Learning a new language with Advent of Code

Gordon Lawrenz

Motivation

- Why Learning a new language?
  - Typescript is great, but not the best fit for all problems
  - Learning different language enables you to think about problems differently
    - new concepts
    - high vs low level
    - allocations
    - performance

Motivation

- Why Rust?
  - I know and used C/C++, but I don't really like them
  - Memory safety
  - Pretty good support for many different use cases (https://wiki.mozilla.org/Areweyet)
  - A lot of functionality built in and extendable
  - Big community
  - WebAssembly/WASM looks promising
  - Fits my use cases: CLI Tools, Servers, Game Programming

  - BLAZINGLY FAST

Motivation

- Why Advent of Code?
  - You actually need to program to learn a programming language
  - Provides a clear path
  - Challenging problems
  - Great community

Advent of Code

- What actually is Advent of Code
  - Series of 25 puzzles over dezember each year
  - Created by Eric Wastl
  - A puzzle consists of two parts
    - You only get access to part 2 if you have solved part 1
    - Part 1 – a base case
    - Part 2 – an extended case with some caveats (e.g. computationally expensive)
    - For each part you get a test input
- https://adventofcode.com/

How to start

- There are good tutorial for most languages
- Many people use either so called...
  - "starter templates"
  - "scaffolds"
- But you can also just
  - create a local project for you language
  - download the inputs & tests

