



Programming in an fMRI Scanner: A Report from the Field

Steven Tanimoto
Paul Allen Sch. of Comp. Sci. & Eng.
University of Washington
Seattle, Washington, USA
tanimoto@uw.edu

Rob Thompson
Dept. of Computer Science
Colorado School of Mines
Golden, CO, USA
rthompson@mines.edu

Todd Richards
(Prof. Emeritus) Dept. of Radiology
University of Washington
Seattle, Washington, USA
toddr@uw.edu

Cheri Yates
(Retired Pediatric Neuropsychologist)
Seattle Children's Hospital
Seattle, Washington, USA
cheri.yates@gmail.com

Virginia Berninger
(Professor Emerita) Dept. Educ. Psych.
University of Washington
Seattle, Washington, USA
vwb@uw.edu

ABSTRACT

A preliminary study is reported in which six human subjects were scanned while performing a simple form of computer programming. Functional magnetic resonance imaging typically precludes the use of ordinary computing equipment due to safety concerns when items with metallic parts are in the scanning room. We used a special visual programming environment, “Kokopelli’s World,” for which an optics-only trackball was sufficient. This research methodology is described. Aspects of the experience of programming, by one of the authors, in the fMRI scanner are reported. An analysis of the data from the scanning is presented, and brain areas where activity for coding exceeded activity for a control task were identified and are presented here. These observations suggest hypotheses and directions for future research. One hypothesis is that particular aspects of programming are inherently and measurably pleasurable.

CCS CONCEPTS

• **Human-centered computing** → Pointing devices; *User studies*; *Field studies*; • **Software and its engineering** → *Software development techniques*; • **Social and professional topics** → **Computational thinking**; • **General and reference** → **Experimentation**;

KEYWORDS

programming, fMRI, coding in scanner, visual language, trackball, brain imaging, brain region, Kokopelli’s World, programming experience, scientific poem

ACM Reference Format:

Steven Tanimoto, Rob Thompson, Todd Richards, Cheri Yates, and Virginia Berninger. 2022. Programming in an fMRI Scanner: A Report from the Field. In *Companion Proceedings of the 6th International Conference on the Art, Science, and Engineering of Programming (‘Programming’ 22 Companion)*, March 21–25, 2022, Porto, Portugal. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3532512.3535220>



This work is licensed under a [Creative Commons Attribution International 4.0 License](https://creativecommons.org/licenses/by/4.0/).

‘Programming’ 22 Companion, March 21–25, 2022, Porto, Portugal
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9656-1/22/03.
<https://doi.org/10.1145/3532512.3535220>

1 INTRODUCTION

The question “What makes programmers feel the way they do when they’re programming?” posed by the PX/22 workshop organizers, seeks more understanding about the cognitive and affective processes involved in computer programming. Psychologists and neuroscientists have been studying brain processes involved in computer programming as an important subarea of studying cognition. Understanding how the brain functions during programming may lead to improvements in the teaching of programming, the tooling provided to programmers or better appreciation of an important kind of creative activity.

Functional magnetic resonance imaging (fMRI) is a relatively recent means to localize, within the human brain, where cognition is actively happening at a particular time. By engaging a person in a task, sequences of 3D digital images can be made in which high-levels of brain activity, attributable to the task, are marked with special colors.

A number of studies have been done on code comprehension in a scanner, including the early one by Siegmund et al [10]. Subsequent comprehension studies include that by Floyd et al [3] and Ivanova et al [5]. These studies have identified areas of the brain that are active when a person is reading and analyzing computer source code.

However, to date, few studies have been performed in which a human is scanned with fMRI while actively constructing a computer program rather than reading or thinking about it. A major reason for the paucity of such research involves the question of physical safety of having traditional computing equipment in the same room as the scanner. The strong and changing magnetic fields created during the scanning process can cause metallic objects to be violently moved. Another difficulty is that the body motions associated with speaking or even operating keyboards can lead to inaccuracies in the imaging.

One notable study used a special keyboard in which most metal was eliminated, but special shielded metal wires carried signals to the scanner control room[7]. In this 2020 study, brain activity arising in performing coding was compared with brain activity arising when writing English words and sentences. Coding was found to involve primarily the right-brain hemisphere, whereas the prose writing involved primarily the left hemisphere.

Here, our paper reports on a proof-of-concept study performed in the summer of 2016 (near the end of a funded project addressing specific learning disabilities in written language) for a new research methodology involving fMRI brain scanning of people actively creating (not just reading or analyzing) computer programs. As far as we know, no earlier studies involved any sort of active computer programming in an fMRI scanner. Data collected from our study was limited, because the grant was ending, and also because of personnel transitions. Nonetheless, this experience report may be of interest to other researchers on the psychology of programming.

This paper presents the unique methodology that was developed in that study, and it offers a first-hand account of the experience of the first author performing simple program-writing tasks while being scanned in an fMRI scanner. Brain activity data for computer coding, compared with a drawing activity, are presented based on scans of children actively performing the tasks. Finally, the paper discusses the possible consequences of fMRI studies of active programming activity on improving the experience of programming for students, end-user programmers, and professional software developers in the future.

2 EXPERIMENTAL SETUP

In most respects the equipment we used for this study was standard for scanning. However, some of the computing environment was particular to this study, and we describe that below.

Functional MRI was performed on a Philips Medical Systems 3T scanner (version 5.1.7) located at the University of Washington Department of Radiology Diagnostic Imaging Center with the following scan parameters: echo-planar pulse sequence, repetition time 2000 milliseconds, echo time 25 milliseconds, acquisition matrix 80x80x33, field of view (ap, fh, rl) [millimeters], 240.000 132.000 240.000, EPI factor 35, total number of dynamics 507, scan duration 1030 seconds (i.e., 17 minutes, 10 seconds). It is illustrated in Fig. 1.

