Attribution/License

- Original Materials developed by Mike Shah, Ph.D. (www.mshah.io)
- This slideset and associated source code may not be distributed without prior written notice
- This slideset may not be mined using AI or algorithms and otherwise generating derivative products without permission.

Please do not redistribute slides/source without prior written permission.

DConf 2025

London, UK

All in on DLang: Why I pivoted to D for web, teaching, and graphics in 2025 and beyond!

Web:	mshah.io
YouTube	www.youtube.com/c/MikeShah
Social:	mikeshah.bsky.social
Courses:	courses.mshah.io
Talks:	http://tinyurl.com/mike-talks

40 minutes | Audience (For All)
10:00 - 10:40 Tues, Aug 19, 2025

Abstract (Which you already read :))

Talk Abstract: D is a general purpose programming language capable of building any type of software—from scripts to highly concurrent low-latency applications. Over the past year, Mike has put D to the test, developing all of his software from scripts, to web, to graphics in the D programming language. In this talk, he will discuss why he is ‘All in on DLang’, and why he is investing his time in D, which he believes makes him more competitive and a better engineer building better software.

In the talk, Mike will show a variety of software projects developed in the last year including: web 2.0, scripting, and even heavy-duty graphics rendering and game engine development for teaching. Audience members will leave with some inspiration, anecdotes of how he overcame various engineering challenges, and the motivation for why he thinks D should also be their primary programming language.

Your Tour Guide for Today

Mike Shah

- **Current Role:** Teaching Faculty at **Yale University**

  (Previously Teaching Faculty at Northeastern University)
  - **Teach/Research**: computer systems, graphics, geometry, game engine development, and software engineering.
- **Available for:**
  - **Contract work** in Gaming/Graphics Domains
    - e.g. tool building, plugins, code review
  - **Technical training** (virtual or onsite) in Modern C++, D, and topics in Performance or Graphics APIs
- **Fun**:
  - Guitar, running/weights, traveling, video games, and cooking are fun to talk to me about!

Web

www.mshah.io

YouTube

https://www.youtube.com/c/MikeShah

Non-Academic Courses

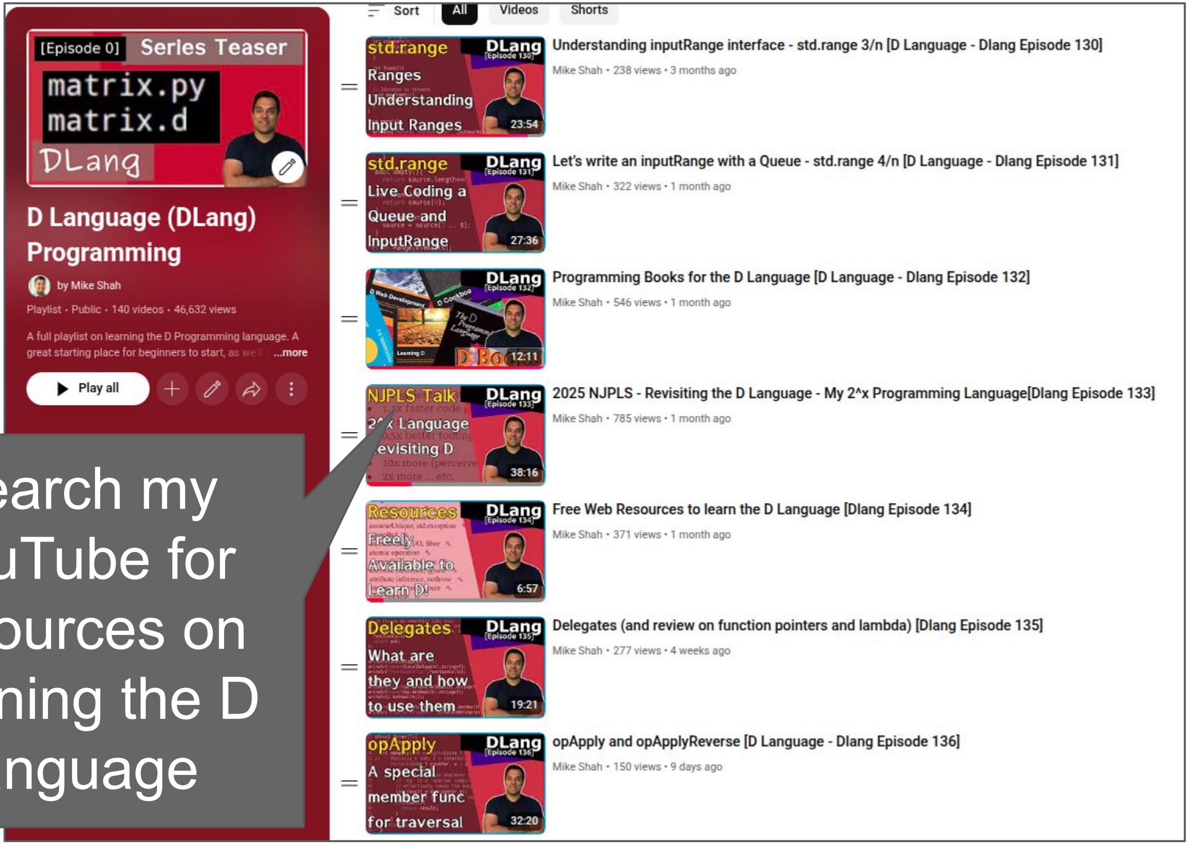
courses.mshah.io

Conference Talks

http://tinyurl.com/mike-talks

Find my programming content on YouTube

https://www.youtube.com/c/mikeshah



D Language (DLang) Programming

by Mike Shah

Playlist · Public · 140 videos · 46,632 views

A full playlist on learning the D Programming language. A great starting place for beginners to start, as we'll ...more

Understanding inputRange interface - std.range 3/n [D Language - Dlang Episode 130]
Mike Shah • 238 views • 3 months ago
23:54

Let's write an inputRange with a Queue - std.range 4/n [D Language - Dlang Episode 131]
Mike Shah • 322 views • 1 month ago
27:36

Programming Books for the D Language [D Language - Dlang Episode 132]
Mike Shah • 546 views • 1 month ago
12:11

2025 NJPLS - Revisiting the D Language - My 2^x Programming Language[Dlang Episode 133]
Mike Shah • 785 views • 1 month ago
38:16

Free Web Resources to learn the D Language [Dlang Episode 134]
Mike Shah • 371 views • 1 month ago
6:57

Delegates (and review on function pointers and lambda) [Dlang Episode 135]
Mike Shah • 277 views • 4 weeks ago
19:21

opApply and opApplyReverse [D Language - Dlang Episode 136]
Mike Shah • 150 views • 9 days ago
32:20

Search my YouTube for resources on learning the D language

Web

www.mshah.io

YouTube

https://www.youtube.com/c/MikeShah

Non-Academic Courses

courses.mshah.io

Conference Talks

http://tinyurl.com/mike-talks

Find my programming content on YouTube
https://www.youtube.com/c/mikeshah

D Language (DLang) Programming

by Mike Shah

Playlist • Public • 140 videos • 46,632 views

A full playlist on learning the D Programming language. A great starting place for beginners to start, as we'll ...more

Understanding inputRange interface - std.range 3/n [D Language - Dlang Episode 130]

Mike Shah • 238 views • 3 months ago

Free Web Resources to learn the D Language [Dlang Episode 134]

Mike Shah • 371 views • 1 month ago

Delegates (and review on function pointers and lambda) [Dlang Episode 135]

Mike Shah • 277 views • 4 weeks ago

opApply and opApplyReverse [D Language - Dlang Episode 136]

Mike Shah • 150 views • 9 days ago

Okay -- enough about me -- on to the talk!

Web
www.mshah.io

YouTube
https://www.youtube.com/c/MikeShah

Non-Academic Courses
courses.mshah.io

Conference Talks
http://tinyurl.com/mike-talks

So what’s this talk about -- “All in on Dlang” ? (1/2)

- I am going to show you some highlights of a few projects and code snippets of projects I’m working on in wildly different domains:
  - Web
  - Scripts
  - Game Engines
  - Graphics
  - Teaching
- But the real thing this talk is about is...

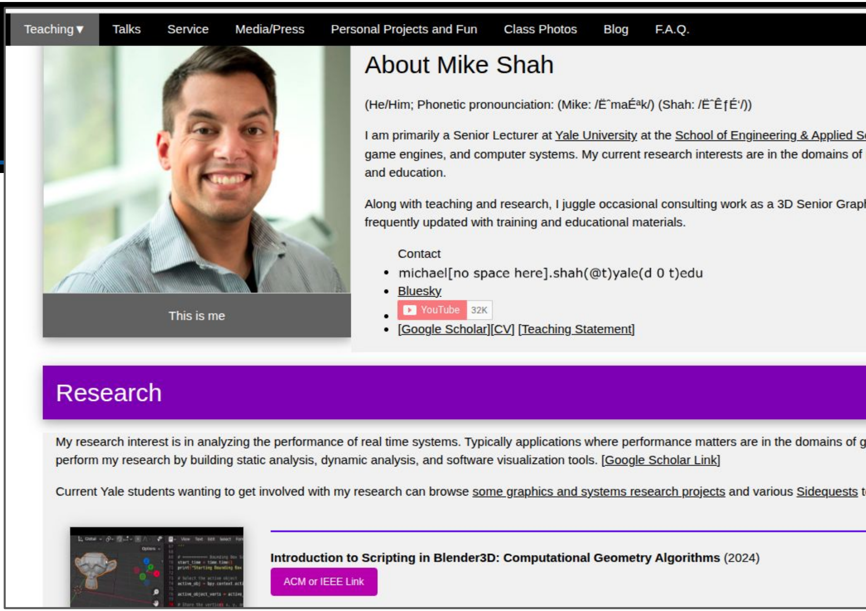
So what’s this talk about -- “All in on Dlang” ? (2/2)

- I am going to show you some highlights of a few projects and code snippets of projects I’m working on in wildly different domains:
  - Web
  - Scripts
  - Game Engines
  - Graphics
  - Teaching
- But the real thing this talk is about is...
- This is a talk about ignoring the noise.
  - Or rather finding the “right signal” in the noise when it comes to choosing a tool for programming and going all in.
- For me that tool has been the D language

Web Development in DLang: Part 1

Building a Web 2.0 Site

- For my web programming needs:
  - I needed to build a personal site and a website for my courses I teach
    - Reliability needs to be high (no crashing!)
    - Needs to handle ‘bursts’ of users (100 or so students) clicking around, loading images, etc.
    - And be cost efficient and secure



About Mike Shah

(He/Him; Phonetic pronounciation: (Mike: /ËˆmaÉªk/) (Shah: /ËˆÊƒÉ‘/))

I am primarily a Senior Lecturer at Yale University at the School of Engineering & Applied S
game engines, and computer systems. My current research interests are in the domains of
and education.

Along with teaching and research, I juggle occasional consulting work as a 3D Senior Grap
frequently updated with training and educational materials.

Contact

- michael[no space here].shah(@t)yale(d 0 t)edu
- Bluesky
- YouTube 32K
- [Google Scholar][CV] [Teaching Statement]

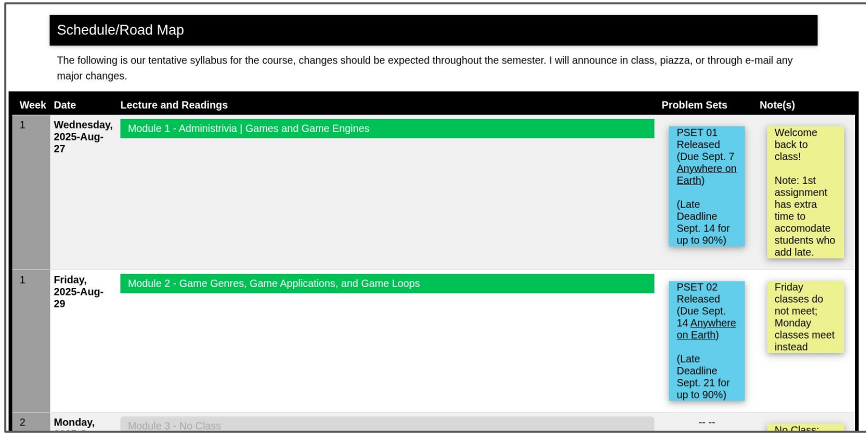
Research

My research interest is in analyzing the performance of real time systems. Typically applications where performance matters are in the domains of g
perform my research by building static analysis, dynamic analysis, and software visualization tools. [Google Scholar Link]

Current Yale students wanting to get involved with my research can browse some graphics and systems research projects and various Sidequests t

Introduction to Scripting in Blender3D: Computational Geometry Algorithms (2024)

ACM or IEEE Link



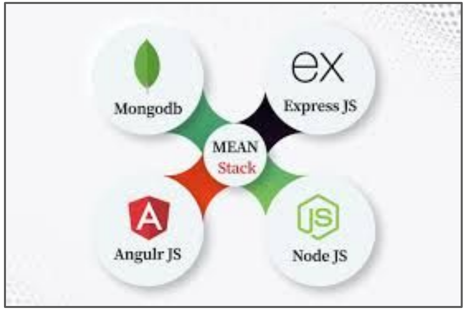
Schedule/Road Map

The following is our tentative syllabus for the course, changes should be expected throughout the semester. I will announce in class, piazza, or through e-mail any major changes.

Week	Date	Lecture and Readings	Problem Sets	Note(s)
1	**Wednesday, 2025-Aug-27**	Module 1 - Administrivia | Games and Game Engines	PSET 01 Released (Due Sept. 7 Anywhere on Earth) (Late Deadline Sept. 14 for up to 90%)	Welcome back to class! Note: 1st assignment has extra time to accomodate students who add late.
1	**Friday, 2025-Aug-29**	Module 2 - Game Genres, Game Applications, and Game Loops	PSET 02 Released (Due Sept. 14 Anywhere on Earth) (Late Deadline Sept. 21 for up to 90%)	Friday classes do not meet; Monday classes meet instead
2	**Monday,**	Module 3 - No Class	-- --	No Class:

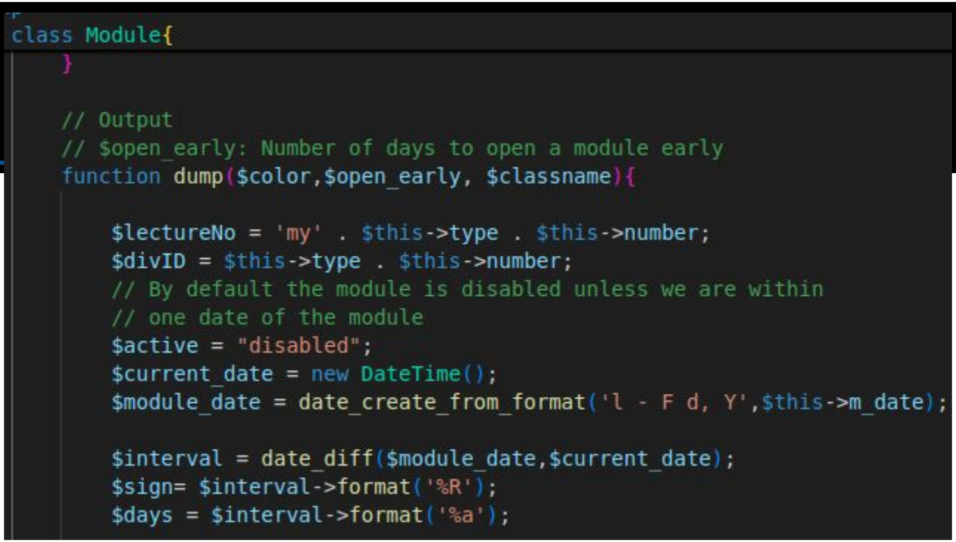
Web Development

- When I think about web development, these are the images and acronyms that come to mind
  - They are very popular
  - There are many books on these frameworks, technologies, and scripting languages
- Many of these programming stacks have existed for quite some time, and I would not be critiqued for choosing one

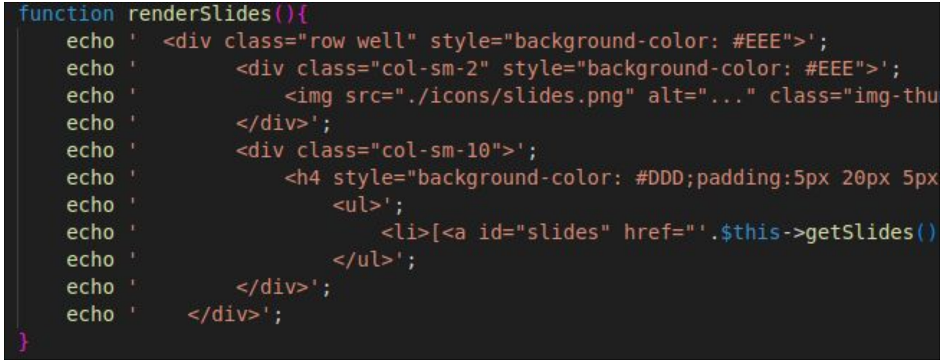


So I tried PHP for a bit… (2022-2024)

- You can see what it looks like
  - Just echo'ing out html from a few classes … oops!
  - And anyone who saw this code told me:
    - "hey, just use [Laravel/Drupel/WordPress/ some other framework]"
- Frankly I was not very excited about my codebase
  - I was not very curious/motivated to dive into frameworks that are constantly changing
    - The cognitive overhead is already quite high of learning another language
    - It's very healthy to use different languages, but I did not feel I could leverage much of my D knowledge here where I want to continue building expertise.



