

Relativitization: an interstellar social simulation framework and a turn-based strategy game

(Adrian) Lai Kwun Hang
Leiden University



Motivation

A bit of self intro

- ▼ I was educated in physics
- ▼ I did gravitational waves research under the LIGO scientific collaboration during my master
- ▼ And I started to learn about open source software when I messed around with supercomputers



Credit: ESO/L. Calçada/M. Kornmessey, CC-BY-4.0

A bit of self intro

- ▼ I decided to do something in social science for my PhD, so I came to a field called “quantitative science studies”
- ▼ I am supposed to work on social modeling, simulation, and data analysis
- ▼ And I keep contributing to open source software as a hobby

Then my primary PhD project became a disaster

- ▼ Partially because of the pandemic
- ▼ The data that I am supposed to rely on is still not ready
- ▼ The data + analysis infrastructure is a mess

Rescue my PhD

- ▼ Need something that I can work it out by myself
- ▼ Know something about physics and social modeling
- ▼ Also kind of familiar with software engineering, thanks to my open source contribution hobby

Rescue my PhD

- ▼ Need something that I can work it out by myself
- ▼ Know something about physics and social modeling
- ▼ Also kind of familiar with software engineering, thanks to my open source contribution hobby
- ▼ (Astro)physics + social modeling + software engineering =
Interstellar social simulation framework

Interstellar social simulation – big questions

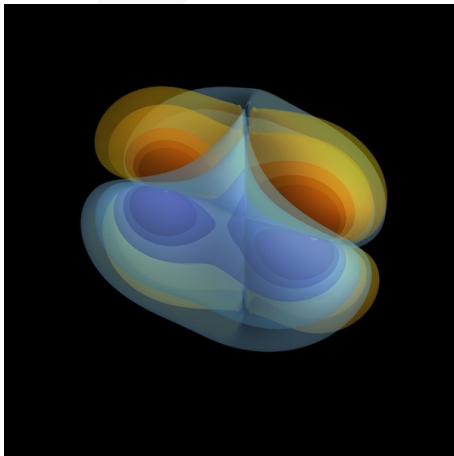
- ▼ Will human civilization become interstellar?
- ▼ Are we alone? Does extraterrestrial intelligence exist?
- ▼ Social theory that remains to be a good approximation?
 - ▼ Utility seeking
 - ▼ Collective action
 - ▼ ...
- ▼ Use simulation for exploration

Agent based modeling

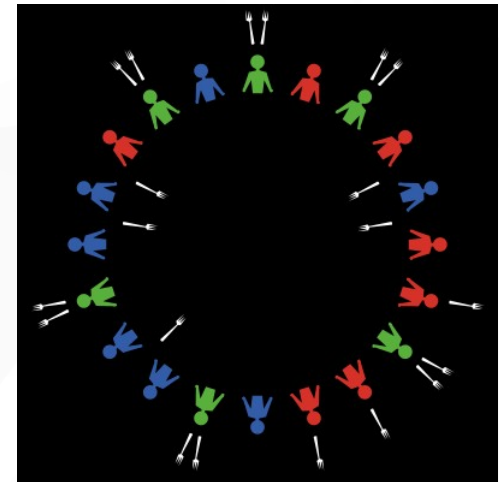
- ▼ Design rules for individual agents to capture human behaviors
- ▼ Put agents in a computer world to interact with each other
- ▼ Interpret the macroscopic simulation outcome
- ▼ There are many existing agent-based simulation framework

Why a new framework?

