



MoonBit

大语言模型时代的软件开发起点

叶子航



极客邦科技 2024 年会议规划

促进软件开发及相关领域知识与创新的传播



访问大会官网



参会咨询

目录

- 月兔开发平台介绍
- 大模型在大型开发平台项目上的主要难点
- 月兔设计理念与关键思考

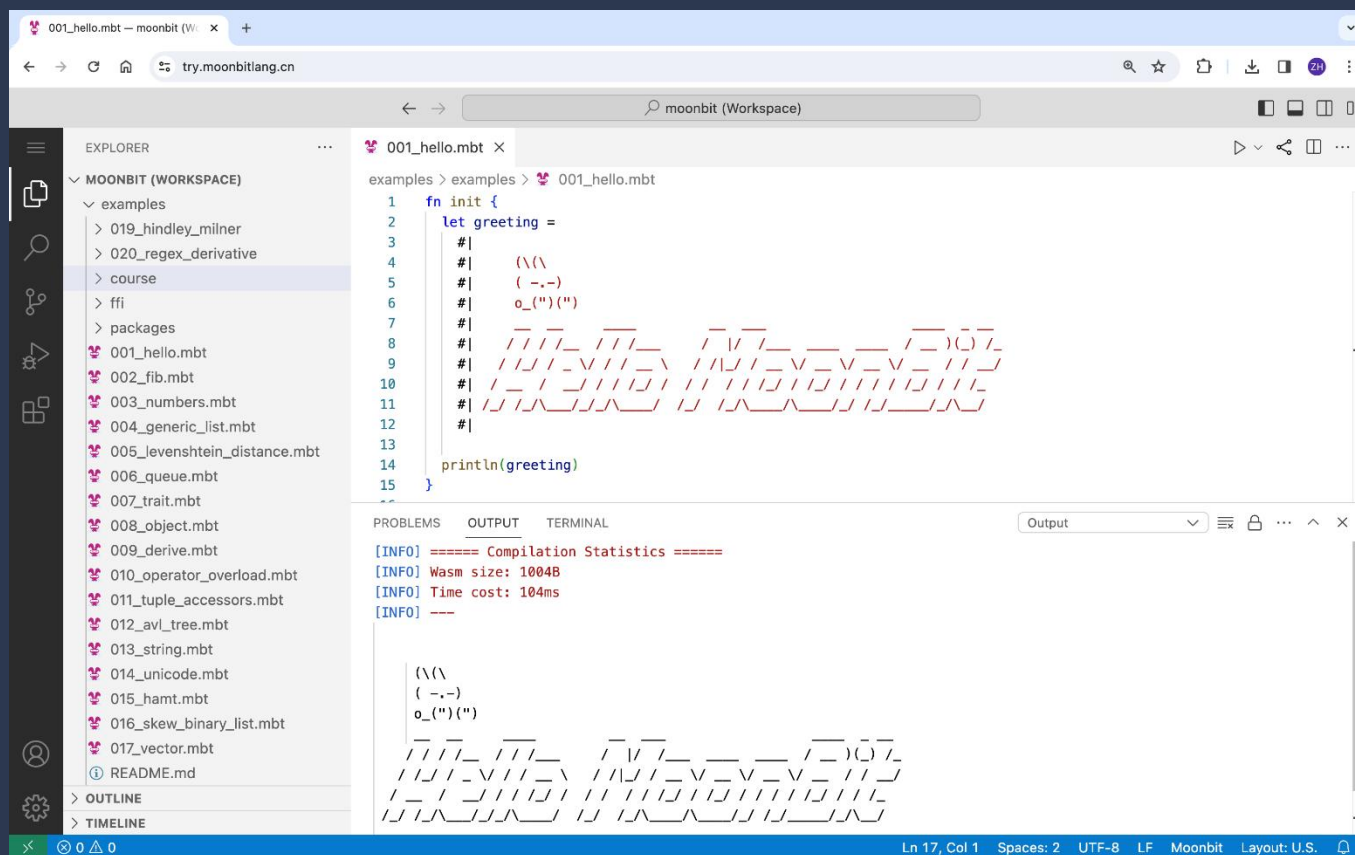


MoonBit

- AI时代下的云原生开发平台

- 多范式编程语言

- 全工具链开发





MoonBit

- 月兔 集成环境
- 月兔 调试器

MoonBit 支持
云原生调试

MoonBit 项目展示

MoonBit Gallery

Welcome to the MoonBit gallery.



