mathNEWS

VOLUME 146 • ISSUE 6

JULY 30, 2021

9 770705 041004

mastHEAD

"WHAT IS ONE THING YOU'RE LOOKING FORWARD TO DOING IN THE FALL TERM?"

The fate of **mathNEWS** in the term to follow feels uncertain, up in the air. I've only felt this way once before. The fifth issue of Volume 142 was published on March 13, 2020. That was the day the University announced that all on-campus activities would be suspended for the week of March 14 to March 23. Last Fall, I had an opportunity to enter MC to do some administrative business for **mathNEWS**. On the display stand outside our office, there was a neat stack of issues, about a foot thick. I wonder how many people ever got to pick up a physical copy of 142.5 — the last issue to date to even have a physical copy. But that stack will still be there come September, and depending on when / if we can work out printing paper issues again, it might stay there for even longer.

I was returning as an editor for Volume 143, and at the time, I had no idea how the term was going to be. I didn't know if RDP had been enabled on the **mathNEWS** computer and if we were even going to be able to put together issues. I didn't know how production nights would look. I didn't know if anyone would write for us. Back in March, the future of **mathNEWS** — and the future of everything else — seemed so impenetrably murky.

I will be returning as an editor for Volume 147, and **mathNEWS** has once again found itself in the dark about how the coming term will shake out. Will there be printed issues? Will there be in-person production nights? Will there be ***pizza***?

To paraphrase something I said in the last issue of Volume 143: looking back, all my worry was for nothing. In case you couldn't tell, **mathNEWS** has been doing just fine these last 4 volumes. Amazing, actually. And it's all thanks to each of our wonderful contributors. They make **mathNEWS** the funny, clever, geeky rag that it is. There's no **mathNEWS** without them.

Barring a "😹 — Cat with Tears of Joy" variant on our hands in the near future, it seems that Ontario is slowly returning to the way things used to be. There *will* eventually be a day where we'll hold an in-person production night again, with pizza, even. I hope you'll consider coming. Me? I wouldn't miss it for anything. Till then,

clarifiED
Editor, **mathNEWS**

CC	Honestly? Just going out with friends in Waterloo.
Deriving for Dick	Waking up and realizing I should've been on campus by then.
Predap	Finally moving back out and back to Waterloo
boldblazer	Online courses again
me	Play minecraft 1.17 with **mathNEWS**
aphf	Hunting and gathering my own sustenance in the wilds of Waterloo.
tendstofortytwo	Licking people's faces without a mask again.
cy	Releasing my **mathNEWS** recipe book woooooo
A cool pen name	Going to in person prod nights at jeff's house and hanging out with his cats
jeff	Defending my house from unsolicited prod nights
Abald Man	In-person classes *nyah*
Finchey	Ordering the largest shawarma platter at iPotato with ten lines of spicy sauce, then shitting myself to death the morning after.
waldo@<3.LE-GASP.ca	Hugging my friends <3 [I miss people a LOT]… and maybe an IN-PERSON **mathNEWS** production night?!
Narf Dert	Infiltrating Golden Words for the glory of **mathNEWS**!
In-Person MATH 136 Final Exam	Row row row reduce Gently down the page! One mistake and you'll be baked, So check at every stage!
bipED	Autumn
god ⚡ peED	Enjoying Shawerma Plus in its sacred homeland.
clarifiED	Seeing friends!
terrifiED	Row row row your boat Gently down the stream Merrily merrily verily merrily In-person school finally isn't a dream

ARTICLE OF THE ISSUE

Ah, Article of the Issue, the highest, fortnightly honour we mighty editors can bestow. Well, this time around, it goes to jeff for C++ is just Racket, if you think about it (parts 1 *and* 2)!

clarifiED
Editor, **mathNEWS**

I'm looking forward to licking strangers' faces too.

CLARA XI, mathNEWS EDITOR FOR SPRING 2021
ALONG WITH TERRY CHEN, KEVIN TRIEU, AND YANG ZHONG

mathASKS 146.6

FEATURING mathNEWS *EDITOR BIPED*

CY: WHY IS mathNEWS THE BEST UNIVERSITY PUBLICATION EVER?

A publication is nothing without contributors, and it's always the writers and artists that make **mathNEWS** the best university publication ever. That said, I think **mathNEWS**' low barrier to entry and freeform structure let people express themselves more fully than other university publications. There is no fixed idea of what **mathNEWS** should be like, so it doesn't get stale or stuck. Personally, I like how every **mathNEWS** writer has a style that shines through.

TENDSTOFORTYTWO: WHAT'S THE BIGGEST UPSIDE AND DOWNSIDE OF BEING AN EDITOR?

Upside: It feels like real work. I find that in my computer programming jobs I have always felt disconnected, like I didn't know what or who I'm really doing this for, even when I work for companies I really respect and on work I really enjoy. I know who I edit **mathNEWS** for, and all of the annoyances that we deal with are for the readers and writers. There's a sense of concreteness and fulfillment to this kind of job.

Downside: I have to use Adobe products.

A COOL PEN NAME: HOW DO YOU MOTIVATE YOURSELF TO DO THINGS?

I'm not qualified to answer this question. There are things that are easy for me to do (like **mathNEWS** editing work), and it's easy for me to spend hours doing those things. For other things (like doing PD assignments or writing e-mails), I can burn a whole day procrastinating and get nearly nothing done. I try to do something I find fulfilling every day, and I try to accept that hard things will take me a longer time than I expect.

For creative endeavours, try enforcing deadlines on yourself. **mathNEWS** is good for this. But don't take it too hard if you fail, it's normal. And if you're really suffering when you try to do something, try taking a step back and thinking about why you feel that way. Generally, it's good to keep in touch with what you're thinking and how you're feeling, and notice how those things relate to each other.

But really, I'm not qualified; read On Advice by cy.

CLARIFIED: WHAT IS YOUR FAVOURITE CLASS YOU'VE TAKEN SO FAR AT UW?

It's either CS246E or CS365 (with Eric Blais, in particular). I think if I had taken the latter course in person it would win handily, but as it is CS246E was more fun and compelling just because I took it IRL. I think of CS246E as the most fun CS course I've taken, and CS365 as the most fun math course I've taken. I highly recommend both.

CLARIFIED: WHICH ONE OF THE OTHER EDITORS IS YOUR FAVOURITE?

terrifiED, for introducing me to **mathNEWS**, and not because he's the most senior and powerful or anything like that.

ABALD MAN: RANK THE TOP 10 mathNEWS WRITERS IN ORDER.

1. CC, with two articles containing the word 'farticle'
2.
3.
4.
5.
6.
7.
8.
9.
10. Everyone else, with zero articles containing the word 'farticle'

JEFF: RANK THE TOP 10 mathNEWS READERS IN ORDER.

1. You, the reader <3
2. Me
3. The **mathNEWS** WordPress conversion script
4. That person I saw in RCH once
5. Reddit user uw_raccoon
6. Mark Giesbrecht
7. Every fictional character in **mathNEWS** who reads **mathNEWS**
8. Everyone else who has submitted anything for **mathNEWS**
9. Everyone else who has read at least one **mathNEWS** article
10. Everyone else who has at least glimpsed a **mathNEWS** issue

BOLDBLAZER: WOULD YOU CHANGE YOUR EDITOR NAME IN THE WORST CASE SCENARIO WHERE YOU LOSE ONE OR MORE OF YOUR LEGS?

I don't think there's precedent for a **mathNEWS** editor changing their name. If something like that happened to me, I think I will ask to swap my name with another editor's.

πLLOW PRINCESS: WHERE IS YOUR FAVOURITE TOILET ON CAMPUS?

I haven't been to enough of them to really know. I remember 1A, when I was taking the GO bus home. I had walked from DC to SLC and back to buy a ticket, and it was foggy and SLC was full of mud from construction. I waited for about 20 minutes in the DC lobby, and I remember going to the long and very disgusting toilets behind that big round room, it was either room 1934 or 1935. I would say that those are my favourite.

CIX: WHAT IS SOMETHING YOU REGRET?

I regret forgetting my native language. I came to Canada with a decent-for-my-age grasp of it and turned my back to it afterwards. I can still dredge up some everyday words out of some primal instinct, usually if someone speaks to me in that language while I'm distracted. I wish I had put in the effort to maintain my ability fully.

CC: IF YOU COULD ONLY WRITE ONE MORE PIECE OF WRITING IN THE REST OF YOUR LIFE, WHAT WOULD YOU WRITE?

I'd like to say "I have a fantastically large project that I've been mulling over for the past few years, whose creation will take me a very long time. I would work on that and retire satisfied", and I do have something like that. I think. For me, admitting that I'm working on some project when that project is still in its infancy feels a little distasteful. So I won't give any more details than what I've written here… maybe you're reading this some time in the future and now know what you're talking about… I know what you're looking for… Check the display Bible in St. Jerome's……

BEYOND META: FAVOURIE WHIMSICAL MODE OF TRANSPORTATION?

Long-distance train rides. As a child I once took the train from BC to Ontario; the trip took several days and I don't remember much, and I would likely get severely motion-sick if I took it again, but it was an experience that really stuck with me. I also took a train from Ottawa to Toronto, leaving at the crack of dawn, in my first year to get to UW, which invoked feelings in me that are literally indescribable. I've also taken some trains in Asia—those rides were shorter, either due to distance or speed. I can see why some people hold deep interest in trains.

ME: WHAT IS YOUR FAVORITE ON CAMPUS EATERY?

You know, I always meant to go to more of these places, but I never managed to. I guess I got donuts from the SCH Tim Hortons a few times? Does that count?

Doesn't being a mathNEWS editor seem fun? Don't YOU want to have fun? Apply to be an editor today!

A mathNEWS EDITOR WHO IS
~~DYING INSIDE~~ DEFINITELY HAVING
LOTS OF FUN RIGHT NOW

A COOL PEN NAME: WHOSE BLOOD DID YOU HAVE TO SACRIFICE TO BECOME mathNEWS EDITOR?

:)

BOLDBLAZER: IF "BIP" WERE A VERB, WHAT DEFINITION WOULD YOU GIVE IT?

To exist for a short moment.

PREDAP: ARE YOU FEATHERLESS?

AFAIK.

mathNEWS EDITORS IN A K-POP GROUP

blackBOX is a co-ed 4-member group under **mathNEWS** Entertainment. The members consist of terrifiED, clarifiED, god⚡peED, and bipED. The debuted on May 21, 2021 with their EP "**mathNEWS** With You."

Fandom Name: editED

Fandom Colors: pastel pink & black

Member Profiles:

terrifiED:

- Producer
- Center
- Sub Vocalist
- Face of the Group

clarifiED:

- Main Dancer
- Sub Vocalist
- Visual

god⚡peED:

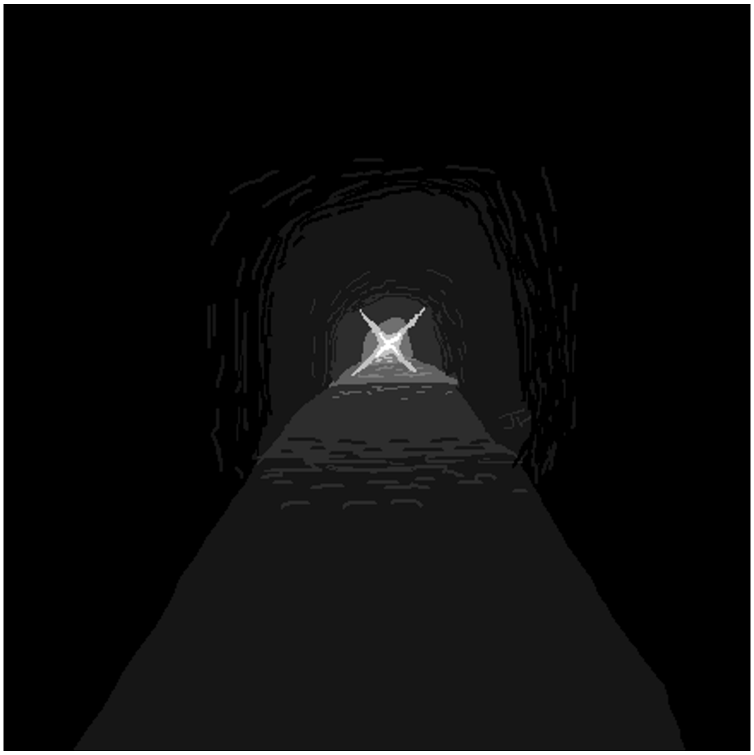
- Main Rapper
- Lead Dancer
- Lead Vocalist

biPED:

- Main Vocalist
- Lead Rapper
- Maknae

Deriving for Dick

LIGHT AT THE END OF THE TUNNEL



DRAWN BY ME

2020 for me has to be a wash. I remember every single day felt like a year. And now the entire year of 2021 has felt like a day.

Do you remember how it started? I remember the overwhelming dread, every minute; every day a tsunami of terrible events traumatizing us concurrently. From attending club meetings one day to a week later where everyone is placed on house arrest. Events cancelled, lectures moved online, co-op positions on hold, libraries closed, borders closed. Empty sheaves, Empty buses, Empty restaurants to Empty residences. Then comes mask mandates, social distancing, indoor limits, and more restrictions.

Then there was the new normal. We thought it's only going to be 6 weeks. I, a sweet summer child back then, believed we would be back by summer or Fall 2020 at the latest. Didn't happen though, agonizingly watching the curve to bend, to fall. It took some time but our wish did come true. The number was down in July, the government started to ease restrictions. Maybe we were going to be back in the fall, after all, Maybe that was light at the end of the tunnel.

That light was an on-coming freight train. Without a vaccine, without herd immunity, managing this pandemic is like managing a nuclear reactor. Covid spread exponentially, like supercritical fission. The restrictions and masks were the control rods. The harsher the rules the slower the spread, the reaction. But then you destroy the economy. Therefore, every level of government had to play the balancing act of making the pandemic manageable to hospitals while keeping the economy alive. That was the problem — we become too greedy, we become too complacent, they were too incompetent. When the pandemic heated up again during the fall, our leaders reacted retroactively instead of being proactive. That brings us to the second wave.

During the first wave, I was terrified, but part of me was also intrigued to experience history in the making. It's fascinating to live a reenactment of *Contagion*. Again, the government locked everything down. Walking in the empty streets, looking at the boarded-up buildings, I was downright depressed, disappointed, isolated, hopeless. I become extremely unhealthy, being a social recluse without any means for me to stay connected. Friends I made in university — slowly losing them, becoming the people I used to know. I experienced the loneliest Christmas and New Year ever in my life.

