Introduction to ZIG

Andrew Kelley

GOTO: Copenhagen 2022

Speaker notes

These are my secret speaker notes! 😱

Let's see what kind of programming experience y'all have.

Warm up the audience, let’s find out what kind of programming language experience y’all have.

Raise your hand if you have experience with...

1. Java or Go
2. Python, Perl, JavaScript, or similar
3. C, C++, or Rust
4. Other

Repeat into the mic about how many people raise their hands.

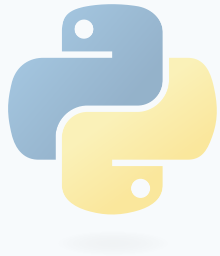
Let's see what kind of programming experience y'all have.

Let's see what kind of programming experience y'all have.

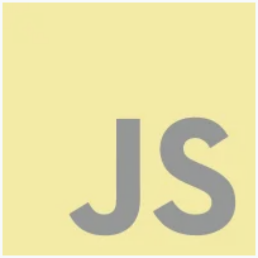
Let's see what kind of programming experience y'all have.

Java
GO
PROGRAMMING REPUBLIC OF
PERL
JS
THE
C
PROGRAMMING
LANGUAGE
C++
R

Let's see what kind of programming experience y'all have.



PROGRAMMING REPUBLIC OF
PERL



THE
C
PROGRAMMING
LANGUAGE

other

SPOT THE BUG

```
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

Speaker notes

Hook: Here is some Zig code.

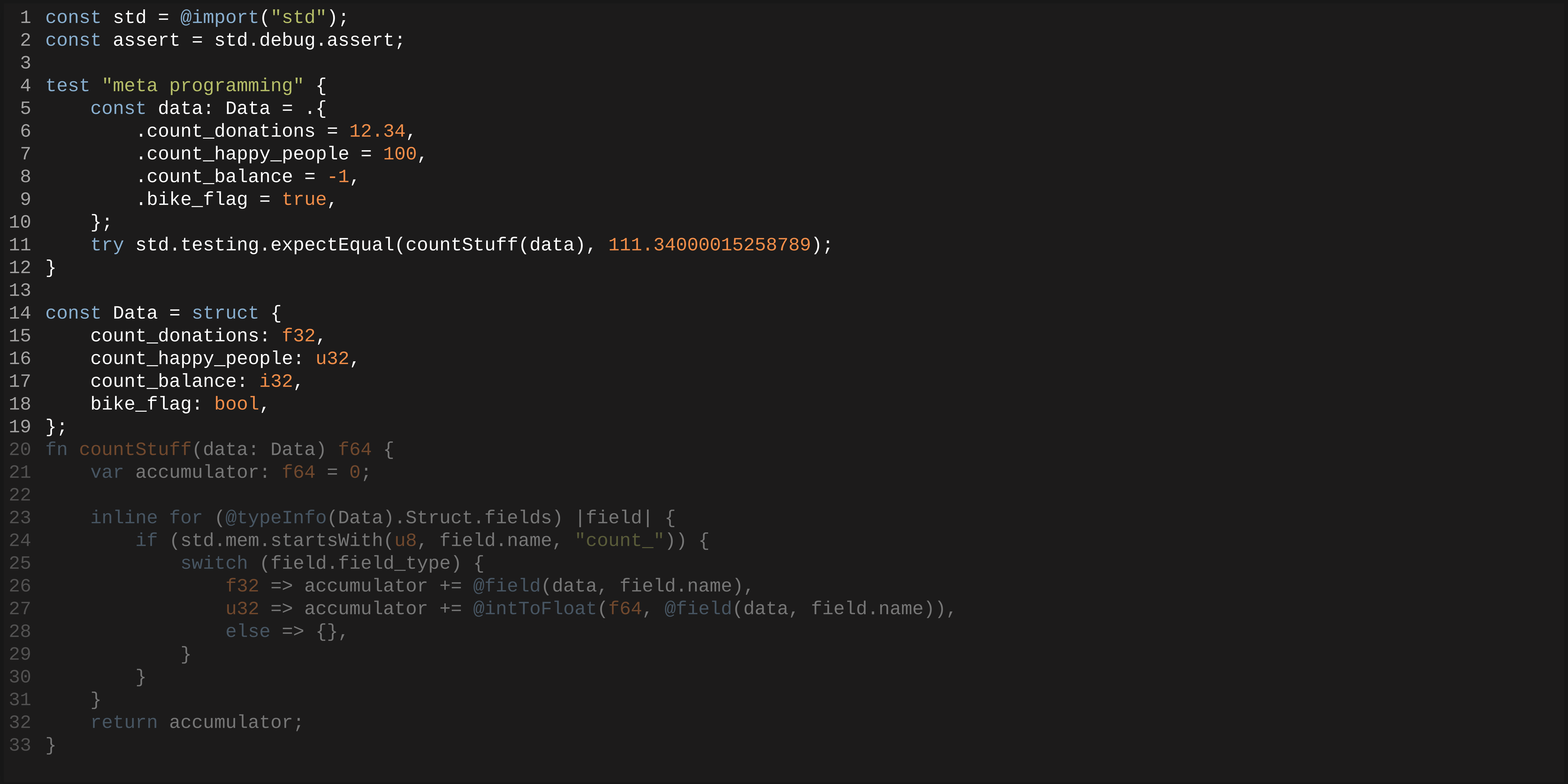
1. Raise your hand if this is your first time ever seeing Zig code.
2. Keep your hand up if you can spot the bug.

Hopefully at least one person successfully debugs code in a language they have never seen before.

"That’s the power of Zig."

Hint: off by 1

SPOT THE BUG

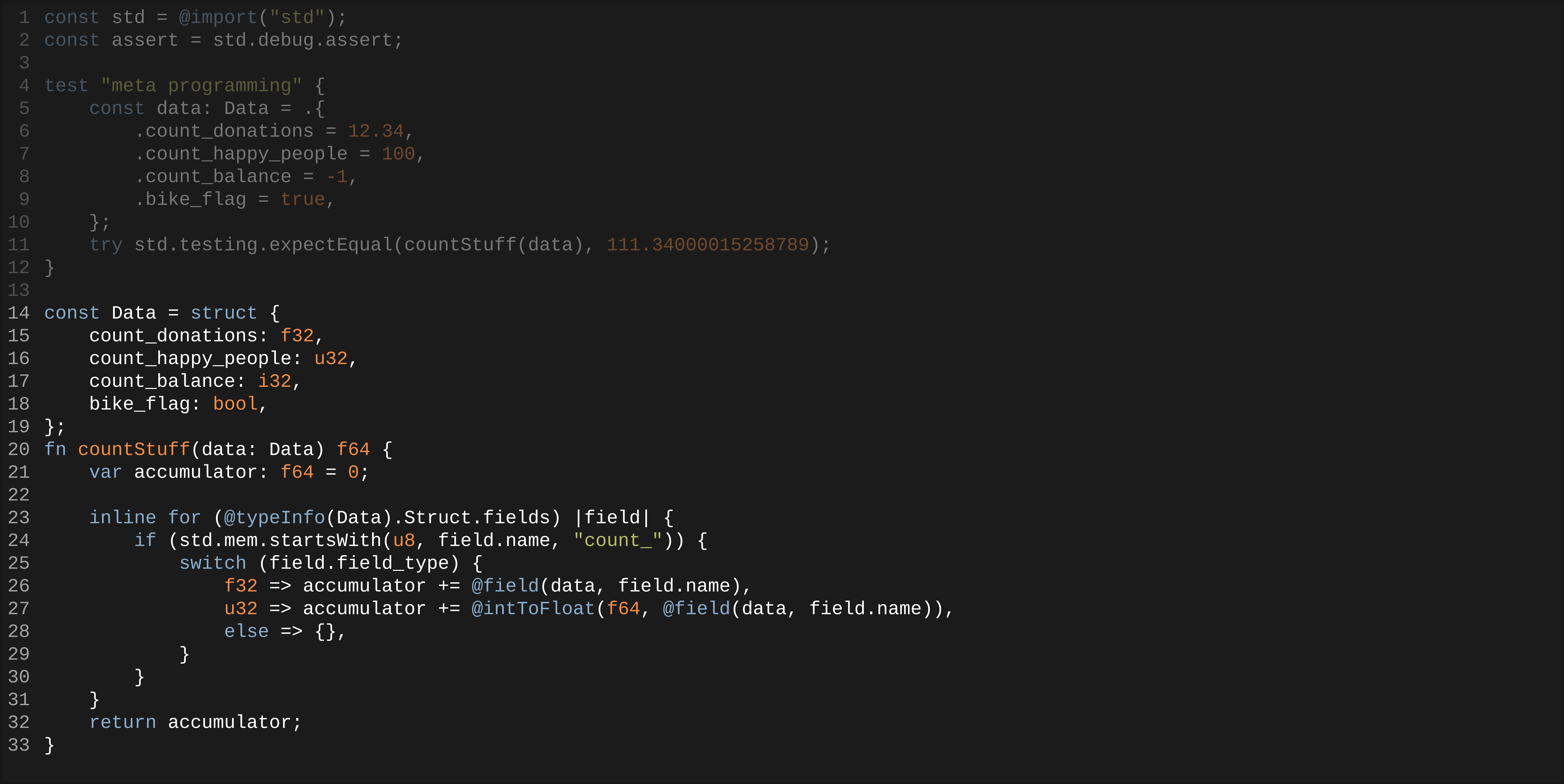


```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG



```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => {},
            }
        }
    }
    return accumulator;
}
```

SPOT THE BUG

```zig
const std = @import("std");
const assert = std.debug.assert;

test "meta programming" {
    const data: Data = .{
        .count_donations = 12.34,
        .count_happy_people = 100,
        .count_balance = -1,
        .bike_flag = true,
    };
    try std.testing.expectEqual(countStuff(data), 111.34000015258789);
}

const Data = struct {
    count_donations: f32,
    count_happy_people: u32,
    count_balance: i32,
    bike_flag: bool,
};
fn countStuff(data: Data) f64 {
    var accumulator: f64 = 0;

    inline for (@typeInfo(Data).Struct.fields) |field| {
        if (std.mem.startsWith(u8, field.name, "count_")) {
            switch (field.field_type) {
                f32 => accumulator += @field(data, field.name),
                u32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                i32 => accumulator += @intToFloat(f64, @field(data, field.name)),
                else => @compileError("unhandled struct field type"),
            }
        }
    }
    return accumulator;
}
```

Speaker notes

Hook: Here is some Zig code.

1. Raise your hand if this is your first time ever seeing Zig code.
2. Keep your hand up if you can spot the bug.

Hopefully at least one person successfully debugs code in a language they have never seen before.

"That’s the power of Zig."

Hint: off by 1

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. ***Secret Surprise***

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. *Secret Surprise*

TALK OVERVIEW

1. What is the Zig Project?
2. Maintain It With Zig
3. How to Predict the Future
4. Zig in Action
5. A Taste of Zig
6. *Secret Surprise*

What is the Zig project?

Speaker notes

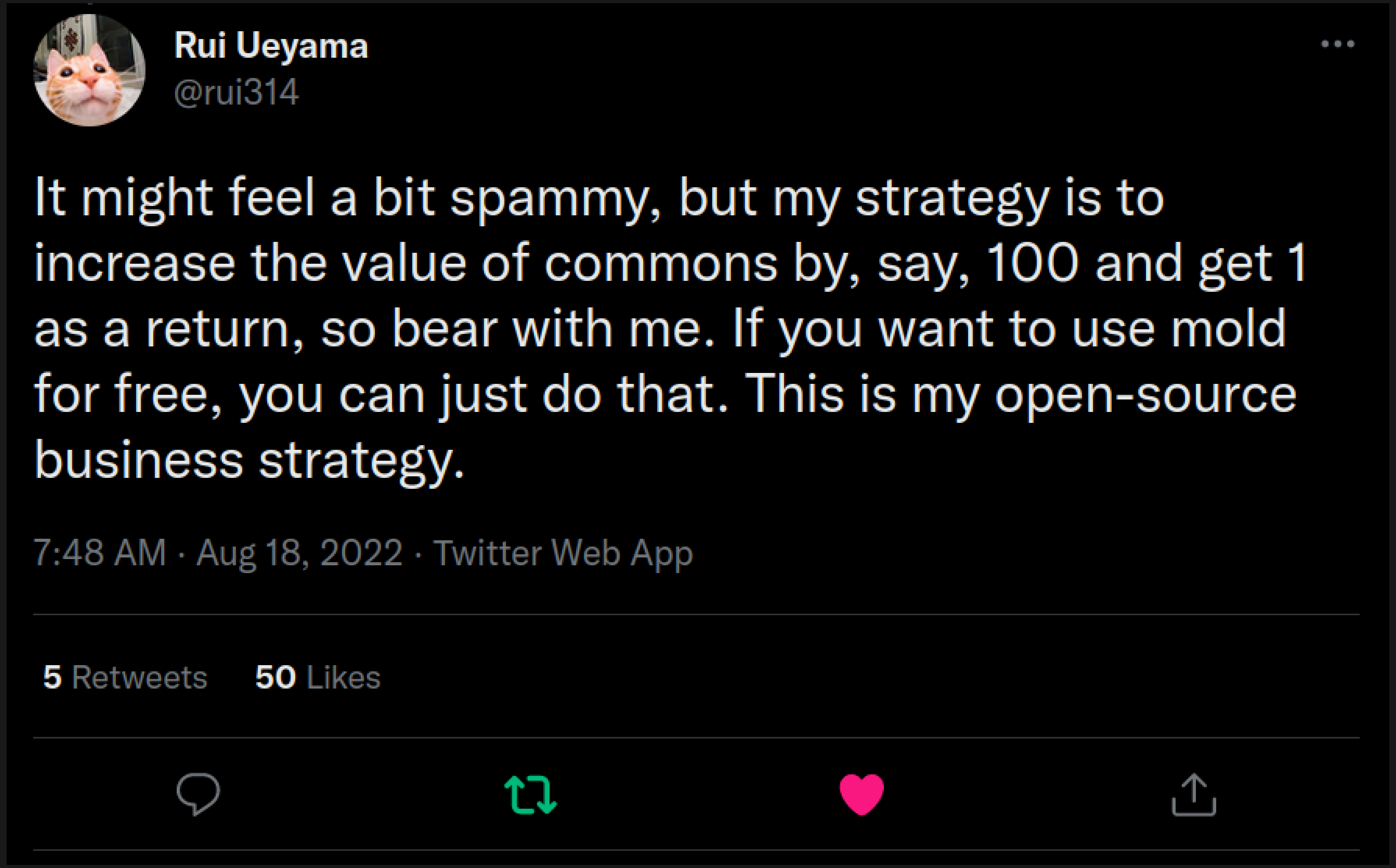
We got cute space lizards

What is the Zig project?

ZIG

A general-purpose programming language and toolchain for maintaining **robust**, **optimal**, and **reusable** software.

What is the Zig project?



Rui Ueyama
@rui314

It might feel a bit spammy, but my strategy is to increase the value of commons by, say, 100 and get 1 as a return, so bear with me. If you want to use mold for free, you can just do that. This is my open-source business strategy.

7:48 AM · Aug 18, 2022 · Twitter Web App

5 Retweets **50** Likes

Speaker notes

I love this quote from Rui about the mold project

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future

Speaker notes

SDL will be mentioned later

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
    - Example: Memory Allocation
    - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry
    - Tooling such as zig cc

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry
  - Tooling such as zig cc
    - Better defaults

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry
  - Tooling such as zig cc
    - Better defaults
  - Robust, high performance open source libraries for the industry to use

What is the Zig project?

- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry
  - Tooling such as zig cc
    - Better defaults
  - Robust, high performance open source libraries for the industry to use
    - Not only Zig but **all languages** via the C ABI

What is the Zig project?

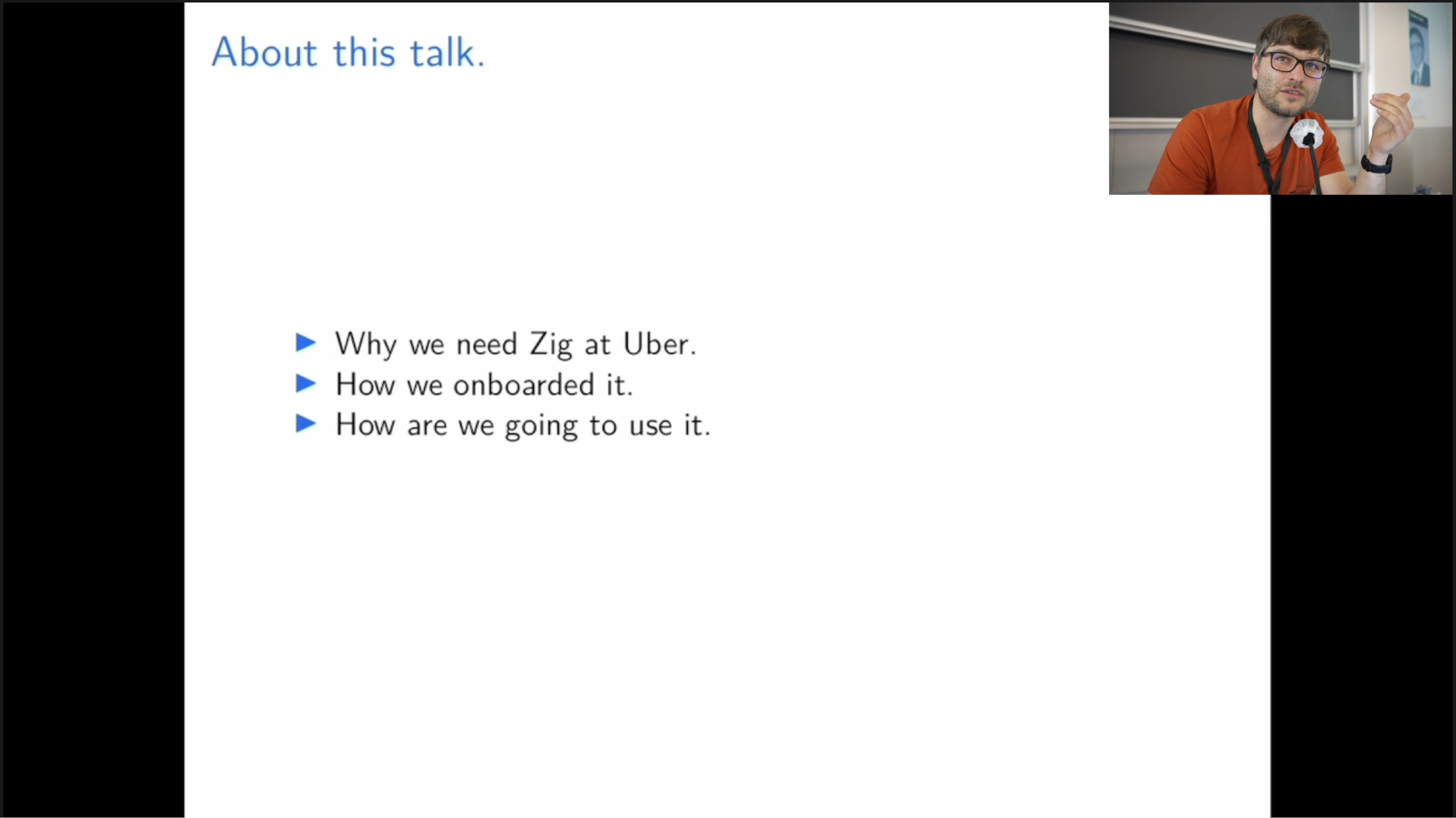
- Increase the utility of the commons, bringing human technology incrementally into the future
- Re-examine fundamental building blocks of software
  - Example: Memory Allocation
  - Example: Dependency on libc
- Raise the standards of software as a craft throughout the industry
  - Tooling such as zig cc
    - Better defaults
  - Robust, high performance open source libraries for the industry to use
    - Not only Zig but **all languages** via the C ABI
  - Providing guidance to students in ethics and skills

Maintain It With Zig 🔧

Maintain It With Zig 🔧

Level 1: Drop in `zig cc`

HERMETIC BUILDS

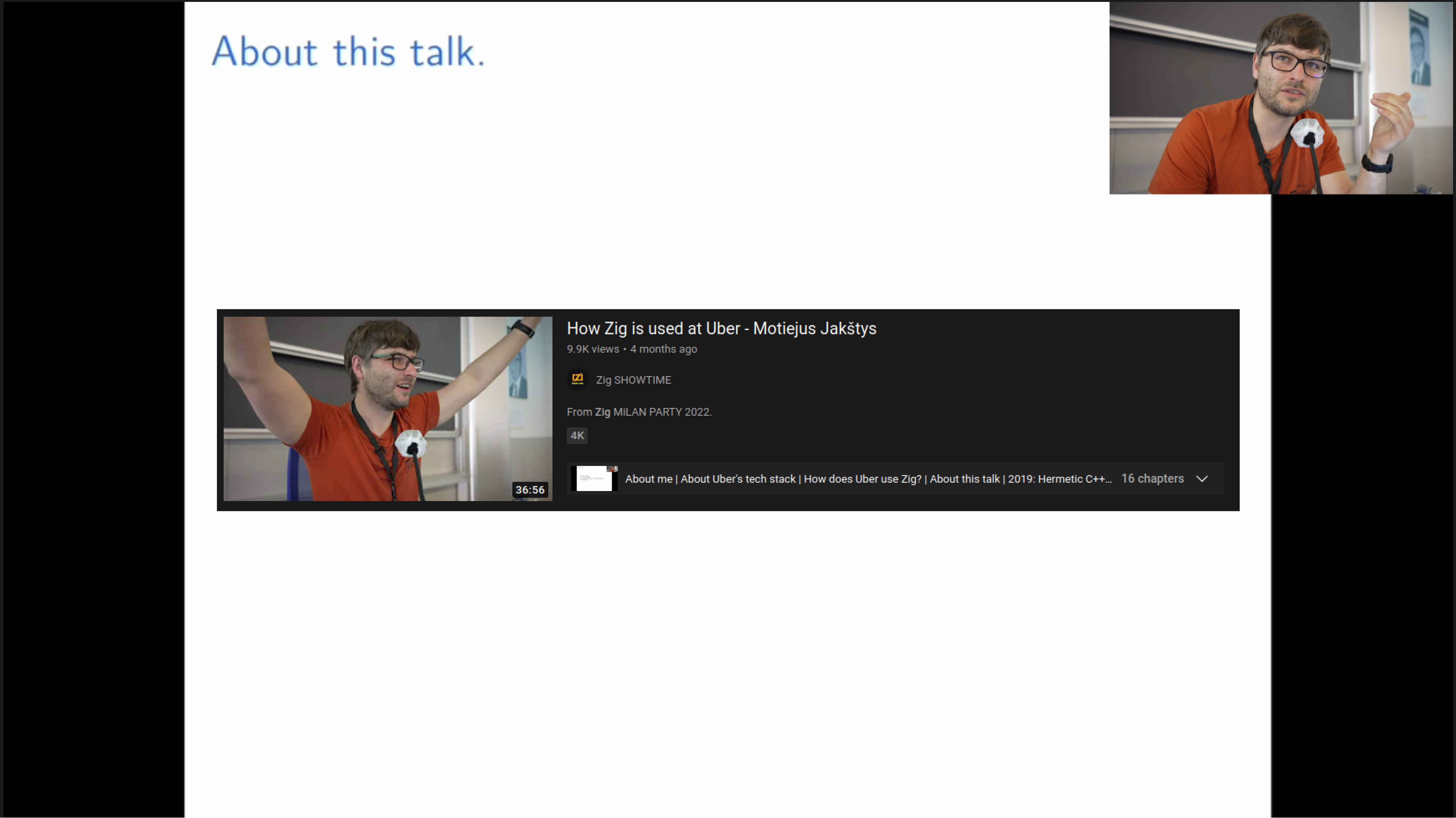


About this talk.

- Why we need Zig at Uber.
- How we onboarded it.
- How are we going to use it.

Level 1: `zig cc`

HERMETIC BUILDS



About this talk.

How Zig is used at Uber - Motiejus Jakštys

9.9K views • 4 months ago

Zig SHOWTIME

From **Zig** MiLAN PARTY 2022.

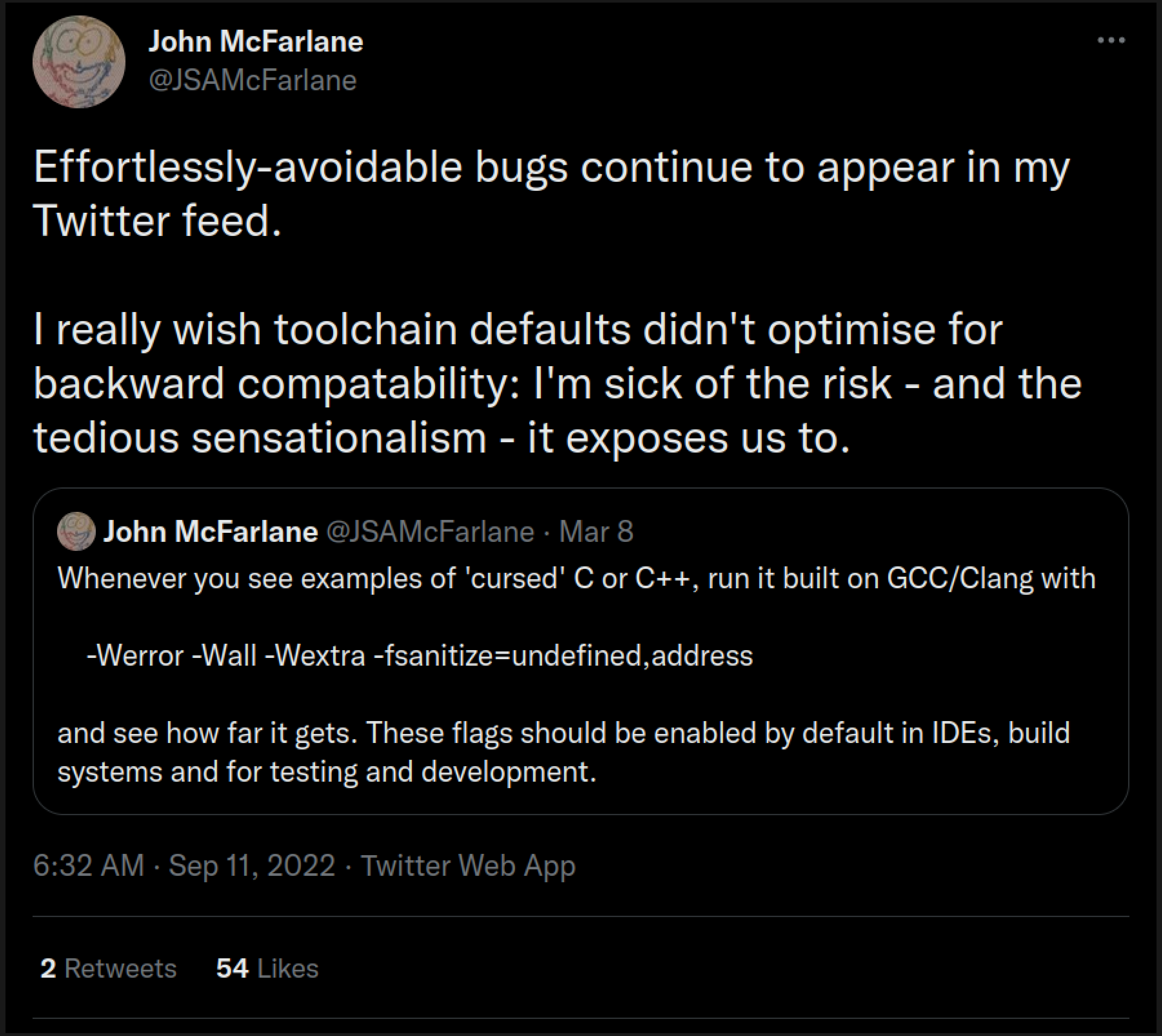
4K

36:56

About me | About Uber's tech stack | How does Uber use Zig? | About this talk | 2019: Hermetic C++... 16 chapters

Level 1: `zig cc`

UBSAN BY DEFAULT



John McFarlane
@JSAMcFarlane

Effortlessly-avoidable bugs continue to appear in my Twitter feed.

I really wish toolchain defaults didn't optimise for backward compatability: I'm sick of the risk - and the tedious sensationalism - it exposes us to.

> **John McFarlane** @JSAMcFarlane · Mar 8
>
> Whenever you see examples of 'cursed' C or C++, run it built on GCC/Clang with
>
> -Werror -Wall -Wextra -fsanitize=undefined,address
>
> and see how far it gets. These flags should be enabled by default in IDEs, build systems and for testing and development.

6:32 AM · Sep 11, 2022 · Twitter Web App

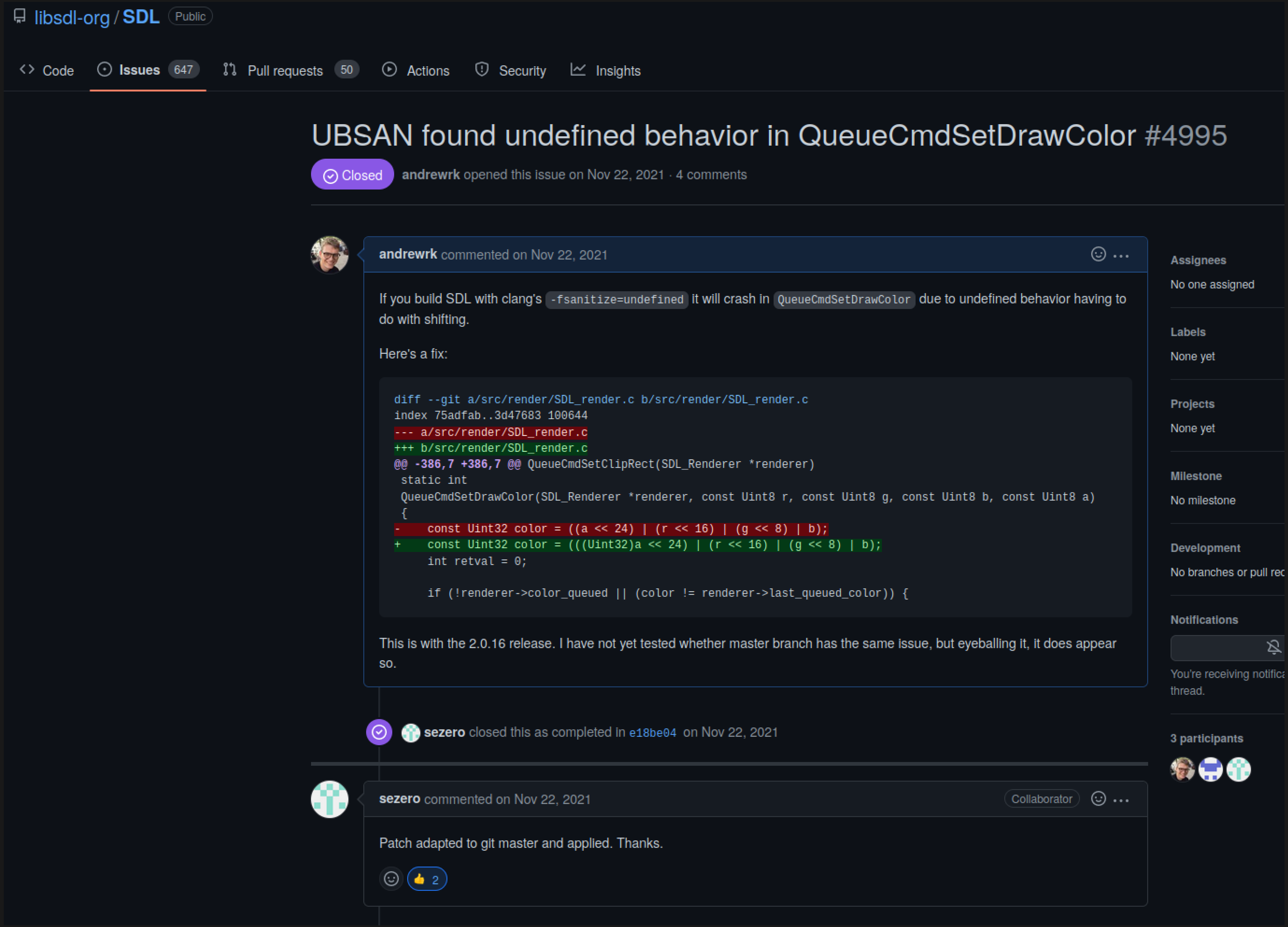
2 Retweets **54** Likes

Speaker notes

I'm happy to report that

1. Using zig cc as a drop-in C compiler is growing in popularity
2. It enables UBSAN by default
3. Real world bugs are being caught and fixed as a result!

UBSAN BY DEFAULT



UBSAN found undefined behavior in QueueCmdSetDrawColor #4995

Closed andrewrk opened this issue on Nov 22, 2021 · 4 comments

andrewrk commented on Nov 22, 2021

If you build SDL with clang's `-fsanitize=undefined` it will crash in `QueueCmdSetDrawColor` due to undefined behavior having to do with shifting.

Here's a fix:

```
diff --git a/src/render/SDL_render.c b/src/render/SDL_render.c
index 75adfab..3d47683 100644
--- a/src/render/SDL_render.c
+++ b/src/render/SDL_render.c
@@ -386,7 +386,7 @@ QueueCmdSetClipRect(SDL_Renderer *renderer)
 static int
 QueueCmdSetDrawColor(SDL_Renderer *renderer, const Uint8 r, const Uint8 g, const Uint8 b, const Uint8 a)
 {
-    const Uint32 color = ((a << 24) | (r << 16) | (g << 8) | b);
+    const Uint32 color = (((Uint32)a << 24) | (r << 16) | (g << 8) | b);
     int retval = 0;

     if (!renderer->color_queued || (color != renderer->last_queued_color)) {
```

This is with the 2.0.16 release. I have not yet tested whether master branch has the same issue, but eyeballing it, it does appear so.

sezero closed this as completed in `e18be04` on Nov 22, 2021

sezero commented on Nov 22, 2021 Collaborator

Patch adapted to git master and applied. Thanks.

👍 2

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull rec

Notifications

You're receiving notifica
thread.

3 participants

CROSS-COMPILATION

Level 1: `zig cc`

CROSS-COMPILATION

Level 1: `zig cc`

```
$ zig cc -o hello hello.c -target x86_64-windows
```

CROSS-COMPILATION

Level 1: `zig cc`

```
$ zig cc -o hello hello.c -target x86_64-windows
```

```
$ zig cc -o hello hello.c -target aarch64-macos
```

CROSS-COMPILATION

Level 1: zig cc

```
$ zig cc -o hello hello.c -target x86_64-windows
```

```
$ zig cc -o hello hello.c -target aarch64-macos
```

```
$ zig c++ -o hello hello.cpp
```

CROSS-COMPILATION

Level 1: `zig cc`

```
$ zig cc -o hello hello.c -target x86_64-windows
```

```
$ zig cc -o hello hello.c -target aarch64-macos
```

```
$ zig c++ -o hello hello.cpp
```

```
$ zig cc -o hello hello.c -target aarch64-linux-gnu.2.31
```

CROSS-COMPILATION

Level 1: `zig cc`

```
$ zig cc -o hello hello.c -target x86_64-windows

$ zig cc -o hello hello.c -target aarch64-macos

$ zig c++ -o hello hello.cpp

$ zig cc -o hello hello.c -target aarch64-linux-gnu.2.31

$ zig cc -o hello hello.c -target aarch64-linux-musl
```

Level 1: `zig cc`

BUILT-IN CACHING

Speaker notes

Zig needs to cache global stuff such as libc and compiler-rt so naturally it extends this feature to basic C objects too.

TRIVIAL INSTALLATION

Speaker notes

It once took me full 24 hours and 2 system restarts.

A "full 27 GB installation"?!