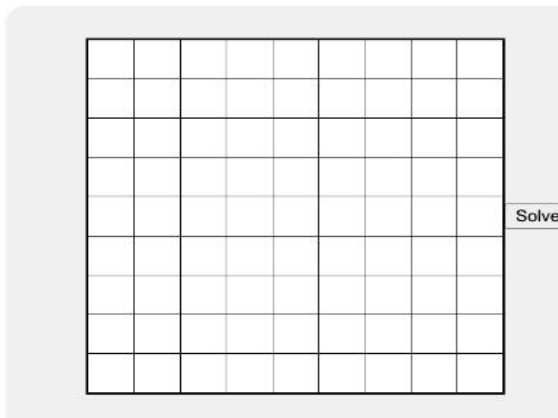
Tetris

A traditional tetris game development in MoonBit using web canvas API. You can change the MoonBit code and see the result on the fly.



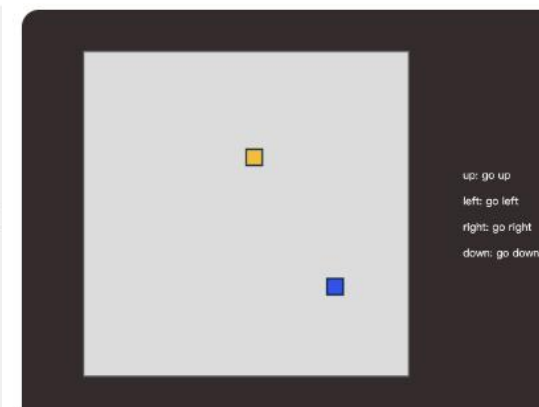
Mario

A mario game development in MoonBit using web canvas API. You can change the MoonBit code and see the result on the fly.



Sudoku

A sudoku solver written in MoonBit



Snake

使用 MoonBit 编写的贪吃蛇游戏

构建系统

构建系统



基于 n2 (Rust)

工具链

100%自研核心

编译器

OCaml

IDE

测试

静态分析

Coverage

MoonBit Docs

...

