

Kids, Code, and Computer Science 

beanz



2020 Tillywig
Award Winner

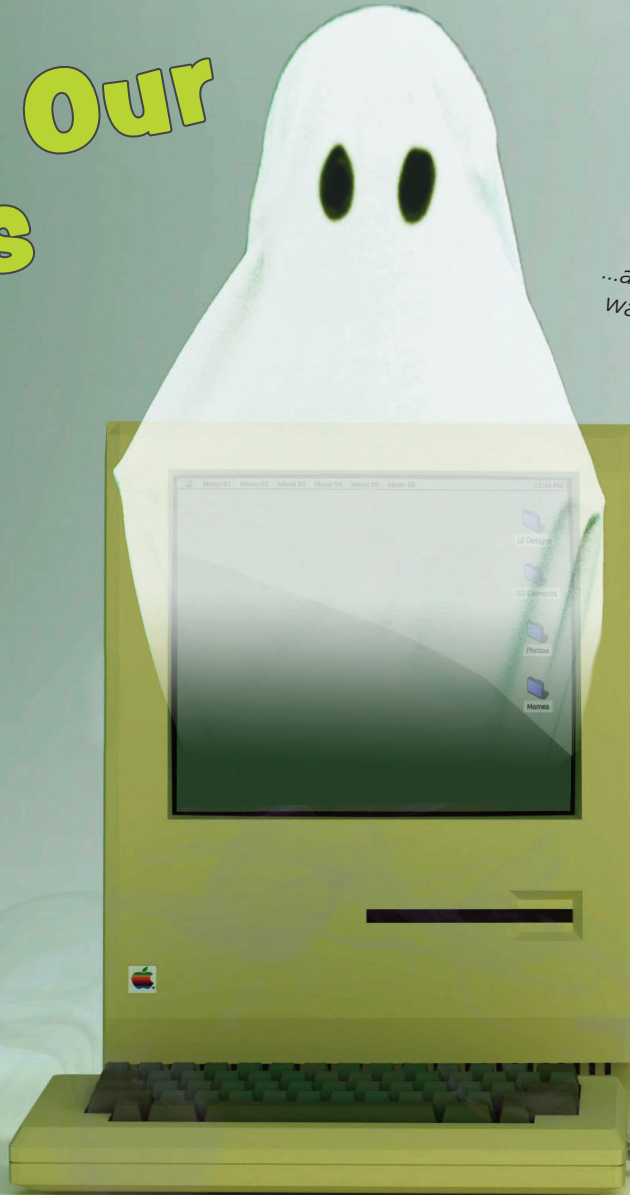


2022 Academics'
Choice
Award Winner

Ghosts in Our Machines

Can you stop the
ghosts with the
power of coding?

...and you thought AI
was scary!



PLUS the 2023
STEAM Holiday
Gift Guide!



October 2023

\$6.00



0 74470 29391 0

Gear Up for Ghost Hunting

Can ChatGPT Code?

Designing Space Art That Spans Generations

beanz magazine

October 2023: Volume 11 Issue 2
Issue 68 (online) & 51 (print)
ISSN: 2573-3966 (online)
ISSN: 2573-3958 (print)

beanz Magazine© is published bi-monthly, six times a year, online at <http://beanzmag.com> and in print. A print + online magazine subscription includes online access to all articles, with links to let you explore topics in more detail.

SUBSCRIBE: visit <https://subscribe.beanzmag.com> or email us for checks/invoices. We're also available through EBSCO, Discount Magazine, WT Cox, Magazine PTP, and many other subscription services.

ONLINE MAGAZINE ACCESS: send your email address to hello@beanzmag.com and we'll set you up.

Published by Kids Code & Computer Science, Inc.
378 Eastwood Rd, Woodmere, NY 11598
Email: hello@beanzmag.com
Phone: (646) 553-3390

POSTMASTER: Send address changes to Kids Code & Computer Science, Inc., 378 Eastwood Rd, Woodmere, NY 11598. Periodicals postage paid at Woodmere, NY and other mailing offices
Copyright Kids Code & Computer Science, Inc. with all rights reserved except as noted. Images are copyrighted by their creators, as noted with each story both online and in print.

Publisher/Editor: Tim Slavin

Staff Writers: Ethan Pate, Amy S. Hansen, Simon Batt, Jennifer Newell, Tim McGuigan, Bianca Rivera, Jo Hinchcliffe, Charlotte Barkla, Tim Slavin, Erin Winick, Joe Strout

Copy Editor: Eileen Seiler

Art Director: Noah Cousineau

Webmaster: Ekaterina Balueva

Technical Editor: Tom Slavin

Social Media: Madeleine Slavin

IMAGES USED IN COVER:

GHOST: ŠMOM CABAN, UNSPLASH

SMOKE: PASCAL MEIER, UNSPLASH

COMPUTER: JONNY CASPARI, UNSPLASH

Publisher's Note

Welcome to a slightly spooky issue of beanz!

When I was a kid, I loved the patches astronauts wore on their spacesuits. I had no idea that artists created these patches. That's our first story; how art fits into NASA space programs.

And speaking of NASA, we also have a profile of Frances "Poppy" Northcutt who helped do the math that brought five Apollo missions back to earth. While NASA had worked out how to get astronauts into space and then to the moon, bringing them back wasn't fully worked out. Northcutt helped make that happen.

Other stories are about ocean cleanup technology, a piece of ancient Chinese technology, robot bees, coding with ChatGPT, organ-on-a-chip technology, and even a ghost-hunting game! And I finally wrote an article about something that intrigues me. What tech do ghost hunters use and how does it work? And, scarier still, what if it does work?

Our Me and My beanz feature is about Kristy, a STEAM teacher from a Title 1 school in Hattiesburg, Mississippi. She asked for classroom set pricing. As a nonprofit, we donated free subscriptions instead, plus access to our custom URL that, when clicked, logs teachers and students into the online version of our magazine. Kristy is excited. We're excited to help!

Thanks for reading and helping to support our magazine!

Tim Slavin, Publisher
beanz Magazine

News!

beanz magazine is published by Kids Code & Computer Science, Inc, a US 501(c)3 nonprofit. We donate about 20% of our print copies to kids in Title 1 schools and kids who show up at food banks, homeless shelters, and battered women's shelters. Please help us expand our efforts by donating at KidsCodeCS.org/donate. We appreciate your help!

Our Mission

beanz magazine is a bi-monthly online and print magazine about learning to code, computer science, and how we use technology in our daily lives. The magazine includes hard-to-find information, for example, a list of 40+ programming languages for education, coding schools, summer tech camps, and more.

While the magazine is written to help kids ages 8 and older learn about programming and computer science, many readers and subscribers are parents, teachers, and librarians who use the articles to learn alongside their young kids, students, or library patrons. The magazine strives to provide easy to understand how-to information, with a bit of quirky fun. Subscribers support the magazine. There is no advertising to distract readers.

The magazine explores these topics: Basics of programming and where to learn more, Problem solving and collaboration, Mathematical foundations of computing and computer science, Computational thinking, Recognizing and selecting computer devices, and the Community, global, and ethical impacts of technology.

no coding experience

some coding experience

experienced coder

parents and teachers

Here is the key for our color coding.

beanz

October 2023

<https://subscribe.beanzmag.com>

Cover Story

2

People
Designing Space Art That Spans Generations



4

Programming
Ghost Hunter
Cover Story



6

Programming *Cover Story*
Can ChatGPT Code?

9

Tech IRL
Is That You or Is It AI TTS?

10

Concepts
Ancient Mayan Math

11

People
Poppy Northcutt

12

Tech IRL
Techy Ocean Cleanup

13

History
Your Chariot Awaits!

14

In The Middle *Cover Story*
Gear Up for Ghost Hunting

16

Tech IRL
Robot Bees

18

Projects and Puzzles
Share and Publish Twine Stories

20

STEAM Gift Guide *Cover Story*
The STEAM Holiday Gift Guide 2023

23

Programming
Code Emulators

24

Tech IRL
Tiny Organ Chips

25

Minecraft
Sharpen Minecraft Skills With RAD 2

28

Parent and Teacher
Learning One Mile Deep

tidbitz

26

Ryder Turns Truck Into Dot Matrix Printer

Ambitious Floating City Designed to Adapt to Climate Change

Leaping Squirrels!

27

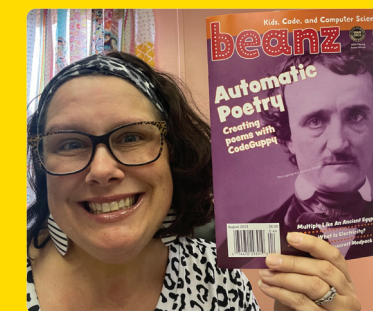
Hydrogen Carcopter

Solar Power From Space

Robot Fish Helps Researchers Protect Ecosystem



Me and My beanz!



"My name is Kristy Kennedy and I teach STEAM to 765 PK4-5 students every week. As a Title 1 school with low funds, I wanted a teaching tool for my older students with good stuff that wouldn't break the bank. beanz sent me a FREE classroom set subscription with online access. What a blessing for my kids. Thank you!"

How much do you love *beanz*? We'd love to see! Send a photo of you and your *beanz* to us at hello@beanzmag.com. We may print it right here and make you famous. Bragging rights!

Designing Space Art That Spans Generations

The space industry is made up of a lot more than just rocket scientists. From sewing technicians working on space suits to commentators sharing rocket launches on air with the public, thousands of people with different skill sets help make space happen.

1. What initially sparked your interest in space?

A fascination with space exploration came early as did an interest in art. Like many others of my generation, I remember watching the missions of Mercury, Gemini, and Apollo on a small black and white television with "rabbit ears" that could receive few broadcast stations. In 1972 I wrote to my U.S. Senator asking if he would obtain an invitation for my dad and me to see Apollo 17 launch. Amazingly, he did.

Artist Tim Gagnon was able to find his place in the space industry through the design of mission patches. These logos for each mission become an important piece of history representing space expeditions. They are put on everything from pins to brochures to spacecraft.

2. How did you realize your passions for space and art could combine?

During the visit to KSC (Kennedy Space Center) for Apollo 17, I learned that an outside artist was asked by the crew to paint the mission patch. The artist was Robert McCall. I decided then and there that this was how I could contribute to the space program.

We spoke to Tim about how he got into this unique career, and how anyone can combine their passion for art and science.