All was not lost yet. The vaccine research was at an accelerated pace. By February 2021, many vaccines were already approved. The cases started to drop again. Once again, I could see the light at the end of the tunnel. But new variants came. The vaccine rollout was slow because we lacked foresight. We repeated the same mistake from last year. The third and the worst wave happened. More panic, more lockdowns, watching this movie too many times. I wished I would get Covid and get it over with.

The vaccines came in clutch, just in time to save me from a permanent solution. My eyes were glued to Vaccine Hunters. Finished my compo combo PZ/MD shots by July. Right now, I can feel America in my veins. And now over 70% of the eligible are vaccinated. Finally, the light at the end of the tunnel?

Honestly, I would not dare to guess. The Delta and Lambda variants are ravaging previously vaccinated countries. The fall term is still mostly blended or online. Even if we recover by end of this year, we've lost too much time and opportunities we would've had if we were on campus. Many of us never get to say goodbye. Even then, nothing compares to the unlucky ones who lost their loved ones in one of the worst disasters that happened in this century.

As we exit this tunnel, we enter another, much deeper tunnel as Lytton BC of all places, reached 50°C *in May*. When we look up to see the blood-red moon tinted by the smoke from the west, and the Covid which tormented us for the last 2 years: both are reminders of the consequences of the exploitation done by humanity.

The old normal has died, long live the new normal.

me

C++ IS JUST RACKET, IF YOU THINK ABOUT IT (PART 1)

Going into third year soon, I can't say I haven't yearned for the days of Racket and Haskell more than once. After all, pure functional languages are where the real beauty of programming lies. I mean, just look at this:

```
((λ (y) ((λ (x) (+ x y)) 3)) 4) ; ⇒ 7
```

Abstract. Beautiful. Inspired. Elegant. Oh, the days of not having to worry about nasty things like “side-effects” and “writing programs that actually *do* things”. But we can't go back. Nowadays, it's C++ that's all the rage—an imperative, object-oriented language. Blegh.

But really, C++ is only imperative for the closed-minded. If you think about it, C++ is basically just Racket, right? …No? In fact, C++ does a remarkable job at being a pure functional programming language. And we don't even need to use the fancy constructs like if clauses, functions, or even variables, to get it! What remains, then, you ask?

Templates! Templates are a beautiful feature of C++ that let you abstract out certain details of structs and functions (like types and certain constants), and “fill in the blanks” at compile time.

```
template<typename TypeToSub>
struct Node {
    TypeToSub data;
    ListNode* next;
};
```

This is the generic linked-list structure you've probably seen in a first-year CS course, but it has the “type” of the data abstracted out as a parameter. So, **Node** is now the name of a template struct that takes one parameter. If we wanted to declare a linked-list of integers called “x”, we would write something like “`Node<int> x`”. The type of `x` is `Node<int>`, and this type is created at compile-time. The compiler will basically take the code for the `Node` template, replace all instances of “`TypeToSub`” with “`int`”, and name it “`Node<int>`”.

These templates can also be used on functions, but we won't use that here. One thing that we *will* use is the ability to write *template aliases*, i.e. a template in conjunction with the *using* keyword.

```
template<typename T>
using NodeAlias = Node<T>;
```

Racket is nothing but lies.

BRAD LUSHMAN

A last important detail is that templates can take value parameters, not just type parameters. Here's an example of a template struct that takes both kinds of parameters, making them accessible through a member type alias and a constant member field in the struct:

```
template<typename T, T N>
struct Integral_Constant {
    using type = T;
    static const T value = N;
};
// ...
Integral_Constant<int, 3> x;
```

So, now, the integer “3” now becomes a type. What? Yeah, a type. `Integral_Constant<int, 3>` is a type, on the same level as our old friends `int`, `float*`, and so on. Moreover, we can access its parameters through the “type” and “value” defined for it. In particular, `Integral_Constant<int, 3>::type` will be an alias for `int`, and `Integral_Constant<int, 3>::value` will be the constant `3`. Trying to instantiate these types in variables/objects is kind of pointless, since they won't do anything useful. This might seem like a strange concept, but it's really important for what's to come.

To wrap up this first part, we introduce the central theme of “tricking the compiler into doing big computations for us”. We'll illustrate this by writing a template that lets us compute N! (N factorial) at compile time. That means the compiler does all the work for us, and bakes the final result into our compiled program. By definition,

$$N! = \begin{cases} N(N-1)! & \text{if } N > 0 \\ 1 & \text{if } N = 0 \end{cases}$$

So we'll probably want our computation to follow a similar recursive structure. Let's see the recursive case:

```
template<int N>
struct Fact {
    static const int value = N * Fact<N-1>::value;
};
```

That's not so bad! To figure out what the constant “value” should be for `Fact<N>` at compile time, the compiler is forced to figure out what the constant “value” should be for `Fact<N-1>`, and it'll keep having to figure this out until it gets down to the base case, `Fact<0>`. But we haven't actually written anything to get the compiler to stop at 0 yet, so this won't work. How can we do this? Well, C++'s template facilities allow for a form of primitive pattern matching through *template specialization*. Basically, we can “override” the definition of a template for specific parameters. Here's how we do that here, for `Fact`:

```
template<>
struct Fact<0> {
    static const int value = 1;
};
```

It's very literal and we put the “specialized” parameters after the struct name, mimicking what kind of “form” we want to match against. The brackets after “template” are empty because there are no parameters now—they've all been specialized. The compiler will pick the most specialized option available, based on the given parameters.

So that's all we need for factorial! We can now compute N! completely at compile time, affording us instant run time. It's important that the N we give is some constant expression that can be determined at compile-time too, but this is still really powerful. Here's how we'd compute the factorial of 10:

```
    const int x = Fact<10>::value;
```

With our template, the compiler will do all the work computing 10!, and after it figures out `Fact<10>::value`, it'll embed the result directly in its place, i.e. 3628800.

There's something weird but interesting here: “`Fact`” really is just a struct. It's a structure, and we usually use those for storing and grouping together data in some useful way. Here, though, we're treating it more like a *function*, where the arguments are given inside angle-brackets, and `::value` evaluates the result. From now on, we'll call this sort of structure a “meta-function”. And since this is evaluated at compile time, we have no side-effects. Curious!

I haven't actually gotten to explaining how any of this relates to Racket yet. There are some flavours of Racket like recursion, a loose notion of pattern matching, and the promise of no side-effects, but we have yet to relate it to Racket in a concrete way. There's a whole rabbit hole you could go down with template metaprogramming, but we'll just go over what's needed to uncover the truth of a pure functional programming language at the core of C++. I think we have that now, so in the next part we'll get into how this all relates to a pure-functional language, and how C++ is really pure functional.

jeff

Midterms are like bears. If you are chased by a bear, you don’t have to run faster than the bear, you just have to run faster than your friends.

PROF. DAVID WAGNER

profQUOTES

CS 240: MARK PETRICK

> “ This is so cool that I am writing it in red.

CS 251: STEPHEN MANN

> “ The valid bits of the cache got absorbed onto the ink on the paper.

ENGL 306A: CLIVE FORRESTER

> “ I actually call [course evaluations] the “customer service exit survey.”

MATH 239: MARTIN PEI

> “ If you find a fast algorithm for solving [the Traveling Salesman Problem], let me know secretly and together we can take over the world.

LAST CALL FOR RECIPES

Hello to all **mathNEWS** writers who like food,

I am looking for recipes from **mathNEWS** writers. Please send a recipe that is meaningful to you, along with your writer name, to mathnews@gmail.com. For example, a recipe for a personal “comfort food”. It does not actually have to taste good, but it must be food. Shorter recipes preferred. You can write the recipe yourself or send it from a link. Optional: you can also include a few sentences describing why this recipe is meaningful to you.

By sending me the recipe, you agree for me to put your recipe, or a shortened version, and your few sentences, or a shortened version, in a future **mathNEWS** article. Your **mathNEWS** name will be credited. But don't get too excited yet; you won't see your article until the next on-campus volume (fingers crossed for fall!).

I want recipes from those who have written at least one article. If you have not but wish to submit a recipe, you can send it to me anyway and write an article next issue.

cy

THE FUNNIEST JOKE I'VE EVER TOLD

“Hey, I think I'm gonna write a **mathNEWS** article today.”

A cool pen name

A BRIEF OVERVIEW OF MY BOOKMARKS TOOLBAR

THINGS I FOUND USEFUL AT ONE POINT, SO YOU MIGHT AS WELL

It's no secret to anyone who's seen me using a web browser that I have my fair share of bookmarks, which I keep in a permanently visible Bookmarks Bar on the top of my screen. Call me a boomer, I just like the comfort of them being there for me when nobody else is. Although people have often marveled at my collection, neatly arranged into folders such that the list doesn't spill off the screen in either the Y or X direction, none have asked me what bookmarks I actually have in there.

I presume you all have been just too shy, so here's what you've always wanted to know; a bunch of links that I keep near to my heart in hopes that they will come in handy when I least expect them. Note that I only mention links that I figure will be useful/interesting to the average UW math student.

GAMES

- **BlueMaxima Flashpoint (**https://bluemaxima.org/flashpoint/**):** Ever wondered what happened to your favorite Flash game after Adobe retired the Flash Player? If you're lucky, the game found refuge in BlueMaxima's Flashpoint. Flashpoint is a Flash game player and archive, where you can download all the games you used to enjoy, and play them in a secure environment that probably shouldn't give you viruses I think. It's a great way to relive your childhood if you spent any significant time playing online Flash games, check it out when you can!
- **QWOP (**http://www.foddy.net/Athletics.html**):** In this harrowing sports action game, you play as the sole representative of your small country in the Olympics. The struggles you faced to get to this worldwide stage are nothing compared to what you will see at the 100m race event. Stretch your calves and thighs, this is going to be one hell of a run. I hope this description makes it sound like some sort of a blockbuster AAA game, because while QWOP definitely isn't that, it really should be.
- **2048 (**https://play2048.co/**):** It's that game that went viral a few years back. Swipe numbers on the board in different directions, to add them together to form powers of 2. Can you reach 2048? I sure can't, but it's fun to keep trying (even if my assignment deadlines loom hauntingly close as I continue to waste time doubling numbers…).
- **Cubed (**https://web.archive.org/web/20200204013820/http://rappdaniel.com/experiment/cubed/**):** You'll notice that this is a Wayback Machine link, and that's because sadly the original domain name seems to have expired. God bless the Internet Archive though, because they preserved this fun 3D shape driving/avoiding game which looks like it was someone's (frankly awesome) intro-to-game-design project. The social media/live scoreboard features don't work, but honestly I think that's a good thing. The game is super enjoyable, though I will mention that it had terrible framedrops, and eventually caused the Firefox tab it was playing in to crash.

FUN

- **Boil The Frog (**http://boilthefrog.playlist-machinery.com/**):** The title (and the meaning behind it) are uncharacteristically gruesome (though I guess an accurate description) for what this tool is — it takes two songs as input, and creates a playlist that starts at song 1, and then plays songs that slowly transition from the style of song 1 to the style of song 2, ending with song 2 itself. It's surprisingly good at what it does, and in my experience a great way to discover new music based on two songs you already like.
- **Digitally signed Windows XP AHCI/RAID drivers (**https://www.win-raid.com/t11f23-Modded-Intel-AHCI-and-RAID-Drivers-digitally-signed.html**):** Most somewhat-modern computers use SATA hard drives. These can work in two formats — the older, slower IDE format, or the newer, faster AHCI format. Windows XP does not contain the right drivers to use the AHCI format by default, nor does it contain the drivers to use a multi-drive configuration via RAID. So if you want to use Windows XP on a modern computer, this will be an essential resource for you to get the maximum performance from your system.

LEARN

I have links to *way too many* learning resources (all of them CS-y, feel free to skip if you're not CS) that I found and then never used, so I will group them rather than mention each one individually:

IDEAS FOR SIDE PROJECTS:

- https://github.com/florinpop17/app-ideas/blob/master/README.md
- https://what-to-code.com/
- https://github.com/danistefanovic/build-your-own-x/blob/master/README.md
- https://github.com/tuvtran/project-based-learning

LEETCODE/CONTEST PROGRAMMING:

- http://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf
- https://cses.fi/book/book.pdf

STUFF I REALLY WANT TO LEARN/MAKE AT SOME POINT:

- https://littleosbook.github.io/
- https://en.wikipedia.org/wiki/ELIZA
- https://rustwasm.github.io/docs/book/introduction.html
- https://book.systemsapproach.org/

ENTERTAINMENT

- **The Story of the Vox Machina fansub (**https://www.youtube.com/watch?v=kKixWUIgmKA**):** A friend suggested that I start watching Critical Role, a bunch of people playing Dungeons and Dragons for fun. Apparently this is a way to catch up to the story so far. I tried watching it, honestly didn't vibe with it much. But the quality seems great, so if D&D is your thing, maybe check it out!
- **FicSave (**http://ficsave.xyz/**):** Ever wanted to read fanfiction from the comfort of your ebook reader/an airplane/someplace that doesn't have internet? I did, and that's when I found this site — it converts your online fanfiction into a readable file. I converted the Albus Potter fanfiction series by NoahPhantom to EPUB and read it all on my Kindle with this tool. Highly recommend that series, by the way. It's got a fantastic storyline, worldbuilding, original characters, and plot.
- **Debugging Stories (**https://github.com/danluu/debugging-stories**):** This is a collection of links to stories and blog posts people made of debugging bugs of all kinds — weird, complicated, funny, disastrous, you name it, it's probably there. And if not, please submit a PR to them so I can read it next time I visit.

USEFUL STUFF

This is quite possibly the most vague category I have, but it has quite a lot of, well, useful stuff in it. A lot of is actually specific to me — Linux drivers for my laptop's fingerprint reader, Linux drivers for my mouse, Linux drivers for my laptop's dual GPU setup… you know what, I think we should stop laughing at my operating system choice and start looking at the other links, a lot of which are fairly useful/interesting to most people.