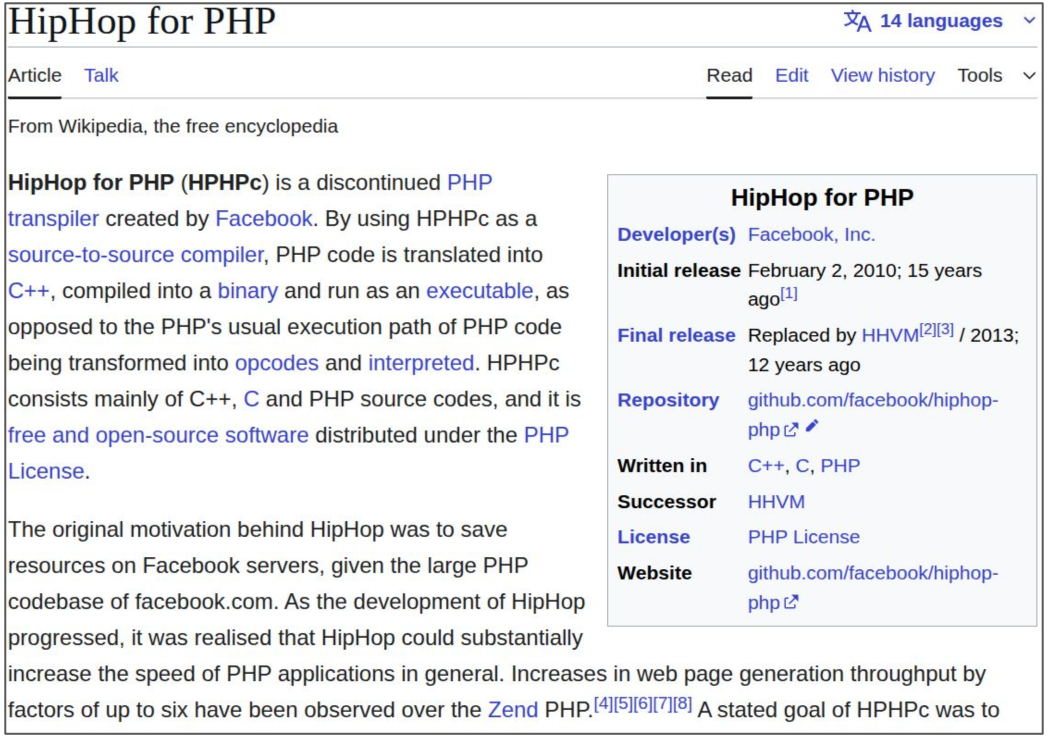
```
class Module{
    }

    // Output
    // $open_early: Number of days to open a module early
    function dump($color,$open_early, $classname){

        $lectureNo = 'my' . $this->type . $this->number;
        $divID = $this->type . $this->number;
        // By default the module is disabled unless we are within
        // one date of the module
        $active = "disabled";
        $current_date = new DateTime();
        $module_date = date_create_from_format('l - F d, Y',$this->m_date);

        $interval = date_diff($module_date,$current_date);
        $sign= $interval->format('%R');
        $days = $interval->format('%a');
```



```
function renderSlides(){
    echo '  <div class="row well" style="background-color: #EEE">';
    echo '      <div class="col-sm-2" style="background-color: #EEE">';
    echo '          <img src="./icons/slides.png" alt="..." class="img-thu
    echo '      </div>';
    echo '      <div class="col-sm-10">';
    echo '          <h4 style="background-color: #DDD;padding:5px 20px 5px
    echo '              <ul>';
    echo '                  <li>[<a id="slides" href="'.$this->getSlides()
    echo '              </ul>';
    echo '      </div>';
    echo '    </div>';
}
```

Compiled Language Web Development

- One specific example that has always stuck in my head for a long time came from Facebook*
- I remember thinking this ‘HipHop’ project was indeed ‘hip’.
  - This project (and several other ones) seemed like such common sense to me, use a compiled language
  - But I wanted to skip the whole ‘writing a transpiler’ part, and just write in native code to start
    - -- this seemed easier



HipHop for PHP

From Wikipedia, the free encyclopedia

HipHop for PHP (**HPHPc**) is a discontinued PHP transpiler created by Facebook. By using HPHPc as a source-to-source compiler, PHP code is translated into C++, compiled into a binary and run as an executable, as opposed to the PHP's usual execution path of PHP code being transformed into opcodes and interpreted. HPHPc consists mainly of C++, C and PHP source codes, and it is free and open-source software distributed under the PHP License.

HipHop for PHP	
Developer(s)	Facebook, Inc.
Initial release	February 2, 2010; 15 years ago[1]
Final release	Replaced by HHVM[2][3] / 2013; 12 years ago
Repository	github.com/facebook/hiphop-php
Written in	C++, C, PHP
Successor	HHVM
License	PHP License
Website	github.com/facebook/hiphop-php

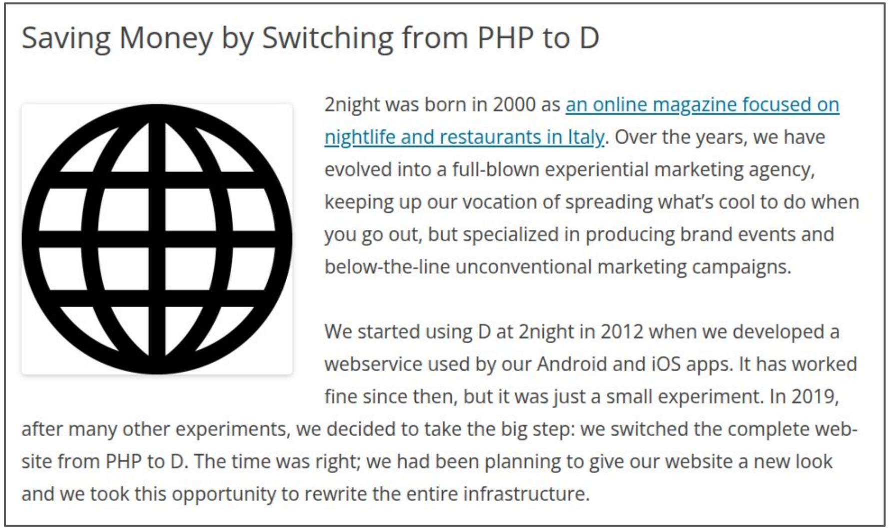
The original motivation behind HipHop was to save resources on Facebook servers, given the large PHP codebase of facebook.com. As the development of HipHop progressed, it was realised that HipHop could substantially increase the speed of PHP applications in general. Increases in web page generation throughput by factors of up to six have been observed over the Zend PHP.[4][5][6][7][8] A stated goal of HPHPc was to

https://en.wikipedia.org/wiki/HipHop_for_PHP

*The company is now known as ‘Meta’

Do compiled languages save money?

- So I was excited -- I could just use ‘D’ for my web development needs now.
  - Time invested in web programming would mean improving my D programming skills
- Before committing, I had one more thought -- and some folks in the D community were already a step ahead of me -- would it be more cost effective?
- Here’s the note from the article:
  - *“Switching from PHP to D meant we could cut in half the instance size of each Amazon AWS machine in our cloud. “*



Saving Money by Switching from PHP to D

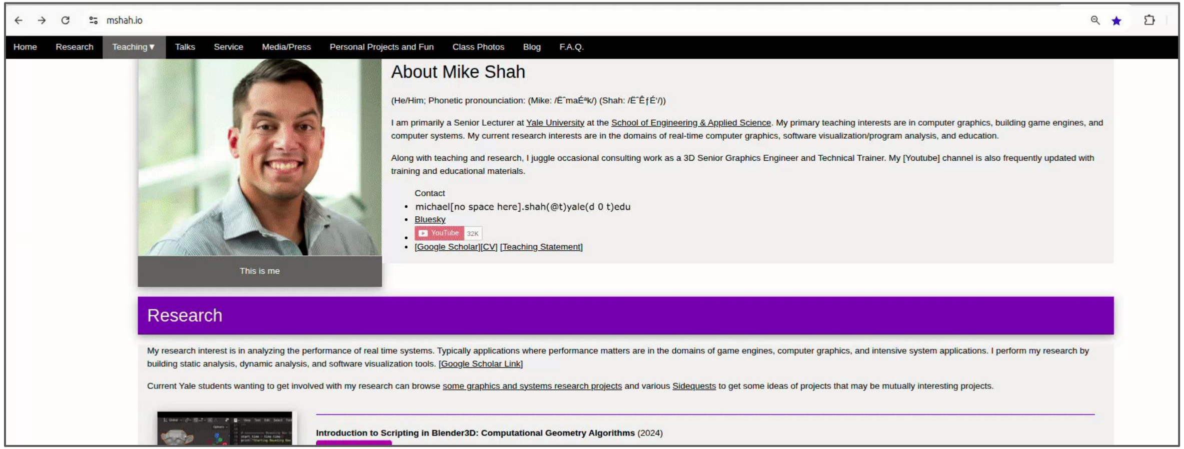
2night was born in 2000 as an online magazine focused on nightlife and restaurants in Italy. Over the years, we have evolved into a full-blown experiential marketing agency, keeping up our vocation of spreading what’s cool to do when you go out, but specialized in producing brand events and below-the-line unconventional marketing campaigns.

We started using D at 2night in 2012 when we developed a webservice used by our Android and iOS apps. It has worked fine since then, but it was just a small experiment. In 2019, after many other experiments, we decided to take the big step: we switched the complete website from PHP to D. The time was right; we had been planning to give our website a new look and we took this opportunity to rewrite the entire infrastructure.

https://dlang.org/blog/2019/09/30/saving-money-by-switching-from-php-to-d/

So I did it -- my website is in D (1/5)

- My website and courses are generated using D code.



About Mike Shah

(He/Him; Phonetic pronounciation: (Mike: /ËˆmaÉªk/) (Shah: /ËˆÊƒÉ‘/))

I am primarily a Senior Lecturer at Yale University at the School of Engineering & Applied Science. My primary teaching interests are in computer graphics, building game engines, and computer systems. My current research interests are in the domains of real-time computer graphics, software visualization/program analysis, and education.

Along with teaching and research, I juggle occasional consulting work as a 3D Senior Graphics Engineer and Technical Trainer. My [Youtube] channel is also frequently updated with training and educational materials.

Contact

- michael[no space here].shah(@t)yale(d 0 t)edu
- Bluesky
- YouTube 32K
- [Google Scholar][CV] [Teaching Statement]

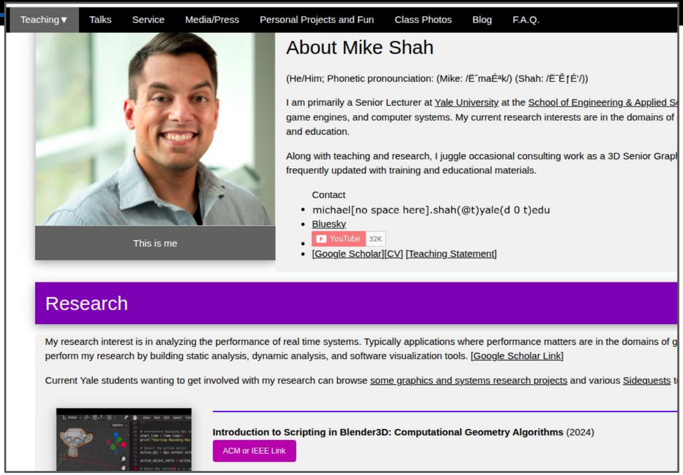
This is me

Research

My research interest is in analyzing the performance of real time systems. Typically applications where performance matters are in the domains of game engines, computer graphics, and intensive system applications. I perform my research by building static analysis, dynamic analysis, and software visualization tools. [Google Scholar Link]

Current Yale students wanting to get involved with my research can browse some graphics and systems research projects and various Sidequests to get some ideas of projects that may be mutually interesting projects.

Introduction to Scripting in Blender3D: Computational Geometry Algorithms (2024)



Teaching ▾ Talks Service Media/Press Personal Projects and Fun Class Photos Blog F.A.Q.

About Mike Shah

(He/Him; Phonetic pronounciation: (Mike: /ËˆmaÉªk/) (Shah: /ËˆÊƒÉ‘/))

I am primarily a Senior Lecturer at Yale University at the School of Engineering & Applied S
game engines, and computer systems. My current research interests are in the domains of
and education.

Along with teaching and research, I juggle occasional consulting work as a 3D Senior Grapl
frequently updated with training and educational materials.

Contact

- michael[no space here].shah(@t)yale(d 0 t)edu
- Bluesky
- YouTube 32K
- [Google Scholar][CV] [Teaching Statement]

This is me

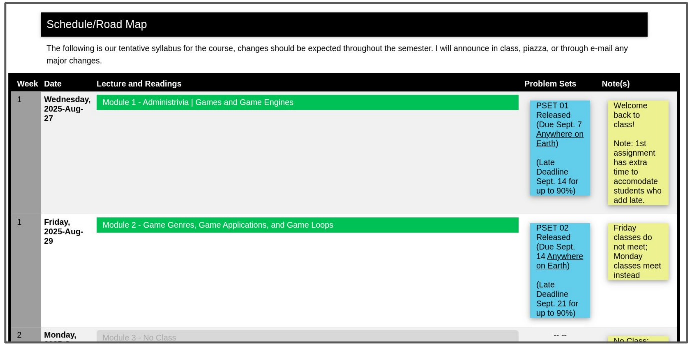
Research

My research interest is in analyzing the performance of real time systems. Typically applications where performance matters are in the domains of g
perform my research by building static analysis, dynamic analysis, and software visualization tools. [Google Scholar Link]

Current Yale students wanting to get involved with my research can browse some graphics and systems research projects and various Sidequests t

Introduction to Scripting in Blender3D: Computational Geometry Algorithms (2024)

ACM or IEEE Link



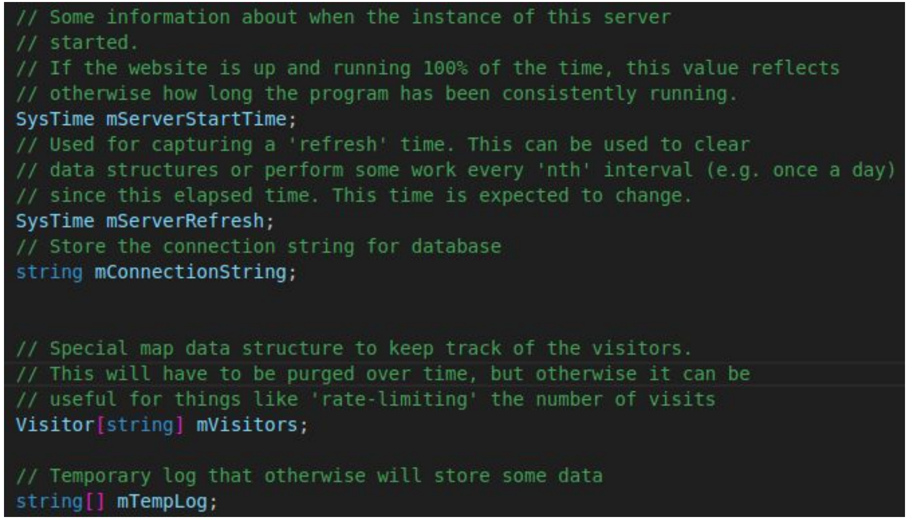
Schedule/Road Map

The following is our tentative syllabus for the course, changes should be expected throughout the semester. I will announce in class, piazza, or through e-mail any major changes.

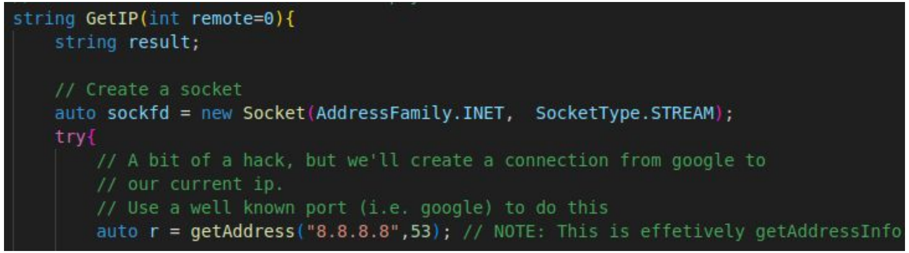
Week	Date	Lecture and Readings	Problem Sets	Note(s)
1	**Wednesday, 2025-Aug-27**	Module 1 - Administrivia | Games and Game Engines	PSET 01 Released (Due Sept. 7 Anywhere on Earth) (Late Deadline Sept. 14 for up to 90%)	Welcome back to class! Note: 1st assignment has extra time to accomodate students who add late.
1	**Friday, 2025-Aug-29**	Module 2 - Game Genres, Game Applications, and Game Loops	PSET 02 Released (Due Sept. 14 Anywhere on Earth) (Late Deadline Sept. 21 for up to 90%)	Friday classes do not meet; Monday classes meet instead
2	**Monday,**	Module 3 - No Class	-- --	No Class:

So I did it -- my website is in D (2/5)

- My web toolstack consists primarily of handyhttpd and Phobos (D standard library)
  - handyhttpd was easy to get started in, relatively well documented, and I could abstract on top of it
- I also used dpq2 for a postgresql database
  - This *just worked* and mirrored the libpq-dev C library for the most part, so I could benefit from the documentation there.
- Some findings -- to the next slide!



```
// Some information about when the instance of this server
// started.
// If the website is up and running 100% of the time, this value reflects
// otherwise how long the program has been consistently running.
SysTime mServerStartTime;
// Used for capturing a 'refresh' time. This can be used to clear
// data structures or perform some work every 'nth' interval (e.g. once a day)
// since this elapsed time. This time is expected to change.
SysTime mServerRefresh;
// Store the connection string for database
string mConnectionString;


// Special map data structure to keep track of the visitors.
// This will have to be purged over time, but otherwise it can be
// useful for things like 'rate-limiting' the number of visits
Visitor[string] mVisitors;

// Temporary log that otherwise will store some data
string[] mTempLog;
```

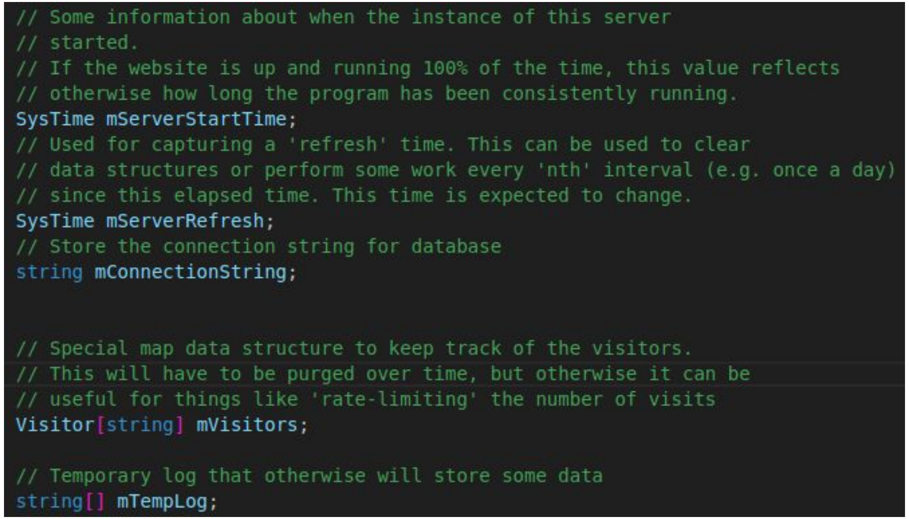


```
string GetIP(int remote=0){
    string result;

    // Create a socket
    auto sockfd = new Socket(AddressFamily.INET,  SocketType.STREAM);
    try{
        // A bit of a hack, but we'll create a connection from google to
        // our current ip.
        // Use a well known port (i.e. google) to do this
        auto r = getAddress("8.8.8.8",53); // NOTE: This is effetively getAddressInfo
```

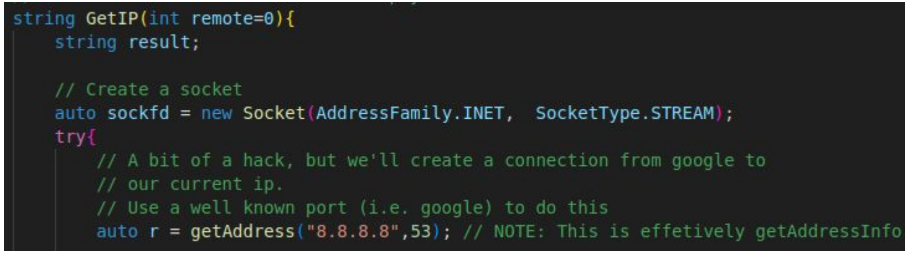
Here it is -- D Code! I don't have to remember another standard library, and I can reuse code I have!

