

Reflection: “Twenty Things to Do With a Computer” Revisited

[Glen Bull](#), N. Rich Nguyen, Jo Watts, & Rachel Gibson
University of Virginia

Michael Littman
Princeton University

In 1971 Seymour Papert and Cynthia Solomon published “Twenty Things to Do With a Computer” (MIT Artificial Intelligence Memo No. 248 / Logo Memo No. 3). To celebrate the 50th anniversary of this influential paper, we are republishing it in this issue of the *CITE Journal*. To accompany this republished paper, we are also including a reflection by Brian Harvey and Jens Möenig, the inventors of the educational programming language Snap!, regarding ways in which educational computing is both the same and different from activities of a half century ago, and an essay by Gary Stager describing a book of readings inspired by the original paper.

We are also pleased to publish an updated article by Cynthia Solomon, one of the authors of the original article. The article, “(More Than) Twenty Things to Do With TurtleStitch,” describes current work of Cynthia Solomon and her coauthor, Susan Klimczak, with computer-controlled embroidery machines. The original “Twenty Things” paper suggested a number of activities in which the computer interacted with the physical world. The current work with embroidered computer designs is very much in that tradition.

In an interview with Terry Gross on the public radio show *Fresh Air*, William Gibson famously commented that the future is here – it just is not evenly distributed. This insight was stimulated by the transition from an urban environment to a rural environment when he was a boy. It is certainly the case that when Seymour Papert and Cynthia Solomon began collaborating on development of the educational language Logo, together with the computer scientist Wallace Feurzeig, they had access to computational facilities in the M.I.T. Artificial Intelligence Laboratory and at Bolt, Beranek, and Neuman that would not be widely available in public schools for decades.

The paper that Papert and Solomon wrote was stimulated by their work with Logo in schools in the greater Boston area. Some of the activities were based on actual activities of children in schools, while other suggested activities would be challenging to implement even today. As computers became more widely available in schools, their work influenced a generation of educational researchers.

Gary Stager has collected reflections from more than 50 of these educators in a book of readings, *Twenty Things to Do With a Computer: Forward 50* (Stager, 2021). In the foreword to this book, Cynthia Solomon noted,

Among our collaborators in this endeavor were Marvin Minsky (a founder of the field of Artificial Intelligence) and a slew of MIT AI Lab hackers – students and staff. This collaboration started in the 1960s and continued into the 21st century, and was joined by a coterie of interesting teachers, learners, and researchers from all over the world.

A decade after the original “Twenty Things” paper was published, Robert Taylor (1980) published a book of readings about ways in which computers might be used in schools. *The Computer in the School: Tutor, Tool, Tutee* suggested that the computer might be used as a tutor, as a tool, or as a *tutee*, in which a child learned by teaching the computer (through programming). Of these three imagined uses, use of the computer as a tool by far exceeds the other two uses today. Early computer users had to write their own code. However, tools such as Photoshop for image editing, Ableton for creating music, and MATLAB for mathematical modeling were soon developed. Today the term *computer application* has been shortened to *app*. Countless applications are now available in app stores, giving rise to the popular phrase, “There’s an app for that!”

Papert (1980) published expanded descriptions of many of the activities introduced in the “Twenty Things” paper in his book, *Mindstorms: Children, Computers, and Powerful Ideas*. In the foreword to *Mindstorms*, Papert recounted the way in which he used the gears of farm equipment as models for exploring mathematical concepts. Papert viewed the computer as a more malleable tool for creating “objects to think with” of this kind.

Papert introduced the term “computational thinking” in *Mindstorms*. He describes it as a method of using a computer to explore a problem. Jeanette Wing (2006) later popularized and expanded use of the term to encompass solving problems by employing the fundamental concepts of computer science. Today two of the primary reasons advanced for learning to code are (a) workforce development, acquisition of a skill that may lead to remunerative employment, and (b) underlying computational concepts learned in the process may be useful in problem solving generally.