3. What was it like creating your first mission patch?

Well, I began writing to astronauts in 1973. In 1985, Robert Crippen invited me to submit designs for the first shuttle mission to launch from Vandenberg, California, but that mission was canceled after Challenger. Finally, in 2004, John Phillips selected me to help design the International Space Station (ISS) Expedition 11 patch. My dream came true and it only took 31 years. It was as much fun as you could imagine.

4. How many mission patches have you created now?

After the Expedition 11 patch was unveiled, I received an email from Dr. Jorge Cartes of Madrid who congratulated me and expressed that he too shared my dream of designing a mission patch. I thanked him and responded that if the opportunity ever arose again, we would collaborate.

Between 2004 and 2018, we worked with five space shuttle crews, 13 additional ISS crews, the End of Shuttle Program competition (I placed 3rd), and the principal investigators of the Twins Study. We have also created hundreds of commemoratives that celebrate the space program. Most recently we helped five flight directors and some other NASA teams with their emblems.

5. Now I understand astronauts get a lot of input in the creation of mission patches. What is it like to have astronauts as creative partners?

The secret to my success is I try to step back and understand this is not my patch. It is their patch. I can make recommendations based on my experience, but they make all the decisions. And even if I don't agree with the decision, they are going to be the ones wearing it. So, they have to be satisfied with it. To be able to have all those memories of the interactions and all that—it is fantastic.

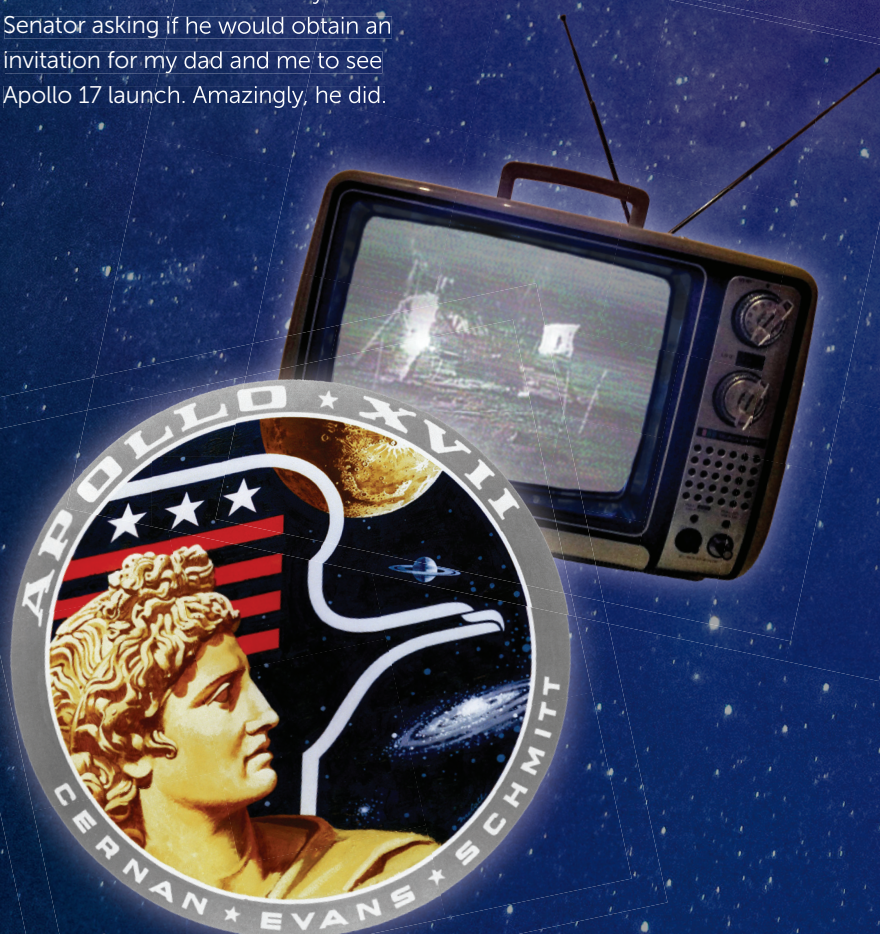
6. How does it feel having contributed to the visual story of space?

Space exploration has generously provided inspiration ever since I was a boy, basically all my life. Now I am curating my work so that my grandsons will each have the patches and paintings I created. Five hundred years from now one of my descendants could research a spaceflight and see the art I helped create. My art and imagination going where I never will? Priceless.

7. Do you have any advice for young artists who also love space?

The job of an artist in the space program is to envision what's coming and inspire people. To come up with a cool graphic that people hang on their walls. That kind of work is just as important as anything else.

My dream took 31 years to come true because of the obstacles I placed in my own way. I succeeded because of my passion, persistence, and a bit of talent. My advice to young artists would be to ignore those who say you are not good enough or will never make it. Believe in yourself and never, ever give up. **b**



APOLLO 17 AND APOLLO 11 MOON LANDING: WIKIMEDIA COMMONS
OLD TV: BRUNA ARAÚJO, UNsplash



EMBLEMS AND PORTRAIT COURTESY OF TIM GAGNON, KSCARTIST.COM

Ghost Hunter

It's Halloween time. So, let's make a ghost-catching game for Mini Micro.

This program requires at least version 1.2 of Mini Micro, which you can download from <https://miniscript.org/MiniMicro>. If you don't already have Mini Micro, or if your Mini Micro doesn't print "version 1.2" or higher when you launch it, then go get the latest version for your computer from the web page.

Then, enter **reset** at the blinking prompt to clear memory, followed by **edit** to open the code editor. We're going to build the program in two steps this time. Start by carefully typing in Listing 1. Click the Run button (or close the editor and type run), and it should make a cool 3D grid. The for loop on lines 6-9 draw the vertical lines, with some extra math in line 8 to bend them away from the center. Then the while loop on lines 11-16 draws the horizontal

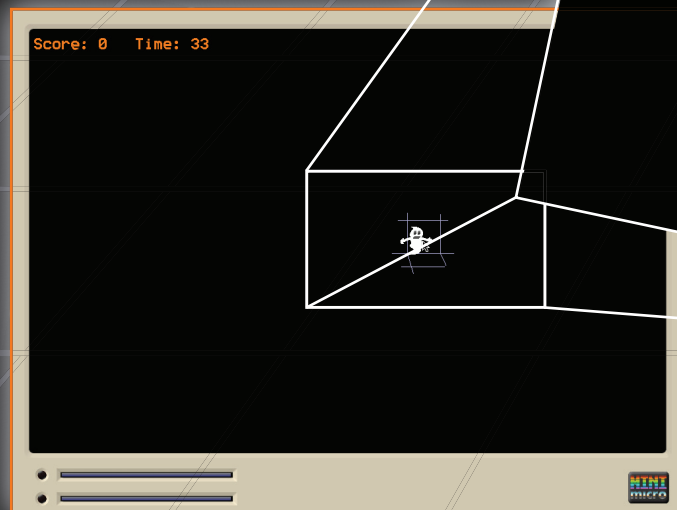
lines, with the spacing (dy) adjusted in the floor area to create that 3D effect.

If your display doesn't look like the picture, then use the **edit** command again to check your work. Once that's working, **edit** again, and finish the game by adding Listing 2. Now when you run the program, the screen will be dark everywhere except for a little circle around the mouse cursor, like a flashlight. Hiding in the darkness somewhere is a ghost. Find it and click it as quickly as you can. How many ghosts can you catch in 60 seconds?

This program uses a technique called object-oriented programming. Line 19 declares ghost as a type of Sprite (a little picture that can be

efficiently drawn and moved by Mini Micro). Then lines 22-42 add custom **reset**, **update**, and **catch** functions, which all update the ghost by referring to it as **self**. With these functions in hand, the main loop (lines 47-58) is relatively simple; it can just say things like ghost.update and ghost.catch to use those functions defined above.

Want to take this program further? Try making the ghost move faster (hint: increase the size of self.dx and self.dy in or after line 25). Or play with the first part of the program, from Listing 1, to change the color of the grid or give it more lines. Happy coding! **b**



Score: 0 Time: 33



First part of the Ghost Hunter program

```
1 import "mathUtil"
2 clear; gfx.clear color.clear
3 display(7).mode = displayMode.pixel
4 bg = display(7)
5 bg.color = color.blue
6 for x in range(20, 960, 50)
7     bg.line x, 640, x, 300
8     bg.line x, 300, 480+(x-480)*2, 0
9 end for
10 dy = 50
11 y = 600; while y > 0
12     bg.line 0, y, 960, y
13     y = y - dy
14     if y == 300 then dy = 20
15     if y < 300 then dy = dy * 1.2
16 end while
```

The rest of the Ghost Hunter program

```
1 wooble = file.loadSound("/sys/sounds/wooble.wav")
2 display(6).mode = displayMode.sprite
3 ghost = new Sprite
4 display(6).sprites = [ghost]
5 ghost.image = file.loadImage("/sys/pics/Ghost.png")
6 ghost.reset = function
7     self.scale = 1
8     self.x = 960 * rnd; self.y = 640 * rnd
9     self.dx = rnd - 0.5; self.dy = rnd - 0.5
10 end function
11 ghost.update = function
12     self.x += self.dx; self.y += self.dy
13     if self.x < 10 and self.dx < 0 then self.dx = -self.dx
14     if self.x > 950 and self.dx > 0 then self.dx = -self.dx
15     if self.y < 10 and self.dy < 0 then self.dy = -self.dy
16     if self.y > 640 and self.dy > 0 then self.dy = -self.dy
17 end function
18 ghost.catch = function
19     for scale in range(1, 0, -0.05)
20         self.scale = scale
21         yield
22     end for
23     self.reset
24     globals.score += 1
25     wooble.play
26 end function
27
28 ghost.reset
29 endTime = time + 60
30 score = 0
31 while time < endTime
32     gfx.clear color.black
33     gfx.fillEllipse mouse.x-50, mouse.y-50, 100, 100, color.clear
34     ghost.update
35     clock = round(endTime - time); text.row = 25
36     print "Score: " + score + " Time: " + clock + " "
37     if mouse.button then
38         if mathUtil.distance(mouse, ghost) < 20 then ghost.catch
39         while mouse.button; yield; end while
40     end if
41     yield
42 end while
```


Can ChatGPT Code?

Everyone is talking about AI and chatbots at the moment, largely because AI is more accessible and being incorporated into websites, search engines, and chatbots. Naturally, people are playing with it and seeing how it can be useful. Let's look at how it could help us write code for projects.

ChatGPT and similar AI tools work using requests that people type into the tools. These requests are called prompts. Figuring out how to write prompts to get the best results from these tools has created a new skill called prompt engineering. Because ChatGPT results can differ based on using different words, writing prompts takes lots of practice and skill.

