics and Science Teaching, 26*(2), 155-173.
- Zacharia, Z. C., Olympiou, G., & Papaevripidou, M. (2008). Effects of experimenting with physical and virtual manipulatives on students' conceptual understanding in heat and temperature. *Journal of Research in Science Teaching, 45*(9), 1021-1035.