However, if the list of twenty things to do with a computer is revisited, another reason for learning to code emerges. In the examples described by Papert, the personal engagement and enjoyment experienced by children who create poetry, art, and music is evident. The success of a school orchestra is measured not by the number of students who go on to become

professional musicians but by the extent to which music enriches their lives while in school and beyond.

These activities are undertaken for personal enjoyment rather than because they are required for work or school. Many of the educators and researchers who participated in the first wave of educational computing are now retired. Yet, many of these individuals continue to use the tools that they developed for personal enrichment after retirement.

Cynthia Solomon noted,

... After 50 years of playing with turtles, I am deeply absorbed with a turtle that makes designs for computerized embroidery machines. I use TurtleStitch, an extension of the Snap! programming language. TurtleStitch follows the conventions of Snap! with a few additional features specifically for embroidery. (Stager, 2021, p. i)

Snap! is a computationally advanced computing language inspired by Scratch, itself a web-based language inspired by Logo. Brian Harvey stated, “Languages in the Logo family, including Scratch and Snap!, take the position that we’re not in the business of training professional computer programmers. Our mission is to bring programming to the masses.”

This mission statement implies that some people will use Snap! for coding voluntarily and not because they are required to do so. A further corollary is that Snap! is not designed solely for children or for students, but for a broader community. (In contrast, Scratch is very much designed for children.) This fact is reflected in the widely used Snap! forum, which has participants from almost every age and decade of life, from the early teens to participants in their eighties and nineties. Through the Snap! forum, an inclusive community of users are able to connect, create, and innovate, regardless of their gender, ethnicity, or coding experience.

The word *technology* has roots in the Greek word *tekhnē* referring to “art, skill, or craft” (as in weaving or fabricating). Today technology is often associated with STEM applications (science, technology, engineering, and mathematics) and development of workforce skills. However, *techne* was originally also associated with the arts as well (hence, the acronym STEAM).

In this regard we highlight another application of technology separate from STEM and workforce development – uses of technology designed to amuse and delight us. We refer to these uses as *tehkne amuse* (pronounced “tek-nay amuse-say”) – applications of technology that engage and delight us separate from any higher purpose other than personal enrichment.

Terms such as “casual computing,” “creative computing,” and “recreational computing” have also been used in various ways to describe these uses of technology and computing. However, each of these terms has an extensive history and considerable baggage associated with it. Therefore, in the interest of a term that is neutral without prior

associations, we have adopted the term *tekhne amuséto* to describe such uses in the examples that follow.

This approach seems consistent with the spirit of the original Papers and Solomon (1971) paper. In a subsequent expansion of the activities suggested in this paper, Papert introduced the term *computational thinking* in relation to use of a computer for this type of exploration. However, this term has also been appropriated for applications that are tightly coupled to workforce development.

With that in mind, here are twenty examples of *tekhne amusé* – uses of technology designed to amuse and delight. Many of these activities were originally piloted in the Charlottesville Laboratory School (Lab School), a collaboration between the Charlottesville City Schools and the University of Virginia. In some cases, these projects were undertaken in collaboration with the Smithsonian Institution. These activities were subsequently incorporated into two computing courses that we have developed: Designing Digital Art and Music and An Introduction to Design through Making. The design through making activities are organized around the three themes of Sight, Sound, and Motion. The art and music course is organized around the parallel themes of art (sight or vision), music (sound), and animation (motion). We also have included several additional activities by colleagues whom we admire.

Sight and Vision - Art

1. Create Art - The images in Figure 1 are examples of abstract artwork from Mark Rothko and Jackson Pollock, as well as examples of art created in Snap! and inspired by the style of the two painters.

Figure 1. *Art in the Style of the Artist Rothko*



The image on the left in Figure 2 is a photograph of a window box with flowers. The image on the right is an impressionist painting of the flowerbox created through a technique known as pointillism. A Snap! program can create a painting by placing tiny dots on the canvas to create an impression of the original painting.