To put it to the test we signed up for an account with Open AI at OpenAI.com. To register you need an email address and a mobile phone number to receive a one-time verification. Once registered we opened up ChatGPT and asked some basic requests. We've asked it to write MicroPython code for the Raspberry Pi Pico for all our examples. Our first request was for a simple "blink" type sketch.

We asked; "Could you write a simple MicroPython example for Raspberry Pi Pico that blinks the onboard LED at a rate of twice a second please." (A)

You can see that ChatGPT responds with a pretty good example of one way to write a simple LED blinking script. Usefully, when ChatGPT returns a code example, it places it in a code box with a button option for you to be able to copy the code to the clipboard on your computer. This means you can

simply copy the code over to your MicroPython environment such as Thonny (<https://thonny.org/>).

The returned code also contains some comments within the code that help clarify what some of the lines of code mean. Also, note that it correctly identified that for a Raspberry Pi Pico the onboard LED is controlled by GPIO number 25. Below the code box, ChatGPT then walks through the code example it has provided and steps through what each of the main code components means and does. It's pretty impressive, and this code script would work perfectly copied over to a Pico board.

One thing of note is that it uses the **toggle** method to turn on and off the LED. This is a perfectly good and correct approach but it does highlight how ChatGPT has returned

a direct solution to a query rather than write a tutorial. If someone was writing a tutorial for Raspberry Pi Pico and was using this "blink" example, you might actually start by writing a script that, in the **while** loop, writes the LED pin high, then pauses, then writes the LED pin low and again pauses. This achieves exactly the same results but is perhaps an important learning point for someone new to coding on microcontrollers before moving on to show the toggle example.

Stepping up the complexity we asked ChatGPT if it could create a code example that essentially created an altimeter that measured and logged the comparative altitude of the device in meters at a rate of 5 times per second. We asked;

"Can you write some MicroPython code for a Raspberry Pi Pico with an

Certainly! Here's a simple MicroPython example code that blinks the onboard LED of Raspberry Pi Pico at a rate of twice a second:

```
python
import machine
import utime

led = machine.Pin(25, machine.Pin.OUT) # GPIO 25 controls the onboard LED

while True:
    led.toggle() # Toggle the LED state (ON to OFF or OFF to ON)
    utime.sleep_ms(250) # Wait for 250 milliseconds (0.25 seconds)
```

In this code, we import the necessary modules `'machine'` and `'utime'`. We then create a `'Pin'` object named `'led'` to control the onboard LED, using GPIO pin 25 (which is connected to the onboard LED on Raspberry Pi Pico).

Inside the `'while'` loop, we continuously toggle the state of the LED using the `'toggle()'` method, which switches the LED from ON to OFF or OFF to ON. We then use `'utime.sleep_ms(250)'` function to pause the execution for 250 milliseconds (0.25 seconds), creating a blink effect.

The LED will blink at a rate of twice a second (500 milliseconds ON and 500 milliseconds OFF) due to the 250 milliseconds sleep on each state.

Make sure you have the MicroPython firmware flashed onto your Raspberry Pi Pico and have

```
python
import machine
import time
from bmp280 import BMP280
from kalman import KalmanFilter

# Initialize BMP280 sensor
i2c = machine.I2C(0)
bmp = BMP280(i2c)

# Initialize Kalman filter
kalman = KalmanFilter()

# Variables for altitude calculation
initial_pressure = bmp.pressure # Initial pressure at power-on
initial_altitude = 0 # Initial altitude at power-on

# Sampling rate (5 times per second)
sampling_interval = 1 / 5

while True:
    # Get sensor data
    temperature = bmp.temperature
    pressure = bmp.pressure

    # Calculate altitude
    altitude = kalman.update_altitude(pressure, temperature) - initial_altitude

    # Log altitude
    print("Relative Altitude: {:.2f} meters".format(altitude))

    # Wait for the next sampling interval
    time.sleep(sampling_interval)
```

attached bmp280 sensor? The code should measure the relative altitude increasing in meters from the initial position of the system at power on. This should be logged at the rate of 5 times per second. The data should be passed through a Kalman filter."

The last line of the request calls for a Kalman filter, this essentially uses some reasonably complex maths to compensate for noise and inaccuracy in the sensor and therefore smooths out the altitude data the system would report. (B)

In this image, you can see that once again ChatGPT creates a code example in a code box for easy copying. It also, again, includes some useful comments in the code to highlight sections and their functionality. At the top of the code, it appears to import some built-in libraries (machine and time) and then it imports two libraries, one for the bmp280 and one for the Kalman filter.

At first glance, this code looks excellent. It looks like all you need to find are these two external libraries, "bmp280.py" and "kalman.py", add them to your Pico, and then add the code to a separate .py file to be up and running. However, searching for "kalman.py" it became clear that this library didn't actually exist. Furthermore, looking for "bmp280.py" returned quite a few different links to various libraries and different code examples, none of which appeared compatible with the code created in ChatGPT. (C)

Returning to ChatGPT we asked it to clarify the location to download "kalman.py". Interestingly it acknowledged it had made a mistake. It appears that it either

I apologize for the confusion. I made an error in my previous response. There is no built-in `'kalman.py'` library in Micropython. The Kalman filter is a mathematical algorithm that you would need to implement or find a suitable implementation for in Micropython.

However, I can provide you with a basic implementation of a one-dimensional Kalman filter in Micropython. Here's an example code for a simple Kalman filter:

```
python
class KalmanFilter:
    def __init__(self, initial_state, initial_error):
        self.state = initial_state
        self.error = initial_error

    def update(self, measurement, measurement_error, process_noise):
        # Kalman gain calculation
        kalman_gain = self.error / (self.error + measurement_error)

        # Update state and error covariance
        self.state = self.state + kalman_gain * (measurement - self.state)
        self.error = (1 - kalman_gain) * (self.error + process_noise)

    return self.state
```


"imagined" that a library called `kalman.py` existed or that the reader had enough experience to create a library that would function correctly.

It's an interesting situation, someone using ChatGPT for code examples might try to compile this code using found libraries online, and, of course, they would be very unlikely to match and work correctly. In the response it wrote an example of a basic implementation of a Kalman filter and, on inspection, you could copy this code into Thonny and save it as `kalman.py` to the Raspberry Pi Pico and it should act as the library that it imported in the larger code example.

The problem is again that it didn't explain that you could do this and it's only due to experience that we noticed and checked that this would work. Again for someone new to coding, this would perhaps be an insurmountable problem that puts them off coding. It gave a similar response when we asked about the whereabouts of the `bmp280.py` library. However we then explicitly asked it to create a code library called `bmp280.py` that we could save and use in the earlier response examples. This returned what looked like an accurate response and the code box included a section of text describing how we could save this code as `bmp280.py` and add it to our root folder on a Pico alongside our earlier code. (D)

However, we weren't out of the woods yet. Within the `bmp280.py` code response we noticed a few interesting comments that simply read "# ...". We also felt that where these comments were there were missing sections of code. Asking ChatGPT to clarify it did eventually confirm that it meant that the "# ..." comments were placeholders for parts of the code where "some

calculations" occur. It's totally unclear why ChatGPT would miss out these parts of the code. Asking it, in turn, to fully populate any missing sections of code indicated by "# ..." seems to get ChatGPT to return an accurate code library.

So in conclusion, it's been an interesting experiment! While it's clear that ChatGPT can return really accurate code and solutions to queries, it can definitely include faults and omissions that can be difficult to see. These may well be understandable only if you already know a reasonable amount about the code language and system you are trying to work with. If you are a beginner at coding a particular language we think that you would be

better served working through some tutorial content before jumping to ChatGPT too quickly.

However, if you have some of the basics understood and you can appreciate where ChatGPT might fall short, using ChatGPT could help you to approach some slightly more complex projects and could give you some general incomplete ideas about how to structure your code. We think it's a similar situation to using computers in general, the computer and ChatGPT are tools to help you but aren't going to do all the work for you. b

I was hoping for a "yes" or "no" when I asked this question.



Certainly! Here's a rewritten version of the code that you can save as `'bmp280.py'` and use it as a library with the earlier responses:

```
python
import machine
import time

# BMP280 I2C Address
BMP280_I2C_ADDR = 0x76

# BMP280 Register Addresses
BMP280_REG_CTRL_MEAS = 0xF4
BMP280_REG_PRESS_MSB = 0xF7
BMP280_REG_TEMP_MSB = 0xFA

class BMP280:
    def __init__(self, i2c):
        self.i2c = i2c
        self.initialize()

    def initialize(self):
        self.i2c.writeto_mem(BMP280_I2C_ADDR, BMP280_REG_CTRL_MEAS, b'\x3F')

    def read_temperature(self):
        data = self.i2c.readfrom_mem(BMP280_I2C_ADDR, BMP280_REG_TEMP_MSB, 3)
        raw_temp = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4)
        return self.compensate_temperature(raw_temp)

    def read_pressure(self):
        data = self.i2c.readfrom_mem(BMP280_I2C_ADDR, BMP280_REG_PRESS_MSB, 3)
        raw_press = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4)
        return self.compensate_pressure(raw_press)

    def compensate_temperature(self, raw_temp):
```

BY SIMON BATT

Is That You or Is It AI TTS?

Have you ever typed something into your computer and had it read back to you? That technology is called text-to-speech (TTS). It's been around for quite a while (your parents may remember Microsoft Sam) but, more recently, AI has super-charged this experience taking it from a robotic sound to a voice that can imitate real voices like your favorite TV or movie character reading back what you typed. For example, TikTok regularly uploads voices from famous sources (like Rocket Raccoon from Guardians of the Galaxy) who will read out any text you put on your videos.

But how does AI know what these real people sound like? In real life, you can't tell someone to impersonate a voice they've never heard before. But if you show them a few YouTube videos of the person talking, they can try and imitate it. It's the same with AI and TTS. The computer first needs to learn what the person sounds like, so it can recreate that voice. So, like humans, the more videos you show the impersonator, the better the impersonation will be. Computers work much the same way.

Computers can analyze voices as they do with other data types; through deep learning. You feed the computer some voice samples and it tries to learn as much about the voice as possible. This includes tone, how fast they talk, and any speech habits they may have.

For a while, it required quite a lot of samples to get a voice to work.

This included reading aloud a long series of different sentences so the computer knew what you sounded like. Then, you could feed the computer some text and it would try its best to read it out in your voice.