So I did it -- my website is in D (3/5)

- As mentioned utilizing phobos has come in handy
  - I have been able to reuse some of my client/server networking code
- Handling exceptions for errors has been useful
- Other abstractions like creating maps for visitors and logs (that refresh over time) were very easy
  - Things again I was not thinking of when using Php -- why?
    - The answer is cognitive overload, I'm too busy wrestling with learning the language



```d
// Some information about when the instance of this server
// started.
// If the website is up and running 100% of the time, this value reflects
// otherwise how long the program has been consistently running.
SysTime mServerStartTime;
// Used for capturing a 'refresh' time. This can be used to clear
// data structures or perform some work every 'nth' interval (e.g. once a day)
// since this elapsed time. This time is expected to change.
SysTime mServerRefresh;
// Store the connection string for database
string mConnectionString;


// Special map data structure to keep track of the visitors.
// This will have to be purged over time, but otherwise it can be
// useful for things like 'rate-limiting' the number of visits
Visitor[string] mVisitors;

// Temporary log that otherwise will store some data
string[] mTempLog;
```

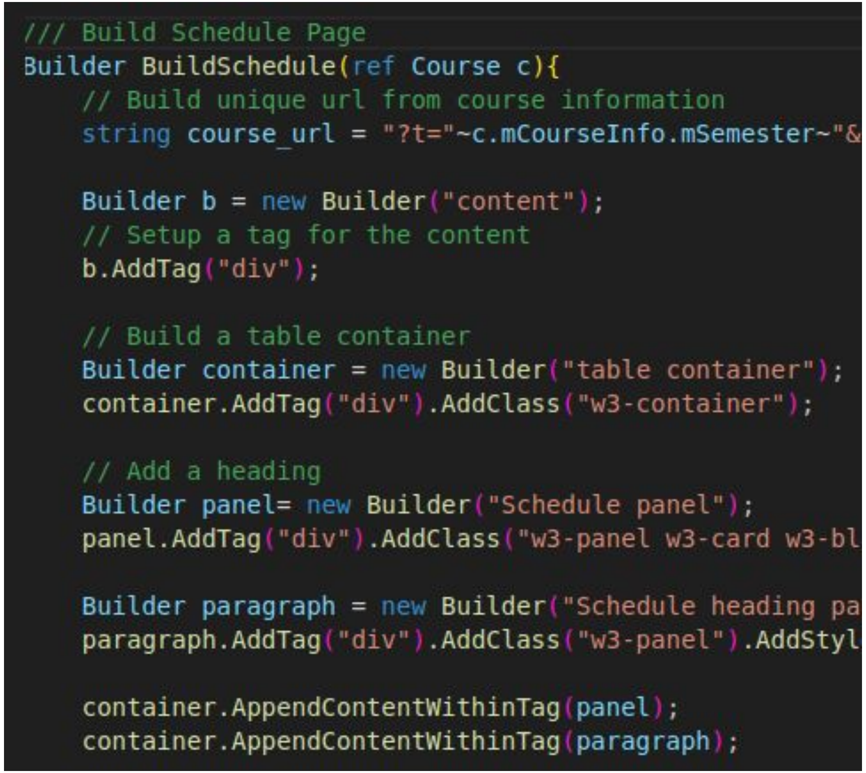


```d
string GetIP(int remote=0){
    string result;

    // Create a socket
    auto sockfd = new Socket(AddressFamily.INET,  SocketType.STREAM);
    try{
        // A bit of a hack, but we'll create a connection from google to
        // our current ip.
        // Use a well known port (i.e. google) to do this
        auto r = getAddress("8.8.8.8",53); // NOTE: This is effetively getAddressInfo
```

Here it is -- D Code! I don't have to remember another standard library, and I can reuse code I have!

So I did it -- my website is in D (4/5)

- Much of the website otherwise uses a sort of ‘builder’ pattern for generating html.
    - This is nice because I’m “echo’ing” out less html, can build strings more efficiently (either at compile-time, reserving memory, Appender, etc.)
        - I could have also done this in php, however, with D I can leverage compile-time function execution (CTFE) and templates for code generation
        - This should emulate something like vibe.d already does and potentially provide more performance improvements

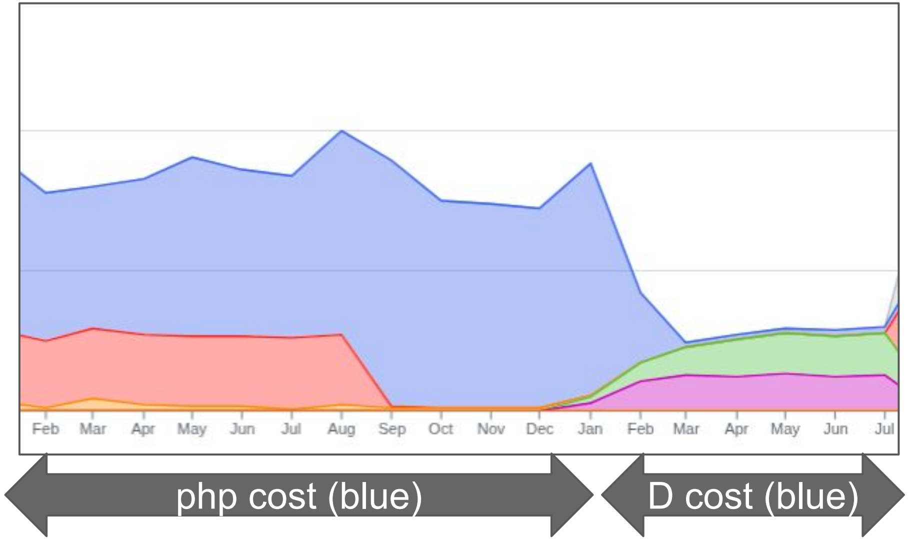


```
/// Build Schedule Page
Builder BuildSchedule(ref Course c){
    // Build unique url from course information
    string course_url = "?t="~c.mCourseInfo.mSemester~"&

    Builder b = new Builder("content");
    // Setup a tag for the content
    b.AddTag("div");

    // Build a table container
    Builder container = new Builder("table container");
    container.AddTag("div").AddClass("w3-container");

    // Add a heading
    Builder panel= new Builder("Schedule panel");
    panel.AddTag("div").AddClass("w3-panel w3-card w3-bl

    Builder paragraph = new Builder("Schedule heading pa
    paragraph.AddTag("div").AddClass("w3-panel").AddStyl

    container.AppendContentWithinTag(panel);
    container.AppendContentWithinTag(paragraph);
```

Here it is -- D Code! I don’t have to remember another standard library, and I can reuse code I have!

So I did it -- my website is in D ... and I saved money (5/5)

- While not admittingly a controlled study:
  - The cost is effectively free ($0.00x) for the bandwidth/compute verses about $0.24 per day.
- There's probably a more rigorous study to be done here
  - I do not see a reason currently to change languages/tech stack from D
- Using other technologies like nginx and potentially building my own in-memory cache are things I am thinking about.

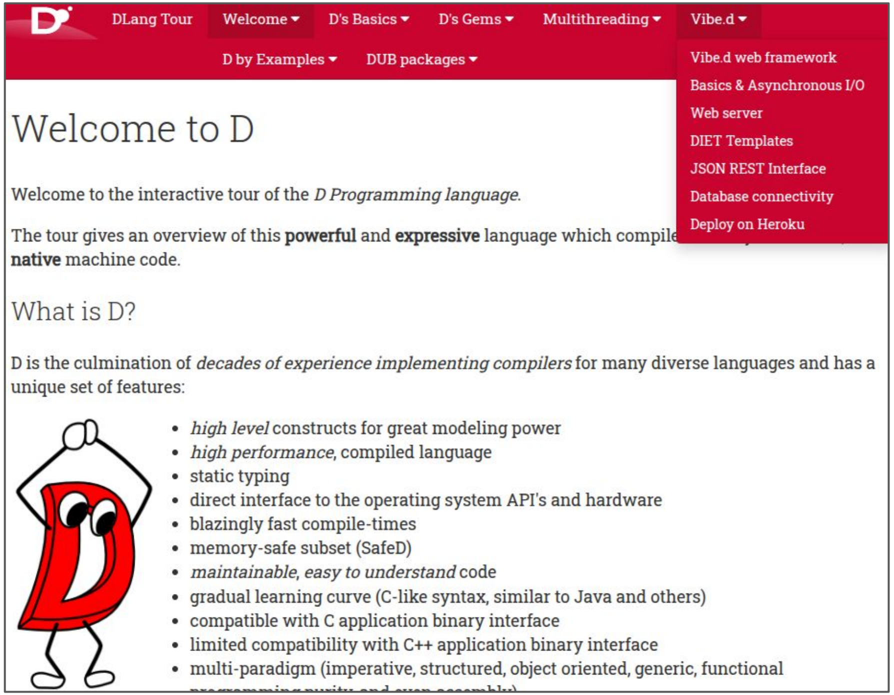


php cost (blue)

D cost (blue)

Note: Vibe.d

- Vibe.d is probably one of the more successful web frameworks that’s worth shouting out if you’re new and watching this-- I would encourage folks to check it out.
  - I’m glad a web framework is mentioned on the DLang tour
    - It is good to show off D’s flexibility
- There are also many other web frameworks, database, http server, and wasm libraries available
  - https://code.dlang.org/?sort=added&category=library.web&skip=0&limit=20



Welcome to D

Welcome to the interactive tour of the *D Programming language*.

The tour gives an overview of this **powerful** and **expressive** language which compile **native** machine code.

What is D?

D is the culmination of *decades of experience implementing compilers* for many diverse languages and has a unique set of features:

- *high level* constructs for great modeling power
- *high performance*, compiled language
- static typing
- direct interface to the operating system API's and hardware
- blazingly fast compile-times
- memory-safe subset (SafeD)
- *maintainable, easy to understand* code
- gradual learning curve (C-like syntax, similar to Java and others)
- compatible with C application binary interface
- limited compatibility with C++ application binary interface
- multi-paradigm (imperative, structured, object oriented, generic, functional programming purity and even assembly)

What’s next for D and web programming?

- I will record some tutorials on web programming and networking for YouTube at some point to help folks get started
- I suspect I will do more in the gaming space with D for networking
- This DConf 2014 talk by Funatics discusses D as a server for a mobile games company is one (of several) other inspirations for folks watching this talk.
    - (Mobile Gameserver Backend in D) https://dconf.org/2014/talks/dilly.html

Scripting with DLang: Part 2

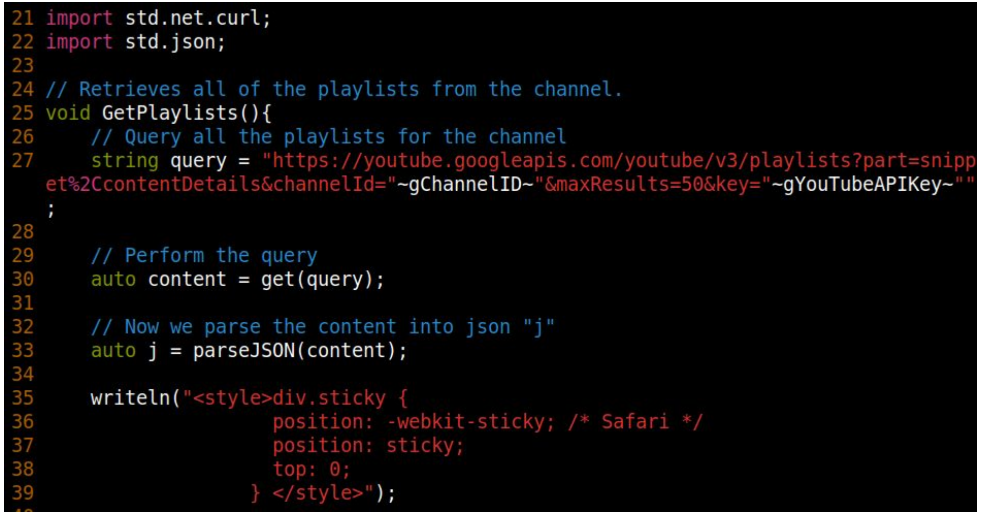
D as a scripting language

- I’ve stated this before -- but D has replaced every Python3* script that I use to use
    - ad-hoc build systems
    - Scripts using curl and various REST API’s to retrieve data
    - etc.
- I’ll just flip through a few little utilities so you get the idea.
- **The main point** -- is that often at work (industry or academic) I’ve been able to use whatever language I want for utilities.
    - I’ve found rdmd (and ldmd2) to be very effective

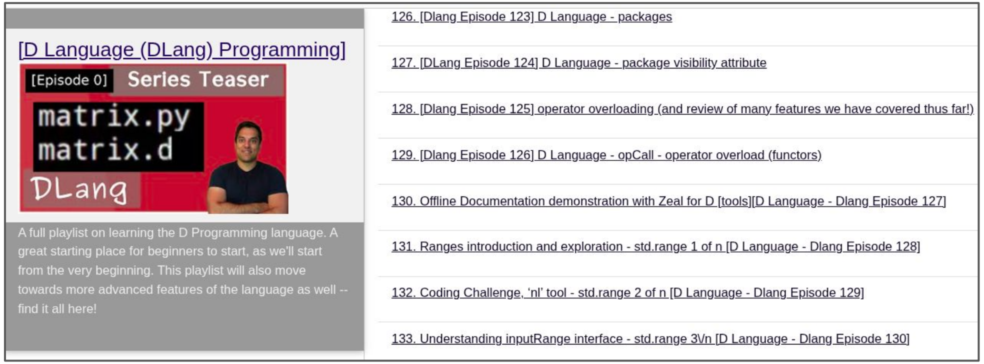
*Python3 is a lovely language, but through experience I’ve found I can iterate with D as fast, with equally malleable, but more scalable and performant code for tasks/scripts I use very frequently

Script 1 - Curl API to grab YouTube videos

- 127 line of code
- The code you see is enough to get a list of videos from a youTube channel
- Then you can parse the json
    - (This is a common thing to do with REST APIs)
- Then I have a single page with all my YouTube videos to easily ‘ctrl+f’ through
    - https://courses.mshah.io/pages/YouTubeLibrary#DLanguage(DLang)Programming



```d
import std.net.curl;
import std.json;

// Retrieves all of the playlists from the channel.
void GetPlaylists(){
    // Query all the playlists for the channel
    string query = "https://youtube.googleapis.com/youtube/v3/playlists?part=snippet%2CcontentDetails&channelId="~gChannelID~"&maxResults=50&key="~gYouTubeAPIKey~"";

    // Perform the query
    auto content = get(query);

    // Now we parse the content into json "j"
    auto j = parseJSON(content);

    writeln("<style>div.sticky {
                    position: -webkit-sticky; /* Safari */
                    position: sticky;
                    top: 0;
                } </style>");
```



[D Language (DLang) Programming]

A full playlist on learning the D Programming language. A great starting place for beginners to start, as we'll start from the very beginning. This playlist will also move towards more advanced features of the language as well -- find it all here!

126. [Dlang Episode 123] D Language - packages

127. [DLang Episode 124] D Language - package visibility attribute

128. [Dlang Episode 125] operator overloading (and review of many features we have covered thus far!)

129. [Dlang Episode 126] D Language - opCall - operator overload (functors)

130. Offline Documentation demonstration with Zeal for D [tools][D Language - Dlang Episode 127]

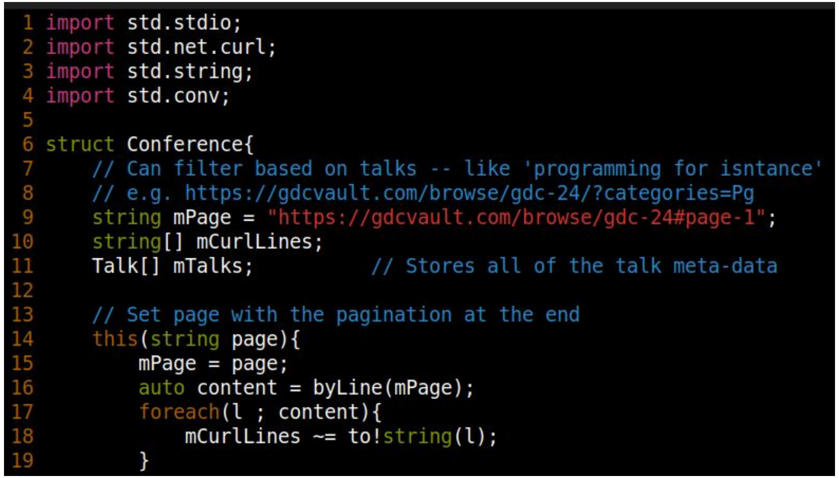
131. Ranges introduction and exploration - std.range 1 of n [D Language - Dlang Episode 128]

132. Coding Challenge, 'nl' tool - std.range 2 of n [D Language - Dlang Episode 129]

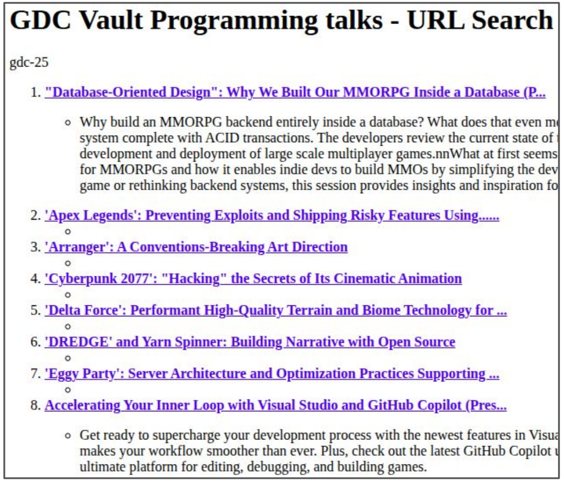
133. Understanding inputRange interface - std.range 3\n [D Language - Dlang Episode 130]

Script 2 - Game Development Conference (GDC) Talk Aggregator

- 110 lines of code
- I wrote a little URL aggregator to find talks I might be interested in regarding Game Development from the gdcvault.
    - Why? It was easy, and the gdcvault website was too hard to search.
    - Now I have every talk on 1 page, and can again, use ‘ctrl+f’ or vim chrome extensions to search



```
import std.stdio;
import std.net.curl;
import std.string;
import std.conv;

struct Conference{
    // Can filter based on talks -- like 'programming for isntance'
    // e.g. https://gdcvault.com/browse/gdc-24/?categories=Pg
    string mPage = "https://gdcvault.com/browse/gdc-24#page-1";
    string[] mCurlLines;
    Talk[] mTalks;          // Stores all of the talk meta-data

    // Set page with the pagination at the end
    this(string page){
        mPage = page;
        auto content = byLine(mPage);
        foreach(l ; content){
            mCurlLines ~= to!string(l);
        }
```