Figure 2. Art Emulating a Pointillist Painting



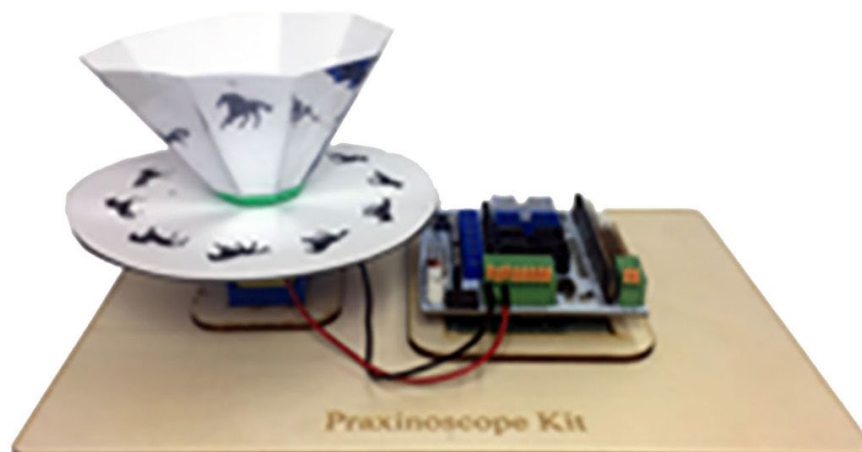
2. Create a Digital Animation - The photographer Muybridge projected a series of animated images before a paying audience for the first time at the Chicago World's Fair in 1893. The original images that Muybridge captured can be reanimated with a computer – or a series of modern-day images can be drawn or captured and animated (Figure 3).

Figure 3. Muybridge Animated Images



3. Create a Movie Machine - The same images can be brought life with a mechanical movie machine, also known as a *praxinoscope*. This machine can be created with cardstock and a Silhouette die cutting machine. A microcontroller and a motor are used to make the movie machine spin (Figure 4).

Figure 4. Reconstruction of a Praxinoscope



4. Make a Movie Projector - Combine an animation machine (described in the previous activity) with a magic lantern to create a movie projector (Figure 5). In 1893 Muybridge presented moving images projected on a screen at the 1893 Chicago's World's fair.

Figure 5. *Reconstruction of a Moving Image Projector*



This was the first time that an audience paid to see moving pictures. Muybridge subsequently met with Edison, who was intrigued by the possibilities, and subsequently invented the kinetoscope, forerunner of today's motion pictures.

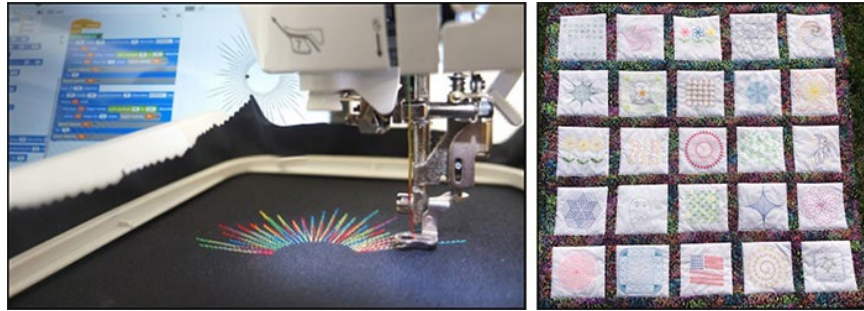
5. Use Machine Learning to Create Art - Today, machine learning is employed in countless technologies and software. In this example, a machine learning method called *neural style transfer* takes two images – a content image of a peacock and a style image such as an artwork by Van Gogh – and fuse them together to create an output image that looks like the content image, but painted in a pattern of the style image (Figure 6). Neural style transfer is a fun technique that demonstrates the capabilities and internal representation of a neural network, which may generate interest and curiosity to gain a better understanding of how machine learning works.