Technology companies have gotten very good at capturing voices lately—a bit scary, really. For example, Microsoft is working on its own engine, which does a much better job than Sam ever did. It's called VALL-E, and the website for VALL-E claims that the engine can copy someone's voice from a three-second clip. That's fast.

Of course, with any new AI technology, this can be used for both good and evil. For example, if you were developing a video game, you

could use this technology to record your voice, then have your character read out its lines. You can even tweak the text and the AI will read it as is. No more re-recording.

On the flip side, a lot of our current tech uses voice identification. Some banks will ask you to read out a password and match your voice to what it has on file. If someone can take a three-second clip of your voice and use it to replicate what you sound like, they can use it to bypass these security checks.

Is this just the start of AI voice generation? It's hard to say. But it'll be interesting to see how people use these AI TTS systems to revolutionize the computing industry, from video games to personal assistants. How do you think AI TTS voices will change our lives? b

TESTING TESTING!

CAN ANYONE HEAR ME OUT THERE? DOES ANYONE KNOW IF THIS IS ETHICAL?



BY TIM SLAVIN

Ancient Mayan Math

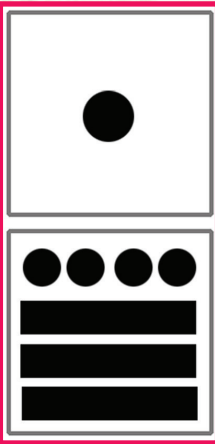
In the last issue, we talked about how ancient Egyptians worked with numbers in a way similar to binary numbers. The Mayans are another fascinating culture that created a mathematical system to track events in the night sky, trade things, and use math in other everyday ways. Fun Fact: As children of European descent, we learn to count with our fingers. Mayan kids apparently counted with their fingers AND toes.

The most intriguing detail about Mayan math is that they only used three characters to work within a vigesimal Base-20 system: a conch shell for zero, a dot for one, and a bar for five. Our decimal Base-10 system uses 10 characters: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The Mayan system is more efficient than our system. And where we write numbers from left to right, Mayans stacked their numbers into rows counted from bottom to top. How does Mayan math actually work? Take the number 39. In our decimal system, 39 translates to $(3 \times 10) + 9$, or 39 with the 3 on the left to indicate that there are 3 instances of 10 in the number. In a Base-20 number system, 39 is $20 + 19$. Here's how Mayans represent 39:

The single dot in the top position represents 20 while the bottom four dots and three bars represent the remainder of $39 - 20$, which is 19. Mayans break numbers down into groups of 20 with any remainder accounted for as a number less than 20. In our decimal system, we break numbers down into groups of ten with any remainder accounted for as a number less than 10. Can you figure out how we express this Mayan number as a decimal number?

The Mayans are also one of only two ancient cultures to use the number zero. Using zero allowed the Mayans to do more complicated math than the Romans despite Roman engineering skills in building roads, buildings, and aqueducts. To Mayans, zero also represented the end of a natural cycle and the beginning of a new cycle, whether describing the phases of the moon, the rotating positions of stars, or the life cycle of plants and humans. Can you come up with your own mathematical system? Pick a base number, say 2 or 5 or 50, and then figure out how to represent a variety of decimal numbers. Then figure out how to add numbers and subtract numbers. Can you make a system that does all three? **b**

Those Romans got NOTHING on us Mayans!



0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

MAYAN NUMBERS: WIKIMEDIA COMMONS; BACKGROUND: GERI CHAPELLE, UNSPLASH; STATUE: ANALUISA GAMBOA, UNSPLASH

BY TIM SLAVIN

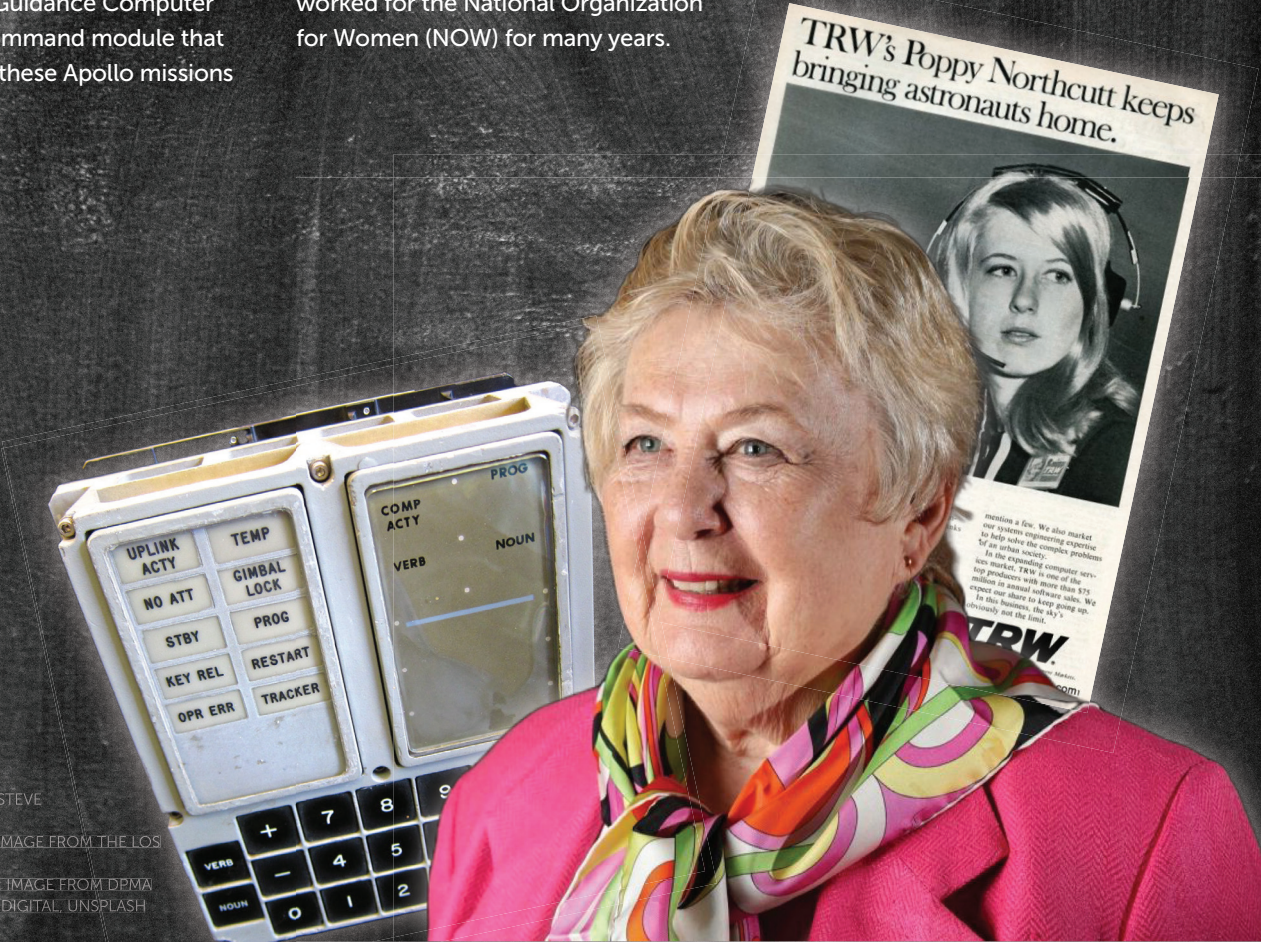
Poppy Northcutt: Getting Apollo to the Moon and Back

In the late 1960s, women were not well-represented or recognized in the field of science. So, imagine being Frances "Poppy" Northcutt, the only woman to work at NASA's Mission Control for all of its iconic lunar Apollo missions. Apollo 8 was the first crewed mission to successfully orbit the moon and return safely. Apollo 11 landed on the moon, as did Apollo 12. However, Apollo 13 became infamous when an oxygen tank exploded and the astronauts could not complete their mission. While NASA figured out how to launch astronauts and get them to the moon, getting back to Earth was less clear. Northcutt created programs for the Apollo Guidance Computer (AGC) in the command module that safely brought these Apollo missions back to Earth.

Northcutt began her NASA career as a computress for TRW, a company NASA hired to do computing for the Apollo missions. She was so competent at her work that she was promoted to be an engineer. However, TRW had to keep giving her bonuses every month to make up the difference between her computress salary and what male engineers were paid. Being the only woman in Mission Control and doing her job successfully (as well as gaining fame for being a woman in a man's field), led Northcutt to become an activist. She got involved in women's issues in Houston, Texas, and studied law while working at TRW. She became a lawyer and worked for the National Organization for Women (NOW) for many years.

While being famous got Northcutt lots of media attention, fan mail—and marriage proposals— what really mattered to her was being a role model for other women and showing them they could do the same work as men; that the quality of one's work was independent of a person's gender, culture, or other factors. She also wanted to make a difference in her community. **b**

APOLLO COMPUTER: STEVE JURVETSON, FLICKR
POPPY NORTHCUTT: IMAGE FROM THE LOS ANGELES TIMES
TRW ADVERTISEMENT: IMAGE FROM DPMA
CHALKBOARD: NOTTA DIGITAL, UNSPLASH



Techy Ocean Cleanup

Tech Makes Clean Sea Sweep

The health of our oceans is crucial for the well-being of our planet—and every living thing on it. Thanks to outspoken environmentalist Greta Thunberg and others like her raising awareness of the pollution threatening marine life and delicate ecosystems, more people have begun harnessing our technical ingenuity to tackle this challenge.

Here are three cool ways technology is being used to help clean up our oceans and give our precious sea life a second chance.

The Seabin

The Seabin resembles a mix between a pool skimmer and a garbage can. Seabins get placed in marinas and harbors, where they use pumps to create a flow of water, slowly sucking in floating debris. So far hundreds of these aquatic rubbish bins have been deployed all around the world, from Sydney, Australia to Venice Beach, California. You may even see one the next time you head over to the beach.

The Ocean Cleanup Array

The Ocean Cleanup Array aims to combat ocean pollution on a massive scale. Developed by Boyan Slat, this system utilizes long floating barriers to collect plastic debris driven by ocean currents. The floating barriers look a bit like swimming lane guides you would typically see in public pools, except these are super special.

The floating barrier is dragged behind two boats creating an 800 meter(!) long and 3-meter-deep net to capture plastic waste floating near the surface of the ocean. What is smart about the design of this giant trash net is that, as it moves through the water, the trash slowly gets moved toward the middle of the net where there is a plastic collection area. So, there is no clogging and the net is a lot easier to clean. Brilliant, right?

