



10-2018

Using Magic in Computing Education and Outreach

Ronald I. Greenberg

Loyola University Chicago, Rgreen@luc.edu

Dale F. Reed

The University of Illinois at Chicago

Author Manuscript

This is a pre-publication author manuscript of the final, published article.

Recommended Citation

Ronald~I. Greenberg and Dale~F. Reed. Using Magic in Computing Education and Outreach. Proceedings of 2018 IEEE Frontiers in Education Conference.

This Article is brought to you for free and open access by the Faculty Publications at Loyola eCommons. It has been accepted for inclusion in Computer Science: Faculty Publications and Other Works by an authorized administrator of Loyola eCommons. For more information, please contact ecommons@luc.edu.



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License](https://creativecommons.org/licenses/by-nc-nd/3.0/).

Using Magic in Computing Education and Outreach

Ronald I. Greenberg
Department of Computer Science
Loyola University Chicago
Chicago, Illinois 60611–2147
Email: rig@cs.luc.edu

Dale F. Reed
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois 60607-7053
Email: reed@uic.edu

Abstract—This special session explores the use of magic tricks based on computer science ideas; magic tricks help grab students’ attention and can motivate them to invest more deeply in underlying CS concepts. Error detection ideas long used by computer scientists provide a particularly rich basis for working such “magic”, with a CS Unplugged parity check activity being a notable example. Prior work has shown that one can perform much more sophisticated tricks than the relatively well-known CS Unplugged activity, and these tricks can motivate analyses across a wide variety of computer science concepts and are relevant to learning objectives across grade levels from 2nd grade through graduate school. These tricks have piqued the interest of past audiences and have been performed with the aid of online implementations; this conference session will demonstrate enhanced implementations used to illuminate the underlying concepts rather than just to perform the tricks. The audience will participate in puzzling out how to apply relevant concepts as we work through a scaffolded series of tricks centering on error detection and correction. The implementations also provide a useful model for incorporating greater interaction than is typically found in current innovative online interactive textbooks. In addition, they are samples for possible programming assignments that can motivate students using CS Unplugged activities to actively pursue deep programming experiences.

I. SPECIAL SESSION GOALS

This special session will engage the audience in exploring “magic” tricks that can be used to enhance computer science education and outreach at graduate, undergraduate, and pre-college levels. In addition, this session will demonstrate interactive tools that can be used to illuminate the application of computing concepts to performing the magic tricks.

This session should appeal to traditional educators at a wide range of grade levels, outreach professionals, and designers of textbooks and educational tools.

While some of the tricks to be explored can provide jumping off points into fairly deep technical topics, the basic mechanics can be understood by those with a background in basic math. Concrete connections to specific discrete mathematics and computing concepts will be pointed out so that educators can use these fun tricks to reinforce student learning. More generally, these activities can help develop logical thinking and problem solving skills that can be applied in other areas.

II. BACKGROUND

Previous authors have used “magic” tricks to pique student interest and ultimately teach computer science concepts, e.g., [1], [2], [3], [4], [5]. Probably the most well-known

tricks are a number-guessing trick based on binary number representation [6] within the MATHmaniaCS site [7] and an error-detection trick based on simple parity checks [8] that is part of CS Unplugged [9].¹ The CS Unplugged trick is rated as suitable for ages 7 and up, but, with its simplicity, there comes a significant likelihood that a more advanced audience may find it rather easy and obvious. It also has become relatively widely known through use in outreach work and has engendered simple online implementations [8], [11]. (The latter source includes a video demo showing how one can make the trick less obvious, but it encourages misleading the viewer, as does the original CS Unplugged version.)

Recent work [12] describes more advanced tricks inspired by the CS Unplugged error-detection trick that can motivate learning about several additional concepts. One version has been successfully used in many outreach interactions, and strongly positive survey responses have been obtained after presenting six versions as an escalating series of explorations. At the ITiCSE conference and in undergraduate classes, a questionnaire asked for responses on a five-level Likert scale to the statements “I enjoyed this presentation”, “I learned from this presentation”, and, depending whether a student or an educator, “I would like to learn more about the ideas behind this presentation” or “I am likely to use ideas from this presentation in education/outreach work”. All 20 ITiCSE respondents indicated “agree” or “strongly agree” for at least one question as did all but three of the 62 classroom respondents.