Example

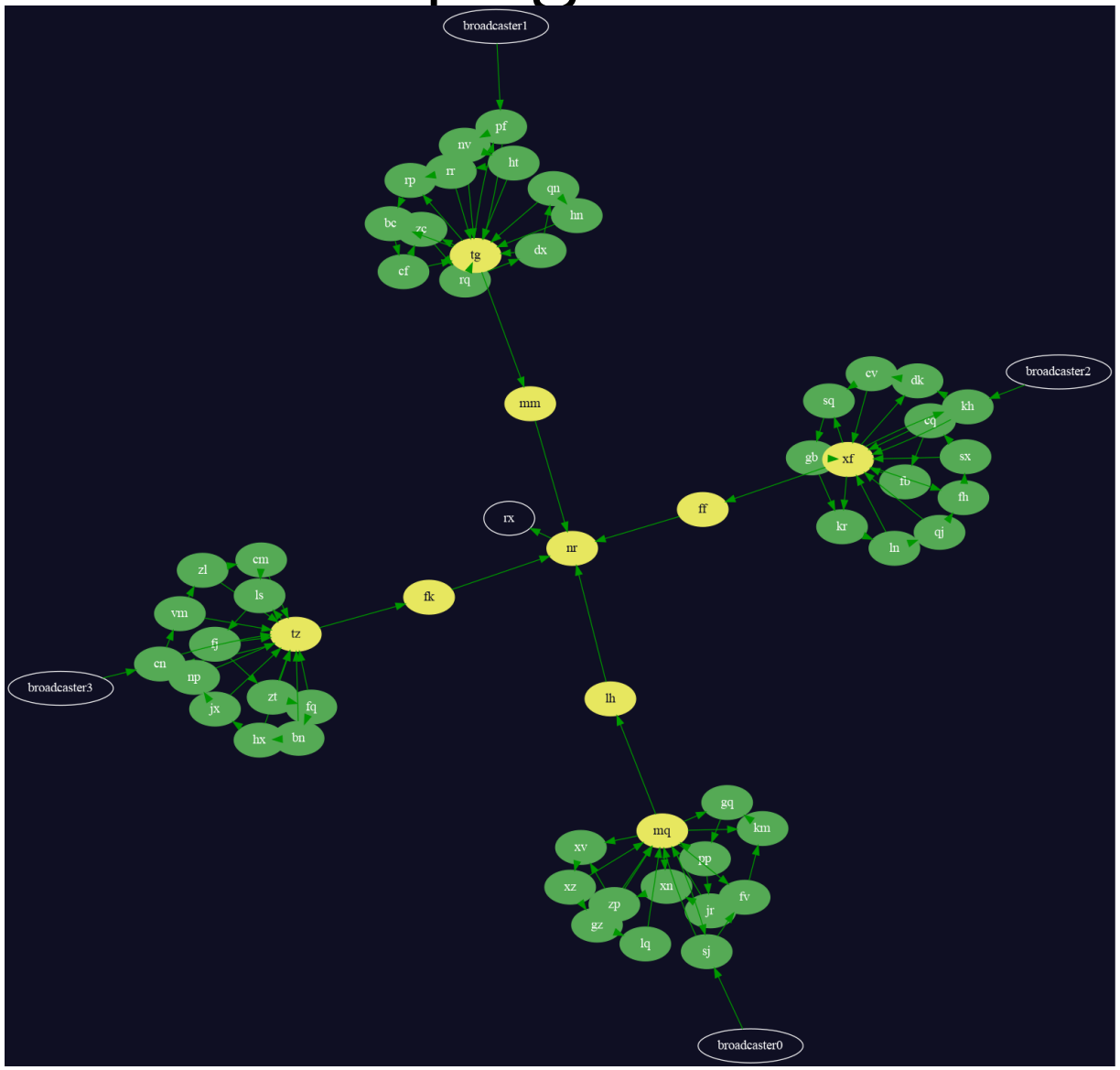
Day 2: Cube Conundrum

- Input:
  - Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue; 2 green
  - Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 red; 1 green, 1 blue
- Game:
  - In each game three sets of cubes are revealed from the bag (and then put back again).
- Part 1:
  - Determine which games would have been possible if the bag had been loaded with only 12 red cubes, 13 green cubes, and 14 blue cubes. What is the sum of the IDs of those games?
- Part 2:
  - For each game, find the minimum set of cubes that must have been present. What is the sum of the power of these sets?

Day 20: Pulse Propagation

- Input:
  broadcaster -> a, b, c
- %a -> b
- %b -> c
- %c -> inv
- &inv -> a
- Puzzle represents an electrical circuit:
  - Graph with different type of modules sending high/low signals
  - Broadcaster broadcasts
  - % => Flip Flop
    - on or off state
    - high pulse is ignored; low pulse flips on/off state
    - if on/of state is switched sends signal: on => sends high; off => sends low
  - & => Conjunction
    - remembers inputs
    - if all inputs are high sends low pulse; otherwise sends high pulse