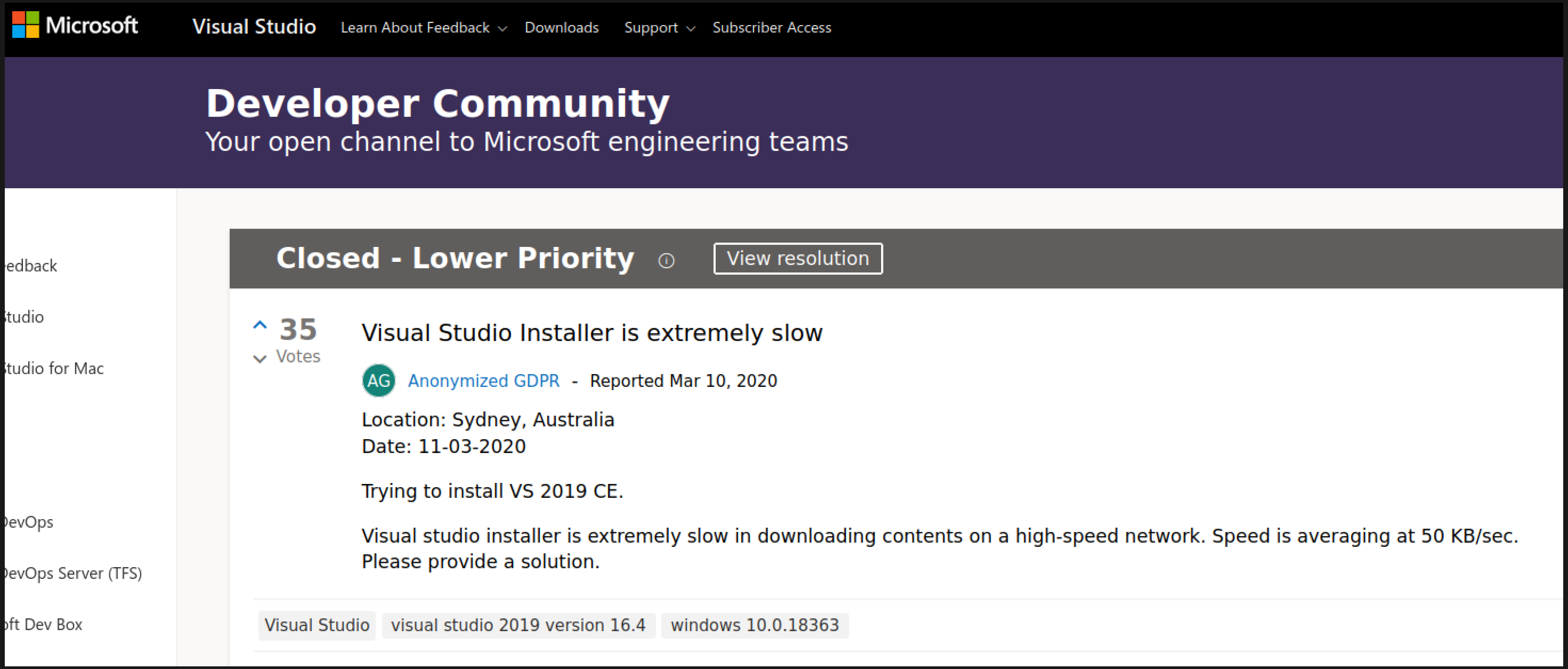
Level 1: `zig cc`

TRIVIAL INSTALLATION

How many people here have gone through the trouble to install MSVC?

Level 1: `zig cc`

TRIVIAL INSTALLATION



Developer Community

Your open channel to Microsoft engineering teams

Closed - Lower Priority

Visual Studio Installer is extremely slow

AG Anonymized GDPR - Reported Mar 10, 2020

Location: Sydney, Australia
Date: 11-03-2020

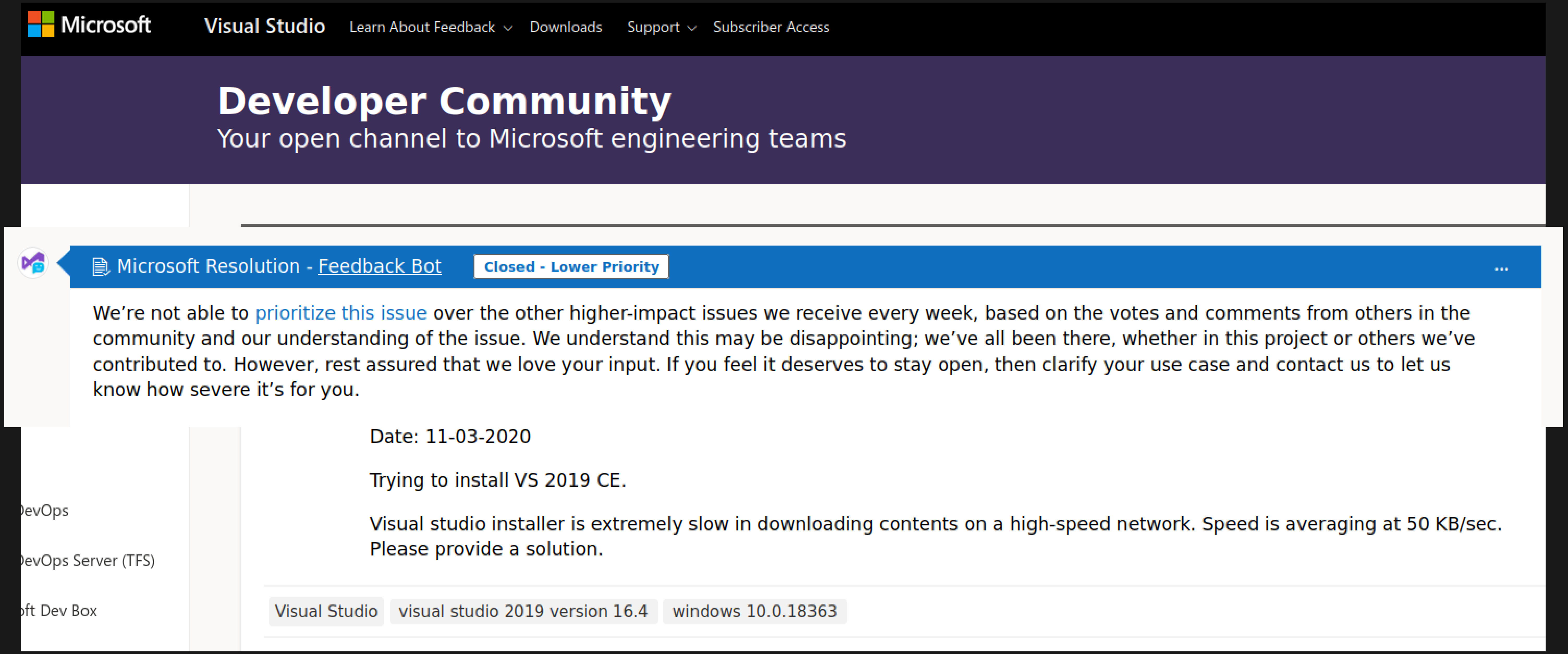
Trying to install VS 2019 CE.

Visual studio installer is extremely slow in downloading contents on a high-speed network. Speed is averaging at 50 KB/sec. Please provide a solution.

Visual Studio | visual studio 2019 version 16.4 | windows 10.0.18363

Level 1: `zig cc`

TRIVIAL INSTALLATION



Developer Community

Your open channel to Microsoft engineering teams

Microsoft Resolution - Feedback Bot **Closed - Lower Priority**

We're not able to prioritize this issue over the other higher-impact issues we receive every week, based on the votes and comments from others in the community and our understanding of the issue. We understand this may be disappointing; we've all been there, whether in this project or others we've contributed to. However, rest assured that we love your input. If you feel it deserves to stay open, then clarify your use case and contact us to let us know how severe it's for you.

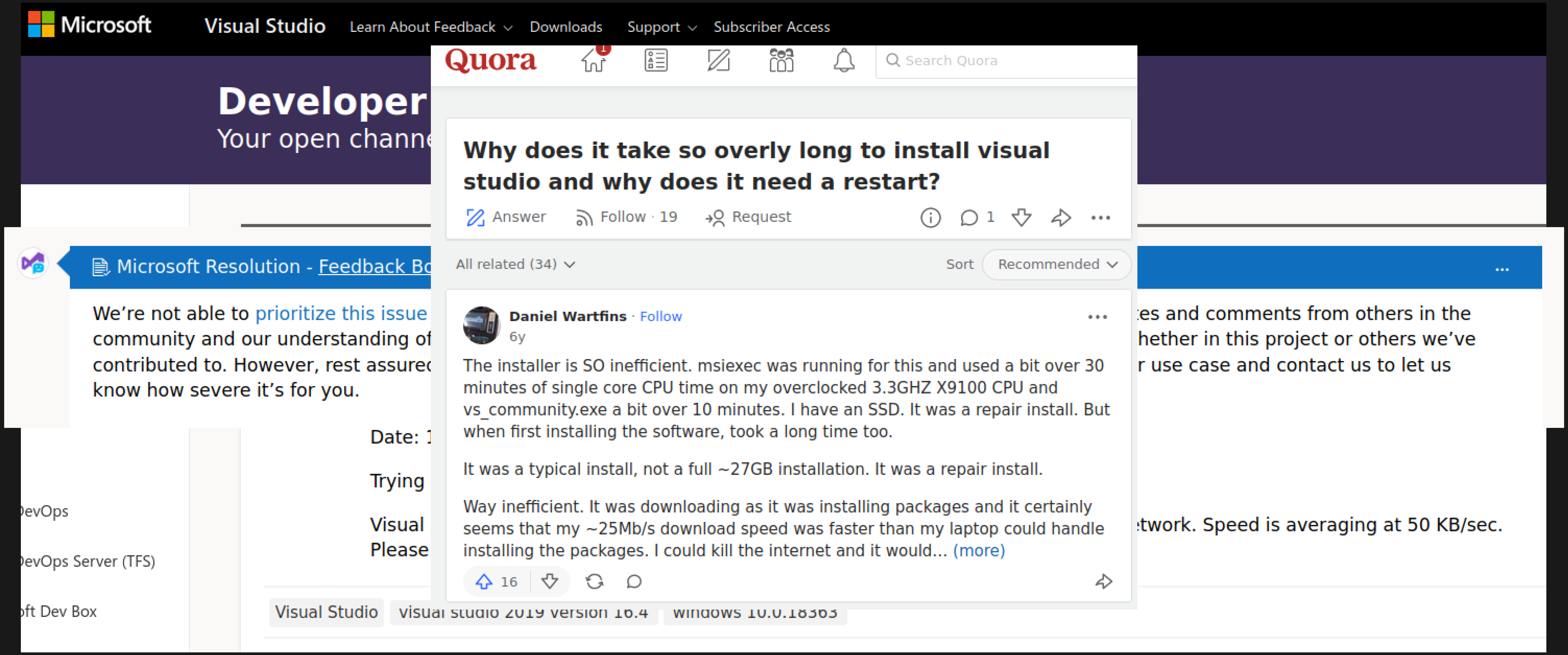
Date: 11-03-2020

Trying to install VS 2019 CE.

Visual studio installer is extremely slow in downloading contents on a high-speed network. Speed is averaging at 50 KB/sec. Please provide a solution.

Level 1: `zig cc`

TRIVIAL INSTALLATION



Microsoft Resolution - Feedback Bo

We're not able to prioritize this issue community and our understanding of contributed to. However, rest assured know how severe it's for you.

tes and comments from others in the hether in this project or others we've r use case and contact us to let us

Date: 1

Trying

Visual
Please

twork. Speed is averaging at 50 KB/sec.

Why does it take so overly long to install visual studio and why does it need a restart?

Daniel Wartfins · Follow
6y

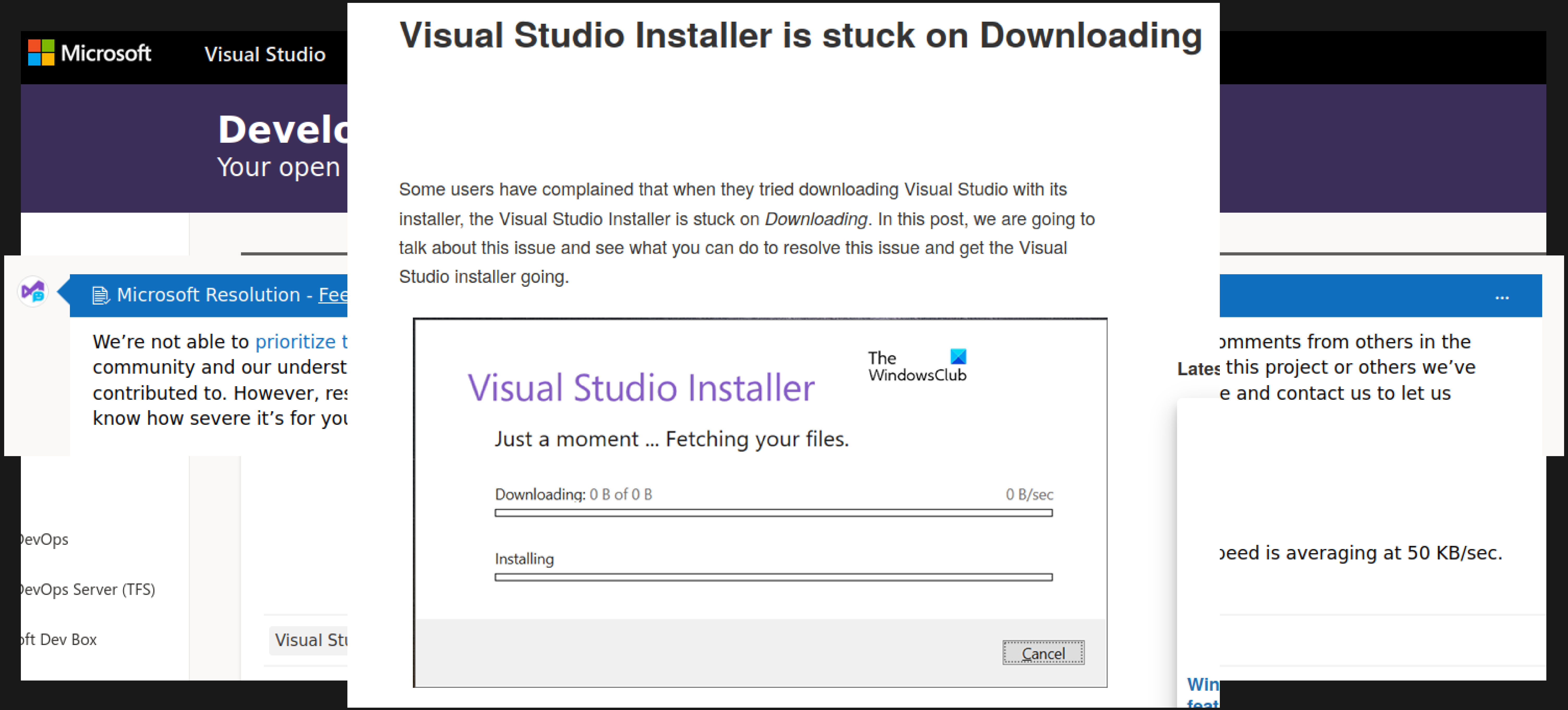
The installer is SO inefficient. msiexec was running for this and used a bit over 30 minutes of single core CPU time on my overclocked 3.3GHZ X9100 CPU and vs_community.exe a bit over 10 minutes. I have an SSD. It was a repair install. But when first installing the software, took a long time too.

It was a typical install, not a full ~27GB installation. It was a repair install.

Way inefficient. It was downloading as it was installing packages and it certainly seems that my ~25Mb/s download speed was faster than my laptop could handle installing the packages. I could kill the internet and it would… (more)

Level 1: `zig cc`

TRIVIAL INSTALLATION



Develo
Your open

Microsoft Resolution - Fee

We're not able to prioritize t
community and our underst
contributed to. However, res
know how severe it's for you

omments from others in the
Lates this project or others we've
e and contact us to let us

peed is averaging at 50 KB/sec.

Visual Studio Installer is stuck on Downloading

Some users have complained that when they tried downloading Visual Studio with its installer, the Visual Studio Installer is stuck on *Downloading*. In this post, we are going to talk about this issue and see what you can do to resolve this issue and get the Visual Studio installer going.

Just a moment ... Fetching your files.

Downloading: 0 B of 0 B 0 B/sec

Installing

TRIVIAL INSTALLATION

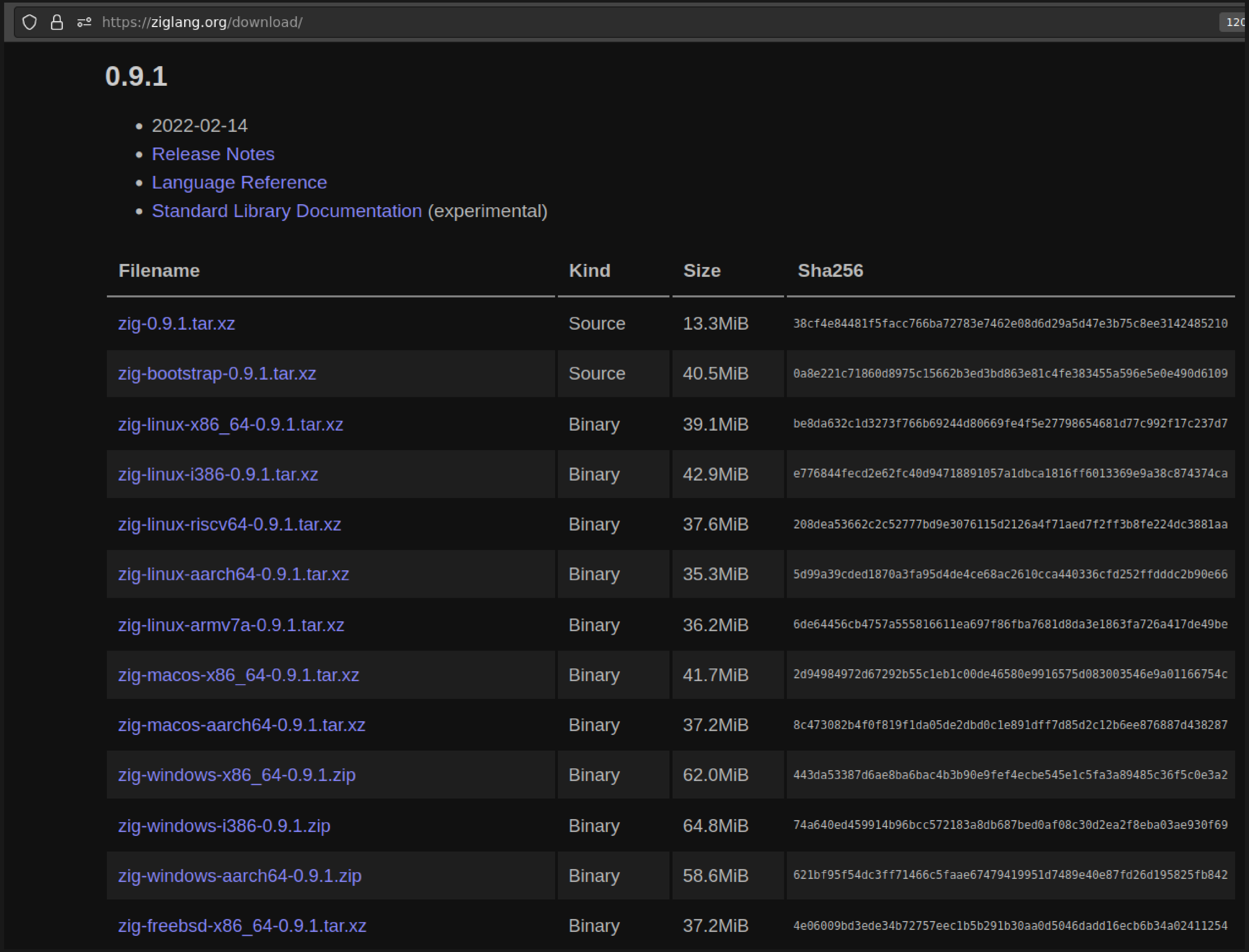
Speaker notes

Just unzip it. No installation process. No registry edits. Multiple versions live harmoniously side by side.

This would be extremely easy to say, bundle along with an IDE.

Level 1: `zig cc`

TRIVIAL INSTALLATION



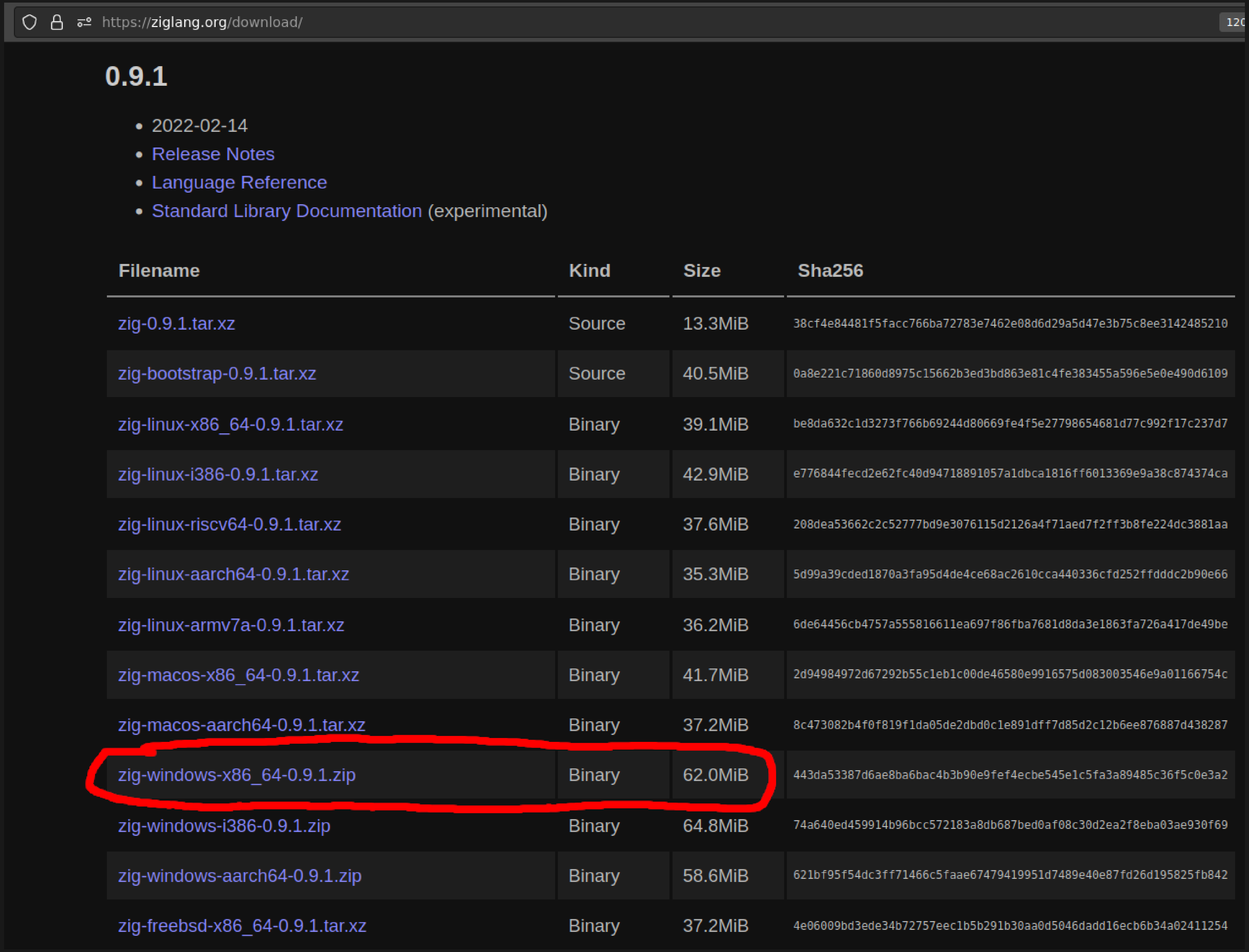
0.9.1

- 2022-02-14
- Release Notes
- Language Reference
- Standard Library Documentation (experimental)

Filename	Kind	Size	Sha256
zig-0.9.1.tar.xz	Source	13.3MiB	38cf4e84481f5facc766ba72783e7462e08d6d29a5d47e3b75c8ee3142485210
zig-bootstrap-0.9.1.tar.xz	Source	40.5MiB	0a8e221c71860d8975c15662b3ed3bd863e81c4fe383455a596e5e0e490d6109
zig-linux-x86_64-0.9.1.tar.xz	Binary	39.1MiB	be8da632c1d3273f766b69244d80669fe4f5e27798654681d77c992f17c237d7
zig-linux-i386-0.9.1.tar.xz	Binary	42.9MiB	e776844fecd2e62fc40d94718891057a1dbca1816ff6013369e9a38c874374ca
zig-linux-riscv64-0.9.1.tar.xz	Binary	37.6MiB	208dea53662c2c52777bd9e3076115d2126a4f71aed7f2ff3b8fe224dc3881aa
zig-linux-aarch64-0.9.1.tar.xz	Binary	35.3MiB	5d99a39cded1870a3fa95d4de4ce68ac2610cca440336cfd252ffdddc2b90e66
zig-linux-armv7a-0.9.1.tar.xz	Binary	36.2MiB	6de64456cb4757a555816611ea697f86fba7681d8da3e1863fa726a417de49be
zig-macos-x86_64-0.9.1.tar.xz	Binary	41.7MiB	2d94984972d67292b55c1eb1c00de46580e9916575d083003546e9a01166754c
zig-macos-aarch64-0.9.1.tar.xz	Binary	37.2MiB	8c473082b4f0f819f1da05de2dbd0c1e891dff7d85d2c12b6ee876887d438287
zig-windows-x86_64-0.9.1.zip	Binary	62.0MiB	443da53387d6ae8ba6bac4b3b90e9fef4ecbe545e1c5fa3a89485c36f5c0e3a2
zig-windows-i386-0.9.1.zip	Binary	64.8MiB	74a640ed459914b96bcc572183a8db687bed0af08c30d2ea2f8eba03ae930f69
zig-windows-aarch64-0.9.1.zip	Binary	58.6MiB	621bf95f54dc3ff71466c5faae67479419951d7489e40e87fd26d195825fb842
zig-freebsd-x86_64-0.9.1.tar.xz	Binary	37.2MiB	4e06009bd3ede34b72757eec1b5b291b30aa0d5046dadd16ecb6b34a02411254

Level 1: `zig cc`

TRIVIAL INSTALLATION



0.9.1

- 2022-02-14
- Release Notes
- Language Reference
- Standard Library Documentation (experimental)

Filename	Kind	Size	Sha256
zig-0.9.1.tar.xz	Source	13.3MiB	38cf4e84481f5facc766ba72783e7462e08d6d29a5d47e3b75c8ee3142485210
zig-bootstrap-0.9.1.tar.xz	Source	40.5MiB	0a8e221c71860d8975c15662b3ed3bd863e81c4fe383455a596e5e0e490d6109
zig-linux-x86_64-0.9.1.tar.xz	Binary	39.1MiB	be8da632c1d3273f766b69244d80669fe4f5e27798654681d77c992f17c237d7
zig-linux-i386-0.9.1.tar.xz	Binary	42.9MiB	e776844fecd2e62fc40d94718891057a1dbca1816ff6013369e9a38c874374ca
zig-linux-riscv64-0.9.1.tar.xz	Binary	37.6MiB	208dea53662c2c52777bd9e3076115d2126a4f71aed7f2ff3b8fe224dc3881aa
zig-linux-aarch64-0.9.1.tar.xz	Binary	35.3MiB	5d99a39cded1870a3fa95d4de4ce68ac2610cca440336cfd252ffdddc2b90e66
zig-linux-armv7a-0.9.1.tar.xz	Binary	36.2MiB	6de64456cb4757a555816611ea697f86fba7681d8da3e1863fa726a417de49be
zig-macos-x86_64-0.9.1.tar.xz	Binary	41.7MiB	2d94984972d67292b55c1eb1c00de46580e9916575d083003546e9a01166754c
zig-macos-aarch64-0.9.1.tar.xz	Binary	37.2MiB	8c473082b4f0f819f1da05de2dbd0c1e891dff7d85d2c12b6ee876887d438287
zig-windows-x86_64-0.9.1.zip	Binary	62.0MiB	443da53387d6ae8ba6bac4b3b90e9fef4ecbe545e1c5fa3a89485c36f5c0e3a2
zig-windows-i386-0.9.1.zip	Binary	64.8MiB	74a640ed459914b96bcc572183a8db687bed0af08c30d2ea2f8eba03ae930f69
zig-windows-aarch64-0.9.1.zip	Binary	58.6MiB	621bf95f54dc3ff71466c5faae67479419951d7489e40e87fd26d195825fb842
zig-freebsd-x86_64-0.9.1.tar.xz	Binary	37.2MiB	4e06009bd3ede34b72757eec1b5b291b30aa0d5046dadd16ecb6b34a02411254

Level 1: `zig cc`

TRIVIAL INSTALLATION

```
$ time bash -c 'wget https://ziglang.org/download/0.9.1/zig-windows-x86_64-0.9.1.zip && unzip zig-windows-x86_64-0.9.1.zip'
real  0m4.884s
user  0m1.422s
sys 0m0.570s
```

Maintain It With Zig 🔧

Level 2: Exploit the Zig Build System

Level 2: `zig build`

MAINTAIN IT WITH ZIG 🔧

Create a build.zig script in a **real language**, using a **declarative API**.

```zig
const std = @import("std");
const Builder = std.build.Builder;

pub fn build(b: *Builder) void {
    const mode = b.standardReleaseOptions();
    const windows = b.option(bool, "windows", "create windows build") orelse false;

    const exe = b.addExecutable("tetris", null);
    exe.addCSourceFile("src/main.c", &[_][]const u8{"-std=c11"});
    exe.addCSourceFile("stb_image-2.22/stb_image_impl.c", &[_][]const u8{"-std=c99"});
    exe.addIncludePath("stb_image-2.22");
    exe.setBuildMode(mode);

    if (windows) {
        exe.setTarget(.{
            .cpu_arch = .x86_64,
            .os_tag = .windows,
            .abi = .gnu,
        });
    }

    exe.linkSystemLibrary("c");
    exe.linkSystemLibrary("glfw");
    exe.linkSystemLibrary("epoxy");
    exe.install();

    const play = b.step("play", "Play the game");
    const run = exe.run();
    run.step.dependOn(b.getInstallStep());
    play.dependOn(&run.step);
}
```

Level 2: `zig build`

MAINTAIN IT WITH ZIG 🔧

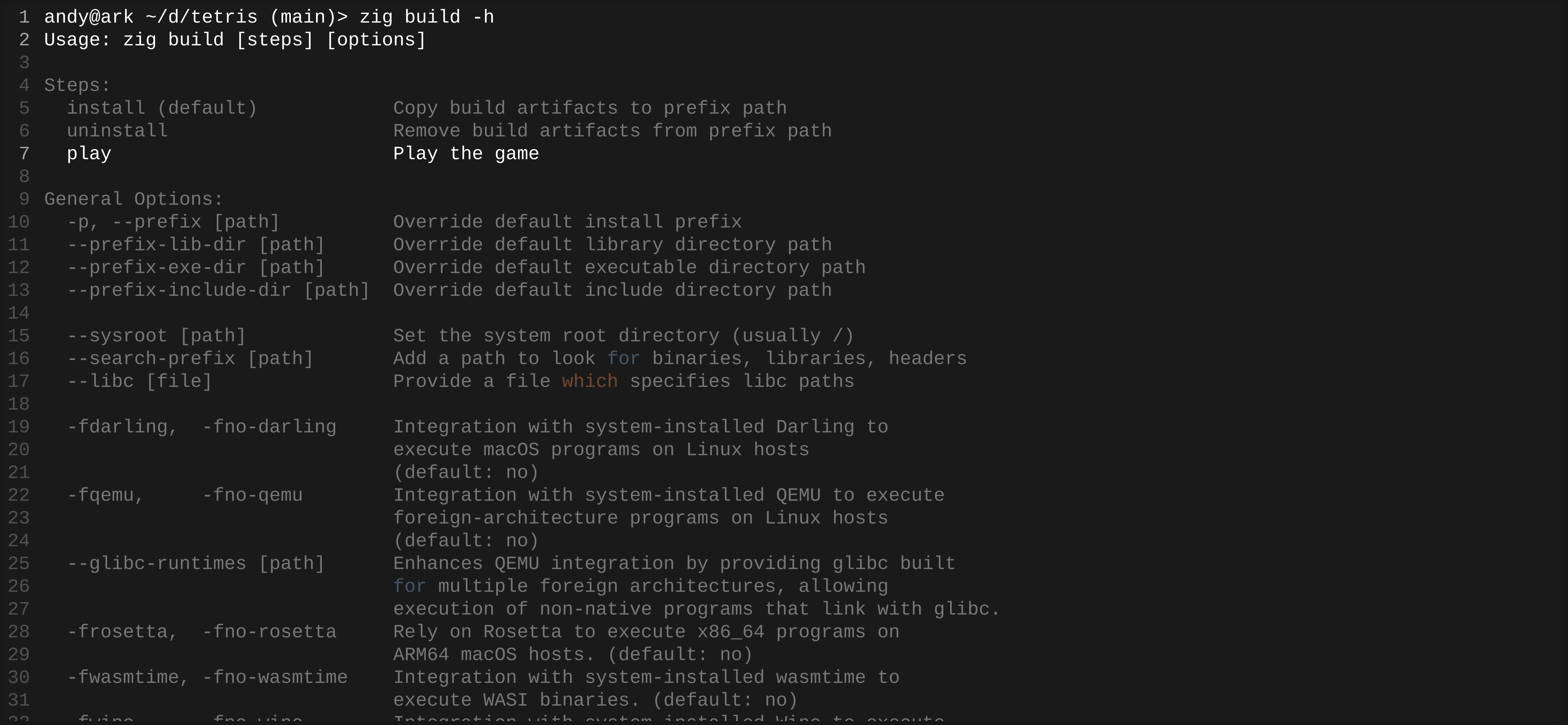
Create a build.zig script in a **real language**, using a **declarative API**.

```
const std = @import("std");
const Builder = std.build.Builder;

pub fn build(b: *Builder) void {
    const mode = b.standardReleaseOptions();
    const windows = b.option(bool, "windows", "create windows build") orelse false;

    const exe = b.addExecutable("tetris", null);
    exe.addCSourceFile("src/main.c", &[_][]const u8{"-std=c11"});
    exe.addCSourceFile("stb_image-2.22/stb_image_impl.c", &[_][]const u8{"-std=c99"});
    exe.addIncludePath("stb_image-2.22");
    exe.setBuildMode(mode);

    if (windows) {
        exe.setTarget(.{
            .cpu_arch = .x86_64,
            .os_tag = .windows,
            .abi = .gnu,
        });
    }

    exe.linkSystemLibrary("c");
    exe.linkSystemLibrary("glfw");
    exe.linkSystemLibrary("epoxy");
    exe.install();

    const play = b.step("play", "Play the game");
    const run = exe.run();
    run.step.dependOn(b.getInstallStep());
    play.dependOn(&run.step);
}
```

Level 2: `zig build`

MAINTAIN IT WITH ZIG 🔧

Use `zig build` for a **consistent**, **cross-platform** command-line interface.