Loudspeakers near the human subject's ears and a microphone near the mouth allowed spoken communication between the experiment staff (who are in the control room) and the subject (who is in the scanner, in the scanner room).

The computer that hosted the experimental programming tasks and control tasks was a normal Windows PC, and it was housed in the control room of the scanner.

The computer display was arranged in a special layout so that the human subject could lie supine and look straight up to a mirror that reflected the computer's display screen. The actual display device was also in the control room.

The computer input device was a special trackball, designed and manufactured to be safe within the magnetic fields of fMRI scanners[4]. We used the Current Designs Trackball 2, illustrated in Fig. 2. The trackball was positioned over the abdomen of the human subject during each trial, so that it could easily be operated by the subject, and far enough from the head so that the hand movements involved in operating the trackball would not cause any significant head movements that might interfere with the obtaining of sharp



Figure 1: The functional magnetic resonance imaging scanner at the University of Washington Medical Center.



Figure 2: The Trackball 2 model from Current Designs is MRI safe, and constructed without metal. The connecting cable shown is a fiber-optic cable with 12 fibers. Not shown is a special interface that connects the fiber cable to a standard USB cable.

images of the brain. The trackball was connected to the computer by a special fiber-optic cable: first, from the trackball to a special interface located in the control room, and second, via a standard wired cable from the special interface to a USB port on the computer. The fiber-optic cable contains no metal that might either interfere with the scanner or be affected by it.

The programming environment for the tasks was a specially developed blocks-oriented language aimed at a population of students

¹Additional information about the scanner can be found in the compliance document linked from <https://www.usa.philips.com/healthcare/resources/support-documentation/dicom-magnetic-resonance-imaging>

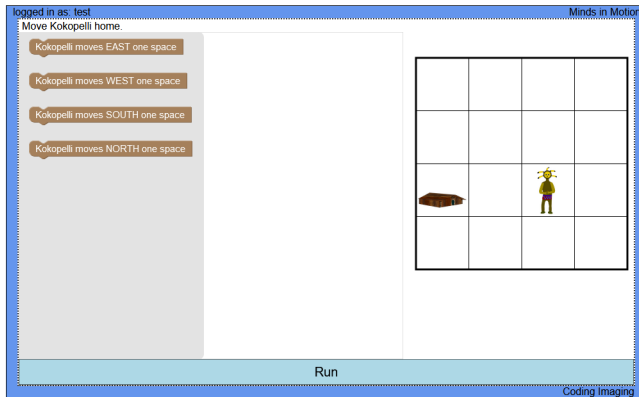


Figure 3: A screen shot of the programming environment called Kokopelli's World used in the study. A set of blocks is given on the left part of the screen. These are to be placed in the programming space in the middle of the screen (blank at the beginning). A 2D rendering of the microworld is shown on the right.

that included both normally developing and learning-disabled students. The programming environment is called “Kokopelli's World.” It was created by the second author (Thompson) for use by the research project led by author Berninger. The environment includes a simple 2D microworld, and both the programming space and the microworld can be seen in the screenshot in Fig. 3. Parts of Kokopelli's World were described in relation to the integrated teaching of reading, writing, and coding[12] and in relation to teaching students with specific learning disabilities in reading or writing[13]. The programming language in Kokopelli's World offers variable-free constructs for imperative statements, conditionals, looping, and user input, somewhat similar to those in Pattis' Karel the Robot[8] but without user-defined functions. Details about the environment can be found in Thompson's Ph.D. dissertation[11]. This environment is also described further in the next section.

3 SCANNING AND COGNITIVE ACTIVITY SCHEDULE

3.1 Overview

Here we describe the timing and sequencing of the cognitive activities that human subjects were instructed to perform while being scanned. There is a mix of programming activities, resting activity, and control (connect-the-dots) activity, arranged to work in a within-subjects experiment design. The resting activities are included to help separate the periods of programming or control activity from one another and to provide contrasts within the image sequence so that activities and their corresponding images could be reliably matched.

This study called for a set of students to be scanned while performing the intended activities. Prior to that, the first author served as a test subject (let us call him Subject Zero or S_0) in order to make sure that the planned activity schedule would work.

In total, six subjects were scanned. Two were adults and four were children. Each child's activity began in a mock scanner in a

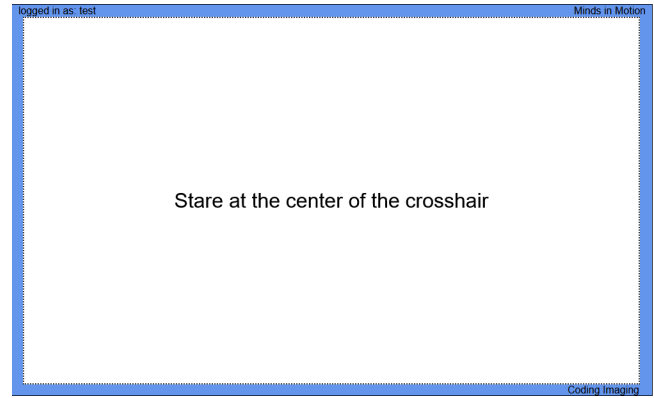


Figure 4: Written instructions to the human subject for the cross-hair activity. This activity was important in preparing the subject to stay still, stay calm, clear the brain of thoughts, and rest the brain.

special room set up to welcome juvenile subjects and get them to be comfortable with the idea of lying down in a scanner.

Subject S_0 skipped the mock-scanner phase of the activity and began with the actual in-scanner portion of the schedule. The first scan was conducted on July 8, 2016 with S_0 and proceeded according to plan without problems. The remaining subjects were scanned between the months of August and October of that year.

Each subject completed four cycles of activity and was scanned for approximately 17 minutes. This time was allocated as described in the following subsections.