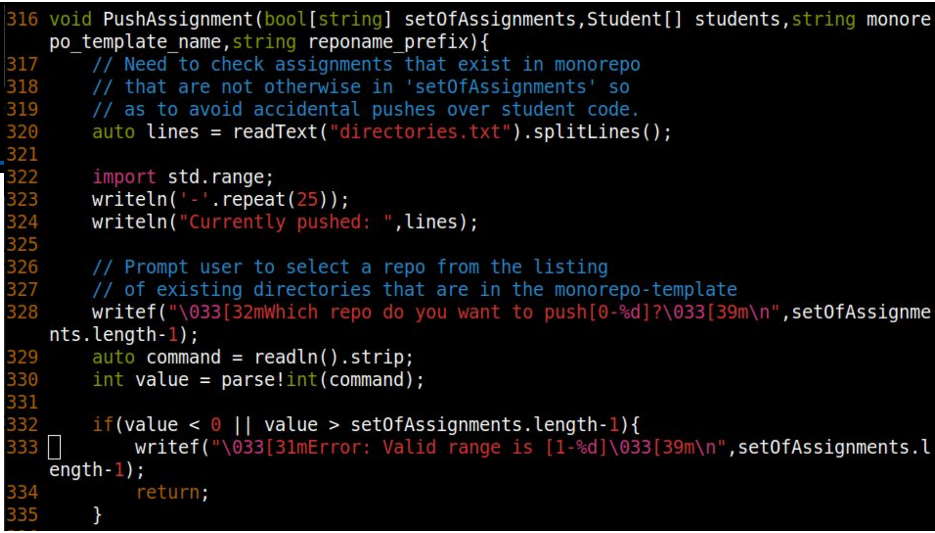
GDC Vault Programming talks - URL Search

gdc-25

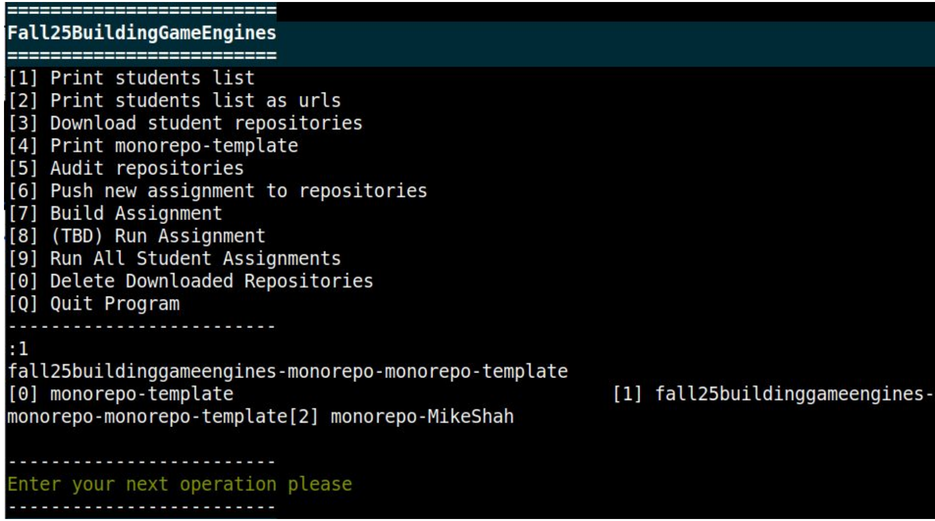
1. "Database-Oriented Design": Why We Built Our MMORPG Inside a Database (P...
    - Why build an MMORPG backend entirely inside a database? What does that even m
system complete with ACID transactions. The developers review the current state of
development and deployment of large scale multiplayer games.nnWhat at first seems
for MMORPGs and how it enables indie devs to build MMOs by simplifying the dev
game or rethinking backend systems, this session provides insights and inspiration fo
2. 'Apex Legends': Preventing Exploits and Shipping Risky Features Using......
    -
3. 'Arranger': A Conventions-Breaking Art Direction
    -
4. 'Cyberpunk 2077': "Hacking" the Secrets of Its Cinematic Animation
    -
5. 'Delta Force': Performant High-Quality Terrain and Biome Technology for ...
    -
6. 'DREDGE' and Yarn Spinner: Building Narrative with Open Source
    -
7. 'Eggy Party': Server Architecture and Optimization Practices Supporting ...
    -
8. Accelerating Your Inner Loop with Visual Studio and GitHub Copilot (Pres...
    - Get ready to supercharge your development process with the newest features in Visua
makes your workflow smoother than ever. Plus, check out the latest GitHub Copilot
ultimate platform for editing, debugging, and building games.

Script 3 - Github API

- 434 lines of code
- Github Classroom is what I use in my courses to deliver assignments to students (so they learn git)
- Provided is the command line tool I use to update repositories, gather assignments, and run student code.



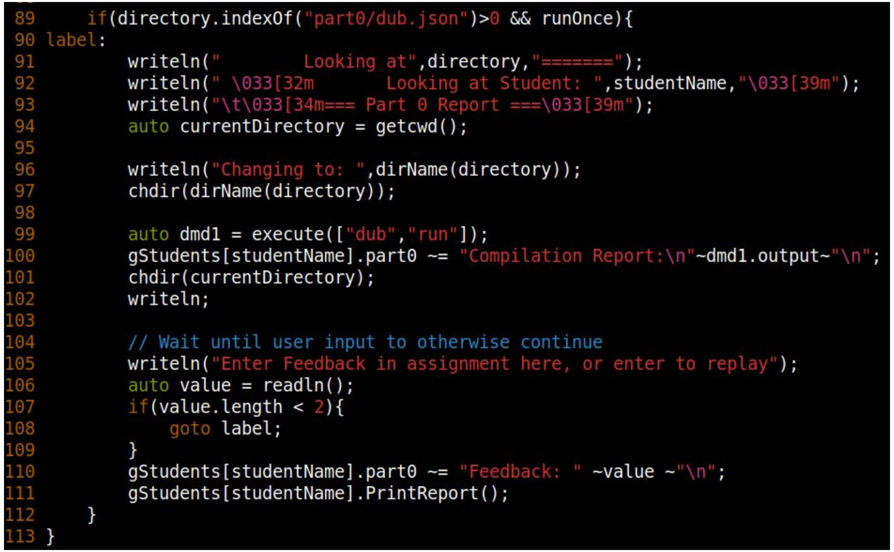
```
void PushAssignment(bool[string] setOfAssignments,Student[] students,string monorepo_template_name,string reponame_prefix){
    // Need to check assignments that exist in monorepo
    // that are not otherwise in 'setOfAssignments' so
    // as to avoid accidental pushes over student code.
    auto lines = readText("directories.txt").splitLines();

    import std.range;
    writeln('-'.repeat(25));
    writeln("Currently pushed: ",lines);

    // Prompt user to select a repo from the listing
    // of existing directories that are in the monorepo-template
    writef("\033[32mWhich repo do you want to push[0-%d]?\033[39m\n",setOfAssignments.length-1);
    auto command = readln().strip;
    int value = parse!int(command);

    if(value < 0 || value > setOfAssignments.length-1){
        writef("\033[31mError: Valid range is [1-%d]\033[39m\n",setOfAssignments.length-1);
        return;
    }
```



```
=========================
Fall25BuildingGameEngines
=========================
[1] Print students list
[2] Print students list as urls
[3] Download student repositories
[4] Print monorepo-template
[5] Audit repositories
[6] Push new assignment to repositories
[7] Build Assignment
[8] (TBD) Run Assignment
[9] Run All Student Assignments
[0] Delete Downloaded Repositories
[Q] Quit Program
-------------------------
:1
fall25buildinggameengines-monorepo-monorepo-template
[0] monorepo-template                                   [1] fall25buildinggameengines-
monorepo-monorepo-template[2] monorepo-MikeShah

-------------------------
Enter your next operation please
-------------------------
```

Script 4 - Autograder / Feedback Scripts

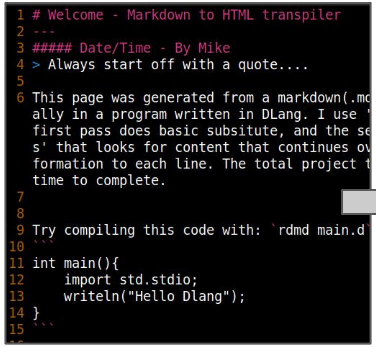
- 141 lines of code
- One of the beautiful things about D is that there exists a default package manager -- dub.
    - ‘dub’ provides a standard way to build projects
- This script executes ‘dub’ separate process (using std.process)
- I can also re-run assignments interactively
    - I’m highlighting this -- because it’s probably my first and only use of **goto** in my history of D code.
        - It was there when I needed it!



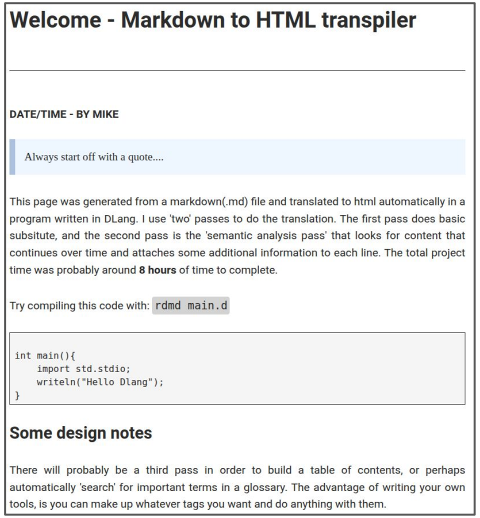
```
    if(directory.indexOf("part0/dub.json")>0 && runOnce){
label:
        writeln("        Looking at",directory,"=======");
        writeln(" \033[32m       Looking at Student: ",studentName,"\033[39m");
        writeln("\t\033[34m=== Part 0 Report ===\033[39m");
        auto currentDirectory = getcwd();

        writeln("Changing to: ",dirName(directory));
        chdir(dirName(directory));

        auto dmd1 = execute(["dub","run"]);
        gStudents[studentName].part0 ~= "Compilation Report:\n"~dmd1.output~"\n";
        chdir(currentDirectory);
        writeln;

        // Wait until user input to otherwise continue
        writeln("Enter Feedback in assignment here, or enter to replay");
        auto value = readln();
        if(value.length < 2){
            goto label;
        }
        gStudents[studentName].part0 ~= "Feedback: " ~value ~"\n";
        gStudents[studentName].PrintReport();
    }
}
```

Script 5 - markdown to html transpiler

- 366 lines of code, 8 hours of work
- Mostly using an associative array, and textual substitution



~~~
# Welcome - Markdown to HTML transpiler
---
##### Date/Time - By Mike
> Always start off with a quote....

This page was generated from a markdown(.md
ally in a program written in DLang. I use '
first pass does basic subsitute, and the se
s' that looks for content that continues ov
formation to each line. The total project t
time to complete.


Try compiling this code with: `rdmd main.d`
```
int main(){
    import std.stdio;
    writeln("Hello Dlang");
}
```
~~~



Welcome - Markdown to HTML transpiler

DATE/TIME - BY MIKE

> Always start off with a quote....

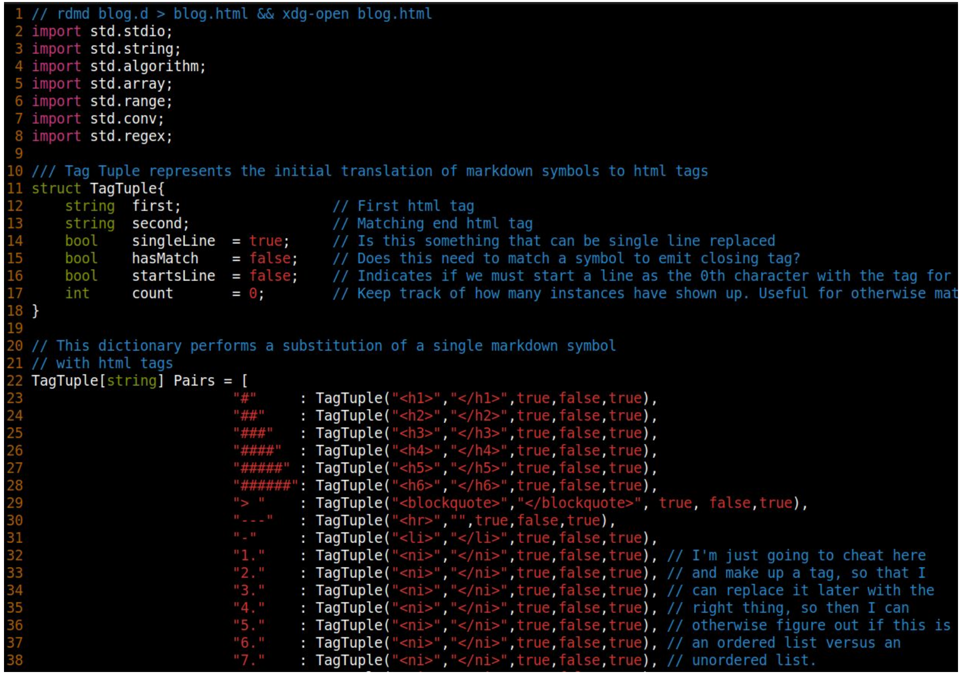
This page was generated from a markdown(.md) file and translated to html automatically in a program written in DLang. I use 'two' passes to do the translation. The first pass does basic subsitute, and the second pass is the 'semantic analysis pass' that looks for content that continues over time and attaches some additional information to each line. The total project time was probably around **8 hours** of time to complete.

Try compiling this code with: `rdmd main.d`

```
int main(){
    import std.stdio;
    writeln("Hello Dlang");
}
```

Some design notes

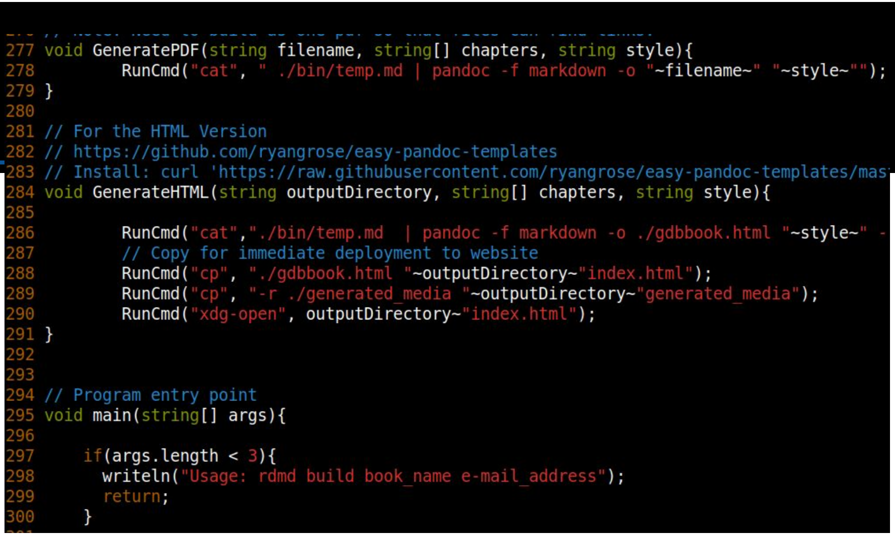
There will probably be a third pass in order to build a table of contents, or perhaps automatically 'search' for important terms in a glossary. The advantage of writing your own tools, is you can make up whatever tags you want and do anything with them.



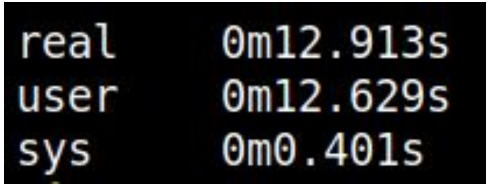
```
// rdmd blog.d > blog.html && xdg-open blog.html
import std.stdio;
import std.string;
import std.algorithm;
import std.array;
import std.range;
import std.conv;
import std.regex;

/// Tag Tuple represents the initial translation of markdown symbols to html tags
struct TagTuple{
    string  first;                  // First html tag
    string  second;                 // Matching end html tag
    bool    singleLine  = true;     // Is this something that can be single line replaced
    bool    hasMatch    = false;    // Does this need to match a symbol to emit closing tag?
    bool    startsLine  = false;    // Indicates if we must start a line as the 0th character with the tag for
    int     count       = 0;        // Keep track of how many instances have shown up. Useful for otherwise mat
}

// This dictionary performs a substitution of a single markdown symbol
// with html tags
TagTuple[string] Pairs = [
                        "#"     : TagTuple("<h1>","</h1>",true,false,true),
                        "##"    : TagTuple("<h2>","</h2>",true,false,true),
                        "###"   : TagTuple("<h3>","</h3>",true,false,true),
                        "####"  : TagTuple("<h4>","</h4>",true,false,true),
                        "#####" : TagTuple("<h5>","</h5>",true,false,true),
                        "######": TagTuple("<h6>","</h6>",true,false,true),
                        "> "    : TagTuple("<blockquote>","</blockquote>", true, false,true),
                        "---"   : TagTuple("<hr>","",true,false,true),
                        "-"     : TagTuple("<li>","</li>",true,false,true),
                        "1."    : TagTuple("<ni>","</ni>",true,false,true), // I'm just going to cheat here
                        "2."    : TagTuple("<ni>","</ni>",true,false,true), // and make up a tag, so that I
                        "3."    : TagTuple("<ni>","</ni>",true,false,true), // can replace it later with the
                        "4."    : TagTuple("<ni>","</ni>",true,false,true), // right thing, so then I can
                        "5."    : TagTuple("<ni>","</ni>",true,false,true), // otherwise figure out if this is
                        "6."    : TagTuple("<ni>","</ni>",true,false,true), // an ordered list versus an
                        "7."    : TagTuple("<ni>","</ni>",true,false,true), // unordered list.
```

Script 6 - Book Builder

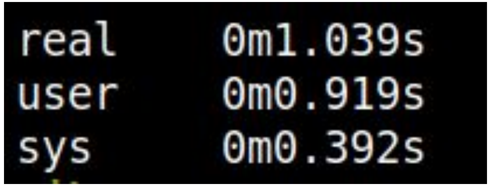
- 219 lines of code
- Build system for Pandoc to generate a book I'm working on.
- The real magic is **.parallel** for data processing in this case
    - About a 12x speedup by adding .parallel
        - Note: Variations of .parallel(1) or .parallel(4) yielded same result



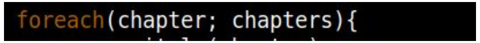
```
// Note: Need to build as one pdf so that files can find links.
void GeneratePDF(string filename, string[] chapters, string style){
        RunCmd("cat", " ./bin/temp.md | pandoc -f markdown -o "~filename~" "~style~"");
}

// For the HTML Version
// https://github.com/ryangrose/easy-pandoc-templates
// Install: curl 'https://raw.githubusercontent.com/ryangrose/easy-pandoc-templates/mas
void GenerateHTML(string outputDirectory, string[] chapters, string style){

        RunCmd("cat","./bin/temp.md  | pandoc -f markdown -o ./gdbbook.html "~style~" -
        // Copy for immediate deployment to website
        RunCmd("cp", "./gdbbook.html "~outputDirectory~"index.html");
        RunCmd("cp", "-r ./generated_media "~outputDirectory~"generated_media");
        RunCmd("xdg-open", outputDirectory~"index.html");
}


// Program entry point
void main(string[] args){

    if(args.length < 3){
      writeln("Usage: rdmd build book_name e-mail_address");
      return;
    }
```



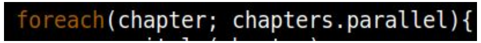
```
real    0m12.913s
user    0m12.629s
sys     0m0.401s
```



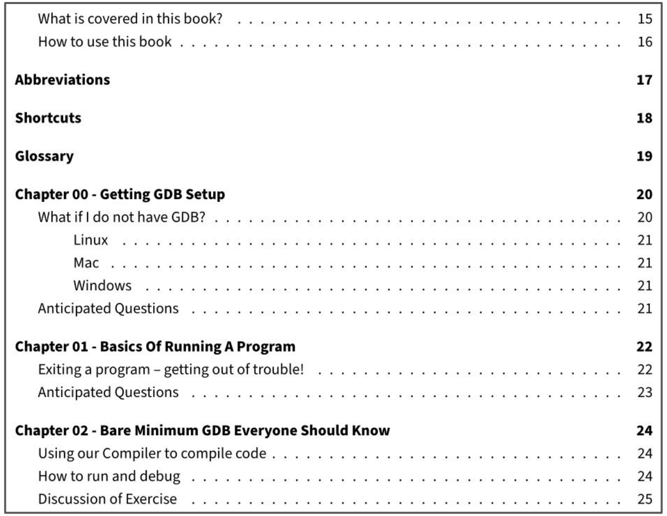
```
real    0m1.039s
user    0m0.919s
sys     0m0.392s
```



```
foreach(chapter; chapters){
```