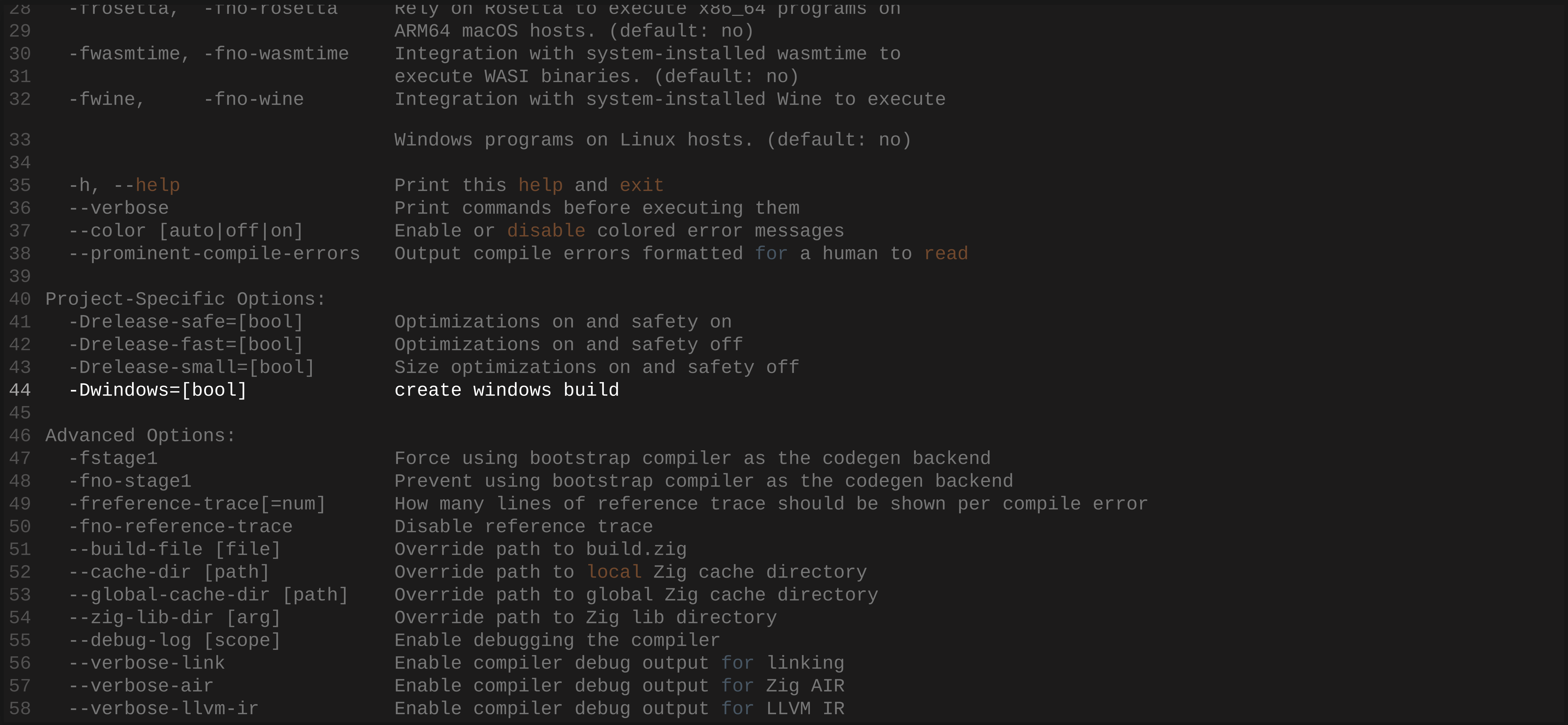


```
andy@ark ~/d/tetris (main)> zig build -h
Usage: zig build [steps] [options]

Steps:
  install (default)            Copy build artifacts to prefix path
  uninstall                    Remove build artifacts from prefix path
  play                         Play the game

General Options:
  -p, --prefix [path]          Override default install prefix
  --prefix-lib-dir [path]      Override default library directory path
  --prefix-exe-dir [path]      Override default executable directory path
  --prefix-include-dir [path]  Override default include directory path

  --sysroot [path]             Set the system root directory (usually /)
  --search-prefix [path]       Add a path to look for binaries, libraries, headers
  --libc [file]                Provide a file which specifies libc paths

  -fdarling,  -fno-darling     Integration with system-installed Darling to
                               execute macOS programs on Linux hosts
                               (default: no)
  -fqemu,     -fno-qemu        Integration with system-installed QEMU to execute
                               foreign-architecture programs on Linux hosts
                               (default: no)
  --glibc-runtimes [path]      Enhances QEMU integration by providing glibc built
                               for multiple foreign architectures, allowing
                               execution of non-native programs that link with glibc.
  -frosetta,  -fno-rosetta     Rely on Rosetta to execute x86_64 programs on
                               ARM64 macOS hosts. (default: no)
  -fwasmtime, -fno-wasmtime    Integration with system-installed wasmtime to
                               execute WASI binaries. (default: no)
  -fwine,     -fno-wine        Integration with system-installed Wine to execute
```

Level 2: `zig build`

MAINTAIN IT WITH ZIG 🔧

Use `zig build` for a **consistent**, **cross-platform** command-line interface.



```
  -frosetta,  -fno-rosetta     Rely on Rosetta to execute x86_64 programs on
                               ARM64 macOS hosts. (default: no)
  -fwasmtime, -fno-wasmtime    Integration with system-installed wasmtime to
                               execute WASI binaries. (default: no)
  -fwine,     -fno-wine        Integration with system-installed Wine to execute
                               Windows programs on Linux hosts. (default: no)

  -h, --help                   Print this help and exit
  --verbose                    Print commands before executing them
  --color [auto|off|on]        Enable or disable colored error messages
  --prominent-compile-errors   Output compile errors formatted for a human to read

Project-Specific Options:
  -Drelease-safe=[bool]        Optimizations on and safety on
  -Drelease-fast=[bool]        Optimizations on and safety off
  -Drelease-small=[bool]       Size optimizations on and safety off
  -Dwindows=[bool]             create windows build

Advanced Options:
  -fstage1                     Force using bootstrap compiler as the codegen backend
  -fno-stage1                  Prevent using bootstrap compiler as the codegen backend
  -freference-trace[=num]      How many lines of reference trace should be shown per compile error
  -fno-reference-trace         Disable reference trace
  --build-file [file]          Override path to build.zig
  --cache-dir [path]           Override path to local Zig cache directory
  --global-cache-dir [path]    Override path to global Zig cache directory
  --zig-lib-dir [arg]          Override path to Zig lib directory
  --debug-log [scope]          Enable debugging the compiler
  --verbose-link               Enable compiler debug output for linking
  --verbose-air                Enable compiler debug output for Zig AIR
  --verbose-llvm-ir            Enable compiler debug output for LLVM IR
```

Level 2: `zig build`

PACKAGE MANAGER

Coming soon: fulfill C/C++/Zig dependencies with a built-in package manager

Speaker notes

How many steps does your README have in the "building from source" section?

Maintain It With Zig 🔧

Level 3: Write a component in Zig

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

Updating the build script

```zig
const std = @import("std");

pub fn build(b: *std.build.Builder) void {
    const target = b.standardTargetOptions(.{});
    const mode = b.standardReleaseOptions();

    const exe = b.addExecutable("foobar", null);
    exe.addCSourceFile("src/bar.c", &.{"-std=c99"});
    exe.setTarget(target);
    exe.setBuildMode(mode);
    exe.linkLibC();
    exe.install();
}
```

Speaker notes

no dependencies are introduced; the project already depended on zig

WRITING NEW CODE IN ZIG

Updating the build script

```zig
const std = @import("std");

pub fn build(b: *std.build.Builder) void {
    const target = b.standardTargetOptions(.{});
    const mode = b.standardReleaseOptions();

    const exe = b.addExecutable("foobar", null);
    exe.addCSourceFile("src/bar.c", &.{"-std=c99"});
    exe.setTarget(target);
    exe.setBuildMode(mode);
    exe.linkLibC();
    exe.install();
}
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

Updating the build script

```
const std = @import("std");

pub fn build(b: *std.build.Builder) void {
    const target = b.standardTargetOptions(.{});
    const mode = b.standardReleaseOptions();

    const exe = b.addExecutable("foobar", "src/foo.zig");
    exe.addCSourceFile("src/bar.c", &.{"-std=c99"});
    exe.setTarget(target);
    exe.setBuildMode(mode);
    exe.linkLibC();
    exe.install();
}
```

Speaker notes

no dependencies are introduced; the project already depended on zig

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

Updating the build script

```zig
const std = @import("std");

pub fn build(b: *std.build.Builder) void {
    const target = b.standardTargetOptions(.{});
    const mode = b.standardReleaseOptions();

    const exe = b.addExecutable("foobar", "src/foo.zig");
    exe.addCSourceFile("src/bar.c", &.{"-std=c99"});
    exe.setTarget(target);
    exe.setBuildMode(mode);
    exe.linkLibC();
    exe.install();
}
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

foo.zig

```zig
const std = @import("std");

export fn dump_stack_trace() void {
    std.debug.dumpCurrentStackTrace(null);
}

export fn sum_array(ints_ptr: [*]const i32, ints_len: usize) i32 {
    return sumArray(ints_ptr[0..ints_len]);
}

fn sumArray(ints: []const i32) i32 {
    var sum: i32 = 0;
    for (ints) |int| {
        sum += int;
    }
    return sum;
}

pub const _start = void;
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

foo.zig

```
const std = @import("std");

export fn dump_stack_trace() void {
    std.debug.dumpCurrentStackTrace(null);
}

export fn sum_array(ints_ptr: [*]const i32, ints_len: usize) i32 {
    return sumArray(ints_ptr[0..ints_len]);
}

fn sumArray(ints: []const i32) i32 {
    var sum: i32 = 0;
    for (ints) |int| {
        sum += int;
    }
    return sum;
}

pub const _start = void;
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

foo.zig

```zig
const std = @import("std");

export fn dump_stack_trace() void {
    std.debug.dumpCurrentStackTrace(null);
}

export fn sum_array(ints_ptr: [*]const i32, ints_len: usize) i32 {
    return sumArray(ints_ptr[0..ints_len]);
}

fn sumArray(ints: []const i32) i32 {
    var sum: i32 = 0;
    for (ints) |int| {
        sum += int;
    }
    return sum;
}

pub const _start = void;
```

WRITING NEW CODE IN ZIG

foo.zig

```zig
const std = @import("std");

export fn dump_stack_trace() void {
    std.debug.dumpCurrentStackTrace(null);
}

export fn sum_array(ints_ptr: [*]const i32, ints_len: usize) i32 {
    return sumArray(ints_ptr[0..ints_len]);
}

fn sumArray(ints: []const i32) i32 {
    var sum: i32 = 0;
    for (ints) |int| {
        sum += int;
    }
    return sum;
}

pub const _start = void;
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

foo.zig

```zig
const std = @import("std");

export fn dump_stack_trace() void {
    std.debug.dumpCurrentStackTrace(null);
}

export fn sum_array(ints_ptr: [*]const i32, ints_len: usize) i32 {
    return sumArray(ints_ptr[0..ints_len]);
}

fn sumArray(ints: []const i32) i32 {
    var sum: i32 = 0;
    for (ints) |int| {
        sum += int;
    }
    return sum;
}

pub const _start = void;
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

bar.c

```c
#include <stdio.h>

void dump_stack_trace(void);
int sum_array(int *ptr, size_t len);

static void foo(int dump) {
    if (dump) {
        dump_stack_trace();
    }
}

int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

bar.c

```c
#include <stdio.h>

void dump_stack_trace(void);
int sum_array(int *ptr, size_t len);

static void foo(int dump) {
    if (dump) {
        dump_stack_trace();
    }
}

int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

bar.c

```c
#include <stdio.h>

void dump_stack_trace(void);
int sum_array(int *ptr, size_t len);

static void foo(int dump) {
    if (dump) {
        dump_stack_trace();
    }
}

int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

Level 3: Mixing C/C++/Zig

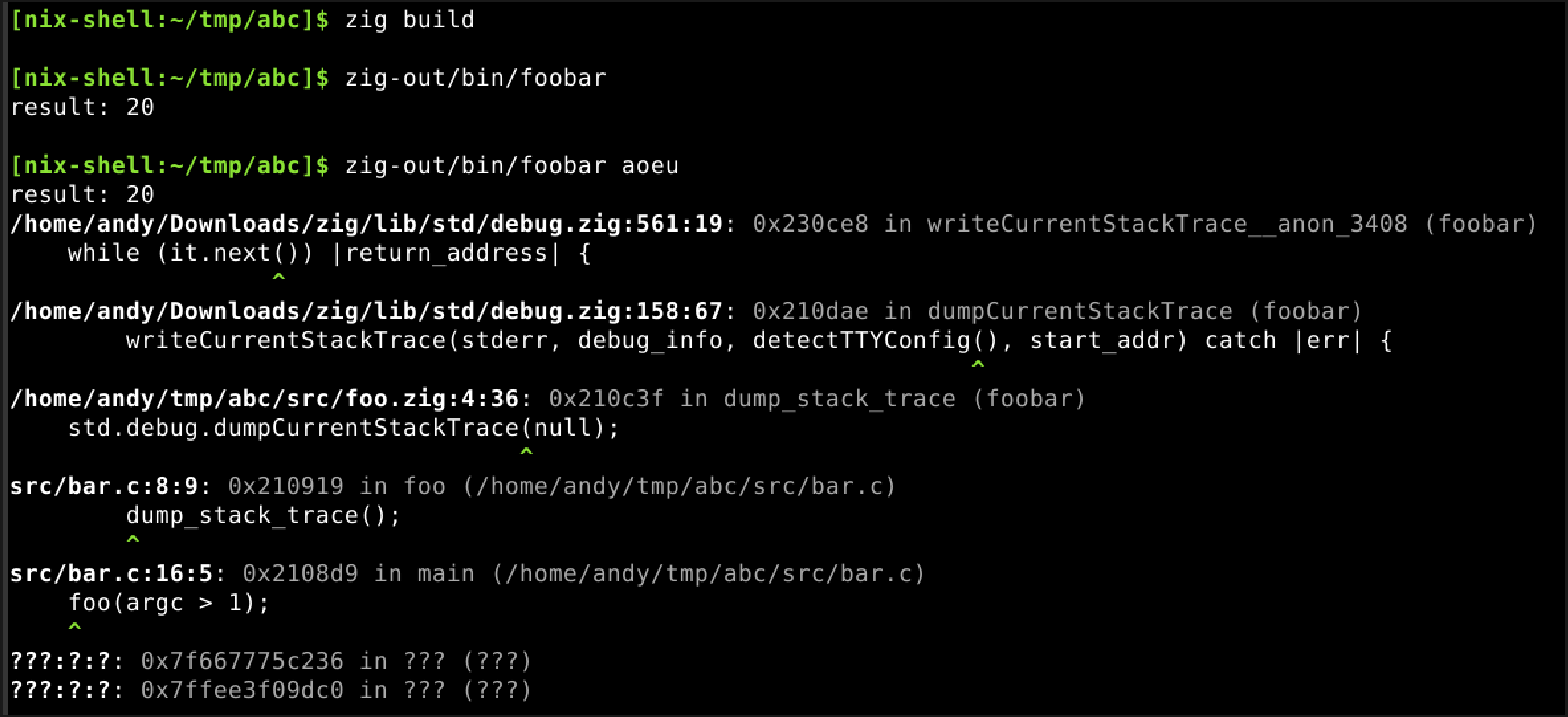
WRITING NEW CODE IN ZIG

bar.c

```c
#include <stdio.h>

void dump_stack_trace(void);
int sum_array(int *ptr, size_t len);

static void foo(int dump) {
    if (dump) {
        dump_stack_trace();
    }
}

int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG



```
[nix-shell:~/tmp/abc]$ zig build

[nix-shell:~/tmp/abc]$ zig-out/bin/foobar
result: 20

[nix-shell:~/tmp/abc]$ zig-out/bin/foobar aoeu
result: 20
/home/andy/Downloads/zig/lib/std/debug.zig:561:19: 0x230ce8 in writeCurrentStackTrace__anon_3408 (foobar)
    while (it.next()) |return_address| {
                  ^
/home/andy/Downloads/zig/lib/std/debug.zig:158:67: 0x210dae in dumpCurrentStackTrace (foobar)
        writeCurrentStackTrace(stderr, debug_info, detectTTYConfig(), start_addr) catch |err| {
                                                                  ^
/home/andy/tmp/abc/src/foo.zig:4:36: 0x210c3f in dump_stack_trace (foobar)
    std.debug.dumpCurrentStackTrace(null);
                                   ^
src/bar.c:8:9: 0x210919 in foo (/home/andy/tmp/abc/src/bar.c)
        dump_stack_trace();
        ^
src/bar.c:16:5: 0x2108d9 in main (/home/andy/tmp/abc/src/bar.c)
    foo(argc > 1);
    ^
???:?:?: 0x7f667775c236 in ??? (???)
???:?:?: 0x7ffee3f09dc0 in ??? (???)
```

WRITING NEW CODE IN ZIG

LTO

```c
int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

```
$ zig build -Drelease-fast
$ objdump -d zig-out/bin/foobar -Mintel | vim -
```

```
0000000000216380 <main>:
  216380: 53                    push   rbx
  216381: 89 fb                 mov    ebx,edi
  216383: bf 19 26 20 00        mov    edi,0x202619
  216388: be 14 00 00 00        mov    esi,0x14
  21638d: 31 c0                 xor    eax,eax
  21638f: e8 4c 00 00 00        call   2163e0 <printf@plt>
  216394: 83 fb 02              cmp    ebx,0x2
  216397: 7c 0a                 jl     2163a3 <main+0x23>
  216399: bf 60 0d 20 00        mov    edi,0x200d60
  21639e: e8 fd e8 fe ff        call   204ca0 <debug.dumpCurrentStackTrace>
  2163a3: 31 c0                 xor    eax,eax
  2163a5: 5b                    pop    rbx
  2163a6: c3                    ret
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

LTO

```
int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

```
$ zig build -Drelease-fast
$ objdump -d zig-out/bin/foobar -Mintel | vim -
```

```
0000000000216380 <main>:
  216380: 53                    push   rbx
  216381: 89 fb                 mov    ebx,edi
  216383: bf 19 26 20 00        mov    edi,0x202619
  216388: be 14 00 00 00        mov    esi,0x14
  21638d: 31 c0                 xor    eax,eax
  21638f: e8 4c 00 00 00        call   2163e0 <printf@plt>
  216394: 83 fb 02              cmp    ebx,0x2
  216397: 7c 0a                 jl     2163a3 <main+0x23>
  216399: bf 60 0d 20 00        mov    edi,0x200d60
  21639e: e8 fd e8 fe ff        call   204ca0 <debug.dumpCurrentStackTrace>
  2163a3: 31 c0                 xor    eax,eax
  2163a5: 5b                    pop    rbx
  2163a6: c3                    ret
```

Level 3: Mixing C/C++/Zig

WRITING NEW CODE IN ZIG

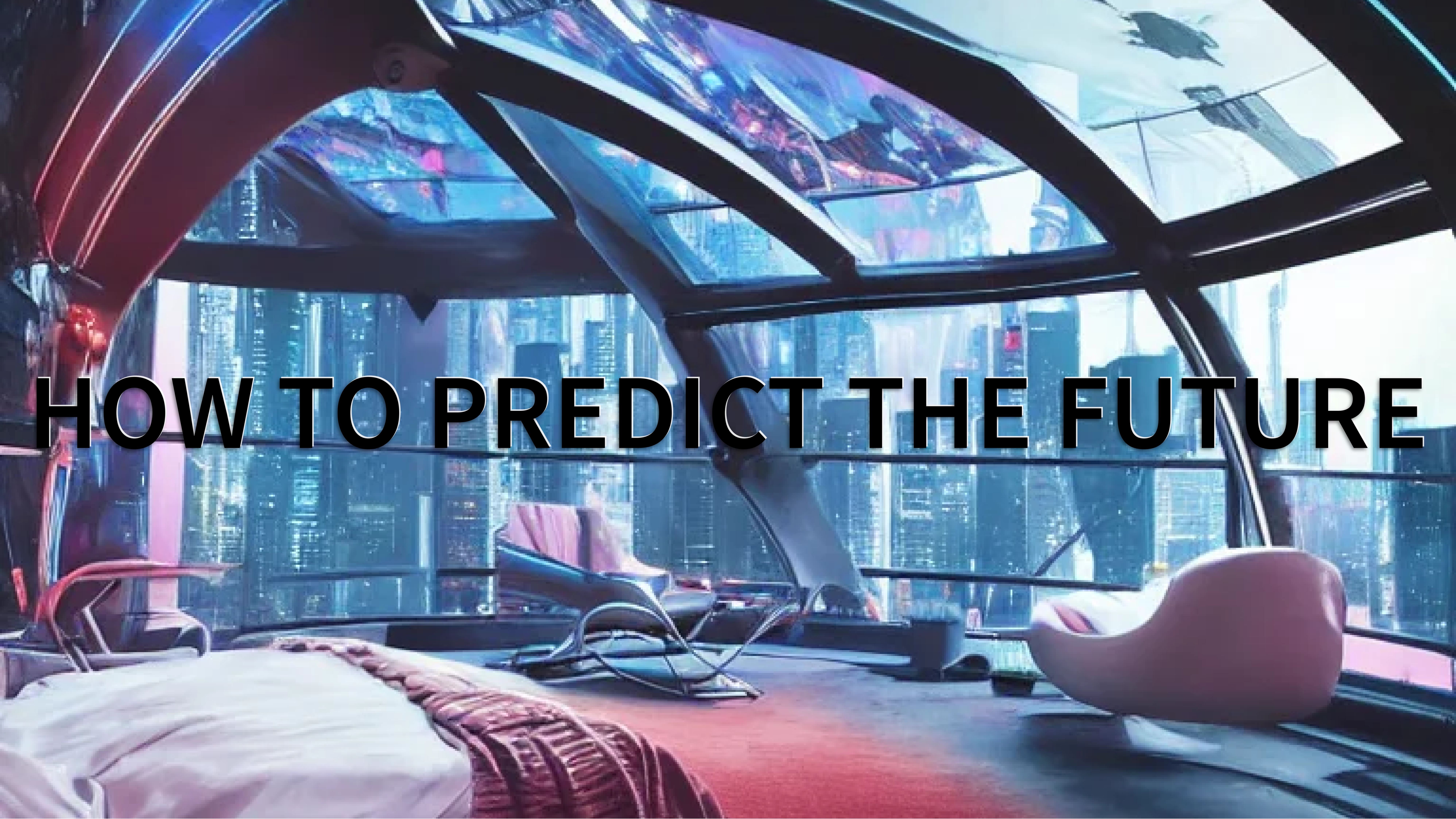
LTO

```c
int main(int argc, char **argv) {
    int array[5] = {1, 0, 4, 5, 10};
    int result = sum_array(array, 5);
    printf("result: %d\n", result);
    foo(argc > 1);
    return 0;
}
```

```
$ zig build -Drelease-fast
$ objdump -d zig-out/bin/foobar -Mintel | vim -
```

```
0000000000216380 <main>:
  216380: 53                      push   rbx
  216381: 89 fb                   mov    ebx,edi
  216383: bf 19 26 20 00          mov    edi,0x202619
  216388: be 14 00 00 00          mov    esi,0x14
  21638d: 31 c0                   xor    eax,eax
  21638f: e8 4c 00 00 00          call   2163e0 <printf@plt>
  216394: 83 fb 02                cmp    ebx,0x2
  216397: 7c 0a                   jl     2163a3 <main+0x23>
  216399: bf 60 0d 20 00          mov    edi,0x200d60
  21639e: e8 fd e8 fe ff          call   204ca0 <debug.dumpCurrentStackTrace>
  2163a3: 31 c0                   xor    eax,eax
  2163a5: 5b                      pop    rbx
  2163a6: c3                      ret
```

HOW TO PREDICT THE FUTURE



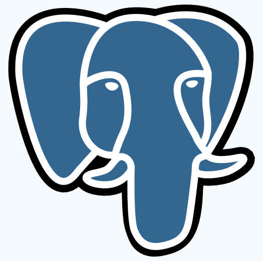
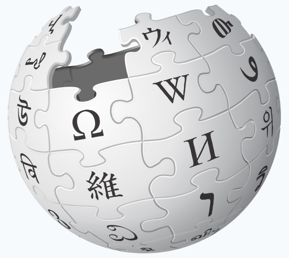
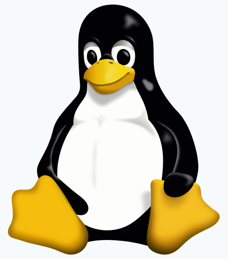
Let's play a game.

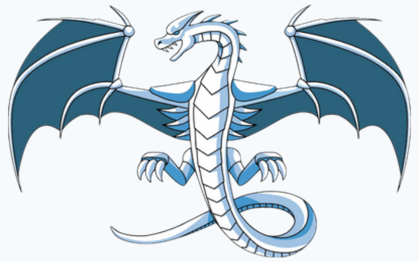
CATEGORY A	CATEGORY B

Speaker notes

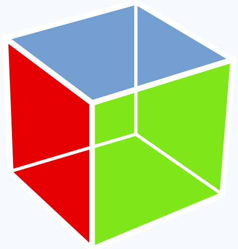
Can anyone guess what the two categories are?

CATEGORY A



StableDiffusion



blender®

CATEGORY B

mongoDB®

Google

macOS

SQLite

OpenAI

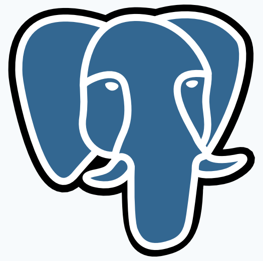
Qt

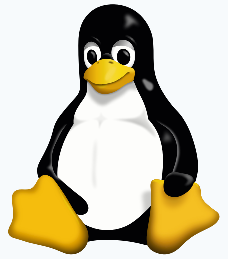
MySQL™

Speaker notes

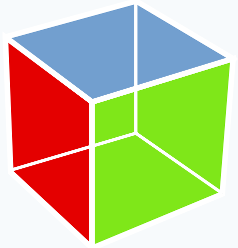
Can anyone guess what the two categories are?

NON-PROFITS





StableDiffusion



blender®

FOR-PROFITS

mongoDB®

Google

macOS

SQLite

OpenAI

Qt

MySQL™

Speaker notes

The red, green, blue box is GTK

OpenAI, the company behind DALL-E, is in fact a closed-source, for-profit company.

Anyone surprised about SQLite? We'll come back to that one.

Non-Profits vs VC-backed Startups

Non-Profits vs VC-backed Startups

- Profit = Revenue - Expenses

Non-Profits vs VC-backed Startups

- Profit = Revenue - Expenses
- In a for-profit company, profit goes to the owners.

Non-Profits vs VC-backed Startups

- Profit = Revenue - Expenses
- In a for-profit company, profit goes to the owners.

Success is defined by the **owners gaining wealth**.

Non-Profits vs VC-backed Startups

Non-Profits vs VC-backed Startups

- At a non-profit, excess revenue is reinvested into the mission statement.

Non-Profits vs VC-backed Startups

- At a non-profit, excess revenue is reinvested into the mission statement.

Success is defined by **fulfillment of the mission**.

VC BACKED STARTUPS

VC BACKED STARTUPS

It's very easy to predict the future of VC-backed startups.

VC BACKED STARTUPS

It's very easy to predict the future of VC-backed startups.

VC BACKED STARTUPS

It's very easy to predict the future of VC-backed startups.

Year 0

They spend VC money to make a seductive, unsustainable product that customers flock to.

VC BACKED STARTUPS

It's very easy to predict the future of VC-backed startups.

Year 5

The VCs squeeze their grip, the startup starts going to shit. Employees and customers suffer as they prepare their "exit strategy".

VC BACKED STARTUPS

It's very easy to predict the future of VC-backed startups.

Year 7

Roll a d100. 99/100 they get bought by a large corporation as a talent acquisition. The company effectively dies, leaving customers stranded, or awkwardly absorbed by the corporation.

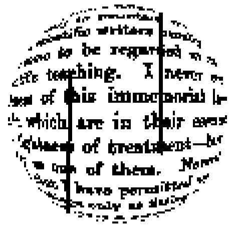
VC BACKED STARTUPS

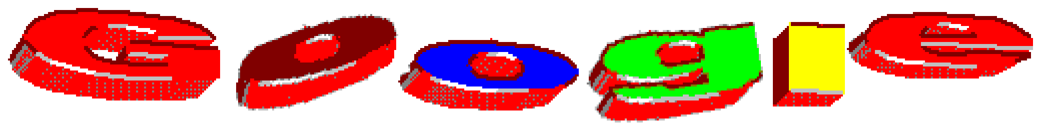
It's very easy to predict the future of VC-backed startups.

Year 20

1/100 they hockey stick. Twenty years pass and they are a large corporation.

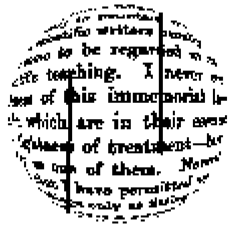
Recognize these?



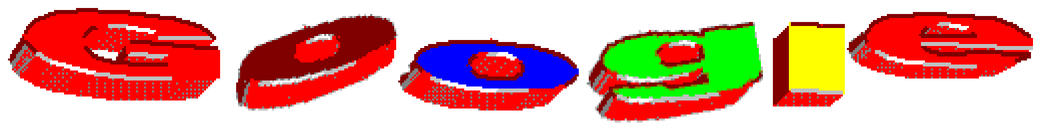


source: Wikipedia

Recognize these?



Wikipedia
501(c)(3) Non-Profit



Google
Limited Liability Company

source: Wikipedia

Speaker notes

The wikipedia logo is from year 2000.

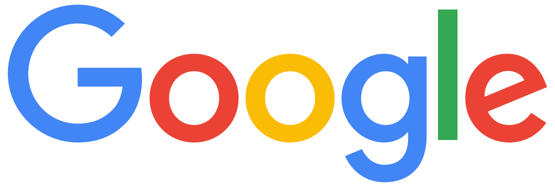
The google logo is from year 1997.

These were contemporary early web websites. Both highly regarded.

Recognize these?

Wikipedia
Still a 501(c)(3) Non-Profit

Still awesome!



Google
Publicly Traded since 2004

Evil Large Corporation

source: Wikipedia

Speaker notes

how many people in here used FitBit?

congratulations, google has your data whether you like it or not!

shady gym taking my money. simple bank was great until it wasn't. stopped supporting check sending

PRIVATELY OWNED BUSINESSES

Speaker notes

Sometimes it stays a family business, but that depends on brittle factors.

SQlite has a Consortium that companies can pay to join to get better support. This seems like an excellent business model, and has been working well so far.

SQLite is well-regarded. But we have yet to see the owners of Hipp, Wyrick & Company, Inc. reach retirement age. I wouldn't get surprised if the company gets sold.

EPIC acquired RAD

PRIVATELY OWNED BUSINESSES

- A bit more stable.

PRIVATELY OWNED BUSINESSES

- A bit more stable.
- But probably the owner wants to retire and cash out.

PRIVATELY OWNED BUSINESSES

- A bit more stable.
- But probably the owner wants to retire and cash out.
- Case Study: Progressive Roofing

PRIVATELY OWNED BUSINESSES

- A bit more stable.
- But probably the owner wants to retire and cash out.
- Case Study: Progressive Roofing
- Case Study: RAD Game Tools

PRIVATELY OWNED BUSINESSES

- A bit more stable.
- But probably the owner wants to retire and cash out.
- Case Study: Progressive Roofing
- Case Study: RAD Game Tools
- Case Study: SQLite

PRIVATELY OWNED BUSINESSES

- A bit more stable.
- But probably the owner wants to retire and cash out.
- Case Study: Progressive Roofing
- Case Study: RAD Game Tools
- Case Study: SQLite
- From the customer’s perspective, you have a bit more time to enjoy before the inevitable outcome.

ZIG SOFTWARE FOUNDATION

Speaker notes

we don't have any runway to worry about

we have no VCs wanting a return on their investment

we can focus on our mission

I'm not the boss - the board of directors is. so you, the customer, know that ZSF will still operate according to the same mission when I'm gone.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.
- We have released a useful, working product.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.
- We have released a useful, working product.
- We have happy, talented, self-directed staff.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.
- We have released a useful, working product.
- We have happy, talented, self-directed staff.
- We have an ambitious roadmap.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.
- We have released a useful, working product.
- We have happy, talented, self-directed staff.
- We have an ambitious roadmap.
- We have a secure future.

ZIG SOFTWARE FOUNDATION

- We have achieved financial stability.
- We have released a useful, working product.
- We have happy, talented, self-directed staff.
- We have an ambitious roadmap.
- We have a secure future.

We have already succeeded!

ZIG IN ACTION

ZIG IN ACTION

- Zig is general-purpose, which means it gives you the tools to generate the best possible machine code for the target, whether it is hardware or a virtual machine.

ZIG IN ACTION

- Zig is general-purpose, which means it gives you the tools to generate the best possible machine code for the target, whether it is hardware or a virtual machine.
- This makes it applicable to anything resembling a Vonn Neumann machine.

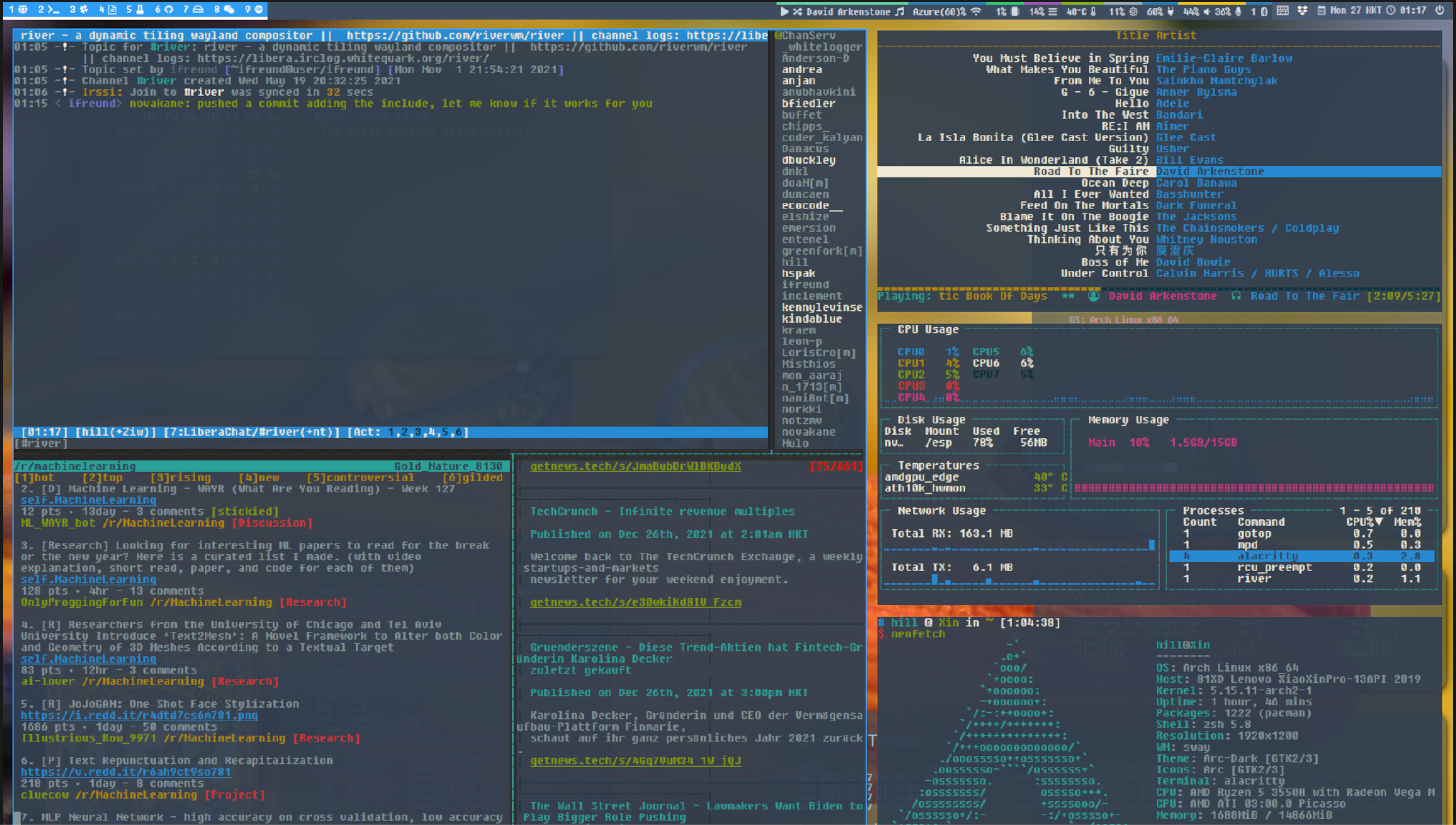
ZIG IN ACTION

- Zig is general-purpose, which means it gives you the tools to generate the best possible machine code for the target, whether it is hardware or a virtual machine.
- This makes it applicable to anything resembling a Vonn Neumann machine.
- But there are some cases where Zig truly excels thanks to its conservative language design choices.

ZIG IN ACTION

- Low-Level Infrastructure
- Libraries to be used by higher level languages
- High-performance applications
  - Real-Time Digital Audio Processing
  - Video Games
  - Databases
- Resource-Constrained Environments
  - WebAssembly
  - Operating Systems
  - Embedded development

RIVER WINDOW MANAGER



```
1  2 >_  3  4  5  6  7  8  9        ▶ David Arkenstone ♫  Azure(60)%  1%  14% ≡  40°C  11%  60%  44%  36%  1      Mon 27 HKT  01:17
```

```
river - a dynamic tiling wayland compositor ||  https://github.com/riverwm/river || channel logs: https://libe
01:05 -!- Topic for #river: river - a dynamic tiling wayland compositor ||  https://github.com/riverwm/river
          || channel logs: https://libera.irclog.whitequark.org/river/
01:05 -!- Topic set by ifreund [~ifreund@user/ifreund] [Mon Nov  1 21:54:21 2021]
01:05 -!- Channel #river created Wed May 19 20:32:25 2021
01:06 -!- Irssi: Join to #river was synced in 32 secs
01:15 < ifreund> novakane: pushed a commit adding the include, let me know if it works for you