3.2 Activity Setup Phase

Human Subject Preparation: Student is welcomed and taken to the mock scanner room and acquainted with the idea of doing activities inside a scanner.

Setup: Coauthor Richards sets up the actual scanner, supervises the positioning of the human subject in the scanner, and launches the computer program written by coauthor Thompson that drives the cognitive activities. Scanning begins, and the special computer program displays the instructions for the baseline rest task to the human subject.

Baseline Rest Task: Instructions for the cross-hairs task are displayed (see Fig. 4), and the subject fixates on the cross-hairs on the screen (see Fig. 5) for 18 seconds. This task is included because it facilitates calibration of the imaging data, and because it helps prepare the human subject to stay calm during the subsequent coding and control tasks.

3.3 Cycle 1 Schedule

Line Drawing: Subject engages in the control task (“connect-the-dots”) for 2 minutes. Connecting pairs of dots with straight line segments served as the control activity for this study. The actual drawing is facilitated by using a standard “rubber-band” interaction style. The activity shares some features with the programming activities (the use of the trackball and the associated sensory-motor activity), yet allows a distinction from the programming activity

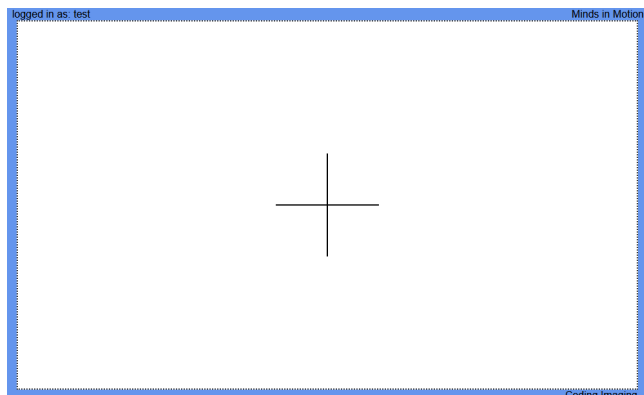


Figure 5: The actual cross-hair image shown to the subject. This image is minimal and to the point of helping the subject have a “blank” state of mind prior to the next, more demanding, cognitive activity.

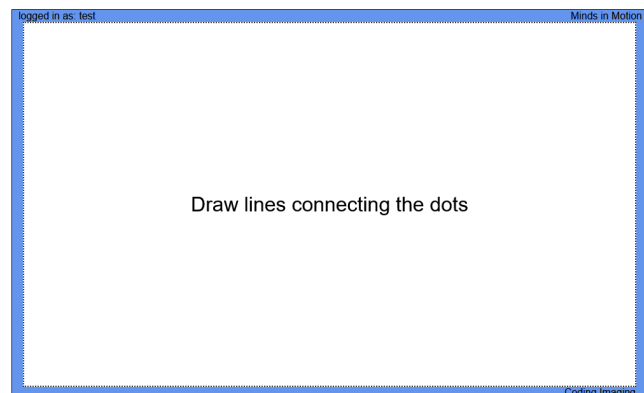


Figure 6: Instructions for the line-drawing (connect-the-dots) activity.

that would involve planning and/or logic. The displayed instructions for the connect-the-dots activity are shown in Fig. 6), and an example of what the screen looks like while drawing a line is shown in Fig. 7.

Simplest Coding: Subject engages with two (easy) levels of difficulty in sequencing instructions to the computer. Two minutes are allocated for the 10 short trials:

- 1-cell-away:** (4 versions). Kokopelli is only one cell away from his destination, and a program consisting of a single block is sufficient to solve the problem.
- 2-cells-away:** (8 versions). Kokopelli is two cells away from home. Two program blocks are required.

Instructions for the simple programming tasks can be seen in Fig. 8. After the problem has been solved, the screen looks like what’s shown in Fig. 9, with Kokopelli in the same square as the longhouse.

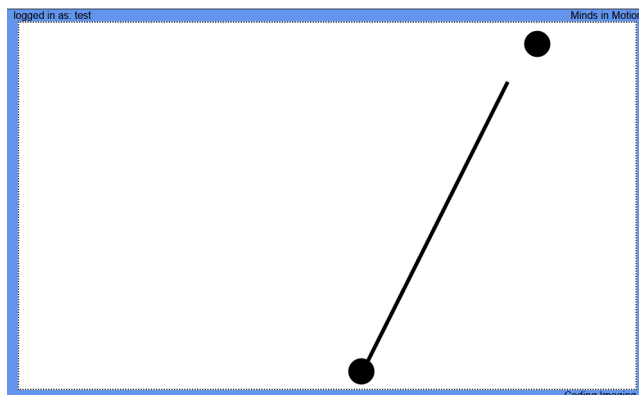


Figure 7: A screen shot taken during one of the line-drawing (connect-the-dots) activity instances.

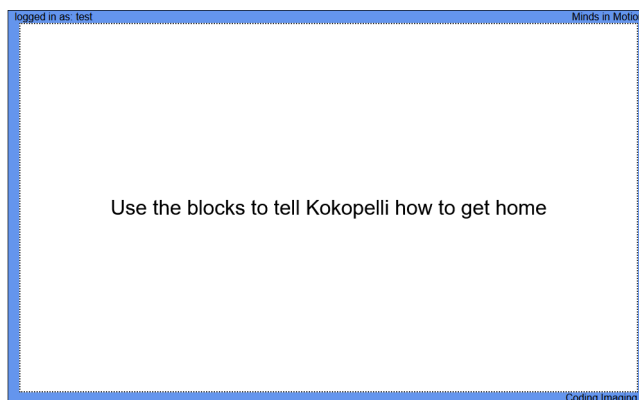


Figure 8: Instructions for the basic control-flow programming exercise.

3.4 Cycle 2 Schedule

Line Drawing: Again, the subject engages in the connect-the-dots task for 2 minutes.

