

# Python vs Rust...

(for simulation)

Alisa Dammer

ML Engineer at

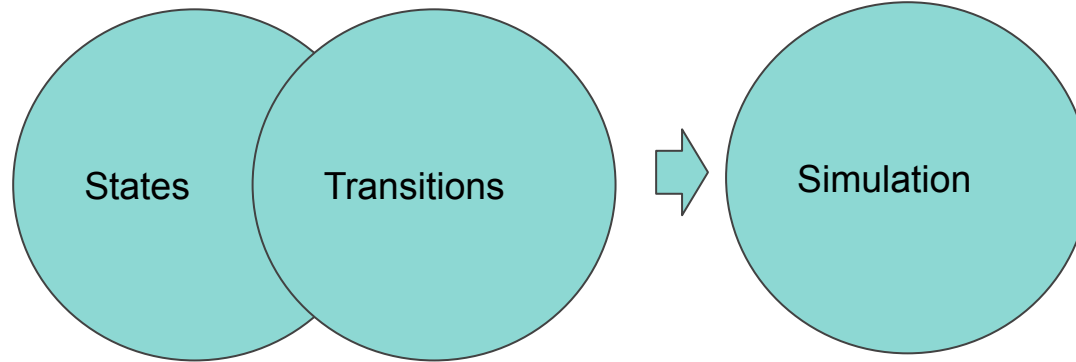


Repo: [https://github.com/Alisa-lisa/conferences/tree/master/EP\\_2019](https://github.com/Alisa-lisa/conferences/tree/master/EP_2019)

# What is Simulation?



Approximate imitation



# Types



## Continuous

- Physics
- Biology
- Chemistry
- Advanced engineering systems

## Discrete-event

- Taxi
- Post Office
- Manufacturing pipeline
- Network protocols

## Mixed

- Forestry
- Health care
- Higher complexity systems

# Tools

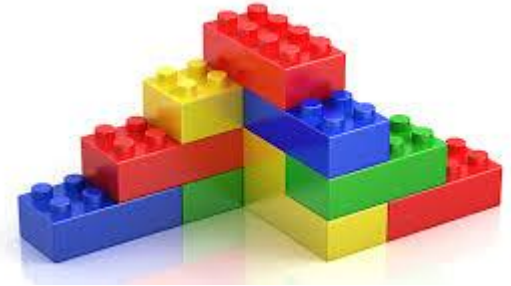


- Frameworks (GUST)
- Libraries (SimPy)
- Game Engines (Unity)
- Programming languages

# Important points



© Enache Dumitru Bogdan • www.free-cartoon-clipart.blogspot.com



# Scenario



- Spawn  $0..N$  taxi requests with  $P$  chance
- Request can be assigned to a FREE car only
- Request gets cancelled after  $X$  seconds, if not assigned
- Cars are either FREE or OCCUPIED
- 1 day of simulation

# Criteria



## Objective:

- Amount of code
- Testing simplicity
- Documentation generation (API or usage manual)
- Performance
- Memory usage
- Ecosystem
- Language versions (Major updates, breaking changes, etc.)