Llama2



AI



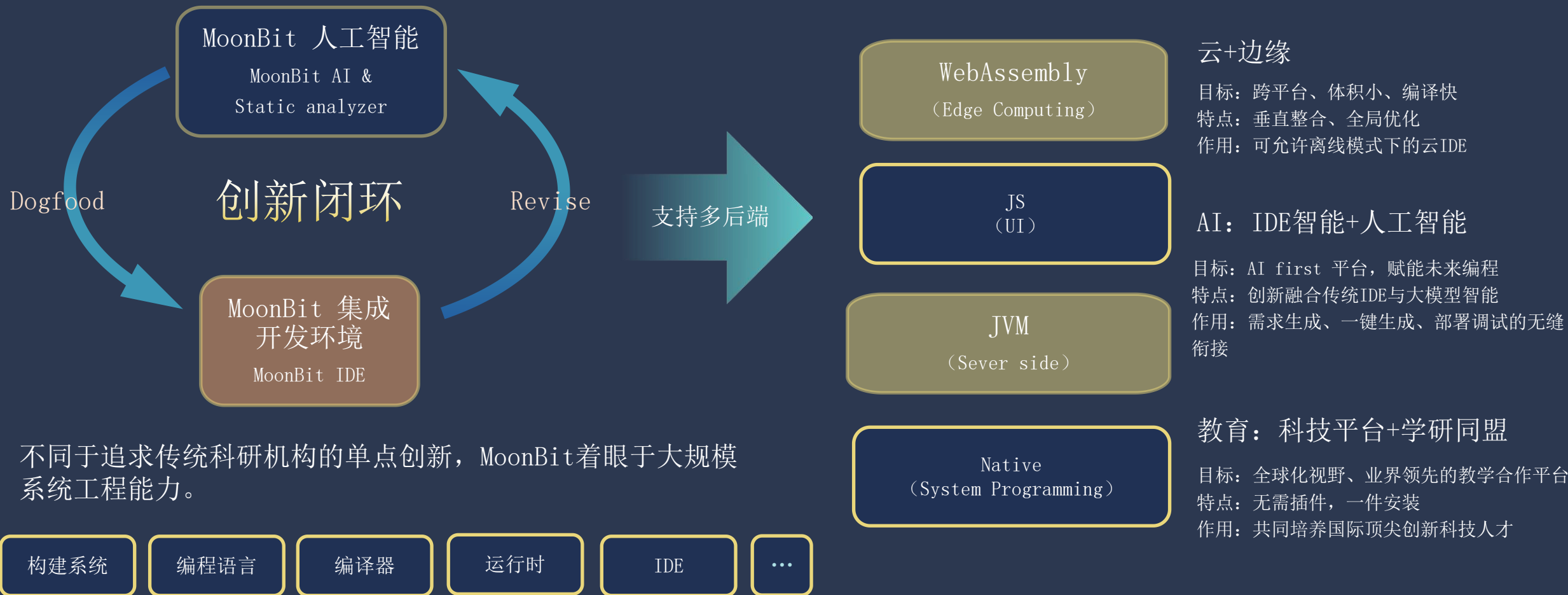
IDE



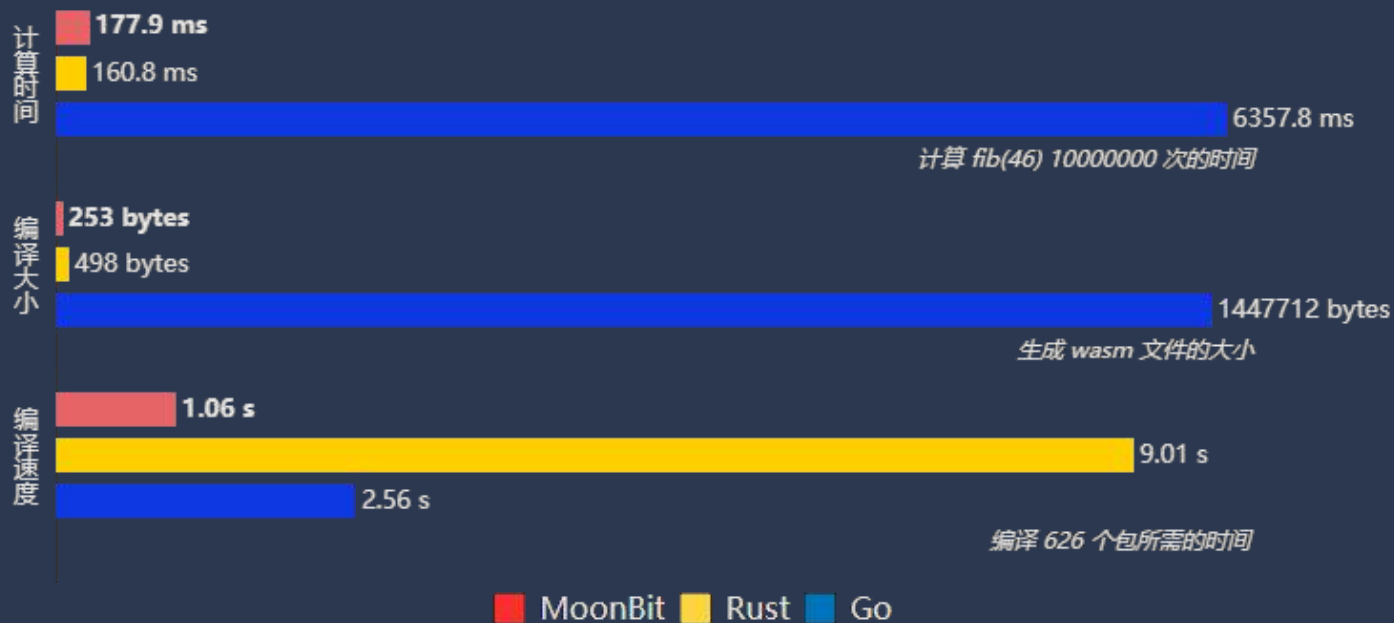
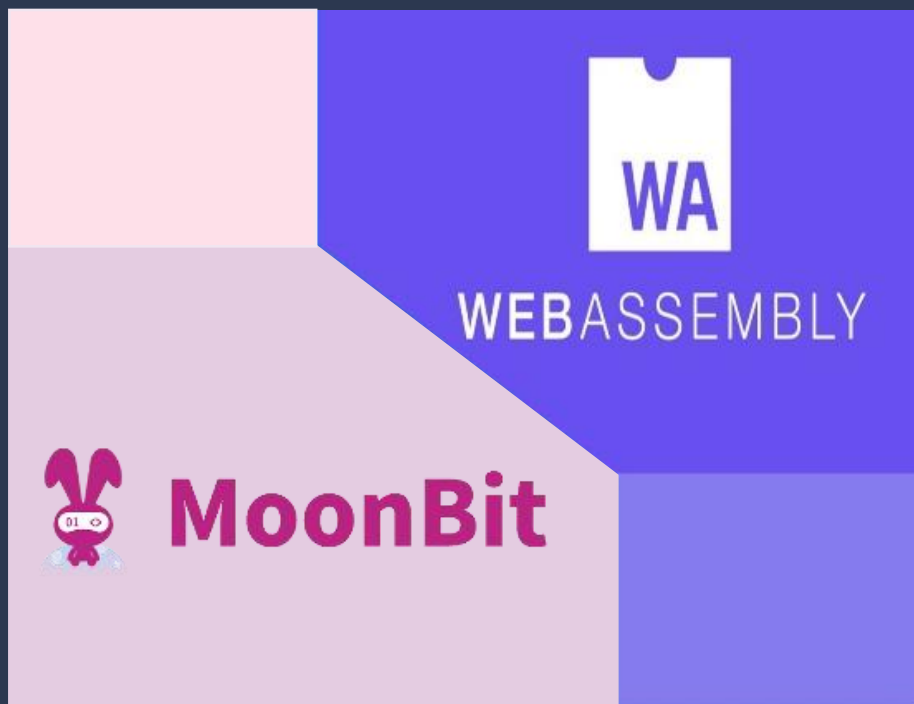
llama.cpp