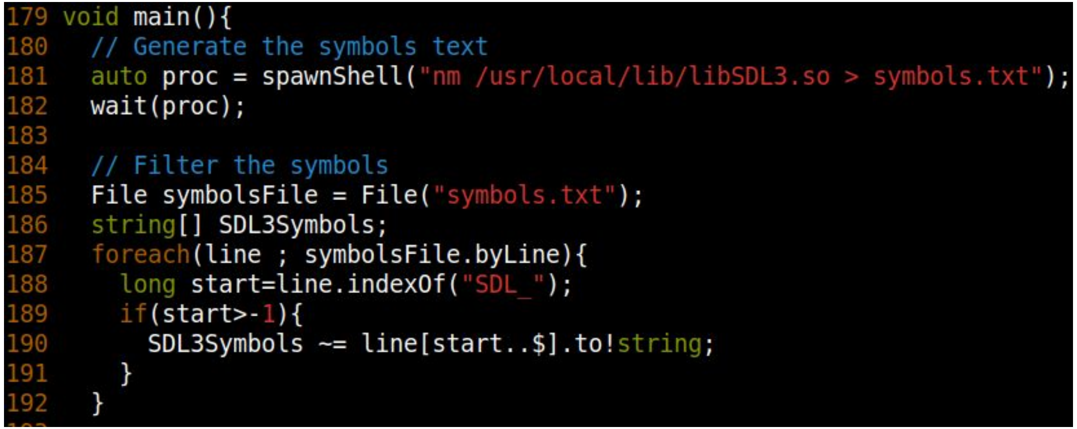


```
foreach(chapter; chapters.parallel){
```

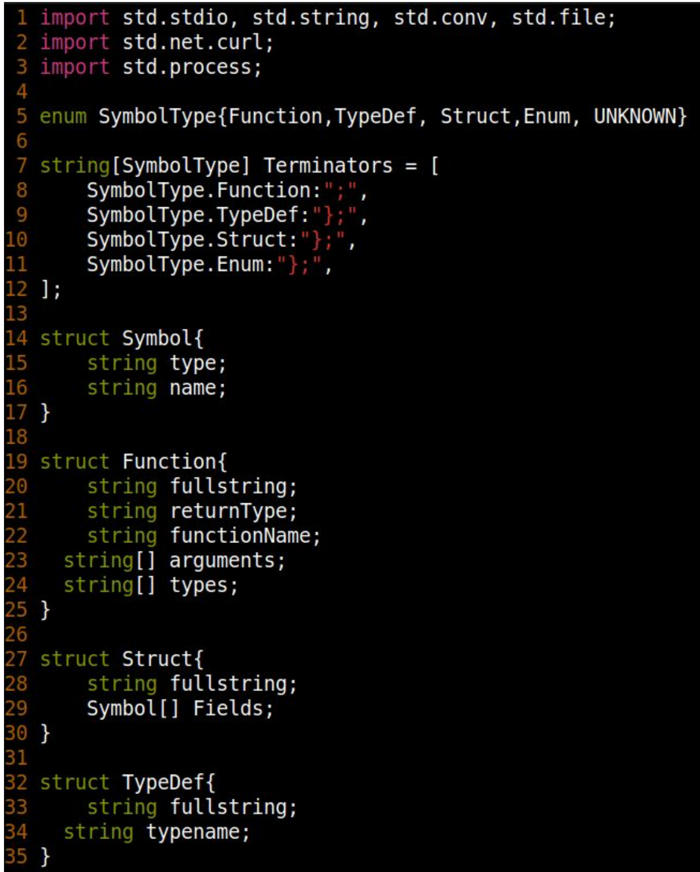


Script 7 - SDL3 C Binding Generator (1/2)

- This is a half-finished script that uses ‘nm’ to look up the names of symbols in a library (e.g. libSDL3)
- The (wild) idea is then to then generate an SDL3 binding all in one file with documentation



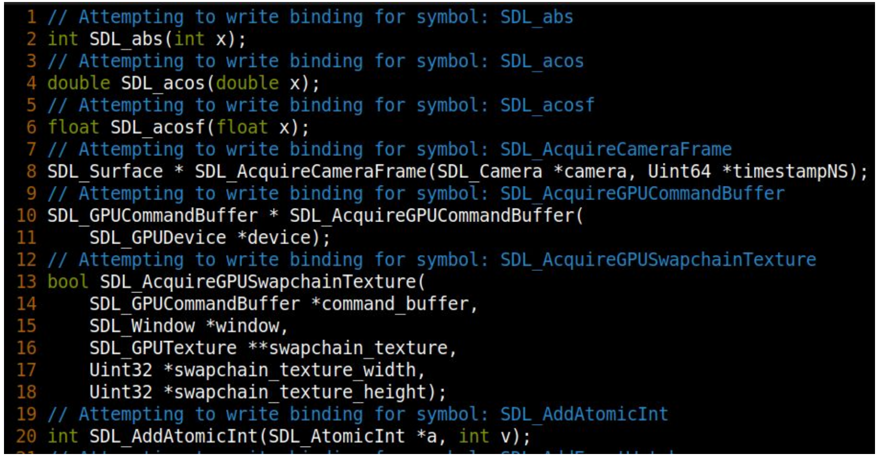
```
void main(){
  // Generate the symbols text
  auto proc = spawnShell("nm /usr/local/lib/libSDL3.so > symbols.txt");
  wait(proc);

  // Filter the symbols
  File symbolsFile = File("symbols.txt");
  string[] SDL3Symbols;
  foreach(line ; symbolsFile.byLine){
    long start=line.indexOf("SDL_");
    if(start>-1){
      SDL3Symbols ~= line[start..$].to!string;
    }
  }
```



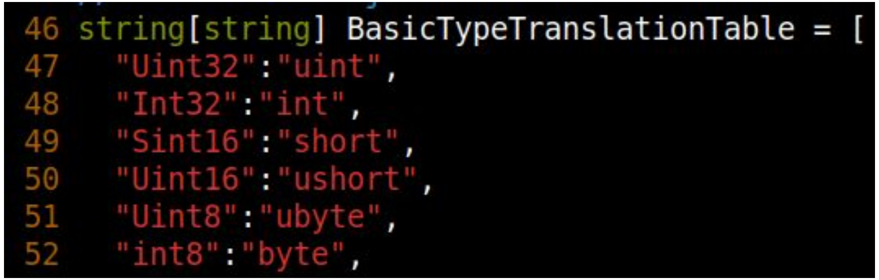
```
import std.stdio, std.string, std.conv, std.file;
import std.net.curl;
import std.process;

enum SymbolType{Function,TypeDef, Struct,Enum, UNKNOWN}

string[SymbolType] Terminators = [
    SymbolType.Function:";",
    SymbolType.TypeDef:"};",
    SymbolType.Struct:"};",
    SymbolType.Enum:"};",
];

struct Symbol{
    string type;
    string name;
}

struct Function{
    string fullstring;
    string returnType;
    string functionName;
  string[] arguments;
  string[] types;
}

struct Struct{
    string fullstring;
    Symbol[] Fields;
}

struct TypeDef{
    string fullstring;
  string typename;
}
```

Script 7 - SDL3 C Binding Generator (2/2)

- The resulting file is something like what you get on the top
  - The .html page for each function is also downloaded, and the ‘synopsis’ of the command will eventually be inlined above the function name in a comment.
- Some of the common things (e.g. translating types like ‘unsigned int’ to ‘uint’ are then taken care of.
  - Custom types a work in progress
  - The ‘BasicTypeTranslationTable’ otherwise is added to as new types are found, and an equivalent typedef.



```
// Attempting to write binding for symbol: SDL_abs
int SDL_abs(int x);
// Attempting to write binding for symbol: SDL_acos
double SDL_acos(double x);
// Attempting to write binding for symbol: SDL_acosf
float SDL_acosf(float x);
// Attempting to write binding for symbol: SDL_AcquireCameraFrame
SDL_Surface * SDL_AcquireCameraFrame(SDL_Camera *camera, Uint64 *timestampNS);
// Attempting to write binding for symbol: SDL_AcquireGPUCommandBuffer
SDL_GPUCommandBuffer * SDL_AcquireGPUCommandBuffer(
    SDL_GPUDevice *device);
// Attempting to write binding for symbol: SDL_AcquireGPUSwapchainTexture
bool SDL_AcquireGPUSwapchainTexture(
    SDL_GPUCommandBuffer *command_buffer,
    SDL_Window *window,
    SDL_GPUTexture **swapchain_texture,
    Uint32 *swapchain_texture_width,
    Uint32 *swapchain_texture_height);
// Attempting to write binding for symbol: SDL_AddAtomicInt
int SDL_AddAtomicInt(SDL_AtomicInt *a, int v);
```



```
string[string] BasicTypeTranslationTable = [
    "Uint32":"uint",
    "Int32":"int",
    "Sint16":"short",
    "Uint16":"ushort",
    "Uint8":"ubyte",
    "int8":"byte",
```

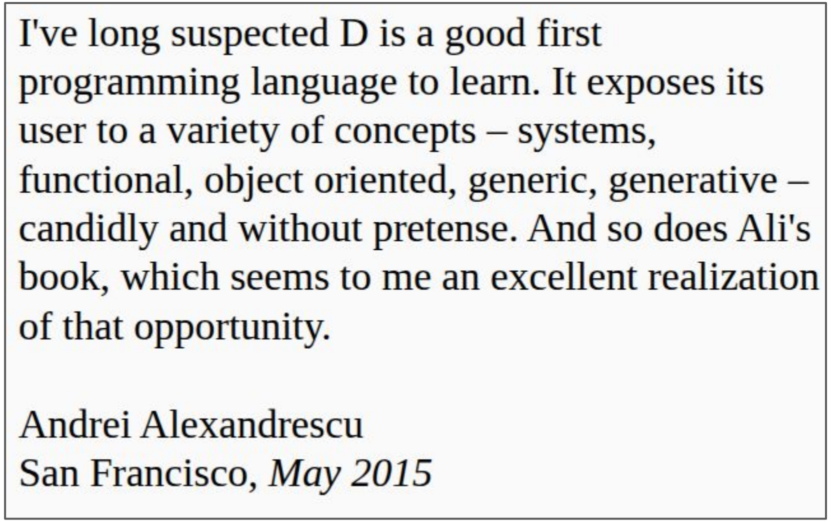
What’s next for D and Scripting?

- Continue moving forward -- perhaps these scripts will get a web interface when other folks need them.

Teaching with DLang: Part 3

Teaching D Lang at University

- The **purpose** of this section is to simply tell others **there is a lot of life in university with the D language.**
  - And I think there should be even more teaching with D -- it’s a language that scales from beginner to expert.



I've long suspected D is a good first programming language to learn. It exposes its user to a variety of concepts – systems, functional, object oriented, generic, generative – candidly and without pretense. And so does Ali's book, which seems to me an excellent realization of that opportunity.

Andrei Alexandrescu
San Francisco, *May 2015*

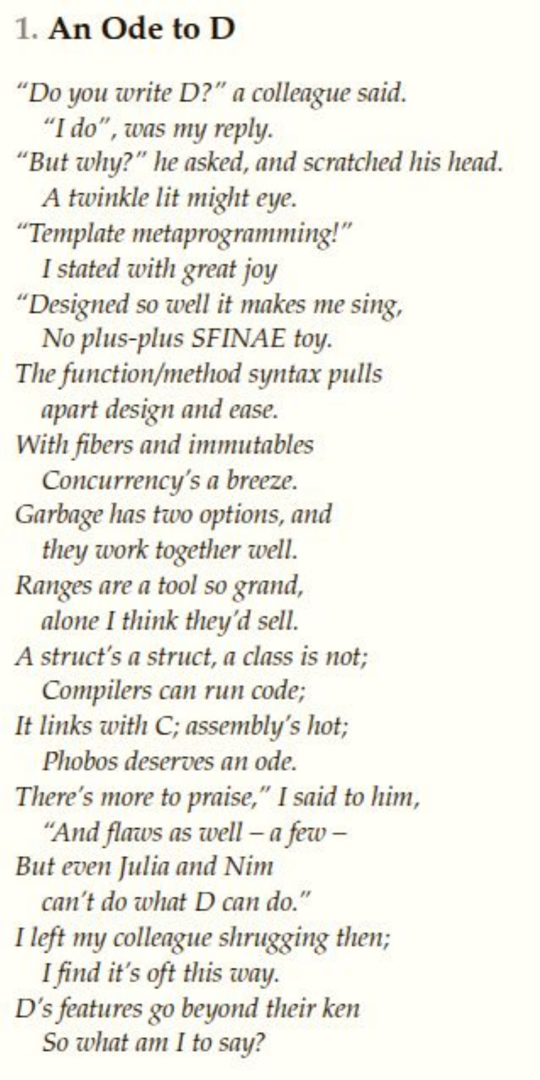
From Ali Çehreli *Programming in D* book
http://ddili.org/ders/d.en/index.html

Quick List of Academics (that I know of) using D

- Ben Jones - University of Utah
  - coming up later this week!
- Razvan Nitu, Eduard Staniloiu, et al.
  - https://github.com/Dlang-UPB/D-Summer-School
- Brian Callahan - Rensselaer Polytechnic Institute
  - https://briancallahan.net/
- Chuck Allison - Utah Valley University / Parkland
  - (Previous host of previous DConf conference!)
  - D:#A#Programming#Language#for#Our#Time
- Timon Gehr - Research at ETH Zürich
- Gas Dynamics toolkit from The University of Queensland
  - https://gdtk.uqcloud.net/docs/introduction/about-the-toolkit/
- RSUH
  - https://dlang.org/blog/2022/02/19/how-i-taught-the-d-programming-language-at-a-russian-university/
- Zach Yedidia at DConf 2023 at Stanford
- **And many more...**

(Aside) from Luther Tychonievich

- Poem to the D language
  - https://luthert.web.illinois.edu/blog/posts/730.html
  - Faculty at University of Illinois Urbana-Champaign
- For you to read later tonight before bedtime :)



1. An Ode to D

"Do you write D?" a colleague said.
"I do", was my reply.
"But why?" he asked, and scratched his head.
A twinkle lit might eye.
"Template metaprogramming!"
I stated with great joy
"Designed so well it makes me sing,
No plus-plus SFINAE toy.
The function/method syntax pulls
apart design and ease.
With fibers and immutables
Concurrency's a breeze.
Garbage has two options, and
they work together well.
Ranges are a tool so grand,
alone I think they'd sell.
A struct's a struct, a class is not;
Compilers can run code;
It links with C; assembly's hot;
Phobos deserves an ode.
There's more to praise," I said to him,
"And flaws as well – a few –
But even Julia and Nim
can't do what D can do."
I left my colleague shrugging then;
I find it's oft this way.
D's features go beyond their ken
So what am I to say?

Why is D better for teaching?

- I’ll give you the short version I gave at the New Jersey Programming Language Symposium hosted by Jane Street this past **May of 2025**
- Let’s have a look -- **next slide**
  - Note: Another talk was also given at the Illinois Computer Science Teaching Workshop in **June of 2025**
  - Note: Slides available on my website for both talks: https://mshah.io

2025 NJPLS - Revisiting the D Language - My 2^x Programming Language[Dlang Episode 133]

787 views • 1 month ago

Mike Shah

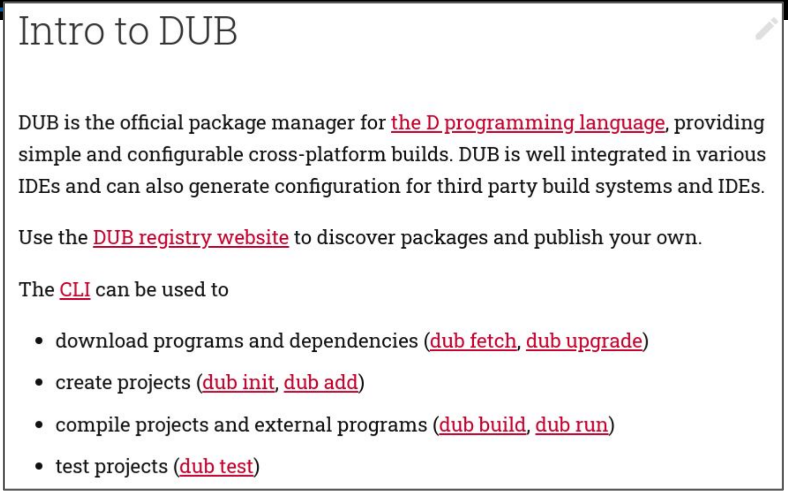
▻Lesson Description: This is a rerecording of a recent talk I gave at the New Jersey Programming Language Seminar on the D ...

14 chapters Introduction | The only reason to switch programming languages | First impression of D | History of D |...

I re-recorded the talk here otherwise: https://www.youtube.com/watch?v=PJf0etigg7o

Exponential Gains in Teaching - How? (1/2)

- **‘dub’** is the built-in package manager and build system
- Having a package manager / build system is just necessary today
  - I do still show students how to compile on the command-line however!
- (Yes, I learned about **Greenspun’s rule** recently)



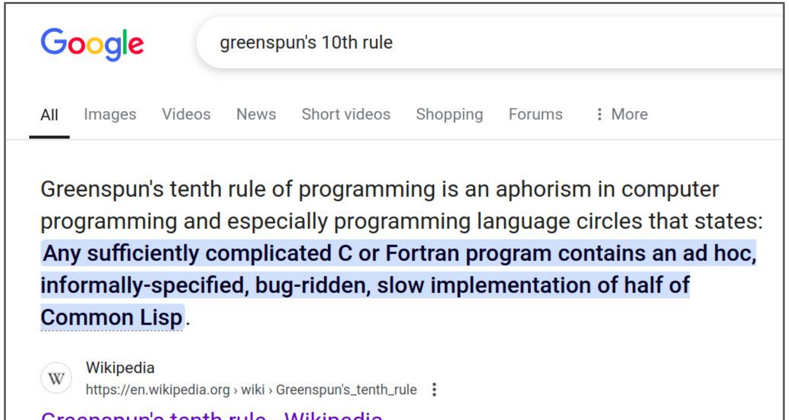
Intro to DUB

DUB is the official package manager for the D programming language, providing simple and configurable cross-platform builds. DUB is well integrated in various IDEs and can also generate configuration for third party build systems and IDEs.

Use the DUB registry website to discover packages and publish your own.