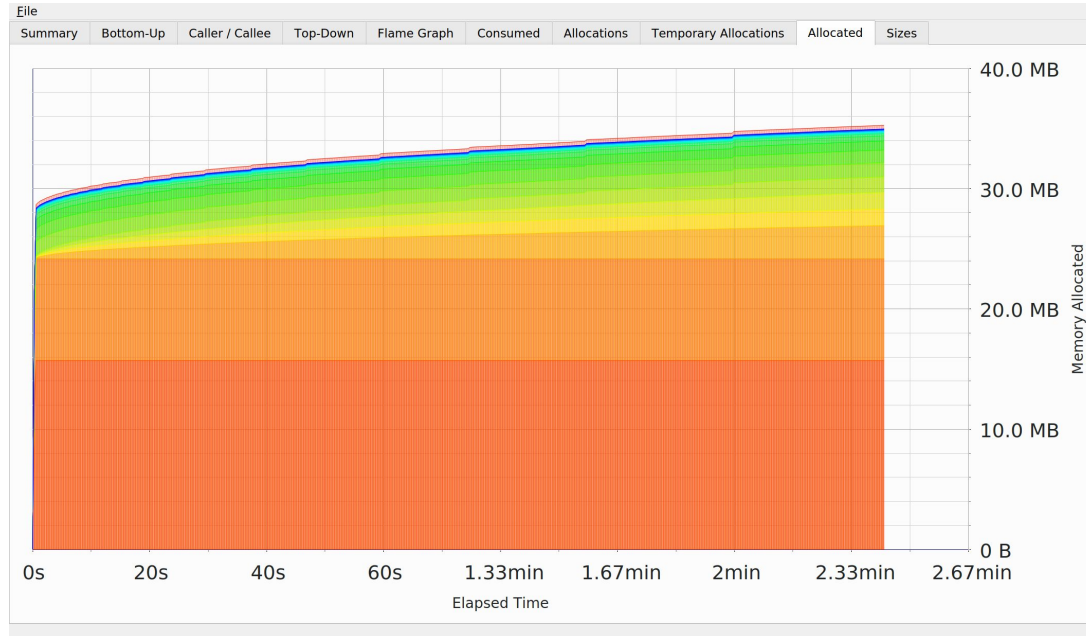
## Subjective:

- Code simplicity
- Development speed

# Python



- Lines: 94,
- Performance: 209.036s+-16.96s
- Memory usage:

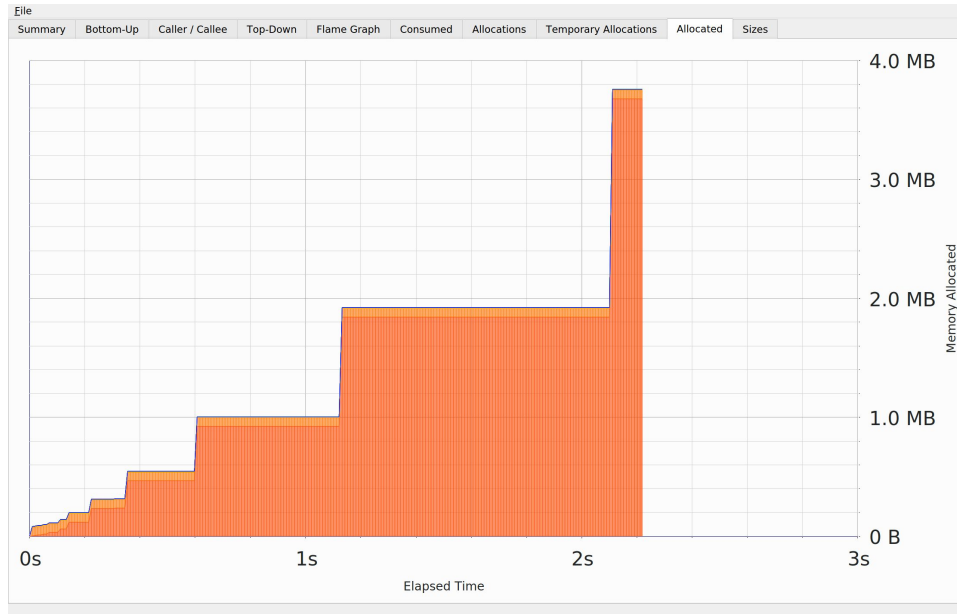




# Rust



- Lines: 160
- Performance: 154.5ms +- 4.4ms
- Memory usage:



# Comparison



criteria	Python	Rust
Amount of code	Green	Red
Test Simplicity	Green	Green
Documentation	White	Green
Memory efficiency	Red	Green
Performance	Red	Green
Ecosystem	Green	Red
Versions	Green	White
Simplicity	Green	Red
Development speed	Green	Red

# Conclusion



What do you want to reach and what are your pain points?