- **MathSoc Work Term Report Template (**https://github.com/MathSoc/mathWorkReportTemplate**):** Turns out MathSoc had a LaTeX, fully-compatible-with-faculty-specs, work term report template all this time! Who knew? Especially useful if you haven't figured out a template yourself yet, or if you've been using Word/Google Docs and want to move to LaTeX.
- **Dual boot Windows and Ubuntu with disk encryption (**https://www.mikekasberg.com/blog/2020/04/08/dual-boot-ubuntu-and-windows-with-encryption.html**):** At my previous workplace, for insurance reasons I had to keep my work laptop's hard drive encrypted at all times. Also, my employer wanted me to have a dual-boot Windows/Linux environment, where I spent most of my time in Linux but spent some in Windows if I had to debug a Windows-specific issue. The trick was that both these OSs had to be encrypted, and neither Windows nor Ubuntu supports this. This blog post by Mike Kasberg came in clutch here, and I was able to encrypt my fresh Ubuntu installation and keep the factory Windows 10 install Bitlocker encrypted.
- **Intel Processor Label Request Form (**https://app.smartsheet.com/b/form?EQBCT=946510ad2aab4ad2a4f9fd73f6fc1c26**):** I know the URL is a bit iffy, and to be honest I don't know where I got the link. But I know this form works; I filled it and uploaded a picture of my old ThinkPad T60's BIOS screen showing the processor name and serial number; and sure enough, I received an Intel® Core™ 2 Duo sticker in the mail a few days later, completely free of charge. Which I promptly stuck on my new Core™ i7 laptop for funsies. I also found out that there's a limit to how old a CPU sticker you can get; apparently they don't stock Pentium™ 4 Mobile stickers anymore.
- **DeTeXify (**https://detexify.kirelabs.org/classify.html**):** Ever been in the middle of writing a paper in LaTeX and suddenly forget the command for a particular symbol? Ever wondered if the infinity symbol \inf, \infin, \infty, or \infinity (or WLOG any other symbol)? This website would be a godsend; you draw a symbol and it tells you what the command is. It's that simple.
- **Pivotal Web Services:** Presumably I bookmarked their homepage because I liked some web services they were providing. Sadly, they seem to have been acquired by VMWare. Oh well.

TO READ

I have very high hopes of myself. You see, as a little child in Drusselstein I used to read all the time. And that led me to believe in a delusion; that I would keep reading as I grew older. Having still not grown out of it, I collect a list of academic papers and books that I will definitely probably maybe get to next week if I'm free (I won't be).

- **Register Allocation For Programs in SSA-Form (**https://link.springer.com/chapter/10.1007%2F11688839_20**):** Back in CS 241E, we had to compile our programming language Lacs down to MIPS assembly code, which meant converting memory accesses into loads and stores. The strategy I used for doing this in my assignments probably wasn't optimal (a lot of registers went completely unused, and some registers were loaded/stored into the same location repeatedly), and we were told that an optimal strategy wasn't expected. That said, this paper was linked in the course notes as an example of a good strategy.

- **Paxos Made Simple (https://lamport.azurewebsites.net/pubs/paxos-simple.pdf):** Paxos is a very commonly used distributed consensus algorithm, and I worked a lot with distributed consensus algorithms in my first co-op. But my work involved Raft (https://raft.github.io/), an algorithm whose paper I have read, and whose creation was motivated *solely* by the fact that Paxos is so hard to understand. Leslie Lamport, the creator of Paxos, wrote this paper to explain Paxos simply, and perhaps to rebut the idea that it was so hard to understand.
- **It Can't Happen Here by Sinclair Lewis:** This is a short fiction book describing the rise of a nationalist dictator in the United States. I found it when I was looking for “more books like 1984”, and some people drew parallels between this and Donald Trump's presidency.
- **The Wisdom of James Mickens (https://mickens.seas.harvard.edu/wisdom-james-mickens):** James Mickens is a professor of computer science at Harvard, and he has some extremely funny and insightful essays on distributed systems, processor design, and more. I've read some of them and thoroughly enjoyed them; highly recommend over your next cup of coffee.

And I think that's all! There are a few more folders and a few more links in these ones, but these were the ones I found to be most relevant to at least some **mathNEWS** readers. So I hope they're useful to you! If so, feel free to hit the like button, subscribe to **mathNEWS**, and send comments to spam@tendstofortytwo.tk. I read every email, so if you have something to say then get typing!

tendstofortytwo

TIE GUARD PART 6: LOOSE ENDS

Pulling open the door to the BlackBox offices, Bhavya raced inside. Glancing from side to side, she saw a few other employees dressed similarly to the man who had attacked her earlier coming down the hallway. But it was a fairly narrow hallway.

That was good. She could take them one by one.

The first one went for a strike with the side of his palm; she caught the arm and kneed him in the groin, knocking him down. She sucker punched the second in the face before he got a chance to attack, stunning him, then kicked his chest to knock him back into the one behind him. By the time the third one had gotten up from behind the limp body of her coworker, Bhavya was ready to grab her and throw her back the way she had came.

As the last attacker crashed into the floor, Bhavya made her way into a room at the back, where a laptop sat, open and with an essay half-completed. Opening up Outlook, she quickly typed off a confession message, which she sent to CECA, to make it look like a whistle blower. She even found a couple more pieces of evidence to add to her case, including some memos from the manager of the laptop’s owner. Pressing the Send button, she turned to leave, only to see someone else standing in the doorway: a tall white man with greying hair and a black suit and tie.

“I’m not going to stop you,” the man said. “I don’t want to get my butt kicked in, since you dealt with all my guards. But you may have ruined my livelihood, and that of everyone else working here. I hope you think about that when you try to sleep tonight.”

Grinning, Bhavya walked past the man. “You tried to scam co-op students out of jobs and grades. Your workers will find another job. I think I’ll sleep just fine.”

❦

As soon as Bhavya arrived back at her apartment, she flopped back on her couch. Ally smiled, looking at her friend and roommate from across the room. “Tough day?”

“Yeah, but it feels like a weight’s been lifted.” Bhavya said, sighing.

“That’s good to hear. Got an assignment done?”

“Something like that,” she replied, letting herself enjoy the comfort of the couch cushions. Getting a notification on her phone, she pulled up her email to see a message from CECA acknowledging BlackBox’s fraudulent postings and committing to redoubling their efforts in catching similar postings in the future. She smiled. All in a day’s work.

Predap

PMC PROBLEM SOLUTION

Hello friends,

The time has come to say goodbye. The term is coming to an end, the **mathNEWS** creativity has evaporated, and I am well on my way to moving to Waterloo to start my second year afresh. It has been an absolute pleasure. I do not know where the winds will take me, or whether destiny will allow me to return to the wonderful power of PMC executivehood. But know this — I will treasure the moments we have shared this term in my heart, always.

Congratulations to Sri and Nick for submitting beautiful solutions to the last problem of a beautiful term. Such a beautiful term, in fact, that I am forever promising to never commit to any extracurriculars over co-op because I'll have "*soo* much free time" ever again. To savor this unique experience. What a sacrifice indeed.

Sri's solution, a classic and one of the prettiest math ideas I've ever encountered: "Let's assume that the statement is true. We know that $\sqrt{2}$ is irrational. Consider $a = b = \sqrt{2}$. Let $c = a^b$. From our assumption, c is irrational. Again from our assumption we get: c^a is irrational. However $((\sqrt{2})^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^2 = 2$, ie. $c^a = 2$ which is rational. CONTRADICTION!"

Nick's solution, an absolutely **hot** piece of math:

Lemma: There exist real numbers which are not the root of any polynomial with integer coefficients.

Proof: Let $\mathcal{A}$ be the set of real numbers which are the root of a polynomial with integer coefficients. First, note that $\mathbb{Z}$ is countable, so $\mathbb{Z}^n$ is countable for any n. We may uniquely identify a polynomial by its coefficients, so the map from n -tuples of integers to degree n polynomials is surjective. Therefore, there is a surjection $\bigcup_{n\geq 1} \mathbb{Z}^n \to \mathbb{Z}[x]$, and since the countable union of countable sets is countable, $\bigcup_{n\geq 0} \mathbb{Z}^n$ is countable; therefore, $\mathbb{Z}[x]$ is countable. Next, consider the map $\varphi : \mathbb{Z}[x] \to \bigcup_{n\geq 1} \mathcal{A}^n$ defined by $0 \mapsto 0$ and $0 \neq p(x) \mapsto (a_1, \ldots, a_n)$ where $a_1, \ldots, a_n$ are the real roots of $p(x)$. Each element of $\mathcal{A}$ appears in at least one tuple. Since there are countably many tuples (by the countability of $\mathbb{Z}[x]$) each with finitely many elements, there must be a total of at most countably many elements in the tuples together, and therefore at most countably many elements in $\mathcal{A}$. However, $\mathbb{R}$ is uncountable, so there must exist elements in $\mathbb{R} \setminus \mathcal{A}$, as desired. Note that all such numbers are irrational, since $\frac{p}{q}$ is the root of $qx - p$.

Theorem: There exist irrational numbers a and b such that a^b is rational.

Proof: By the lemma, transcendental numbers exist and are irrational, so choose an arbitrary one a. By basic calculus, the function a^x is continuous on all of $\mathbb{R}$. So it satisfies the intermediate value property on the interval $[0, 1]$. Pick a rational number $\frac{p_0}{q_0} \in (\min(1, a), \max(1, a))$, where $p_0, q_0 \in Z$. Since $f(x) = a^x$ satisfies the intermediate value property, there exists a $b \in [0, 1]$ such that $a^b = \frac{p_0}{q_0}$. Note that $b \neq 0$ since $\frac{p_0}{q_0} \neq 1$. If b were rational, say $b = \frac{p_1}{q_1}$ with $p_1, q_1 \in \mathbb{Z}$, $p_1, q_1 \neq 0$, then $a^{p_1} = \left(\frac{p_0}{q_0}\right)^{q_1}$, so $q_0^{q_1} a^{p_1} - p_0^{q_1} = 0$. This would imply that a is a root of the polynomial $p(x) = q_0^{q_1} x^{p_1} - p_0^{q_1}$, which contradicts that $a \notin \mathcal{A}$. Therefore, b is irrational, as desired. Note that this argument can be applied to any rational number in the image of a, which gives a much stronger result; if a is transcendental and $a^b \in \mathbb{Q}$, then b is irrational or zero. Since there is a rational number other than 1 in $\min(a^x, a^{x+\varepsilon}), \max(a^x, a^{x+\varepsilon})$ for any $\varepsilon > 0$, this implies the set of irrational b such that a^b is rational is dense in $\mathbb{R}$. Neat!

That's all for the Spring 2021 term folks — wait shit no actually that's not it. A couple announcements:

The **Over 9000 contest** is happening August 6th! Fill in this form if you're interested: https://forms.gle/YdVckdJzrtaqdzDF6

The last **Prof Talk** is happening August 17th! Professor Pechenik will be giving a C&O talk and completing our triad of prof talks for the first time in 293804 years. Watch out for a club email.[1]

That's all folks (for real)!

VP Propaganda, PMC

[1] To join the mailing list, email us (me) at pmclub@gmail.com.

N THINGS THAT ARE OVER 9000

- The number 9001
- The population of Mississippi Mills, ON
- The population of the U.S. state of Mississippi
- The number of words in this issue of **mathNEWS**
- The number of stars in the universe
- The depth of the Marianas Trench in metres
- The highest possible score in Pac-Man
- The number of years since the Last Glacial Period
- The number 9002
- The estimated number of lions alive today
- The mass of a cruise ship anchor in kilograms
- The number of monthly listeners Cardi B has on Spotify
- The retail price of a Rolex Datejust
- The number 9003
- Son Goku's power level in the English dub of the episode "The Return of Goku" from the anime *Dragon Ball Z*

Vegeta

AN INTERVIEW WITH CY AND GIRAFARIG, GAME DEVELOPERS AND FOUNDERS OF CHILI TRUMPET

CC: TODAY WE HAVE CY AND GIRAFARIG FROM CHILI TRUMPET JOINING US FOR AN INTERVIEW! CHILI TRUMPET IS A GAME DEVELOPMENT STUDIO THAT HAS BEEN CREATING ASSORTED WEB-BASED GAMES AND PUBLISHING THEM IN mathNEWS.

CC: WHAT IS THE ORIGIN OF THE NAME CHILI TRUMPET?

girafarig: This was a response given to us from the **mathNEWS** idea bot Sungmin Chee.

CC: WOW, THAT WAS NOT WHAT I WAS EXPECTING — BUT FOR SOME REASON I'M NOT SURPRISED. WHAT INSPIRED YOU TO MAKE VIDEO GAMES?

cy: We've always wanted to make games because we enjoy playing games but we never really had the time to get around to doing it, and we both felt like our skills weren't developing as game designers. The only way to get good at something is to practice it, so we thought "we need to force ourselves to make games," and one of the best ways of doing that is to give yourself a deadline. With **mathNEWS**, there's a real deadline.

CC: CAN YOU GAVE A BRIEF OVERVIEW YOUR GAME DEVELOPMENT PROCESS? WHAT ARE THE LARGE STEPS THAT YOU TAKE WHEN PUTTING TOGETHER A GAME?

girafarig: We start working on the next game right after release of our current game. So far, all of the ideas for the games we've had are pretty much on the spot. What we do is we take our boilerplate we made for our first game—all vanilla JavaScript, simple game loops, importing images and stuff. Then we paste that into a new folder and then we start bashing out whatever we feel like.

Hopefully by a week in, we have some idea of where the game is actually going, and the second week is doing the art stuff and polishing. Neither of us are very good at art, so the art's kind of the secondary part of it, as well as the visual design, and that takes up the bulk of the time. I would say we work a lot more in the second week than the first because of that.

CC: WHAT'S YOUR RATIONALE BEHIND YOUR DECISION TO WRITE ALL OF YOUR GAMES IN JAVASCRIPT IN THE WEB BROWSER INSTEAD OF A LARGER ENGINE LIKE UNITY OR UNREAL?

cy: The decision was primarily because we want the games to be easy to play—not that the games are easy, but that you can run them on your computer easily. A lot of the time you see on the Internet, "I made a game!", and then you try to run it and it takes an hour and you still can't get it to run on your computer because actually releasing an executable is hard, especially for amateur programmers, so going with a browser makes it possible for just anyone to just click on the link and play it.

girafarig: Neither of us have any experience with the bigger things like Unity or Godot or what have you so we just decided to go with this—this is just vanilla JavaScript, so not even a JavaScript game library or anything.

CC: OF ALL THE GAMES THAT CHILI TRUMPET HAS MADE SO FAR, WHICH ONE ARE YOU PROUDEST OF AND WHY?

cy: I would say I'm most proud of the latest game that we made, which is *Goose of All Time*, because I think it is the most interesting game that we've made and the one that is most conducive to being an actual game that you can buy on Steam—not that we're gonna do that, but I think the idea is there for an engaging game that can have lots of content versus our other games (which I think are good too, but are kind of constrained by what they are—they can't really get any bigger).

