

Как (не) написать фронтенд на Rust

О себе

Ilya Bogdanov

JavaScript

~~Rust~~ Developer



Кто мы и что делаем

Как можно написать фронтенд на Rust?

Как можно не написать фронтенд на Rust?

Конец?

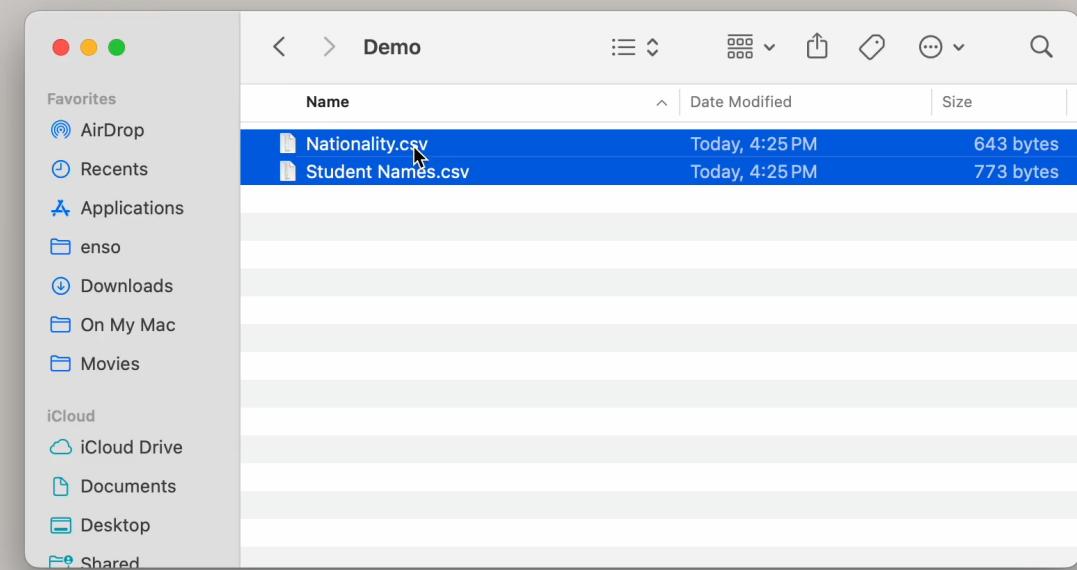
Проблема

ID	Name
1	Emma Johnson
2	Ethan Wilson
3	Olivia Davis
4	Liam Miller
...	...

ID	Nationality	Classroom
1	USA	A
2	Canada	A
3	France	B
4	Germany	E
...

Classroom	Nationality	Count
A	USA	5
A	Canada	2
A	France	3
B	France	1
...

Решение



Enso Language

ФУНКЦИОНАЛЬНЫЙ

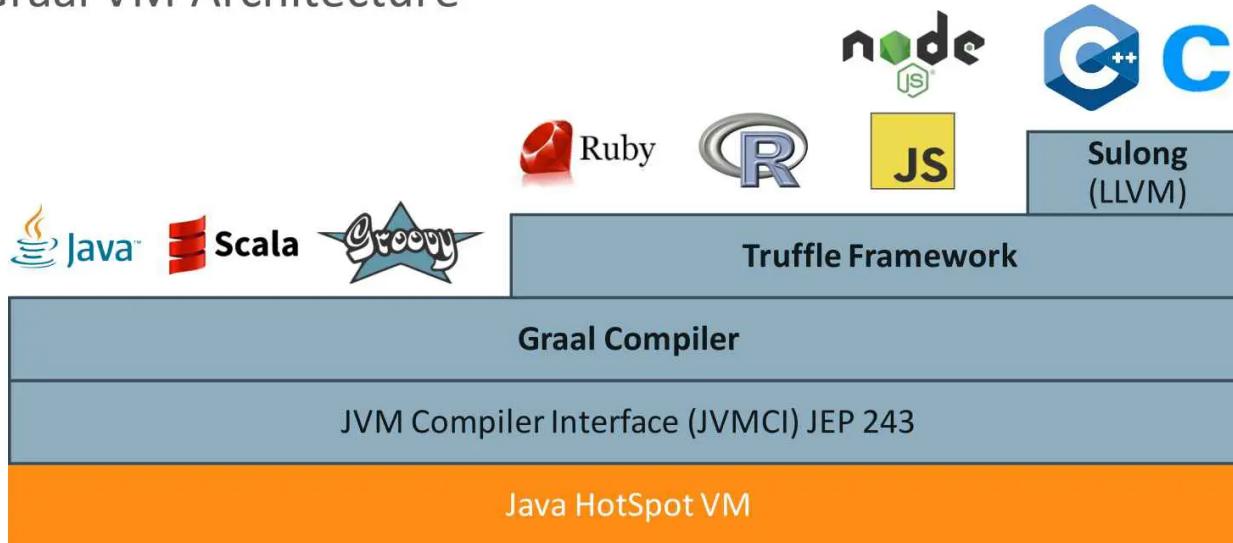
```
c > ≡ Complex.ensa
1   from Standard.Base import all
2
3   type Complex
4       ....Value re:Number im:Number
5
6   .....new re=0 im=0 = Complex.Value re im
7
8   .....+ self (that:Complex) =
9       .....re =self.re + that.re
10      .....im = self.im + that.im
11      .....Complex.Value re im
12
13 main =
14     .....v1 = (Complex.new 1 2) + (Complex.new 3 4)
15     .....v2 = (Complex.new 1 5) + 3.to Complex
16     .....v3 = (Complex.new 1 5) + 3
17     .....[ v1, v2, v3 ]
18
19 Complex.from (that: Number) = Complex.new that 0
20
```

Текстовое + визуальное представление

```
src > Main.enso
1  from Standard.Base import all
2  from Standard.Table import all
3  from Standard.Database import all
4  from Standard.AWS import all
5  import Standard.Visualization
6
7 main = .
8 ....operator1 = enso_project.data/"Nationality.csv" . read
9 ....operator2 = enso_project.data/"Student Names.csv" . read
10 ....operator3 = operator2.join operator1 Join_Kind.Inner [(Join_Condition.Equals 'Student ID')]
11 ....operator4 = operator3.aggregate [(Aggregate_Column.Group_By 'Classroom'), (Aggregate_Column.Group_By 'Nationality'),
12      (Aggregate_Column.Count)]
13 ....operator5 = operator4.order_by [(Sort_Column.Name 'Classroom')]
14
```

