Rust

Group 7 - Jaeseok Choi

Overview

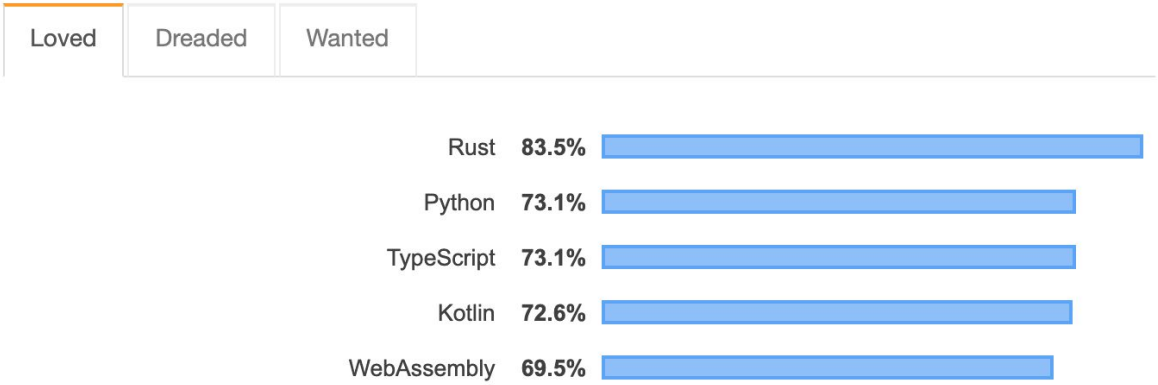
- Introduction
- Evaluation Criteria
- Project Overview

What is Rust?

- System-level programming language
- High level of performance & safety
- Syntactically similar to C++, but provides memory safety
- “A language empowering everyone to build reliable and efficient software” - Official Rust Website

*Sources: https://en.wikipedia.org/wiki/Rust_(programming_language) & https://www.rust-lang.org/

Most Loved Language



* Source: https://insights.stackoverflow.com/survey/2019#community

Why use Rust?

- Memory Management
- Concurrency
- Functionality
- Performance
- Community
- Available Libraries
- Developer Cost

1. Memory Management

- Ownership
  - A discipline for access control that programmers try to follow, but
  - Rust's compiler checks statically for programmers
- Ownership Rules
  - Each value in Rust has a variable called *"owner"*
  - There can only be one owner at a time
  - When the owner goes out of scope, the value will be dropped

*Source: https://doc.rust-lang.org/book/ch04-01-what-is-ownership.html

Memory Management Example 1

```
fn main() {
    let a = 4;
}
```

Memory Management Example 2

```
let s1 = String::from("hello");
let s2 = s1;

println!("{}, world!", s1);
```

```
----------------------------Compile--------------------------------

2 |     let s1 = String::from("hello");
  |     -- move occurs because `s1` has type
`std::string::String`, which does not implement the `Copy` trait
3 |     let s2 = s1;
  |              -- value moved here
4 |
5 |     println!("{}, world!", s1);
  |                            ^^ value borrowed here after move
```

*Source: https://doc.rust-lang.org/book/ch04-01-what-is-ownership.html

2. Concurrency

- Use **threads** to run code simultaneously
- Use **message passing** to transfer data between threads
  - Channel
- **Shared-state** concurrency
- **Extensible** concurrency with *sync* and *send* traits

*Source: https://doc.rust-lang.org/book/ch16-00-concurrency.html

3. Functionality

- Operating Systems
- Browsers
- Game Engine
- File Systems
- Simulation Engines for Virtual Reality
- Any software applications that require high performance and safety

*Source:https://research.mozilla.org/rust/#:~:text=What%20is%20Rust%3F,simulation%20engines%20for%20virtual%20reality.

4. Performance

	Binary-trees (secs / mem)	N-body (secs / mem)	Reverse-complement (secs / mem)
Rust	3.48 / 199,856	5.98 / 688	1.69 / 994,998
C++	3.96 / 113,236	7.70 / 1,612	1.89 / 499,672
Go	25.19 / 627,604	21.26 / 1,888	3.73 / 826,488

*Source: https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/rust.html

5. Community

- This-Week-In-Rust
  - Weekly updates from Rust community including news and blog posts
- 93,682 Github repositories
- 500 Stack Overflow responses
- 426 jobs found on indeed.com

*Sources: https://this-week-in-rust.org/, https://www.indeed.com/q-Rust-Developer-jobs.html, https://stackoverflow.com/search?q=rust, & https://github.com/search?q=rust

6. Available Libraries

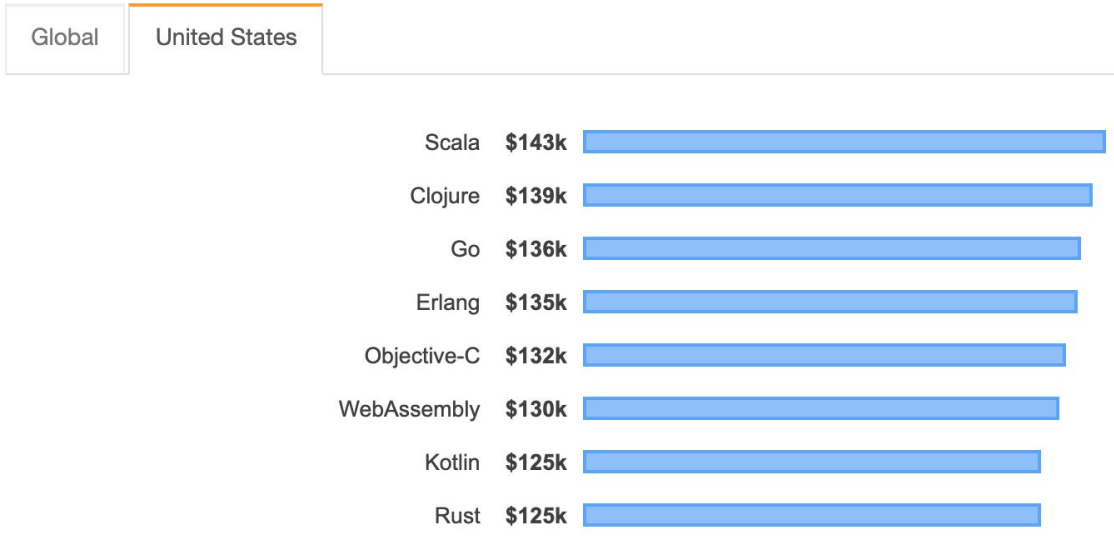
Crates.io - Rust's open source libraries website

42,134 crates* available to download

*crate: a special word for *library* in Rust

*Source: https://crates.io/

7. Developer Cost



*Source: https://insights.stackoverflow.com/survey/2019#top-paying-technologies

Project Overview - Game *21*

- Objective
- Constraints
- Features
- Technology used
- Application

1. Objective

- Similar to ‘Snake Game’, but more complexity added
  - Starts with a small triangle (head) at a random point
  - Needs to catch a moving rectangle (body)
  - Circles (enemies) are also moving around in random directions
  - If contact with circles, game ends
  - Must reach either a body length of 21 or 21 seconds to win the game

2. Constraints

- Minimal design of entities
  - Rectangles, circles, triangles etc.
- Desktop game only

3. Features

- Fun and simple
- Fast execution
- Can play with both a keyboard or a mouse

4. Technology Used

- Piston - a modular open source game engine
- Graphics, Glutin_Window, and many other crates

Piston

5. Application

- Basic game implemented on OSX, Linux, etc.

Questions?