- ▼ Existing social simulation (agent-based) framework, e.g. NetLogo, Repast, Mesa, etc.
 - ▼ Not trivial to deal with the physics
- ▼ Existing physics simulation framework, e.g., Einstein toolkit
 - ▼ Not created for social simulation



Credit: North-German Supercomputing Alliance, CC-BY-4.0



Credit: Uri Wilensky, CC BY-NC-SA 3.0

The framework

The physics: special relativity

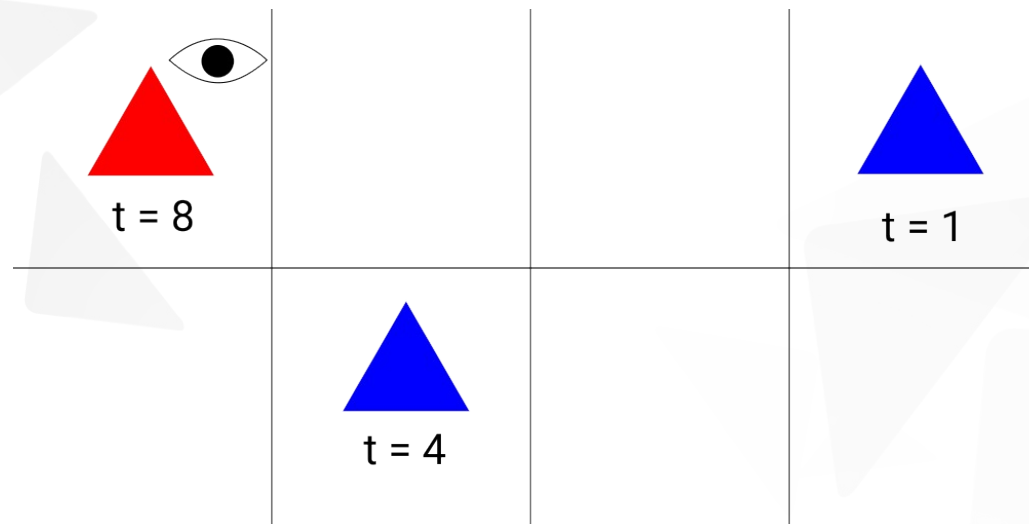
- ▼ Sufficient when spatial scale = ~ 100 light years
- ▼ Time delay: nothing can travel faster than the speed of light
- ▼ Time dilation: moving clocks tick slower

The framework needs to deal with

- ▼ What can an agent see?
- ▼ How an agent interacts with other agents?
- ▼ How the model changes the state of an agent?

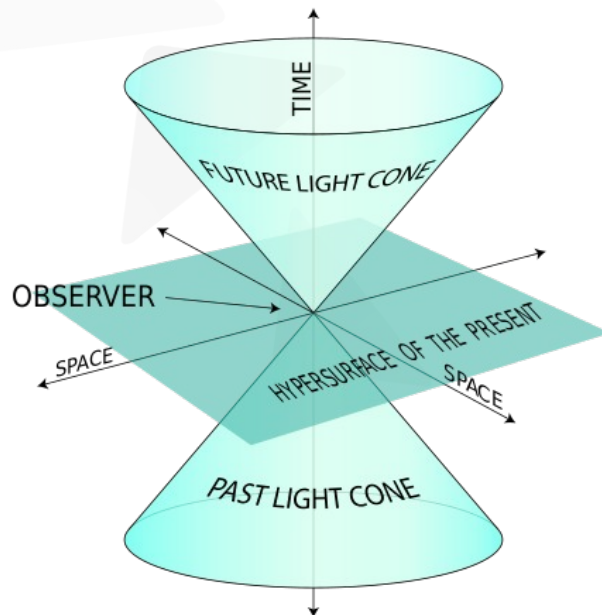
What can an agent see?

- Information travels at the speed of light, so every agent sees the past of other agents



What can an agent see?