GraalVM

Graal VM Architecture



ORACLE

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. |

13

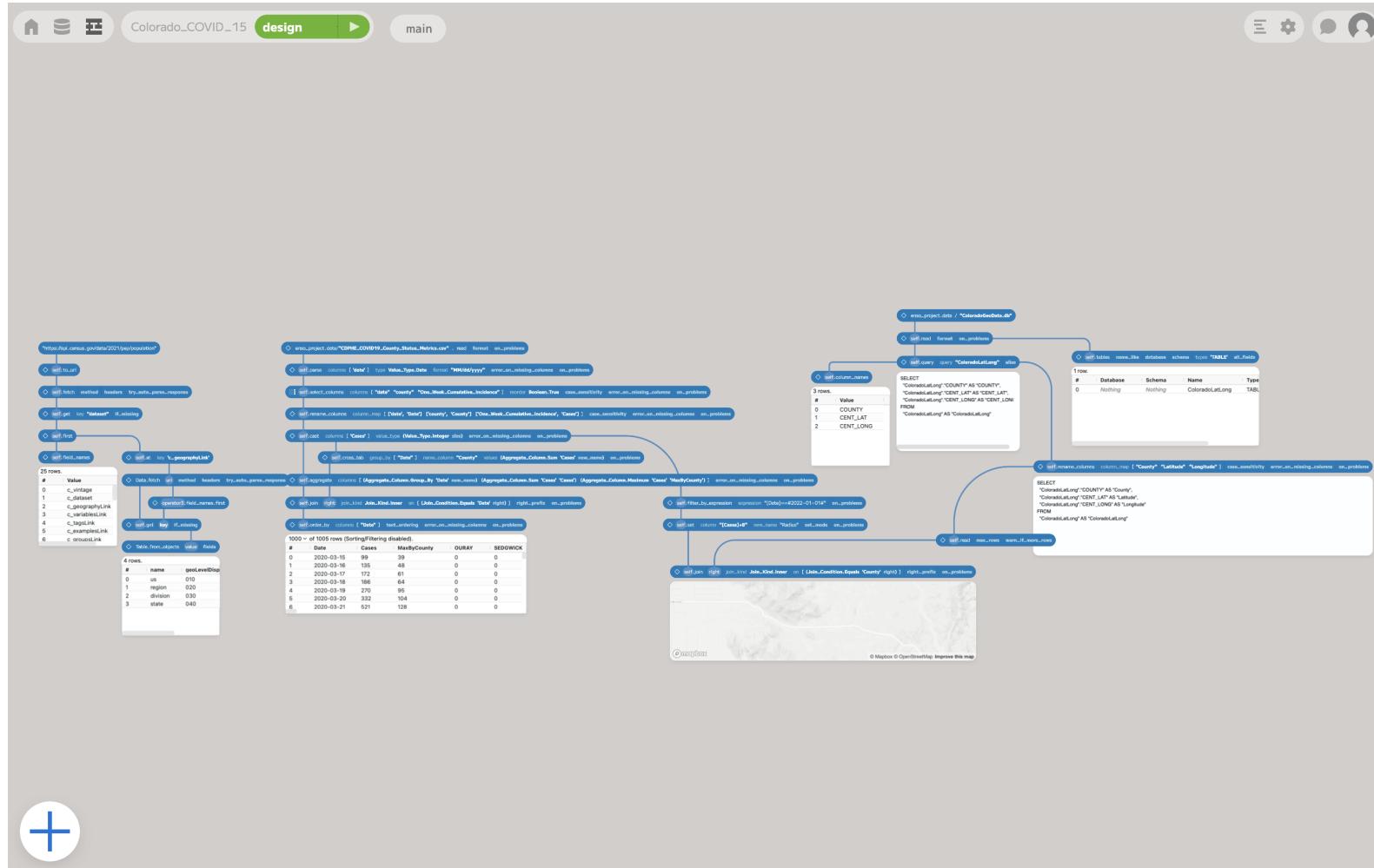
<https://rieckpil.de/whatis-graalvm/>

Polyglot

```
rc > ≡ Polyglot.enso
1  type Foo
2  |....Foo_Data vec
3
4  compare_tco a b = case a.vec.length == b.vec.length of
5  |....False -> a.vec.length . compare_to b.vec.length
6  |....True -> cmp = (a.vec.at ix) . compare_to (b.vec.at ix)
7
8  foreign js generate_js_array = """
9  |....return [1, 2, 3, 4, 5]
10 |"""
11
12 foreign python generate_py_array = """
13 |....return [1, 2, 3, 4, None]
14 |"""
```

```
rc > ≡ Polyglot2.enso
1  polyglot java import java.lang.System as Java_System
2
3  exit : Integer -> Nothing
4  exit code = Java_System.exit code
5
```

Enso IDE



Требования к техническому стеку

Требования к техническому стеку

1. Гибкость и кастомизируемость UI

Требования к техническому стеку

1. Гибкость и кастомизируемость UI
2. Производительность

Требования к техническому стеку

1. Гибкость и кастомизируемость UI
2. Производительность
3. Кроссплатформенность

Кто мы и что делаем

**Как можно написать
фронтенд на Rust?**

Как можно не написать фронтенд на Rust?

Конец?

WebAssembly (WASM)

WebAssembly (WASM)

- Стандарт VM и байт-кода к ней

WebAssembly (WASM)

- Стандарт VM и байт-кода к ней
- Работает во всех современных браузерах

WebAssembly (WASM)

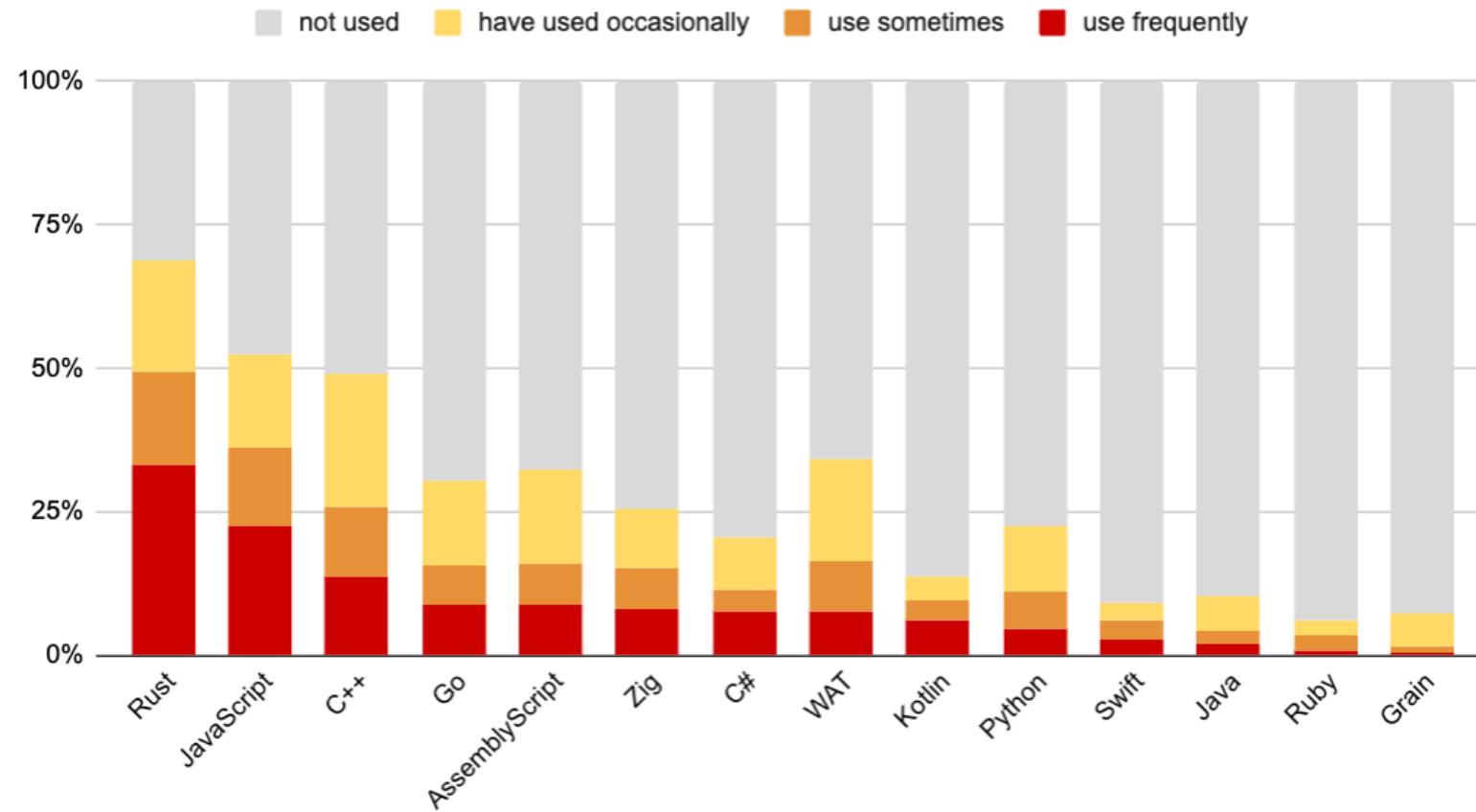
- Стандарт VM и байт-кода к ней
- Работает во всех современных браузерах
- Быстрый

WebAssembly (WASM)

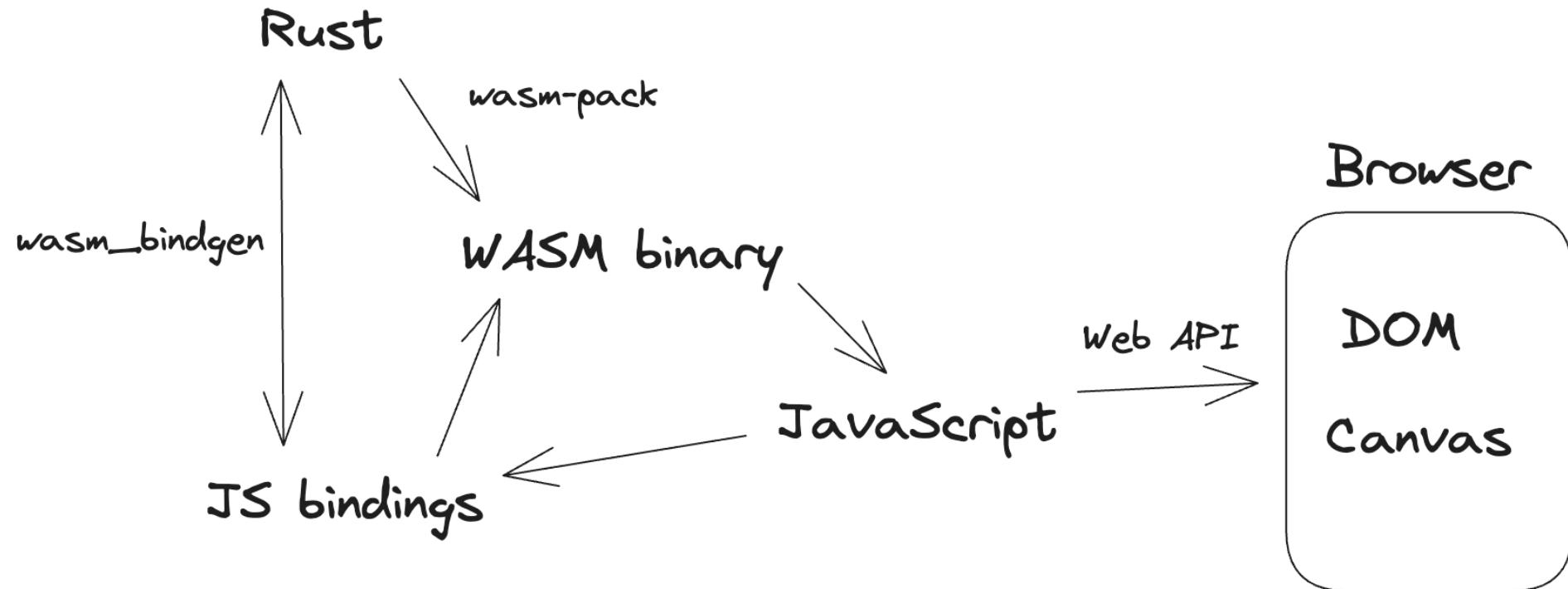
- Стандарт VM и байт-кода к ней
- Работает во всех современных браузерах
- Быстрый
- Безопасный