 [01:17] [hill(+Ziw)] [7:LiberaChat/#river(+nt)] [Act: 1,2,3,4,5,6]
[#river]
```

```
@ChanServ
 _whitelogger
 Anderson-D
 andrea
 anjan
 anubhavkini
 bfiedler
 buffet
 chipps_
 coder_kalyan
 Danacus
 dbuckley
 dnkl
 doaN[m]
 duncaen
 ecocode__
 elshize
 emersion
 entenel
 greenfork[m]
 hill
 hspak
 ifreund
 inclement
 kennylevinse
 kindablue
 kraem
 leon-p
 LorisCro[m]
 Misthios
 mon_aaraj
 n_1713[m]
 nani8ot[m]
 norkki
 notzmv
 novakane
 Nulo
```

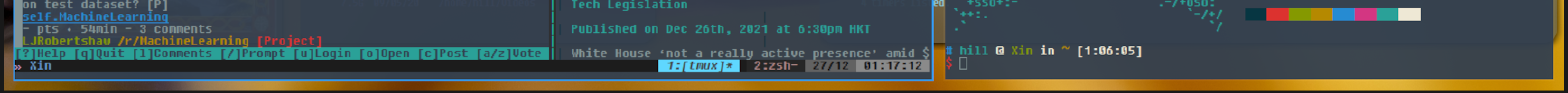
```
                                      Title Artist
-----------------------------------------------------------------------------
          You Must Believe in Spring Emilie-Claire Barlow
           What Makes You Beautiful The Piano Guys
                     From Me To You Sainkho Namtchylak
                    G - 6 - Gigue Anner Bylsma
                             Hello Adele
                     Into The West Bandari
                           RE:I AM Aimer
 La Isla Bonita (Glee Cast Version) Glee Cast
                            Guilty Usher
       Alice In Wonderland (Take 2) Bill Evans
                 Road To The Faire David Arkenstone
                        Ocean Deep Carol Banawa
                 All I Ever Wanted Basshunter
              Feed On The Mortals Dark Funeral
            Blame It On The Boogie The Jacksons
          Something Just Like This The Chainsmokers / Coldplay
              Thinking About You Whitney Houston
                          只有为你 庾澄庆
                        Boss of Me David Bowie
                     Under Control Calvin Harris / HURTS / Alesso
-----------------------------------------------------------------------------
Playing: tic Book Of Days  **   David Arkenstone   Road To The Fair [2:09/5:27]
```

```
                         OS: Arch Linux x86 64
┌─ CPU Usage ────────────────────────────────────────────────────────────┐
│                                                                        │
│ CPU0    1%  CPU5    6%                                                 │
│ CPU1    4%  CPU6    6%                                                 │
│ CPU2    5%  CPU7    5%                                                 │
│ CPU3    0%                                                             │
│ CPU4    0%                                                             │
└────────────────────────────────────────────────────────────────────────┘
┌─ Disk Usage ──────────────┐ ┌─ Memory Usage ────────────────────────────┐
│Disk  Mount  Used  Free    │ │                                           │
│nv…   /esp   78%    56MB   │ │ Main   10%   1.5GB/15GB                   │
└───────────────────────────┘ │                                           │
┌─ Temperatures ────────────┐ │                                           │
│amdgpu_edge         40° C  │ │                                           │
│ath10k_hwmon        33° C  │ │###########################################│
└───────────────────────────┘ └───────────────────────────────────────────┘
┌─ Network Usage ─────────────────────┐ ┌─ Processes ─────────── 1 - 5 of 210 ─┐
│                                     │ │ Count   Command            CPU%▼  Mem% │
│ Total RX: 163.1 MB                  │ │ 1       gotop              0.7    0.0  │
│                                     │ │ 1       mpd                0.5    0.3  │
│                                     │ │ 4       alacritty          0.3    2.8  │
│ Total TX:   6.1 MB                  │ │ 1       rcu_preempt        0.2    0.0  │
│                                     │ │ 1       river              0.2    1.1  │
└─────────────────────────────────────┘ └──────────────────────────────────────┘
```

```
/r/machinelearning                                              Gold Nature 8130
[1]hot    [2]top    [3]rising    [4]new    [5]controversial    [6]gilded
 2. [D] Machine Learning - WAYR (What Are You Reading) - Week 127
 self.MachineLearning
 12 pts • 13day - 3 comments [stickied]
 ML_WAYR_bot /r/MachineLearning [Discussion]

 3. [Research] Looking for interesting ML papers to read for the break
 or the new year? Here is a curated list I made. (with video
 explanation, short read, paper, and code for each of them)
 self.MachineLearning
 128 pts • 4hr - 13 comments
 OnlyProggingForFun /r/MachineLearning [Research]

 4. [R] Researchers from the University of Chicago and Tel Aviv
 University Introduce ‘Text2Mesh’: A Novel Framework to Alter both Color
 and Geometry of 3D Meshes According to a Textual Target
 self.MachineLearning
 83 pts • 12hr - 3 comments
 ai-lover /r/MachineLearning [Research]

 5. [R] JoJoGAN: One Shot Face Stylization
 https://i.redd.it/r4dtd7cs6m781.png
 1686 pts • 1day - 50 comments
 Illustrious_Row_9971 /r/MachineLearning [Research]

 6. [P] Text Repunctuation and Recapitalization
 https://v.redd.it/r6ah9ct9so781
 218 pts • 1day - 8 comments
 cluecow /r/MachineLearning [Project]

 7. MLP Neural Network - high accuracy on cross validation, low accuracy
```

```
| getnews.tech/s/JmaBubDrW1BKBydX                        [75/661]
|
|------------------------------------------------------------------
-----------------------------------|
| TechCrunch - Infinite revenue multiples
|
| Published on Dec 26th, 2021 at 2:01am HKT
|
| Welcome back to The TechCrunch Exchange, a weekly
 startups-and-markets
| newsletter for your weekend enjoyment.
|
| getnews.tech/s/e30wkiKd8IV_Fzcm
|
|------------------------------------------------------------------
-----------------------------------|
| Gruenderszene - Diese Trend-Aktien hat Fintech-Gr
ünderin Karolina Decker
| zuletzt gekauft
|
| Published on Dec 26th, 2021 at 3:00pm HKT
|
| Karolina Decker, Gründerin und CEO der Vermögensa
ufbau-Plattform Finmarie,
| schaut auf ihr ganz persönliches Jahr 2021 zurück
.
| getnews.tech/s/4Gq7VuM34_1W_jQJ
|
|------------------------------------------------------------------
-----------------------------------|
| The Wall Street Journal - Lawmakers Want Biden to
 Play Bigger Role Pushing
```

```
# hill @ Xin in ~ [1:04:38]
$ neofetch
                   -`
                  .o+`                   hill@Xin
                 `ooo/                   --------
                `+oooo:                  OS: Arch Linux x86_64
               `+oooooo:                 Host: 81XD Lenovo XiaoXinPro-13API 2019
               -+oooooo+:                Kernel: 5.15.11-arch2-1
             `/:-:++oooo+:               Uptime: 1 hour, 46 mins
            `/++++/+++++++:              Packages: 1222 (pacman)
           `/++++++++++++++:             Shell: zsh 5.8
          `/+++ooooooooooooo/`           Resolution: 1920x1200
         ./ooosssso++osssssso+`          WM: sway
        .oossssso-````/ossssss+`         Theme: Arc-Dark [GTK2/3]
       -osssssso.      :ssssssso.        Icons: Arc [GTK2/3]
      :osssssss/        osssso+++.       Terminal: alacritty
     /ossssssss/        +ssssooo/-       CPU: AMD Ryzen 5 3550H with Radeon Vega M
   `/ossssso+/:-        -:/+osssso+-     GPU: AMD ATI 03:00.0 Picasso
                                         Memory: 1688MiB / 14866MiB
```



```
on test dataset? [P]
self.MachineLearning
- pts • 54min - 3 comments
LJRobertshaw /r/MachineLearning [Project]
[?]Help [q]Quit [l]Comments [/]Prompt [u]Login [o]Open [c]Post [a/z]Vote
» Xin
```

```
Tech Legislation

Published on Dec 26th, 2021 at 6:30pm HKT

White House ‘not a really active presence’ amid $
1:[tmux]*  2:zsh-  27/12  01:17:12
```

```
# hill @ Xin in ~ [1:06:05]
$ 
```

Speaker notes

River Window Manager

Bun

Docs Discord GitHub

Bun is a fast all-in-one JavaScript runtime

Bundle, transpile, install and run JavaScript & TypeScript projects — all in Bun. Bun is a new JavaScript runtime with a native bundler, transpiler, task runner and npm client built-in.

Install Bun CLI v0.1.8 (beta)

macOS x64 & Silicon, Linux x64, Windows Subsystem for Linux

```
curl https://bun.sh/install | bash
```

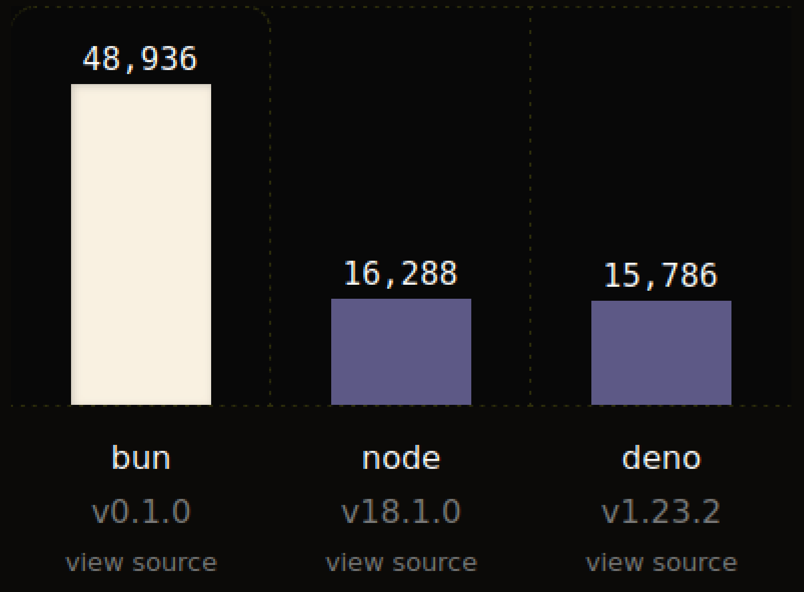
copy

Show script source

`Bun.serve` `bun:sqlite` `bun:ffi`

Server-side rendering React

HTTP requests per second (Linux AMD64)



Why is Bun fast?

An enormous amount of time spent profiling, benchmarking and optimizing things. The answer is different for every part of Bun, but one general theme: ZIG's low-level control over memory and lack of hidden control flow makes it much simpler to write fast software. Sponsor the Zig Software Foundation.

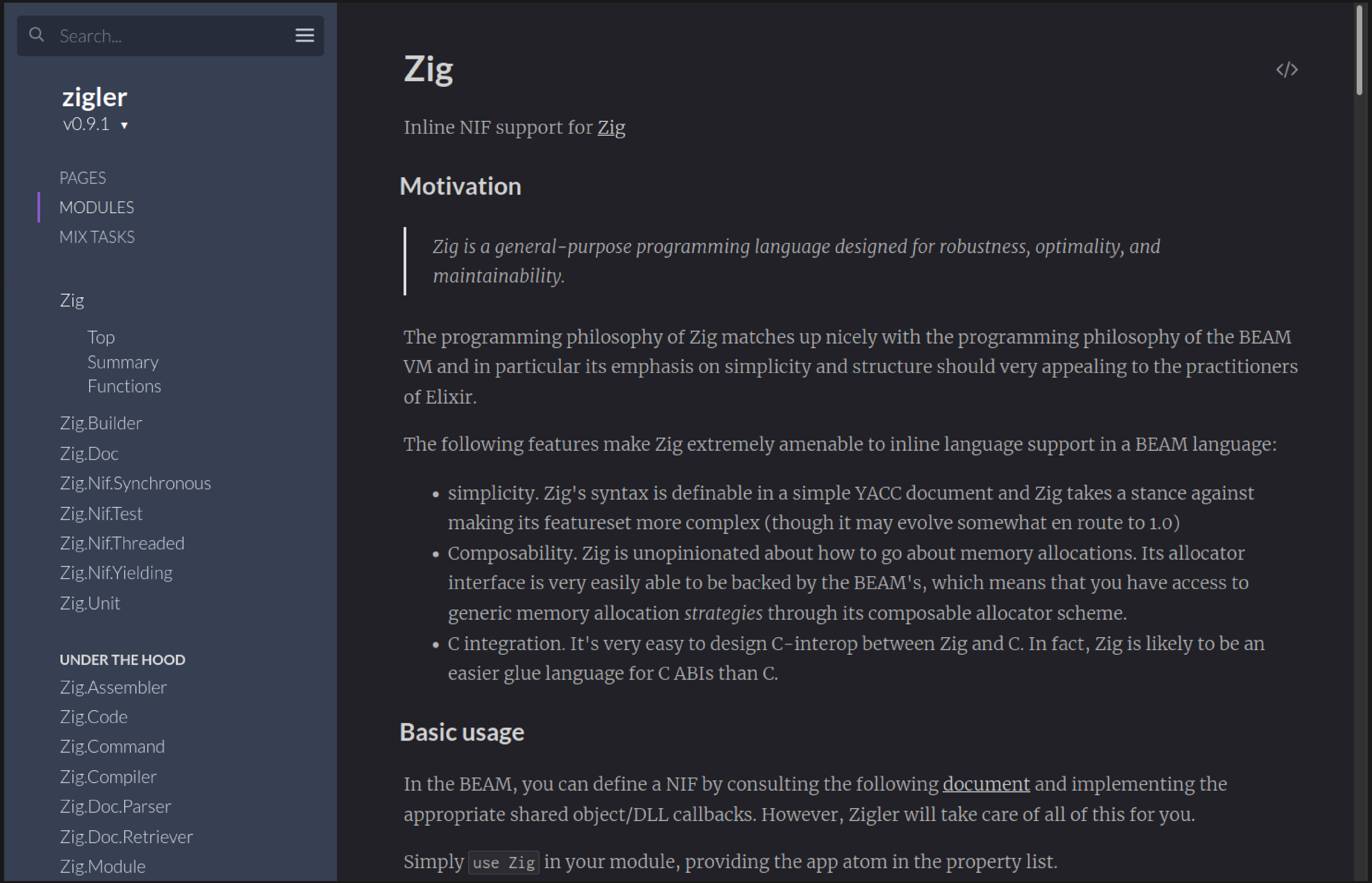
Speaker notes

Very cool of Jarred to put this section on the homepage for Bun. Thanks!

ZIG IN ACTION

- Low-Level Infrastructure
- **Libraries to be used by higher level languages**
- High-performance applications
  - Real-Time Digital Audio Processing
  - Video Games
  - Databases
- Resource-Constrained Environments
  - WebAssembly
  - Operating Systems
  - Embedded development

ZIGLER



Zig

Inline NIF support for Zig

Motivation

> *Zig is a general-purpose programming language designed for robustness, optimality, and maintainability.*

The programming philosophy of Zig matches up nicely with the programming philosophy of the BEAM VM and in particular its emphasis on simplicity and structure should very appealing to the practitioners of Elixir.

The following features make Zig extremely amenable to inline language support in a BEAM language:

- simplicity. Zig's syntax is definable in a simple YACC document and Zig takes a stance against making its featureset more complex (though it may evolve somewhat en route to 1.0)
- Composability. Zig is unopinionated about how to go about memory allocations. Its allocator interface is very easily able to be backed by the BEAM's, which means that you have access to generic memory allocation *strategies* through its composable allocator scheme.
- C integration. It's very easy to design C-interop between Zig and C. In fact, Zig is likely to be an easier glue language for C ABIs than C.

Basic usage

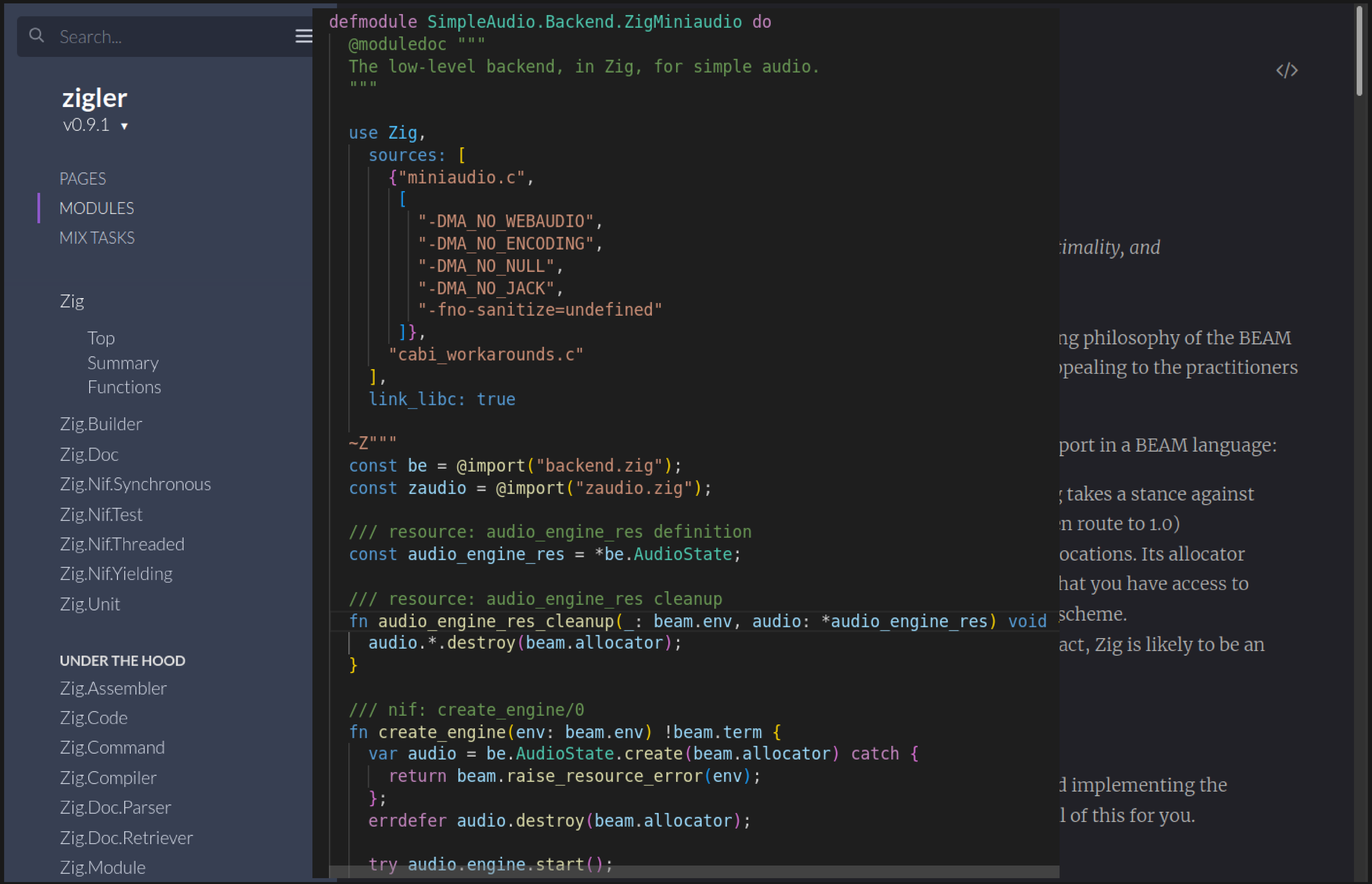
In the BEAM, you can define a NIF by consulting the following document and implementing the appropriate shared object/DLL callbacks. However, Zigler will take care of all of this for you.

Simply `use Zig` in your module, providing the app atom in the property list.

Speaker notes

The level of degree of smooth integration that Isaac accomplished is incredible. Highly recommended to check this project out if you're an Erlang user.

ZIGLER



```
defmodule SimpleAudio.Backend.ZigMiniaudio do
  @moduledoc """
  The low-level backend, in Zig, for simple audio.
  """

  use Zig,
    sources: [
      {"miniaudio.c",
       [
         "-DMA_NO_WEBAUDIO",
         "-DMA_NO_ENCODING",
         "-DMA_NO_NULL",
         "-DMA_NO_JACK",
         "-fno-sanitize=undefined"
       ]},
      "cabi_workarounds.c"
    ],
    link_libc: true

  ~Z"""
  const be = @import("backend.zig");
  const zaudio = @import("zaudio.zig");

  /// resource: audio_engine_res definition
  const audio_engine_res = *be.AudioState;

  /// resource: audio_engine_res cleanup
  fn audio_engine_res_cleanup(_: beam.env, audio: *audio_engine_res) void
    audio.*.destroy(beam.allocator);
  }

  /// nif: create_engine/0
  fn create_engine(env: beam.env) !beam.term {
    var audio = be.AudioState.create(beam.allocator) catch {
      return beam.raise_resource_error(env);
    };
    errdefer audio.destroy(beam.allocator);

    try audio.engine.start();
```

:imality, and

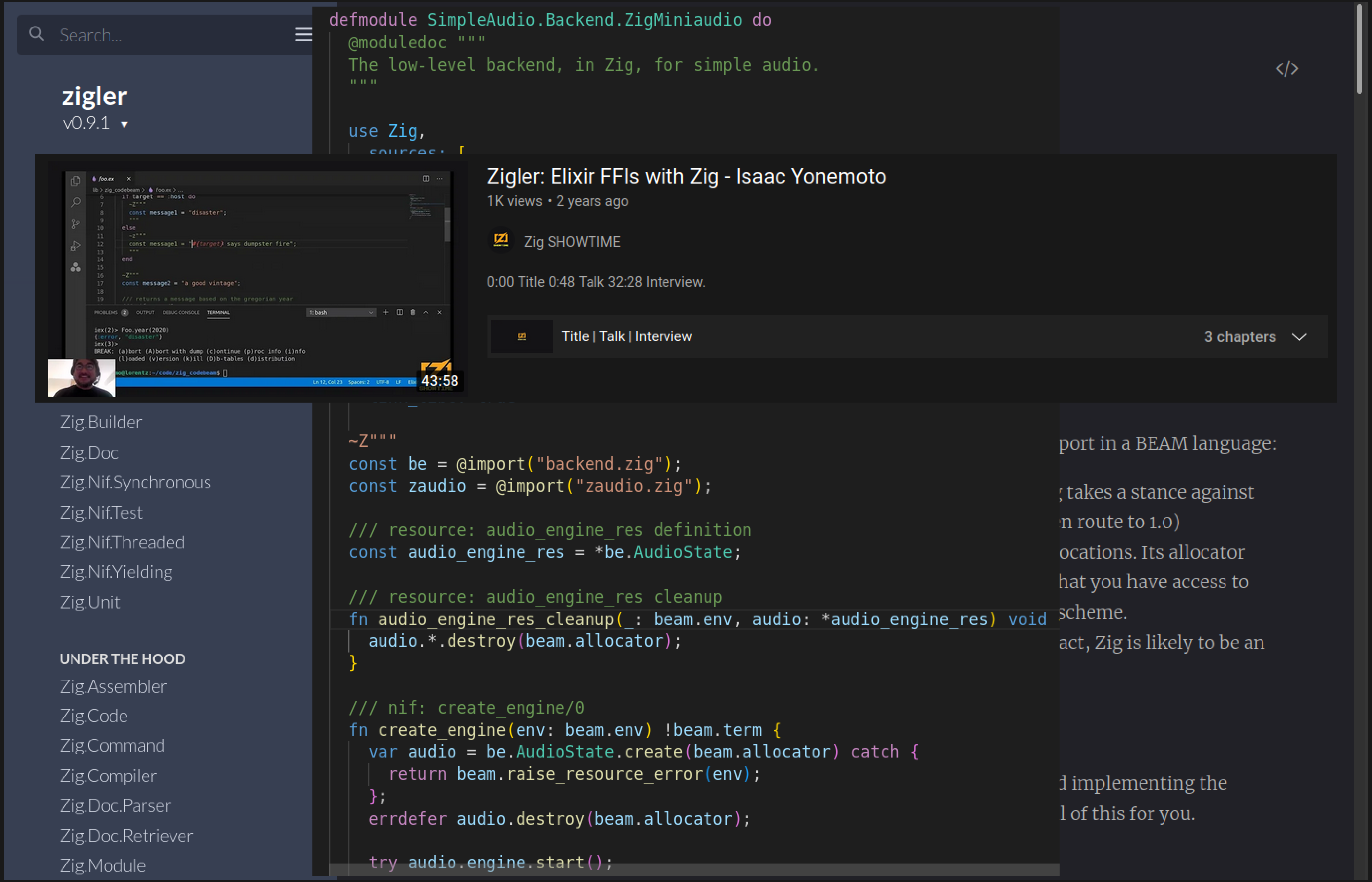
ng philosophy of the BEAM
)pealing to the practitioners

port in a BEAM language:

; takes a stance against
n route to 1.0)
ocations. Its allocator
hat you have access to
scheme.
act, Zig is likely to be an

d implementing the
l of this for you.

ZIGLER



Search...

zigler
v0.9.1 ▾

Zig.Builder
Zig.Doc
Zig.Nif.Synchronous
Zig.Nif.Test
Zig.Nif.Threaded
Zig.Nif.Yielding
Zig.Unit

UNDER THE HOOD
Zig.Assembler
Zig.Code
Zig.Command
Zig.Compiler
Zig.Doc.Parser
Zig.Doc.Retriever
Zig.Module

```
defmodule SimpleAudio.Backend.ZigMiniaudio do
  @moduledoc """
  The low-level backend, in Zig, for simple audio.
  """

  use Zig,
    sources: [

  ~Z"""
  const be = @import("backend.zig");
  const zaudio = @import("zaudio.zig");

  /// resource: audio_engine_res definition
  const audio_engine_res = *be.AudioState;

  /// resource: audio_engine_res cleanup
  fn audio_engine_res_cleanup(_: beam.env, audio: *audio_engine_res) void
    audio.*.destroy(beam.allocator);
  }

  /// nif: create_engine/0
  fn create_engine(env: beam.env) !beam.term {
    var audio = be.AudioState.create(beam.allocator) catch {
      return beam.raise_resource_error(env);
    };
    errdefer audio.destroy(beam.allocator);

    try audio.engine.start();
```

43:58

Zigler: Elixir FFIs with Zig - Isaac Yonemoto

1K views • 2 years ago

Zig SHOWTIME

0:00 Title 0:48 Talk 32:28 Interview.

Title | Talk | Interview

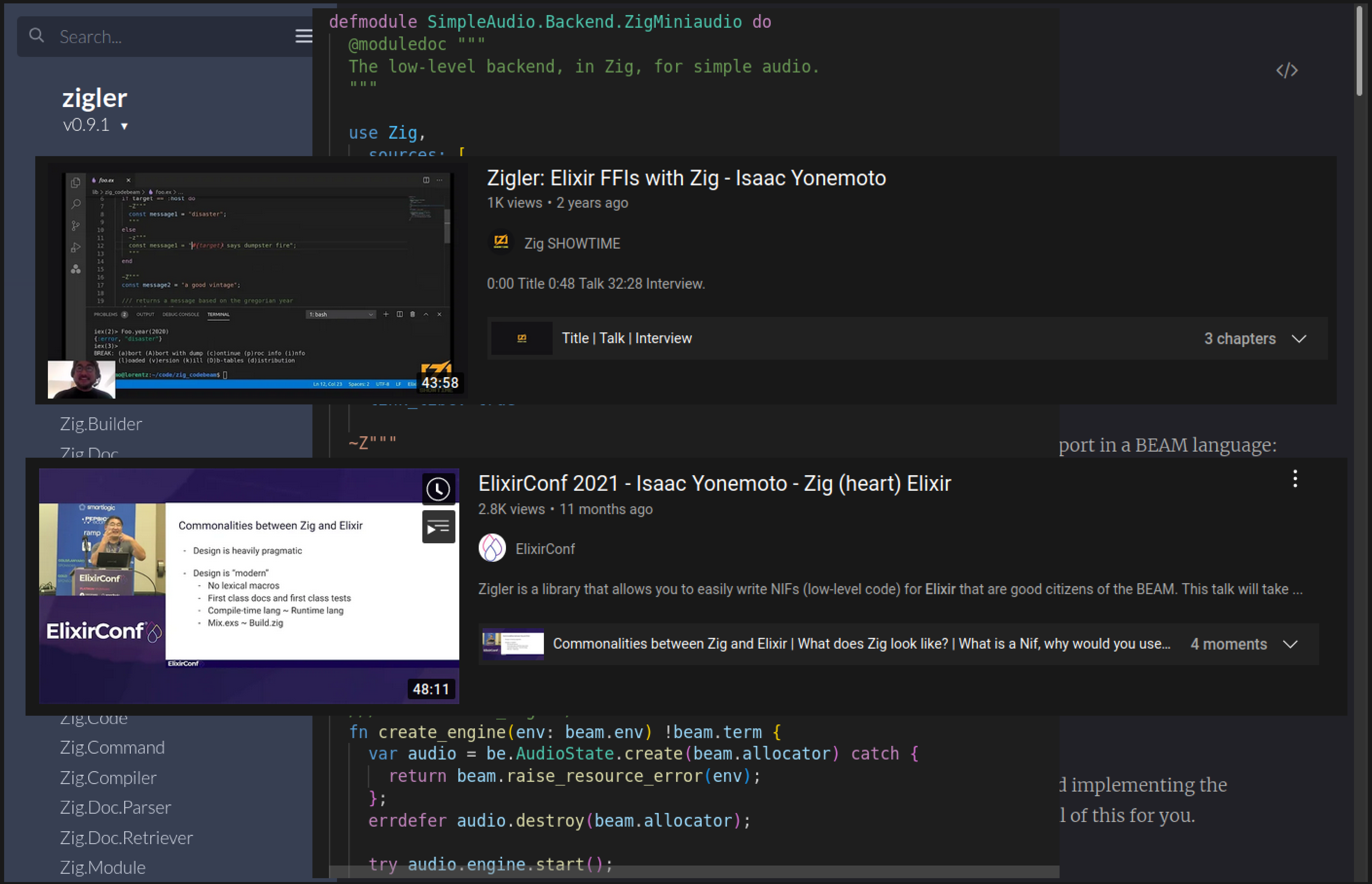
3 chapters

port in a BEAM language:

takes a stance against
n route to 1.0)
ocations. Its allocator
hat you have access to
scheme.
act, Zig is likely to be an

d implementing the
l of this for you.

ZIGLER



Search...

zigler
v0.9.1 ▾

Zig.Builder
Zig.Doc
Zig.Code
Zig.Command
Zig.Compiler
Zig.Doc.Parser
Zig.Doc.Retriever
Zig.Module

```
defmodule SimpleAudio.Backend.ZigMiniaudio do
  @moduledoc """
  The low-level backend, in Zig, for simple audio.
  """

  use Zig,
    sources: [
```

```
  ~Z"""
```

```
  fn create_engine(env: beam.env) !beam.term {
    var audio = be.AudioState.create(beam.allocator) catch {
      return beam.raise_resource_error(env);
    };
    errdefer audio.destroy(beam.allocator);

    try audio.engine.start();
```

port in a BEAM language:

d implementing the
l of this for you.

Zigler: Elixir FFIs with Zig - Isaac Yonemoto
1K views • 2 years ago
Zig SHOWTIME
0:00 Title 0:48 Talk 32:28 Interview.
Title | Talk | Interview — 3 chapters
43:58

```
      if target == :host do
        ~Z"""
        const message1 = "disaster";
        """
      else
        ~Z"""
        const message1 = "#{target} says dumpster fire";
        """
      end

      ~Z"""
      const message2 = "a good vintage";