Longer Programs: Subject engages with longer versions of the same types of problems as in Cycle 1. Two minutes are allocated for three trials:

- 2-cells-away:** This repeats the format last used in Cycle 2.
- 3-cells-away:** This increases the length of the required solution by 1.
- 4-cells-away:** This again increases the length of the required solution.

Rest and Calibration: An 18-second period of staring at the cross-hair pattern ends this cycle.

3.5 Cycle 3 Schedule

Line Drawing: This cycle again begins with the connect-the-dots task for 2 minutes.

Paths with Obstacles: Subjects solve one problem involving a length-4 path around obstacles and additional problems with longer paths, also avoiding obstacles. Correct ordering of the blocks used

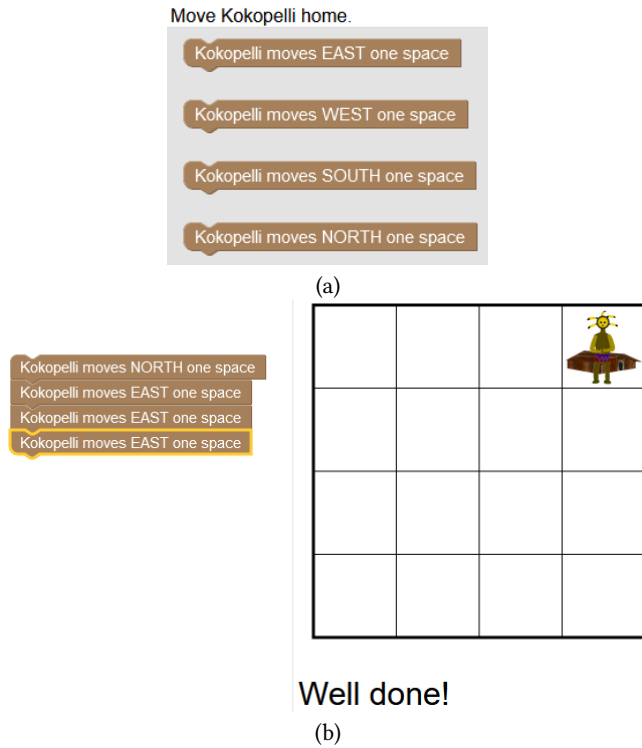


Figure 9: The menu (a) and a solution (b) for an exercise on getting Kokopelli home to the longhouse. Both the solution program (with four blocks) and the goal state in the microworld are shown.

is very important in solving these problems. Fig. 10 and Fig. 11 show challenges involving obstacles: first an easy case then one that is a bit more challenging.

3.6 Cycle 4 Schedule

Line Drawing: Again the subject begins with the connect-the-dots task for 2 minutes.

Programs with Loops: First the subject is given a problem to create a program with the loop (for example, that shown in Fig. 12). No additional steps other than those in the loop are needed. After that a problem is given in which a loop is needed, and there is an step to be taken (see Fig. 13). This was the most complex coding task used in this study.

3.7 Completion Image

After Cycle 4 is complete, the subject is congratulated for finishing the activities and shown the image of Mount Rainier (Fig. 14). Thus each subject ends his or her engagement in the scanner with the computer with this attractive image of Mount Rainier, a local landmark that is usually hidden in the clouds and typically much appreciated when clearly visible. The reason to have this was to help insure that our subjects left with calm and positive feelings about the experience in the scanner, regardless of how they might have felt during the exercises.

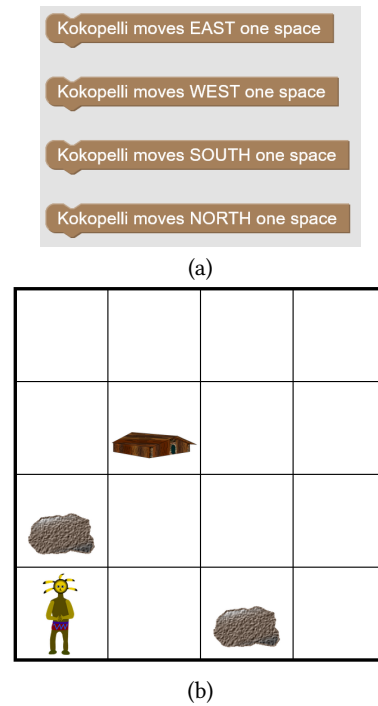


Figure 10: The menu (a) and initial microworld (b) for a simple challenge involving an obstacle.

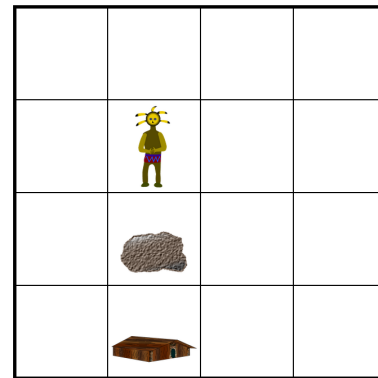


Figure 11: The initial microworld for a more challenging obstacle avoidance exercise.