While an HTML and JavaScript implementation to perform the six magic tricks online was provided previously [12], we now provide an enhanced implementation illuminating the underlying concepts rather than just serving as a platform for the “all-knowing” presenter to work his magic. This new tool can serve as a prototype for incorporating a greater degree of interactivity than is typical in current interactive digital learning products. While great strides are being made in development of online interactive textbooks with positive outcomes for students [13], [14], and web design learning platforms have been around even longer, e.g., [15], [16], most interaction is of a limited nature. With the exception of running computer programs or HTML code during the teaching of programming or web design, interactivity is typically limited to clicking to run pre-defined animations or checking attempted answers to pre-defined questions. By contrast, the tool described here

¹Each of these tricks may have a longer history, but these references are among those with the longest online trail. See also [10] for some history.

shows how each magic trick is performed based on whatever the magician's subject (a student) may do.

The session description includes an overview of how the tricks are conducted, and shows a sample graphic produced by the illumination tool, but without fully giving away the secrets, which will be unveiled in the special session.

III. NOVELTY

The use of magic tricks is a particularly engaging medium for students in instructional and outreach settings. This session will focus on an escalating series of related magic tricks centering on error detection and correction, scaffolded in a way that allows the audience to build on prior knowledge at each step, reasoning out each successive new wrinkle.

The interactive tool for visualizing the application of underlying concepts to perform the tricks successfully is also a valuable model for designing interactive textbooks. While significant progress is being made in development of interactive textbooks by providers such as zyBooks, the tools to be demonstrated in this session involve a qualitatively higher degree of interactivity. As indicated in Section II, most of the ideas behind these magic tricks have been exposed previously, but the full interactive visualizations are new.

IV. INTERACTION IN SPECIAL SESSION

The session will involve steady interaction of the audience members with the presenter and each other. Audience members will be organized into groups of two or three for in-group discussion and sharing out to the full audience.

A series of up to five magic tricks will be performed, with a similar pattern of interaction for each. First, one of the pre-defined groups will be brought to the presentation computer to serve as volunteer subjects as the presenter performs the trick using a computer implementation. The presenter will repeat the trick a few times to give audience members a chance to figure out how the trick is performed. Each time, the audience will be given an opportunity to engage in group discussion to puzzle out the secret, and the presenter will give hints as necessary. The first group that believes they have found the secret will be invited to perform and explain the trick, with help offered if needed. Finally, the presenter will demonstrate an interactive visualization of the trick that fully exposes the underlying computations, and he may demonstrate at a larger scale that is more difficult to manage mentally.

V. DESCRIPTION OF SPECIAL SESSION

A. Conduct of Tricks and Student Learning Topics

As the series of magic tricks is performed, we will note a number of concepts in discrete mathematics and computer science that are being applied. We will also note directions for deeper analysis of the probabilities of certain events occurring during the tricks and of the computational complexity of performing the tricks most efficiently.

The tricks will generally follow the outline in Figure 1, and access to the demonstrations and the full HTML/JavaScript

- 1) A volunteer is given a grid of tiles, each of which is black or white but can be flipped to the opposite color. The volunteer is allowed to flip tiles to make the grid sufficiently "random" that it would be hard to memorize.
- 2) The magician makes a small adjustment to the grid that effectively organizes it. This adjustment will be specified by the computer for convenience, though it usually could be done manually by the magician.
- 3) The volunteer flips a tile while the magician looks away.
- 4) The magician looks back at the board and announces which tile was flipped.

Fig. 1. The basic steps in conducting the magic tricks discussed here. For each version of the trick, we will indicate how steps are detailed or modified.

source will be available on the Loyola eCommons and more immediately at <http://rig.cs.luc.edu/~rig/fie>.

- Trick Version 1 is the CS Unplugged trick; a 5×5 grid of tiles is presented at Step 1 of Figure 1, and the magician performs Step 2 by adding a sixth row and column.

As audience members work out the secret behind this trick version (or use prior knowledge), they will see it can teach students about the XOR (exclusive or) operator, parity checking, and a common computer science technique of pinpointing a cell of interest in a two-dimensional array by specifying row and column indices.

A sample visualization of how the magician completes the trick for Version 1 is shown in Figure 2 and will be explained in the special session; participants will also see that variations on this trick can be performed at much larger scale and with much more sophisticated features that motivate learning about several other discrete mathematics and computing topics. In all of the more advanced tricks, the magician does not add any tiles to the grid but only flips a small number of them.