In terms of like playability or fun-ness, I like *WRITE A GREAT* ***mathNEWS*** *ARTICLE IN THREE EASY MINUTES* just because I'm very good at typing, so I like playing that game the most.

girafarig: The problem I have with answering this sort of question is that being satisfied or "most proud of the game" has a lot of different axes. I would also say that I like *Goose of All Time* the most, but I'm also the most disappointed in what we did with *Goose of All Time* because some other games are "complete". *WRITE A GREAT* ***mathNEWS*** *ARTICLE IN THREE EASY MINUTES*, or *leaving home* or even *Virtual Goose 2021*—despite being kind of a mess gameplay wise—all feel like rounded-out experiences, while in *Goose of All Time*, some of the decisions I made felt more impromptu and less like I had time to flesh things out. But, the end result is very good and I'm very happy with it.

CC: WHAT IS YOUR FAVOURITE GAME?

cy: That's a hard question to answer, and I don't know if I have a favourite of all time. Recently, the games that I enjoyed were *Baba Is You, Dicey Dungeons,* and *Celeste.*

girafarig: This is not the deepest cut ever or most impressive underground hit, but I like *FTL*—I think that is my favourite game, both as a player and as a designer.

CC: SO, NOW THAT CHILI TRUMPET HAS MADE ITS FIRST MOVES IN THE VIDEO GAME DEVELOPMENT SCENE, WHAT ARE YOUR NEXT PLANS?

girafarig: Chili Trumpet as an entity? I'm not attached to the name; it's just something handy I picked for the Gitlab website. Going back to school, I probably won't be working on many other games for a while. I have some bigger projects just in my head, for the most part.

cy: In terms of games, I'm not sure I have anything I'd immediately want to do. Very generally speaking, I would like to make games with a narrative. The idea of making a game with a really good narrative has always interested me because I like to write. Most games with a very heavy narrative element I haven't really liked, and I feel like there is so much more potential for blending a narrative with a game that hasn't been explored yet. Individually, you're going to keep seeing me write in **mathNEWS** and stuff like that.

girafarig: Here's my project, then: this is completely opposite of a narrative game, but I've always wanted to make a traditional 2D fighting game, so if any of you are artists or animators, please contact me.

CC: DO YOU HAVE ANY ADVICE FOR PEOPLE WHO WANT TO START MAKING VIDEO GAMES?

cy: On the programming side of things: if you are attending UW, as I'm assuming you are, then learning programming and taking CS courses is a good place to start. Taking a course like CS246 or CS246E really helped me get better and faster at writing programs. In fact, the final project of CS246 has you make a game and CS246E has you make Vim, which is basically a game in the way it's structured. Then read up on the game loop and other game programming patterns.

girafarig: Design is kind of a murky thing to talk about or to even give advice for 'cause I don't think there's any universal way to make a good game design-wise, because if there was I would be very happy and also not a game designer.

Just make games. That's such a cliché piece of advice. Everyone says that, you know, "just do it." Just make the games. What else is there to do besides make the games? Make the games and then whether you *don't* like it or not, you have to finish it. You have to finish the game. There's nothing to do but finish the game. There's no bullshit where you make a game halfway, and say "this game's obviously not going to be fun." No, you gotta finish the game and say "oh it's fun because of this, this and this, and it's not fun because of this, this and that," and then you write that down and then you make another game. That's how you do it.

CC: AS WE WRAP UP, DO YOU HAVE ANYTHING YOU'D LIKE TO SAY TO THE READERS OF mathNEWS?

cy: I'd like to say thanks for reading our articles and playing our games!

girafarig: And if this inspires you, I would be very happy if you put your own games in **mathNEWS**! Someone did that for their Game Jam, in issue 146.5!

CC: mathNEWS DOES ACCEPT A VERY WIDE VARIETY OF CONTENT! CY, GIRAFARIG AT CHILI TRUMPET, THANK YOU FOR YOUR TIME, THANK YOU FOR YOUR INSIGHT, AND THANK YOU FOR THE GAMES!

The above is an edited abridged transcription of an interview between **mathNEWS** *writer CC and developers cy and girafarig from Chili Trumpet, held on July 23, 2021.*

Reach out to Chili Trumpet at `mathnews@gmail.com` *(emails will be forwarded to cy and girafarig).*

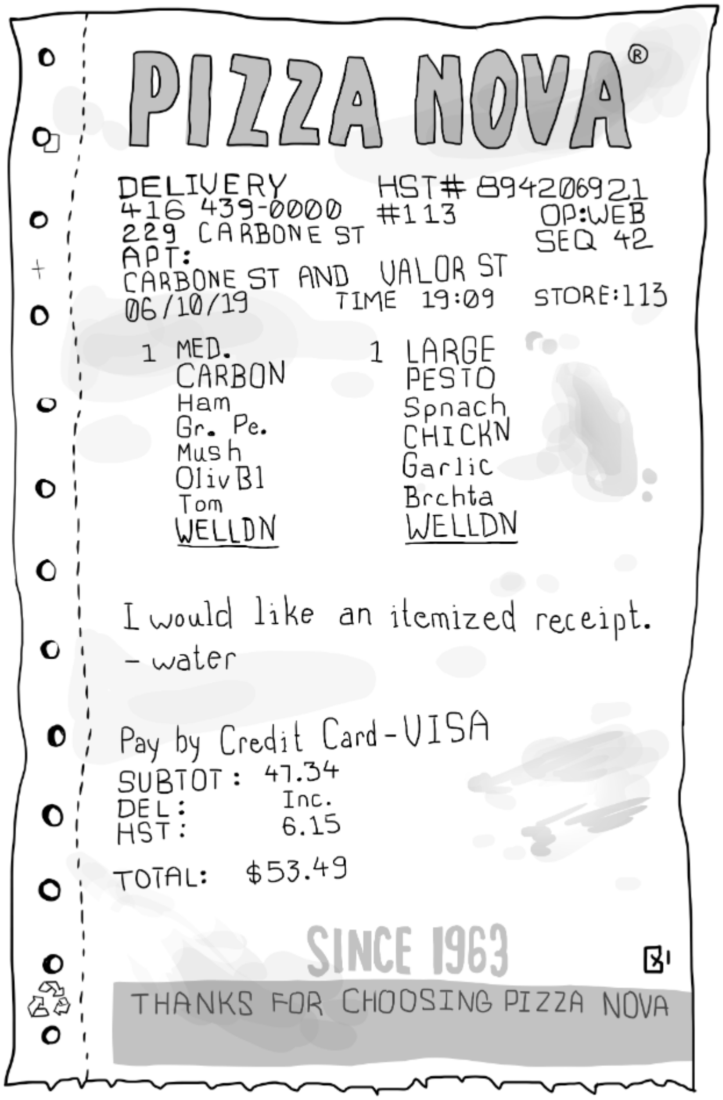
Play Chili Trumpet's games at `https://chilitrumpet.gitlab.io/`.

CC

CHEESE

> **slams a block of red Leicester cheese next to a block of cheddar cheese on to a wooden cutting board on a wooden table**
> **CHEESE.**
>
> James May, 2019

boldblazer



PIZZA NOVA®

DELIVERY HST# 894206921
416 439-0000 #113 OP:WEB
229 CARBONE ST SEQ 42
APT:
CARBONE ST AND VALOR ST
06/10/19 TIME 19:09 STORE:113

1 MED.
CARBON
Ham
Gr. Pe.
Mush
OlivBl
Tom
WELLDN

1 LARGE
PESTO
Spnach
CHICKN
Garlic
Brchta
WELLDN

I would like an itemized receipt.
- water

Pay by Credit Card - VISA
SUBTOT: 47.34
DEL: Inc.
HST: 6.15
TOTAL: $53.49

SINCE 1963

THANKS FOR CHOOSING PIZZA NOVA

FINAL RETROSPECTIVE

>>> Our last game, Oh no we lost all the articles, will be played at the **mathNEWS** End Of Term event, for all who have contributed to **mathNEWS**. You can play Goose Of All Time v2.0 here: `https://chilitrumpet.gitlab.io/goatv2`

girafarig:

Sorry, **mathNEWS** readers, the main game this time is only playable at the End Of Term event. This is due to both implementation factors and because this game works best in a close and controlled (by me personally) environment. Hey, one more reason to write for **mathNEWS**!

It's a pretty simple multiplayer 'board game'. I blatantly stole the idea from an already existing game, and modified it with some **mathNEWS** trimmings. This was the first time we used any sort of framework or library for JavaScript, and boy it was not a fun experience. The final product is essentially a socket.io tutorial thing with a bunch of stuff tacked on, and I intend to turn my server on to run a game off it and turn it off after. It's a pile of hacks. But it works, that's all that matters anyway.

On another topic: If you liked G.O.A.T. last issue, I hope you'll enjoy v2, which is a small expansion of the original. There are a few more enemies, a bunch more moves, and some other interesting features. I really got into a groove after finishing the original, and I like all of these moves a lot (well, I like 'Brain Freeze' a little less). It feels incredible that at the end of making G.O.A.T. I felt that I had totally run out of move ideas, but after some time a bunch more popped up in my head.

Due to some of the features being a bit hard to test, I haven't comprehensively went through this game yet. Maybe expect bugs a little. But cy did fix that awful 1-frame rendering bug that plagued our lives for several weeks, which is exciting.

There's not much to say otherwise. I do feel that expansions of games always veer into self-indulgent territory, where the core ideas have already been fleshed out and all that's left is adding more gimmicks. I said in that interview earlier that my favourite game is FTL, and I'll expand on that by specifying *vanilla* FTL, without Advanced edition. I don't think that expansion added very much. v2 of this game does add some replayability and incentive to beat the game multiple times with different restrictions, which is probably the biggest thing in making this game more 'complete'? Once you're done with a game, it always feels weird to add more to it, you can never tell what's appropriate and that's filler.

And finally, a reflection on this term: we did it! We made six games, one every two weeks. It wasn't as neatly cut as that, with some being especially small and some overrunning their 'allotted' times, but they all came out on time. I'm happy with all of them, and they've all satisfied my expectations in different ways.

Virtual Goose 2021 has problems with its difficulty and its uniformity — I never managed to fully capture the spark I was after. But it was our first game, it has a coherent idea and a decent execution, and the polish was good. It's still the game I relate with this project the most.

Great **mathNEWS** Article was a bit simplistic and we made it too hard again, but it's a game with a lot of visceral fun. In the course of making the game, I got so much better at playing it that it felt somewhat unreal. I think it's the game that most fully realizes that we started out to do when we made it.

leaving home was the smallest game, and one where I never fully felt immersed into creating; it was just too short. But it does well in what it's focused on, and I think it's the most popular game with the writers. I think it's the most narratively-strong game in this list, as well.

Super Little Bug was… disappointing. I wanted others to like this game as much as I did. This game was the one that took us the most work to push all its disparate parts together, and I'm proud of the levels and the art and the fine-tuning by both of us it took to get it feeling right. I suppose it's the opposite of leaving home in this way.

G.O.A.T. is really the climax here, being the biggest and most replayable and most flashy game of them all. But in this way it's also the most incomplete; I think I talk about this in CC's interview as well. There's so many directions I want it to go in, the effects could do with a lot of work, there are so many little things I should be adding to make it better, I keep thinking. I don't know where to stop, and that's why I made v2 in the first place. I see a glimpse of 'real big game' game development through this one.

Finally, Oh no we lost the articles is the epilogue. It's a straightforward game that's hard to not be fun, mostly because it's multiplayer, whose biggest challenge in creation are simple technical issues. Since I haven't really played this game yet, I guess I don't truly know how I feel about it. Maybe it won't be popular, or maybe it'll break ten minutes before we start playing, who knows?

Thanks for coming along with us. Thanks to all the **mathNEWS** writers who played our games on production nights. And if you want to make a game, you should do it.

cy:

I didn't have as much of a hand in this final game, since it was fairly simple to program and I happened to have a bunch of stuff on my plate these two weeks. So I think I'll reflect on the term…

It's been a lot of work, some fortnights more than others. I missed a lot of time I could have been spending with family, or outside biking, or playing video games. There were many times I regretted this. But, I always scheduled for "fun". As I've said, the programming really isn't that hard.

Near the start of a game, I'd only work on the game for about two hours a day. Then the time spent on the game would gradually increase, until the weekend right before prod night. We'd probably work on the game all Saturday, with final touches on Sunday.

But it was never too hard. I never felt like I'd "bitten off more than I could chew". That's partly because we had some experience working on projects together before. And partly because I genuinely love, and crave, to do things like this. It's almost carnal.

I'm currently taking a creative writing course right now (ENGL335—I highly recommend!) and one of our craft readings pondered why writers choose to write. To paraphrase, the text proclaimed that we don't write for fun or pleasure; engaging in the act of writing isn't as enjoyable as other pastimes that are designed to be enjoyable. Instead, writers write because they have some need to.

I agree wholeheartedly, but it misses something: that getting something done and accomplishing is euphoric. It's a high. Yep, you heard it here first; receiving an Article of the Issue is better than sex.

In that way, working on programming projects is absolutely addicting compared to writing, because (if you're doing it right), it's broken up into small, logical chunks. You just need to write *one* function and there it is, that high. You could stop there, or just write one more…

So I would attribute my success over the course of this project to both of these: prior experience and my own disposition that lends itself well to these sorts of things. The final factor was the unique situation that girafarig and I were in: we were living at home due to COVID, and working at fairly non-stressful co-ops, giving us plenty of spare time. I'll be really lucky if I can get to do something similar again during my undergrad.

And yes, overall, I would say this was a massive success. I am proud, elated, etc., of what we've accomplished. If you're interested in doing something similar, please, go for it! Don't worry (as I did) about letting other people down, or about stopping in the middle of the term. That's all okay. I'll be here cheering you on.