4 DATA COLLECTION AND ANALYSIS

Human subjects were recruited and prepared for scanning by coauthor Yates. In addition to subject S_0 and one 27-year-old subject (S_5), there were four children scanned. A larger sample was intended but funding ran out before such a group could be recruited and studied. In addition to those actually scanned, some other children were recruited and prepared, but they became concerned when they heard noise from the actual scanner, and then they left the study without being scanned. Although the overall research center that sponsored this study addressed the needs of students

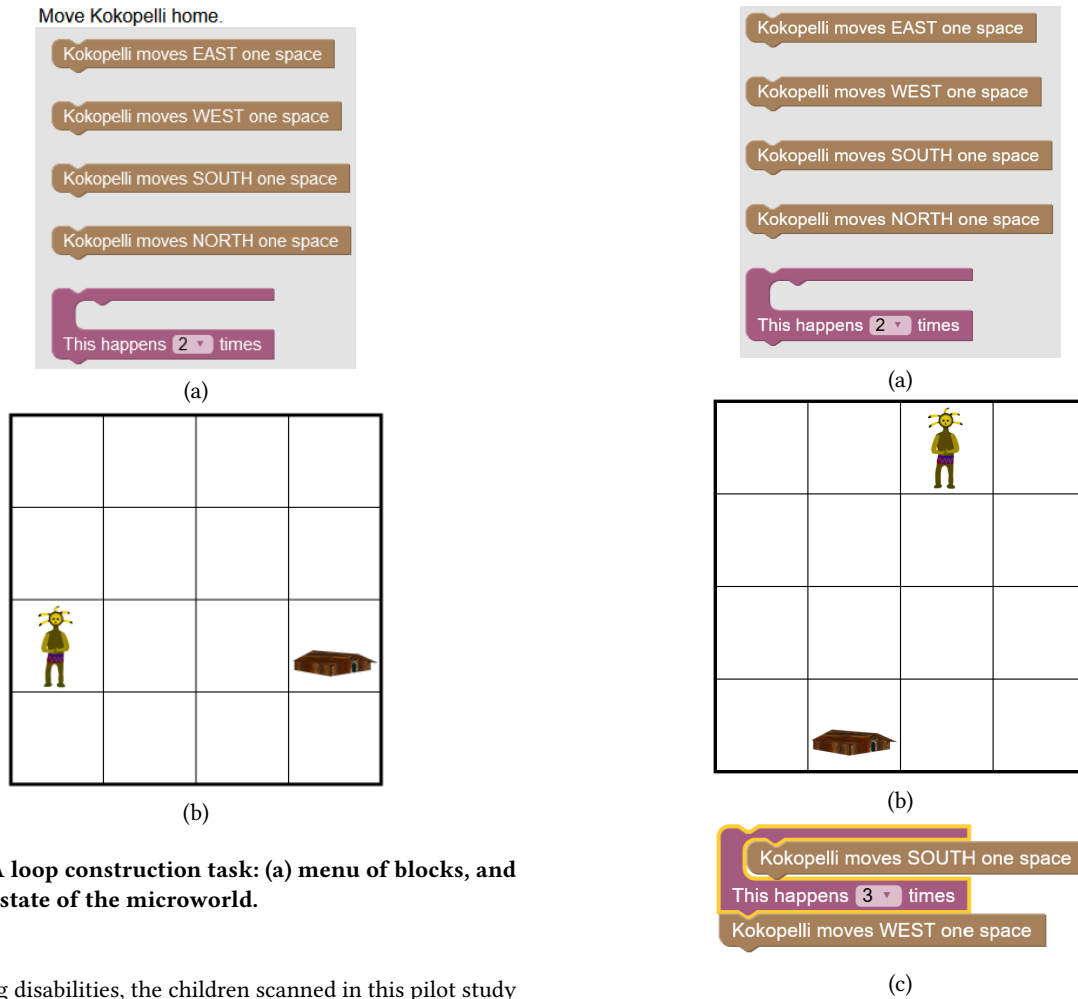


Figure 12: A loop construction task: (a) menu of blocks, and (b) starting state of the microworld.

with learning disabilities, the children scanned in this pilot study were not recruited from any learning-disabled population.

Dr. Yates trained the children on a simpler version of the computer programming activities described in Thompson et al[13]. The children practiced simple coding of sentences on personal computers. Dr. Yates checked that the students could comprehend the If-Then sentences and could move the cursor. (It turned out that If-Then sentences were not used during the exercises; however, this this assessment served as a proxy for readiness to engage in the exercises.) She also conducted a practice session with the children before they entered the scanner. The four children that were scanned had the following ages and genders:

Subject no.	age	gender
S_1	12 years 11 months	female
S_2	14 years 0 months	female
S_3	10 years 8 months	male
S_4	10 years 9 months	male

5 OBSERVATIONS

5.1 Report from the Field

Programming can be performed in an fMRI scanner without a normal keyboard and mouse, provided there is some other method available to point and click. Having the programming be visual (e.g.,

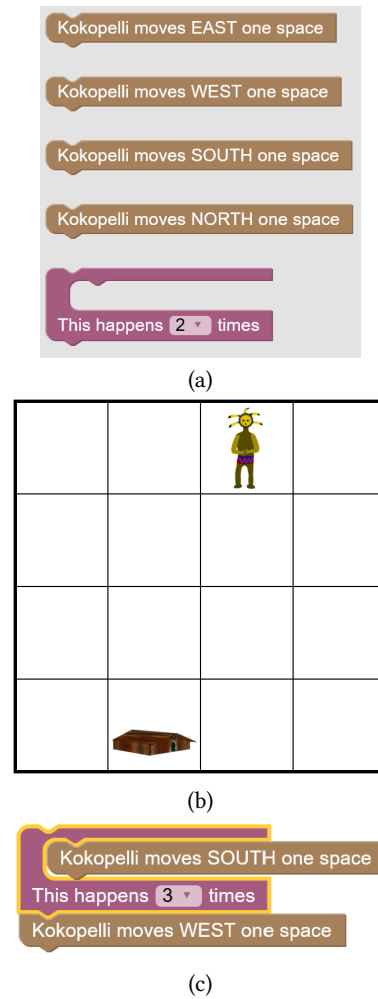


Figure 13: A second loop construction task, this time requiring an additional step: (a) menu of blocks, (b) starting state of the microworld, and (c) blocks program that handles the task.



Figure 14: Post-exercises reward screen.