Rust – самый популярный язык для WASM

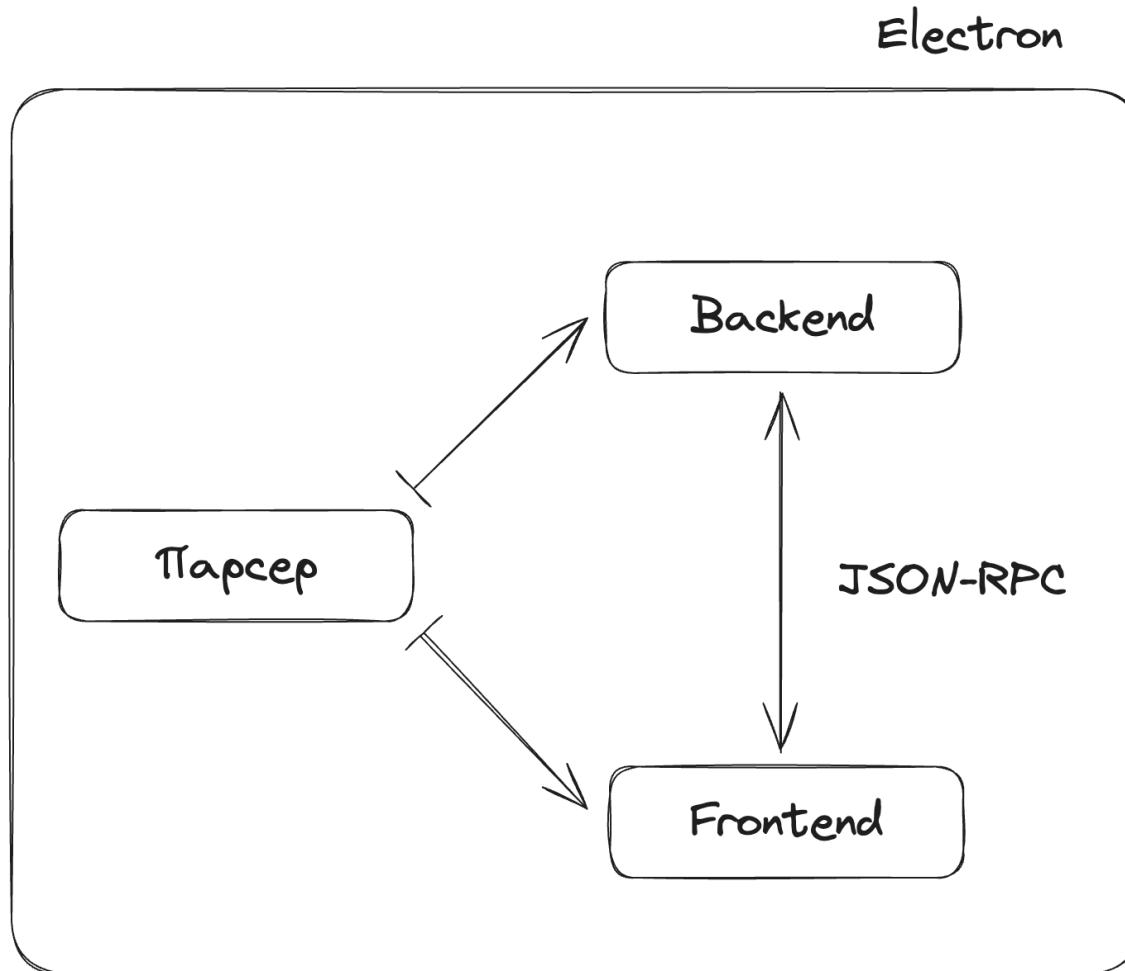
Current WebAssembly language usage (n=303)



<https://blog.scottlogic.com/2023/10/18/the-state-of-webassembly-2023.html>



Архитектура



Парсер

Парсер

- Scala → Rust

Парсер

- Scala → Rust
- 100 Мб → 4 Мб

Парсер

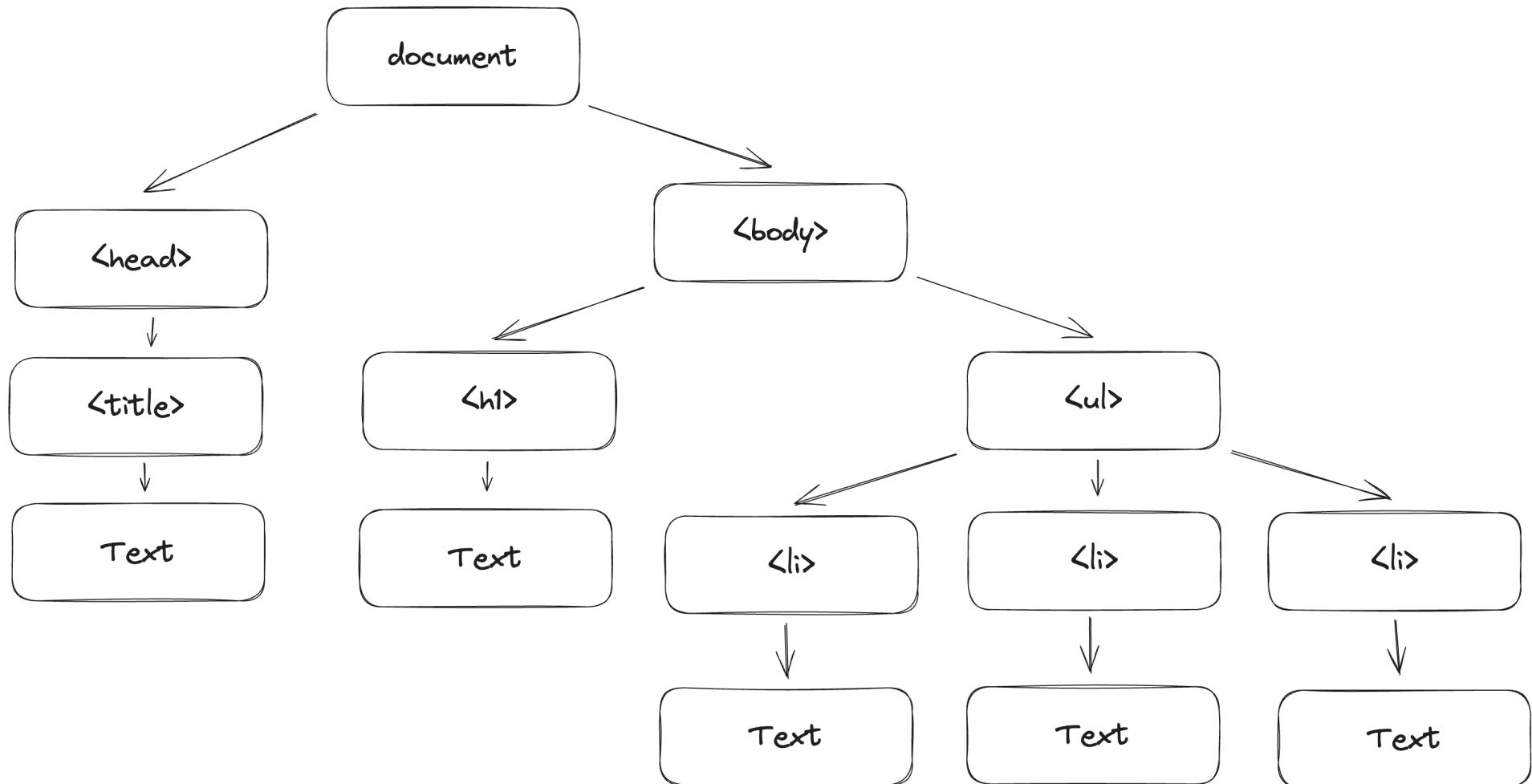
- Scala → Rust
- 100 Мб → 4 Мб
- fn parse(string: &str) -> Result<Ast>

Парсер

- Scala → Rust
- 100 Мб → 4 Мб
- fn parse(string: &str) -> Result<Ast>
- JNI-обертка для использования в бекенде

EnsoGL

DOM



```
// Get the element with id "myElement"
let element = document.getElementById("myElement");

// Change the text content of the element
element.textContent = "Hello, World!";

// Change the background color of the element
element.style.backgroundColor = "blue";
```



Canvas

```
function draw() {  
  const canvas = document.getElementById("canvas");  
  if (canvas.getContext) {  
    const ctx = canvas.getContext("2d");  
    ctx.beginPath();  
    ctx.arc(75, 75, 50, 0, Math.PI * 2, true); // Outer circle  
    ctx.moveTo(110, 75);  
    ctx.arc(75, 75, 35, 0, Math.PI, false); // Mouth (clockwise)  
    ctx.moveTo(65, 65);  
    ctx.arc(60, 65, 5, 0, Math.PI * 2, true); // Left eye  
    ctx.moveTo(95, 65);  
    ctx.arc(90, 65, 5, 0, Math.PI * 2, true); // Right eye  
    ctx.stroke();  
  }  
}
```