目标：世界级编程语言及工具链



MoonBit 平台



数据来源: <https://github.com/moonbitlang/moonbit-docs/tree/main/benchmark/fibonacci>

MoonBit拥抱AI：不经意间撞上的风口

盖茨访谈OpenAI Sam Altman
Sam谈他对AI最兴奋的事情
以及未来发展的期待



but coding is probably the area,
the single area from a productivity gain we're most
excited about today,
massively deployed and at scaled usage.

MoonBit: Explore the Design of an AI-Friendly Programming Language

Anonymous Author(s)*

ABSTRACT



LLM4Code

@llm4code

The 1st International Workshop on Large Language Models for Code
Co-located with @ICSEconf 2024

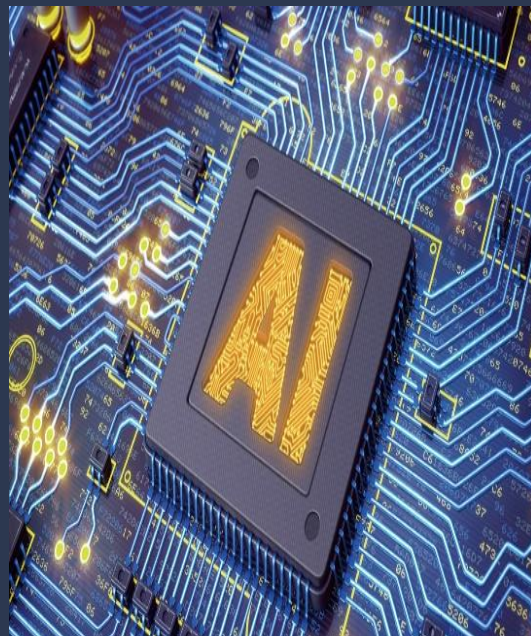
📍 Lisbon, Portugal 🔗 llm4code.github.io 📅 Joined October 2023