The CLI can be used to

- download programs and dependencies (dub fetch, dub upgrade)
- create projects (dub init, dub add)
- compile projects and external programs (dub build, dub run)
- test projects (dub test)



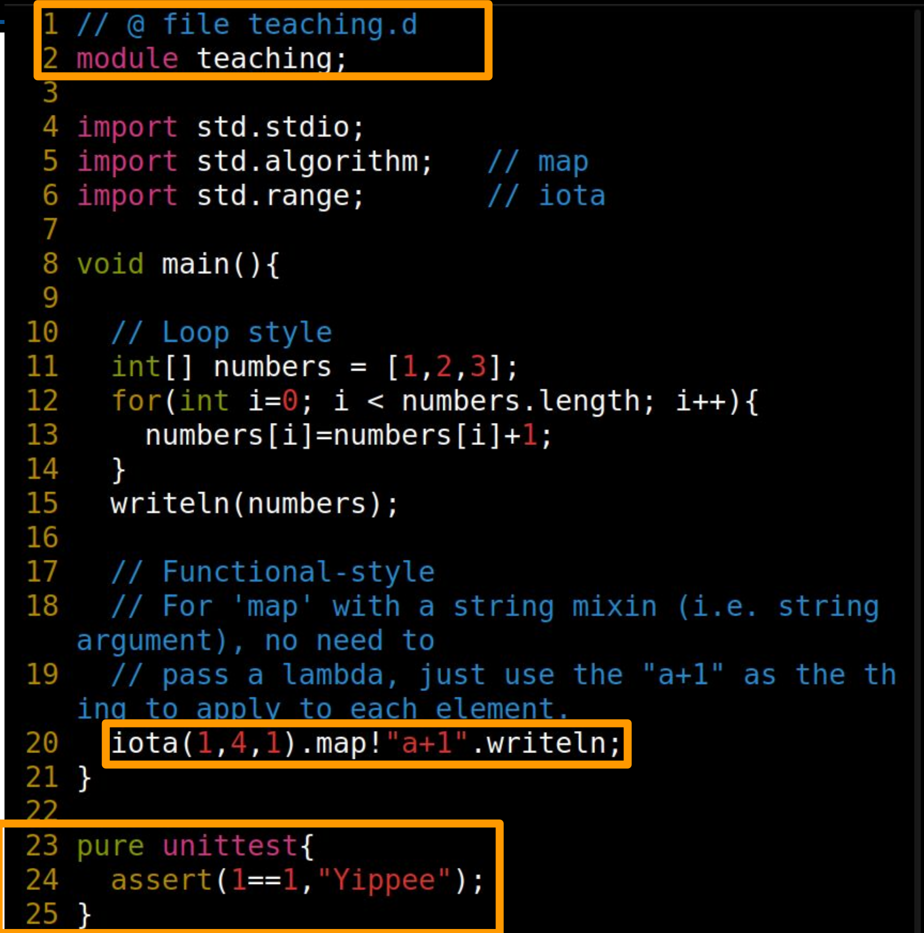
Greenspun's tenth rule of programming is an aphorism in computer programming and especially programming language circles that states: Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp.

Wikipedia
https://en.wikipedia.org › wiki › Greenspun's_tenth_rule

Greenspun's tenth rule - Wikipedia

Exponential Gains in Teaching - How? (2/2)

- **Modules instead of header files** is a big when for both iteration, and management.
- **Multiple paradigms**
  - I get to talk about things like concurrency, OOP -- specifically message passing, functional programming, generic programming, etc.
- **unit testing** built-in
  - Should show unit tests for doing test-driven development
    - (note: Tests can be annotated with ‘pure’)
- And much more!



```d
// @ file teaching.d
module teaching;

import std.stdio;
import std.algorithm;    // map
import std.range;        // iota

void main(){

  // Loop style
  int[] numbers = [1,2,3];
  for(int i=0; i < numbers.length; i++){
    numbers[i]=numbers[i]+1;
  }
  writeln(numbers);

  // Functional-style
  // For 'map' with a string mixin (i.e. string argument), no need to
  // pass a lambda, just use the "a+1" as the thing to apply to each element.
  iota(1,4,1).map!"a+1".writeln;
}

pure unittest{
  assert(1==1,"Yippee");
}
```

Courses where I changed languages to use D

- Spring 2023
    - **Software Engineering,** C++ --> D
- Fall 2024
    - **Building Game Engines**, C++ --> D
- Spring 2025
    - **Real-Time Computer Graphics,** C++ --> D
- Fall 2025
    - **Building Game Engines,** now fully in D
    - **Computer Systems,** C --> A mix of C, D, and Rust
        - (D has the ‘importC’ compiler which you can use as a C compiler)
- Note:
    - Most courses that have a final project I allow students to choose their language
    - Almost all choose D (otherwise some choose C++)

(Old news) How Generally Students Respond?

- Generally most students are open to learning a new language
    - Some are disappointed to not be learning C++ in my graphics/games courses initially
    - Most by the end of the semester report being happy.
- Almost all students who previously used C++ previously reported enjoying using and collaborating on group projects in D more than C++.
- **Even better** -- you can hear directly the students perspective
- D Conf 2023:
    - YouTube: https://www.youtube.com/live/wXTlafzlJVY?si=Xpy6g5h4wtIUrt2E&t=7711
    - Link to Conference Talk Description: https://dconf.org/2023/index.html

DCONF '23
August 29 - September 1

A Semester at University: Teaching Software Engineering in DLang

Social: @MichaelShah
Web: mshah.io
Courses: courses.mshah.io
YouTube: www.youtube.com/c/MikeShah

Presentor: Mike Shah, Ph.D.
11:00-11:30, Wed, Aug. 30, 2023
Audience: Everyone!

What’s next for D and teaching?

- It is time for another D programming book to start getting drafted.
- This is actively on my mind directly after I finish another book project in the works (which will have a chapter on D) -- give me a little bit more time on this :)

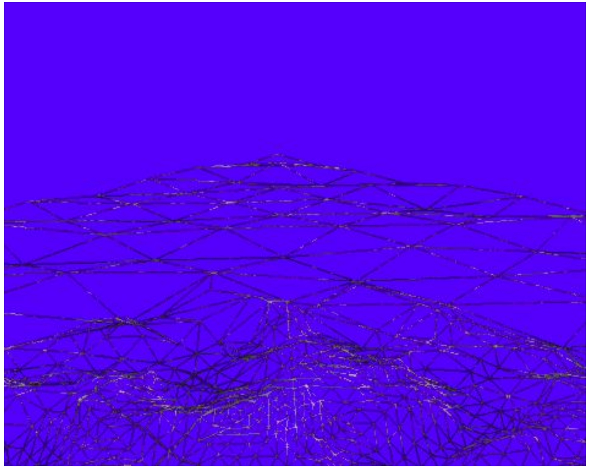
Research DLang: Part 4

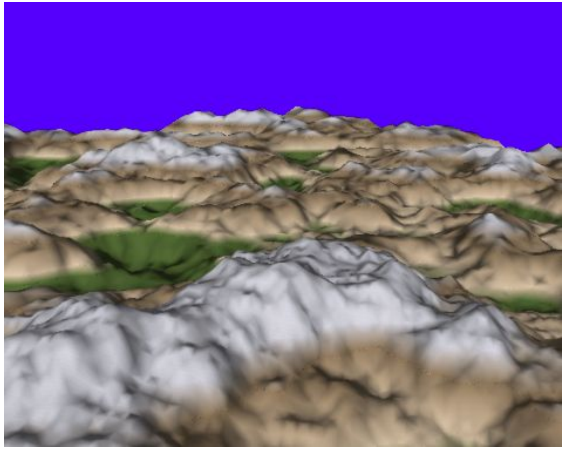
2025 Bachelor of Science Senior Thesis Projects using D

- Matthew Q. - Decoupling Semantic Routines from AST Nodes in the DMD Compiler for D
  - **Special shout out to Razvan Nitu for leading this student**
- Ali U. - Real-Time Terrain Tessellation: A Comparative Study of LOD Approaches
- Arnav N. - Boosting Performance: A D-Based Alternative to C++ Boost Algorithms
- Ron C. - Lightweight Pseudo-3D Rendering with Sprite Stacks and Bump Maps
- Simon J. - Level of Detail for 3D Procedural Terrain Generators

These students each have 3 semesters worth of D programming experience leaving school (amongst other internships and programming language experience) -- keep an eye out if you are hiring!

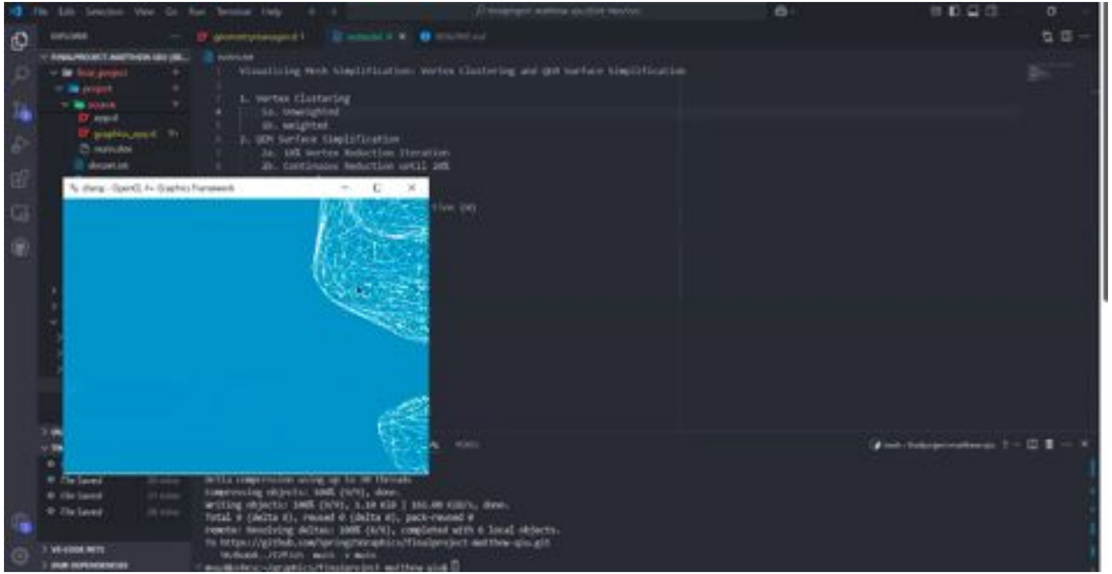
Some Screenshots of some student research work (in D)





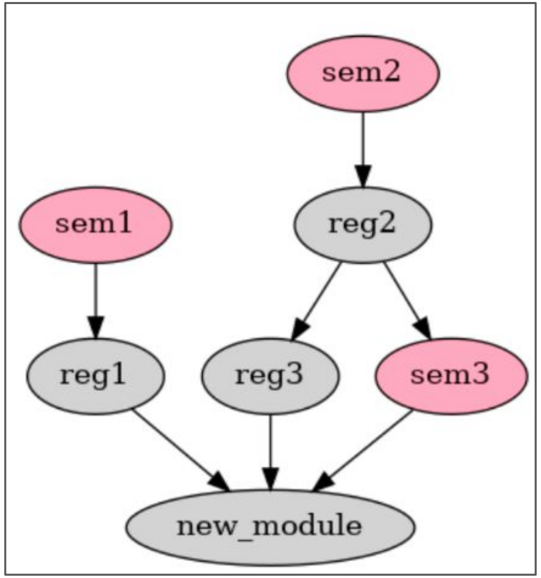
- Simon J. - Water, fresnel equations, level of detail algorithms, noise generation algorithms

Some Screenshots of some student research work (in D)



```
Visualizing Mesh Simplification: Vertex Clustering and QEM Surface Simplification

1. Vertex Clustering
    1a. Unweighted
    1b. Weighted
2. QEM Surface Simplification
    2a. 10% Vertex Reduction Iteration
    2b. Continuous Reduction until 10%
```

```
Delta compression using up to [illegible] threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.18 KiB | 151.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/[illegible]/finalproject-matthew-qiu.git
   [illegible]  main -> main
```



- Matthew Q. - Mesh simplification, and work with Razvan on [Decoupling Semantic Routines from AST Nodes in the DMD Compiler for D]

What’s next for D and Research?

- More students will be completing their senior thesis with me every year.
    - If your company would like to collaborate, I like having students work on real stuff as part of their projects (or doing internships) within the bounds of students completing things for their degree.
- I will continue to march along with research using D.

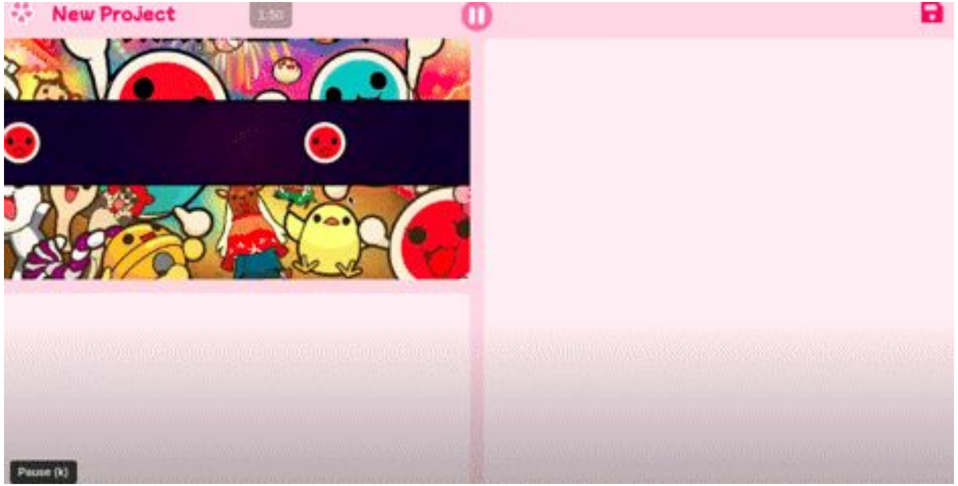
Game Engines with DLang: Part 5

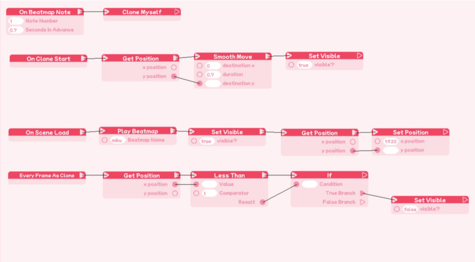
Building Game Engines

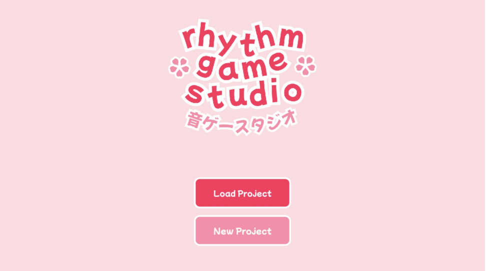
- As mentioned, I am teaching a course on building games and game engines primarily in D
  - (Students explore some C libraries, and also incorporating scripting languages into their engines as well)
- I will show a few samples of the outcomes of the type of things students build in a semester without any prior D programming experience.
  - **Next slide**

Yoonity Engine

- Members
  - Eric Y.
- Video
  - https://www.youtube.com/watch?v=XJGRJ9V-M4w
- Notes
  - https://ericyoondotcom.github.io/RhythmGameStudio/pages/

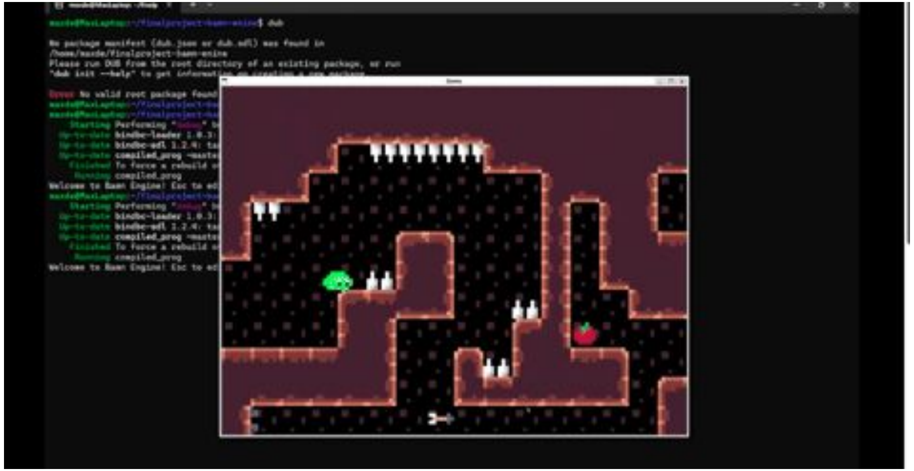




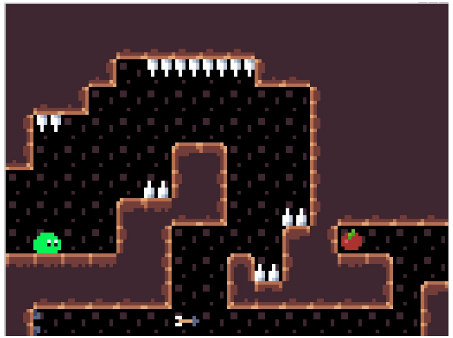


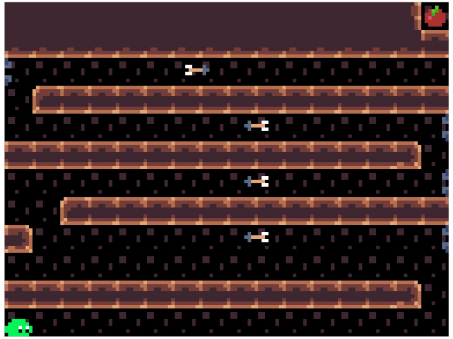
Bamn-engine

- Members
  - Brian B.
  - Andrew F.
  - Max O.
  - Nicholas S.
- Video
  - https://www.youtube.com/watch?v=7RJcTSc5v3I
- Notes
  - https://andrewfu1.github.io/bamn-website/



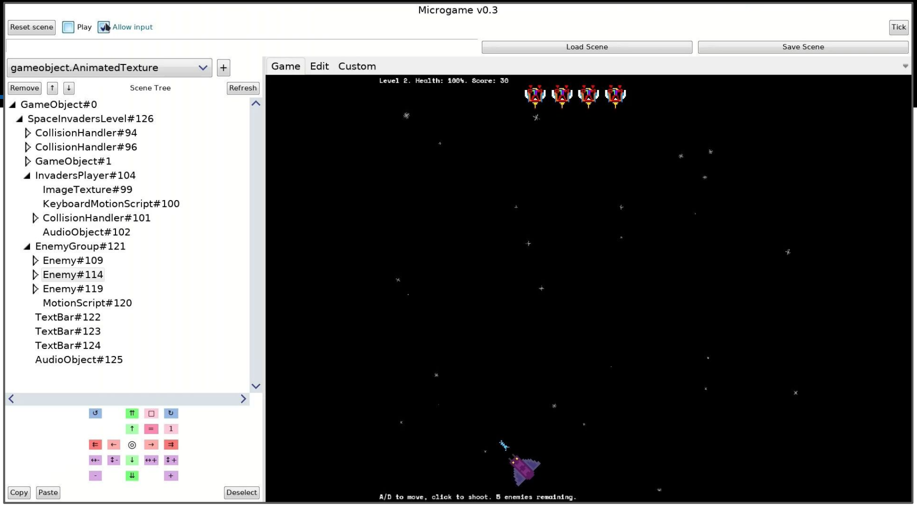
```
maxde@MaxLaptop:~/finalproject-bamn-enine$ dub

No package manifest (dub.json or dub.sdl) was found in
/home/maxde/finalproject-bamn-enine
Please run DUB from the root directory of an existing package, or run
"dub init --help" to get information on creating a new package.

Error No valid root package found
maxde@MaxLaptop:~/finalproject-ba
maxde@MaxLaptop:~/finalproject-ba
    Starting Performing "debug" b
  Up-to-date bindbc-loader 1.0.3:
  Up-to-date bindbc-sdl 1.2.4: ta
  Up-to-date compiled_prog ~maste
    Finished To force a rebuild o
     Running compiled_prog
Welcome to Bamn Engine! Esc to ed
maxde@MaxLaptop:~/finalproject-ba
    Starting Performing "debug" b
  Up-to-date bindbc-loader 1.0.3:
  Up-to-date bindbc-sdl 1.2.4: ta
  Up-to-date compiled_prog ~maste
    Finished To force a rebuild o
     Running compiled_prog
Welcome to Bamn Engine! Esc to ed
```





Micro Engine

- Members
  - Alexa S.
- Video
  - https://www.youtube.com/watch?v=TCuYpCA5oGk
- Notes
  - Full scene graph serialization, demonstration of multiple game types, live editing, forked version of DLangGUI, very nice full editor for building out the game, uses JS-like scripting at run-time



```
◢ GameObject#0
  ◢ SpaceInvadersLevel#126
    ▷ CollisionHandler#94
    ▷ CollisionHandler#96
    ▷ GameObject#1
    ◢ InvadersPlayer#104
        ImageTexture#99
        KeyboardMotionScript#100
      ▷ CollisionHandler#101
        AudioObject#102
    ◢ EnemyGroup#121
      ▷ Enemy#109
      ▷ Enemy#114
      ▷ Enemy#119
        MotionScript#120
      TextBar#122
      TextBar#123
      TextBar#124
      AudioObject#125
```

Level 2. Health: 100%. Score: 30

A/D to move, click to shoot. 5 enemies remaining.