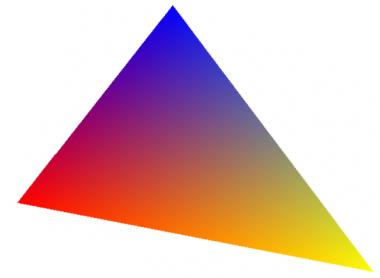
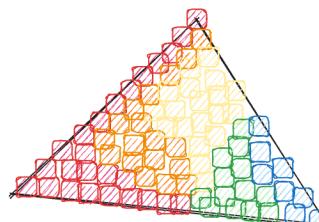
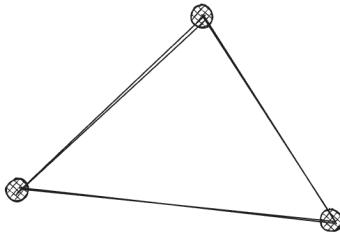
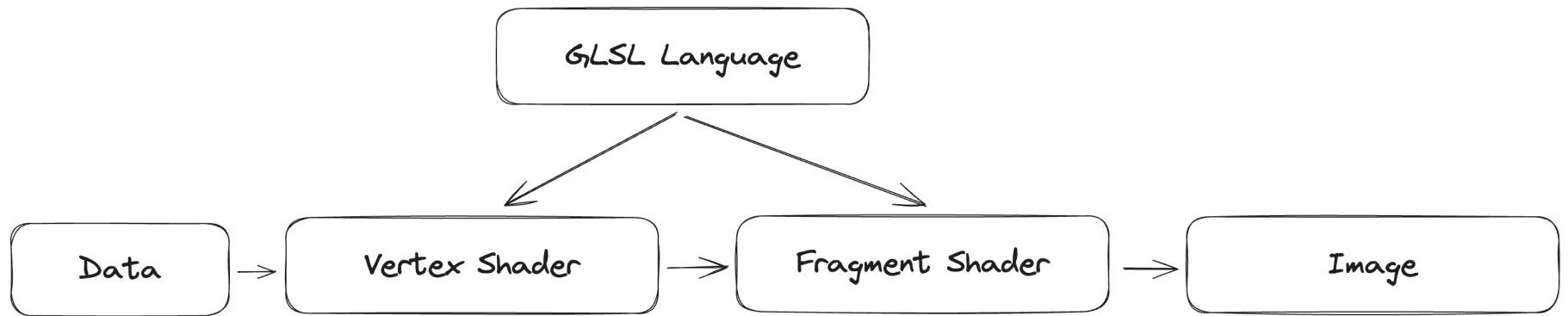
- Огрызок OpenGL ES

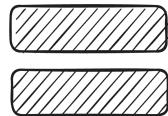
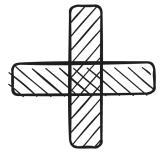
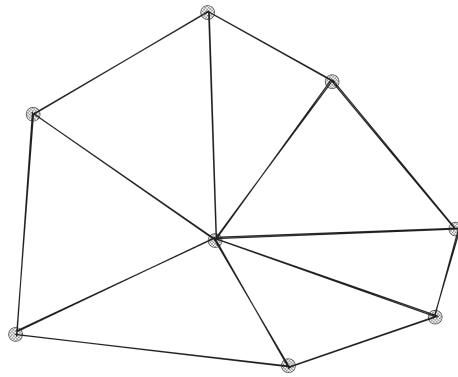
WebGL

- Огрызок OpenGL ES
- Порядком устарел

WebGL

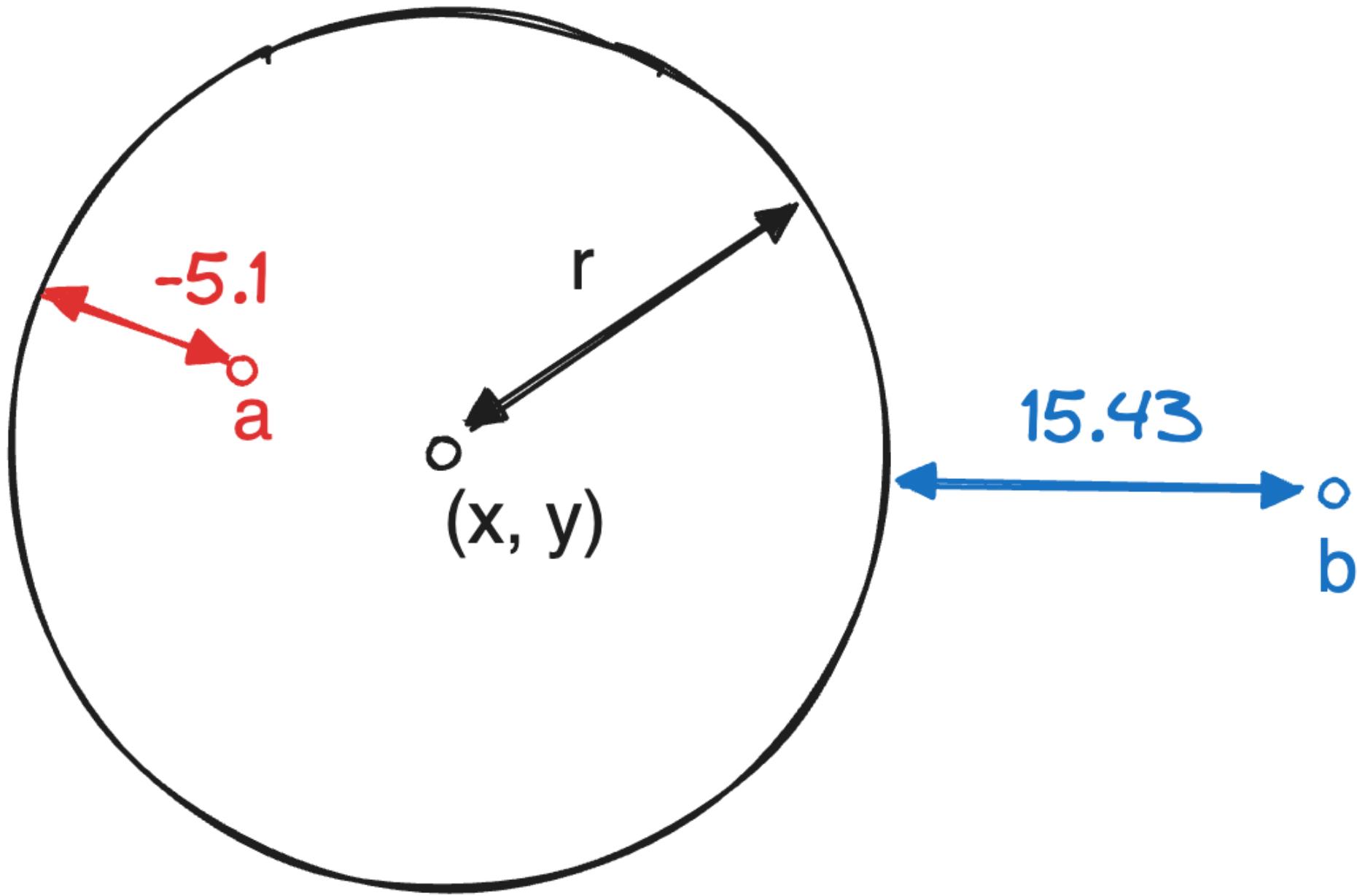
- Огрызок OpenGL ES
- Порядком устарел
- Шейдеры



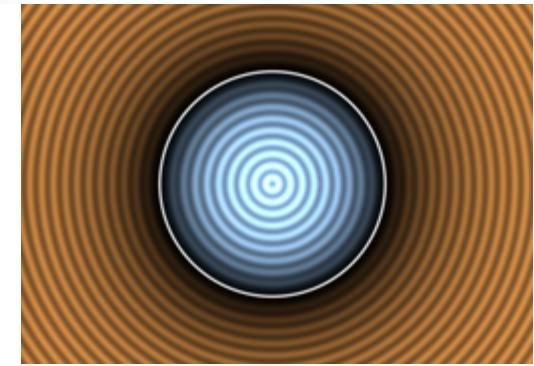


SDF (Signed Distance Function)

$$f(x) = \begin{cases} -d(x, \partial\Omega) & \text{if } x \in \Omega \\ d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases}$$

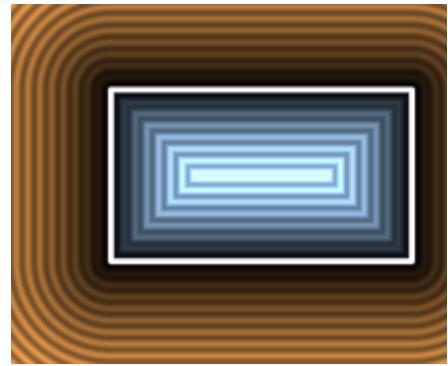


```
float sdCircle( vec2 p, float r )  
{  
    return length(p) - r;  
}
```



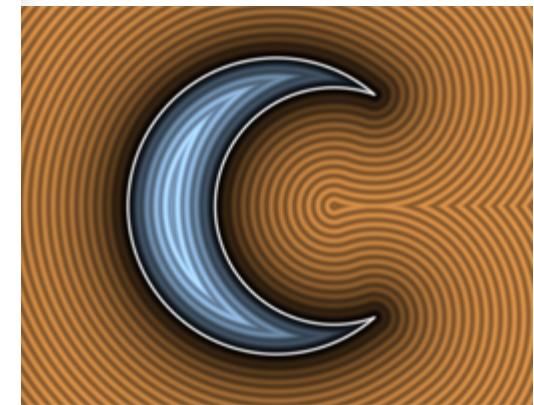
<https://iquilezles.org/articles/distfunctions2d/>

```
float sdBox( in vec2 p, in vec2 b )
{
    vec2 d = abs(p)-b;
    return length(max(d,0.0)) + min(max(d.x,d.y),0.0);
}
```