- Trick Versions 2a and 2b introduce use of the pigeonhole principle and the generalized pigeonhole principle.

For Version 2a, a manageable scale to start with is an 8×8 grid, and the magician performs Step 2 in Figure 1 by flipping at most three existing tiles. The secret will be explored in the special session, and a culminating visualization will be shown with a 12×12 grid and at most five flips by the magician.

In Version 2b, a small variation allows for a higher ratio of grid size to the maximum number of magician flips.

Note that in each case, the magician flips "at most" a certain number of tiles. Depending on the initial (random) arrangement, the magician may be able to manage with even fewer flips. These two versions of the trick motivate an analysis of the probability distribution of the number of tiles that the magician needs to flip.

- Trick Version 3 introduces the idea of representing numbers using different choices of radix, as well as techniques for counting combinatorial arrangements, particularly counting permutations and, more generally, using the multiplication principle. Here, we will proceed as in Version 2b but with the volunteer making an additional secret flip before Step 2 in Figure 1 and the magician revealing both flips at Step 4. The

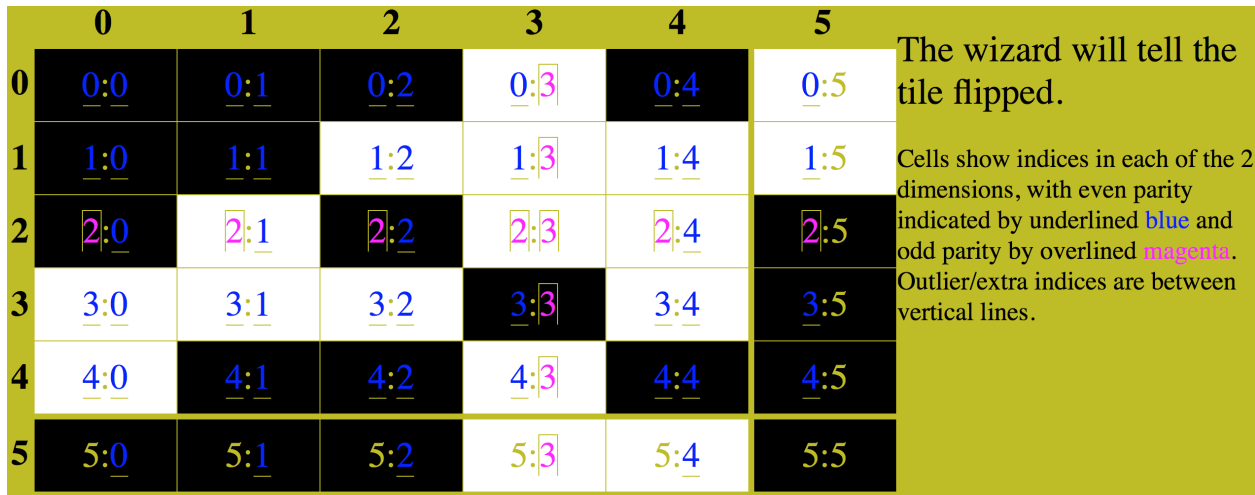


Fig. 2. A sample illuminated run for Version 1 showing the calculations of row and column parities that the magician can use to identify the secret flip at tile (2,3). Note that the color choices accommodate most color vision deficiencies, and the underlining and overlining are redundant but ease interpretation in a grayscale printout.

audience will quickly deduce or be informed early on that, in this version of the trick, there is a “cheating” communication when the computer performs Step 2 (and adds some apparent showmanship at Step 4); in the session, we will work out how the necessary information is being communicated without violating information-theoretic bounds.

- Trick Version 4 introduces a use of the principle of unique multiplicative inverses mod a prime (as is important in various number-theoretic algorithms). Here, we will observe that once we allow a “cheating” communication, we can use just a single organizing flip in an 11×11 grid (or larger) in Step 2 of Figure 1 and simply reverse the order of Steps 2 and 3. There are many ways to choose the organizing flip that constitutes the secret communication, but we will see that the relevant discrete mathematics concept provides a convenient but non-obvious mechanism for conveying the necessary information.

- Trick Version 5 introduces new concepts involving multidimensional representation and correspondences to two-dimensional representations and can motivate additional analyses for advanced students [12], [17].

In this version of the trick, we outlaw “cheating” communications and show that we can still complete the trick in an arbitrarily large grid with just one organizing flip in Step 2 of Figure 1! We will start with an 8×8 grid and show a culminating visualization with a 16×16 grid!

