On Computational Thinking: A Perspective From Rural Saskatchewan

Episode 7: Tools For Computational Thinking Script

Stephen Hadden

Educational Technology and Design, University of Saskatchewan

ETAD 992: Project

Dr. Paula MacDowell

April 30, 2023

Fun tools, fun tools, Fun tools! This final episode showcases different tools and processes that I have included in my practice. I even have a few things that I am still figuring out how to incorporate. I'm Stephen Hadden and this is the final episode in this series on computational thinking.

There are lots of tools out on the internet for exploring computational thinking. The opportunities for exploring concepts like coding are endless, and range from free, to rather expensive. But coding is not the only computational exploration out there. Let's take a look at a few things.

Coding

We will start with coding. Coding is a bit problematic because this is an area that can be the most intimidating. Which language do we use? Which platform? What software has to be downloaded? What lessons are available? These are all valid questions. There are a number of tools that are pretty hands off, meaning, the students have tools to explore, and you don't need to do a bunch of prep work. And they are fun because the teacher can engage with the lessons as well as the students.

Block based programming languages are excellent places to start for both teachers and for students (Arfé et al., 2019; Rode et al., 2015; Vegas et al., 2021). Block based coding means that students build their code with colour coded blocks that have certain functions or roles. The blocks are connected together like Lego to cause some on screen object (often called a sprite) to carry out some action.

Scratch

This is great for most elementary grades and very useful in high school computer science. There is a Scratch Jr. on iPads for younger children. They have tutorials and extensions for use with robotics. It's one of the original block-based languages, and it is great for exploring coding (Scratch, 2018).

Code.org

Code.org has a series of courses for K-12 that can be used for computer instruction that can probably fill the whole year. There are project opportunities for art, mobile applications, and games. The courses start with block coding for younger grades and progressively ramp up the complexity including shifting to text-based programming for older grades (Code.org, 2023).

MakeCode

MakeCode has a number of different entry points. MakeCode Arcade is really fun for building games. These games can be played on computer or downloaded to devices like the PyGamer (Microsoft MakeCode, n.d.-a). MakeCode is the coding environment for the Micro:bit which is a great inexpensive microcontroller - microcontrollers are small little computers that can attach to other sensors and items like motors to build fun little robotic or automated devices (Micro:bit Educational Foundation, 2019; Microsoft MakeCode, n.d.-b). MakeCode has a portal to Minecraft coding if you students love using Minecraft. All of these environments have tutorials to help teachers and students become comfortable with the tools, and provide lots of extension opportunities (Microsoft, 2023).

Text-based Languages

As students get older and more experienced, they can shift to other programming languages. Python is a common language to explore after block based languages - tools like MakeCode include the ability to switch between block based language and languages like Python and JavaScript (Microsoft MakeCode, n.d.-b).

Robotics

Robotics is another great way to explore computational thinking. There are Robotics kits that take all sorts of different forms and possibilities. Organizations like SaskCode have curated viable options for specific grade groupings. Their offerings include Robot Mice for grades K-2, Ozobots for grades 3-4, Edison robots or Micro:bit for grades 5-6 and Arduino microcontrollers for grades 7-12 (Saskatoon Industry Education Council, n.d.). The complexity of the robots and their programming increase through the grades allowing for a progression to explore computational thinking skills using a variety of tools. For the younger grades coding is not part of the process. This makes the tools more accessible - easing that point of entry for the teacher and the student.

Dash and Dot by Wonder Workshop

I have used Dash and Dot Robots with Kindergarten and grade 1. Our grade 1 teacher does code club with grades 1-3 with the Dash and Dot Robots. They explore controlling the robot like a remote controlled object, and they use line drawing in an iPad app to direct the robots through a maze. The Dash and Dot robots can be programmed, and they have little attachments that can be fun to play with (Wonder Workshop Inc., 2019).

Micro:bit

I love working with the Micro:bits - we have had rock, paper, scissors tournaments and built digital pets with Micro:bits in grade 5 (Microsoft MakeCode, n.d.-b), and we have built counters to determine RPM on windmills for grade 9 science (InkSmith Limited, 2023).

Media Production and Fabrication

Some of the devices we put in the hands of the students are computing powerhouses - iPads are just incredibly powerful. They can be used to edit video, audio, photos, and create a variety of digital media products. You can use apps like Pages, keynote or Canva to complete presentations or develop posters. You can have students provide oral or video recorded responses. This can be done simply by recording audio or video using the built in camera and voice recorder apps on iPads, or you can get students to develop their editing skills with iMovie, Adobe Rush, or Garage Band.

TinkerCAD

If you want to have fun with 3D shapes, let the students mess around with the design tools in TinkerCAD - TinkerCAD can be used on computer or iPad (Autodesk, 2023). This 3D design program can lead to some fun creative results that can even be fabricated on 3D printers - it takes some time and finesse, but having that physical final product is so rewarding.

Excel or Google Sheets

Actually, I decided to give this a try with the grade 3s the other day. We needed to do some graphing of temperatures for a science activity. So, I showed the students how Google sheets can build graphs while you are typing in information, and that you can change the look of the graph to best fit your purposes. It's a simple application of computational thinking by selecting a computer tool to complete a task that can be done by hand but doesn't need to be. (Especially when the computer can do it better).

Making

Making is a great way to explore aspects of computational thinking - especially when you can incorporate technology. Many skills can be explored by building cardboard pinball tables, creating little stuffed animals with basic light circuits included. Having the opportunity to develop ideas and work towards a solution through a design thinking process or engineering process helps students grow in problem solving, iterate through prototypes and troubleshoot while working toward a physical product. It may not always be technology focused, but these tasks connect solidly to computational thinking concepts and practices (Rode et al., 2015).