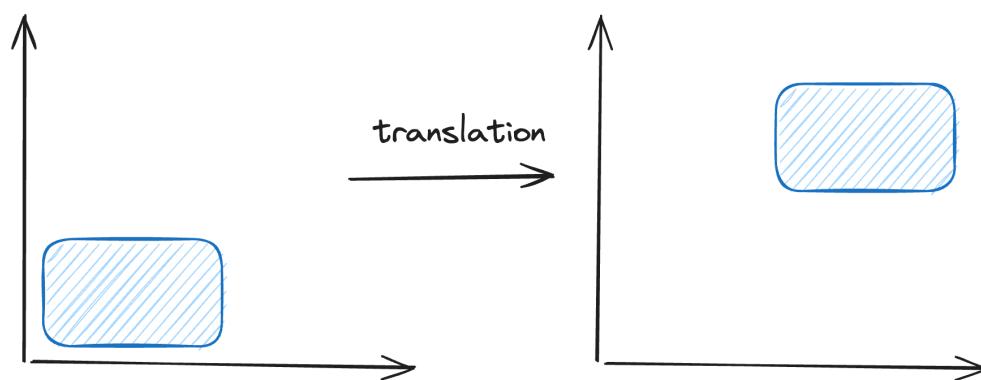
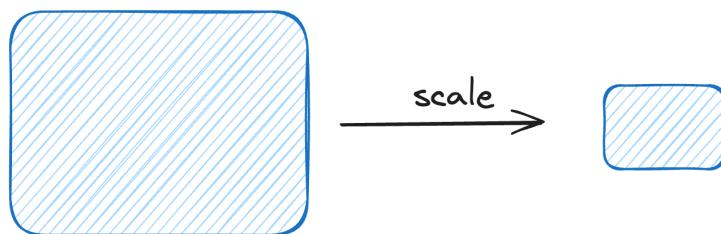
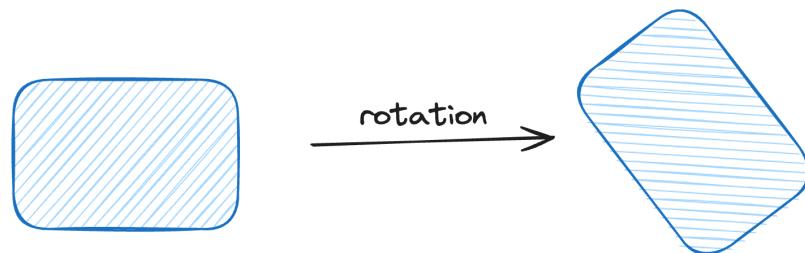


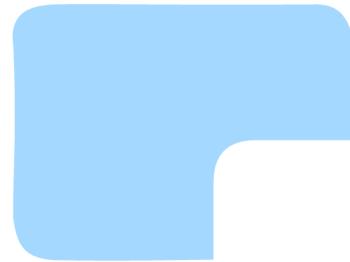
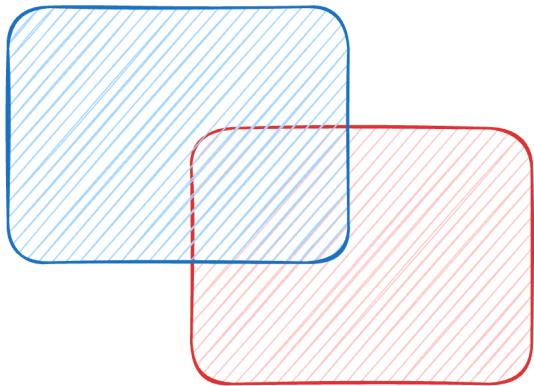
<https://iquilezles.org/articles/distfunctions2d/>

```
float sdMoon(vec2 p, float d, float ra, float rb )
{
    p.y = abs(p.y);
    float a = (ra*ra - rb*rb + d*d)/(2.0*d);
    float b = sqrt(max(ra*ra-a*a,0.0));
    if( d*(p.x*b-p.y*a) > d*d*max(b-p.y,0.0) )
        return length(p-vec2(a,b));
    return max( (length(p)-ra),
                -(length(p-vec2(d,0))-rb));
}
```