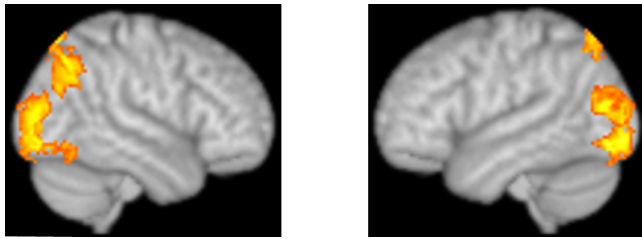


Figure 15: Projections of right-brain and left-brain activity where the coding activity levels exceeded those of the control task (connect-the-dots). The activation levels used were averages based on data from the four juvenile subjects. High-activity regions are highlighted in orange.

placing blocks that represent program elements) can cut down or eliminate the need for entering text. The Current Design Trackball 2 was adequate for our programming tasks in the scanner.

It is interesting to note that programmers in virtual reality today often face the same problem — a normal keyboard is not available within VR (although it could be in Augmented Reality). Virtual keyboards are available, but they can only be operated using a pointer and controller button.

If S_0 had had a normal-layout keyboard and arms-to-keyboard positioning, he might have had to sit up, and therefore also have a different mirroring arrangement for the display. Computer keyboards, however, are still generally unapproved for in-scanner use for the safety reason that they contain metal parts. These parts may interact with the strong magnetic fields from the scanner producing effects that could include physical forces that induce motion and lead to injuries, heat that could cause burns to the subject and damage to the keyboard, or interference with imaging.

The physical experience of programming in the scanner was surprisingly not bad. Programming while lying on one's back worked out well. It was comfortable using the trackball, with it located above the stomach. The screen was clearly visible without either eyestrain or muscle strain, due to the mirroring arrangement. Although a scanning subject is dressed in only hospital scrubs, a blanket is provided that keeps one warm enough. What could be more luxurious for a programmer than coding in bed?²

The first author has written separately, using a poetic form, about his experience of coding in the fMRI scanner. His coauthors suggested that it be included in this paper. It can be found in the appendix.

5.2 Results from the fMRI Images

The analysis for the fMRI data was carried out using FEAT (fMRI Expert Analysis Tool) Version 6.00, part of FSL (FMRIB's Software Library, www.fmrib.ox.ac.uk/fsl). The Z (Gaussianised T/F) statistic images were thresholded using clusters determined by $Z > 2.3$ and a (corrected) cluster significance threshold of $P=0.05$ [14].

²While this was the first author's thought upon reflection, others might react differently, and scanning experiences will vary according to locations, practices, and prior conceptions.

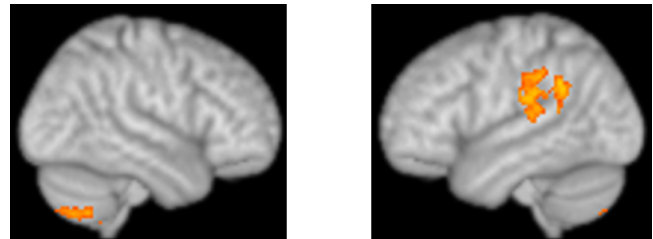


Figure 16: Projections of right-brain and left-brain activity where the control activity (connect-the-dots) levels exceeded those of the coding tasks. As in Fig. 12, the activation levels used were averages based on data from the four juvenile subjects, and high-activity regions are highlighted in orange.

Across all six participants (4 children and 2 adults), Dr. Richards identified common brain regions of fMRI BOLD (blood-oxygen-level-dependent) activation for the coding task (programming) performed during brain scanning. This was facilitated by first performing co-registration of the individual brain images to the standard Harvard/Oxford cortical atlas distributed with the FMRIB Software Library. The 12 common regions included Middle Frontal Cortex, Precentral Gyrus, Superior Parietal Lobule, Lateral Occipital Cortex-superior division, Lateral Occipital Cortex-inferior division, Intracalcarine Cortex, Precuneus Cortex, Cuneal Cortex, Lingual Gyrus, Temporal Occipital Fusiform Cortex, Occipital Fusiform Gyrus, and Occipital Pole. Collectively, these included brain regions involved in working memory, executive functions, language processing, and visual processing. This analysis shows that in spite of the age differences among the six participants, there were 12 brain regions of common brain activation.

Next, the programming task (coding) was compared to the control task (connect-the-Dots) during brain imaging for the four children (novice programmers) whom Dr. Yates had provided instruction in programming on a computer outside the scanner prior to performing the coding and control tasks during brain imaging. The goal of these analyses performed by Dr. Richards was to identify the brain regions in each of the four children that were unique for programming compared to the control task of connecting the dots. Nine regions of unique BOLD activation were identified for Coding > Connect-the-Dots: Angular Gyrus, Lateral Occipital Cortex-superior division, Lateral Occipital Cortex-inferior division, Intracalcarine Cortex, Precuneus Cortex, Lingual Gyrus, Occipital Fusiform Gyrus, Supracalcarine Cortex, and Occipital Pole. See Fig. 15 for Brain Image with BOLD activation in these 9 brain regions uniquely involved in computer coding.

Finally, the control task (connect-the-Dots) was compared to the programming task (coding) during brain imaging for the same four children to identify the brain regions in each of the four children that were unique for the control task compared to programming task. Six regions of unique BOLD activation were identified for Connect-the-Dots > Coding: Postcentral Gyrus, Supramarginal Gyrus-anterior division, Supramarginal Gyrus-posterior division, Angular Gyrus, Parietal Operculum Cortex, and Planum Temporale. See Fig. 16 for the Brain Image with BOLD activation in these 6 brain regions uniquely involved in connect-the-dots. The Angular Gyrus,

which is a hub with many brain networks, probably is unique in both comparisons because different networks are involved.

Note that for the analyses identifying unique brain activation for coding > control task and for control task > coding for the four children, Dr. Richards had excluded the rest (cross-hair) activities. After aggregating the images separately for the coding task and the connect the dots task, he showed the regions of unique activation in orange highlights on side views of the brain in Figures 15 and 16.