B. Agenda

We intend to work through all or most of the following:

- Introduction; organizing audience into groups.
- Trick Version 1: CS Unplugged (6×6 grid after magician adds last row and column). Culminating visualization 8×8 .
- Trick Version 2:
 - Version 2a: 8×8 grid; magician flips at most 3 tiles. Culminating visualization with 12×12 and at most 5 flips by magician.

- Version 2b: 10×10 grid; magician flips at most 3 tiles. Culminating visualization with 13×13 and at most 4 flips by magician.
- Trick Version 3: 10×10 grid; magician flips 3 tiles and “guesses” two secret tiles (with “cheating” communication).
- Trick Version 4: 11×11 grid with magician flipping 1 tile (with “cheating” communication).
- Trick Version 5: 8×8 ; magician flips 1 tile and no cheating. Culminating visualization with 16×16 and 1 flip by magician.
- Conclusion, audience comments, questions, and suggestions.

C. Outcomes

Audience members will acquire tools they can use in education and outreach activities. There also will be an opportunity at the end of the session for audience members to reflect on the activities and briefly offer any of their own suggestions for variations, other magic tricks that advance computing education and outreach, or potential for similar approaches in other fields. Hopefully, they will be motivated to continue with additional work in a similar vein.

VI. FURTHER WAYS TO ADVANCE STUDENT LEARNING

We noted in Section II that a simple, early error detection trick is included in CS Unplugged [8], [9]. Recent thinking and some research by the CS Unplugged developers and others indicates that to achieve benefit in computer science education, it is important to explicitly link the CS Unplugged activities to traditional computer science activities, e.g., programming. (See discussion with Eileen Kraemer at the conclusion of Tim Bell’s SIGCSE keynote [18].) Moreover, CS Unplugged developers have begun to focus on creating programming exercises motivated by CS Unplugged activities [18].

The work presented in this special session can help educators to form a bridge from error detection magic tricks to computer programming. Student assignments can be created from contemplating the computer implementations used here and

in [12]. We also note that we are starting to see quantifiable positive results from at least one new computing curriculum that makes some use of CS Unplugged activities [19], [20], [21], though these studies have not focused on the specific roll of unplugged activities and suggest that many factors are important to broadening the appeal of CS courses.

It is of particular interest that material provided here can be used as a gentle bridge from unplugged to online activities. One of the original motivations for CS Unplugged was to provide opportunity to students lacking computing resources or electricity, and this motivation remains important even today and in unexpected circumstances (e.g., Puerto Rico post-hurricane, per comments by Joseph Carroll-Miranda at the conclusion of Tim Bell’s SIGCSE keynote [18]). Even on the US mainland, many schools have difficulty finding funds for computers or obtaining the reliable internet access that is desired for some teaching curricula and tools; sometimes internet access is even limited due to policy restrictions to safeguard students from inappropriate content. Computer programming work along the lines used for this special session can powerfully leverage web browser technology and even teach students about web design but without any need for internet connectivity! The implementations used are all programmed using just HTML and JavaScript, and all this work can be done entirely without an internet connection (as long as any web browser is installed).

Another interesting direction that will typically call for more online connectivity during development is an assignment to write a smartphone app to carry out magician activities. Without using the computer to show the illuminations, a human magician probably will not be able to manage Version 5 of the trick for a grid larger than 8×8 . But we could use a smartphone app to photograph the final state of a larger grid and perform the trick (with the smartphone in airplane mode under the control of an audience member to prove that it is not receiving any cheating communication from the computer being used to display the grid). Creating such an app performing the core parity checking operations should be reasonably manageable for even a beginning programmer. An additional interesting challenge for this app is to appropriately recognize the features of the grid in the photo, even though it is hard to take a perfectly straight picture, and to determine the status of each tile.

VII. CONCLUSION

We have demonstrated a powerful interactive tool for visualizing the underlying concepts of several sophisticated error-detection magic tricks. These tricks have been highly engaging for many audiences, and these visualizations provide the potential for a very high degree of interaction in learning about related computer science concepts.

ACKNOWLEDGMENTS

The presenters are supported in part by National Science Foundation grants CNS-1738691, CNS-1543217, CNS-1542971, and CNS-1738515.