A big thanks to everyone who read our articles and played our games and gave feedback and told us how much they liked our games and bugtested and waited patiently while we were bugfixing and hyped up our games and referenced our games in conversation and made fanart of our games and supported us. You have over-inflated my ego and made me feel, just a little, that maybe possibly, we could, sort of, *make it?*

cy and girafarig

A SHITTY MOVIE IDEA PART 2: ANYTHING IS BETTER ON A PLANE

Welcome back to this new edition of the segment I shamelessly stole from god ϟpeED. I'm **methNEWS**, and I'm here to ~~ruin~~ ~~defile~~ improve the cinematic classic known as *Romeo and Juliet: Sealed with a Kiss*. That's right, this $400,000 budget animated movie about 2 star-crossed seal lovers was so impactful it sent shockwaves through the space-time continuum that inspired Shakespeare to write his tragic play based on it. Anyways, my idea is to make a new version of the classic tale of Romeo and Juliet, but on a plane. In this story, the Capulets and Montagues are two families of passengers who have been bitter rivals ever since the plane took off. The members of the families have been blinded by so much hatred and gross airplane food that most of them forgot how this feud even started. However, the elders of the Premium Economy seats speak of an ancient prophecy regarding the Capulets that is responsible for the conflict:

The Capulets' riches will last them forever
Their family tree will survive thick and thin
But the man with the blade born not of checked-in baggage
Will cut down their lineage and end their whole kin

Rumors say, this terrible prophecy has caused the Capulets to distrust all other families on the plane. In fact, the members of the Capulets are forbidden from even talking to the Montagues—let alone court them. However, there are two lovers that are trying to overcome this hatred and animosity against all odds: Romeo and Juliet. They must keep their love in secret or face consequences worse than death. This is especially difficult since this particular plane has like 60 seats at most, and basically no private enclosed space.

The movie opens on Romeo, wooing Juliet with a love sonnet:

Oh my sweet Juliet
My love can't be quenched
Just like the Sahara
Can't be helped with a drink
Why must our forefathers
In their constant fighting
Bring our truest passion
To destruction's brink
No matter though, my love
We won't let them stop us
We won't let our feelings
Grow colder or shrink
To make me as pure
As you and your beauty
I have washed my gonads
In the airplane sink

Juliet answers with an entrancing poem of her own:

Oh Romeo my partner
My forbidden love
When I'm with you I feel high
As the clouds we're above
And my truest feelings
Words cannot explain
To be with you forever
I'd crash this whole plane
I'll bring our two families
Right into the ground
Leave no child or witness
So we'll never be found
It doesn't matter who we are
What matters is our plan
Any day could be our last
We must enjoy it while we can

After 30 more minutes of pointless romantic wailing I genuinely couldn't be bothered to write about, Romeo and Juliet meet on the plane's balcony to recreate the iconic balcony scene. Now, this would be a bit hard to write since stepping outside while flying at top speed would mean both of our protagonists would tragically and immediately die from about 7 different reasons all simultaneously. However, we fix this physics-related issue by simply explaining that Romeo was secretly wielding the infinity gauntlet in a post-credit scene. Nonetheless, the wind going at 700 km/h through the balcony still muffles the sound of the two lovers' discussion. Here is the transcript of their obstructed conversation:

Romeo:

Oh Juliet, I want to … …
… hard and until ….
… matter who's watch….
…. … … … orange juice

Juliet:

Romeo, we must … …. …
…. … and you can't …
Until I … … … scream
… … … and the lobster too

Romeo and Juliet share a kiss, and enter back into the plane to return to their families. Meanwhile, Scumbagio—a shady man with questionable motives, enters the scene and starts singing a subtle song foreshadowing his true intentions.

I am the movie's villain
This is the role I'm filling
I want to kill protagonists
That makes me the antagonist
But you won't know this, I insist
Until the movie's major twist
My evil is hidden so well
The audience will never tell

It is revealed that Scumbagio is secretly in love with Juliet as well, but is also forbidden from seeing her due to limited aisle space on the plane. Therefore, Scumbagio plans to poison Romeo with expired pretzels in order to become the only man in Juliet's life. However, right before Romeo goes to eat the poisoned food, he hears that the in-flight entertainment system started playing a 45 minute cut of Avatar in glorious 4-inch display iMax. While Romeo is away to watch the movie, Juliet eats his pretzels and falls down immediately.

Romeo comes back to his seat and is shocked to find Juliet, lying on the floor. In his state of pure horror, he starts singing:

Juliet! how could you eat my pretzels you fat bit–
What is this? Why does it seem your soul has just been
snitched?
Why do you lie so still on the floor of this cursed plane?
And why I didn't check your pulse will never be explained
Oh well then, I will assume you died here right away
And on this flight without you I can't bear to stay
I always knew I would die by the blade of love's dismay
So I will end my life with the knife I found on this tray

Romeo picks up a plastic knife he found on a lunch tray and commits suicide in a scene so graphic, so violent, and so gruesome, the **mathNEWS** editors sent me a Cease and Desist order before I could even start writing it. *[Editor's Note: Apparently the gag order is still in the mail.]* But right after Romeo takes his final breath, Juliet wakes up.

This nap I just woke up from made my head feel like it's hit
And now it's time to go and take the world's worst smelling shi-
Oh Romeo!! Why are you covered in a pool of blood?
The tears this sight has caused are enough to start a flood
Without you in my life I don't think I will ever manage
|And I don't want to pay the fine for your blood's carpet damage
Who knew a bag of pretzels would cause this pain to unfold
And thus I add another kill to plane food's deadly toll

Juliet takes the plastic knife from Romeo's body and ends her life in a scene equally as horrid, which will have to remain censored until obscenity laws catch up to my incredible artistic talent. The shot remains centered on the two lovers' bodies for about 10 minutes, which is just enough time for the audience to get that the prophecy has come true and then set a reminder to nominate the movie for an Oscar. The tragic death of Romeo and Juliet unites the two families in their grievance and causes Scumbagio to jump out of the plane, finally bringing a good ending to the movie. With the families in peace, the flight finally lands back on the ground after an uncountable number of years in the air. However, it is so far gone into the future that apes have taken over the world! The final shot of the movie is of apes entering the plane, armed with guns and riding on horses. Then, the screen fades to black and a movie title appears:

PLANET OF THE APES: ON A PLANE
Coming soon.

To be continued????

methNEWS

sexNEWS: OH WOW MANY QUESTIONS

Welcome back to **sexNEWS**, a biweekly column in which I answer relationship advice questions submitted by you, the readers.

As always, feel free to send your questions to mathnews@gmail.com to be potentially answered in this column. Anonymity is guaranteed[1]. You're also welcome to include additional information to give context that you don't want included in the article if you're worried that your situation is specific enough that fully explaining it would expose you. This column is not restricted to just romantic relationships, we discuss personal relationships as well.

> *How tall are you?*
>
> Short person

What is this, the **mastHEAD**?

> *I plan to marry my boyfriend and become a housewife / programmer for our two-person indie game studio, while he supports us financially. Naturally, we'll split any of the game studio's income, but should he give me half of his income from his day job? (Of course I would then pay for half our expenses.)*
>
> Not gonna make it

Disclaimer: I am not a lawyer, this is not legal advice. Also this is how it would work in Ontario, I dunno about other jurisdictions.

Legally, unless you get divorced and can't agree on the terms of your divorce, you can basically do whatever you want here. If you get divorced and can't agree on the terms of the divorce, then both your and your husband's savings are considered joint for the entire household and get equalized. There may be an entitlement to spousal support, but this would be based on a number of factors. Since you would be both marriage and business partners, it would be a good idea to have a marriage contract and a shareholder's agreement prior to getting married and starting the company respectively.

Practically, in terms of budgeting what you suggested would probably work. Some other possibilities:

- Just have all of your husband's and the game studio's income go into a shared bank account and don't worry about keeping track of who pays for what.
- You husband pays for everything.
- You split your expenses proportionally to your income (only works if your income is greater than zero).

You can do whatever you and your husband think is fair.

> *Do you have any sex tips for someone who enjoys, but cannot orgasm from, penetration beyond the obvious "use your hands" kind of advice?*
>
> 30%

Well if you're enjoying it, what's the problem? Do you feel societal pressure that you need to orgasm? Is your partner upset that you aren't orgasming? If the answer to either of those is yes, then that's something you should think about.

Otherwise, try some sex toys. There are lots of sex toys intended for couples to use, some of those might help you.

And as always, use lots of lube.

> *I have great, satisfying sex all the time. What should I do?*
>
> Has lots and lots of really amazing sex

I'm going to assume that you're having sex so often that it's not leaving you with enough time to do other things (otherwise, I don't see the problem).

Try to do things that leave you feeling less horny. Here are some suggestions:

- Clean the bathroom. Who can be horny after washing a toilet?
- Make dumplings alone. It's fucking exhausting, and you'll be too tired for sex after. Do **not** make the dumplings together, because then it's less work and you might not get sufficiently tired. You can eat the dumplings together though. This suggestion has a huge benefit: you get to eat dumplings.
- Go for a long walk when it's really hot out, until you're both sweaty and tired. Then you won't be horny.
- Do some research on climate change and get sad about how we're screwed.

> *Someone I am interested in refuses to sleep with me unless I go vegan? Is it worth giving up meat to get meat?*
>
> Veghan Varkle

Yes. If they are vegan their cum will taste amazing. Definitely worth it.

*Senior **mathNEWS** Relationship Correspondent*

[1] Unless there is a court order or something, but if I foresee that being an issue I probably won't answer the question in the first place. Canada has unfortunately weak laws protecting journalists, and I don't know if this column even counts as journalism.

warSOC X — THE STANDING COALITION FOR THE COUNTERPROLIFERATION OF ANTI-MATHEMATICS

On a desk in the deepest part of a very drab, very secure facility, there is a pink landline phone. This phone belongs to The Director, and hasn't rung in twenty years. Most directors of such clandestine facilities are very busy, but The Director has had no notable incidents in her long tenure. Each day flows with easy tedium and routine. Today, that will change—the harbinger of such being the deposition of one faux-vellum envelope onto that same desk. It's crumpled and tattered, and several muddy splotches as well as *actual feathers* dot the envelope. It looks like someone's gripped it tightly with claws. Very strange, but the facility's security will have screened it for immediate danger.

The Director, upon returning from one of many coffee breaks in her so-far uneventful day, spots it with eagerness. There's a separate letter, similarly tattered but opened, next to it. She reads this one first. It's addressed to the Prime Minister of Canada.

> *Dear Prime Minister,*
>
> *Theorem-Space, Anti-Mathematics, Mathematics-Manipulation, Rex Sibyllan, Geese.*
>
> *Now that these keywords have caught your attention, please forward the attached letter to your relevant department.*
>
> *Salutations,*
>
> *Lucy*

The Director almost jumps in excitement—at long last, some action! She tears open the second letter in glee.

> *To whom it may concern,*
>
> *I write to bring your attention to the Waterloo Anti-Mathematics Resistance Society, or colloquially, warSOC. Located in room 7111 of the Mathematics & Computer Building at the University of Waterloo, Ontario, this organization has been practising Anti-Mathematics, Theorem-Space access, and other prohibited acts.*
>
> *I humbly request you deal with these lawbreakers as soon as possible.*
>
> *Salutations,*
>
> *Lucy*

Dropping her envelope, The Director lifts the pink phone and rotary dials a three-digit number.

At 24 Sussex Drive, an identical pink phone starts to ring.

❦

Within two hours, strike teams Conjecture and Proposition are en route to Waterloo at two thousand kilometres an hour via a pair of VTOL jets. A dozen agents wait in the belly of each, armed to the teeth with Anti-Mathematics shielding, Theorem-Space equipment, and traditional weaponry. The Director is inside Conjecture's jet, enthusiastically orating a briefing.

“This is the first time in twenty years we have been called to action. Our predecessors built this organization to make sure the innocent citizens never have to witness horrors of Anti-Mathematics again, and we *will* protect the world from this **warSOC**. These criminals know exactly how dangerous the fire they're playing with is, and it's time to put them down. **warSOC** has been hidden from us for who-knows-how-long, and their cunning and subterfuge must be unparalleled.

But we have their location now. No warnings, no warrants—just break their door and take them down.”

❦

Soren, Vigil, and the Jack twins are eating dinner around a table in the **warSOC** clubhouse when a sudden crescendo to a scream of high-speed jet engines sends plates rattling.

“The hell's that?” Soren stands at once and walks toward the clubhouse entrance. Vigil starts running for his computer, and the Jack twins dash towards their ties.

A blast rips a hole in the clubhouse wall with a sharp crack and puff of smoke, and figure after figure in black armour swarm into the clubhouse. Their infrared overlays quickly identify the four **warSOC** members, and a series of soft *puts* accompany the darts that strike each of their bodies.

Vigil makes it to his computer—he only has time to type in a quick command, which activates a small script he wrote one evening many months ago on a whim, before he screams as the taser dart squarely in his back puts him under. One of the Jack twins suffers a similar fate, but the other dodges the dart, and manages a small Anti-Math attack with the tie he's just reached.

Anti-Math alarms go off in the black helmets of the intruders, and they dispatch the second twin with more darts.

Soren isn't hit. He raises his hands, and slowly turns to the light shining through the newly-made hole in the wall, eyes briefly landing on each of the unmoving **warSOC** members. An older woman, tall, slim, and dressed in a black suit and pencil skirt is stepping off a ramp through the hole.

She's wearing heels, which Soren finds absurd. Her voice is steely. “I am here on behalf of the Standing Coalition for the Counterproliferation of Anti-Mathematics. You and all your associates are under arrest for contravention of the Anti-Mathematics Disarmament Treaty."

"You're called SCAM?" Soren blurts incredulously.

"Down." The Director replies, and another taser does just that. She claps slowly, and the agents of strike team Conjecture start to take off their helmets, expressions of relief evident. They were expecting far, far worse. "Good work, everyone! Anyone hurt? First round tonight's on me."

❦

On the rooftop of the adjacent QNC building, Lucy watches the jet and members of strike team Conjecture blast their way into **warSOC**'s clubhouse with great relish. The setting sun frames the attack quite nicely. *Marvellous. Absolutely marvellous, isn't it, Newton?*

I'm just glad they're finally gone. No more killing.

Lucy decides to try a taste of English, borrowed from professor Salis. "Revenge is a dish best served cold."

❦

A high-priority email interrupts Name's work on her math assignment. She looks up from the paper she's writing on. Only one organization's emails have notifications enabled while she's studying.

"You came through, Vigil," Name murmurs as she reads the automated **warSOC** sensor data dump. The pink goose, the Magenta Menace, has made an appearance on the top of QNC. Name's pulled out her tie and is starting the Theorem-Space entry procedure when a second email from Vigil arrives.