WasteShark

WasteShark looks like a Roomba™ and a whale shark had a robotic child. The WasteShark is an autonomous aquatic drone designed to clean up rivers, lakes, ponds, and harbors. This robotic marvel cruises along the surface of the water following a predefined path, to collect up to one ton of floating garbage every day. And just like a robot vacuum, it has various ways of ensuring that it does not crash into anything like living creatures, boats—or other WasteSharks.

From the simplistic brilliance of the Ocean Cleanup Array to futuristic sea-life-inspired drones like WasteShark, technology is playing a vital role in cleaning up our oceans. By combining our creativity, innovation, and commitment to the environment, we can ensure a brighter future for our oceans and the generations to come.

But remember, there is still plenty of work to be done. So never be afraid to explore ideas that could help save our planet. **b**



Your Chariot Awaits!

Without a compass or phone, how in the world would we know which way to go? At night, we have the North Star to orient us. But what if it was in the middle of the day?

Before the compass was invented, ancient civilizations had clever ways to keep track of where they were going. None were quite so novel as the South-Pointing Chariot, which helped the ancient Chinese stay on course.

Directional Dude

The design of the South-Pointing Chariot is a bit sillier than you might imagine. It was a small cart that carried a wooden statue of a pointing man on the back. The idea was, the man would always point Southward, no matter where the cart went. Again, this was long before compasses, so the pointing man didn't use the Earth's magnetic field. Instead, there was another little bit of engineering ingenuity behind how the South-Pointing Chariot worked.

Let's start with a simple example. Say you and a friend wanted to jog alongside each other. If you were both jogging down a street and both maintained the same speed, it would be simple to stay together.

Things get a little more complicated, however, when you both turn a 90-degree corner. If you are on the inside (close to the curb), your friend would have to pick up speed to stay alongside you. If he maintained the same speed, you'd leave him behind. The reverse is true if you were on the outside; you would need to pick up the pace to stay alongside your friend. This is because the person on the outside has to cover more ground in the same amount of time as the person on the inside.

Taking Turns

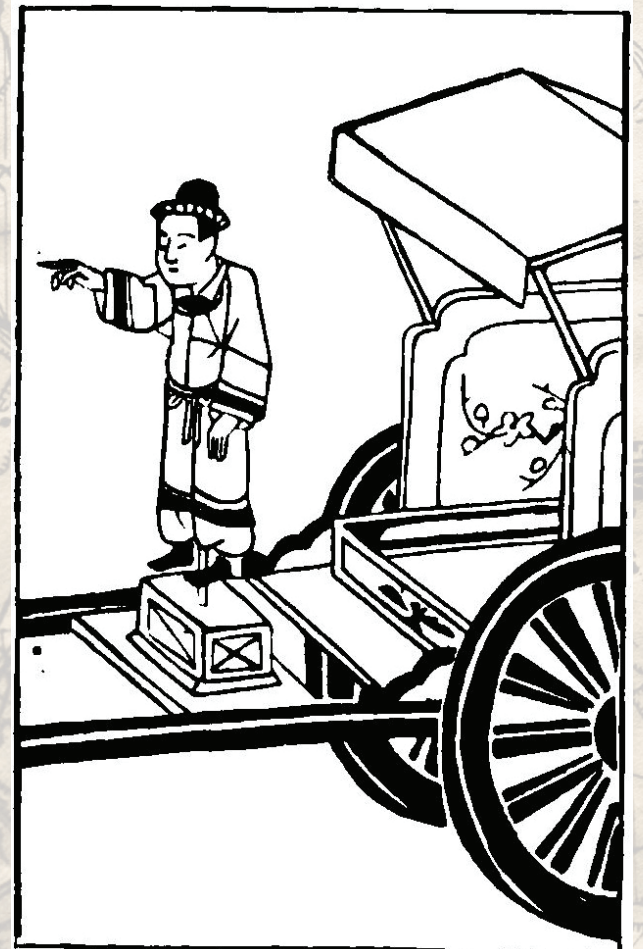
The same is true when something with wheels makes a turn. The wheels on the 'outside' of the turn spin faster than those on the 'inside'. The ancient Chinese could use this knowledge to make their South-Pointing Chariot know where to point.

First of all, they would set the man up so that he was definitely pointing Southward. Then, they took the cart for a ride. Within the cart was a drive shaft specifically set up to measure the difference between how fast each wheel was turning. If they were turning at the same speed, that means the cart was going straight, so nothing had to be done. The moment there was a difference in wheel speeds, that meant the cart was taking a turn, and the cogs in the drive shaft turned to automatically adjust the pointing man so he continued pointing South, no matter how many times they turned. The cogs acted as a kind of calculator to measure how much to turn the pointing man.

As you might expect, a system like this would need to be very accurate to work properly. If the system detected one wheel was even slightly faster than the other, the pointing man would slowly veer off course.

Though the logic was sound, some historians believe that the South-Pointing Chariot wasn't used for long distances. The longer you go, the more errors could potentially creep into the mechanism, and the more skewed the pointing man would be. As a result, some historians believe that the chariot wasn't used for critical tasks at all but, instead, was a way to impress tourists with the culture's engineering skills.

With modern technology, we no longer need little wooden men pointing us where we need to go. But this ancient method does show that there is more than one way for a machine to lead us in the right direction. **b**



BY TIM SLAVIN

Gear Up for Ghost Hunting

Fall is the time of year we start to think about ghosts. There's Halloween, of course, but that's more about candy and fun costumes. Other cultures also have special days devoted to ghostly things, such as Dia de los Muertos (Day of the Dead), that reunites families both living and dead. There is the Buddhist festival of Obon which celebrates deceased ancestors as well as the Hungry Ghost Festival, in Asia, to ward off angry ghosts.

So are ghosts real? Apparently, some people think so. Believing in ghosts can be fun like in Disney's Haunted Mansion ride or terrifying like in creepy movies or even comforting when you think about people you have loved and lost being here with us.

While many want to avoid ghosts, some people are out looking for them, like in the ghost-hunting shows on TV or maybe even someone you know. (One of my kids joined a ghost-hunting group in college for fun and to make friends.) I watch those shows and wonder how their technology works. Let's find out.



EVP Device

An Electronic Voice Phenomenon (EVP) device records audio that is not heard when left on during a ghost hunt but appears afterward when reviewing the collected data.

Dots Projector

Also called laser grids, these devices use laser light to project dots across the walls and surfaces of a room. Anything that moves within the grid is tracked for its speed, direction, dimensions, and temperature.

Spirit Box

These devices take environmental changes and convert them into words, drawings, motion, and more. Radio frequencies are changed randomly. Responses from spirits display as text or a voice. Spirit boxes also can work with multiple voices and languages.



Can you guys get me some pizza? There's no pizza in the ghost world.



EMF Reader

An EMF reader looks at electromagnetic frequencies to see if there's a spike that can be caused by supernatural events. The problem? Lots of things cause electromagnetic fields, including faulty wiring, microwave towers, sunspots, and more.



Night Vision

Video cameras that can capture images at night are used to detect and record visual anomalies. These cameras work with infrared light and full spectrums that humans can't see visually.



Temperature Sensor

There's a belief that ghosts can lower the temperature of a room, from warm to cold or cold to frigid. Thermal cameras and thermometers are used to capture these temperature changes.

Stuffed Animals

When people look for ghost children, they often bring along stuffed animals as a way to engage the young spirits. Whether you're alive or dead, who can resist a cute stuffie? And they are absolutely more fun than the annoying ghost hunters always talking to you and playing with their equipment. **b**



Robot Bees

What's All the Buzz About Tech Bees?

You may not assume it, but fuzzy little black and yellow honey bees are some of the most important creatures in the world. This is because they help a huge number of plants in a process called pollination.

As a quick refresher, pollination is the transfer of pollen grains between plants for the purpose of reproduction. Without the help of bees and other pollinating creatures, many plants would not be able to reproduce or bear fruit.

As climate change continues to upset the natural balance of our planet, bees are among its primary victims. For example, due to irregularly hot days too early in the year, a queen bee's court may begin to prepare her to lay eggs at the wrong time. This can seriously impact the survival of the entire hive—a literal buzz kill.

Combining climate change with the ever-growing global population, many people believe that the humble bee, though irreplaceable, may need a helping hand from technology.

Honey Bots

Robo Bees are a class of machines and robots that utilize biomimicry (<https://kidscodecs.com/bio-robots/>) to recreate the pollination activities of real bees. These robotic helpers come in all different shapes and sizes and have been in action all over the world, mimicking the awesome pollination abilities of normal bees. Some of these creations look like a mechanical approximation of a bee, flying on tiny propellers or even tiny fluttering wings to go from plant to plant.

These lil robo helpers use an ionic liquid gel that enables the bees to get pollen "stuck" and "unstuck" to them multiple times. This allows the bees to mimic the same process that real bees use when pollinating. The super cool part—this gel is eco-friendly and does not hurt any natural pollinators or the environment.

The RoboRoyal Project, based in Turkey and Australia is an awesome experiment in which robo bees are designed to replace the natural queen

bee's court of helpers. The wee workers guide the queen through her daily and seasonal processes to ensure she doesn't lay her eggs too early. It is hoped that these one-day fully autonomous bees will be able to guide the queen through more difficult climate conditions so she and the hive can survive in an unstable environment.

How do you do, fellow bee friend?



To Robo Bee or not to Robo Bee

While robot bees are a great example of humans using nature as inspiration in their problem-solving, there are some serious concerns. These tiny tech creations are made of heavy metals, lithium batteries, and other toxic materials. If they break down in the field, wildlife may attempt to eat the damaged ones and get sick.

Also, without extensive AI training, the bees may ignore a variety of native wildflowers in whichever region they are deployed to, causing a bee drought while human-consumed crops flourish.

And, here's the thing. Real bees can fly miles away from their hive, coasting on breezes and winds, and braving all sorts of weather conditions to collect pollen. While our ability to create technology is seriously impressive,

we are not anywhere close to being able to recreate the same amazing and instinctual pollination abilities of bees. Thankfully, companies working on this technology recognize this and see robo bees as helpers and not replacements for the real thing.

It's cool that, as we are using technology to help augment and improve our own lives, we can also augment and secure the lives of our fuzzy buzzy little friends. **b**

...bzzz



Share and Publish Twine Stories

In our first two articles on Twine, we discussed how to create Interactive Fiction stories, ranging from simple (text and links only) to more complex (using CSS to add styling and code to include multimedia). In this article, we will learn how to publish stories online and share files with others.