      /// returns a message based on the gregorian year
```

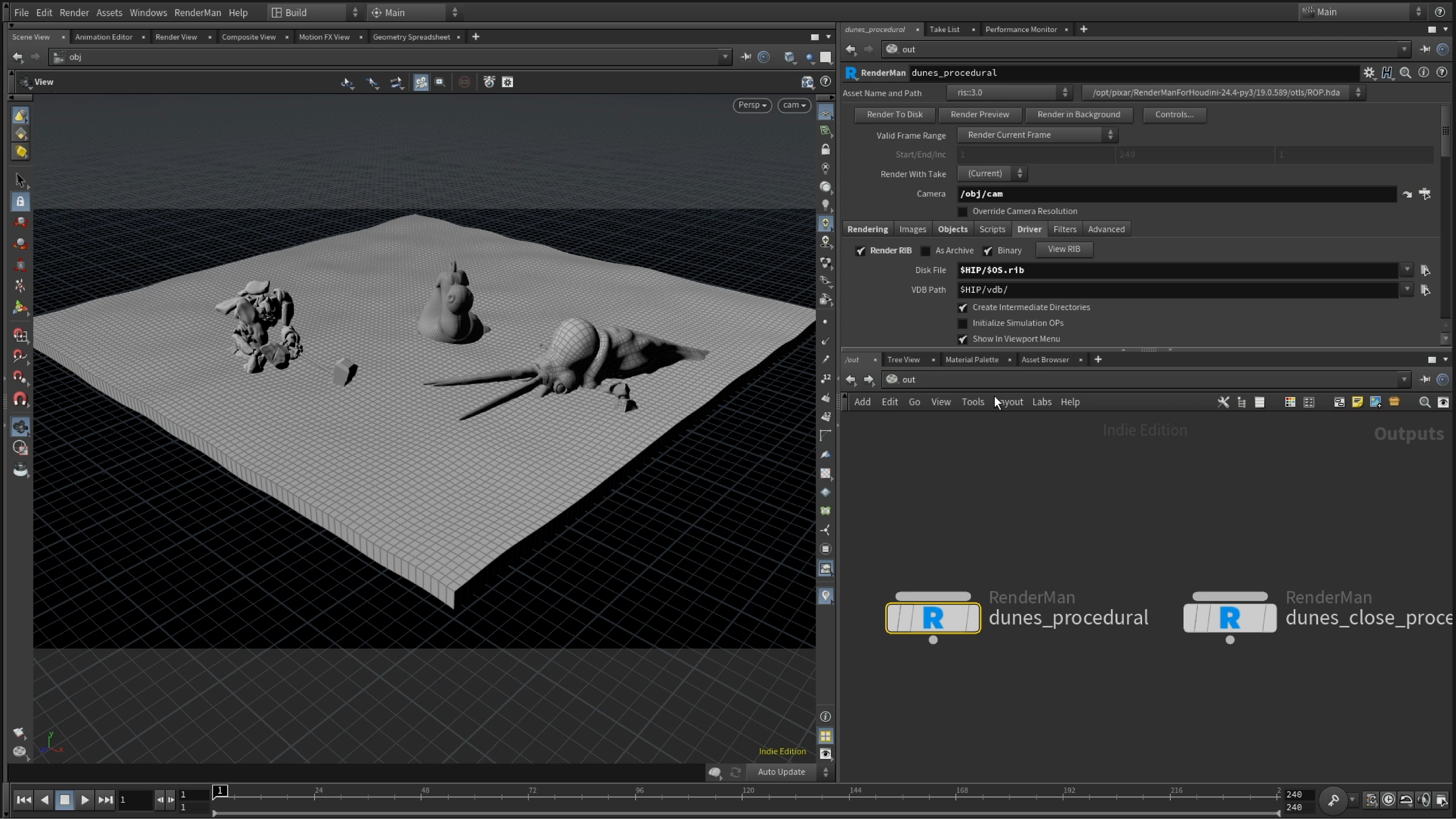
```
iex(2)> Foo.year(2020)
{:error, "disaster"}
iex(3)>
BREAK: (a)bort (A)bort with dump (c)ontinue (p)roc info (i)nfo
       (l)oaded (v)ersion (k)ill (D)b-tables (d)istribution

mo@lorentz:~/code/zig_codebeam$
```

ElixirConf 2021 - Isaac Yonemoto - Zig (heart) Elixir
2.8K views • 11 months ago
ElixirConf
Zigler is a library that allows you to easily write NIFs (low-level code) for **Elixir** that are good citizens of the BEAM. This talk will take ...
Commonalities between Zig and Elixir | What does Zig look like? | What is a Nif, why would you use... — 4 moments
48:11

Commonalities between Zig and Elixir

- Design is heavily pragmatic
- Design is "modern"
  - No lexical macros
  - First class docs and first class tests
  - Compile-time lang ~ Runtime lang
  - Mix.exs ~ Build.zig

VFX PLUGINS



RenderMan dunes_procedural

Asset Name and Path: ris::3.0 /opt/pixar/RenderManForHoudini-24.4-py3/19.0.589/otls/ROP.hda

Valid Frame Range: Render Current Frame

Start/End/Inc: 1 240 1

Render With Take: (Current)

Camera: /obj/cam

□ Override Camera Resolution

☑ Render RIB □ As Archive ☑ Binary

Disk File: $HIP/$OS.rib

VDB Path: $HIP/vdb/

☑ Create Intermediate Directories

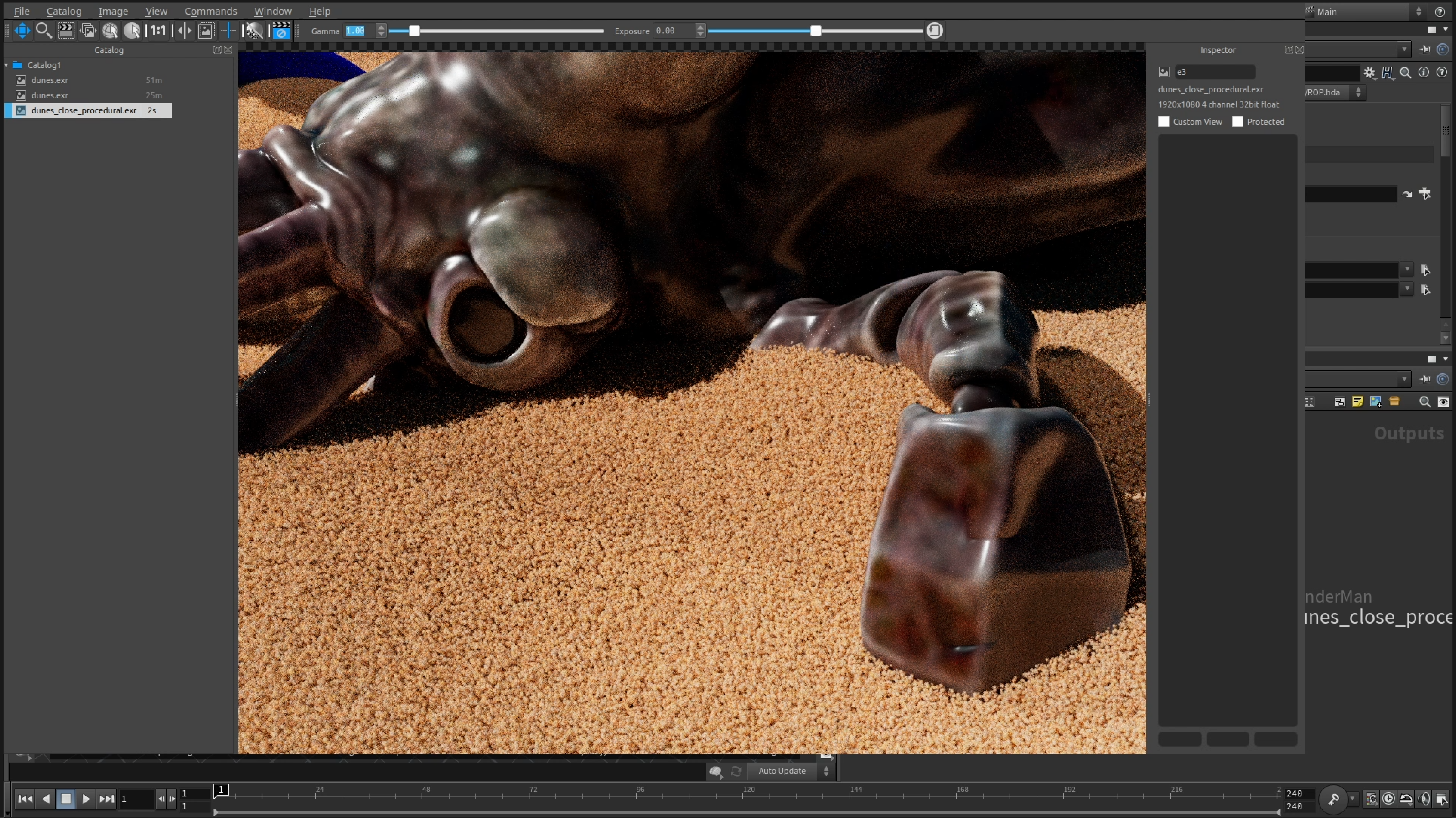
□ Initialize Simulation OPs

☑ Show In Viewport Menu

Speaker notes

Uses procedural generation to save many GiB of bandwidth on the network.

VFX PLUGINS



Gamma 1.00

Exposure 0.00

Catalog1

dunes.exr	51m
dunes.exr	25m
dunes_close_procedural.exr	2s

Inspector

e3

dunes_close_procedural.exr

1920x1080 4 channel 32bit float

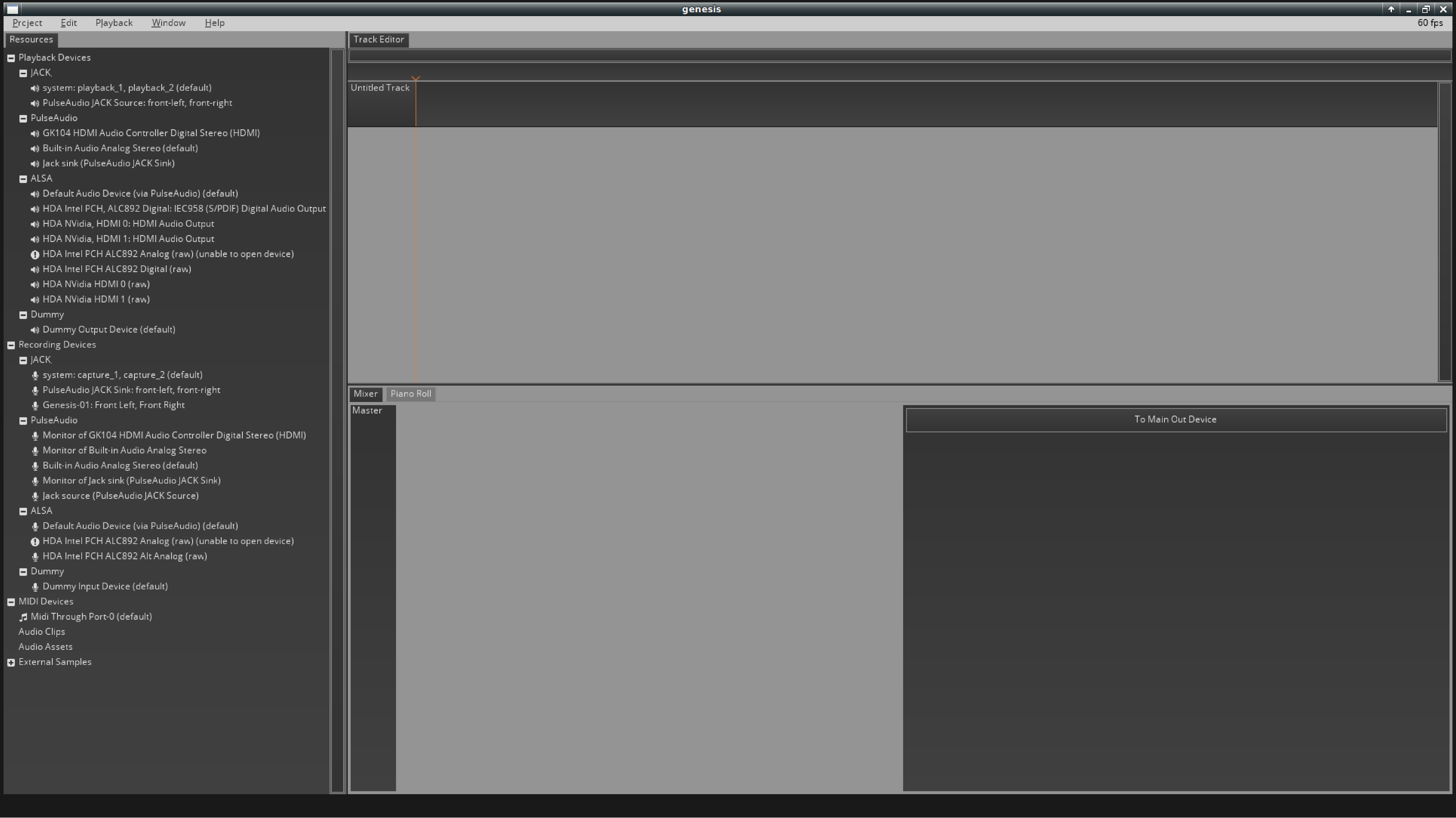
□ Custom View □ Protected

Auto Update

ZIG IN ACTION

- Low-Level Infrastructure
- Libraries to be used by higher level languages
- **High-performance applications**
  - **Real-Time Digital Audio Processing**
  - **Video Games**
  - **Databases**
- Resource-Constrained Environments
  - WebAssembly
  - Operating Systems
  - Embedded development

DIGITAL AUDIO WORKSTATION



- Playback Devices
  - JACK
    - system: playback_1, playback_2 (default)
    - PulseAudio JACK Source: front-left, front-right
  - PulseAudio
    - GK104 HDMI Audio Controller Digital Stereo (HDMI)
    - Built-in Audio Analog Stereo (default)
    - Jack sink (PulseAudio JACK Sink)
  - ALSA
    - Default Audio Device (via PulseAudio) (default)
    - HDA Intel PCH, ALC892 Digital: IEC958 (S/PDIF) Digital Audio Output
    - HDA NVidia, HDMI 0: HDMI Audio Output
    - HDA NVidia, HDMI 1: HDMI Audio Output
    - HDA Intel PCH ALC892 Analog (raw) (unable to open device)
    - HDA Intel PCH ALC892 Digital (raw)
    - HDA NVidia HDMI 0 (raw)
    - HDA NVidia HDMI 1 (raw)
  - Dummy
    - Dummy Output Device (default)
- Recording Devices
  - JACK
    - system: capture_1, capture_2 (default)
    - PulseAudio JACK Sink: front-left, front-right
    - Genesis-01: Front Left, Front Right
  - PulseAudio
    - Monitor of GK104 HDMI Audio Controller Digital Stereo (HDMI)
    - Monitor of Built-in Audio Analog Stereo
    - Built-in Audio Analog Stereo (default)
    - Monitor of Jack sink (PulseAudio JACK Sink)
    - Jack source (PulseAudio JACK Source)
  - ALSA
    - Default Audio Device (via PulseAudio) (default)
    - HDA Intel PCH ALC892 Analog (raw) (unable to open device)
    - HDA Intel PCH ALC892 Alt Analog (raw)
  - Dummy
    - Dummy Input Device (default)
- MIDI Devices
  - Midi Through Port-0 (default)
- Audio Clips
- Audio Assets
- External Samples

Speaker notes

This was my original motivation for working on Zig. I was doing it in C++ but ran into difficult-to-debug undefined behavior. C++ footguns everywhere.

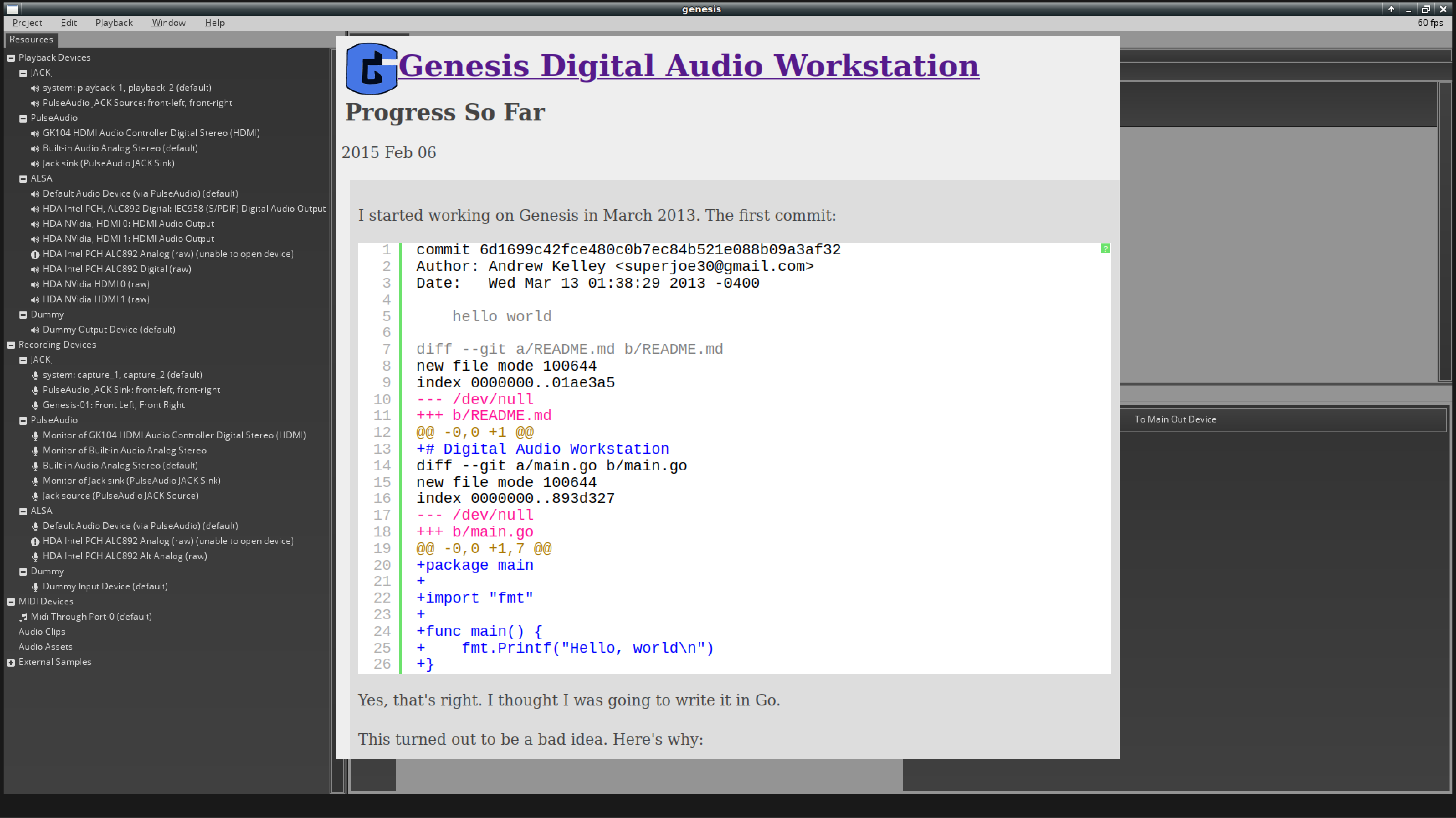
I also tried rewriting this in Rust but found it unproductive.

you can find this, honestly rather unhinged, blog post if you search for it

this was before Rust 1.0 and right now Zig is before 1.0

anyway I don't mean to bash Rust, I just thought this was an amusing old blog post since I didn't know I would be creating Zig at the time

DIGITAL AUDIO WORKSTATION


genesis
Project
Edit
Playback
Window
Help
60 fps
Resources
Playback Devices
JACK
system: playback_1, playback_2 (default)
PulseAudio JACK Source: front-left, front-right
PulseAudio
GK104 HDMI Audio Controller Digital Stereo (HDMI)
Built-in Audio Analog Stereo (default)
Jack sink (PulseAudio JACK Sink)
ALSA
Default Audio Device (via PulseAudio) (default)
HDA Intel PCH, ALC892 Digital: IEC958 (S/PDIF) Digital Audio Output
HDA NVidia, HDMI 0: HDMI Audio Output
HDA NVidia, HDMI 1: HDMI Audio Output
HDA Intel PCH ALC892 Analog (raw) (unable to open device)
HDA Intel PCH ALC892 Digital (raw)
HDA NVidia HDMI 0 (raw)
HDA NVidia HDMI 1 (raw)
Dummy
Dummy Output Device (default)
Recording Devices
JACK
system: capture_1, capture_2 (default)
PulseAudio JACK Sink: front-left, front-right
Genesis-01: Front Left, Front Right
PulseAudio
Monitor of GK104 HDMI Audio Controller Digital Stereo (HDMI)
Monitor of Built-in Audio Analog Stereo
Built-in Audio Analog Stereo (default)
Monitor of Jack sink (PulseAudio JACK Sink)
Jack source (PulseAudio JACK Source)
ALSA
Default Audio Device (via PulseAudio) (default)
HDA Intel PCH ALC892 Analog (raw) (unable to open device)
HDA Intel PCH ALC892 Alt Analog (raw)
Dummy
Dummy Input Device (default)
MIDI Devices
Midi Through Port-0 (default)
Audio Clips
Audio Assets
External Samples
To Main Out Device

Genesis Digital Audio Workstation

Progress So Far

2015 Feb 06

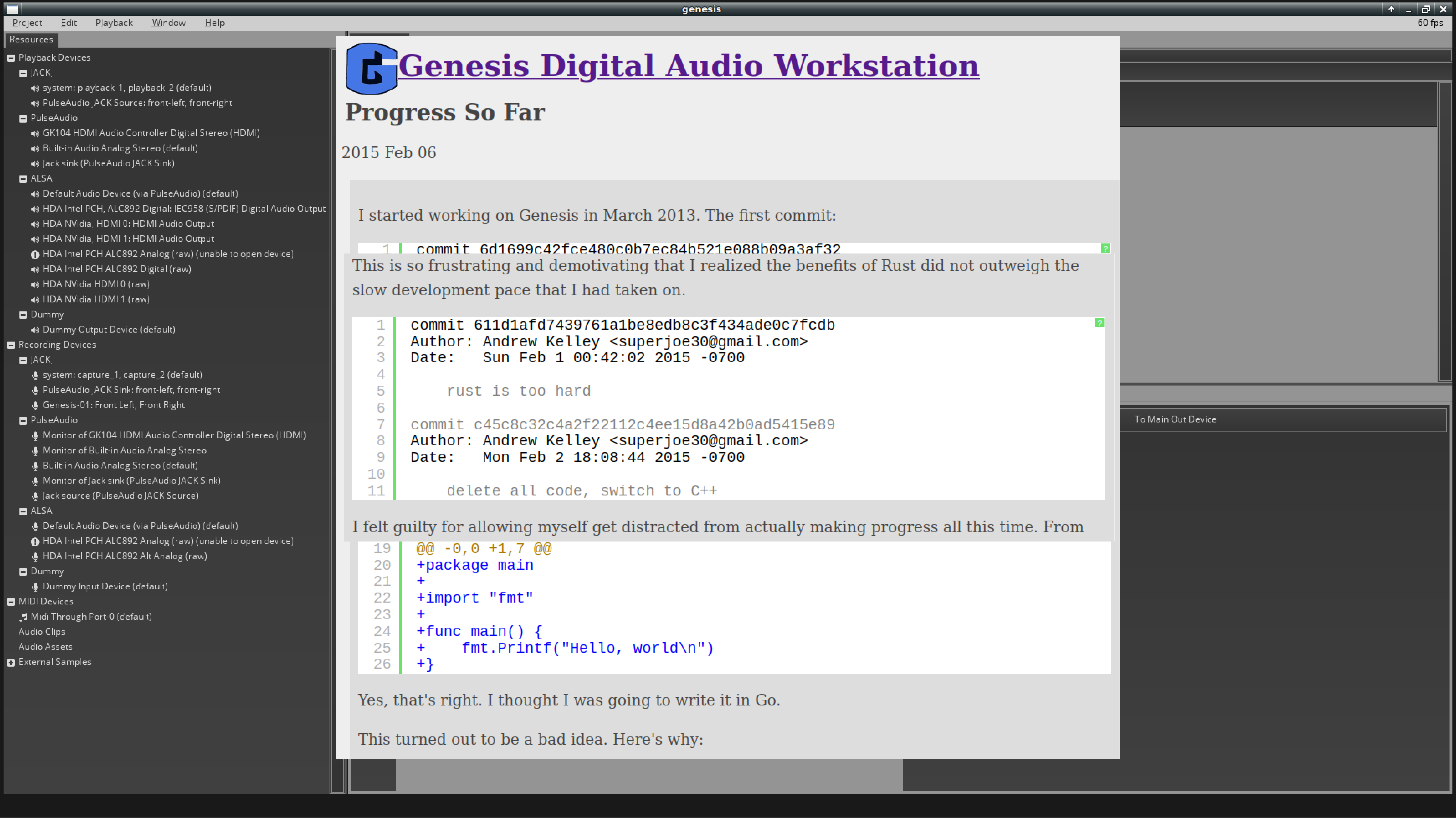
I started working on Genesis in March 2013. The first commit:

```
commit 6d1699c42fce480c0b7ec84b521e088b09a3af32
Author: Andrew Kelley <superjoe30@gmail.com>
Date:   Wed Mar 13 01:38:29 2013 -0400

    hello world

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..01ae3a5
--- /dev/null
+++ b/README.md
@@ -0,0 +1 @@
+# Digital Audio Workstation
diff --git a/main.go b/main.go
new file mode 100644
index 0000000..893d327
--- /dev/null
+++ b/main.go
@@ -0,0 +1,7 @@
+package main
+
+import "fmt"
+
+func main() {
+    fmt.Printf("Hello, world\n")
+}
```

Yes, that's right. I thought I was going to write it in Go.

This turned out to be a bad idea. Here's why:

DIGITAL AUDIO WORKSTATION



genesis
Project
Edit
Playback
Window
Help
60 fps
Resources
Playback Devices
JACK
system: playback_1, playback_2 (default)
PulseAudio JACK Source: front-left, front-right
PulseAudio
GK104 HDMI Audio Controller Digital Stereo (HDMI)
Built-in Audio Analog Stereo (default)
Jack sink (PulseAudio JACK Sink)
ALSA
Default Audio Device (via PulseAudio) (default)
HDA Intel PCH, ALC892 Digital: IEC958 (S/PDIF) Digital Audio Output
HDA NVidia, HDMI 0: HDMI Audio Output
HDA NVidia, HDMI 1: HDMI Audio Output
HDA Intel PCH ALC892 Analog (raw) (unable to open device)
HDA Intel PCH ALC892 Digital (raw)
HDA NVidia HDMI 0 (raw)
HDA NVidia HDMI 1 (raw)
Dummy
Dummy Output Device (default)
Recording Devices
JACK
system: capture_1, capture_2 (default)
PulseAudio JACK Sink: front-left, front-right
Genesis-01: Front Left, Front Right
PulseAudio
Monitor of GK104 HDMI Audio Controller Digital Stereo (HDMI)
Monitor of Built-in Audio Analog Stereo
Built-in Audio Analog Stereo (default)
Monitor of Jack sink (PulseAudio JACK Sink)
Jack source (PulseAudio JACK Source)
ALSA
Default Audio Device (via PulseAudio) (default)
HDA Intel PCH ALC892 Analog (raw) (unable to open device)
HDA Intel PCH ALC892 Alt Analog (raw)
Dummy
Dummy Input Device (default)
MIDI Devices
Midi Through Port-0 (default)
Audio Clips
Audio Assets
External Samples
To Main Out Device

Genesis Digital Audio Workstation

Progress So Far

2015 Feb 06

I started working on Genesis in March 2013. The first commit:

```
commit 6d1699c42fce480c0b7ec84b521e088b09a3af32
```

This is so frustrating and demotivating that I realized the benefits of Rust did not outweigh the slow development pace that I had taken on.

```
commit 611d1afd7439761a1be8edb8c3f434ade0c7fcdb
Author: Andrew Kelley <superjoe30@gmail.com>
Date:   Sun Feb 1 00:42:02 2015 -0700

    rust is too hard

commit c45c8c32c4a2f22112c4ee15d8a42b0ad5415e89
Author: Andrew Kelley <superjoe30@gmail.com>
Date:   Mon Feb 2 18:08:44 2015 -0700

    delete all code, switch to C++
```

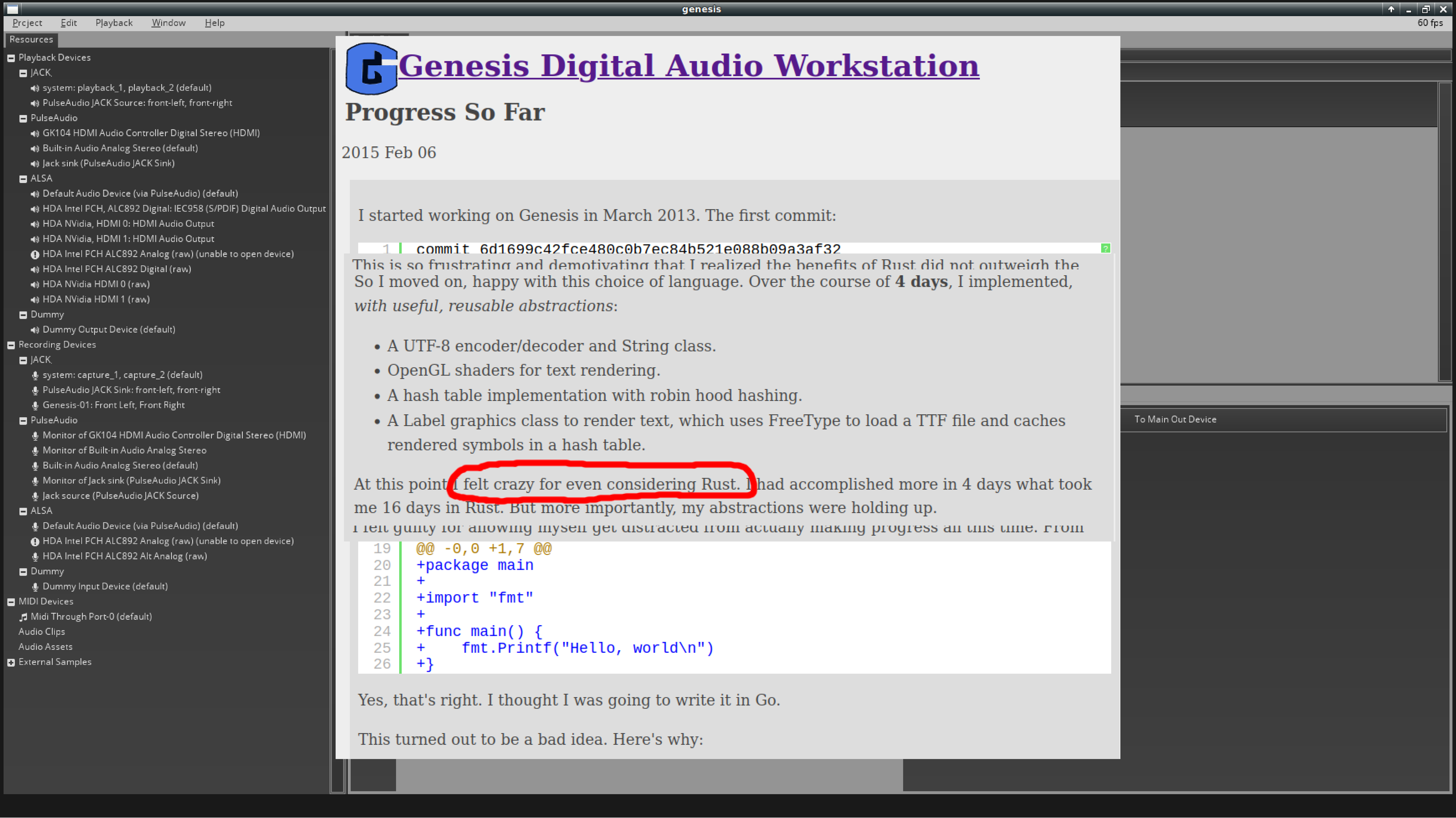
I felt guilty for allowing myself get distracted from actually making progress all this time. From

```
@@ -0,0 +1,7 @@
+package main
+
+import "fmt"
+
+func main() {
+    fmt.Printf("Hello, world\n")
+}
```

Yes, that's right. I thought I was going to write it in Go.

This turned out to be a bad idea. Here's why:

DIGITAL AUDIO WORKSTATION



- Playback Devices
  - JACK
    - system: playback_1, playback_2 (default)
    - PulseAudio JACK Source: front-left, front-right
  - PulseAudio
    - GK104 HDMI Audio Controller Digital Stereo (HDMI)
    - Built-in Audio Analog Stereo (default)
    - Jack sink (PulseAudio JACK Sink)
  - ALSA
    - Default Audio Device (via PulseAudio) (default)
    - HDA Intel PCH, ALC892 Digital: IEC958 (S/PDIF) Digital Audio Output
    - HDA NVidia, HDMI 0: HDMI Audio Output
    - HDA NVidia, HDMI 1: HDMI Audio Output
    - HDA Intel PCH ALC892 Analog (raw) (unable to open device)
    - HDA Intel PCH ALC892 Digital (raw)
    - HDA NVidia HDMI 0 (raw)
    - HDA NVidia HDMI 1 (raw)
  - Dummy
    - Dummy Output Device (default)
- Recording Devices
  - JACK
    - system: capture_1, capture_2 (default)
    - PulseAudio JACK Sink: front-left, front-right
    - Genesis-01: Front Left, Front Right
  - PulseAudio
    - Monitor of GK104 HDMI Audio Controller Digital Stereo (HDMI)
    - Monitor of Built-in Audio Analog Stereo
    - Built-in Audio Analog Stereo (default)
    - Monitor of Jack sink (PulseAudio JACK Sink)
    - Jack source (PulseAudio JACK Source)
  - ALSA
    - Default Audio Device (via PulseAudio) (default)
    - HDA Intel PCH ALC892 Analog (raw) (unable to open device)
    - HDA Intel PCH ALC892 Alt Analog (raw)
  - Dummy
    - Dummy Input Device (default)
- MIDI Devices
  - Midi Through Port-0 (default)
- Audio Clips
- Audio Assets
- External Samples

Genesis Digital Audio Workstation

Progress So Far

2015 Feb 06

I started working on Genesis in March 2013. The first commit:

```
commit 6d1699c42fce480c0b7ec84b521e088b09a3af32
```

This is so frustrating and demotivating that I realized the benefits of Rust did not outweigh the

So I moved on, happy with this choice of language. Over the course of **4 days**, I implemented, *with useful, reusable abstractions*:

- A UTF-8 encoder/decoder and String class.
- OpenGL shaders for text rendering.
- A hash table implementation with robin hood hashing.
- A Label graphics class to render text, which uses FreeType to load a TTF file and caches rendered symbols in a hash table.

At this point I felt crazy for even considering Rust. I had accomplished more in 4 days what took me 16 days in Rust. But more importantly, my abstractions were holding up.

I felt guilty for allowing myself get distracted from actually making progress all this time. From

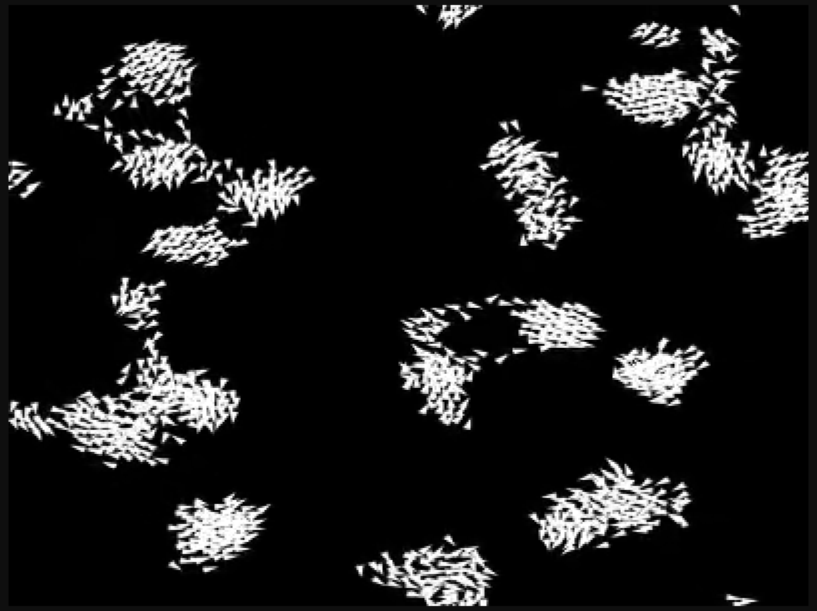
```
@@ -0,0 +1,7 @@
+package main
+
+import "fmt"
+
+func main() {
+    fmt.Printf("Hello, world\n")
+}
```

Yes, that's right. I thought I was going to write it in Go.

This turned out to be a bad idea. Here's why:

Mach

Game engine & graphics toolkit for the future

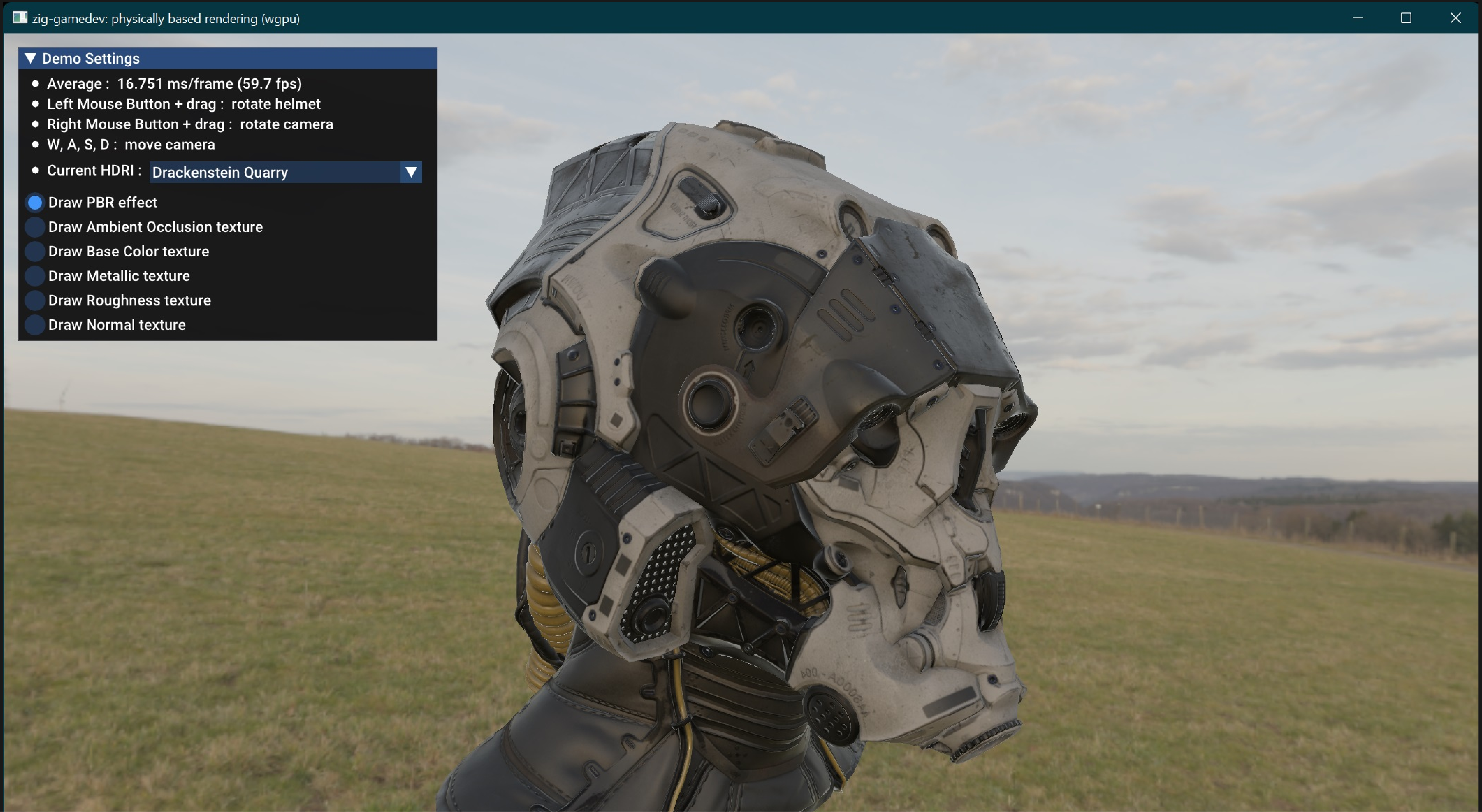


Cross-platform graphics in ~60 seconds