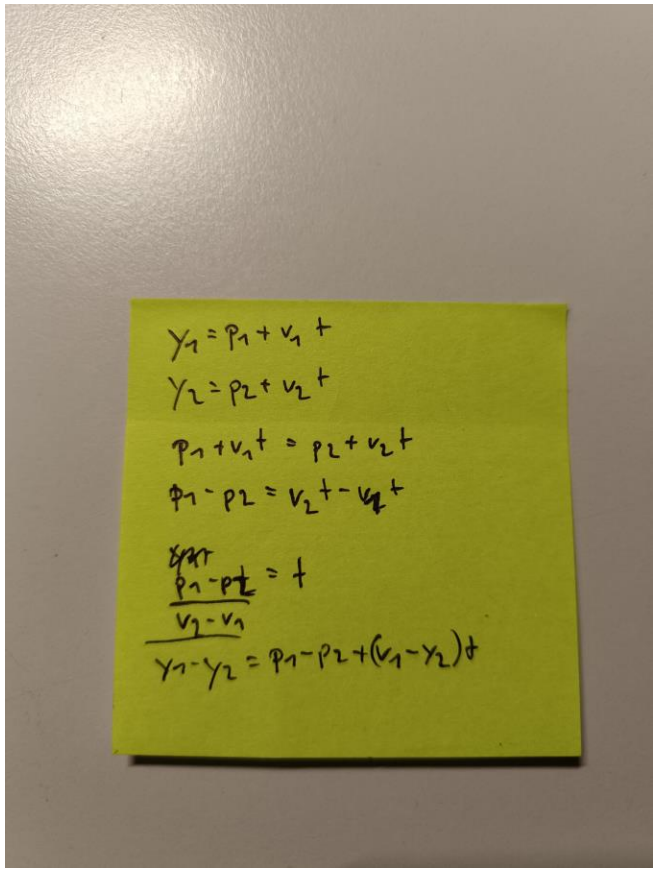
Day 20: Pulse Propagation



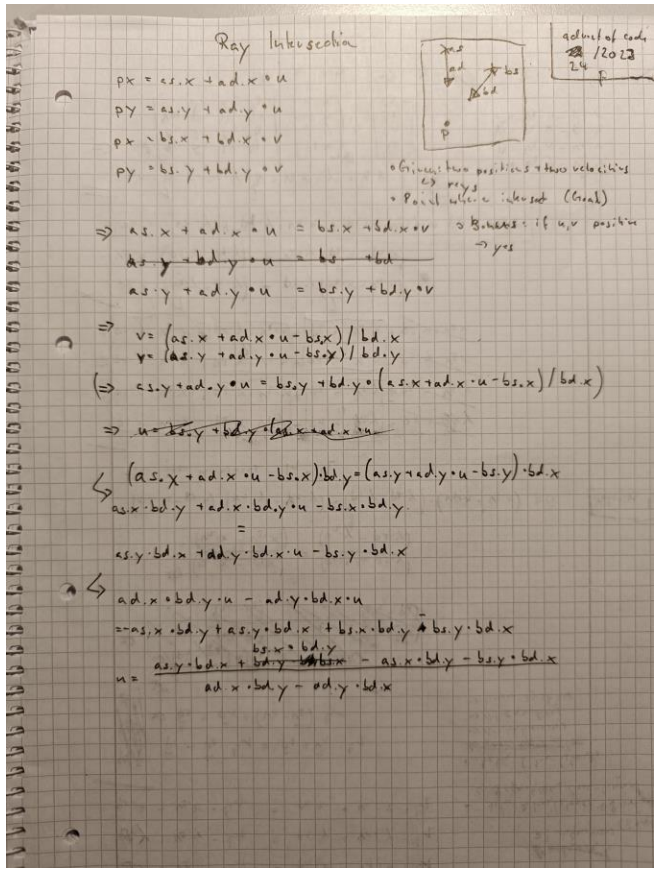
Day 24 – Never Tell Me The Odds

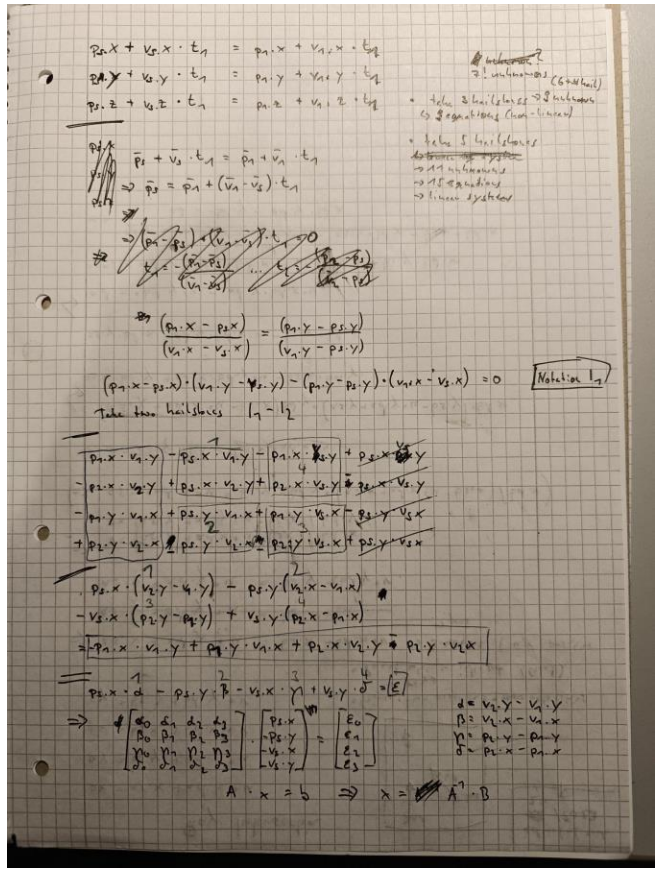
- Input:
  19, 13, 30 @ -2, 1, -2
- 18, 19, 22 @ -1, -1, -2
- 20, 25, 34 @ -2, -2, -4
- 12, 31, 28 @ -1, -2, -1
- 20, 19, 15 @ 1, -5, -3
- Explanation: Hailstones (Hagelkörner) with their position/velocity
- Part 1: Calculate if paths of hailstones paths collidate in a certain area; ignore z
- Part 2: Calculate a hailstone which actually collides with all others

Day 24 – Never Tell Me The Odds



Day 24 – Never Tell Me The Odds





Day 25: Snowverland

- Input
  jqt: rhn xhk nvd
- rsh: frs pzl lsr
- xhk: hfx
- Puzzle represents a bi-directional graph
- Puzzle: find the 3 edges which need to be cut to separate the graph into two

Learnings - Algorithms

- Depth-/Breadth-First-Search (DFS/BFS)
- Dijkstra/A*
- Min-Flow-Cut algorithms like Karger-Stein
- Manhatten-Distance
- Dynamic Programming (a.k.a. divide & conquer with memoization)
- Linear Equation Systems

Learnings - Rust

- Basics
  - Creating & running a project
  - Cargo
  - Parsing input
- Basic data structure
  - Options
  - Vecs
  - HashSet/HashMap
- Creating own data structures
  - Structs
  - Enums
  - Impl
  - match

What I haven't learned yet

- Writing good rust code
- Proper debugging
- Performant Code
- Advanced Data Structures
  - Box – Smart Pointer
  - RefCell – Interior mutability with several owners
  - Rc – Reference Counter
  - Arc – Atomic Reference Counter (Thread-Safe Rc)

Resources

- https://adventofcode.com/
- https://www.reddit.com/r/adventofcode/
- https://www.youtube.com/watch?v=gibVyxpi-qA
  (Eric Wastl - Advent of Code: Behind The Scenes - Leetspeak 2019)

- https://doc.rust-lang.org/book/
- https://github.com/sger/RustBooks