It's important to understand that Twine does not actually host stories and you cannot share your stories using the URLs in the browser version of Twine. However, you have several options for sharing with a select group of people and publishing with the wider, global community.

You can share your stories with a few other people by simply sharing the HTML file. You could email the HTML file as an attachment; share it on a flash drive; upload the file to the cloud and create/share the link; or upload it as an attachment in Google Classroom.

You can publish your story and have the world see it—but remember—do this under the guidance of an adult and with permission. If you go that route, use a Twine-specific hosting service such as **itch.io**. Stories published on itch.io can be either totally private (unlisted and restricted so only the account owner can find them),

unlisted (private but shareable to select individuals), or publicly discoverable through search and browse **(A)**.

When publishing games to itch.io, you need to decide if you want users to have the ability to download your game as an executable file or to play it in the browser. Only allowing browser play will keep users from having access to your source code. If you don't mind players having copies of your work, you can provide the downloadable version of the game **(B)**.

You also can upload your stories online to Google Sites. To do so, you need to first publish your work on itch.

io and then copy your story's itch.io embed code and paste it into embed on Google Sites. From your dashboard in itch.io, go to:

Edit Book -> Distribute -> Embed Game (Direct link)

Back in Google Sites, go to Insert -> Embed and paste the URL of the direct link into the textbox. You can view our game page on Google Sites at **https://sites.google.com/eischools.org/brivera-twine/home (C)**.

You can experiment with Twine by changing anything in our stories' codes to make them your own. You can download our project files online under

Learn More (see Download the HTML project files) at **https://beanzmag.com**.

Once you navigate to those pages on itch.io, you can download the HTML files and then upload them into Twine to test them out. Once in Twine, go to:

Library -> Import -> Choose File **(D)**

Start small by editing the text-based version of the game, Version 1. Try adding your own storyline, adding new passages, and then seeing what happens if you delete passages. Try changing the names of the genres and the books listed in the genres.

Once you feel more confident, move on to Version 2 of the game and edit

the styling. Have fun experimenting because you can always delete the files and start over. **b**



Visibility & access
Use Draft to review your page before making it public. [Learn more about access modes](#)

☐ Draft — Only those who can edit the project can view the page

☐ Restricted — Only owners & authorized people can view the page

☒ Public — Anyone can view the page

Public access settings ×

☒ Disable new downloads & purchases ?

☐ Unlisted in search & browse

A

Look, boss! I got both types of twine! The string AND the stories!

I only needed the stories, but thanks



The Library Book, v2

It's the Friday afternoon before school break. Your parents have to work so your teen sister is "babysitting" you over the break. Sounds like a load of fun. Perhaps grabbing a new book to read over the break isn't the worst idea. You ask your teacher for a hall pass to the library. What could go wrong?

[More information](#) ▾

Download

Download

The Library Book_v2.html 480 kB

B

Embed from the web

[By URL](#) [Embed code](#)

A thumbnail image showing a dimly lit library interior with tall bookshelves filled with books. A small, dark, rectangular object is visible in the center of the image.

Whole page

Show dynamically updating site page.

? Not what you want?

Cancel

Insert

C

New folder

Name	Date modified
Today	
The Library Book_v1	10/16/2022 12:03 PM

File name: HTML Document

Open Cancel

Import Stories ×

To import stories into Twine, upload either an archive or published story file below.

No file chosen

D

The STEAM 2023 Holiday Gift Guide

With the holidays (or a birthday) approaching, you might have someone in your life who is interested in computing—maybe software programming, robotics, or electronics. These STEM/STEAM products make great gifts. You'll definitely find ideas at all different prices and for many different computing interests. Be sure to check out the full guide online at <https://beanzmag.com/steam-gift-guide-2023>.

For more sources:

Wired Magazine's Best STEM Toys 2023 <https://www.wired.com/gallery/best-stem-toys-for-kids/>

The 24 Best STEAM Toys for 2023 <https://www.thespruce.com/best-stem-toys-4153604>

36 Best STEM Toys for Kids, According to Engineers <https://www.goodhousekeeping.com/childrens-products/g5162/best-stem-toys/>

Robots

Little Robot Friends

These small fun robots are very friendly and accessible and an easy way to get kids interested in coding and electronics. Each has its own personality. These cute little robots respond to light, touch, voice, and infrared inputs. They also blink, beep, and buzz. <https://littlerobotfriends.com/> <https://littlerobotfriends.com/collections/little-robot-friends-kits>



Coding

Erase All Kittens!

E.A.K. offers a gentle introduction to professional coding languages, enabling children to play and learn on their own - or alongside parents and teachers. Mario-style gameplay for this online game with interactive dialogue, quirky characters, and kitten gifs makes learning these languages fun for complete beginners and kids aged 8+. It's a blast to play (and learn!). <https://eraseallkittens.com/>



You parents can deal with presents this year, Santa's taking a break!



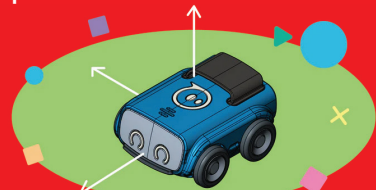
Bitsbox

Kids learn to code by making games on the Bitsbox.com website and then play them on their phones or tablets. The games are quite clever with fun bright graphics. And it's easy for kids to adapt the code once they figure out the game and become comfortable with the code. A new box of games arrives each month. Ages 5 and up. <https://bitsbox.com/>



Indi

Designed for PreK to 2nd-grade kids, this new Sphero robot teaches STEM skills without the need for a computer screen. Cards are used to direct the robot. <https://sphero.com/pages/sphero-indi>



Featured for 2023

Good Housekeeping Amazing Science

This fun book takes you through your home with 83 different science experiments that explain phenomena like stinky socks and mirrors that fog up. It's a fun easy way for kids to see their home in a different light.

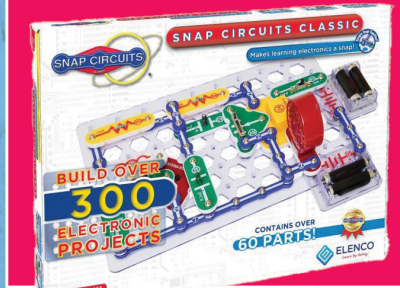
<https://www.goodhousekeeping.com/childrens-products/a36322280/kids-steam-amazing-science-experiments/>



Snap Circuits

This kit lets kids build electronics without having to solder. There's about a hundred different possible combinations. And includes a photo sensor and other fun features.

<https://www.elenco.com/snap-circuits-2/>



Can You Crack the Code?

Security professional Ella Schwartz applies her considerable code-cracking skills to create a fun exploration of how secret codes work, as well as the history of using secret codes.

<https://www.ellasbooks.com/books/can-you-crack-the-code/>



Electronics

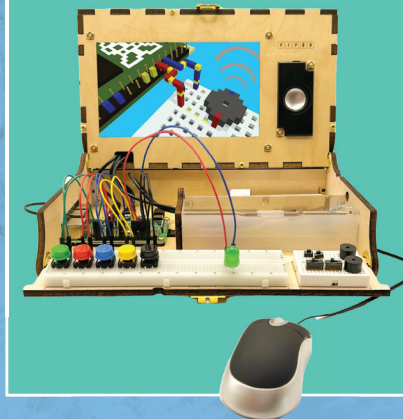
Spintronics

Electrical circuits are everywhere. In our phones, computers, cars, streetlights, and more. But how do they work? Spintronics is a unique game that teaches the basics of electrical flow with a simple board that lets you build circuits. The company also created Turing Tumble which uses marbles to teach how computers work. <https://upperstory.com/spintronics/> <https://upperstory.com/turingtumble/>



Piper

This build-it-yourself computer has evolved into a neat set of electronics projects. You could buy a Raspberry Pi and a bunch of accessories but Piper adds a number of features and makes it easy to build a computer. <https://www.playpiper.com/>



Jewelbots

A unique but really neat idea for kids who want to create friendship bracelets with functionality they can code. <https://jewelbots.com/>



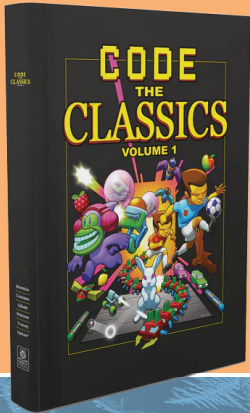
Complete online STEAM gift guide for more ideas at beanzmag.com/steam-gift-guide-2023



Books

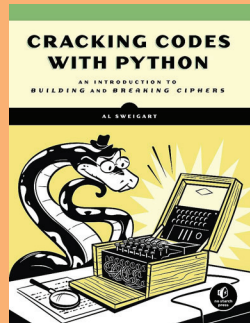
Code the Classics Book

Into retro games? Want to code? The Raspberry Pi Foundation has published a book full of 1970s and 1980s video games, interviews with the creators, and instructions on how to code games inspired by Pong and others from that era. <https://store.rpipress.cc/products/code-the-classics>



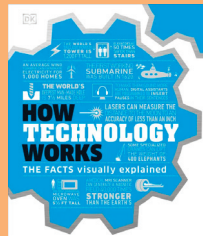
Cracking Codes with Python

This book combines two fun experiences: Python and secret codes. All kinds of ciphers are covered with details about how to use Python to break the ciphers. Could be a fun engrossing sourcebook for someone interested in solving puzzles with code. <https://nostarch.com/crackingcodes>



How Technology Works by Dorling Kindersley Publishing

Have you ever wondered how speed cameras, jet engines, elevators, telephones, MRI scanners, and other technologies work? This amazing book explains visually how almost every human technology actually works, alongside interesting facts. It's the perfect book for anyone who likes to take things apart. And if you like this book, also look up visual dictionaries that provide pictures of common technology and name every part. <https://www.dk.com/us/book/9781465479648-how-technology-works/>



Board Games

Google Cardboard

For \$20-\$30 USD, you can buy and fold together a cardboard headset, then slide in a modern smartphone and use VR apps. While not as immersive as the HTC Vive, the experience is as amazing as more expensive options. <https://vr.google.com/cardboard/> There are a lot more VR headsets and other fun gift ideas to be found in our complete STEAM gift guide online: <https://beanzmag.com/steam-gift-guide-2023!>



Robot Turtles

Robot Turtles is a great board game with some neat extensions like an online community where you can create your own game boards. These games sometimes let you replace the object you direct in the game with a person like your child, or a parent, adding another level of fun and engagement. <https://www.thinkfun.com/robot-turtles/>



Secret Codes

Secret Decoder Deluxe Activity Set

A little less focused on established codes, but has three workbooks that kids can work through and crack the mystery as they go! It's also screen-free fun. <https://www.melissaanddoug.com/secret-decoder-deluxe-activity-set---on-the-go/5238.html>



BY TIM SLAVIN

Code Emulators

Emulators Enhance Language Learning

Learning a programming language can take many forms. There are online classes, for example; you can learn a language through trial and error by creating a software application; or you can buy a book and work through examples.