Figure 6. *Emulation of Art through Machine Learning*



6. Design an Embroidered Quilt - Andrea Mayr-Stalder, Cynthia Solomon, and Susan Klimczak have established a community engaged by computer-controlled embroidery using TurtleStitch, an extension of Snap!. Members of the TurtleStitch community are creating projects like the quilt in Figure 7 designed by Elaine Wolfe (<https://papercraftetc.blogspot.com>).

Figure 7. *An Embroidered Quilt*



7. Make a Computer-Controlled Light Box - In addition, computer-embroidered material, such as a translucent screen, can be further augmented by adding sensors and computer-controlled LEDs, creating a computer-controlled light box (Figure 8).

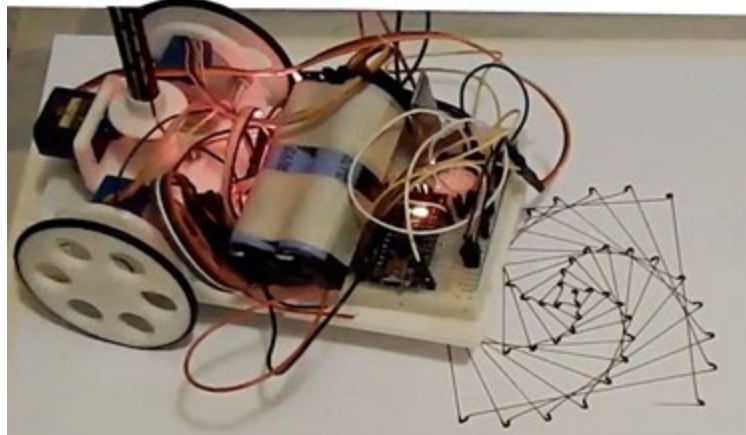
Figure 8. *A Computer-Controlled Light Box*



Sound and Music

8. Compose a Tune - The original “Twenty Things” paper described Logo music extensions. We have created an extension of Snap!, TuneScope, that is a code-based Digital Audio Workstation (DAW). Students have used TuneScope to good effect in creation of original music (Figure 9).

Figure 9. *TuneScope: Creating Music Through Coding*



9. Design a High-Fidelity Speaker - In a related activity, students design and fabricate speakers that can be used to play the music that they create (Figure 10).

Figure 10. *A Speaker Designed and Fabricated With a Computer*



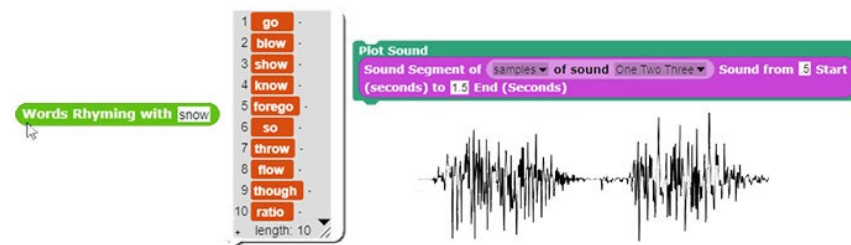
10. Design a Sound Mixer - TuneScope can be used to adjust the volume, tone, and acoustic properties of sounds and music. This can help a student enhance their original compositions by allowing them to balance their music tracks, allowing some instruments to be in the foreground and others in the background (Figure 11). This is similar to what occurs in a recording studio.