```
git clone https://github.com/hexops/mach
cd mach/
zig build run-example-boids
```

Cross-platform graphics, a unified shader language & compute shaders.
Requires zig 0.10.x | known issues

ZIG-GAMEDEV BY MICHAL Z



zig-gamedev: physically based rendering (wgpu)

▼ Demo Settings

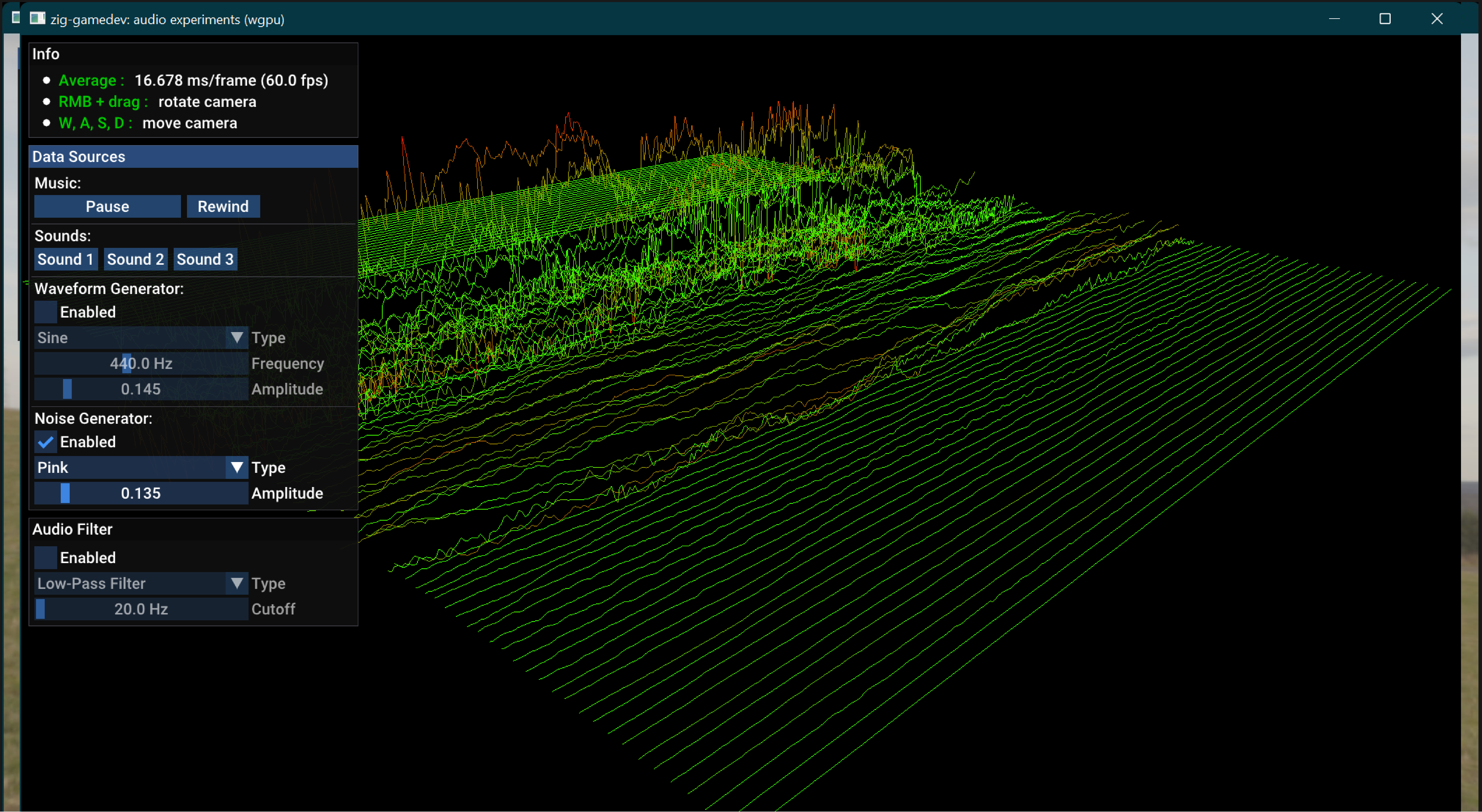
- Average : 16.751 ms/frame (59.7 fps)
- Left Mouse Button + drag : rotate helmet
- Right Mouse Button + drag : rotate camera
- W, A, S, D : move camera
- Current HDRI : Drackenstein Quarry ▼

● Draw PBR effect
○ Draw Ambient Occlusion texture
○ Draw Base Color texture
○ Draw Metallic texture
○ Draw Roughness texture
○ Draw Normal texture

Speaker notes

1. physically based rendering
2. audio experiments
3. bullet physics test
4. procedural mesh

ZIG-GAMEDEV BY MICHAL Z



zig-gamedev: audio experiments (wgpu)

Info

- Average : 16.678 ms/frame (60.0 fps)
- RMB + drag : rotate camera
- W, A, S, D : move camera

Data Sources

Music:

Sounds:

Waveform Generator:

□ Enabled

Type: Sine

Frequency: 440.0 Hz

Amplitude: 0.145

Noise Generator:

☑ Enabled

Type: Pink

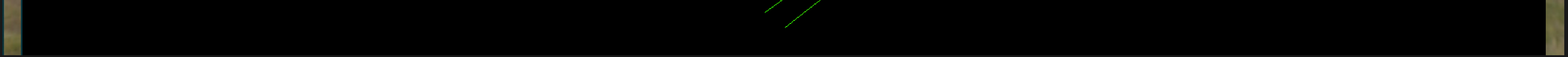
Amplitude: 0.135

Audio Filter

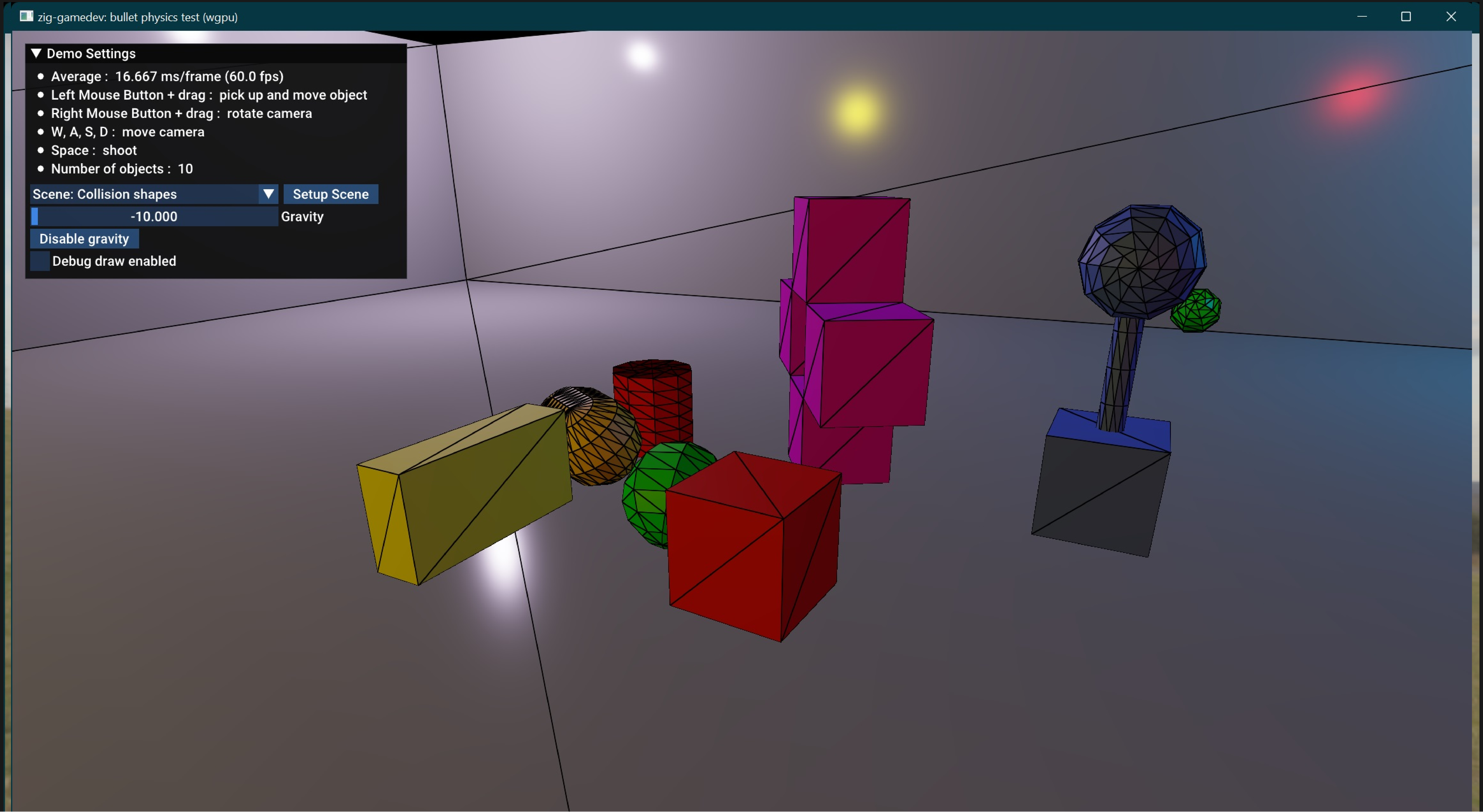
□ Enabled

Type: Low-Pass Filter

Cutoff: 20.0 Hz



ZIG-GAMEDEV BY MICHAL Z



zig-gamedev: bullet physics test (wgpu)

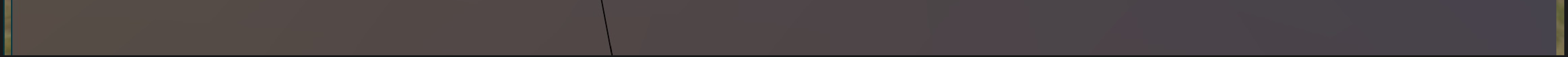
Demo Settings

- Average : 16.667 ms/frame (60.0 fps)
- Left Mouse Button + drag : pick up and move object
- Right Mouse Button + drag : rotate camera
- W, A, S, D : move camera
- Space : shoot
- Number of objects : 10

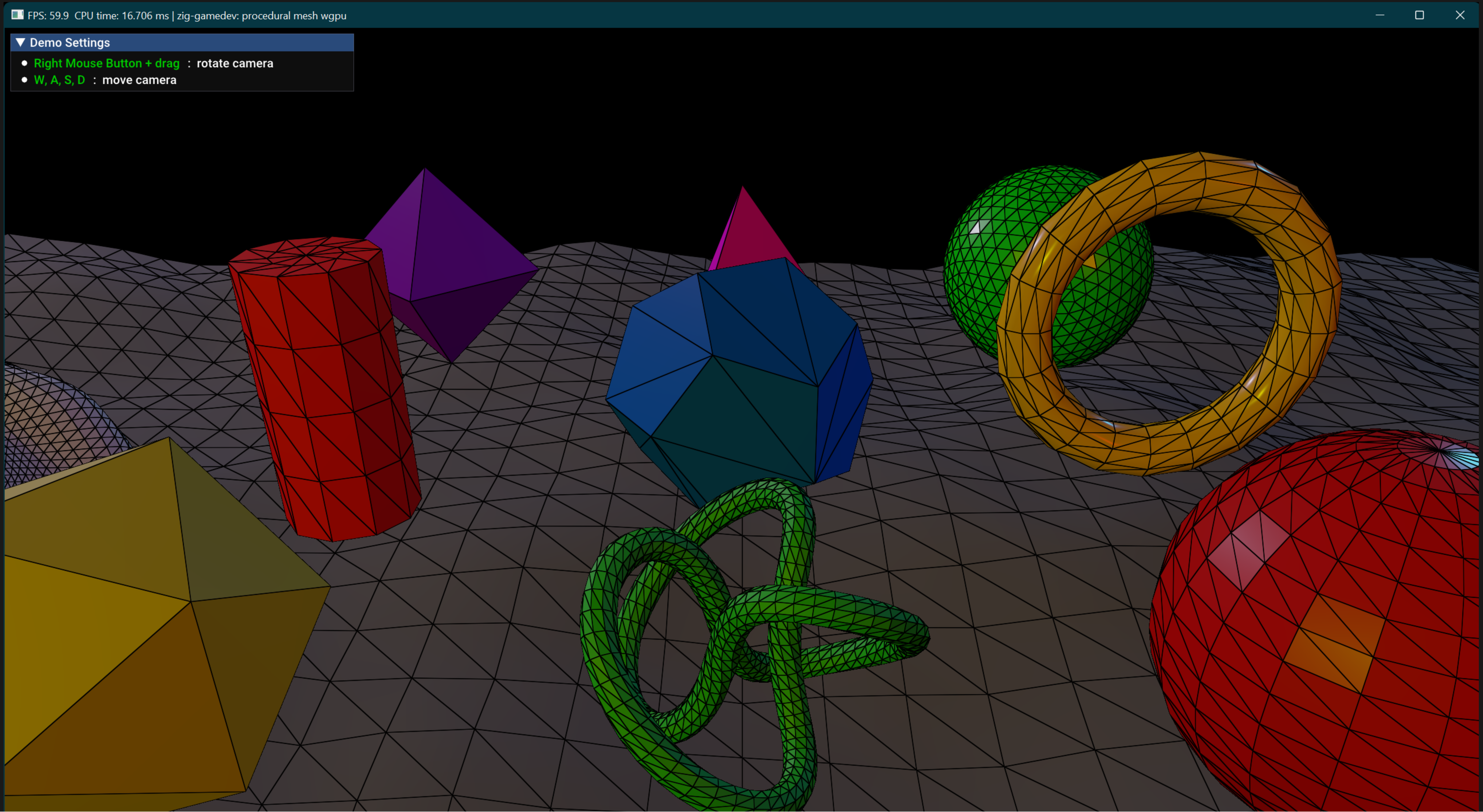
Scene: Collision shapes ▼ Setup Scene

-10.000 Gravity

□ Debug draw enabled



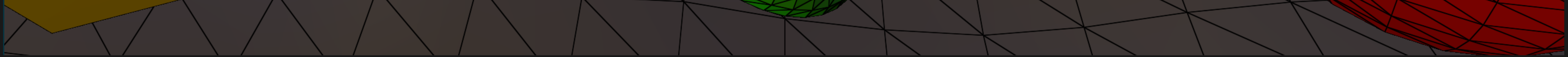
ZIG-GAMEDEV BY MICHAL Z



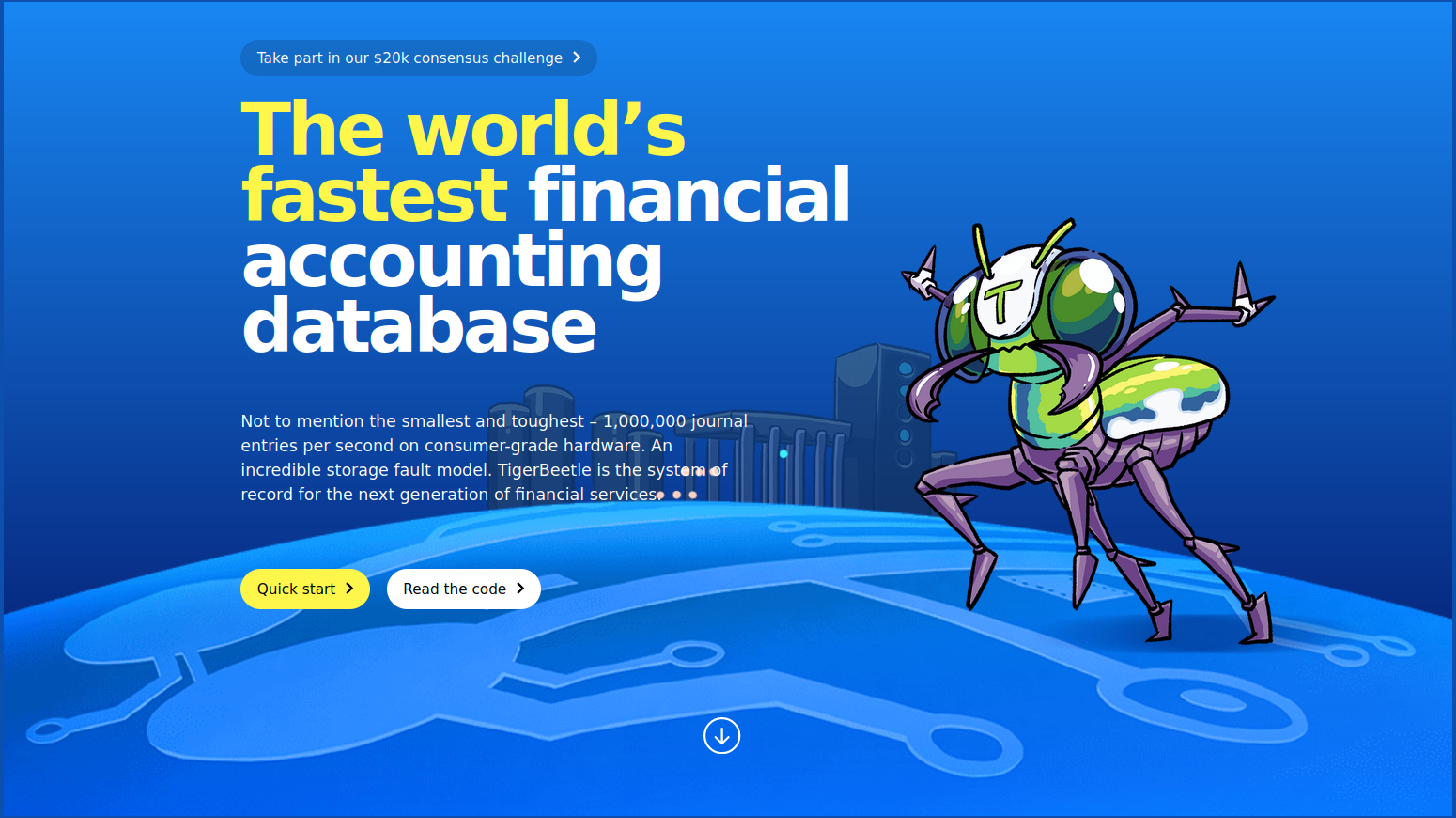
FPS: 59.9 CPU time: 16.706 ms | zig-gamedev: procedural mesh wgpu

▼ Demo Settings

- Right Mouse Button + drag : rotate camera
- W, A, S, D : move camera



TIGERBEETLE



The world's fastest financial accounting database

Not to mention the smallest and toughest – 1,000,000 journal entries per second on consumer-grade hardware. An incredible storage fault model. TigerBeetle is the system of record for the next generation of financial services.

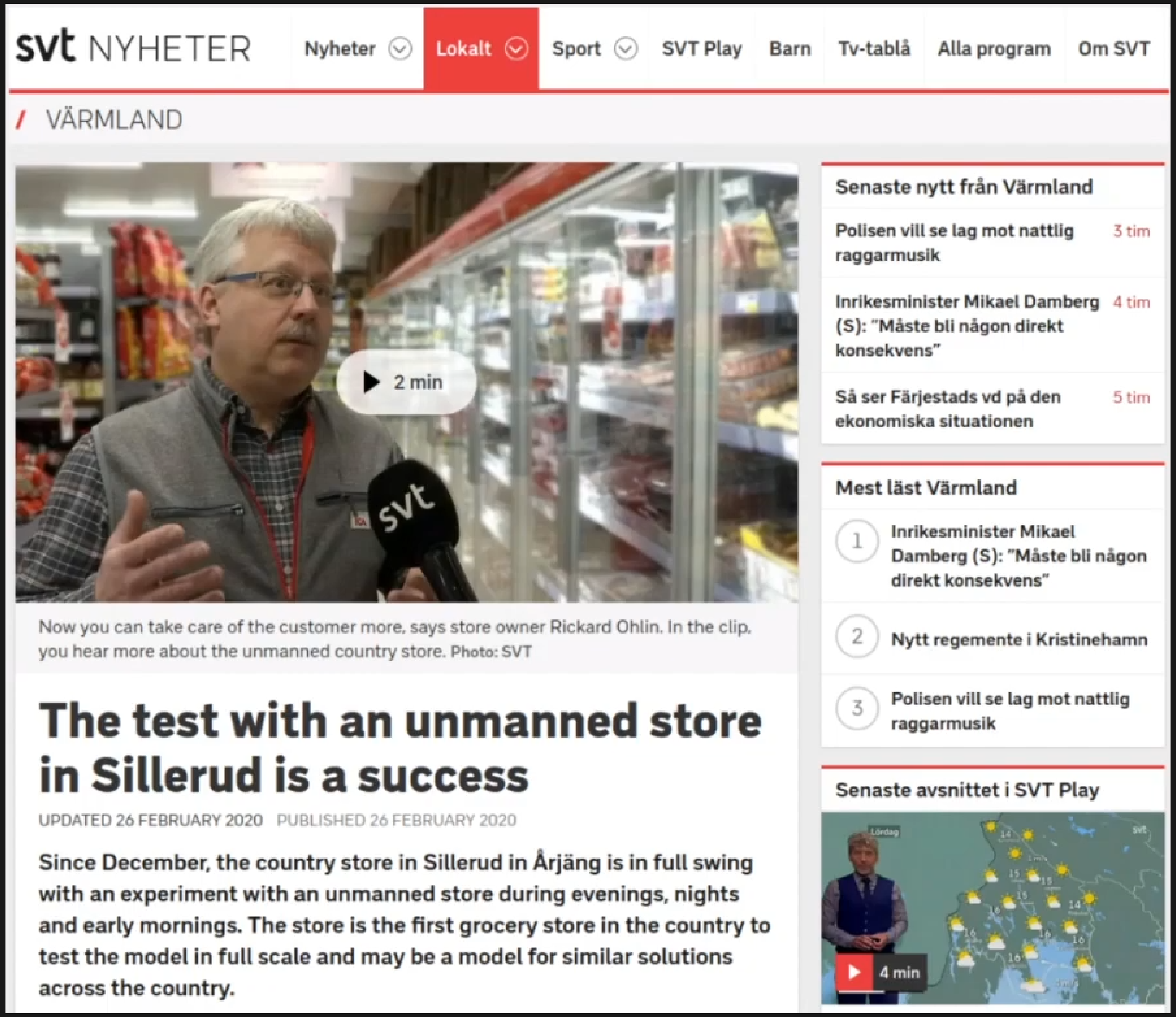
Speaker notes

Mention that they recently gained independence from Coil

ZIG IN ACTION

- Low-Level Infrastructure
- Libraries to be used by higher level languages
- High-performance applications
  - Real-Time Digital Audio Processing
  - Video Games
  - Databases
- Resource-Constrained Environments
  - Operating Systems
  - Embedded development
  - WebAssembly

UNMANNED STORE IN SILLERUD



Now you can take care of the customer more, says store owner Rickard Ohlin. In the clip, you hear more about the unmanned country store. Photo: SVT

The test with an unmanned store in Sillerud is a success

UPDATED 26 FEBRUARY 2020 PUBLISHED 26 FEBRUARY 2020

Since December, the country store in Sillerud in Årjäng is in full swing with an experiment with an unmanned store during evenings, nights and early mornings. The store is the first grocery store in the country to test the model in full scale and may be a model for similar solutions across the country.

Senaste nytt från Värmland

Polisen vill se lag mot nattlig raggarmusik 3 tim

Inrikesminister Mikael Damberg (S): "Måste bli någon direkt konsekvens" 4 tim

Så ser Färjestads vd på den ekonomiska situationen 5 tim

Mest läst Värmland

1 Inrikesminister Mikael Damberg (S): "Måste bli någon direkt konsekvens"

2 Nytt regemente i Kristinehamn

3 Polisen vill se lag mot nattlig raggarmusik

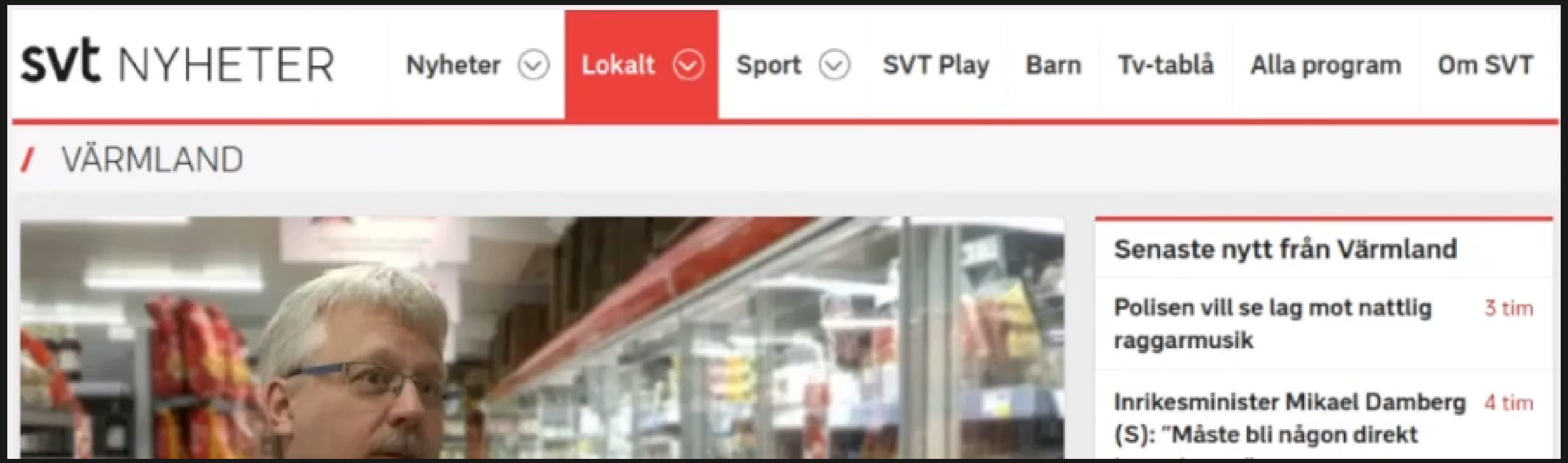
Senaste avsnittet i SVT Play

Speaker notes

The store was open at night with no employees

The test ran for a few months and Jens reported that there were zero bugs! That is a seriously impressive accomplishment.

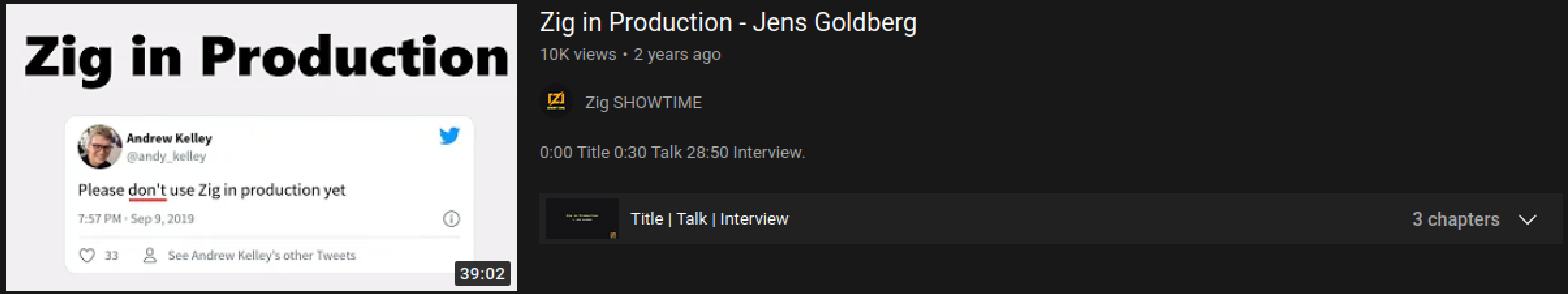
UNMANNED STORE IN SILLERUD



Senaste nytt från Värmland

Polisen vill se lag mot nattlig raggarmusik 3 tim

Inrikesminister Mikael Damberg (S): "Måste bli någon direkt 4 tim



Andrew Kelley
@andy_kelley

Please don't use Zig in production yet

7:57 PM · Sep 9, 2019

33 See Andrew Kelley's other Tweets

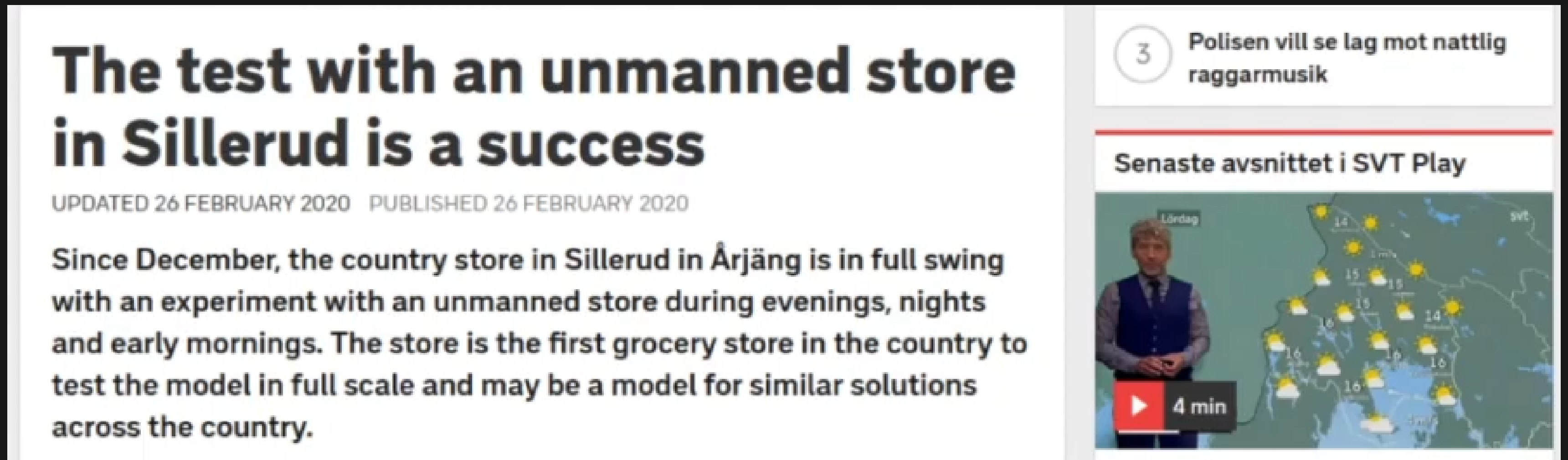
Zig in Production - Jens Goldberg

10K views • 2 years ago

Zig SHOWTIME

0:00 Title 0:30 Talk 28:50 Interview.

Title | Talk | Interview 3 chapters



The test with an unmanned store in Sillerud is a success

UPDATED 26 FEBRUARY 2020 PUBLISHED 26 FEBRUARY 2020

Since December, the country store in Sillerud in Årjäng is in full swing with an experiment with an unmanned store during evenings, nights and early mornings. The store is the first grocery store in the country to test the model in full scale and may be a model for similar solutions across the country.

3 Polisen vill se lag mot nattlig raggarmusik

Senaste avsnittet i SVT Play

§ 📟 Zig Embedded Group 🔋

This group was formed to document and improve the embedded programming experience with the Zig programming language.

§ 🏁 Goals

- Provide documents on how to get started with embedded programming (for absolute newbies)
- Provide example snippets for common opperations on certain architectures (LPC, STM32, AVR, ...)
- Provide example worked through embedded mini-projects
- Create register definition libraries
- Create a common interface/HAL over several architectures
- Create a performant common set of drivers for external platforms

§ 🧑‍🏫 Introduction to embedded programming

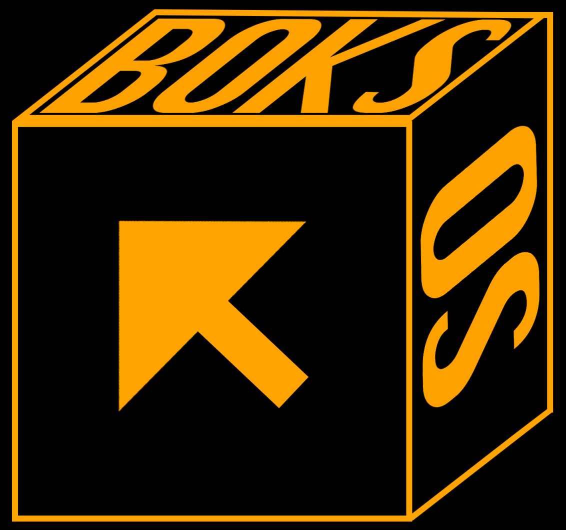
If you've never done any embedded development before, it's a good point to start with one of our tutorials:

- (*WIP*) Embedded Basics: Aims to provide a basic understanding of the embedded environment.
- (*WIP*) Embedded Programming for Beginners: Aims to provide a basic understanding of embedded programming concepts.
- *Coming soon*

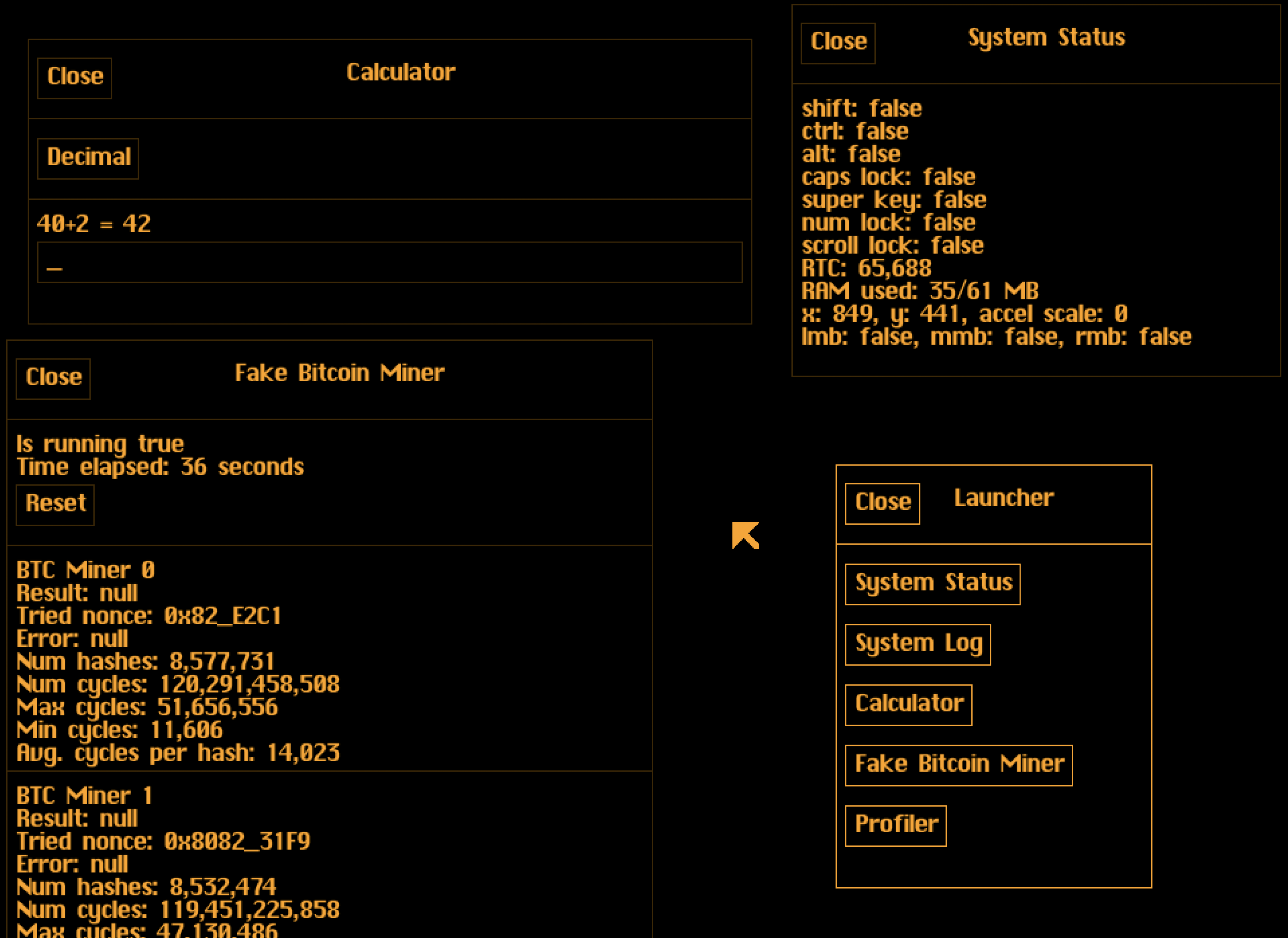
MicroZig

A unified abstraction layer and HAL for several microcontrollers.

BOKSOS



BOKSOS



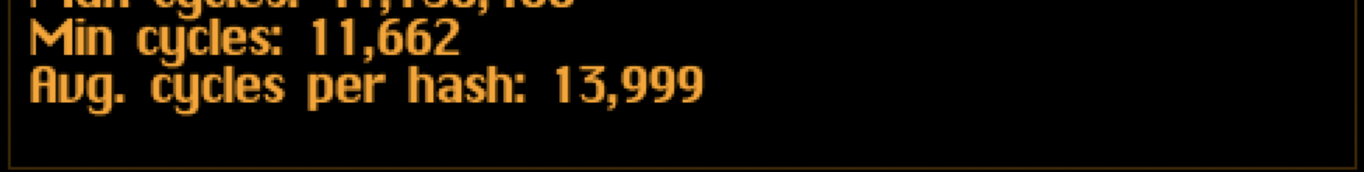
```
40+2 = 42
_
```