6 THREATS TO VALIDITY

Since we only were able to scan six subjects, and among them only four children, we cannot offer statistically strong evidence that the apparent differences in brain activity between the control condition and the coding condition was not due to chance.

Another significant limitation of the study was its reliance on very simple programming tasks, and on its use of a blocks-style language.

Our hypothesis in the next section, that aspects of computer programming may be inherently more pleasurable than playing a mindless game, is rather speculative, and suggested in part by prior research showing that higher volumes of gray matter in the precuneus cortex are associated with higher levels of reported happiness[9].

7 DISCUSSION AND FUTURE WORK

7.1 Happiness and Programming

We observed significant brain activity for the coding task, versus the control task, in the precuneus cortex. This brain region has been associated with positive emotion[1]. Our research suggests a hypothesis that coding is or can be more emotionally positive or fun than a mindless computer game. However, further research is needed to actually test that hypothesis. In addition to scanning subjects performing coding and other activities, the research should make use of other assessment instruments such as questionnaires designed to gauge affect. In the future, scanning might conceivably be used as one form of evidence in answering questions such as “What is the most enjoyable programming language to work with?”

7.2 Practical Benefits

Better understanding of the brain processes involved in computer programming could lead to improving the programming experience. Consider how three categories of programmers might be affected:

novice programmers: By identifying the brain areas most involved in learning to program (including areas related to affect), instructional materials and curricula for teaching programming may be adjusted to follow best practices associated with component activity learning, such as reading, writing, mathematical reasoning, planning, and memory.

end-user programmers: Factors and features that enable productive work in specific domains such as data analysis, architectural design, etc., may be supplemented with supports for those aspects of computer programming that are not already provided in the work environment.

professional software developers: One objective for many developers is reaching and staying in a mental state of “flow”

in which high degrees of focus, concentration, and productivity are reached[2]. An understanding of brain processes for developers in a state of flow may help them structure their workplaces and work strategies more effectively than without such understanding.

Further research is needed to build on these exploratory research findings to conduct studies with larger samples, to identify the common and contrasting brain activation patterns for individuals who are novice compared to expert programmers, and to follow children and youth longitudinally as they learn to become computer programmers and progress from novices to increasingly proficient. Secondly, future research should study professional-level programming in-scanner.

7.3 Writing Code vs. Reading Code

We believe that the in-scanner coding trials reported here were the first cases of any computer programming in an fMRI scanner, although the code-reading trials[10] preceded them. Therefore, one might ask what, if any, differences, in terms of brain processes, exist between code writing and code reading. Both require the understanding of programs – code comprehension. In our examples with Kokopelli’s World, this was a mental simulation of Kokopelli’s motions in the microworld. In the earlier case[10] it was mental simulation of data manipulation. What’s different with writing code, from mere reading, are (a) the process of expressing and externalizing the code (mental composition, followed by muscle control for communicating the code to the system), and (b) responding to the execution (as feedback) and subsequently engaging in an interactive debugging process. Further research should aim to better understand the brain activity that supports the composition, externalization, and debugging components of code writing.

7.4 Coding versus Reading and Writing

Future research should also study the similarity between computer programming and writing in natural language, using fMRI scanning and a variety of specific writing tasks and programming tasks, including (a) conversational writing (as in chat dialog, versus conversational programming as in a REPL (Read-Eval-Print Loop) or a Jupyter notebook[6]; (b) correction of prose for consistent storytelling, versus debugging of code for achieving semantically correct programs, and perhaps (c) writing a legal contract, versus writing a functional program in Javascript or Haskell.

And finally, in addition, future work should study whether people with learning disabilities or dyslexia tend to be equally affected in terms of computer programming versus reading and writing, and whether specific interventions to enhance programming capabilities could then be designed to exploit those commonalities.

ACKNOWLEDGMENTS

Support for the scanning was provided by the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD) under grant P50HD071764.

A A POETIC ACCOUNT OF CODING IN A SCANNER

This appendix presents a personal account of the experience of programming in an fMRI scanner, by the first author. It has been included at the urging of the co-authors, and as a contribution to the spirit of the PX/22 Programming Experience Workshop, as described in the call for papers.

Gray Matter: Thoughts After Seeing My Thoughts While Coding in a Scanner

© Steve Tanimoto, July 10, 2016,
reprinted with permission

Neuroscience is the new frontier,
Like lunar landing in a bygone year.

A brain scan is done just in case
We wish to see thinking take place.

Our research setup's now quite near,
And now we need a volunteer.

Whose brain will be very first
To have its sections all traversed,

While engaging in a task,
In this case coding, must you ask?

Not encryption, fraud or scamming,
It's just computer programming.

Who will be the pilot subject?
Perhaps it's he who doesn't object.

Whose idea was this anyway,
Scan a coder to see what we may?

Someone needs to test the plan.
That someone's brain must have a scan.

But let science and truth be damned;
I don't want my own brain scanned.

Each cook should taste the soup's spice mixture
And a researcher must test the fixture.

Human subjects -- they sign a waiver
And earn a T-shirt as a favor.

But the researcher must go first,
To show the rest that at very worst,
You'll recline for an hour unreimbursed.

So I give in and do my part
To further science's state of the art.

There's one thing that we first must settle,
That could turn out to be a nettle.
MRI scanners don't like metal.

No pins, studs, plates or spring
May be inside that scanner's ring

Or it might quick as a whistle
Turn into a deadly missile.

Dental fillings are OK,
I guess 'cause they can't fly away.

In scrubs and a blanket, I recline,
And the machine proceeds to self-align.

"Keep your head still, do not shift,
Or the images will go adrift."

Alright, I'm ready. Start the scan.
I stare into space as still as I can.

The scan itself proceeds as planned,
The noise no worse than a techno band.