MoonBit团队发表的论文
入选首届国际大型语言模
型应用于代码研讨会
LLM4Code 2024

大模型下语言需要解决的问题



安全性



减少幻觉

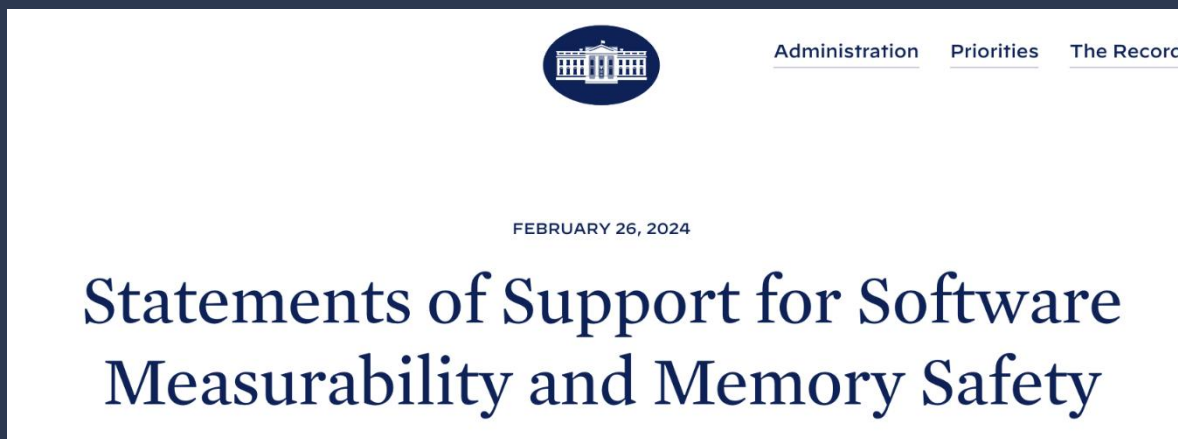


原生的AI人机交互

大模型的安全性

- 软件安全性

FBI—Warning White House Warning
使用内存安全的编程语言



大模型的幻觉

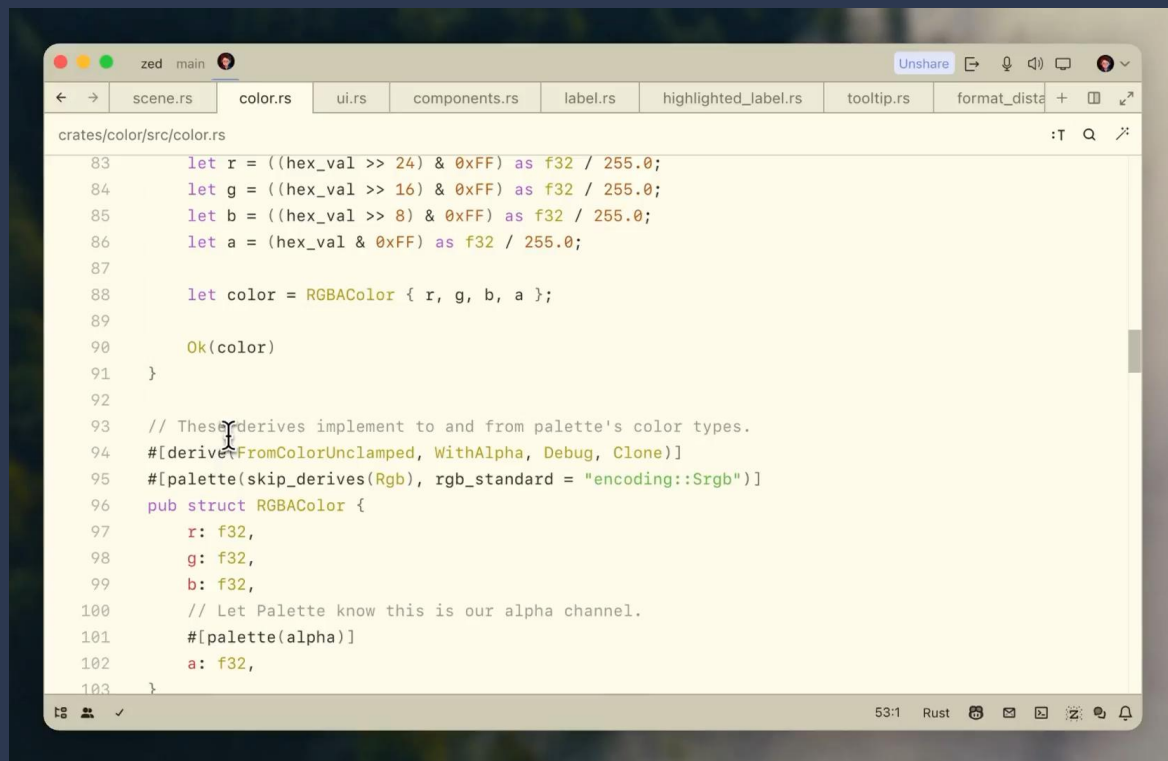
- 对于新的编程语言，如何生成高质量代码

User Message: Print hello world for 5 times.

Assistant:

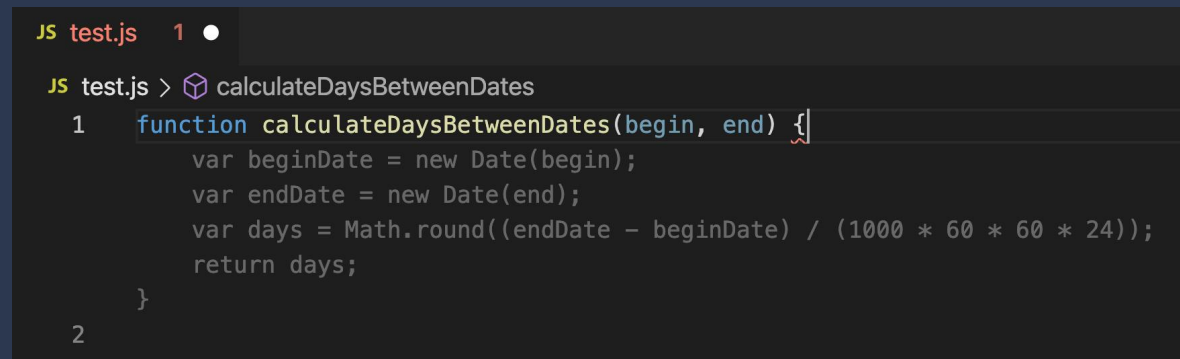
```
fn init {  
    for i in 1..6 {  
        print("Hello world!")  
    }  
}
```