Figure 11. A Sound Mixer Created in TuneScope



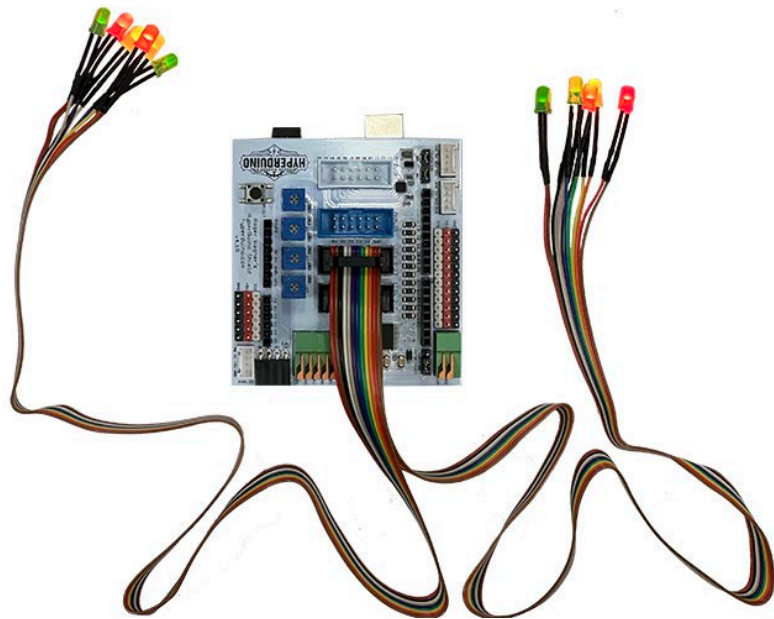
11. Add Lyrics to a Song - After creating a song, TuneScope can be used to generate and add lyrics. There are many online dictionaries that can be accessed to generate lists of rhyming words, which can be implemented in TuneScope to aid in the creation of lyrics. Subsequently, these lyrics can be recorded and edited using a Snap!-based audio editor (Figure 12).

Figure 12. TuneScope Sound Editor



12. Create a Digital Light Show With LEDs - A digital light show synchronized to music is an extension of a music box machine. This can be created using an Arduino and LEDs. Snap4Arduino, an extension of Snap!, can be used to develop a procedure to create the music and control the sequencing of LEDs (Figure 13).

Figure 13. *Microcontroller and Light Emitting Diodes (LEDs)*



13. Create a Mechanical Drum Machine - The girls in the Awesome Blossoms club used their reconstruction of Charles Page's electromagnetic engine to create a mechanical drum machine. A solenoid can be wrapped using magnet wire, then mounted to a platform to be positioned in a variety of ways to hit objects, such as tambourines and bowls, to create a beat pattern. The mechanical drum machine can be played along with as well (Figure 14).

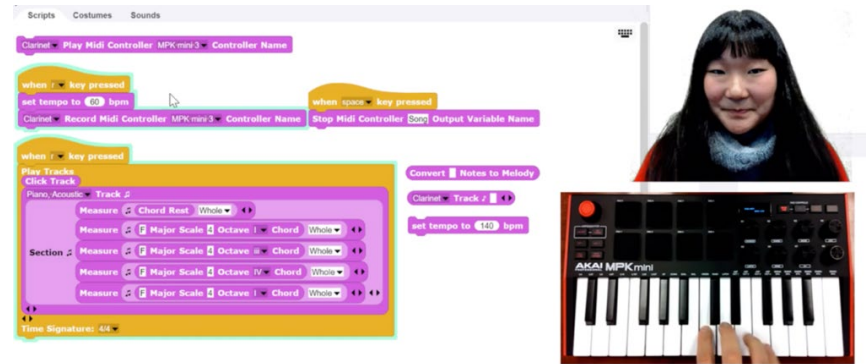
Figure 14. *A Computer-Controlled Drum Machine*



14. Capture Input from a MIDI Controller - Notes played on a MIDI controller like the one show below can be captured in TuneScope, edited,

and played back. The MIDI signal can also be passed through TuneScope to a speaker, allowing the user to play along with prerecorded music (Figure 15).