The coding's of the Blockly kind,
Without a keyboard, it's redefined
As trackball-controlled blocks aligned.

I must program "Kokopelli"
So he'll walk home to his sweet Nellie.

I move some blocks so he goes forth,
First to the east and then to north.

I click Run and he has fun,
And then I get another one.
Now Kokopelli must move west
To get home by the best

Route of which there may be more,
But shortest is the one to score.

My brain is working, on the move,
And I'm caught up in the groove.

But now I have to use a loop
To get to Kokopelli's stoop,

Because it would be just too far
And step-by-step is over par.

The looping block is pretty cool --
A nice long walk through a simple tool.

And after Kokopelli's walked,
As well as stumbled, stopped and balked,

The game is suddenly suspended
And Kokopelli's World is ended.

The MRI scan is complete.
The scanner's bed is in retreat.

I come out from my cyber-land,
Change out of scrubs and try to stand.

It takes a sec, but I'm OK.
The magnet's off, headphones away.

In the control room is our team
Who say the images are clean.

They say my brain is not too alien
And within parameters mammalian.

The sagittal section's pretty weird.
My nose is there but not my beard.

I mean my moustache; it is gone.
May I have the screen redrawn?

And what's that cauliflower there?
"Cerebellum" is that part where
Motion and balance are the main affair.

An hour goes by and then email.
I get the slices in great detail.

One hundred and seventy-six files
Of monochrome spots from tiny tiles.

They should be voxels in a 3D view
But that would require a redo.

But then the next day something comes
And my mind to curiosity succumbs.

Pristine pictures of my brain,
Regions colored that pertain
To the coding-specific thought domain.

Mostly in the lobe occipital,
Where it's visual stuff that's typical,

It's lit up there and in the parietal:
Spatial sense for which it's vital.

There is some light in the frontal lobe,
Perhaps the looping, as we probe.

Soon it hits me, how meta this is,
And what philosophical bliss is.

An aboriginal first meets a mirror.
A deaf person first becomes a hearer.

I myself have become nearer
To where philosophers first get clearer.

I think I see the thought I thought --
The resonance that imaging brought.

My plan for Kokopelli's stroll,
Is onscreen, in-skull, and console.

And now I try to comprehend
What makes my cerebrum attend

To comprehension of attention
To coding, and not to mention:

I wonder therefore I am.
What if Descartes had had a scan?

REFERENCES

- [1] S. An, X. Han, B. Wu, Z. Shi, M. Marks, S. Wang, X. Wu, and S. Han. 2018. Neural activation in resp. to the two sides of emotion. *Neurosci. Ltrs.* 684 (2018), 140–144.
- [2] Mihaly Csikszentmihalyi. 1990. *Flow : the psychology of optimal experience* (first edition ed.). Harper & Row, New York.
- [3] B. Floyd, T. Santander, and W. Weimer. 2017. Decoding the representation of code in the brain: An fMRI study of code review and expertise. In *Proceedings of the 39th International Conference on Software Engineering*. 175–186.
- [4] Current Designs Inc. [n. d.]. Trackball 2. Available at <https://www.curdes.com/mainforp/responsesdevices/variabledevices/hhsc-trk-2.html> (2022/03/06).
- [5] Anna A. Ivanova¹, Shashank Srikant, Yotaro Sueoka, Hope H. Kean, Riva Dhamala, Una-May O'Reilly, Marina U. Bers, and Evelina Fedorenko. 2020. Comprehension of computer code relies primarily on domain-general executive brain regions. *bioRxiv preprint* (2020). <https://doi.org/10.1101/2020.04.16.045732>
- [6] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. 2016. *Jupyter Notebooks-a publishing format for reproducible computational workflows*. Vol. 2016.
- [7] Ryan Krueger, Yu Huang, Xinyu Liu, Tyler Santander, Westley Weimer, and Kevin Leach. 2020. Neurological divide: An fMRI study of prose and code writing. In *Proc. 42nd Int'l Conf. on Software Engineering*. 678–690.
- [8] Richard Pattis. 1981. *Karel The Robot: A Gentle Introduction to the Art of Programming*. John Wiley & Sons, New York.
- [9] Wataru Sato, Takanori Kochiyama, Shota Uono, Yasutaka Kubota, Reiko Sawada, Sayaka Yoshimura, and Motomi Toichi. 2015. The structural neural substrate of subjective happiness. *Scientific Reports* 5 (2015). <https://doi.org/10.1038/srep16891>
- [10] J. Siegmund, C. Kastner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. 2014. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th International Conference on Software Engineering*. 378–389. <https://doi.org/10.1145/2568225.2568252>
- [11] R. Thompson. 2019. *HAWK/KW: An Online System for Studying Dyslexic Children's Reading, Writing, and Programming*. Ph. D. Dissertation. Univ. of Washington.
- [12] R. Thompson, S. Tanimoto, V. Berninger, and W. Nagy. 2016. Coding, reading, and writing: Integrated instruction in written language. In *Proceedings of the 2016 International Symposium on Visual Languages and Human-Centric Computing*. Cambridge, UK, 378–389. <https://doi.org/10.1145/2568225.2568252>
- [13] R. Thompson, S. Tanimoto, R. Lyman, G. Geselowitz, K. Begay, K. Nielsen, W. Nagy, R. Abbott, M. Raskind, and V. Berninger. 2018. Effective instruction for persisting dyslexia in upper grades: Adding hope stories and computer coding to explicit literacy instruction. *Education and Information Technologies* 23 (2018), 1043–1068. <https://doi.org/10.1007/s10639-017-9647-5>
- [14] K. J. Worsley. 2001. Statistical analysis of activation images. In *Functional MRI: An Introduction to Methods*, P. Jezzard, P. M. Matthews, and S. M. Smith (Eds.). Oxford University Press, Oxford, UK, Chapter 14.