大模型的交互



The screenshot shows the Zed code editor with a file explorer at the top displaying a project structure with files like scene.rs, color.rs, ui.rs, components.rs, label.rs, highlighted_label.rs, tooltip.rs, and format_dist.rs. The main editor window shows the content of color.rs, which includes Rust code for handling hex colors and defining an RGBAColor struct. The code is syntax-highlighted, and the editor interface includes standard window controls and a status bar at the bottom.

```
83 let r = ((hex_val >> 24) & 0xFF) as f32 / 255.0;
84 let g = ((hex_val >> 16) & 0xFF) as f32 / 255.0;
85 let b = ((hex_val >> 8) & 0xFF) as f32 / 255.0;
86 let a = (hex_val & 0xFF) as f32 / 255.0;
87
88 let color = RGBAColor { r, g, b, a };
89
90 Ok(color)
91 }
92
93 // These derives implement to and from palette's color types.
94 #[derive(FromColorUnclamped, WithAlpha, Debug, Clone)]
95 #[palette(skip_derives(Rgb), rgb_standard = "encoding:Srgb")]
96 pub struct RGBAColor {
97     r: f32,
98     g: f32,
99     b: f32,
100     // Let Palette know this is our alpha channel.
101     #[palette(alpha)]
102     a: f32,
103 }
```



The screenshot shows a JavaScript code editor with a file named test.js. The code defines a function calculateDaysBetweenDates. The editor has a dark theme and shows line numbers on the left side of the code.

```
JS test.js > calculateDaysBetweenDates
1 function calculateDaysBetweenDates(begin, end) {
    var beginDate = new Date(begin);
    var endDate = new Date(end);
    var days = Math.round((endDate - beginDate) / (1000 * 60 * 60 * 24));
    return days;
}
2
```

<https://code.visualstudio.com/docs/copilot/overview>

<https://zed.dev/>

MoonBit-安全性

- MoonBit的快速静态分析能力、设计上就考虑到了代码三重的安全保障，比如安全的类型系统、WASM的安全沙箱机制，以及我们的死代码删除技术

安全的类型系统

MoonBit通过结构化接口和类型系统的明确性，允许大模型更加准确地识别代码的模式和结构，进而生成更加准确和高效的代码

Wasm安全沙箱

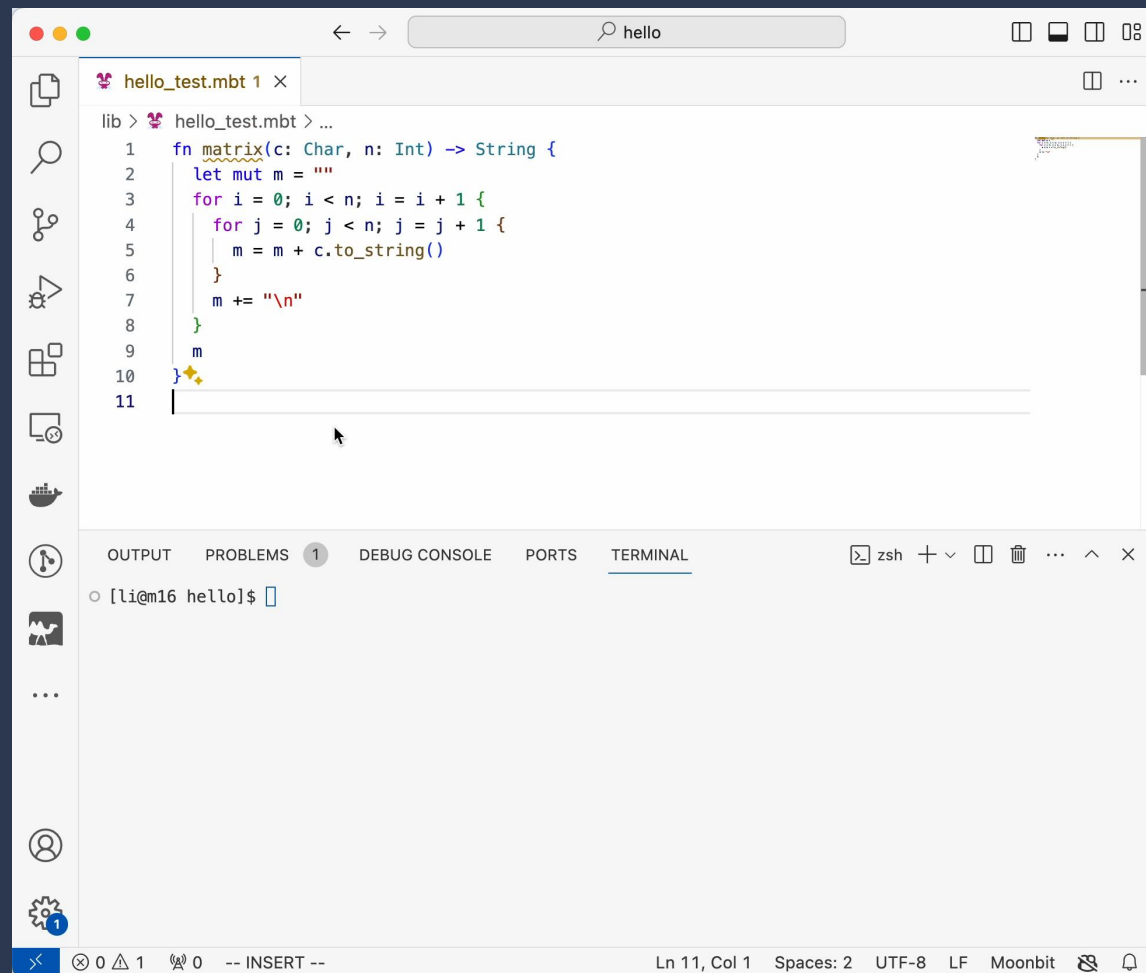
WebAssembly是一种开放的二进制指令格式，用于在现代网络浏览器中以沙箱环境的方式安全、快速地运行代码。它是一种低级别的字节码，类似于Java字节码或.NET的Common Intermediate Language (CIL)，但以Web为中心