The title reads "EXTREME DANGER **warSOC** COMPROMISED DO NOT RETURN TO CLUBHOUSE DO NOT REPLY". Name reads the contents in horror. A livestream of the **warSOC** clubhouse is attached, and she sees black-clothed figures, dust, debris, and her friends, unconscious, being pulled up a ramp through a hole broken in the wall. Vigil's email informs her that her **warSOC** records —along with all of theirs—have been deleted, and she is on her own now. At the bottom of the email, Vigil's attached a small joke: "This'll probably never be used, but I've watched too much *Mission Impossible* lately."

Name feels cold. She starts the Theorem-Space entry procedure once again, and with a flash and rush of air, her room is empty.

❦

Back on the roof of QNC, Newton and Lucy are sharing a moment of triumph. Two years of Anti-Mathematics, of companions lost and hard-fought battles with **warSOC**, all leading to this moment.

I didn't expect it to go so well, Newton comments.

Neither did I, Lucy replies. The rest of Newton's Arith Division is on its way, V-formation flapping through Real-Space to join the two geese on the QNC rooftop. Newton watches them approach with pride, but his pride turns to confusion as the second VTOL jet, property of strike team Proposition, drops into view behind the geese.

Within the jet, the pilots' scanners have identified Arith Division as evolved, Anti-Math-employing geese—all part of strike team Proposition's training: keeping an eye out for the waterfowl surrounding Anti-Math incidents as well as the humans. Turrets on the jet track, lock on, and fire nets, and Arith Division scatters. Geese are struck and entangled, and then gravity does the rest. Like Name, Newton and Lucy can also only watch in horror.

Newton… we may have just woken the lion.

To be continued…

CC

SURFING IS A TERRIBLE SPECTATOR SPORT

The heats are 30 minutes long, largely of dead time. The surfers appear to be waiting for a good wave. Sometimes they would start swimming, and then decide against it. They paddle back out and continue to wait. So then, eventually, the surfers finally actually surf on a wave. They go back and forth for a little bit, and then they mysteriously get some points. The British-sounding commentator talks about how they got good speed or good turns, but really they all did pretty much the same thing. No one does flips, no one does spins, they just turn left and right quickly on a wave. The wave dies down eventually, so the surfer sort of stops. Now we get to wait a minute to see the next guy go. He does the same thing. Were they not wearing different colours? I would not be able to tell they were different people. I imagine colourblind people watching this accidentally assume they have looped the same 20 second surfing clip over and over for 30 minutes. The camera stays far away from the surfers the entire time. No cool camera angles, no dynamic camera movements, no slo-mo replays to show the extreme technical skill required to pull off the world class athletics on display. It's shot like a game of golf. Points are mysteriously distributed now and again. There are judges for surfing, I guess. Points seem to go up quickly at first, but then they level off even faster. Someone wins the heat in the end. They look pretty pleased.

Who tricked me into thinking surfing would be fun to watch.

Aphf

OLYMPIC SPORTS TIER LIST

The 2020 Olympics are finally underway in 2021. For those that watched the opening ceremony, you may have noticed that at one point, the Olympic Motto was shown.

The Olympic Motto is famously “Citius, Altius, Fortius” which is Latin for “Faster, Higher, Stronger”, but the motto was updated in the days before the opening ceremony by adding the word “Communiter” which is Latin for “together” at the end. So now the full motto is “Citius, Altius, Fortius — Communiter” or “Faster, Higher, Stronger — Together”. But the news of this change got me thinking of one thing. How reflective of the three parts of the motto are the actual sports or disciplines in the Olympic programme? How many Olympic sports or disciplines are actually about being faster, higher, and stronger? I will now present my tier list of the sports ranked by how well they fit the Olympic Motto. The tiers used are S, A, B, C, D, and F.

❦

First, let's ask the question, “Is there an Olympic sport that incorporates all three parts of the motto?” I think there is a good candidate for the answer. Sport Climbing is a new sport introduced for the 2020 Olympics. There is only one discipline for these games which combines speed climbing, lead climbing, and bouldering. Speed climbing is literally just climbing a wall in the fastest time, lead climbing is climbing a wall while attached with a rope, and bouldering is free climbing but without being attached by anything. As you can see, being faster, higher, and stronger helps in becoming the winner in this sport. Literally, one of the three parts of Sport Climbing, speed climbing, is where athletes use their strength to climb up a wall the fastest. So, for that reason, I will place this in S tier.

Now to the opposite, and ask what would be the worst sport in terms of the motto. I will nominate Equestrian for that sport and put it into F tier. I think this one is basically the worst Olympic sports in terms of how well it fits in terms of the motto. All three disciplines are based on points, two score points from judges, and the other involves penalty points only. All that this basically determines is which athlete has the better control of their horse. For the most part, you are basically asking generally richer people who can afford horses and the like from generally richer nations to be the ones winning medals.

Next are the sports of mixtures, namely Triathlon, Modern Pentathlon, and Decathlon. They all are a combination of different disciplines but are competed differently. Triathlon involves swimming, cycling, and running all in one go with one final time determining the winner. Decathlon involves ten track and field disciplines[1] that altogether test all three parts of the motto. Modern Pentathlon is a bit troublesome as one of the five disciplines[2] is an Equestrian event that is points based. For that reason, I will place Modern Pentathlon into C tier, while Triathlon is placed in A tier, and Decathlon is placed into S tier.

Immediately, we can then begin placing a lot of sports into A tier just from the fact that purely time is measured to determine the winners. Whoever took the least amount of seconds, i.e. the fastest, simply win. The following sports are thus put into the A tier: Athletics events involving running, Canoe Sprint, all Cycling events except the BMX freestyle, Rowing, and Swimming.

Canoe Slalom also involves getting the fastest time, but there are penalties that add to your time. So the fastest do still win but some conditions apply. For that reason, I will place this into B tier.

Sailing similarly involves getting the fastest time, but there are multiple sailing races whose cumulative placements, not times, determine those that advance to the medal round. So, for each race, first place receives 1 point, second place receives 2 points, and so on, where the lowest cumulative score is considered better. Essentially, it is one step removed from just getting purely the fastest times. For that reason, I will also place this into B tier.

We can now put in a bunch of sports that are essentially about strength. I will put Weightlifting into the A tier since it is the only one that measures the best directly using kilograms. Athletics disciplines involving throwing an object are put into A tier as well as stronger athletes will become the winners. A lot of the martial arts and other contact related sports, I will put into B tier. In these sports, the ones that are stronger do become the winners, but there are still judges that affect how points are scored by athletes which can become more subjective. The following are put into B tier: Boxing, Judo, Karate, Taekwondo, and Wrestling.

The problematic one is “Altius” or “Higher” because not many sports do that. The ones that are purely about reaching the highest include some of the Athletics disciplines like pole vault and high jump which will obviously be put in A tier. This means Athletics except Decathlon is all in A tier.

Next are some sports that measure an aspect that the motto does not reflect: accuracy. These sports are Archery, Golf, and Shooting. Unfortunately, these three will thus be placed into C tier for the mere fact that the motto does not mention accuracy. At least better accuracy gets reflected into winning in each sport with no subjectivity getting in the way.

The team sports are next which are Baseball/Softball, Basketball, Field Hockey, Football (Soccer), Handball, Rugby Sevens, Volleyball (regular and beach), and Water Polo. An individual athlete matters slightly less as the better team as a whole determines the winner. Also, while strength does make a difference, a new element becomes involved in team sports. Good strategy alongside a good team determines winners, and since the motto does not refer to strategy at all, these team sports will be placed into C tier.

The following four sports feature both individual and team events. The sports are Badminton, Fencing, Table Tennis, and Tennis. These four are less about being purely faster, higher, or stronger as there is a lot of technique involved in how athletes contest each sport. Also, as with other team sports, strategy also gets involved like I mentioned above. Thus, these four are also placed into C tier.

The last set of sports are the remaining subjective sports where judges give scores to athletes. It basically means that if you have two equal athletes, one from the host nation, one not from the host nation, more often than not the host nation's athlete will be given a higher score than the other. It is what it is. However, even though the winners are determined from scores given by judges, an athlete being faster, higher, or stronger still get reflected in the scores, so for that reason, I will place these into D tier.

The final tier list is as follows:

S: Decathlon, Sport Climbing

A: Athletics except Decathlon, Canoe Sprint, Cycling events except the BMX freestyle, Rowing, Swimming, Triathlon, Weightlifting

B: Boxing, Canoe Slalom, Judo, Karate, Sailing, Taekwondo, Wrestling

C: Archery, Badminton, Baseball/Softball, Basketball, Fencing, Field Hockey, Football (Soccer), Golf, Handball, Modern Pentathlon, Rugby Sevens, Shooting, Table Tennis, Tennis, Volleyball (regular and beach), Water Polo

D: Artistic Swimming, Diving, Cycling's BMX freestyle, Gymnastics, Skateboarding, Surfing

F: Equestrian

Of course, this tier list in no way dictates if any sports are actually better than others. The only thing this tier list should affect in any way whatsoever is for when you have no idea which sport to watch and none of your favourite sports are currently live. Another option would be to just use a random number generator. It's up to you.

boldblazer

1. The ten parts of Decathlon are:
   100 metres
   Long jump
   Shot put
   High jump
   400 metres
   110 metres hurdles
   Discus throw
   Pole vault
   Javelin throw
   1500 metres
2. The five parts of Modern Pentathlon are:
   Épée fencing
   200 metres freestyle swimming
   Show jumping (Equestrian)
   3.2 kilometres cross-country run with…
   …Pistol shooting every kilometre

> You should be concerned if I were to stop biking to the university.
>
> PROF. BENOIT CHARBONNEAU

EPISODE 24: DYNAMIC MEMORY + CALL FOR APPLICATIONS

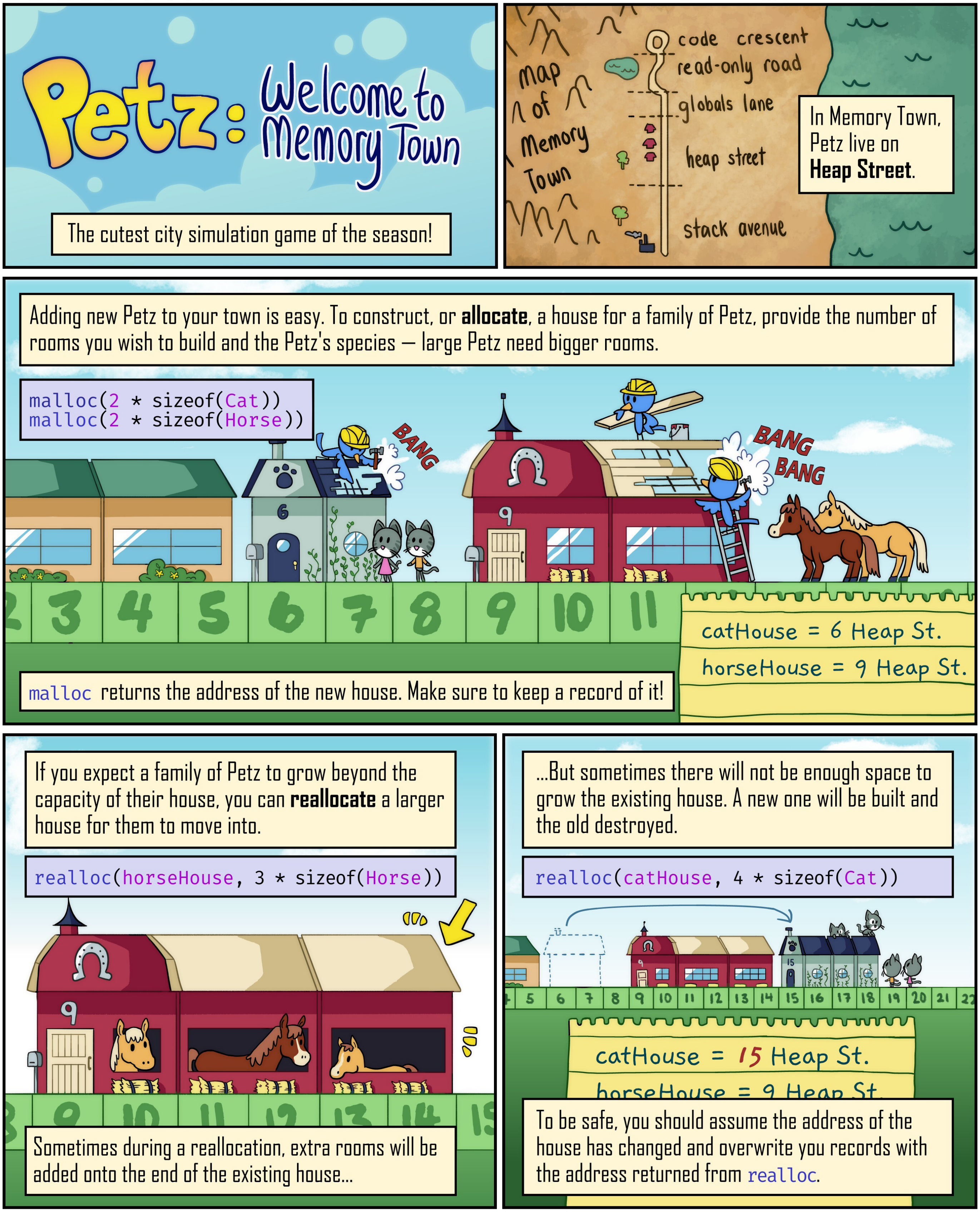
Enjoy Episode 24 of the MathSoc Cartoons series: Dynamic Memory!

Want YOUR name to be on the next MathSoc Cartoon? MathSoc Cartoons is hiring PAID comic writers and artists for Fall 2021! If you love explaining concepts, drawing comics, or telling BAD MATH PUNS, hurry and apply by August 27 at bit.ly/join_mathsoc_cartoons!

MathSoc Cartoons

CS 136: DYNAMIC MEMORY

STORY & ART BY CLARA XI



The cutest city simulation game of the season!

In Memory Town, Petz live on **Heap Street**.

Adding new Petz to your town is easy. To construct, or **allocate**, a house for a family of Petz, provide the number of rooms you wish to build and the Petz's species — large Petz need bigger rooms.

```
malloc(2 * sizeof(Cat))
malloc(2 * sizeof(Horse))
```

```
catHouse = 6 Heap St.
horseHouse = 9 Heap St.
```

`malloc` returns the address of the new house. Make sure to keep a record of it!