Graphics Engines with DLang: Part 6

Graphics Programming with D

- In Spring of 2025 I also made the leap to teach **Real-Time Computer Graphics** programming with D and OpenGL 4.1
  - This means students get a full year of D programming
    - i.e. Most who take ‘Building Game Engines’ join me for Real-Time Graphics
- I’ve made the argument for why I found graphics programming in D more enjoyable previously -- so now I want to show you some results from teaching and my experience.

accu conference 2025

The Case for Graphics Programming Using the D Language

Mike Shah

14:00 - 15:30 Tue, April 1, 2025

90 minutes | Intermediate Audience

Social: mikeshah.bsky.social
Web: mshah.io
Courses: courses.mshah.io
YouTube
www.youtube.com/c/MikeShah
http://tinyurl.com/mike-talks

Mike Shah

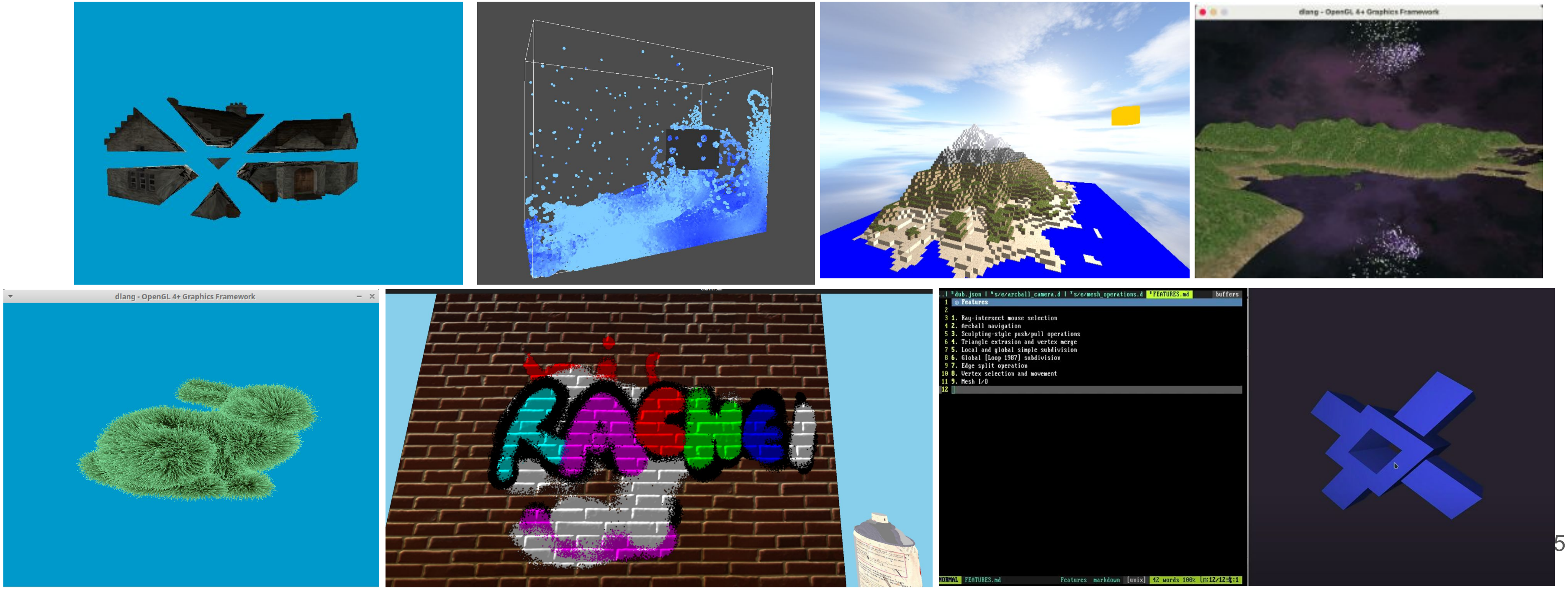
The Case for Graphics Programming Using the D Language - Mike Shah - ACCU 2025
https://www.youtube.com/watch?v=RS3qzDDFMOM

(This includes all the content and more from my DConf 24’ online and DConf 24’ tlak)

Teaching Graphics Programming with D (still image)

- Students did amazing work using D -- here's a sample of things they built in 1-2 weeks at the end of the course

left-to-right starting at the top constructive solid geometry, water simulation, minecraft, fireworks simulation, texture painter, full 3D modeling tool

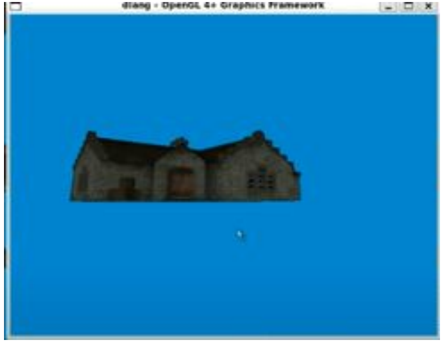

dlang - OpenGL 4+ Graphics Framework
dlang - OpenGL 4+ Graphics Framework

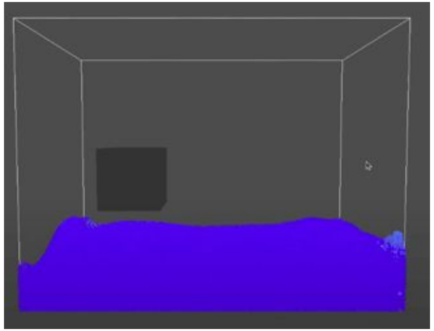
```
..| 5dub.json | 6s/e/arcball_camera.d | 7s/e/mesh_operations.d  8FEATURES.md        buffers
 ⓪ Features

 1. Ray-intersect mouse selection
 2. Arcball navigation
 3. Sculpting-style push/pull operations
 4. Triangle extrusion and vertex merge
 5. Local and global simple subdivision
 6. Global [Loop 1987] subdivision
 7. Edge split operation
 8. Vertex selection and movement
 9. Mesh I/O

NORMAL  FEATURES.md            Features  markdown  [unix]  42 words 100%  ln:12/12☰:1
```

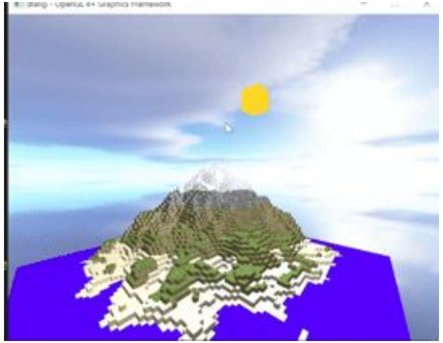
Teaching Graphics Programming with D (animated)

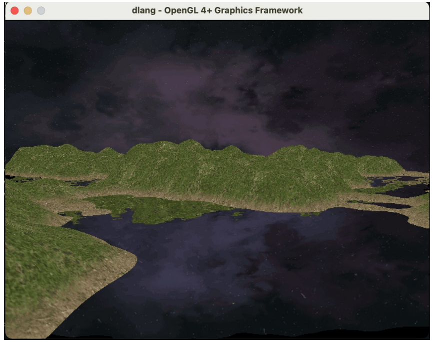
- Students did amazing work using D -- here's a sample of things they built in 1-2 weeks at the end of the course

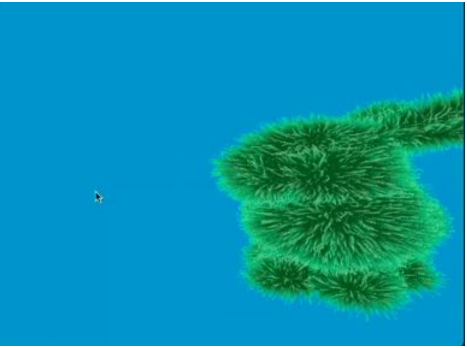
left-to-right starting at the top constructive solid geometry, water simulation, minecraft, fireworks simulation, texture painter, full 3D modeling tool

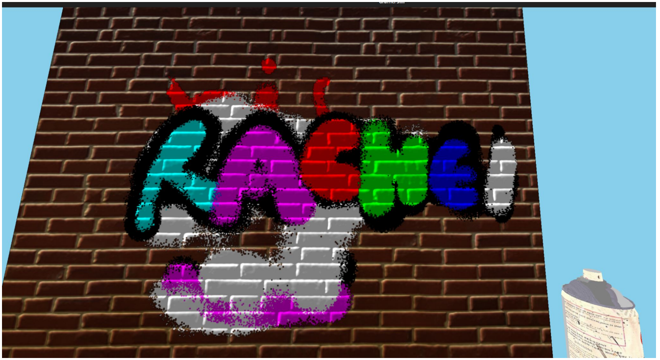


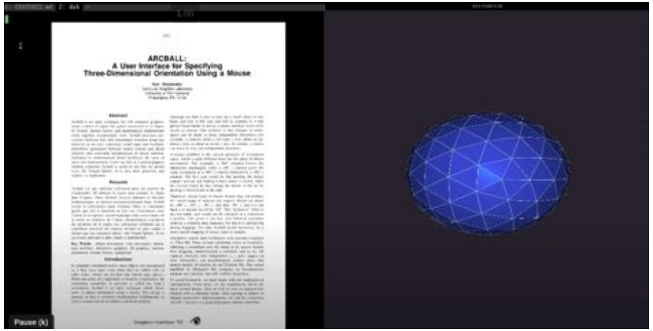






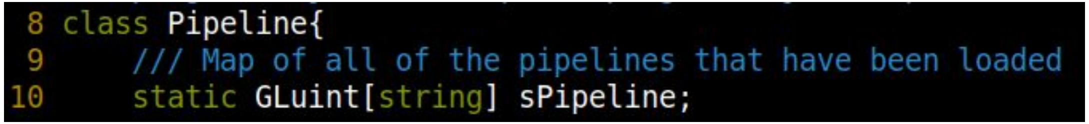




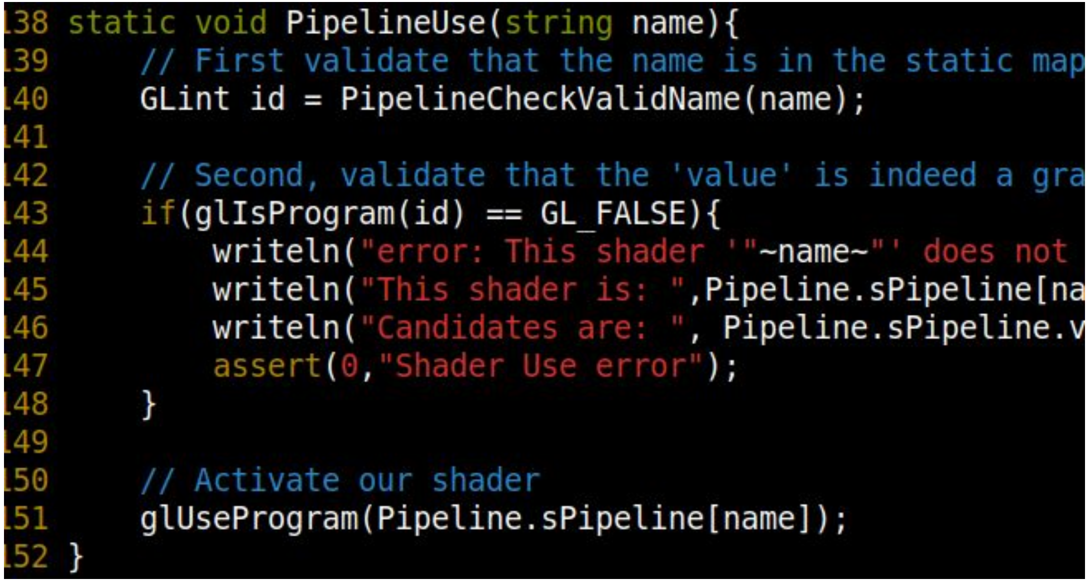

ARCBALL:
A User Interface for Specifying
Three-Dimensional Orientation Using a Mouse
Pause (k)

Some notes on Graphics Programming (1/8)

- From my own notes on graphics programming --
- D has made it relatively easy to find good abstractions for graphics programming.
- I'll share a few snippets that may be useful for folks later on.



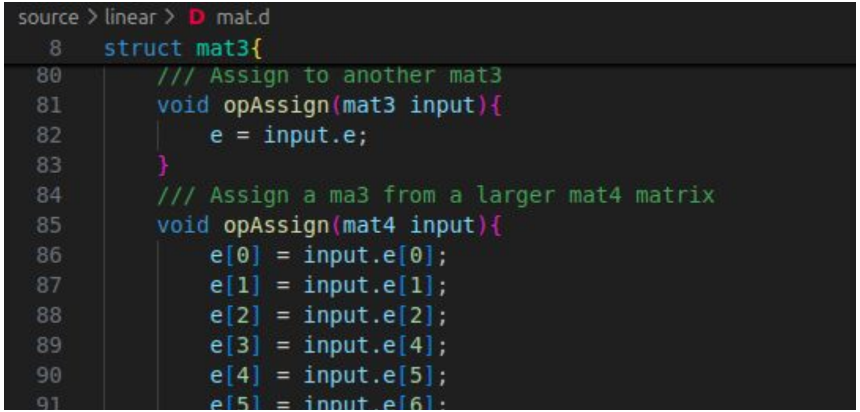
```
class Pipeline{
    /// Map of all of the pipelines that have been loaded
    static GLuint[string] sPipeline;
```



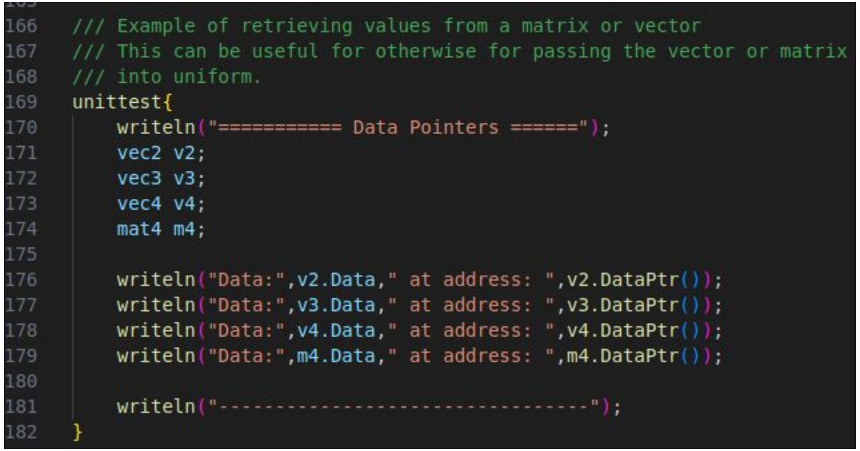
```
static void PipelineUse(string name){
    // First validate that the name is in the static map
    GLint id = PipelineCheckValidName(name);

    // Second, validate that the 'value' is indeed a gra
    if(glIsProgram(id) == GL_FALSE){
        writeln("error: This shader '"~name~"' does not
        writeln("This shader is: ",Pipeline.sPipeline[na
        writeln("Candidates are: ", Pipeline.sPipeline.v
        assert(0,"Shader Use error");
    }

    // Activate our shader
    glUseProgram(Pipeline.sPipeline[name]);
}
```

Some notes on Graphics Programming (2/8)

- There exist several linear algebra libraries, though writing your own is not too hard.
- I was able to replace glm relatively easy, and it just *feels* more understandable.
    - I think operator overloading being easy to ‘grep’ is part of that.
    - unittest (along with code coverage) also a favorite feature I use for:
        - correctness
        - documentation



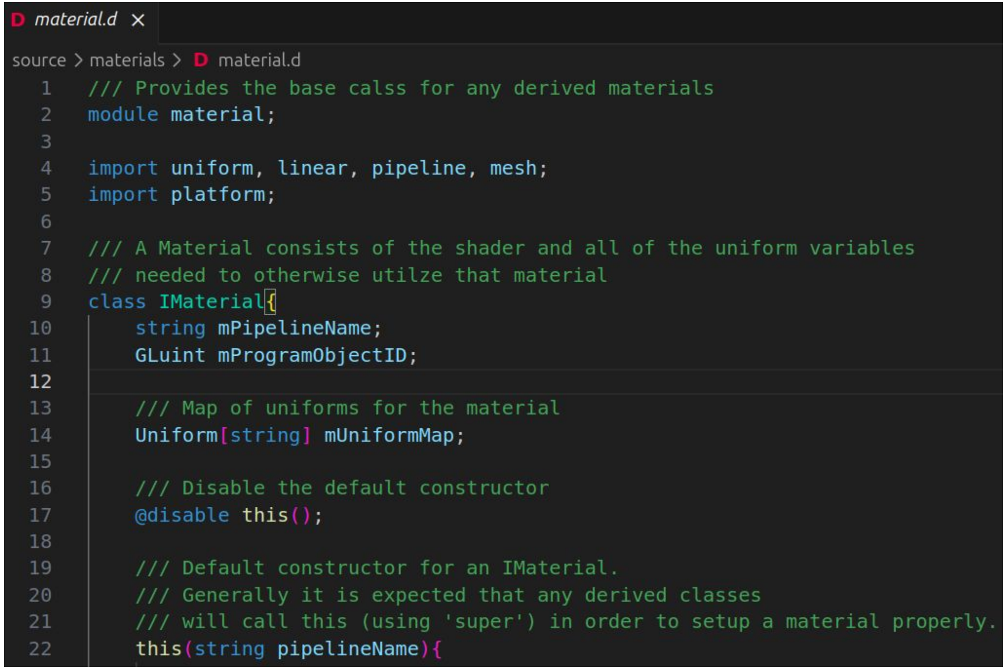
```
struct mat3{
    /// Assign to another mat3
    void opAssign(mat3 input){
        e = input.e;
    }
    /// Assign a ma3 from a larger mat4 matrix
    void opAssign(mat4 input){
        e[0] = input.e[0];
        e[1] = input.e[1];
        e[2] = input.e[2];
        e[3] = input.e[4];
        e[4] = input.e[5];
        e[5] = input.e[6];
```



```
/// Example of retrieving values from a matrix or vector
/// This can be useful for otherwise for passing the vector or matrix
/// into uniform.
unittest{
    writeln("=========== Data Pointers ======");
    vec2 v2;
    vec3 v3;
    vec4 v4;
    mat4 m4;

    writeln("Data:",v2.Data," at address: ",v2.DataPtr());
    writeln("Data:",v3.Data," at address: ",v3.DataPtr());
    writeln("Data:",v4.Data," at address: ",v4.DataPtr());
    writeln("Data:",m4.Data," at address: ",m4.DataPtr());

    writeln("---------------------------------");
}
```