Online programming language emulators are one way to learn a language and practice how to perform specific tasks. You can't always create a full application. But you can noodle

your way to a deeper understanding of a specific language by using an emulator alongside a course, book, or other resource. Search online for the terms emulator, editor, and compiler to find more.

What makes programming so interesting to me is the lifelong learning part. Programming is more an art than a finite science you learn from beginning to end. Instead, coders learn as much as they want or need. And there are so many ways you can learn how to code, and many languages to explore as well.

Years ago, I asked a senior developer which was better: to learn one language in-depth or many languages. His answer: many languages. The reason is that you begin to see common elements all languages use and, thus, learn more about programming than you would with one language. So don't be shy about playing with more languages than the one you're using today.



Python

The canonical website for the Python language includes an online emulator where you can practice using parts of the language.

<https://www.python.org/>
<https://docs.python.org/3/>



JavaScript

The Mozilla Foundation manages development of the JavaScript language but there are several online emulators available.

<https://playcode.io/javascript-compiler>
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>



Swift

Apple's Swift language uses a Playground app instead of an online emulator.

<https://developer.apple.com/documentation/swift-playgrounds/>
<https://developer.apple.com/swift/>
<https://developer.apple.com/documentation/swift>



C#

The W3 Schools website includes an online emulator and editor: the Microsoft C# (C Sharp) language.

https://www.w3schools.com/cs/cs_compiler.php
<https://learn.microsoft.com/en-us/dotnet/csharp/>



Java

W3 Schools also includes an online emulator for Java and the language documentation can be found on the Oracle website.

https://www.w3schools.com/java/java_compiler.asp
<https://docs.oracle.com/en/java/>



Replit

Replit is one of several online services that provide emulators for many different programming languages. JDoodle is another option. And CodePen.io is a really interesting tool to explore.

<https://replit.com/>
<https://replit.com/templates> (Python, Java, Swift, C#, HTML, CSS)

Groovy



Groovy is an object-oriented language similar to Python, but based on Java and maintained by the Apache Software Foundation.

<https://onecompiler.com/groovy/>
<https://groovy-lang.org/documentation.html>
<https://groovy-lang.org/single-page-documentation.html>

Did someone say "groovy", baby?

Eww, please, no. Just stop.



Tiny Organ Chips

Creating new medications and gaining a better understanding of the human body requires a lot of work by scientists. This research can be particularly hard and take a long time if it depends on human or animal test subjects.

The recent development of tissue chips is making this process simpler and taking animals out of the equation.

Liver on Chips (blech!)

Also known as organs-on-a-chip, these small devices are made up of actual human cells and mimic the behavior of individual organs or systems in the human body. They may look more like a small clear circuit board than a lung or heart, but they are able to get very close to the behavior of actual organs.

The chips have three important properties:

- They hold tiny channels that imitate the systems in our body that bring in blood and nutrients and take away debris

- They are 3 dimensional, just like the human body
- They contain many different types of cells because human organs are not made of just one type of tissue

By measuring how these complex chips respond to different drugs, scientists can use them to test new medications. According to the National Institutes of Health (NIH), these chips are showing the potential effects of drugs on various human tissues in a faster and more effective way than current methods. This is particularly helpful in weeding out drugs with side effects that typically would not show up unless tested on humans.

Space makes haste

The NIH started funding research into tissue chips back in 2010 and quickly saw their potential. After the NIH's initial success, they decided to take a big leap: sending tissue chips into space.

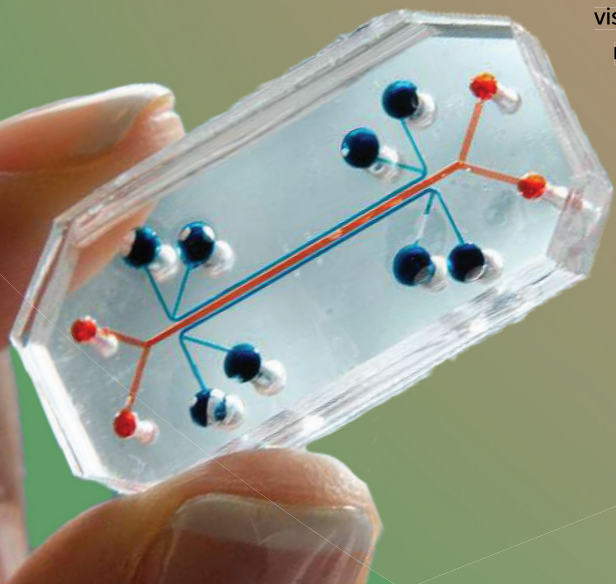
The environment of space is not kind to the human body. From vision changes to bone and muscle loss, microgravity can cause some strange

responses in humans. Some of these changes are similar to the aging process, only they occur much faster in space than on Earth.

While that's not ideal for astronaut health, tissue chips take advantage of this simulated accelerated aging process. Thanks to the Tissue Chips in Space program, many scientists had the chance to launch these devices into orbit. They help scientists learn how different organs and bodily systems respond to spaceflight and test out drugs much faster than is possible in Earth's gravity. In space, researchers can observe changes in a matter of days or weeks as opposed to months or years on our planet. These tested drugs could be used to help future astronauts resist changes caused by microgravity as well as benefit people back on Earth.

The chips sent to space have already studied heart health, joint injuries, the immune system, and much more. Scientists from the Cardinal Heart experiment even confirmed the cells in their heart tissue chips changed as expected in space. They launched a second experiment, Cardinal Heart 2.0, that tested treatments to prevent these changes from happening.

Tissue chip experiments are ongoing with lots of potential to come. **b**



Sharpen Minecraft Skills With RAD 2

If you are familiar with the free-to-play MMO, Runescape, you know your character must reach a specific level of skill (i.e. magic, woodcutting) in order to use the tools related to that skill. Whereas in Minecraft you can use tools freely without meeting goals, which is fun, but less of a challenge. Adding Roguelike Adventures and Dungeons 2 (known as just "RAD 2") to Minecraft changes all that.

RAD 2 adds a Runescape-like system to Minecraft which stops you from using items that are past your level. Found a diamond helmet in the early game? Try putting it on. It will only fall off as your character doesn't have the skill to wear it yet. Weapons and tools are the same, needing specific levels in certain skills to become usable. Working hard to be able to use your tools may not sound like fun, at first. But, if you enjoy this kind of character progression, it can feel very rewarding.

RAD 2 breaks down everything you can do in Minecraft into skills, from attacking to farming to casting spells. And the higher the levels get, the more you can do. This does mean that, from the very start, you'll be stuck with wooden items and leather armor for a bit. And any powerful loot you find will have level requirements before you can use it. But the skill system does make everything you do feel worthwhile.

Sleep Tight or Fight?

Without RAD 2, fighting creatures in Minecraft feels like a chore. You usually only fight them as a necessity; otherwise, you either avoid them or sleep in a bed to prevent them from spawning at night. But with RAD 2, every fight is a chance to level up your skills. You can level your Attack with swords, zap them from afar, and train your Magic, and even if you get hit, your Endurance increases and allows you to wear better armor. As such, you'll always be looking for monsters to fight so you can level up your skills.

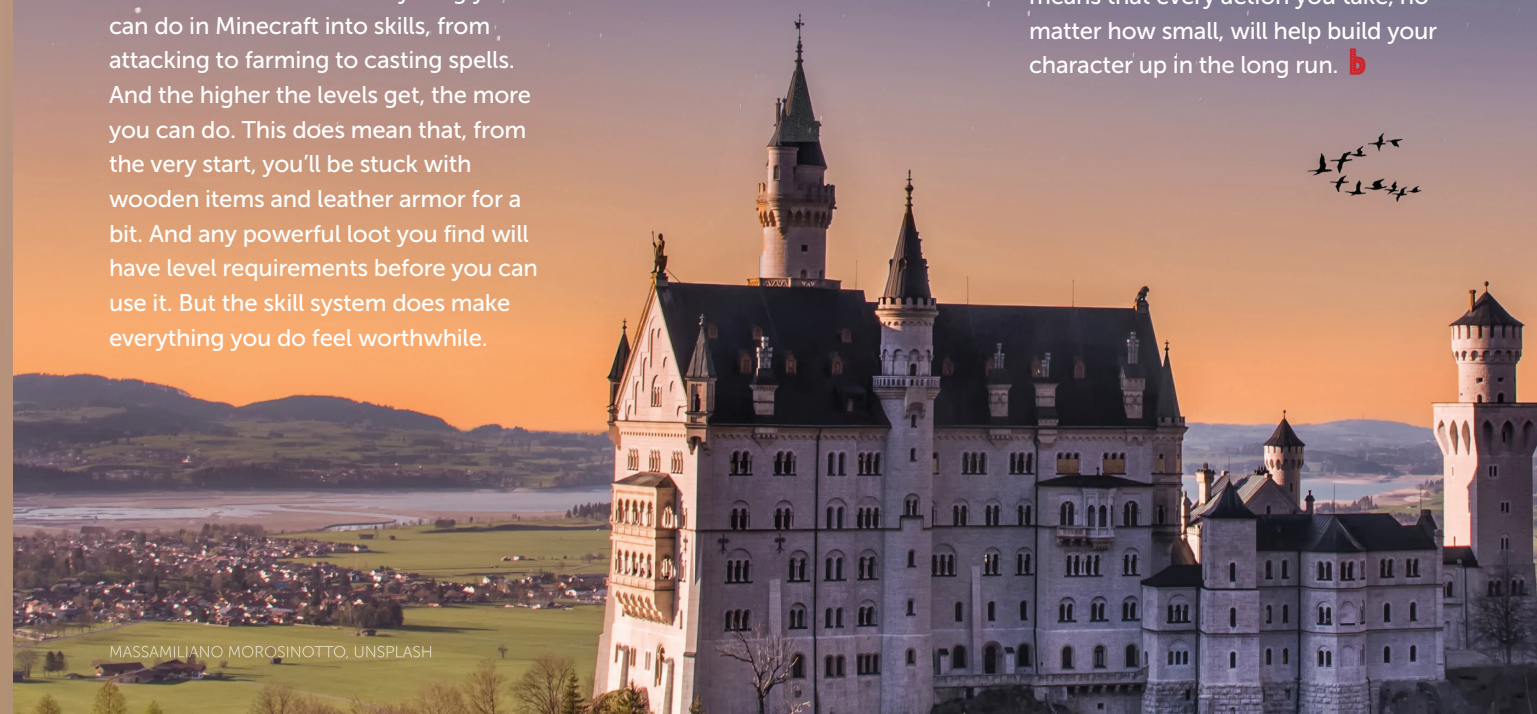
The "Dungeons" part of Roguelike Adventures and Dungeons 2's features mods that generate lots of dungeons, towers, and structures you can raid for loot. And the more you plunder, the more powerful you get through skills alone.