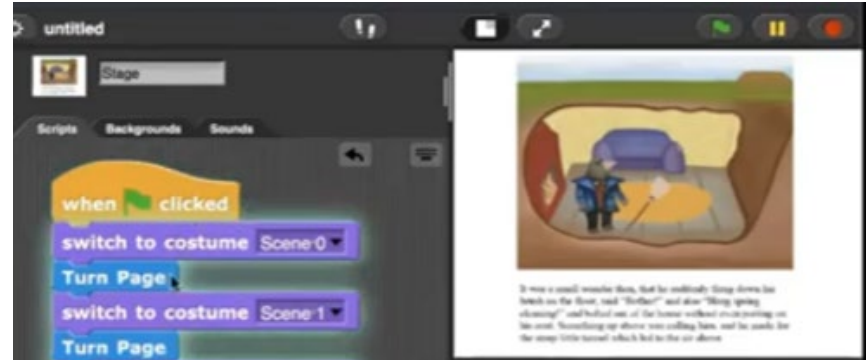
Figure 15. *Capturing Notes From a MIDI Keyboard*



Animation and Motion

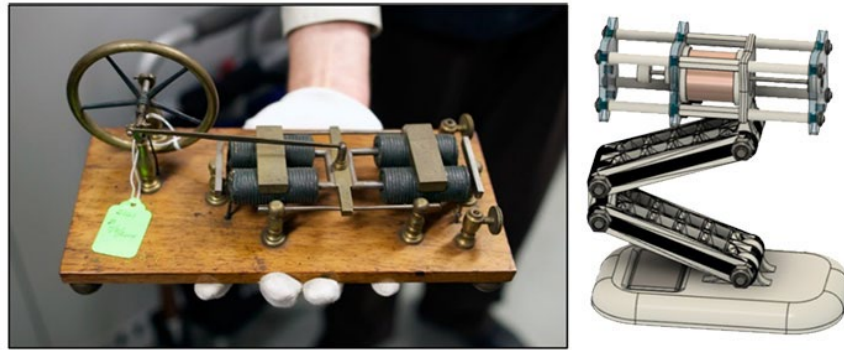
15. Create an Interactive Digital Storybook - Animated images lead naturally to creation of an interactive digital storybook. Addition of speech synthesis or recorded speech can be used to create a talking storybook (Figure 16).

Figure 16. *An Interactive Storybook*



16. Reconstruct a Historic Invention - The invention on the left in Figure 17 is Charles Page's "electromagnetic engine" patented in 1854. The machine on the right is a modern-day reconstruction of this historic invention.

Figure 17. *Reconstruction of a Linear Motor*



17. Use Rotating Magnets to Create a Drum Pattern - As the rotating magnets move past a sensor, the linear motor is activated to create a drum pattern based on the positions of the magnets (Figure 18).

Figure 18. *Creating a Drum Pattern With a Rotating Disk*



18. Synchronize an Animation to Music - Digital animations can be enhanced by synchronizing movements with music. In this application, shapes drift across the screen as music plays in time with the pulsing shapes (Figure 19).

Figure 19. *Synchronized Animations and Music*



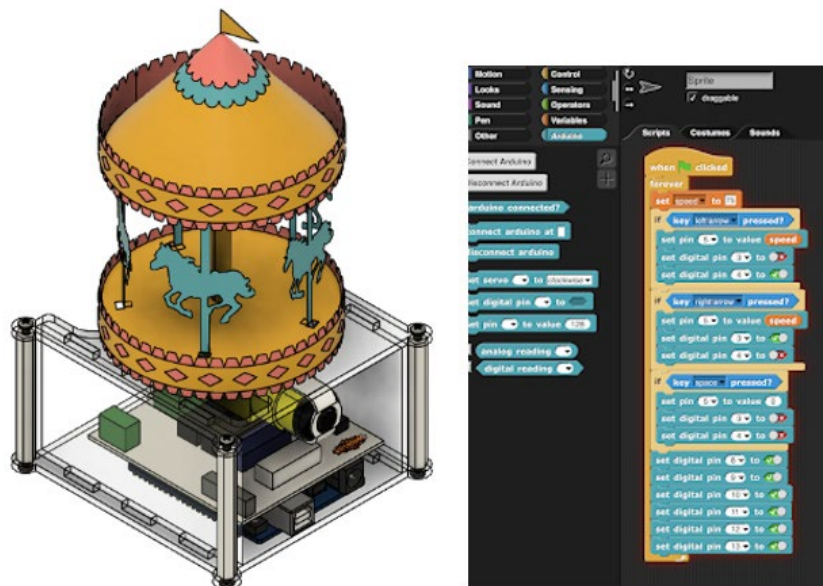
19. Music Visualizer - A music visualizer can be created in Snap! by using a computer's microphone to capture audio samples and plot the spectrum of the sound.