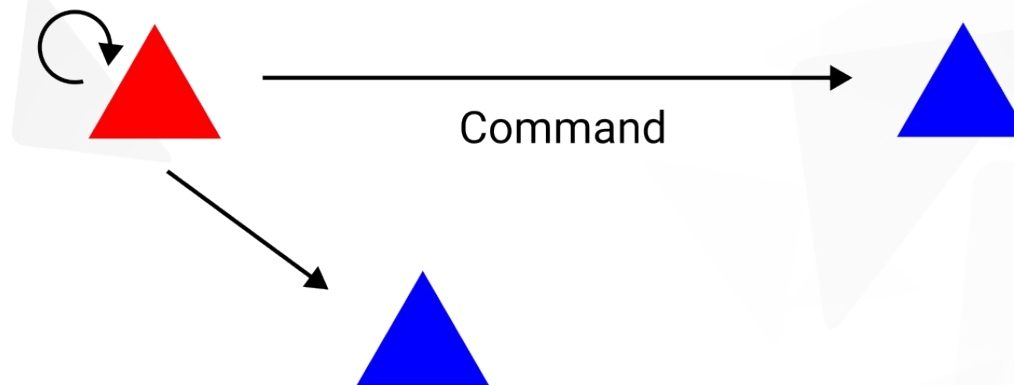
- Information travels at the speed of light, so every agent sees the past of other agents
- Technically, each agent sees the “past light cone”



Credit: K. Aainsqats, CC BY-SA 3.0

How an agent interacts with other agents?

- ▼ Command
 - ▼ Send
 - ▼ Wait for a delay (travels at the speed of light)
 - ▼ Executes on the target
- ▼ Agent may send a command to self



How the model changes the state of an agent?

- ▼ Mechanisms: change the internal state of an agent directly and send commands out
- ▼ Recall: the clock ticks slower when an agent moves

How the model changes the state of an agent?

- ▼ Regular mechanism

- ▼ Process once per turn



- ▼ Dilated mechanism

- ▼ Process once per n turn on average

$$n = \gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}$$

- ▼ The faster an agent move, the less frequent the mechanism is processed

How the model changes the state of an agent?