Mod Mixing

The modpack allows you to specialize in a specific skill area. Whichever one you pick, the game will give you a medal that boosts EXP gain in those skills and gives you some nice equipment to get you started. That way, if you want to achieve a specific goal in the modpack, you can hone in on the one area you want to explore and level it up super quickly.

RAD 2 also mixes in some mods to help change things up. It features "Oh the Biomes You'll Go", which adds a load of new biomes for you to explore and venture through. And "YUNG's Better Portals" makes exploring the Nether a lot more fun, as the only way to escape it is to assemble a beacon using gold blocks.

So if you're tired of speedrunning endgame gear within an hour of starting a new world, give RAD 2 a try. While it does slow things down, it also means that every action you take, no matter how small, will help build your character up in the long run. **b**



tidbitz

Inventor Turns Truck Into Dot Matrix Printer

For most people, a dot matrix printer is an ancient technology from the last millennium that used tiny dots of ink to print letters and images on paper. But what if you could turn a truck into a dot matrix printer to print on a road? That would be neat! YouTuber and inventor "Ryder Calm Down" used his truck, a Raspberry Pi, and drops of water instead of messy ink. The result is amazingly good. **b**

<https://makezine.com/article/maker-news/turning-a-truck-into-a-giant-dot-matrix-printer/>



IMAGE FROM MAKEZINE.COM ARTICLE "TURNING A TRUCK INTO A GIANT DOT MATRIX PRINTER"

Ambitious Floating City Designed to Adapt to Climate Crisis

Japan's N-Ark organization is designing a city for 40,000 people that can adapt to the worst parts of climate change. Dogen City will integrate medical, food, and housing in one location. The project attempts to adapt to ocean environments for architecture, agriculture, and aquaculture. **b**

<https://www.n-ark.jp/en>
<https://mymodernmet.com/n-ark-dogen-city/>

Ahh! Home sweet home!

THIERRY MEIER, UNSPLASH

OH NO! They're going to find out my secret powers!

Leaping Squirrels!

Videos of squirrels leaping from branches across impossibly large gaps are being studied by University of California, Berkeley, researchers to understand how to train robots to perform the same agile feats. Squirrels make leaping decisions in split seconds as they move through trees. **b**

<https://news.berkeley.edu/2021/08/05/leaping-squirrels-parkour-is-one-of-their-many-feats-of-agility/>

YIRAN YANG, UNSPLASH

Hydrogen Carcopter

The pod racer from Star Wars Episode 1 The Phantom Menace has been created (for real) by a French startup, Maca Flight. It uses a hydrogen fuel cell instead of a battery and looks amazingly similar to the Star Wars podracers. It has three pairs of rotors and six motors, all of which use hydrogen to generate electricity. The racer is 23 feet long and has a top speed of 155 mph. **b**

<https://interestingengineering.com/innovation/this-hydrogen-powered-carcopter-is-a-real-life-star-wars-podracer>



MACA
FUTURE IS IN THE AIR

STAR WARS

MACA LOGO: SOURCED FROM MACAFLIGHT.COM
 STAR WARS LOGO: WIKIMEDIA COMMONS



Solar Power From Space

JAXA, the Japanese space agency, has spent years trying to perfect the beaming of solar energy from space down to Earth. In 2015, its scientists beamed 1.8 kilowatts—enough to power an electric kettle—50 meters to a wireless receiver. In 2025, they plan to beam solar energy from small satellites to ground-based arrays hundreds of miles away. **b**

<https://www.engadget.com/japan-will-try-to-beam-solar-power-from-space-by-2025-214338244.html>
<https://www.umass.edu/news/article/new-green-technology-umass-amherst>
<https://onlinelibrary.wiley.com/doi/epdf/10.1002/adma.202300748>

SUN: WIKIMEDIA COMMONS
 KETTLE: ENGIN AKYURT, UNSPLASH

Robot Fish Helps Researchers Protect Ecosystem

Swiss researchers and mechanical engineers have created an autonomous fish robot named Belle that collects data about undersea organisms and their environment. The robot allows scientists to observe life underwater in a way that doesn't scare fish or harm marine life the way boats, propellers, and submarines can. **b**

<https://www.reuters.com/video/watch/robotic-fish-collects-dna-samples-in-the-idOV335228062023RP1>
<https://www.euronews.com/next/2023/06/30/belle-the-inobtrusive-ai-robot-fish-is-helping-researchers-to-protect-our-marine-ecosystems>

I'm already a robot fish! I don't see the big deal here.



STEVE MOHUNDRO, FLICKR

Learning One Mile Deep

Recently, our 6th-grade team embarked on a journey where we didn't make it very far. To be fair, we didn't intend to go anywhere but the 6th-grade wing of our building. Our journey was more like a scrimmage of the mind and was an eye-opening experience for students and educators alike.

For the past few years, our team has replaced our final exam week with an interdisciplinary Project-based unit on space exploration. One way I've heard it described is that, unlike normal classroom content which is a mile long and one inch deep, Project-based Learning (PBL) is one inch long and a mile deep. Immersive education of this sort aims to cover less content but delves into all angles of a subject. While these types of units can be tricky to plan, they also can produce some authentic and meaningful learning opportunities for students.

For our end-of-year unit, we developed a narrative where NASA was soliciting proposals for a Mars mission. Students had to develop a three-phased proposal for creating a settlement on Mars considering many factors like payloads, launch windows, and laws for the colony.

Students' initial attitudes were very relaxed. Humans were on the moon almost 50 years ago; Mars can't be that much more difficult, right? Well, students quickly found out that although a mission to Mars has some similarities to the moon, it is very much different. Using the great Kurzgesagt video, Building a Marsbase is a Horrible Idea, teachers introduced

students to Mars' dust storms, radiation, and frigid temperatures.

We also took a field trip to our local Science Center to see an exhibit on Mars and take notes. The interactive displays on everything from food production to spacesuit technology were a great context for the overall project. We also contacted CMU, our local technical university, and had a speaker visit who works on developing nanosatellites that could one day be deployed on Mars.

By the end of the week, students had researched the hurdles, devised solutions to these problems, built dioramas of their Mars base, drafted a series of laws, and developed a proposal for obtaining water, oxygen, and fertile soil. They gave presentations in front of the entire school for their proposals and a winning proposal was chosen by teachers and administrators.

All in all, it was a successful week, but it wasn't always smooth sailing. As I find with many interdisciplinary projects, the teachers learned just as much as the students. Sometimes well-planned modules really don't make a strong impact, while other more off-the-cuff activities produce some fascinating results. Other times, instructions weren't as detailed as they could be which left room for some groups to stray from their objective and get distracted.

While it can be a challenge to plan this kind of unit, there are a few ways to increase the chances of a fruitful project. First, it really helps to make PBL as different from

normal school as possible. Changing up the schedule, taking field trips, and inviting speakers to come to the school can make learning more immediate for students. Incorporating workshop time where students are given the agency to create in a new environment also has produced some great results. It can be hard to let students work independently, but I've found they can really surprise you when you give them a challenge and the space to work on it.

It also helps to have a strong real-world narrative component to the project. For the whole week, you are creating a tiny world or alternate reality for students. Small things like handing out roles, dressing the part, and holding morning stand-up meetings in character can help to break out of normal school mode. In a way, PBL mimics the model of work in most technology companies where challenges are posted, deadlines are set, and natural consequences work as the primary motivator. Similarly, wrapping up the unit with a presentation or showcase in front of an audience can be quite a motivator for students.

Immersive education, despite its difficulties, can be a great way to capstone a term or a unit. It gives the students the opportunity for application and creation in a unique atmosphere outside the normal school mindset. It also puts the ball in the students' court and gives them the space to create something meaningful, which really is the ultimate goal. **b**



Give a gift of **beanz** magazine at 30% OFF!

What better way to learn than with a childhood friend that generates lasting memories?

Use code **GIVEBEANZ** when subscribing

<https://subscribe.beanzmag.com>



“Sometimes it’s better to leave something alone, to pause, and that’s very true of programming.”

—Joyce Wheeler

Thank you for reading this issue of **beanz**! Check out the links below to read stories from this issue online with links to learn more.

Designing Space Art That Spans Generations

<https://beanzmag.com/designing-space-art-badges>

Ghost Hunter

<https://beanzmag.com/code-ghost-hunter-game>

Can ChatGPT Code?

<https://beanzmag.com/can-chatgpt-code>

Is That You Or Is It AI TTS?

<https://beanzmag.com/how-voice-cloning-works>

Ancient Mayan Math

<https://beanzmag.com/mayan-numerals-explained>

Poppy Northcutt

<https://beanzmag.com/frances-poppy-northcutt-nasa>

Techy Ocean Cleanup

<https://beanzmag.com/cleaning-oceans-with-tech>

Your Chariot Awaits!

<https://beanzmag.com/chinese-south-pointing-chariot>

Gear Up For Ghost Hunting

<https://beanzmag.com/ghost-hunter-technology-equipment>

Robot Bees

<https://beanzmag.com/robo-bee-technology>

Share And Publish

Twine Stories

<https://beanzmag.com/publish-twine-stories>

STEAM Gift Guide

<https://beanzmag.com/steam-gift-guide-2023>

Code Emulators

<https://beanzmag.com/programming-language-emulators>

Tiny Organ Chips

<https://beanzmag.com/organ-on-a-chip>

Sharpen Minecraft Skills with RAD 2

<https://beanzmag.com/minecraft-roguelike-mods>

tidbitz

<https://beanzmag.com/october-2023-news-wire>

Learning One Mile Deep

<https://beanzmag.com/taveling-to-mars-coding>

Sssssssubscribe!
It'ssss Sssso Worth It.



<https://subscribe.beanzmag.com>