REFERENCES

- [1] P. Curzon and P. W. McOwan, “Engaging with computer science through magic shows,” in *Proceedings of the 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 2008, pp. 179–183.
- [2] P. Curzon, P. W. McOwan *et al.*, “The magic of computer science,” <http://www.cs4fun.org/magic>, 2011, accessed March 8, 2016.
- [3] D. D. Garcia and D. Ginat, “Demystifying computing with magic,” in *SIGCSE ’12*. Association for Computing Machinery, 2012, pp. 83–84.
- [4] —, “Demystifying computing with magic, continued,” in *SIGCSE ’13*. Association for Computing Machinery, 2013, pp. 207–208.
- [5] —, “Demystifying computing with magic, part III,” in *SIGCSE ’16*. Association for Computing Machinery, 2016, pp. 158–159.
- [6] C. Heeren, T. Magliery, and L. Pitt, “Binary numbers magic trick,” http://www.mathmaniacs.org/lessons/01-binary/Magic_Trick, 1998, accessed August 31, 2017.
- [7] “MATHmaniaCS - home sweet home,” <http://www.mathmaniacs.org>, accessed August 31, 2017.
- [8] T. Bell, I. Witten, and M. Fellows, “Error detection: Card flip magic,” *CS Unplugged*: <http://csunplugged.org/error-detection>, accessed March 8, 2016.
- [9] T. Bell, I. H. Witten, M. Fellows, R. Adams, J. McKenzie, and S. Jarman, *CS Unplugged: An enrichment and extension programme for primary-aged students*, 2015, http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf.
- [10] T. Bell, F. Rosamond, and N. Casey, “Computer science unplugged and related projects in math and computer science popularization,” in *The Multivariate Algorithmic Revolution and Beyond: Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, ser. Lecture Notes in Computer Science, H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, Eds. Springer-Verlag, 2012, vol. 7370, pp. 398–456.
- [11] D. F. Reed, “Memory trainer,” <https://www.youtube.com/watch?v=67iTJOBw5RA>, Feb. 2010.
- [12] R. I. Greenberg, “Educational magic tricks based on error-detection schemes,” in *Proceedings of the 22nd Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Jul. 2017, pp. 170–175, see also author version with implementation notes and full implementation at http://ecommons.luc.edu/cs_facpubs/149.
- [13] zyBooks, “Press releases archives - zyBooks,” <https://www.zybooks.com/category/press-release>, 2017, accessed August 31, 2017.
- [14] —, “Research - zyBooks,” <https://www.zybooks.com/research>, 2017, accessed August 31, 2017.
- [15] Refsnes Data, “About W3Schools,” <https://w3schools.com>, 2017.
- [16] —, “W3Schools online web tutorials,” <https://w3schools.com>, 2017.
- [17] R. I. Greenberg, “Mentally performing error correction tricks and mapping higher-dimensional cubes to two dimensions,” Manuscript, Jul. 2018.
- [18] T. Bell, “What’s the big idea with CS education in K–12,” Keynote presentation at SIGCSE ’18 available at <https://www.youtube.com/watch?v=VYBbWN-1sAU>, Feb. 2018.
- [19] L. Dettori, R. I. Greenberg, S. McGee, and D. Reed, “The impact of the Exploring Computer Science instructional model in Chicago Public Schools,” *Computing in Science & Engineering (Special Issue: Best of RESPECT 2015)*, vol. 18, no. 2, pp. 10–17, March/April 2016, <https://doi.org/10.1109/MCSE.2016.39>.
- [20] S. McGee, R. McGee-Tekula, J. Duck, R. I. Greenberg, L. Dettori, D. F. Reed, B. Wilkerson, D. Yanek, A. M. Rasmussen, and G. Chapman, “Does a taste of computing increase computer science enrollment?” *Computing in Science & Engineering (Special Issue: Best of RESPECT 2016)*, vol. 19, no. 3, pp. 8–18, April 2017, <https://doi.org/10.1109/MCSE.2017.50>.
- [21] S. McGee, R. McGee-Tekula, J. Duck, C. McGee, L. Dettori, R. I. Greenberg, E. Snow, D. Rutstein, D. Reed, B. Wilkerson, D. Yanek, A. Rasmussen, and D. Brylow, “Equal outcomes 4 all: A study of student learning in ECS,” in *SIGCSE ’18*. Association for Computing Machinery, 2018, pp. 50–55, <https://doi.org/10.1145/3159450.3159529>.