If you expect a family of Petz to grow beyond the capacity of their house, you can **reallocate** a larger house for them to move into.

```
realloc(horseHouse, 3 * sizeof(Horse))
```

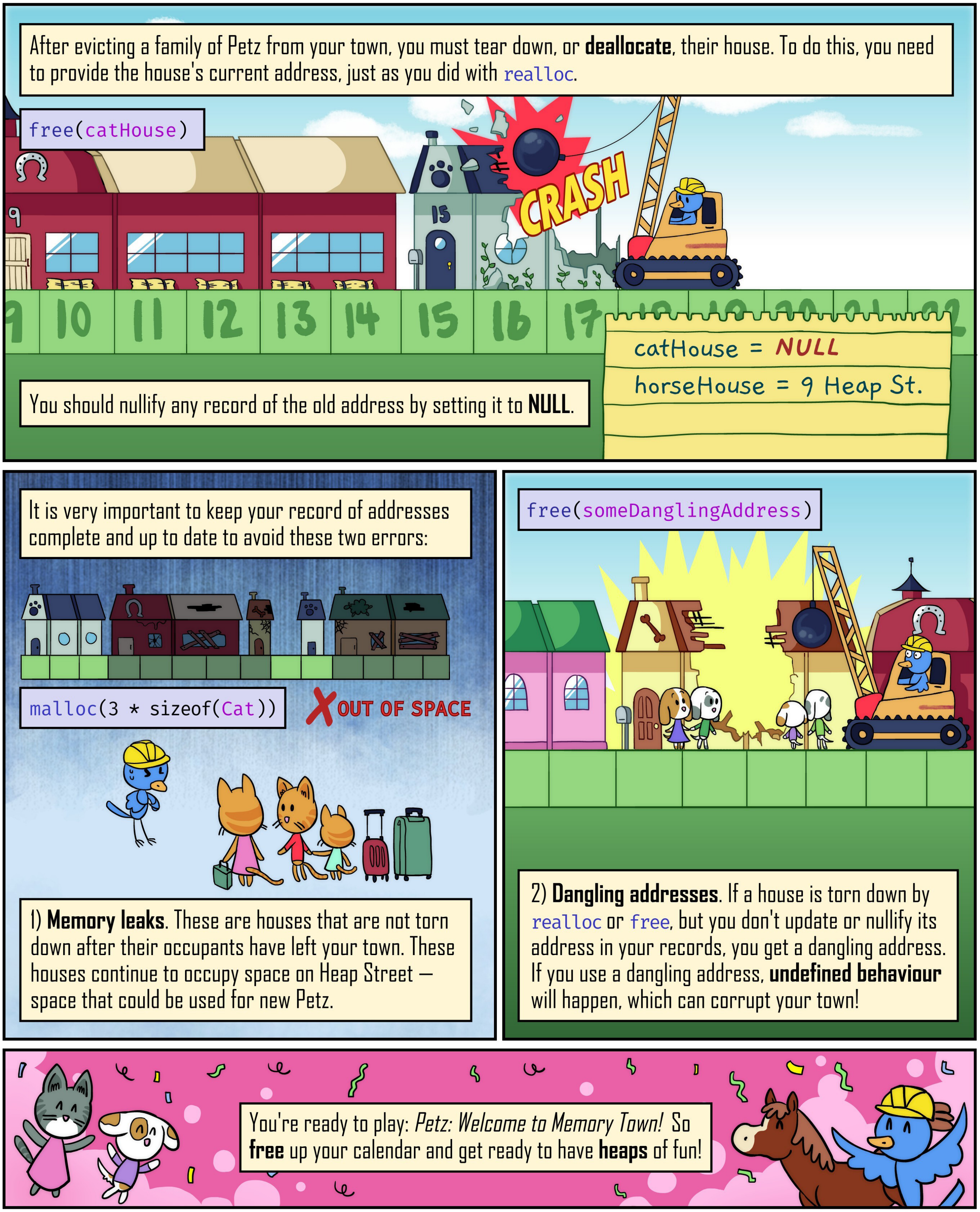
Sometimes during a reallocation, extra rooms will be added onto the end of the existing house...

...But sometimes there will not be enough space to grow the existing house. A new one will be built and the old destroyed.

```
realloc(catHouse, 4 * sizeof(Cat))
```

```
catHouse = 15 Heap St.
horseHouse = 9 Heap St.
```

To be safe, you should assume the address of the house has changed and overwrite you records with the address returned from `realloc`.


After evicting a family of Petz from your town, you must tear down, or **deallocate**, their house. To do this, you need to provide the house's current address, just as you did with realloc.
free(catHouse)
9
15
CRASH
9 10 11 12 13 14 15 16 17
catHouse = NULL
horseHouse = 9 Heap St.
You should nullify any record of the old address by setting it to **NULL**.
It is very important to keep your record of addresses complete and up to date to avoid these two errors:
malloc(3 * sizeof(Cat))
OUT OF SPACE
1) **Memory leaks**. These are houses that are not torn down after their occupants have left your town. These houses continue to occupy space on Heap Street — space that could be used for new Petz.
free(someDanglingAddress)
2) **Dangling addresses**. If a house is torn down by realloc or free, but you don't update or nullify its address in your records, you get a dangling address. If you use a dangling address, **undefined behaviour** will happen, which can corrupt your town!
You're ready to play: *Petz: Welcome to Memory Town!* So **free** up your calendar and get ready to have **heaps** of fun!

WOULDN'T YOU LIKE TO LOOK AT SOME WEIRD-LOOKING MICE: PART 1

THE COMPUTER KIND, NOT THE SQUEAKY MAMMAL KIND

If you have ever been made to use a standard mouse for hours and months on end, you know how much it fucking sucks ass! Primate hands were never meant to be locked in such unnatural and rigid contortions. It's a goddamn repulsion of natural order is what it is! Your hand and wrist pain? It's because of mice. Your trigger finger? Mice. The fact that you'll never live up to the overwhelming expectations of your father despite how hard you try and how desperate you are for his love, his recognition, his pride? ***Mice, mice, mice!***

…

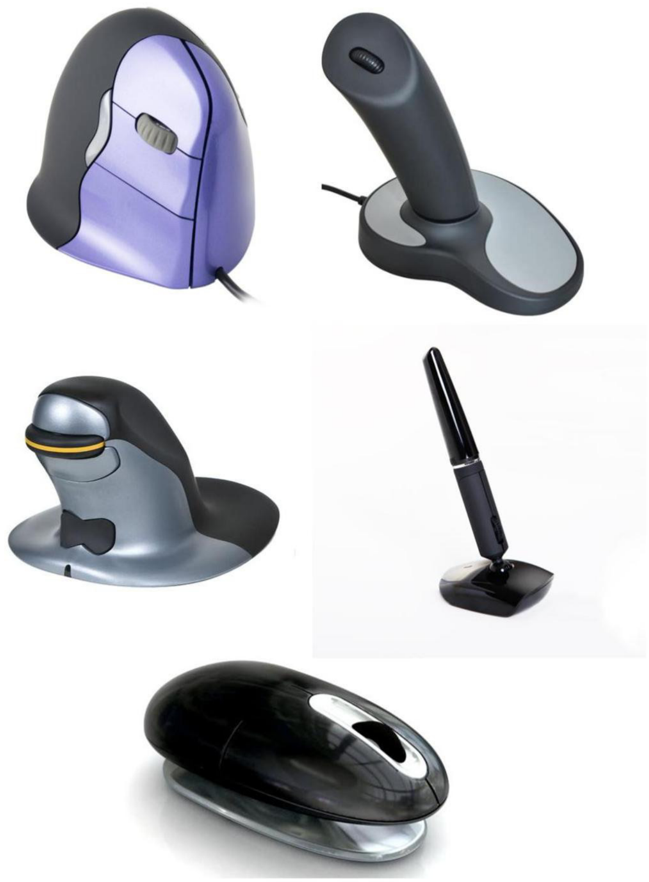
…And so the stage is set for the reveal of the heroes of this drama. *Yes!* Ergonomic and alternative mice! With their provocatively sculpted forms and ingeniously contrived gimmicks, each of their individual manufacturers promises that, indeed, *they've* divined the perfect shape of the pointing tool present in all of our lives. But who's really got the formula for hand-comfort heaven? That's for incorporeal omnipresent entities to know—and for us apes to find out.

Reader, let me be your guide as I take you through an overview of the alternative mice designs on the market today, on a rough scale from weird to weirder. Don't expect trackballs or trackpads or graphics tablets to be on this list. I'm sure you've at least *heard* of those, and also, I'm only covering *mice* designs. Here's one of my rules for living: if it's not sold as a “mouse,” it ain't a mouse!

OPTICAL MICE

Y'know how some mice with have red lights on the bottom? These are mice that convert physical motion of the mouse over a flat surface into cursor movement. Non-optical mice will be covered in Part 2, whenever the fuck I get around to writing that up. Keep your eyes peeled, kiddies!

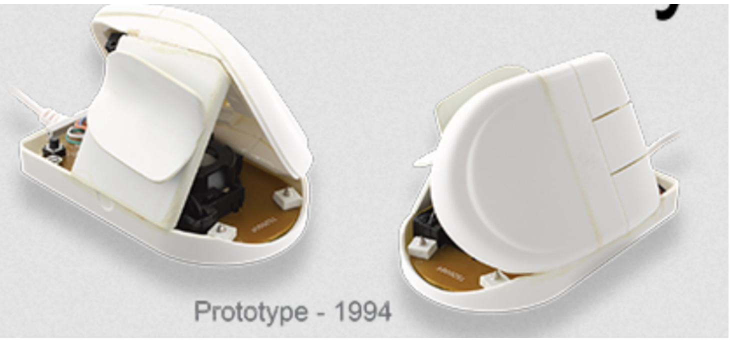
> The most common password in Germany is '123456'. If you're taking this course, you should know better and make that '12345678'.
>
> PROF. FLORIAN KERSCHBAUM



COUNTERCLOCKWISE, FROM TOP-LEFT. *VerticalMouse, Penguin, Whirl, Penclic, Rockstick 1.*

VERTICAL MICE

Take your mouse. Rotate it so that the bottom of it is perpendicular to the surface of the table. Boom—that's vertical mice in a nutshell.



EVOLUENT'S “VERTICALMOUSE” PROTOTYPE. *From what I can gather, the first of its kind.*

These allegedly work by keeping your wrist and forearm in a neutral, “handshake” position. There are gazillions of vertical mice on the market, with various degrees of tilt. That's all I have to say about these… VMs are downright pedestrian in the world of alt mice.

IS THAT A DILDO?

These mice have large, flat bottoms with a tall, upright handle that you grab onto like you're holding a laser gun—allowing for a neutral forearm position. Although some of these mice look like joysticks, you actually move them by shifting around the whole mouse on the desk as you would with a standard mouse.

The star of this category of mice is Posturite's “Penguin,” mainly because it looks like the chestburster from *Alien*. The Penguin also ambidextrous, meaning you can use it comfortably with both hands. Now you can finally give your poor, poor right hand a rest! (Unless you're one of the 10% of the left-handed 10% of the population who use a left-handed mouse, in which case: now you can finally give your poor, poor left hand a rest!)

Other competitors in this category, such as 3M's “Renaissance” and the “Rockstick 1,” are not ambidextrous. They *are* a lot more phallic than the Penguin though—so that's cool. Also, the Rockstick line's distinct attraction is less its form factor and more its mechanism for left/right-button clicks. Instead of clicking a button, you reduce finger fatigue by “rocking” the “stick” left or right. (Don't ergonomic mice manufacturers come up with the best names?)

SPEAKING OF LASER GUNS. *I found this incredible “Gaming Mouse for FPS and TPS Games” on Amazon after I looked up the Penguin. It may not be that good for gaming though: a reviewer gave it 0⁄5 on that metric since “[m]ouse spontaneously combusting while playing can cause you to lose.”*

PEN MICE

Just imagine a mouse, with all of its buttons and shit, in the shape of a pen. You're already 99% of the way there.

There are two main models of pen mice: the “Lychee” and the “Penclic.” The Lychee acts and looks pretty much how you'd expect, with the optical sensor located at the very “tip” of the pen. Meanwhile, the Penclic has a big “base” where the optical sensor is located, so moving it is kind of cumbersome. However, this means you can angle the handle of the Penclic however you want, while the Lychee's sensor starts to skip if you don't hold it upright.

Pen mice are great for precision work, but it's probably not the best idea to switch to this kind of mouse full-time. (Holding a pen for eight hours a day? Have you ever talked to a writer? Wait, hold on… Do writers even write with pen and paper nowadays, or is it all typing?) Don't expect to do a lot of note-taking or drawing with them either, due to how left-click is still a separate, physical button.

ODDS AND ENDS

The following models of mice adhere to principles that are seemingly only cared about by their individual manufacturers. Maybe other manufacturers are so intimidated by the sheer genius to even try their own versions?

- Smartfish's “Whirl” mouse—the mouse is mounted on a base and pivots to provide “fluid, air-like comfort.” Think bobo doll? Or a yoga ball for your hand?
- The “ErgoClick”—incorporates the two premises of “using your palm to click buttons” and “using both hands at once.” This is essentially two separate mice on opposite ends of your keyboard. Your right hand controls the cursor while the left hand controls clicks. As a side note, I wonder if it's possible to rig your computer up to have two cursors controlled by two different mice. Think of the multitasking possibilities!
- The “Hoverstop”—it vibrates when it detects that your hand is on it and you have not clicked or scrolled anything in the last ten seconds. Can the field of ergonomics be really reduced down to gimmicky tricks? Such is the question that the Hoverstop poses to the world…

Finchey

How would you like it if someone threw an exception at you and unwound your stack?

PROF. PETER BUHR

C++ IS JUST RACKET, IF YOU THINK ABOUT IT (PART 2)

Wow, you made it through the first part? You must enjoy pain, huh? Anyway, judgment aside, let's do the rest of this. I really need to convince you that C++ is as pure-functional as it gets. I know you'll believe me. Alright then:

What's at the core of a pure functional programming language? Lists? No, those are great, but not really "core". The real meat and potatoes of a pure functional language is made of three things:

1. Variables
2. Abstraction (lambdas)
3. Application (function application)

That's it. Variables, lambdas, and function application is literally all we need. Nothing else. Not lists, not even numbers! This bare-minimum is what we call the λ-calculus. With these three things alone, we can build anything else we want with these things alone. How? I'll give some of the usual definitions in Racket:

```
(define ZERO (λ (f) (λ (x) x)))
(define SUCC (λ (n) (λ (f) (λ (x) (n (f x))))))
(define TRUE (λ (x) (λ (y) x))
(define FALSE (λ (x) (λ (y) y))
```

Yeah, laugh it off, it says "`SUCC`", it's the word, it's funny. Good? Alright, "succ" means "successor". Let's move on. So aside from that, wow that's dense. So if you wanted three, you'd do `(SUCC (SUCC (SUCC ZERO)))`. You could also define all the usual logical operators, and even an "if-then-else" sort of thing, all with those three λ-calculus pillars.