Electronic Music

I had a whole episode on electronic Music that I built, but it is currently in the drafts folder. Garage band on iPad is a very powerful music tool, you can mess around with instruments, record singing and playing to the device, edit and save musical tracks. It has a pretty steep learning curve but can be really fun to mess around with.

Code.org has a tool called Project Beats that is being testing on their site. You can program drumbeats and select samples and use block-based programming to build songs (project beats). For older students, programs like SonicPi and Estuary allow them to live-code electronic music that changes as the code is changed in a live performance setting (Aaron, n.d.; *Estuary*, n.d.).

Board Games

Having kids play board games is a computational thinking activity. There is strategy involved, you have to follow instructions and rules, there are often multiple paths to victory, and you learn to adapt depending on what is going on around you. Cooperative games like Pandemic, teach about collaboration and problem planning. There are many excellent board games out there along with lots of logic puzzle games like Rush Hour, and Robot Turtles.

Unplugged Activities

Last buy not least, Unplugged activities are logic games, and activities that do not require a computer, but can be considered to be programmatic or connect to computational thinking. Having kids navigate a maze by following instructions, and STEM challenges to build a boat that can hold 100 pennies are a couple of examples. If you have some craft materials, you can carry out unplugged activities. Many coding lesson connect to unplugged activities in their resources. Here are just a few places to find unplugged activities.

Canada Learning Code

When searching for lesson plans on canadalearningcode.ca you can select unplugged under the tools and language search option.

Code.org

Code.org includes unplugged activities in their course options including all the elementary grade options. (Code.org, 2019)

MakeCode

MakeCode arcade has activities like perler bead sprites (Microsoft, 2022), that get you to physically build a pixelated sprite with perler beads, just like you might create one in the MakeCode arcade environment.

CSUnplugged.org

CSUnplugged.org is a project based out of the University of Canterbury in New Zealand which has a bunch of interesting activities (Computer Science Education Research Group, n.d.).

Besides these examples, I will include a google docs link that shows unplugged activities from all sources.

I could probably go on, but that is probably enough to start. Remember to check out the show notes for links. If you are comfortable with coding, then you will find many resources available at your fingertips, if you are a little timid about coding, then try one of these other options. There are lots of ways to get into the ideas of computational thinking, I hope that one of these options might inspire, or be new to you.

With that, we come to the end of the series on computational thinking. It has been a journey to explore the topic, and to see what is out there. I hope that I have been able to provide some information or new tools to whom ever may be listening. Thank you for taking the time to explore the podcast.

References

Aaron, S. (n.d.). *Sonic Pi—The Live Coding Music Synth for Everyone*. Sonic-Pi.Net. Retrieved January 2, 2023, from https://sonic-pi.net/

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in Primary Grades Boosts Children's Executive Functions. *Frontiers in Psychology*, *10*, 2713. https://doi.org/10.3389/fpsyg.2019.02713

Autodesk, Inc. (2023). *Tinkercad | From mind to design in minutes*. Tinkercad. https://www.tinkercad.com/

Code.org. (2019). *CS Fundamentals Unplugged Lessons*. Code.Org. https://code.org/curriculum/unplugged

Code.org. (2023). *Code.org: What will you create?* Code.Org. https://code.org/

Computer Science Education Research Group. (n.d.). *CS Unplugged*. Www.Csunplugged.Org. Retrieved January 2, 2023, from https://www.csunplugged.org/en/

Estuary. (n.d.). Estuary.Mcmaster.Ca. Retrieved January 2, 2023, from https://estuary.mcmaster.ca/

InkSmith Limited. (2023). *Climate Action Kit—Energy*. InkSmith Limited. https://www.inksmith.ca/collections/robotics/products/climate-action-kit-energy

Micro:bit Educational Foundation. (2019). *Micro:bit Educational Foundation*. Microbit.Org. https://microbit.org/

Microsoft. (2022). *Perler Bead Sprites*. Microsoft MakeCode. https://arcade.makecode.com/arts-and-crafts/perler-bead-sprites

Microsoft. (2023). *Minecraft Education Guide | Minecraft Education Edition*. Education.Minecraft.Net. https://education.minecraft.net/en-us/get-started

Microsoft MakeCode. (n.d.-a). *Microsoft MakeCode Arcade*. Microsoft MakeCode Arcade. Retrieved January 2, 2023, from https://arcade.makecode.com/

Microsoft MakeCode. (n.d.-b). *Microsoft MakeCode for micro:bit*. Microsoft MakeCode for Micro:Bit. Retrieved January 2, 2023, from https://makecode.microbit.org/

Rode, J. A., Booker, J., Marshall, A., Weibert, A., Aal, K., von Rekowski, T., El mimouni, H., Sharma, A., Jobs, J., & Schleeter, A. (2015). From computational thinking to computational making. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15*, 401–402. https://doi.org/10.1145/2800835.2800926

Saskatoon Industry Education Council. (n.d.). *SaskCode*. SaskCode. Retrieved January 2, 2023, from https://www.saskcode.ca/

Scratch. (2018). *Scratch—Imagine, Program, Share*. Scratch. https://scratch.mit.edu/

Vegas, E., Hansen, M., & Fowler, B. (2021). *BUILDING SKILLS FOR LIFE How to expand and improve computer science education around the world*. https://www.brookings.edu/wp-content/uploads/2021/10/Building_skills_for_life.pdf

Wonder Workshop Inc. (2019). *Wonder Workshop | Home of Dash, Cue, and Dot – award-winning robots that help kids learn to code.* Wonder Workshop - US. https://www.makewonder.com/