Language Evaluation

Austin Derbique A01967241 CS4700 Fall 2017

Intro: This evaluation will be on the programming language Rust. Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety. https://www.rust-lang.org/en-US/.

Rust Informaiton:

Install Here: https://www.rust-lang.org/en-US/install.html

To Compile: rustc <program.rs>

To Run: ./<program> *You do not need rust installed to be able to run the compiled versions

Program Implementation: I decided to use an old homework assignment that I had written in C++ for my CS1400 class. This assignment was to write a cigarette vending machine. A user could deposit quarters, purchase cigarettes, have someone refill the machine, check the balance, show the menu, or quit. The assignment description is called cigarette_machine_description.pdf located in the same directory as this file. My goal was to recreate this game in Rust. I was successful in recreating an identical version of the game in this new language (same behavior).

Files:

Rust Source Code: rust_version.rs

Rust Compiled Program: rust_version.run

C++ Source Code: c_plus_plus_version.cpp

C++ Compiled Program: c_plus_plus.run

Compiling & Running in Rust:

```
[austin@a01967241-7 finalproject]$ clear
[austin@a01967241-7 finalproject]$ rustc rust_version.rs
[austin@a01967241-7 finalproject]$ ./rust_version
```

Compiling & Running in C++:

```
[austin@a01967241-7 finalproject]$ g++ c_plus_plus_version.cpp -o c_plus_plus.run
[austin@a01967241-7 finalproject]$ ./c_plus_plus.run
```

GamePlay from Rust: Check 3rd page of evaluation

Strengths of Rust: Rust prides itself on being fast, preventing segmentation faults, and guaranteeing thread safety. This means that it is a language that does well when scaled. It features zero-cost abstraction, pattern matching, type inference, move semantics,threads without data races, guaranteed memory safety, minimal run time, trait-based generics, and efficient C bindings. Rust also provides a great package manager called Cargo that can be used for installing useful libraries easily.

Weaknesses of Rust: Currently, there are not many developers that know the Rust language. This makes it difficult for it to become popular. There are less blogs, tutorials, and other places to obtain information on the language. Although Cargo provides a nice set of libraries, there are still fewer than in other languages. There is a known problem with memory leakage with Rust's scope based recursive functions.

Situation: Rust seems to be similar to C++. It would be great in a scaled environment where you have a multithreaded program that relies heavily on memory management. An example of this could be computing information or protein folding applications which are very resource intensive.

Conclusion: From what I've learned about rust over the last few weeks, it's a very useful language that is easy to pick up and learn. Already knowing C++, the transition to Rust was simple. There are some useful quirks to the language that make some tasks rather easy, like pattern matching. It definitely seems to be an up and coming language and it is certain I will be programming in it again.

Example game play footage is below.

Rust GamePlay of Cigarette Machine

```
Welcome to the electronic cigarette machine.

m - display menu options
s - report the machine status
d - drop in quarter
b - check balance
1 - pull the 1st knob
2 - pull the 2nd knob
3 - pull the 3rd knob
4 - pull the 4th knob
r - restock the machine
q - quit the machine
b
You have $0.00 in the machine.

d
You insert a quarter.
d
You insert a quarter.
d
You insert a quarter.
d
You insert a quarter.
d
You insert a quarter.
b
You have $1.25 in the machine.

3
You pull the 3rd knob.
One pack of Gauloises drops from the bottom of the machine

b
You have $0.00 in the machine.

2
You don't have enough money. Thank you for your donation ;)

q







Thanks for playing!!
```