<https://iquilezles.org/articles/distfunctions2d/>

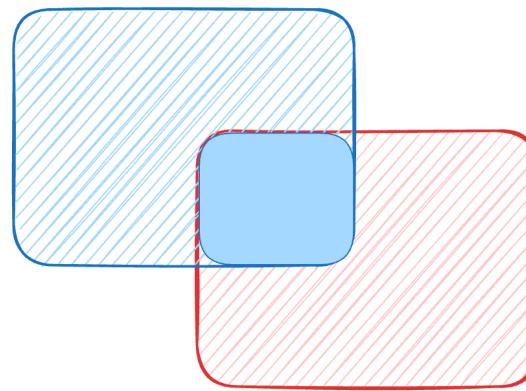




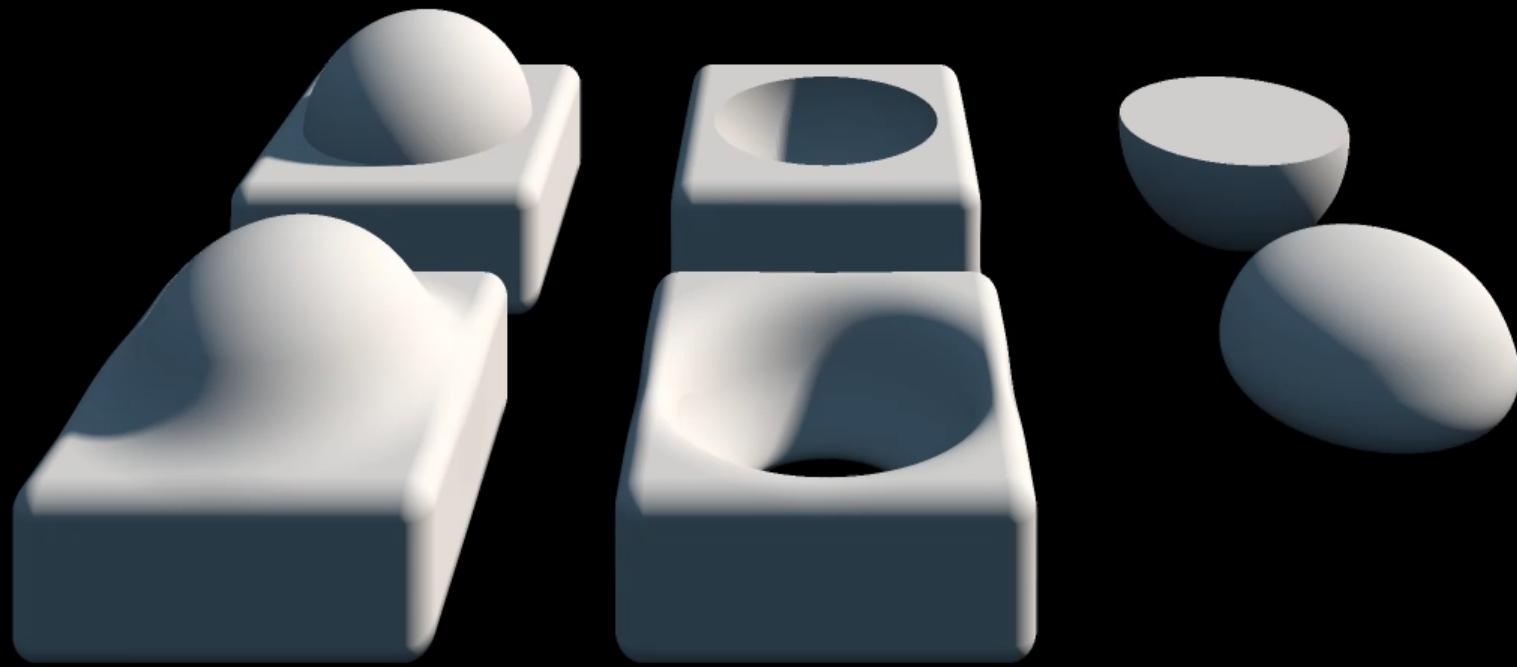
Subtraction = $\max(-d_1, d_2)$



Union = $\min(d_1, d_2)$

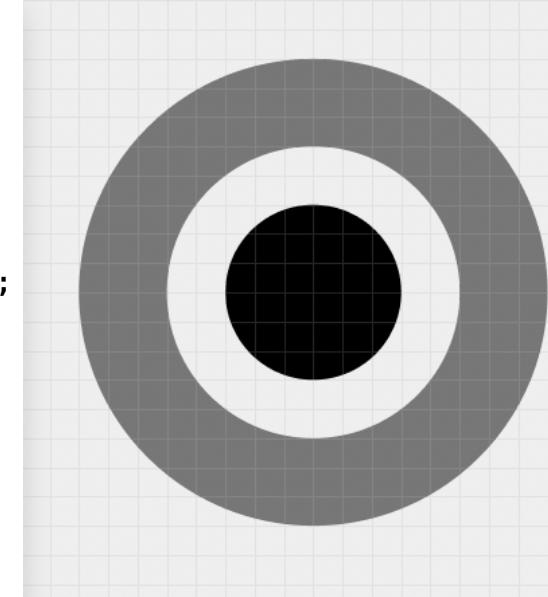


Intersection = $\max(d_1, d_2)$

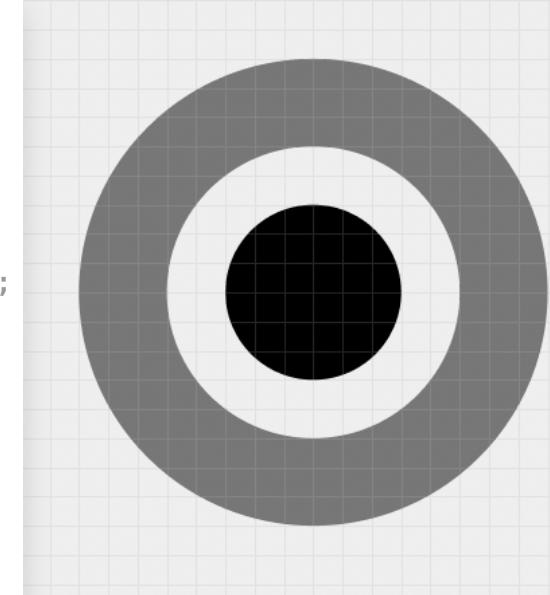


<https://www.shadertoy.com/view/lt3BW2>

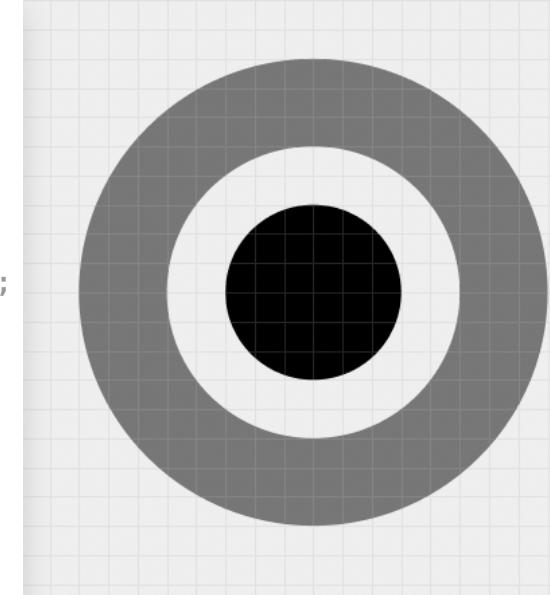
```
1 /// Local scope section button. A dot inside a circle.
2 pub mod local_scope(LocalScope) {
3     ensogl_core::define_shape! {
4         alignment = center;
5         (style: Style) {
6             let dull_alpha: Var<f32> = style.get_number(dull_color_alpha).into();
7             let dull_color = &VIVID_COLOR * &dull_alpha;
8             let dot = Circle(3.0.px());
9             let dot = dot.fill(VIVID_COLOR.gsl());
10            let outer = Circle(8.0.px()) - Circle(5.0.px());
11            let outer = outer.fill(dull_color.gsl());
12            let shape = outer + dot;
13            shape.into()
14        }
15    }
16 }
```



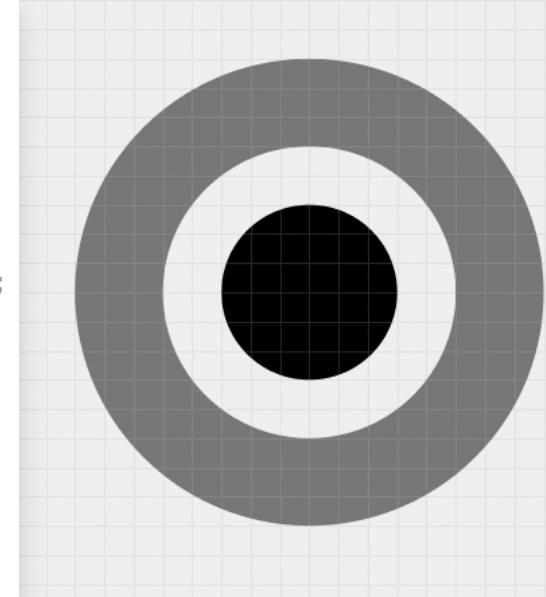
```
1 /// Local scope section button. A dot inside a circle.
2 pub mod local_scope(LocalScope) {
3     ensogl_core::define_shape! {
4         alignment = center;
5         (style: Style) {
6             let dull_alpha: Var<f32> = style.get_number(dull_color_alpha).into();
7             let dull_color = &VIVID_COLOR * &dull_alpha;
8             let dot = Circle(3.0.px());
9             let dot = dot.fill(VIVID_COLOR.gsl());
10            let outer = Circle(8.0.px()) - Circle(5.0.px());
11            let outer = outer.fill(dull_color.gsl());
12            let shape = outer + dot;
13            shape.into()
14        }
15    }
16 }
```



```
1 /// Local scope section button. A dot inside a circle.
2 pub mod local_scope(LocalScope) {
3     ensogl_core::define_shape! {
4         alignment = center;
5         (style: Style) {
6             let dull_alpha: Var<f32> = style.get_number(dull_color_alpha).into();
7             let dull_color = &VIVID_COLOR * &dull_alpha;
8             let dot = Circle(3.0.px());
9             let dot = dot.fill(VIVID_COLOR.gsl());
10            let outer = Circle(8.0.px()) - Circle(5.0.px());
11            let outer = outer.fill(dull_color.gsl());
12            let shape = outer + dot;
13            shape.into()
14        }
15    }
16 }
```



```
1 /// Local scope section button. A dot inside a circle.
2 pub mod local_scope(LocalScope) {
3     ensogl_core::define_shape! {
4         alignment = center;
5         (style: Style) {
6             let dull_alpha: Var<f32> = style.get_number(dull_color_alpha).into();
7             let dull_color = &VIVID_COLOR * &dull_alpha;
8             let dot = Circle(3.0.px());
9             let dot = dot.fill(VIVID_COLOR.gsl());
10            let outer = Circle(8.0.px()) - Circle(5.0.px());
11            let outer = outer.fill(dull_color.gsl());
12            let shape = outer + dot;
13            shape.into()
14        }
15    }
16 }
```



```
// A lot of lines of code...
bool _3286 = mod(input_local.x, input_size.x * 0.5) < (input_size.x * 0.25);
bool _3294 = mod(input_local.y, input_size.y * 0.5) < (input_size.y * 0.25);
bool _3298 = _3286 || _3294;
bool _3305;
if (_3298)
{
    _3305 = !(_3286 && _3294);
}
else
{
    _3305 = _3298;
}
output_color = mix(vec4(0.0, 0.0, 0.0, 1.0), vec4(1.0, 0.0, 0.0, 1.0), bvec4(_3305));
```