```
shift: false
ctrl: false
alt: false
caps lock: false
super key: false
num lock: false
scroll lock: false
RTC: 65,688
RAM used: 35/61 MB
x: 849, y: 441, accel scale: 0
lmb: false, mmb: false, rmb: false
```

```
Is running true
Time elapsed: 36 seconds
```

```
BTC Miner 0
Result: null
Tried nonce: 0x82_E2C1
Error: null
Num hashes: 8,577,731
Num cycles: 120,291,458,508
Max cycles: 51,656,556
Min cycles: 11,606
Avg. cycles per hash: 14,023
```

```
BTC Miner 1
Result: null
Tried nonce: 0x8082_31F9
Error: null
Num hashes: 8,532,474
Num cycles: 119,451,225,858
Max cycles: 47,130,486
```



```
Max cycles: 11,[illegible]
Min cycles: 11,662
Avg. cycles per hash: 13,999
```

WASM-4 GAMES

WASM-4

Build retro games using WebAssembly for a fantasy console

Play Games 🚀 Get Started

A quick video introduction to WASM-4.

WASM-4 GAMES

Game Jam 2 Results

August 26, 2022

Bruno Garcia

The second WASM-4 game jam is a wrap and the results are in! Thanks again to all our judges who each took the time to rate all 26 games. Thanks again to Wasmer for sponsoring a prize fund and helping us with promotion.

The quality of many of the games in this jam were absolutely incredible. It was very difficult to rank the top entries, and we even had a 3-way tie for 3rd place! After much deliberation, the judges sorted out the tie to give us our 3rd and 4th place winners. Since it was extremely close with the tie breaker, I'll be pitching in a small bonus prize for our 5th place winner.

Without further ado, here is the final ranking!

Winners

5th place and $100, Samurai Revenge (wapm link) by Krylan!

4th place and $250, Asteroids 3000 (wapm link) by HitchH1k3r!

3rd place and $500, disk-0 MADNESS (wapm link) by maxcurzi!

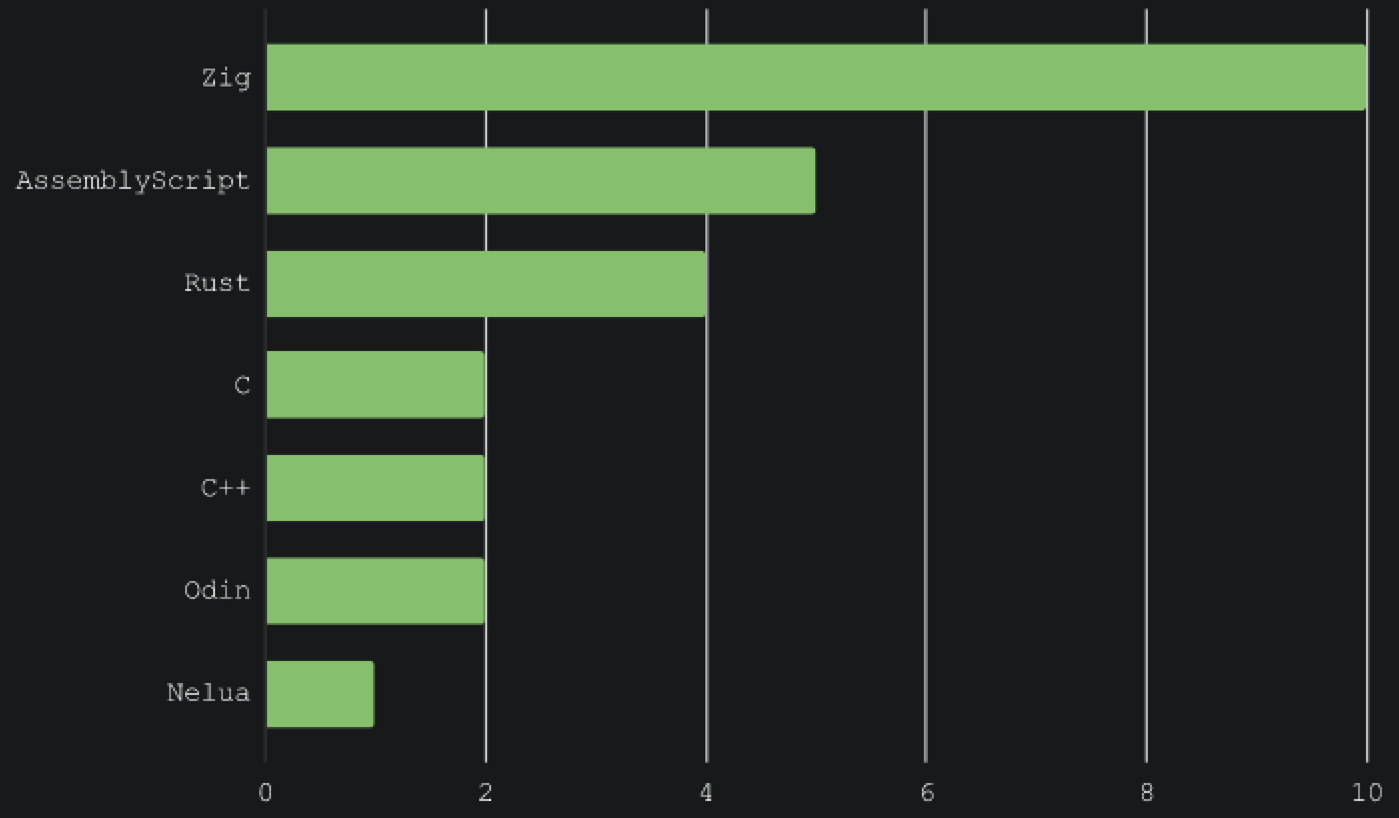
WASM-4 GAMES

Developer Survey

As part of the game submission process, we asked developers to fill out a short, optional survey with some basic questions. Since almost everyone took the time to fill out this survey, we have enough data to share some interesting graphs about the average WASM-4 developer.

Programming Language

We asked developers about the programming language they used to build their jam game:



Zig has really taken over! AssemblyScript also had a strong showing, coming out of nowhere after being almost completely absent from our last jam.

A TASTE OF ZIG

(Not to be confused with A Taste Of India).

Speaker notes

MMM!! 5/5 stars.

SOME HIGHLIGHTS

- ArrayList
- Inline for loops
- MultiArrayList
- AutoArrayHashMap
- C Translation
- Unit Testing
- Untagged Union Safety

ArrayList

```zig
fn ArrayList(comptime T: type) type {
    return struct {
        items: []T,
        capacity: usize,
    };
}
```

zig is a small amount of orthogonal features that function elegantly together

"Focus on debugging your application, not your programming language"

Languages can be controversial but everyone agrees, like it or not, when you use Zig you spend the vast majority of your time dealing directly with your understanding of your application, as opposed to C++ where you're trying to figure out esoteric language rules, or Rust where you're trying to appease the borrow checker with a worthy sacrifice.

ArrayList

```zig
fn ArrayList(comptime T: type) type {
    return struct {
        items: []T,
        capacity: usize,
    };
}
```

ArrayList

```
fn ArrayList(comptime T: type) type {
    return struct {
        items: []T,
        capacity: usize,
    };
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

Inline Loops

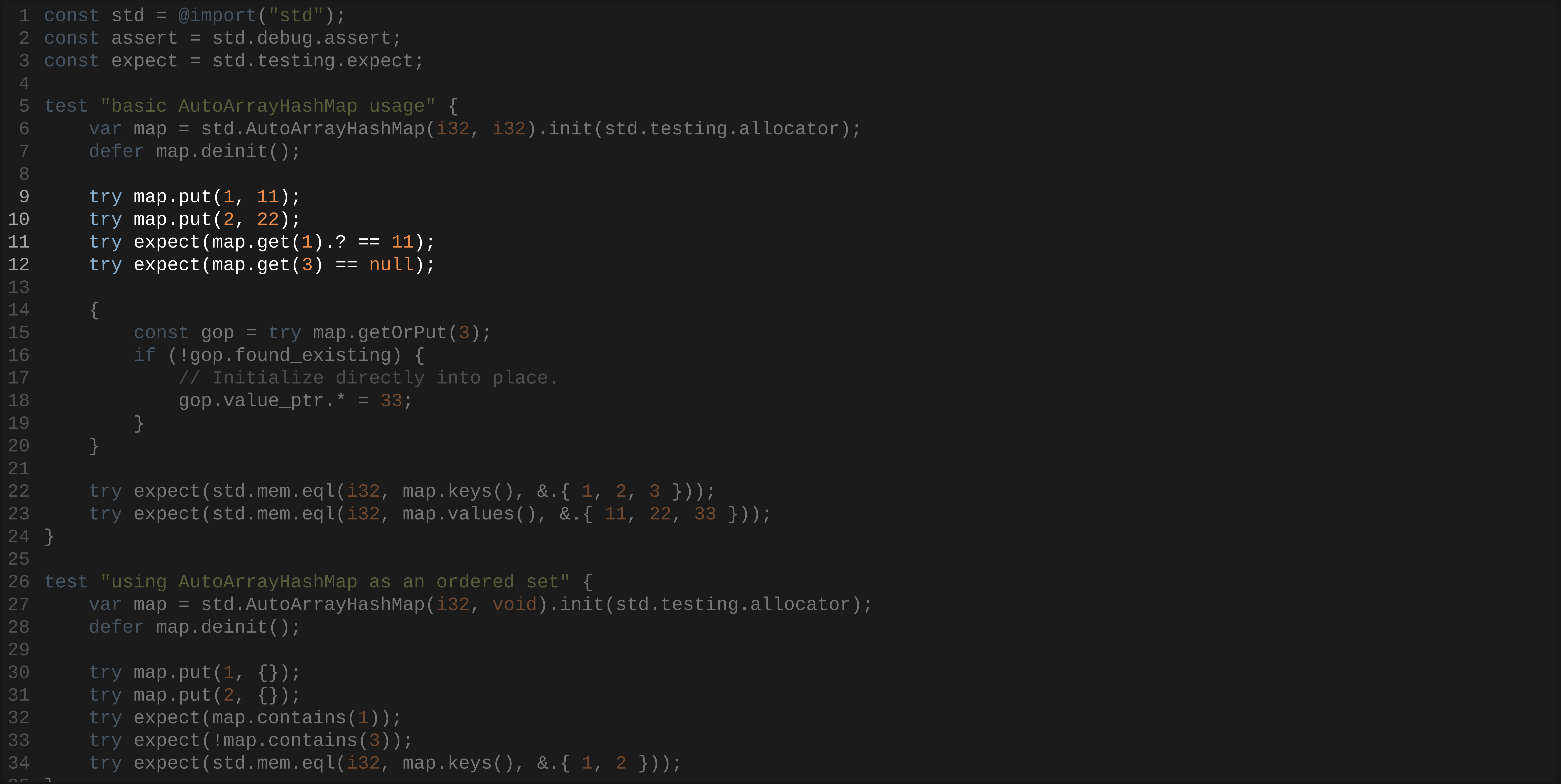
```zig
const std = @import("std");
const assert = std.debug.assert;

const Data = struct {
    foo: Foo,
    bytes: [8]u8,
    ok: bool,
};

const Foo = enum { hello, world };

pub fn main() void {
    var d: Data = .{
        .foo = .world,
        .bytes = "abcdefgh".*,
        .ok = true,
    };
    dump(d);
}

fn dump(data: anytype) void {
    const T = @TypeOf(data);
    inline for (@typeInfo(T).Struct.fields) |field| {
        std.debug.print("{any}\n", .{@field(data, field.name)});
    }
}
```

```
$ zig run test.zig
test.Foo.world
{ 97, 98, 99, 100, 101, 102, 103, 104 }
true
```

AutoArrayHashMap

```
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

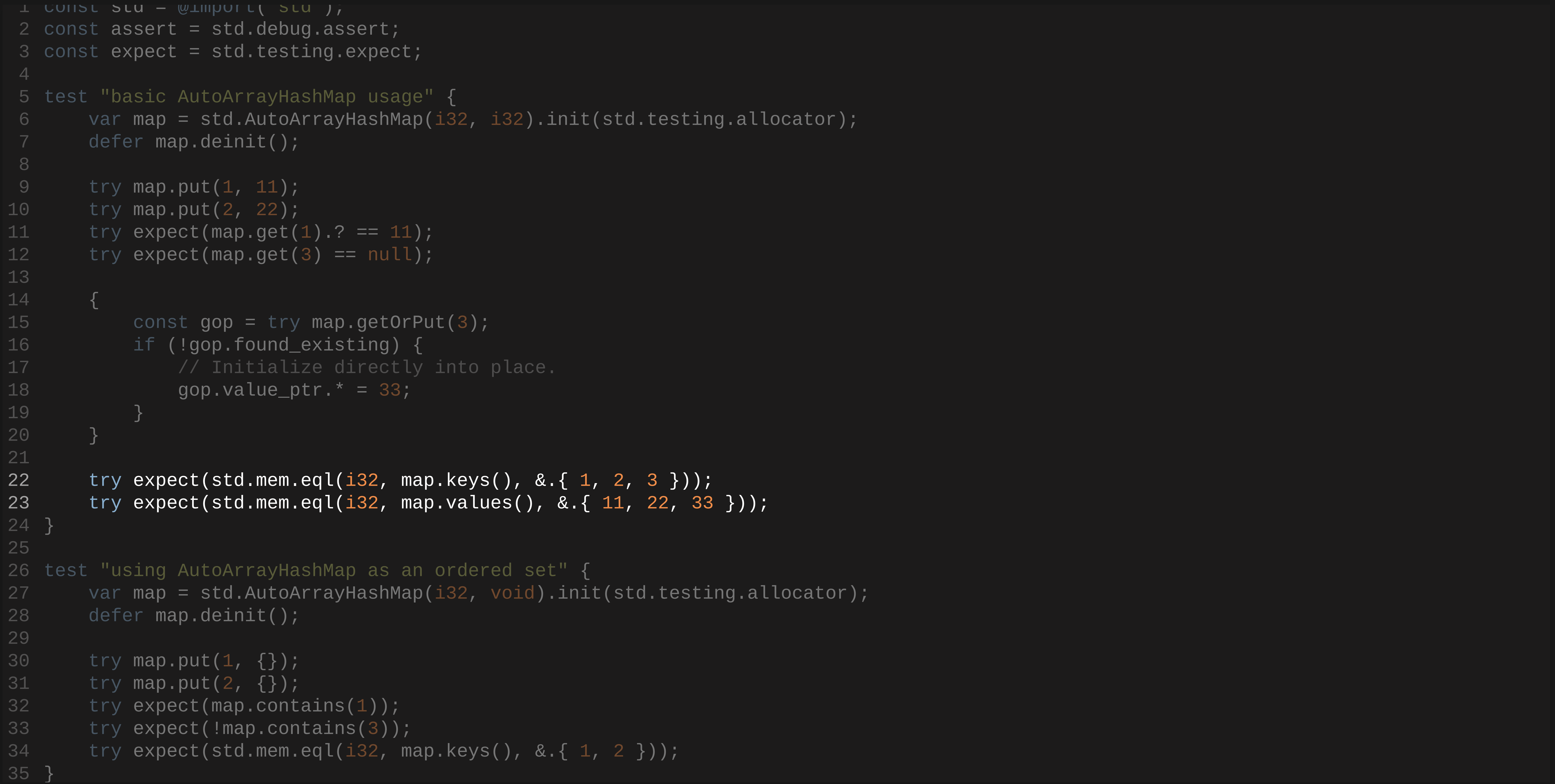


```
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

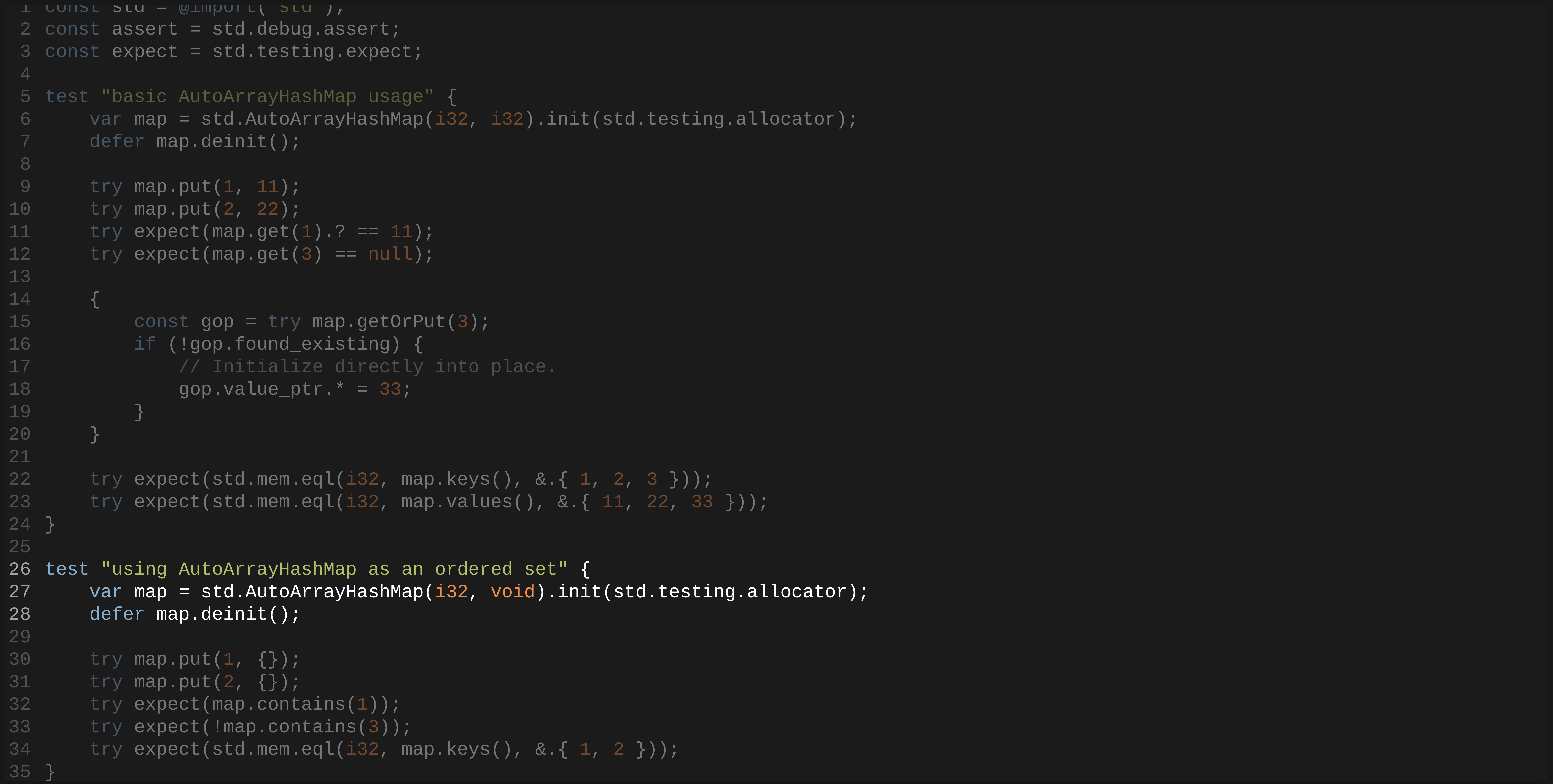
```zig
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap



```
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

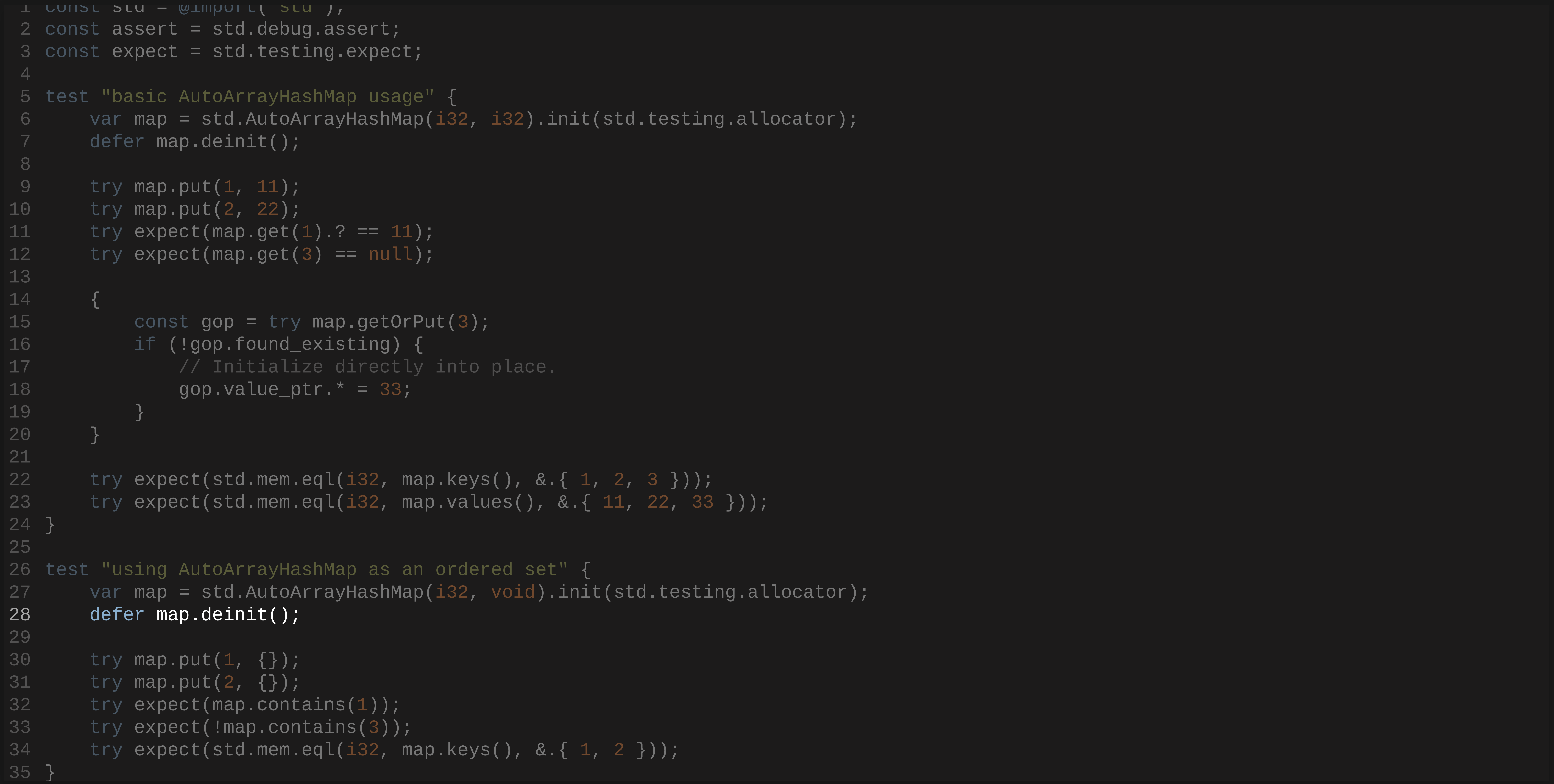


```
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

```zig
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

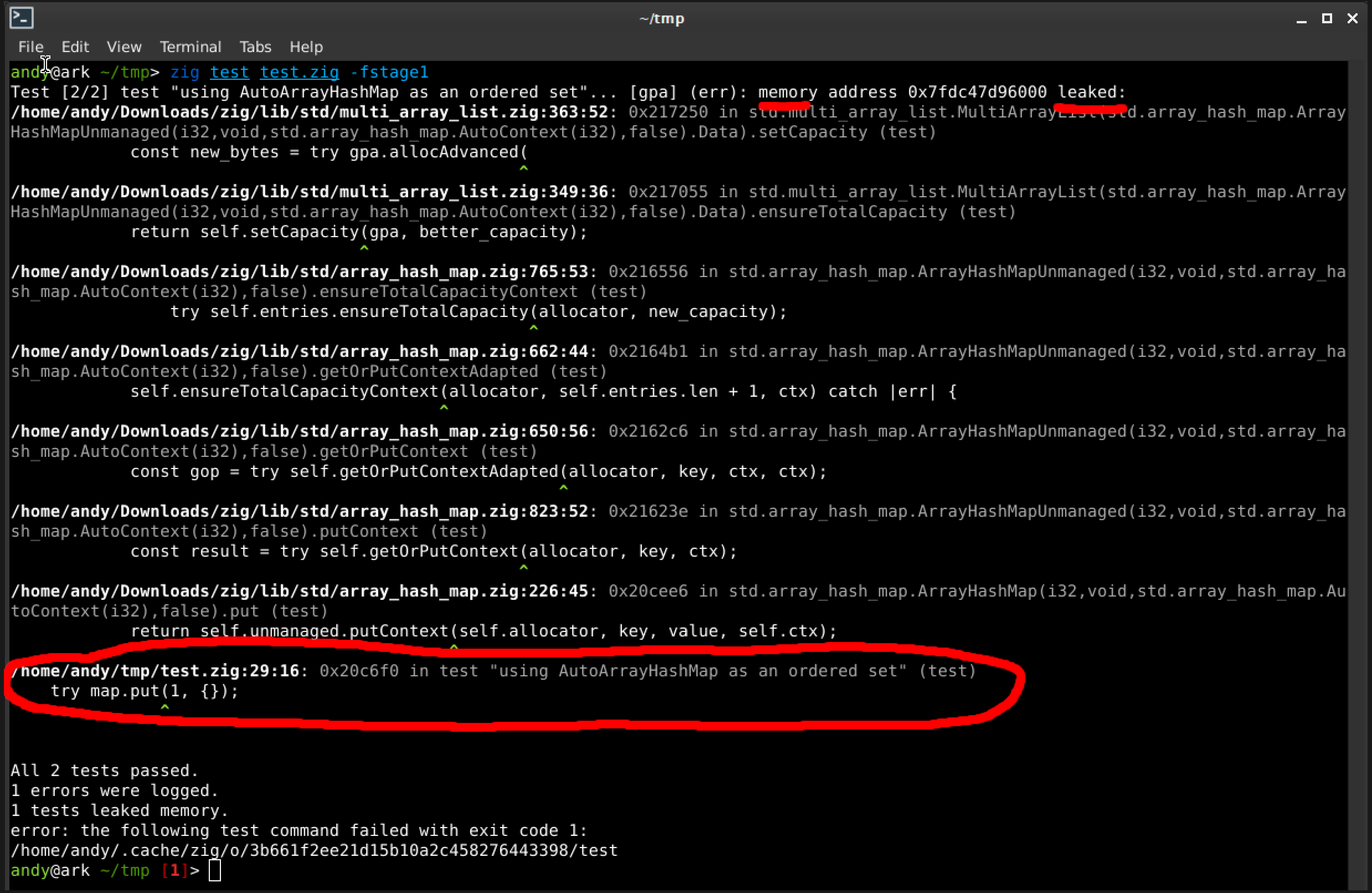


```
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

AutoArrayHashMap

```zig
const std = @import("std");
const assert = std.debug.assert;
const expect = std.testing.expect;

test "basic AutoArrayHashMap usage" {
    var map = std.AutoArrayHashMap(i32, i32).init(std.testing.allocator);
    defer map.deinit();

    try map.put(1, 11);
    try map.put(2, 22);
    try expect(map.get(1).? == 11);
    try expect(map.get(3) == null);

    {
        const gop = try map.getOrPut(3);
        if (!gop.found_existing) {
            // Initialize directly into place.
            gop.value_ptr.* = 33;
        }
    }

    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2, 3 }));
    try expect(std.mem.eql(i32, map.values(), &.{ 11, 22, 33 }));
}

test "using AutoArrayHashMap as an ordered set" {
    var map = std.AutoArrayHashMap(i32, void).init(std.testing.allocator);

    try map.put(1, {});
    try map.put(2, {});
    try expect(map.contains(1));
    try expect(!map.contains(3));
    try expect(std.mem.eql(i32, map.keys(), &.{ 1, 2 }));
}
```

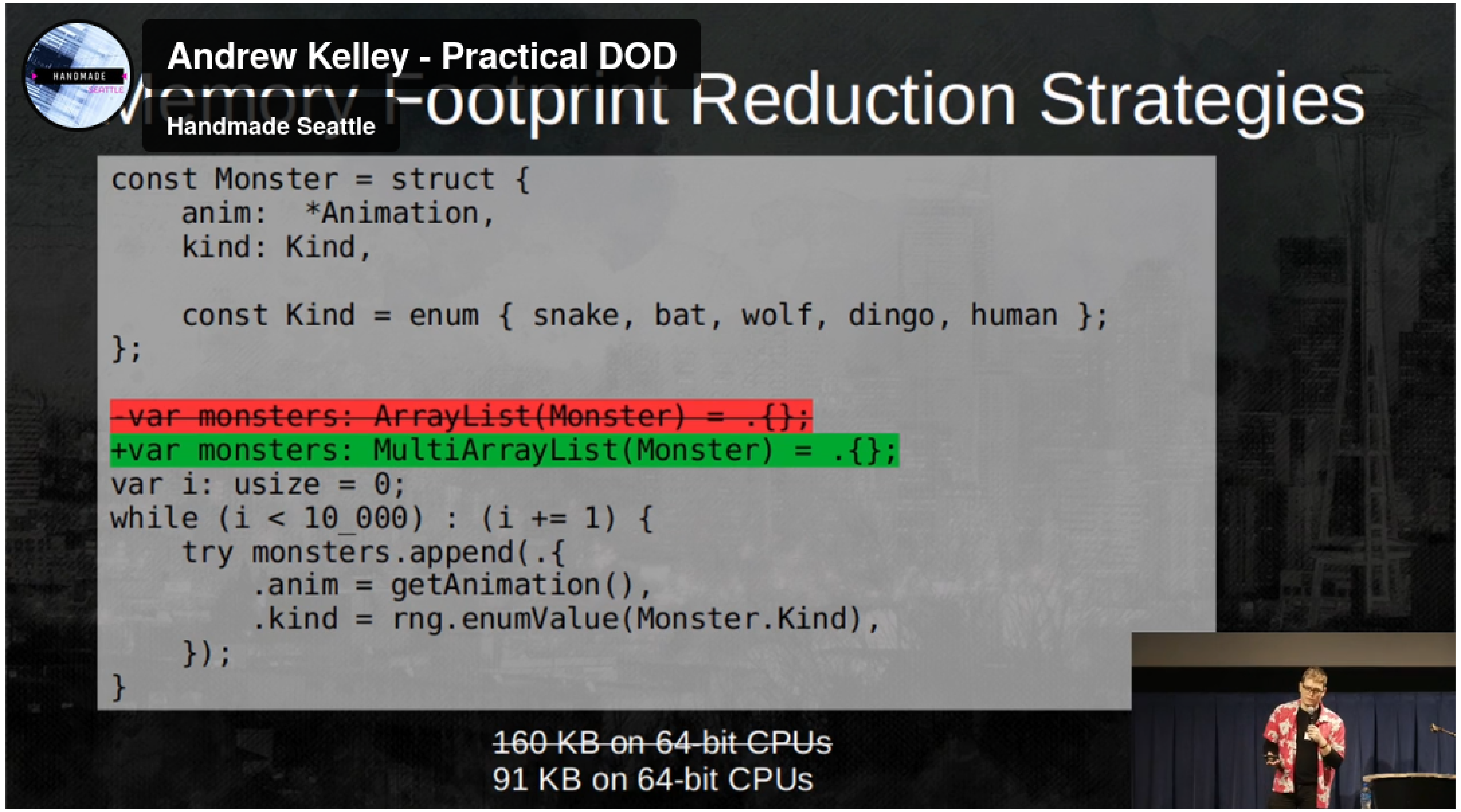
AutoArrayHashMap



```
andy@ark ~/tmp> zig test test.zig -fstage1
Test [2/2] test "using AutoArrayHashMap as an ordered set"... [gpa] (err): memory address 0x7fdc47d96000 leaked:
/home/andy/Downloads/zig/lib/std/multi_array_list.zig:363:52: 0x217250 in std.multi_array_list.MultiArrayList(std.array_hash_map.Array
HashMapUnmanaged(i32,void,std.array_hash_map.AutoContext(i32),false).Data).setCapacity (test)
            const new_bytes = try gpa.allocAdvanced(
                                                   ^

/home/andy/Downloads/zig/lib/std/multi_array_list.zig:349:36: 0x217055 in std.multi_array_list.MultiArrayList(std.array_hash_map.Array
HashMapUnmanaged(i32,void,std.array_hash_map.AutoContext(i32),false).Data).ensureTotalCapacity (test)
            return self.setCapacity(gpa, better_capacity);
                                   ^

/home/andy/Downloads/zig/lib/std/array_hash_map.zig:765:53: 0x216556 in std.array_hash_map.ArrayHashMapUnmanaged(i32,void,std.array_ha
sh_map.AutoContext(i32),false).ensureTotalCapacityContext (test)
                try self.entries.ensureTotalCapacity(allocator, new_capacity);
                                                    ^

/home/andy/Downloads/zig/lib/std/array_hash_map.zig:662:44: 0x2164b1 in std.array_hash_map.ArrayHashMapUnmanaged(i32,void,std.array_ha
sh_map.AutoContext(i32),false).getOrPutContextAdapted (test)
            self.ensureTotalCapacityContext(allocator, self.entries.len + 1, ctx) catch |err| {
                                           ^

/home/andy/Downloads/zig/lib/std/array_hash_map.zig:650:56: 0x2162c6 in std.array_hash_map.ArrayHashMapUnmanaged(i32,void,std.array_ha
sh_map.AutoContext(i32),false).getOrPutContext (test)
            const gop = try self.getOrPutContextAdapted(allocator, key, ctx, ctx);
                                                       ^

/home/andy/Downloads/zig/lib/std/array_hash_map.zig:823:52: 0x21623e in std.array_hash_map.ArrayHashMapUnmanaged(i32,void,std.array_ha
sh_map.AutoContext(i32),false).putContext (test)
            const result = try self.getOrPutContext(allocator, key, ctx);
                                                   ^

/home/andy/Downloads/zig/lib/std/array_hash_map.zig:226:45: 0x20cee6 in std.array_hash_map.ArrayHashMap(i32,void,std.array_hash_map.Au
toContext(i32),false).put (test)
            return self.unmanaged.putContext(self.allocator, key, value, self.ctx);
                                            ^

/home/andy/tmp/test.zig:29:16: 0x20c6f0 in test "using AutoArrayHashMap as an ordered set" (test)
    try map.put(1, {});
               ^


All 2 tests passed.
1 errors were logged.
1 tests leaked memory.
error: the following test command failed with exit code 1:
/home/andy/.cache/zig/o/3b661f2ee21d15b10a2c458276443398/test
andy@ark ~/tmp [1]>
```

MultiArrayList

A Practical Guide to Applying Data-Oriented Design



Memory Footprint Reduction Strategies

```
const Monster = struct {
    anim:  *Animation,
    kind: Kind,

    const Kind = enum { snake, bat, wolf, dingo, human };
};