Obviously, this is as pure as it gets. No fluff, no distractions, just the raw λ-calculus. The question that immediately follows, then, is "how can we do this in C++ with templates"? Well, here it is:

```
template<char Label>
struct Var {};

template<char Param, typename Body>
struct Lambda {};

template<typename Fun, typename Arg>
struct App {};
```

That's it? Well, kind of. These structs are basically completely empty, and exist only to hold the given parameters in their types. So our λ-calculus is going to center entirely around types, not using any values at all. So, let's translate our λ-calculus definitions from earlier to this new C++ template syntax:

```
using ZERO = Lambda<'f', Lambda<'x', Var<'x'>>>;
using SUCC = Lambda<'n', Lambda<'f', Lambda<'x',
                App<Var<'n'>, App<Var<'f'>, Var<'x'>>>>>>;
using TRUE = Lambda<'x', Lambda<'y', Var<'x'>>>;
using FALSE = Lambda<'x', Lambda<'y', Var<'y'>>>;
```

Refined. Mature. Pure functional. A gentleperson's programming language. I'm sure you can already see the advantages over those other languages like Racket and Haskell. Well, we're not done quite yet. We still need some mechanism for actually evaluating these expressions, right? We'd like some way to reduce a function application down to a value of some sort.

The key to doing this is the idea of a *closure*. A closure is basically just a pair that contains a function/lambda, as well as its *environment*. An environment is basically just a list of label-value pairs; a map from labels to the values they represent in a certain scope. We need an environment for each lambda since "x" in a deeper scope might mean something different than "x" in a shallower scope. We'll implement environments as a simple linked-list structure, each node consisting of one label-value mapping.

```
struct EnvEmpty {}; // "empty list"

template<char Label, typename Exp, typename EnvRest>
struct EnvNode {};
```

An example environment might look something like this:

```
EnvNode<'x', some_value_type,
    EnvNode<'y', another_value_type, EnvEmpty>>
```

We'll also want a way to traverse these environments to look up variables and the values to which they map. We'll use template specialization, introduced previously, to implement a basic linear list traversal meta-function:

```
template<char Label, typename Env>
struct EnvNodeFinder {};

template<char Label, typename Exp, typename EnvRest>
struct EnvNodeFinder<Label, EnvNode<Label, Exp, EnvRest>> {
    // case: found
    using value = Exp;
};

template<char Label>
struct EnvNodeFinder<Label, EnvEmpty> {
    // case: not found
};

template<char Label, char OtherLabel, typename Exp,
        typename EnvRest>
struct EnvNodeFinder<Label,
                    EnvNode<OtherLabel, Exp, EnvRest>> {
    // case: recursive case
    using value =
        typename EnvNodeFinder<Label, EnvRest>::value;
};
```

Alright, I'll be honest for a second and say this looks godawful. You can probably parse most of this on your own given the limited explanation I gave of specialization earlier, but one thing that might stand out is the random "`typename`" keyword thrown in after the "`using value =`" bit in the recursive case. Basically this is just because the compiler sometimes can't figure out on its own whether `Thing<T1, T2, …, Tn>::value` is a type or a value, for some struct `X` and parameters `T1, T2, …, Tn`. So we add "`typename`" to explicitly tell the compiler that it's supposed to be a type, and to evaluate it as such.

Okay, nice. The closure definition is very straight-forward now:

```
template<typename Fun, typename Env>
struct Closure {};
```

Again, technically an empty struct that exists solely to hold the function and environment in the type name. With that, we can finally write a big mechanism to put together these pieces and actually evaluate them down to a result. We'll implement this through a meta-function called *Eval*, decidedly not *Evil*. Well, here it is for `Var`:

```
template<typename Exp, typename Env>
struct Eval { using value = Exp; }; // Fallback

template<char Label, typename Env>
struct Eval<Var<Label>, Env> {
    using value =
        typename EnvNodeFinder<Label, Env>::value;
};
```

We're using template specialization here in the same way as before, and we're using our `EnvNodeFinder` meta-function to look up the given variable in the environment `Env`. We don't have to worry about if there are multiple variables with the same label at different scopes, because the algorithm will stop at the first one it finds, and the first one it finds will be the most recently added, which will be the one from the deepest scope. So, it works out without us having to worry about it.

Evaluating lambdas is also pretty easy. We just return the lambda, but wrapped in a closure with the environment:

```
template<char Param, typename Body, typename Env>
struct Eval<Lambda<Param, Body>, Env> {
    using value = Closure<Lambda<Param, Body>>, Env>;
};
```

We've almost achieved enlightenment. All we need now is to figure out how to evaluate applications. What should that involve? The essence is that, for `App<Fun, Arg>` we want to:

1. Evaluate/simplify `Fun`—the lambda—as much as possible.
2. Evaluate/simplify `Arg` as much as possible.
3. Produce the body of `Fun`, with the evaluated param/arg appended to the environment.

We could probably do this all in one big statement, but to have a clearer "separation of duties" and to not make the editors suffer too much *[Editor's Note: *cries*]*, we'll split the last bit of functionality into a helper meta-function:

```
// App: EvalAppHelper
template<typename, typename>
struct EvalAppHelper {};
// necessary unspecialized version with anonymous typenames

template<char Param, typename Body, typename Env,
         typename Arg>
struct EvalAppHelper<Closure<Lambda<Param, Body>, Env>,
                     Arg> {
    using value =
        typename Eval<Body, EnvNode<Param, Arg,
Env>>::value;
};

// App, proper
template<typename Fun, typename Arg, typename Env>
struct Eval<App<Fun, Arg>, Env> {
    using value =
        typename EvalAppHelper<
            typename Eval<Fun, Env>::value,
            typename Eval<Arg, Env>::value
        >::value;
};
```

HELL LANGUAGE. No, this is beautiful. It's perfect, of course it is. Okay. We're just about there. Here are a couple of convenience functions I'm going to define to make things clearer. Note that it uses a GCC compiler macro to print a type, so if you actually bother compiling this stuff, use GCC.

```
#include <iostream> // ugh
#include <string>   // so gross

template<typename T>
using DoEval = typename Eval<T, EnvEmpty>::value;

template<typename T>
void Print() {
    std::string type{__PRETTY_FUNCTION__}; // gcc macro
    std::cout
        << type.substr(23, type.find_first_of(']') - 23)
        << std::endl;
}
```

The most important part there is the "`DoEval`" alias, which just runs `Eval` with an empty environment when we explicitly call it. I'll also throw in a free "if-then-else" definition; see if you can translate it back to Racket and make sense of it:

```
using IFTHENELSE =
    Lambda<'p',
        Lambda<'a',
            Lambda<'b',
                App<App<Var<'p'>, Var<'a'>>,
                    Var<'b'>>>>>;
```

We're finally ready. We've reached the top of the mountain; the ultimate; the universe's magnum opus.

Before, in Racket, we had this:

```
(define mystery (λ (p) (if p 1 3)))
```

This was good, but we can do better. Lay gaze upon the ultimate, one true Pure Functional Programming Language In C++:

```
using MYSTERY =
    Lambda<'p',
        App<App<App<IFTHENELSE, TRUE>,
                App<SUCC, ZERO>>,
            App<SUCC, App<SUCC>, App<SUCC, ZERO>>>>;

Print<DoEval<App<MYSTERY, TRUE>>>();

// ⇒ Closure<
//      Lambda<'f',
//        Lambda<'x',
//          App<Var<'n'>,
//              App<Var<'f'>, Var<'x'>>>>>,
//      EnvNode<'n',
//        Closure<
//          Lambda<'f', Lambda<'x', Var<'x'>>>,
//          EnvNode<'p',
//            Closure<
//              Lambda<'x', Lambda<'y', Var<'x'>>>,
//              EnvEmpty>,
//            EnvEmpty>>,
//         EnvNode<'p',
//           Closure<
//             Lambda<'x', Lambda<'y', Var<'x'>>>,
//             EnvEmpty>,
//           EnvEmpty>>>
```

…which is equivalent to 1, obviously. So there we have it. Compile-time evaluation of our λ-calculus. Instant results at run-time. In C++. What a world we live in! As an included bonus, we now also know that templates are Turing complete, since the λ-calculus is Turing complete. Who needs Racket and Haskell anymore, honestly? The whole functional programming landscape would be so much better if we just used this all the time.

I think this illustrates very clearly that C++ is irrefutably the Best Pure Functional Programming Language In The World. It can do everything the alternatives do, but without any of the distractions and impurities: just as any hypothetical divine entity would have certainly intended.

If you manage to write anything useful with this, then I'll be shocked.

jeff

P.S. Templates are genuinely useful and interesting and have a lot of applications beyond this esoteric λ-calculus nonsense, primarily as a tool for abstraction. This article is the culmination of me spending the last two months going down a rabbit hole about template metaprogramming and all the crazy stuff you can do with it. I'd definitely recommend looking into it if you have the time and if any of the stuff here was even remotely, slightly interesting to you. It'll give you a greater appreciation for how the C++ language really works, and it'll even show you how a lot of common standard library code is actually implemented.

P.P.S. A particularly cool part of this rabbit hole is learning about move semantics and how functions like **std::move** and **std::forward** actually work. Figuring that stuff out will send you down a lot of different lanes trying to reckon with and understand really central parts of the language that don't get talked about very much. You might even end up having to read some sections of the C++ standard at some point!

P.P.P.S. Thank you to the editors for figuring out how the hell to display my code in a column-article format!! *[Editor's Note: You're welcome.]*

A CHALLENGE FOR JEFF

Sure, pure functional programming via the freakin' C++ ***template system*** is cool and all, but I think you can do better. So here's my challenge to you:

…

…

…

…

…

…

Can you implement monads with template metaprogramming?

I eagerly await your next series.

terrifiED

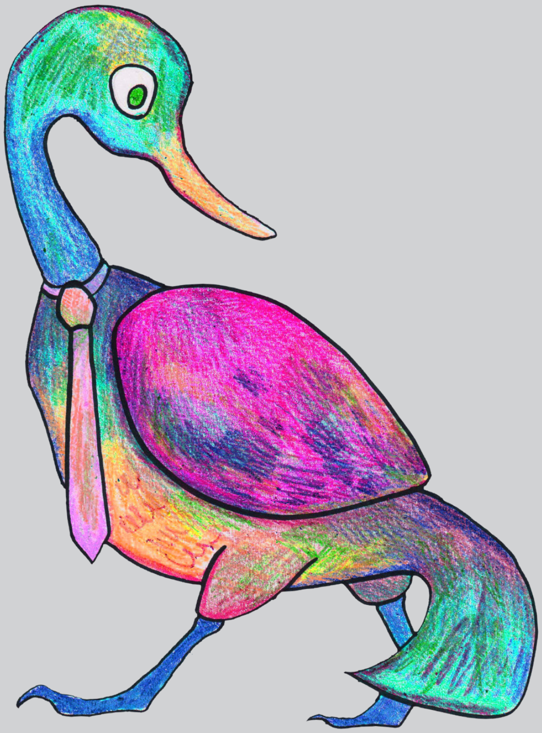
Racket is terrible, Racket wishes it could be Python.

CARMEN BRUNI

WRITER WHO HASN'T CONTRIBUTED TO mathNEWS ALL TERM FRANTICALLY WRITES SOMETHING FOR ISSUE 6

quantum goose

SHINY GOOSE



Cix

N HAIKUS TO END OFF THE ISSUE

TWENTY-SEVEN

Twenty plus seven
The page number of this page
It's an odd number

SHINY GOOSE

Look, a shiny goose
In the left-hand corner there
So iridescent

BOXES OF THE COLOUR BLACK

Pitch-black monotliths
Cradling ends of articles
Words written in white

N THINGS

Why only use N?
Twenty-five other letters
Beg us to use them

N THINGS, AGAIN

Why only use N?
It isn't “N Things” without
That special letter

ANTHOLOGIES

“When are the blasted
anthologies coming out?”
Three words: soon, I hope.

LIFE OF AN EDITOR

Life is sometimes rough
When you're editing **mathNEWS**
But you still do it

LIFE OF AN EDITOR (CONTINUED)

Why? Because **mathNEWS**
is love and **mathNEWS** is life
So it's not so bad

FILLER

Oh thank the heavens
This column's finally filled
My work here is done

clarifiED

lookAHEAD

SUN AUG 1	MON AUG 2	TUE AUG 3	WED AUG 4	THU AUG 5	FRI AUG 6	SAT AUG 7
	Civic Day	Drop/Add appointments for F2021 begin	National Chocolate Chip Cookie Day 🍪	Classes end Drop/Add period for F2021 begins	Drop with WF ends	Final examinations begin
SUN AUG 8	**MON AUG 9**	**TUE AUG 10**	**WED AUG 11**	**THU AUG 12**	**FRI AUG 13**	**SAT AUG 14**
International Cat Day 🐱	Olympics Closing Day Ceremony				Friday the 13th 🎃	

ISSN 0705-0410

UW'S BASTION OF ERUDITE THOUGHT SINCE 1973

mathNEWS is a normally fortnightly publication, funded by and responsible to the undergraduate math students of the University of Waterloo, as represented by the Mathematics Society of the University of Waterloo, hereafter referred to as MathSoc. **mathNEWS** is editorially independent of MathSoc. Content is the responsibility of the **mathNEWS** editors; however, any opinions expressed herein are those of the authors and not necessarily those of MathSoc or **mathNEWS**. Current and back issues of **mathNEWS** are available electronically via the World Wide Web at https://mathnews.uwaterloo.ca. Send your correspondence to: **mathNEWS**, MC3030, University of Waterloo, 200 University Ave. W., Waterloo, Ontario, Canada, N2L 3G1, or to userid mathnews@gmail.com on the Internet.

mathNEWS is overseen by the Board of Publications, an autonomous board of the Federation of Students, University of Waterloo, hereafter referred to as Feds. **mathNEWS** is editorially independent of Feds and the Board of Publications. **mathNEWS** has never been requested to withhold Improper Content as defined under Feds Policy 71.