死代码删除

MoonBit专为高效的死代码消除而设计，并且具有为了轻松删除死代码而设计的标准库结构。通过全局代码优化，MoonBit显著减小了最终生成的Wasm代码大小，提高了安全性并减少了攻击面，并且确保在无服务器计算环境中快速启动

可解释性、可验证性、可测试性

- 内置测试功能



The screenshot shows a code editor window with a file named `hello_test.mbt`. The code is a Rust function `matrix` that takes a character `c` and an integer `n`, and returns a string. The function uses nested loops to build a string `m` by concatenating the character `c` repeated `n` times, separated by newlines. The terminal window at the bottom shows the command prompt `[li@16 hello]$`.

```
lib > hello_test.mbt > ...  
1 fn matrix(c: Char, n: Int) -> String {  
2   let mut m = ""  
3   for i = 0; i < n; i = i + 1 {  
4     for j = 0; j < n; j = j + 1 {  
5       m = m + c.to_string()  
6     }  
7     m += "\n"  
8   }  
9   m  
10 }  
11
```


OUTPUT PROBLEMS 1 DEBUG CONSOLE PORTS TERMINAL
[li@16 hello]\$

可解释性、可验证性、可测试性

- AI参与的开发流程

- 提供测试覆盖率统计工具
- 提供测试功能
- 愿景：AI自主迭代


```
112 + fn op_equal[T : Eq](self : Queue[T], other : Queue[T]) -> Bool {
113 +   if self.length != other.length || self.first != self.first || self.last != other.last {
114 +     return false
115 +   }
116 +   return true
```


 **coderabbitai** (bot) 14 minutes ago Contributor ...

The `op_equal` method for comparing two queues seems to contain a logical error. It incorrectly compares `self.first` with itself instead of comparing `self.first` with `other.first` and `self.last` with `other.last`.

```
- if self.length != other.length || self.first != self.first || self.last != self.last {
+ if self.length != other.length || self.first != other.first || self.last != other.last {
```

► Committable suggestion



 **codecov** (bot) commented 15 minutes ago · edited

Codecov Report

Attention: Patch coverage is 90.47619% with 2 lines in your changes are missing coverage. Please review.


Project coverage is 80.78%. Comparing base (86416c8) to head (fc3056f).

Files	Patch %	Lines
queue/queue.mbt	90.47%	2 Missing 🚩

► Additional details and impacted files

[View full report in Codecov by Sentry.](#)

Have feedback on the report? [Share it here.](#)



可解释性、可验证性、可测试性



MoonBit—减少幻觉

```
/// A simple 2-D geometry library
```

```
/// Computes Euclidean distance between two points.
```

```
pub fn distance(p1: Point, p2: Point) -> Double {
```

```
    /// All content below is invisible to our large language model.
```

```
    /// A 2-D Point.
```

```
    pub struct Point {  
        coord_x: Double  
        coord_y: Double  
    }
```

```
    /// Use FFI to take the square root of a double.
```

```
    pub fn take_sqrt(self: Double) -> Double = "math" "sqrt"
```

▶ 局部重采样

pub $\left[\begin{array}{l} x \\ y \\ fn \end{array} \right.$

▶ 全局重采样

```
pub fn sum (p:Point) {
```

p. $\left[\begin{array}{l} x \\ y \\ co \end{array} \right.$

```
    struct Point  
    {  
        coord_x: Int  
    }
```