-var monsters: ArrayList(Monster) = .{};
+var monsters: MultiArrayList(Monster) = .{};
var i: usize = 0;
while (i < 10_000) : (i += 1) {
    try monsters.append(.{
        .anim = getAnimation(),
        .kind = rng.enumValue(Monster.Kind),
    });
}
```

~~160 KB on 64-bit CPUs~~
91 KB on 64-bit CPUs

Practical Data-Oriented Design

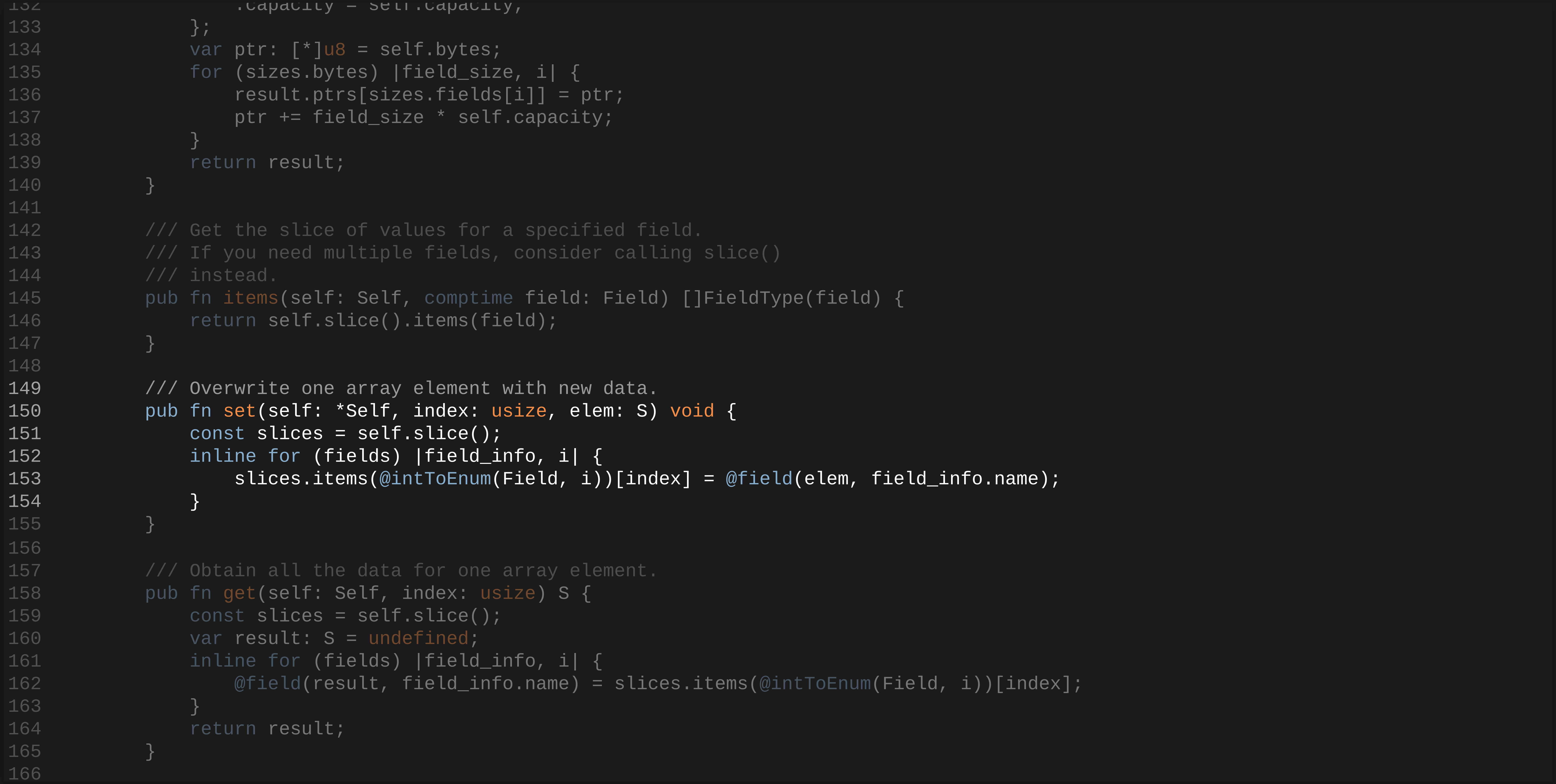
MultiArrayList

```zig
const std = @import("std.zig");
const builtin = @import("builtin");
const assert = std.debug.assert;
const meta = std.meta;
const mem = std.mem;
const Allocator = mem.Allocator;
const testing = std.testing;

/// A MultiArrayList stores a list of a struct type.
/// Instead of storing a single list of items, MultiArrayList
/// stores separate lists for each field of the struct.
/// This allows for memory savings if the struct has padding,
/// and also improves cache usage if only some fields are needed
/// for a computation.  The primary API for accessing fields is
/// the `slice()` function, which computes the start pointers
/// for the array of each field.  From the slice you can call
/// `.items(.<field_name>)` to obtain a slice of field values.
pub fn MultiArrayList(comptime S: type) type {
    return struct {
        bytes: [*]align(@alignOf(S)) u8 = undefined,
        len: usize = 0,
        capacity: usize = 0,

        pub const Elem = S;

        pub const Field = meta.FieldEnum(S);

        /// A MultiArrayList.Slice contains cached start pointers for each field in the list.
        /// These pointers are not normally stored to reduce the size of the list in memory.
        /// If you are accessing multiple fields, call slice() first to compute the pointers,
        /// and then get the field arrays from the slice.
        pub const Slice = struct {
            /// This array is indexed by the field index which can be obtained
            /// by using @enumToInt() on the Field enum
            ptrs: [fields.len][*]u8,
```

Speaker notes

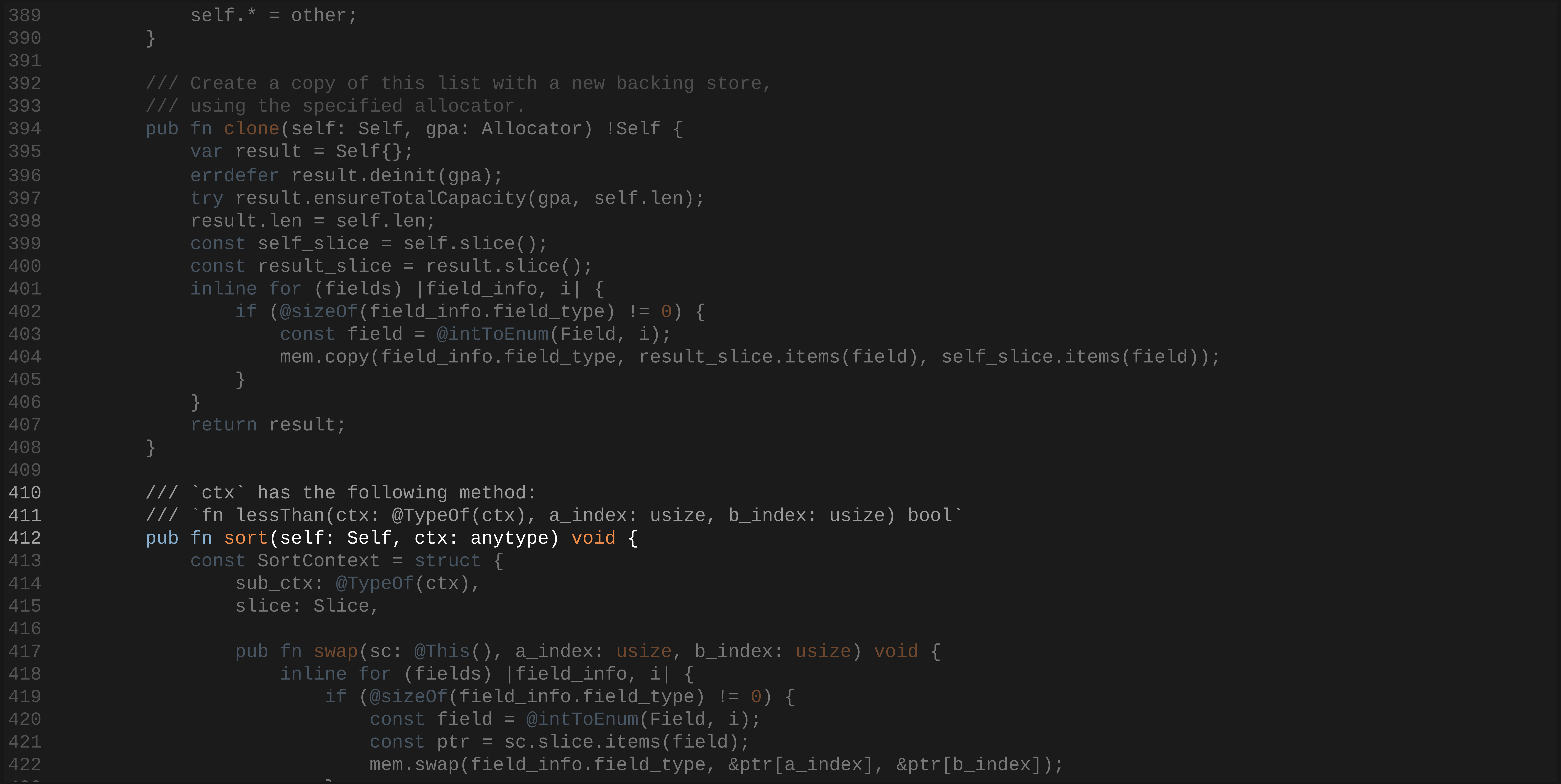
with tests, comments, and blank lines removed it is only 336 lines

MultiArrayList



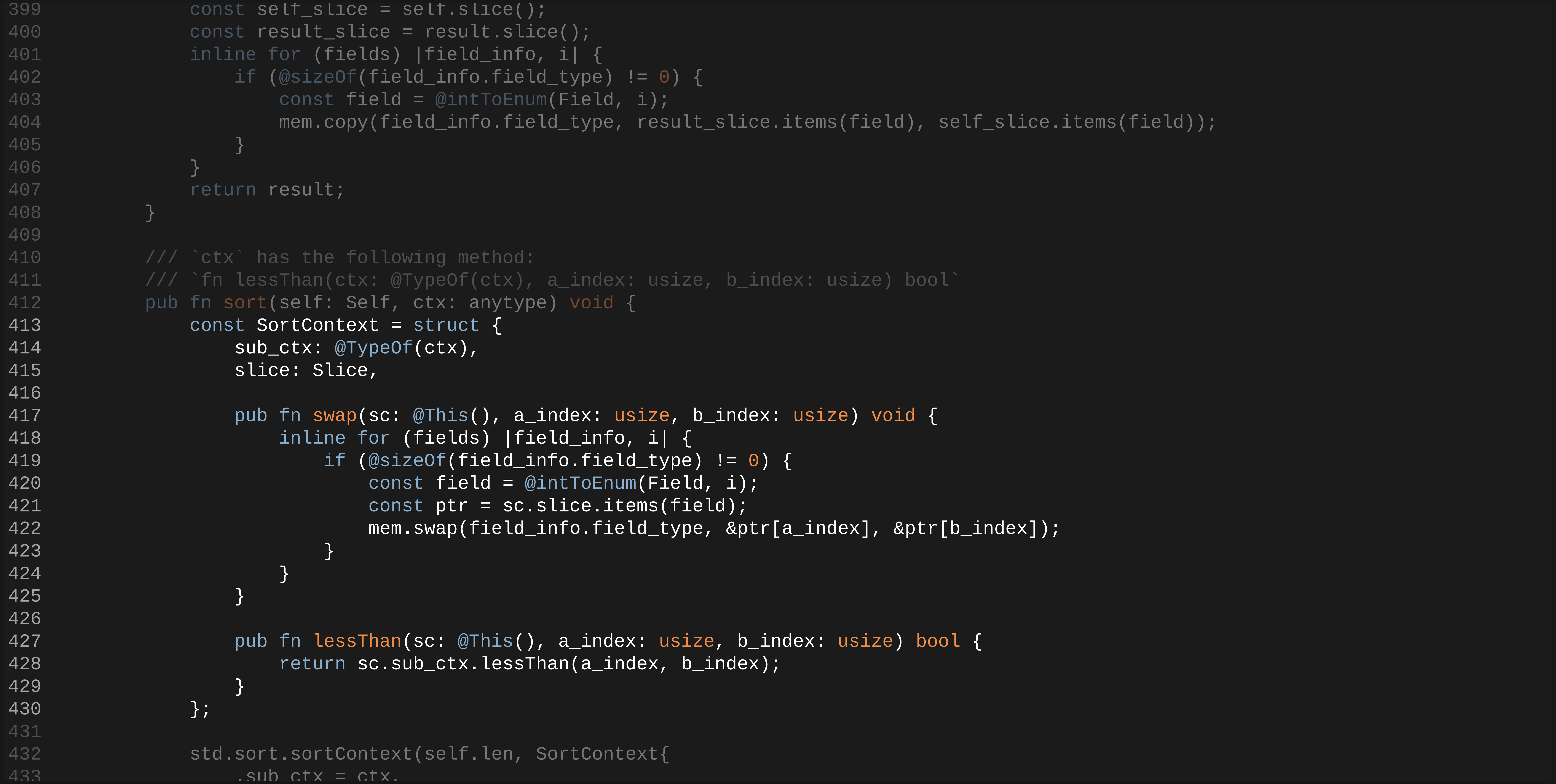
```
            .capacity = self.capacity,
        };
        var ptr: [*]u8 = self.bytes;
        for (sizes.bytes) |field_size, i| {
            result.ptrs[sizes.fields[i]] = ptr;
            ptr += field_size * self.capacity;
        }
        return result;
    }

    /// Get the slice of values for a specified field.
    /// If you need multiple fields, consider calling slice()
    /// instead.
    pub fn items(self: Self, comptime field: Field) []FieldType(field) {
        return self.slice().items(field);
    }

    /// Overwrite one array element with new data.
    pub fn set(self: *Self, index: usize, elem: S) void {
        const slices = self.slice();
        inline for (fields) |field_info, i| {
            slices.items(@intToEnum(Field, i))[index] = @field(elem, field_info.name);
        }
    }

    /// Obtain all the data for one array element.
    pub fn get(self: Self, index: usize) S {
        const slices = self.slice();
        var result: S = undefined;
        inline for (fields) |field_info, i| {
            @field(result, field_info.name) = slices.items(@intToEnum(Field, i))[index];
        }
        return result;
    }
```

MultiArrayList



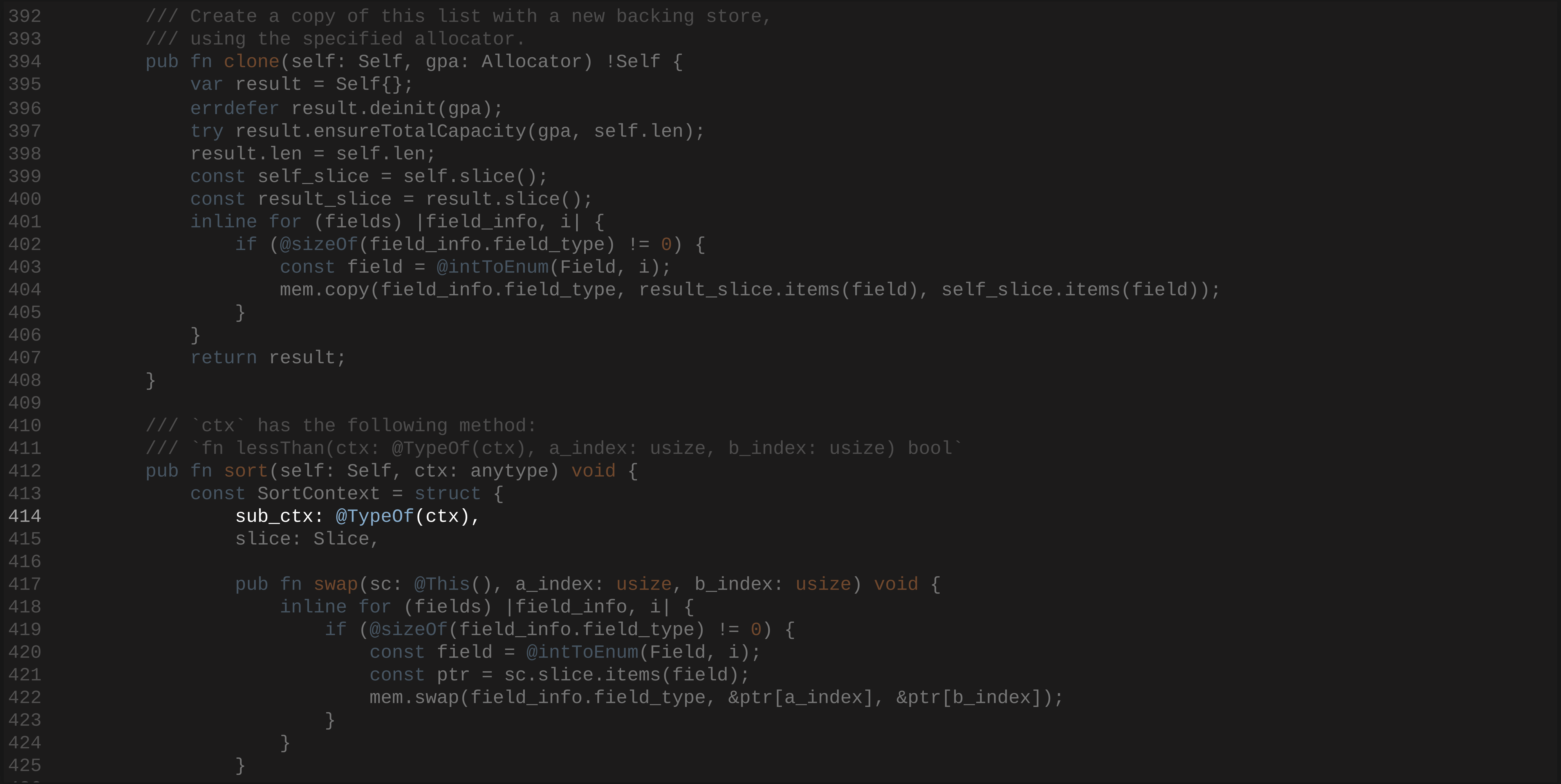
```zig
        self.* = other;
    }

    /// Create a copy of this list with a new backing store,
    /// using the specified allocator.
    pub fn clone(self: Self, gpa: Allocator) !Self {
        var result = Self{};
        errdefer result.deinit(gpa);
        try result.ensureTotalCapacity(gpa, self.len);
        result.len = self.len;
        const self_slice = self.slice();
        const result_slice = result.slice();
        inline for (fields) |field_info, i| {
            if (@sizeOf(field_info.field_type) != 0) {
                const field = @intToEnum(Field, i);
                mem.copy(field_info.field_type, result_slice.items(field), self_slice.items(field));
            }
        }
        return result;
    }

    /// `ctx` has the following method:
    /// `fn lessThan(ctx: @TypeOf(ctx), a_index: usize, b_index: usize) bool`
    pub fn sort(self: Self, ctx: anytype) void {
        const SortContext = struct {
            sub_ctx: @TypeOf(ctx),
            slice: Slice,

            pub fn swap(sc: @This(), a_index: usize, b_index: usize) void {
                inline for (fields) |field_info, i| {
                    if (@sizeOf(field_info.field_type) != 0) {
                        const field = @intToEnum(Field, i);
                        const ptr = sc.slice.items(field);
                        mem.swap(field_info.field_type, &ptr[a_index], &ptr[b_index]);
```

MultiArrayList



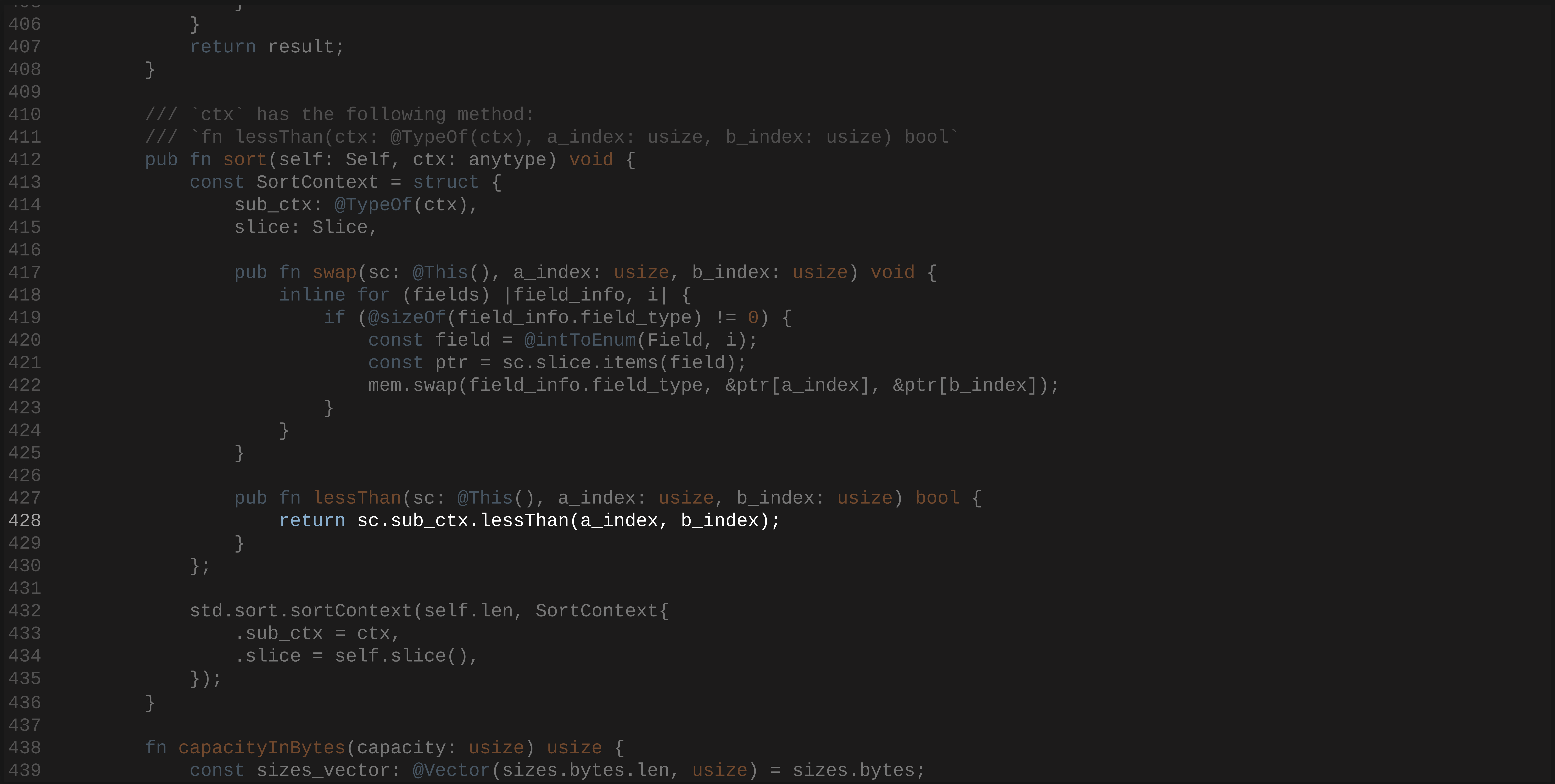
```
            const self_slice = self.slice();
            const result_slice = result.slice();
            inline for (fields) |field_info, i| {
                if (@sizeOf(field_info.field_type) != 0) {
                    const field = @intToEnum(Field, i);
                    mem.copy(field_info.field_type, result_slice.items(field), self_slice.items(field));
                }
            }
            return result;
        }

        /// `ctx` has the following method:
        /// `fn lessThan(ctx: @TypeOf(ctx), a_index: usize, b_index: usize) bool`
        pub fn sort(self: Self, ctx: anytype) void {
            const SortContext = struct {
                sub_ctx: @TypeOf(ctx),
                slice: Slice,

                pub fn swap(sc: @This(), a_index: usize, b_index: usize) void {
                    inline for (fields) |field_info, i| {
                        if (@sizeOf(field_info.field_type) != 0) {
                            const field = @intToEnum(Field, i);
                            const ptr = sc.slice.items(field);
                            mem.swap(field_info.field_type, &ptr[a_index], &ptr[b_index]);
                        }
                    }
                }

                pub fn lessThan(sc: @This(), a_index: usize, b_index: usize) bool {
                    return sc.sub_ctx.lessThan(a_index, b_index);
                }
            };

            std.sort.sortContext(self.len, SortContext{
                .sub_ctx = ctx,
```

MultiArrayList



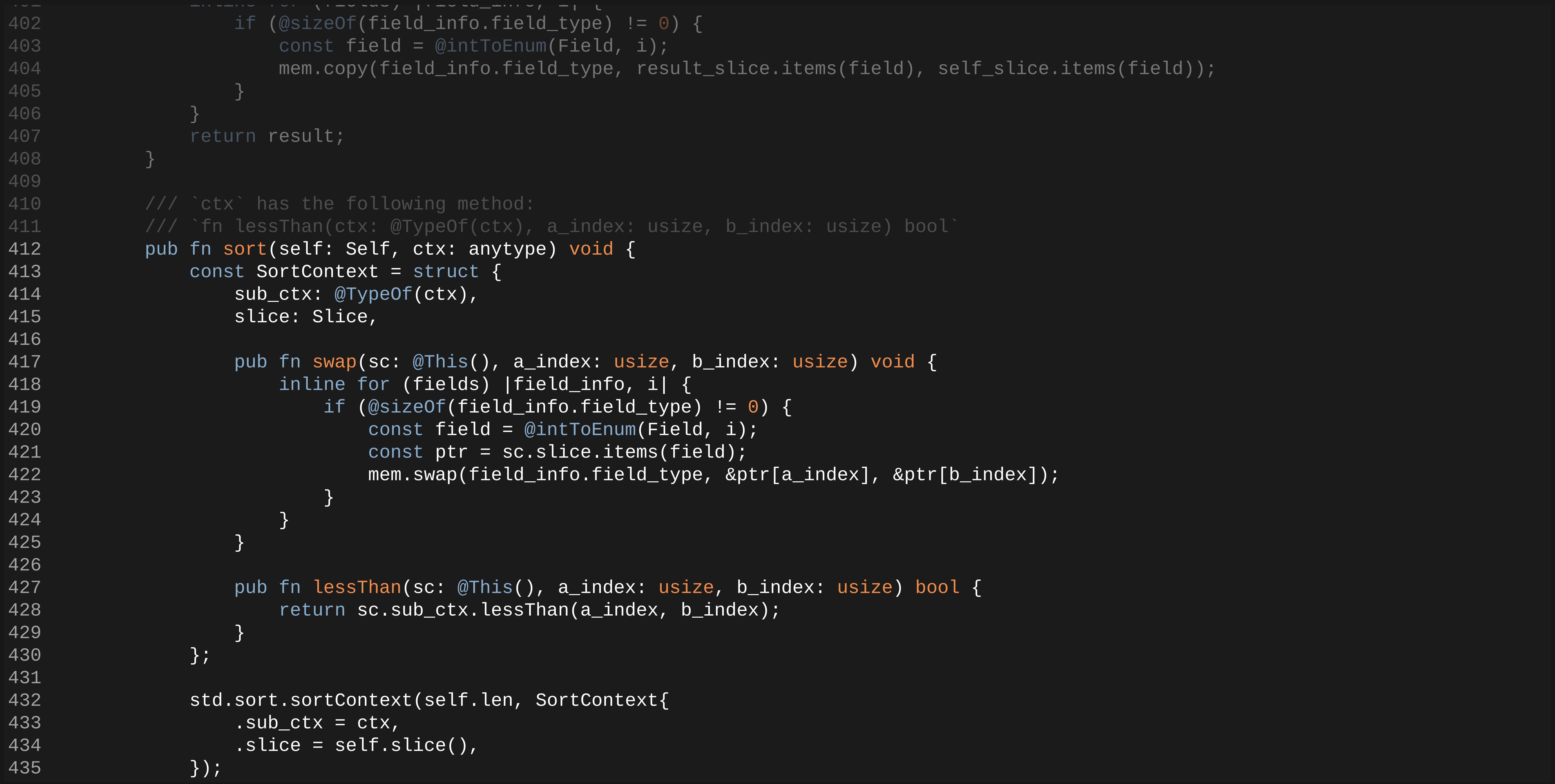
```
        /// Create a copy of this list with a new backing store,
        /// using the specified allocator.
        pub fn clone(self: Self, gpa: Allocator) !Self {
            var result = Self{};
            errdefer result.deinit(gpa);
            try result.ensureTotalCapacity(gpa, self.len);
            result.len = self.len;
            const self_slice = self.slice();
            const result_slice = result.slice();
            inline for (fields) |field_info, i| {
                if (@sizeOf(field_info.field_type) != 0) {
                    const field = @intToEnum(Field, i);
                    mem.copy(field_info.field_type, result_slice.items(field), self_slice.items(field));
                }
            }
            return result;
        }

        /// `ctx` has the following method:
        /// `fn lessThan(ctx: @TypeOf(ctx), a_index: usize, b_index: usize) bool`
        pub fn sort(self: Self, ctx: anytype) void {
            const SortContext = struct {
                sub_ctx: @TypeOf(ctx),
                slice: Slice,

                pub fn swap(sc: @This(), a_index: usize, b_index: usize) void {
                    inline for (fields) |field_info, i| {
                        if (@sizeOf(field_info.field_type) != 0) {
                            const field = @intToEnum(Field, i);
                            const ptr = sc.slice.items(field);
                            mem.swap(field_info.field_type, &ptr[a_index], &ptr[b_index]);
                        }
                    }
                }
```

MultiArrayList



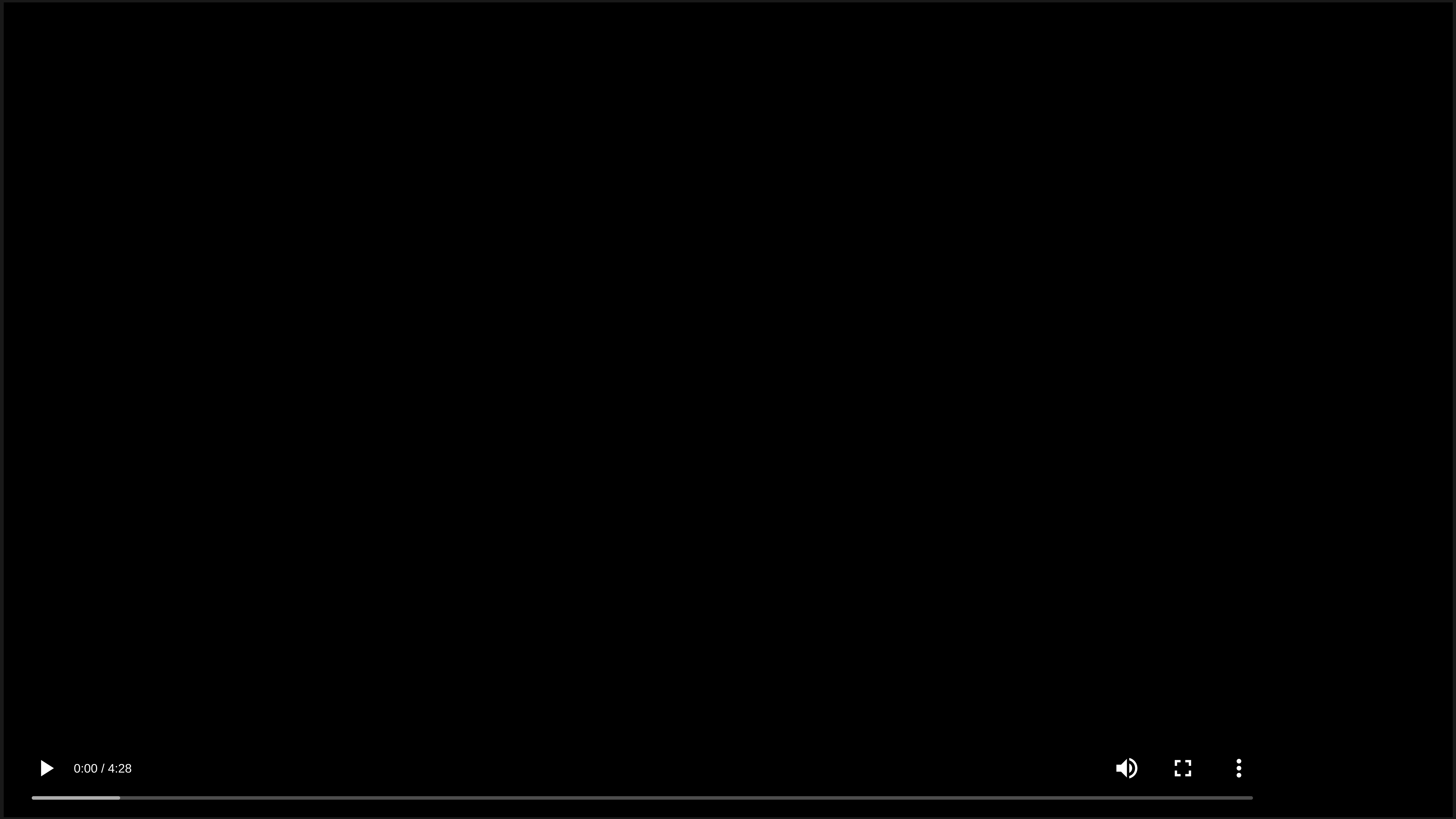
```
                }
            }
            return result;
        }

        /// `ctx` has the following method:
        /// `fn lessThan(ctx: @TypeOf(ctx), a_index: usize, b_index: usize) bool`
        pub fn sort(self: Self, ctx: anytype) void {
            const SortContext = struct {
                sub_ctx: @TypeOf(ctx),
                slice: Slice,

                pub fn swap(sc: @This(), a_index: usize, b_index: usize) void {
                    inline for (fields) |field_info, i| {
                        if (@sizeOf(field_info.field_type) != 0) {
                            const field = @intToEnum(Field, i);
                            const ptr = sc.slice.items(field);
                            mem.swap(field_info.field_type, &ptr[a_index], &ptr[b_index]);
                        }
                    }
                }

                pub fn lessThan(sc: @This(), a_index: usize, b_index: usize) bool {
                    return sc.sub_ctx.lessThan(a_index, b_index);
                }
            };

            std.sort.sortContext(self.len, SortContext{
                .sub_ctx = ctx,
                .slice = self.slice(),
            });
        }

        fn capacityInBytes(capacity: usize) usize {
            const sizes_vector: @Vector(sizes.bytes.len, usize) = sizes.bytes;
```

MultiArrayList



```
            inline for (fields) |field_info, i| {
                if (@sizeOf(field_info.field_type) != 0) {
                    const field = @intToEnum(Field, i);
                    mem.copy(field_info.field_type, result_slice.items(field), self_slice.items(field));
                }
            }
            return result;
        }

        /// `ctx` has the following method:
        /// `fn lessThan(ctx: @TypeOf(ctx), a_index: usize, b_index: usize) bool`
        pub fn sort(self: Self, ctx: anytype) void {
            const SortContext = struct {
                sub_ctx: @TypeOf(ctx),
                slice: Slice,

                pub fn swap(sc: @This(), a_index: usize, b_index: usize) void {
                    inline for (fields) |field_info, i| {
                        if (@sizeOf(field_info.field_type) != 0) {
                            const field = @intToEnum(Field, i);
                            const ptr = sc.slice.items(field);
                            mem.swap(field_info.field_type, &ptr[a_index], &ptr[b_index]);
                        }
                    }
                }

                pub fn lessThan(sc: @This(), a_index: usize, b_index: usize) bool {
                    return sc.sub_ctx.lessThan(a_index, b_index);
                }
            };

            std.sort.sortContext(self.len, SortContext{
                .sub_ctx = ctx,
                .slice = self.slice(),
            });
```

C Integration



SUMMARY

SUMMARY

- **Zig Software Foundation** is a non-profit organization dedicated to improving the craft of software engineering as a whole.

Speaker notes

Our goal is to increase the value of the commons by orders of magnitude. We sincerely invite you to sit tight and reap these benefits, and give nothing in return.

Enjoy these benefits even if you never use the toolchain.

SUMMARY

- **Zig Software Foundation** is a non-profit organization dedicated to improving the craft of software engineering as a whole.
- Zig is a C/C++ **compiler toolchain** and **build system** that can be used to simplify maintenance of your existing projects.

Speaker notes

Enjoy these benefits even if you never use the language.

SUMMARY

- **Zig Software Foundation** is a non-profit organization dedicated to improving the craft of software engineering as a whole.
- Zig is a C/C++ **compiler toolchain** and **build system** that can be used to simplify maintenance of your existing projects.
- Zig is a **simple, powerful programming language** that excels in the most demanding environments.

Speaker notes

The number of Zig users is growing rapidly.

SUMMARY

- **Zig Software Foundation** is a non-profit organization dedicated to improving the craft of software engineering as a whole.
- Zig is a C/C++ **compiler toolchain** and **build system** that can be used to simplify maintenance of your existing projects.
- Zig is a **simple, powerful programming language** that excels in the most demanding environments.

Consider sponsoring us! ziglang.org/zsf

Speaker notes

The number of Zig users is growing rapidly.

QA

Andrew Kelley

andrewkelley.me

exit(0)

Andrew Kelley

andrewkelley.me