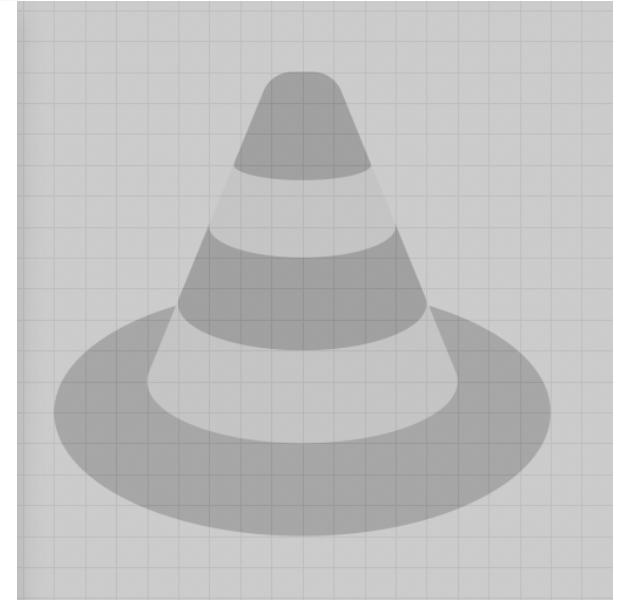
```

/// A road cone.
pub mod unstable(Unstable) {
    ensogl_core::define_shape! {
        alignment = center;
        (style: Style) {
            let x_radius = 8.0;
            let y_radius = x_radius * 0.465;
            let bottom = Ellipse(x_radius.px(), y_radius.px());
            let bottom_pos_y = -SIZE / 2.0 + y_radius;
            let bottom = bottom.translate_y(bottom_pos_y.px());
            let bottom_center = Ellipse(5.0.px(), 2.0.px()).translate_y((-3.0).px());
            let height = 12.5;
            let y_offset = height / 2.0 - 3.0;
            let bottom_triangle = Triangle(10.0.px(), height.px()).translate_y(y_offset.px());
            let bottom_mask = bottom_center + bottom_triangle;
            let bottom = bottom - bottom_mask;

            let x_radius = 4.0;
            let y_radius = 1.5;
            let middle = Ellipse(x_radius.px(), y_radius.px());
            let middle_pos_y = -0.5;
            let middle = middle.translate_y(middle_pos_y.px());
            let height = 10.0;
            let y_offset = middle_pos_y + height / 2.0;
            let middle_triangle = Triangle(8.0.px(), height.px()).translate_y(y_offset.px());
            let middle = middle + middle_triangle;
            let mask_y = middle_pos_y + 2.5;
            let mask = HalfPlane().translate_y(mask_y.px());
            let mask_ellipse = Ellipse(3.0.px(), 1.0.px()).translate_y(mask_y.px());
            let mask = mask + mask_ellipse;
            let middle = middle - mask;

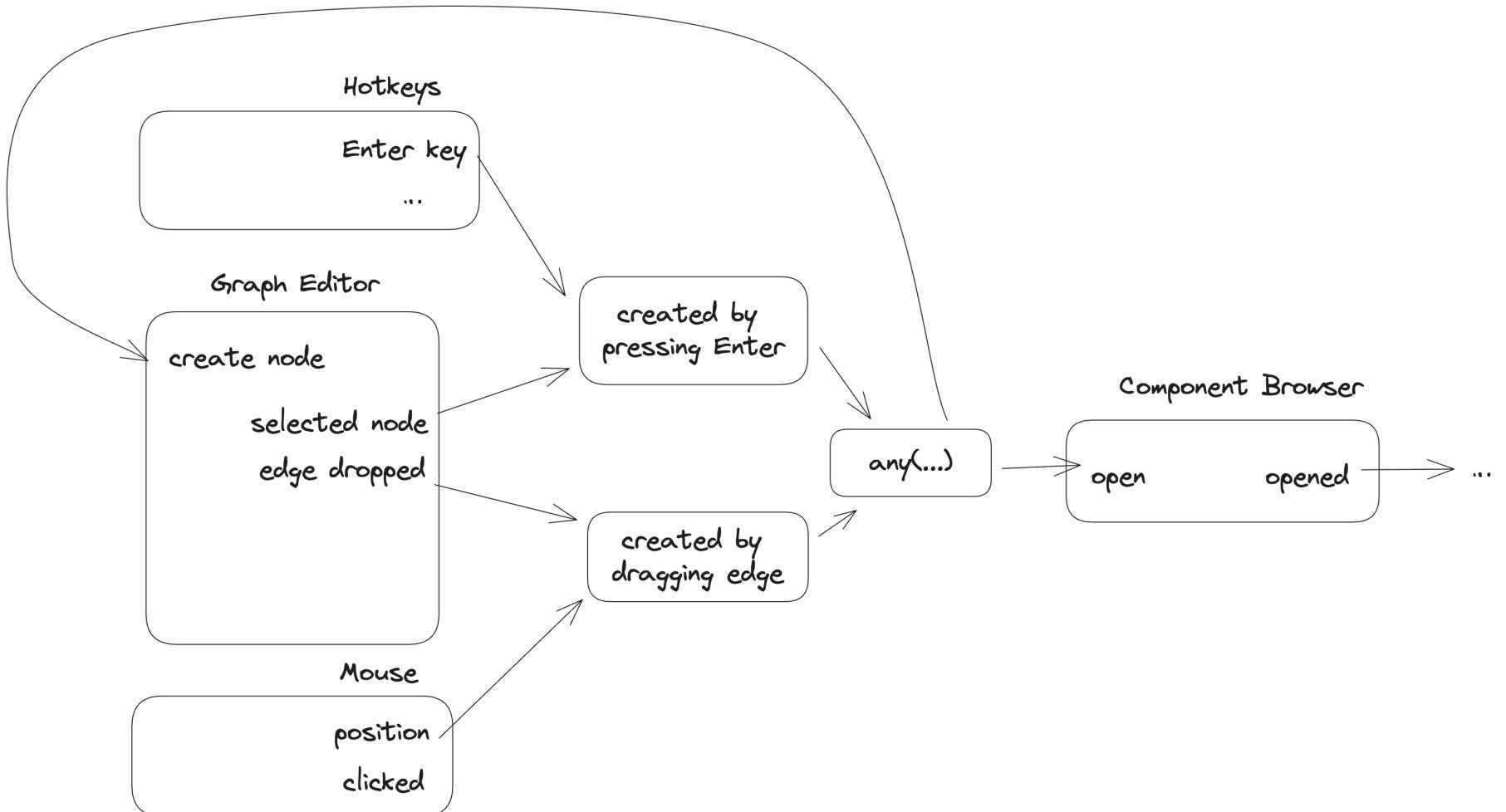
            let height = 5.5;
            let top_pos_y = middle_pos_y + 4.5;
            let y_offset = top_pos_y + height / 2.0;
            let top_triangle = Triangle(4.4.px(), height.px()).translate_y(y_offset.px());
            let triangle_mask = HalfPlane().translate_y((SIZE / 2.0 - 1.0).px());
            let top_triangle = top_triangle - triangle_mask;
        }
    }
}

```



FRP

(Functional Reactive Programming)



```
ensogl_core::define_endpoints! {
    Input {
        set_size (Vector2),
        mouse_nearby (bool),
        click (),
    }
    Output {
        clicked (),
        is_hovered (bool),
        state (State),
        size (Vector2),
    }
}
```

```
1 // === Remove node ===
2
3 frp::extend! { network
4     all_nodes <= inputs.remove_all_nodes.map(f_!(model.nodes.keys()));
5     selected_nodes <= inputs.remove_selected_nodes.map(f_!(model.nodes.all_selected()))
6     nodes_to_remove <- any (all_nodes, selected_nodes);
7     out.node_removed <+ nodes_to_remove;
8 }
```

```
1 // === Remove node ===
2
3 frp::extend! { network
4     all_nodes <= inputs.remove_all_nodes.map(f_!(model.nodes.keys()));
5     selected_nodes <= inputs.remove_selected_nodes.map(f_!(model.nodes.all_selected()))
6     nodes_to_remove <- any (all_nodes, selected_nodes);
7     out.node_removed <+ nodes_to_remove;
8 }
```

```
1 // === Remove node ===
2
3 frp::extend! { network
4     all_nodes <= inputs.remove_all_nodes.map(f_!(model.nodes.keys()));
5     selected_nodes <= inputs.remove_selected_nodes.map(f_!(model.nodes.all_selected()))
6     nodes_to_remove <- any (all_nodes, selected_nodes);
7     out.node_removed <+ nodes_to_remove;
8 }
```

```
1 // === Remove node ===
2
3 frp::extend! { network
4     all_nodes <= inputs.remove_all_nodes.map(f_!(model.nodes.keys()));
5     selected_nodes <= inputs.remove_selected_nodes.map(f_!(model.nodes.all_selected()))
6     nodes_to_remove <- any (all_nodes, selected_nodes);
7     out.node_removed <+ nodes_to_remove;
8 }
```

Кто мы и что делаем

Как можно написать фронтенд на Rust?

Как можно не написать

фронтенд на Rust?

Конец?

Многословность

```
document.getElementById("mousemove").onmousemove = (e) => {
  var rect = e.target.getBoundingClientRect();
  var x = e.clientX - rect.left;
  var y = e.clientY - rect.top;
  console.log(`Mouse position: (${x}, ${y})`);
}
```

```
use wasm_bindgen::{prelude::*, JsCast};
use web_sys::{console, Document, HTMLElement, MouseEvent};

let mousemove = Closure::<dyn Fn(MouseEvent)>::wrap(Box::new(|e| {
    let rect = e
        .target()
        .expect("mouse event doesn't have a target")
        .dyn_into::<HTMLElement>()
        .expect("event target should be of type HTMLElement")
        .getBoundingClientRect();
    let x = (e.clientX() as f64) - rect.left();
    let y = (e.clientY() as f64) - rect.top();
    console::log_1(&format!("Left: {}, Top: {}", x, y));
}));
```

Пересборка

Пересборка

- 20–40 секунд на каждую пересборку

Пересборка

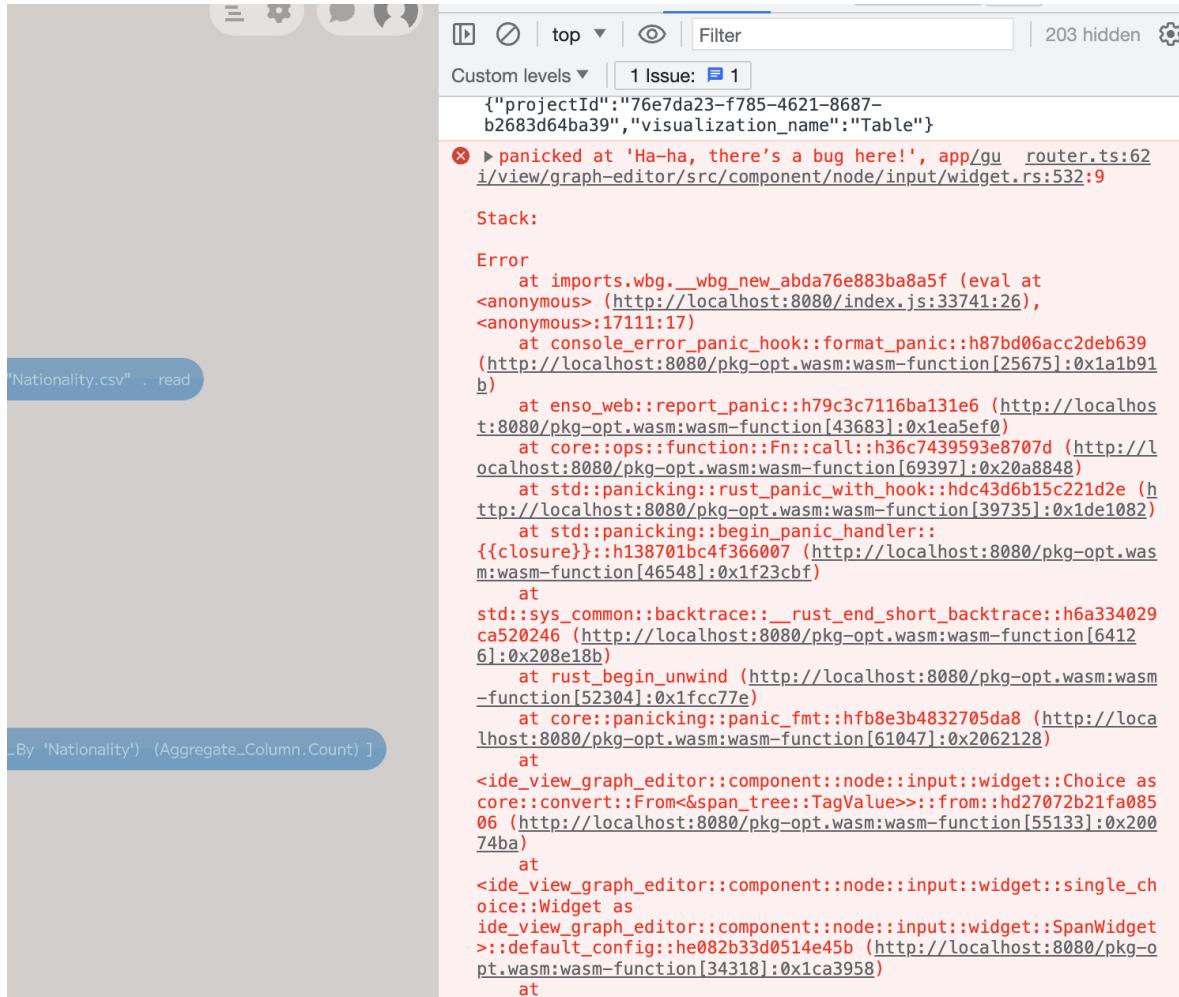
- 20–40 секунд на каждую пересборку
- Демо-сцены