MoonBit-平坦化设计

- 平坦化设计
- 适配Transformer

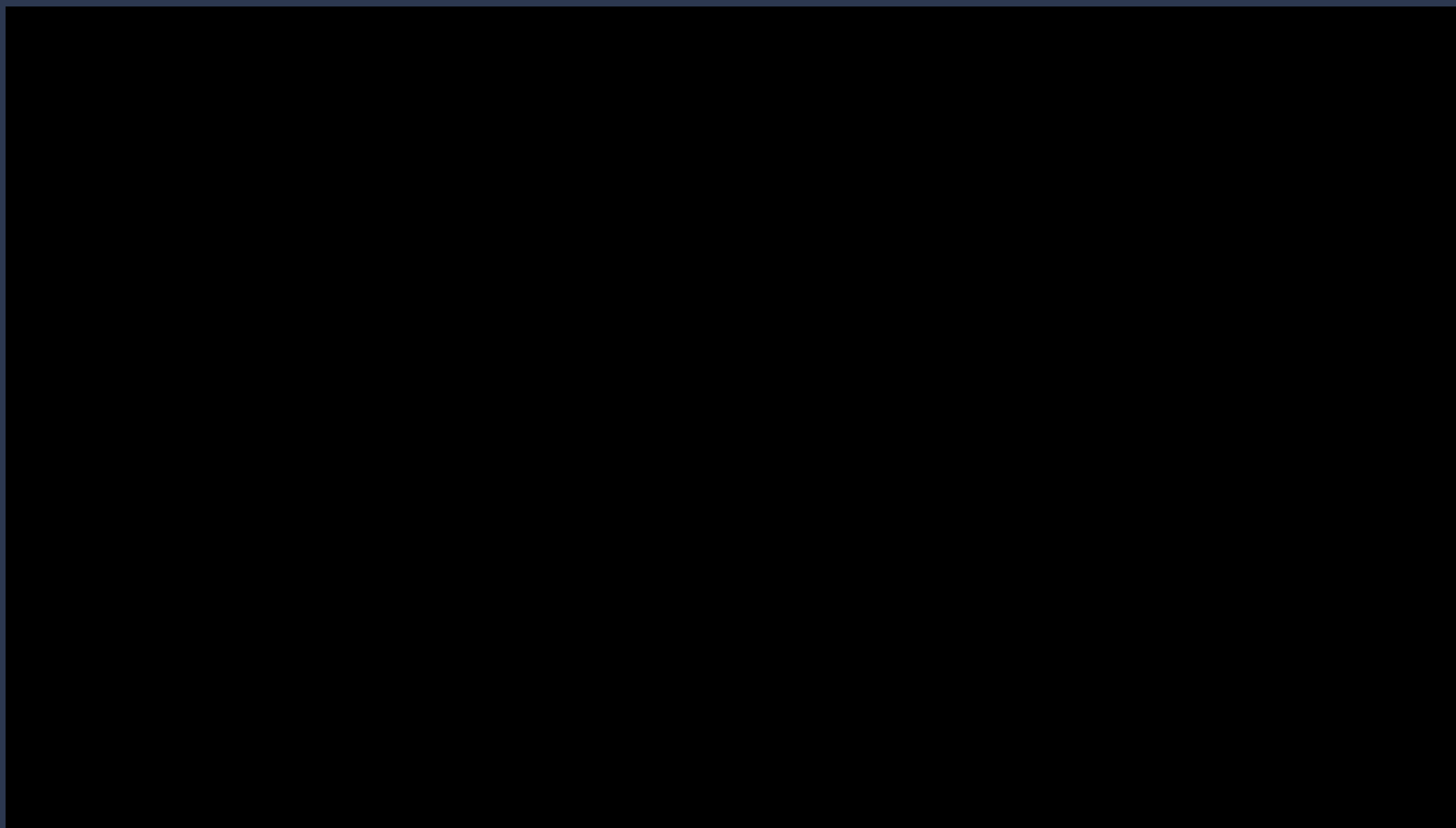
```
struct Llama { ... }  
trait Agent { ... }  
impl LLM for Llama ...  
impl Agent for Llama {  
    fn act(&self) ...  
    fn info(...) ...  
    fn ...  
    fn think(&mut self) {  
        self.generate  
    }  
}
```

(a) Rust

```
struct Llama { ... }  
interface Agent { ... }  
fn act(self: Self) ...  
fn info(...) ...  
fn ...  
fn generate(...) ...  
fn think(self: Self) {  
    self.generate  
}
```

(b) MoonBit

MoonBit AI



MoonBit-原生的AI人机交互

- Block-based 编辑器

```
struct Llama { ... }  
interface Agent { ... }  
fn act(self: Self) ...  
fn info(...) ...  
fn ...  
fn generate(...) ...  
fn think(self: Self) {  
    self.generate
```

(b) MoonBit

MoonBit代码公开路线图



总结

- 全产品协同设计
 - 编程语言：语法、类型系统、……
 - 开发工具：编译器、构建系统、IDE、……
 - 生态构建：测试、代码覆盖率工具、……
- 严肃认真



极客邦科技 2024 年会议规划

促进软件开发及相关领域知识与创新的传播



访问大会官网



参会咨询

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software