Some notes on Graphics Programming (3/8)

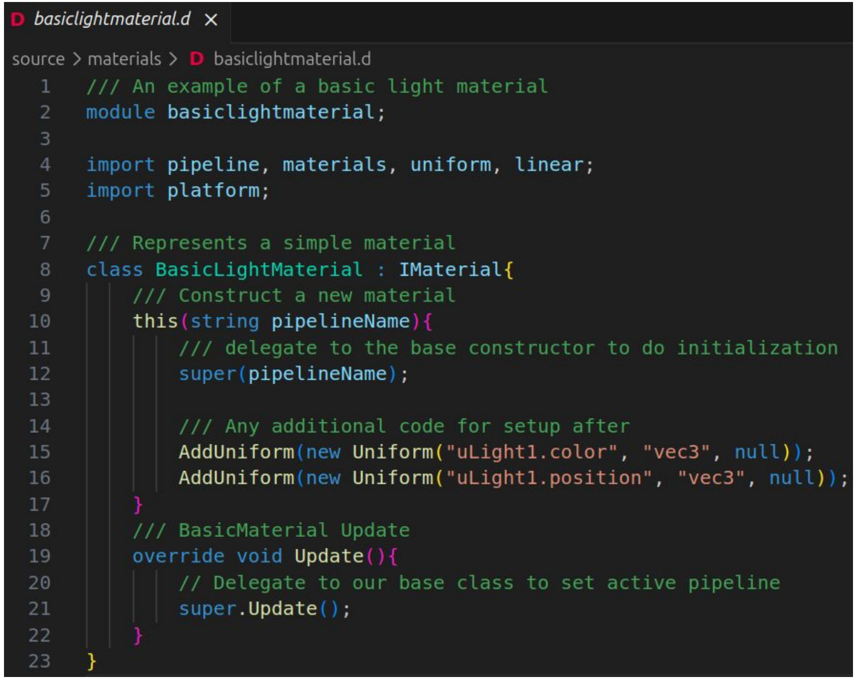
- I’ve leaned into using classes for now for some things in my graphics engine (and ultimately students build something like this themselves)
- I’ve liked using classes (reference semantics) because I care about malleability right now -- while I’m trying to solve a problem



```d
/// Provides the base calss for any derived materials
module material;

import uniform, linear, pipeline, mesh;
import platform;

/// A Material consists of the shader and all of the uniform variables
/// needed to otherwise utilze that material
class IMaterial{
    string mPipelineName;
    GLuint mProgramObjectID;

    /// Map of uniforms for the material
    Uniform[string] mUniformMap;

    /// Disable the default constructor
    @disable this();

    /// Default constructor for an IMaterial.
    /// Generally it is expected that any derived classes
    /// will call this (using 'super') in order to setup a material properly.
    this(string pipelineName){
```

Some notes on Graphics Programming (4/8)

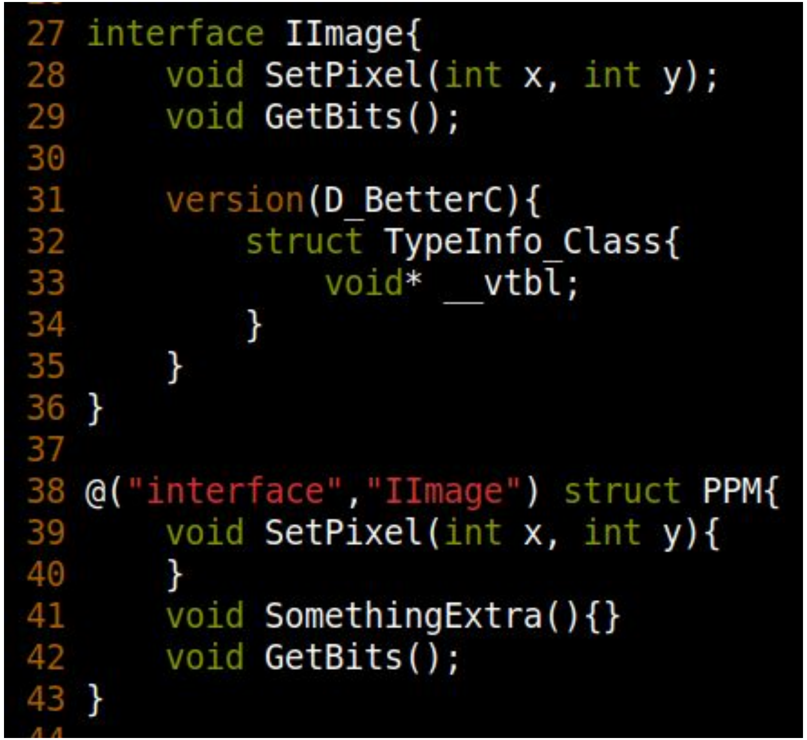
- (For completeness)
- Here’s an example of me implementing a new material
    - This automatically adds other ‘uniforms’ needed to work with the GPU.



```d
/// An example of a basic light material
module basiclightmaterial;

import pipeline, materials, uniform, linear;
import platform;

/// Represents a simple material
class BasicLightMaterial : IMaterial{
    /// Construct a new material
    this(string pipelineName){
        /// delegate to the base constructor to do initialization
        super(pipelineName);

        /// Any additional code for setup after
        AddUniform(new Uniform("uLight1.color", "vec3", null));
        AddUniform(new Uniform("uLight1.position", "vec3", null));
    }
    /// BasicMaterial Update
    override void Update(){
        // Delegate to our base class to set active pipeline
        super.Update();
    }
}
```

Some notes on Graphics Programming (5/8)

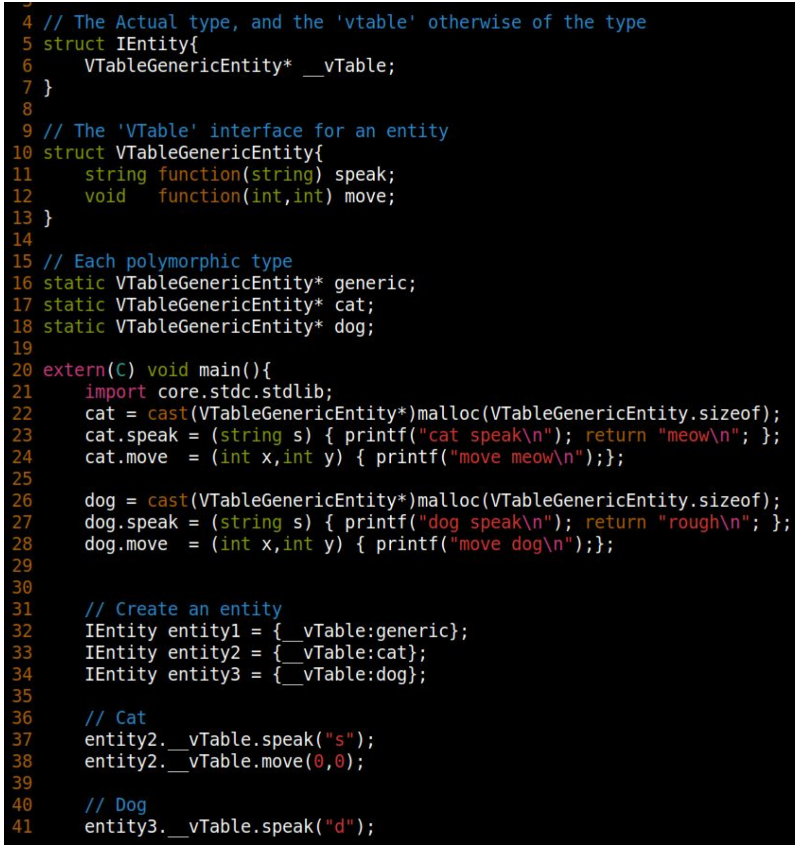
- (Don't worry)
- I wrote a little compile-time reflection thing using User Defined Attributes (UDA) for struct's that should adhere to interfaces if you really don't like classes
    - I think several others have written frameworks/dub packages for this



```
interface IImage{
    void SetPixel(int x, int y);
    void GetBits();

    version(D_BetterC){
        struct TypeInfo_Class{
            void* __vtbl;
        }
    }
}

@("interface","IImage") struct PPM{
    void SetPixel(int x, int y){
    }
    void SomethingExtra(){}
    void GetBits();
}
```

Some notes on Graphics Programming (6/8)

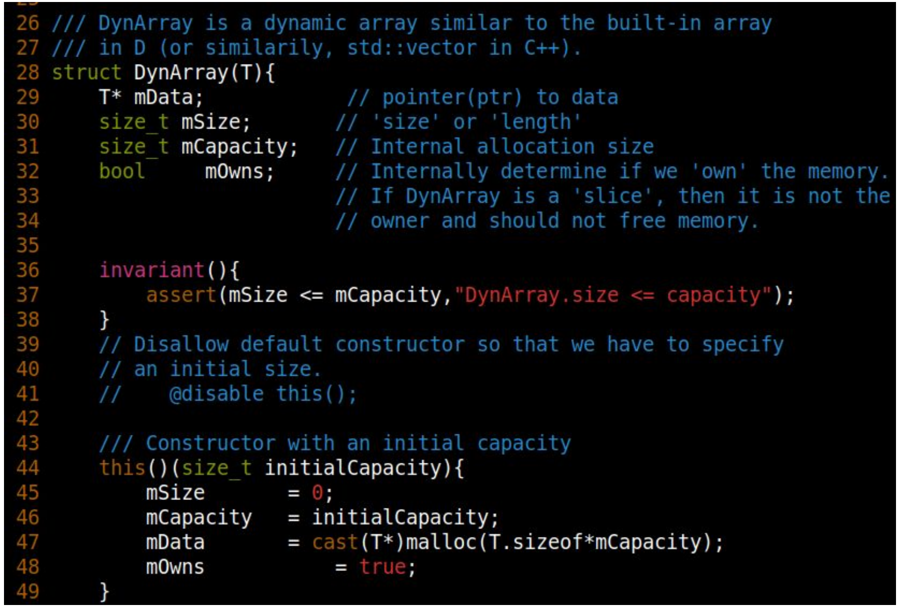
- (Don’t worry part 2)
- Inheritance can also be added in when the time comes if you want to see exactly what you are paying
  - To the right is a minimal example demonstrating the concept



```
// The Actual type, and the 'vtable' otherwise of the type
struct IEntity{
    VTableGenericEntity* __vTable;
}

// The 'VTable' interface for an entity
struct VTableGenericEntity{
    string function(string) speak;
    void   function(int,int) move;
}

// Each polymorphic type
static VTableGenericEntity* generic;
static VTableGenericEntity* cat;
static VTableGenericEntity* dog;

extern(C) void main(){
    import core.stdc.stdlib;
    cat = cast(VTableGenericEntity*)malloc(VTableGenericEntity.sizeof);
    cat.speak = (string s) { printf("cat speak\n"); return "meow\n"; };
    cat.move  = (int x,int y) { printf("move meow\n");};

    dog = cast(VTableGenericEntity*)malloc(VTableGenericEntity.sizeof);
    dog.speak = (string s) { printf("dog speak\n"); return "rough\n"; };
    dog.move  = (int x,int y) { printf("move dog\n");};


    // Create an entity
    IEntity entity1 = {__vTable:generic};
    IEntity entity2 = {__vTable:cat};
    IEntity entity3 = {__vTable:dog};

    // Cat
    entity2.__vTable.speak("s");
    entity2.__vTable.move(0,0);

    // Dog
    entity3.__vTable.speak("d");
```

Some notes on Graphics Programming (7/8)

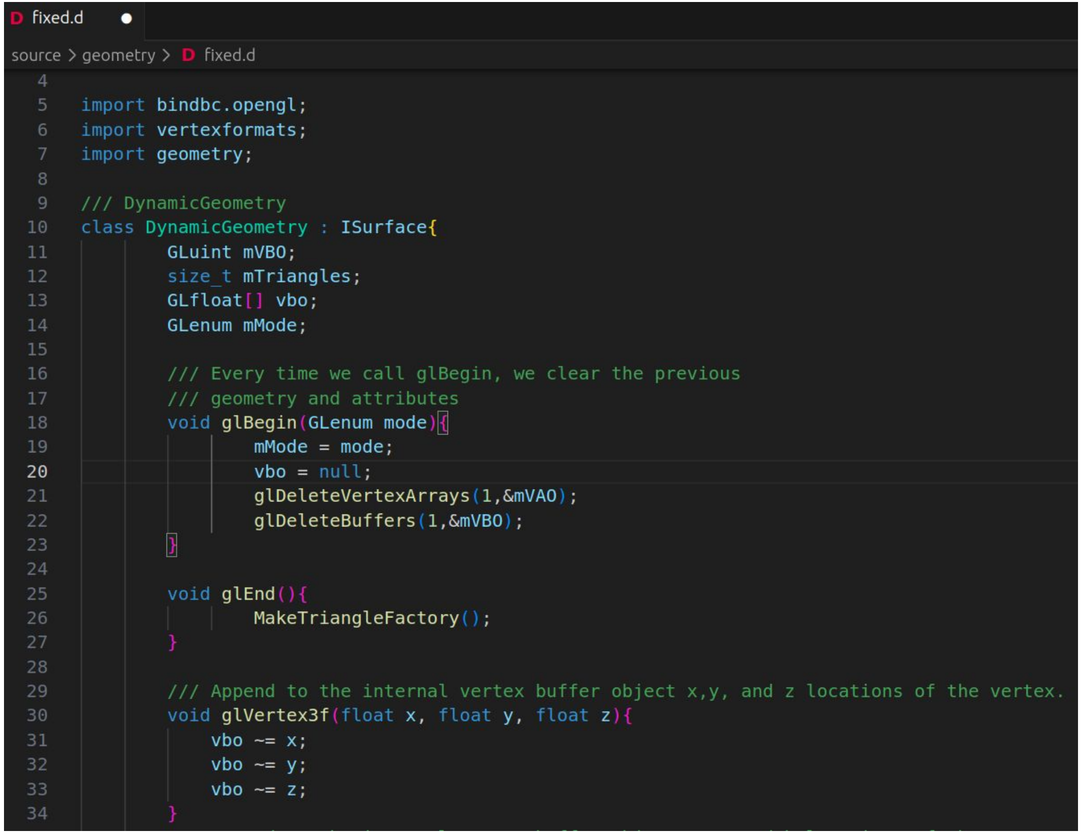
- (Don’t worry part 3)
- I wrote a replacement for built-in arrays if you want to just keep getting rid of things and use D in betterC mode.
- I don’t really use this type though -- I learned to not fear the GC
    - I just don’t allocate in my graphics loop
- When the time comes, I might redesign/revisit this.



```
/// DynArray is a dynamic array similar to the built-in array
/// in D (or similarily, std::vector in C++).
struct DynArray(T){
    T* mData;             // pointer(ptr) to data
    size_t mSize;        // 'size' or 'length'
    size_t mCapacity;    // Internal allocation size
    bool      mOwns;     // Internally determine if we 'own' the memory.
                         // If DynArray is a 'slice', then it is not the
                         // owner and should not free memory.

    invariant(){
        assert(mSize <= mCapacity,"DynArray.size <= capacity");
    }
    // Disallow default constructor so that we have to specify
    // an initial size.
    //     @disable this();

    /// Constructor with an initial capacity
    this()(size_t initialCapacity){
        mSize        = 0;
        mCapacity    = initialCapacity;
        mData        = cast(T*)malloc(T.sizeof*mCapacity);
        mOwns            = true;
    }
```

Some notes on Graphics Programming (8/8)

- When code is malleable, sometimes ‘silly’ ideas arise
- Here’s a snippet of reimplementing an ‘immediate mode’ style rendering on top of the modern API
  - Efficient? No
  - Good for prototyping or dynamic geometry? Moreso
    - Another graphics tool on the toolbelt



```
import bindbc.opengl;
import vertexformats;
import geometry;

/// DynamicGeometry
class DynamicGeometry : ISurface{
    GLuint mVBO;
    size_t mTriangles;
    GLfloat[] vbo;
    GLenum mMode;

    /// Every time we call glBegin, we clear the previous
    /// geometry and attributes
    void glBegin(GLenum mode){
        mMode = mode;
        vbo = null;
        glDeleteVertexArrays(1,&mVAO);
        glDeleteBuffers(1,&mVBO);
    }

    void glEnd(){
        MakeTriangleFactory();
    }

    /// Append to the internal vertex buffer object x,y, and z locations of the vertex.
    void glVertex3f(float x, float y, float z){
      vbo ~= x;
      vbo ~= y;
      vbo ~= z;
    }
```

Conclusion

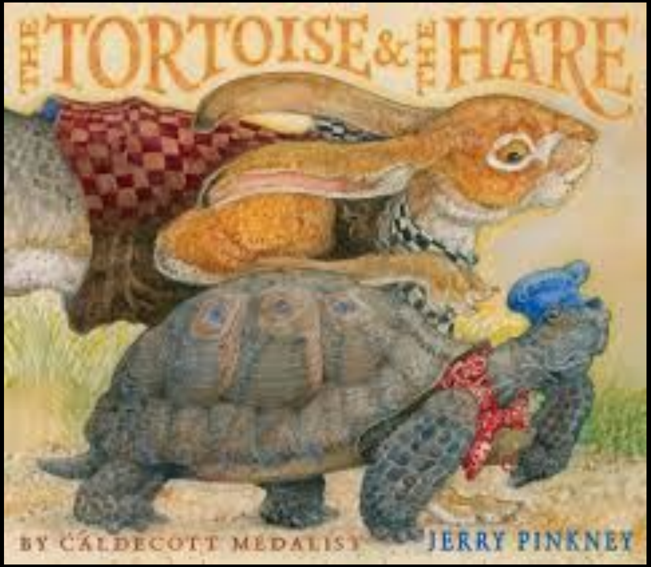
“Most people overestimate what they can do in a day or week, but underestimate what they can do in a year” - a colleague

This quote came from a discussion from an engineer I knew a decade ago while I was doing an internship.

I’ve adapted it a little bit overtime to the current phrasing....

“Most people overestimate what they can do in a day, but the far worse problem is how much they underestimate how much they can learn in a year. Knowledge compounds, slow and steady wins the race.”

“Most people overestimate what they can do in a day, but the far worse problem is how much they underestimate how much they can learn in a year. Knowledge compounds, slow and steady wins the race.”



And some of us may know this story... *Slow and steady wins the race*

I went all in on D in the last year and will continue working in the language.

D is a tool for providing a competitive advantage in many programming domains

Using D has felt more akin to a rocket ship heading for Mars while everyone else is shooting for the moon. (Seriously -- see the forums)

Thank you DConf 2025!

All in on DLang: Why I pivoted to D for web, teaching, and graphics in 2025 and beyond!

Web: mshah.io
YouTube www.youtube.com/c/MikeShah
Social: mikeshah.bsky.social
Courses: courses.mshah.io
Talks: http://tinyurl.com/mike-talks

40 minutes | Audience (For All)
10:00 - 10:40 Tues, Aug 19, 2025

Thank you!