Пересборка

- 20–40 секунд на каждую пересборку
- Демо-сцены

```
./run gui watch --crate-path=app/gui/demo/icons
```

Отладка



Процесс починки бага

Процесс починки бага

1. Учишься воспроизводить

Процесс починки бага

1. Учишься воспроизводить
2. Изучаешь исходники и думаешь

Процесс починки бага

1. Учишься воспроизводить
2. Изучаешь исходники и думаешь
3. Если не придумал, добавляешь везде debug ! ()

Процесс починки бага

1. Учишься воспроизводить
2. Изучаешь исходники и думаешь
3. Если не придумал, добавляешь везде debug ! ()
4. ...

Процесс починки бага

1. Учишься воспроизводить
2. Изучаешь исходники и думаешь
3. Если не придумал, добавляешь везде debug ! ()
4. ...
5. PROFIT

Тестирование

```
use wasm_bindgen_test::*;

wasm_bindgen_test_configure!(run_in_browser);

#[wasm_bindgen_test]
fn test_whole_app() {
    App::new().run();
}
```

```
fn test_adding_node(add_node: impl Fn(&GraphEditor)) {
    let (app, graph_editor) = init();
    assert_eq!(graph_editor.num_nodes(), 0);

    // Adding first node.
    let (node_1_id, node_1) = graph_editor.add_node_by(&add_node);
    graph_editor.assert(Case { node_source: None, should_edit: true });
    graph_editor.stop_editing();
    assert_eq!(graph_editor.num_nodes(), 1);
    next_frame();

    // First node is created in the center of the screen.
    let node_1_pos = node_1.position();
    let screen_center = app.display.default_scene.screen_to_scene_coordinates(Vector3::zeros());
    assert_eq!(node_1_pos.xy(), screen_center.xy());

    // Adding second node with the first node selected.
    graph_editor.model.nodes.select(node_1_id);
    let (_, node_2) = graph_editor.add_node_by(&add_node);
    graph_editor.assert(Case { node_source: Some(node_1_id), should_edit: true });
    assert_eq!(graph_editor.num_nodes(), 2);

    // Second node is below the first and left-aligned to it.
    let node_2_pos = node_2.position();
    assert!(node_2_pos.y < node_1_pos.y);
    assert_eq!(node_2_pos.x, node_1_pos.x);
}
```

«Война за производительность»

«Война за производительность»

- Прекомпиляция шейдеров

«Война за производительность»

- Прекомпиляция шейдеров
- Инструментарий для профилирования

«Война за производительность»

- Прекомпиляция шейдеров
- Инструментарий для профилирования
- Меньше слоёв → меньше draw calls

«Война за производительность»

- Прекомпиляция шейдеров
- Инструментарий для профилирования
- Меньше слоёв → меньше draw calls
- Кэширование SDF-фигур в текстуру

«Война за производительность»

- Прекомпиляция шейдеров
- Инструментарий для профилирования
- Меньше слоёв → меньше draw calls
- Кэширование SDF-фигур в текстуру
- Magic Rectangle

«Война за производительность»

- Прекомпиляция шейдеров
- Инструментарий для профилирования
- Меньше слоёв → меньше draw calls
- Кэширование SDF-фигур в текстуру
- Magic Rectangle
- Оптимизация HTML

Слишком долго

Слишком долго

- Разработка новых фичей и исправление багов занимает слишком много времени

Слишком долго

- Разработка новых фичей и исправление багов занимает слишком много времени
- Мы утопаем в низкоуровневых багах

Слишком долго

- Разработка новых фичей и исправление багов занимает слишком много времени
- Мы утопаем в низкоуровневых багах
- Мы вынуждены реализовывать многие вещи с нуля

Кто мы и что делаем

Как можно написать фронтенд на Rust?

Как можно не написать фронтенд на Rust?

Конец?

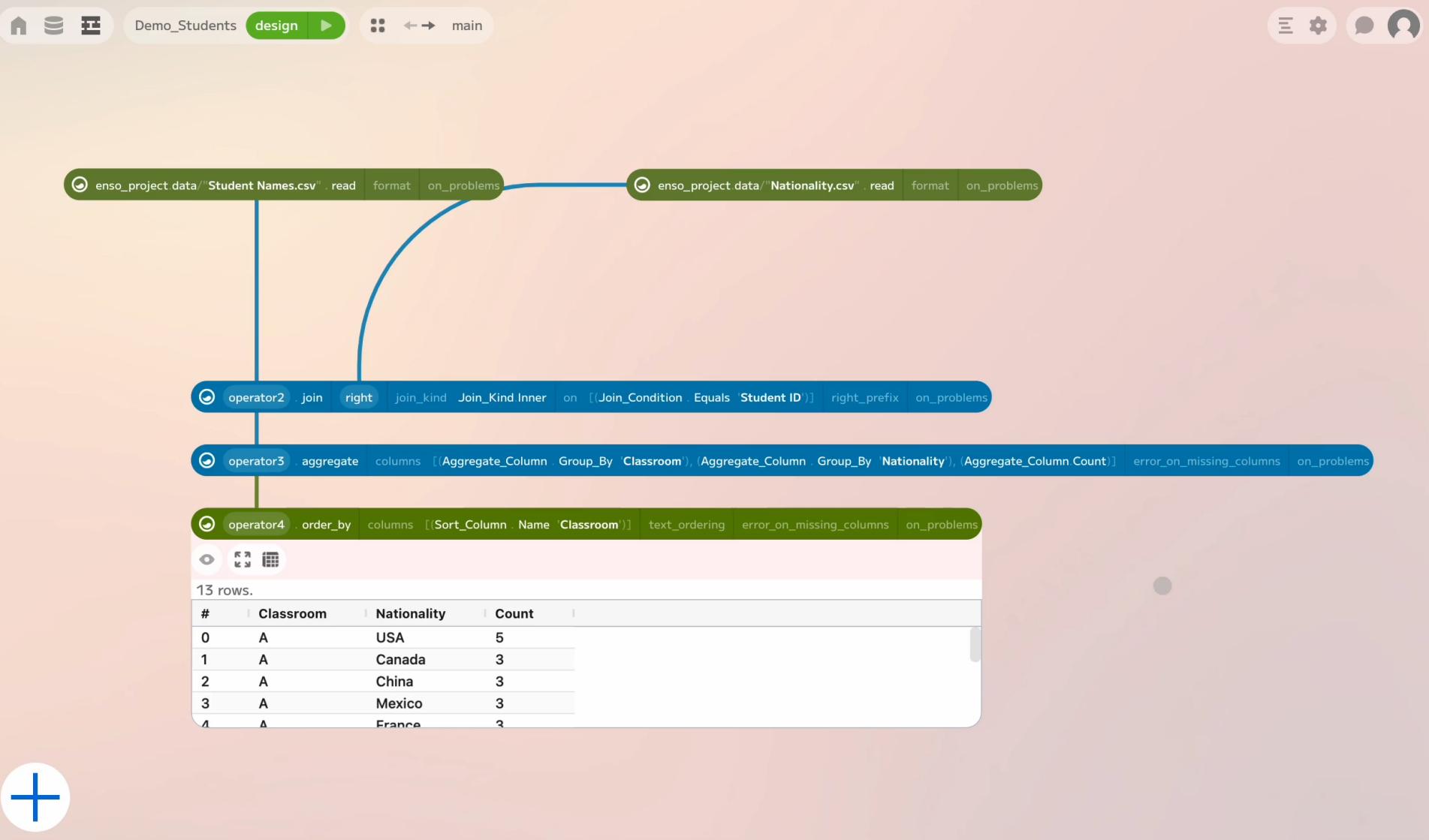
Текущий статус

IDE in Vue #7685

 Open

 51 of 65 tasks

farmaazon opened this issue on Aug 29 · 0 comments



Выводы

Выводы

- У каждого инструмента есть сильные и слабые стороны

Выводы

- У каждого инструмента есть сильные и слабые стороны
- Трезво оценивайте свои возможности

Спасибо за внимание

Слайды: vitvakatu/rustcon2023

Мой Telegram: [@vitvakatu](https://t.me/vitvakatu)

Наш сайт: [Enso.org](https://enso.org)

Наш GitHub: [enso-org/enso](https://github.com/enso-org/enso)