Figure 20. *A Music Visualizer*



20. Create a Computer-Controlled Carousel - Different types of carousel designs can be created on the computer, and then cut and constructed from cardstock using a die cutting machine (Figure 21).

Figure 21. *A Computer-Controlled Carousel*



The design in this illustration was developed by Elaine Wolfe, a talented papercraft artist. The carousel is turned by a motor controlled by a microcontroller using an extension of Snap!, Snap4Arduino. The Hyperduino shield for the microcontroller was developed by Roger Wagner, who has the distinction of writing the first programming manual for the Apple II computer. The enclosure for the microcontroller was developed by Henry Forsythe, who originally began working with us in seventh grade and today is completing engineering school. One of the recurring characteristics of many of these projects is that they were undertaken by individuals working across many disciplines in collaborations that span generations.

Summary

The activities described here lie at the intersection of making and coding. These types of projects do not require guardrails that are necessary for professional applications of code. A program written to control a radiation machine can harm patients if it is not written properly. A program written to create art or music for personal enjoyment does not need similar safeguards. Consequently, many of the concepts related to secure program development highlighted in computer science standards can be bypassed.

These are endeavors worthy of pursuing on their own merits. However, because these activities can be accessible and enjoyable, it may be useful in broadening participation in computing. The field of professionals who enter computing is noticeably less diverse than society as a whole. As a result, decisions that affect everyone's lives are made by a group that is not reflective of society as a whole.

Diffusion of innovation theory has established that a lower threshold increases the chances that someone will try a new idea or method. Enrollment in an initial computer science course often has a high threshold. An introduction to computing in the context of recreational computing can lower the threshold.

This approach to computing can make computing concepts more accessible. The students who do not go on to professional careers in computing may still find that they enjoy coding throughout their lives. Because these students will have more insight and understanding of the operations that underlie computing, they may also be better consumers of code written by others for them.

The original "Twenty Things to Do with a Computer" paper, written 50 years ago, was remarkably prescient. Rereading these activities recalls the joy and whimsy inherent in projects like creation of computer-controlled puppets, juggling, dance, art, music, and games. Retention of the joy and engagement modern-day versions of these activities is a worthwhile goal.

Author Note

Glen Bull is a professor of education at the University of Virginia. N. Rich Nguyen is an assistant professor of computer science at the University of Virginia. Jo Watts is the manager of the Make to Learn Laboratory at the

University of Virginia. Rachel Gibson is a research assistant in the Make to Learn Laboratory. Michael Littman is a professor of engineering at Princeton University.

References

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Papert, S. & Solomon, C. (1972). Twenty things to do with a computer. *Educational Technology*, 12(4), 9-18. <https://www.learntechlib.org/p/162670/>

Stager, G. (2021). *Twenty things to do with a computer: Forward 50*. Constructing Modern Knowledge Press.

Taylor, R. (1980). *The computer in the school: Tutor, tool, tutee*. Teachers College Press.

Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49, 33-35. <https://doi.org/10.1145/1118178.1118215>

Contemporary Issues in Technology and Teacher Education is an online journal. All text, tables, and figures in the print version of this article are exact representations of the original. However, the original article may also include video and audio files, which can be accessed online at <http://www.citejournal.org>