- ▼ Regular mechanism

- ▼ E.g., observe and update stored information

- ▼ Dilated mechanism

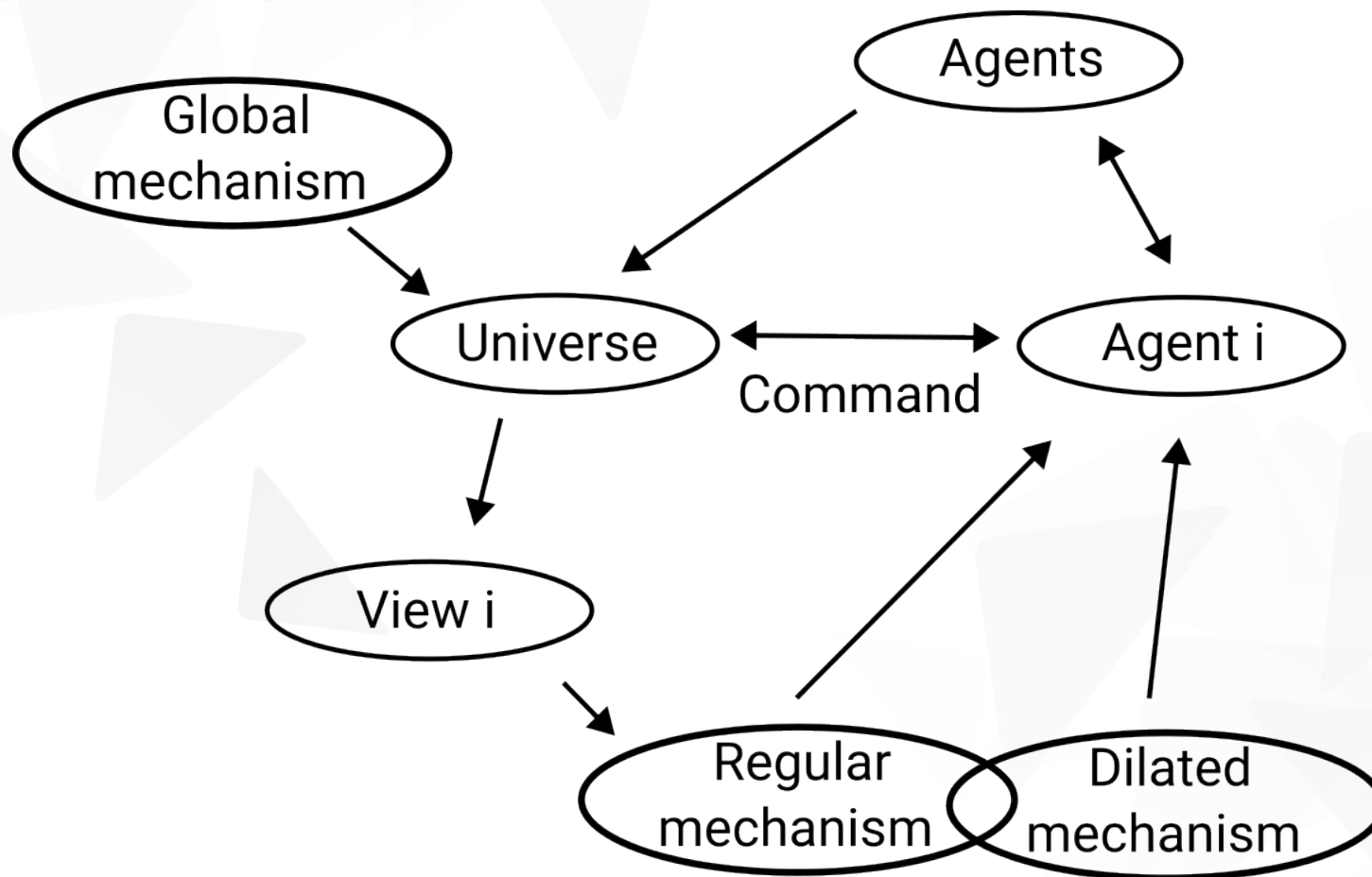
- ▼ E.g. manufacturing

Our simulation framework: Relativitization

- ▼ A framework written in Kotlin
- ▼ Enforce a specific way to implement models to account for the physics
- ▼ Provide functionalities to ease the development of interstellar social models
- ▼ Deal with technical subtleties such as parallelization, serialization, deterministic simulation, etc

Relativitization

- “On social simulation in 4D relativistic spacetime”:
<https://arxiv.org/abs/2206.11019>



How to use the framework

- ▼ Instead of providing a library, we decided that it should be used in the form of source code
- ▼ Technical reason: deserialization from string to object in pure Kotlin
- ▼ Run a gradle task to create a template
- ▼ Build the model on top of the template – implement the data components, commands, and mechanisms



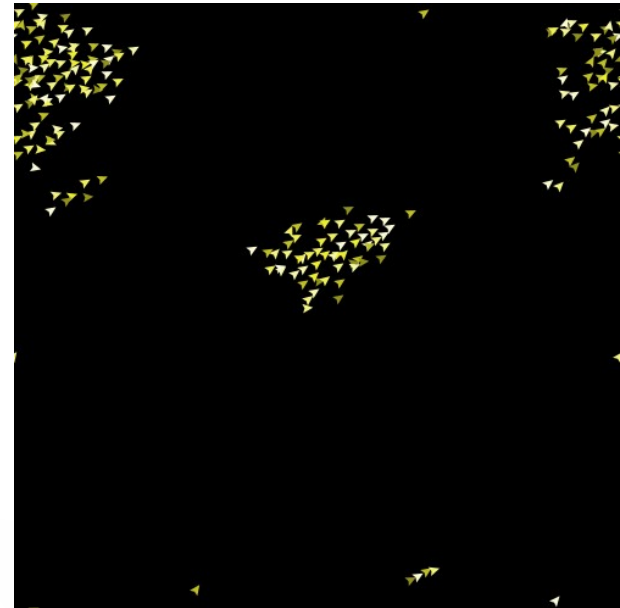
Examples

Flocking model

- Simple alignment rules \Rightarrow interesting macroscopic pattern



Credit: John Holmes, CC BY-SA 2.0



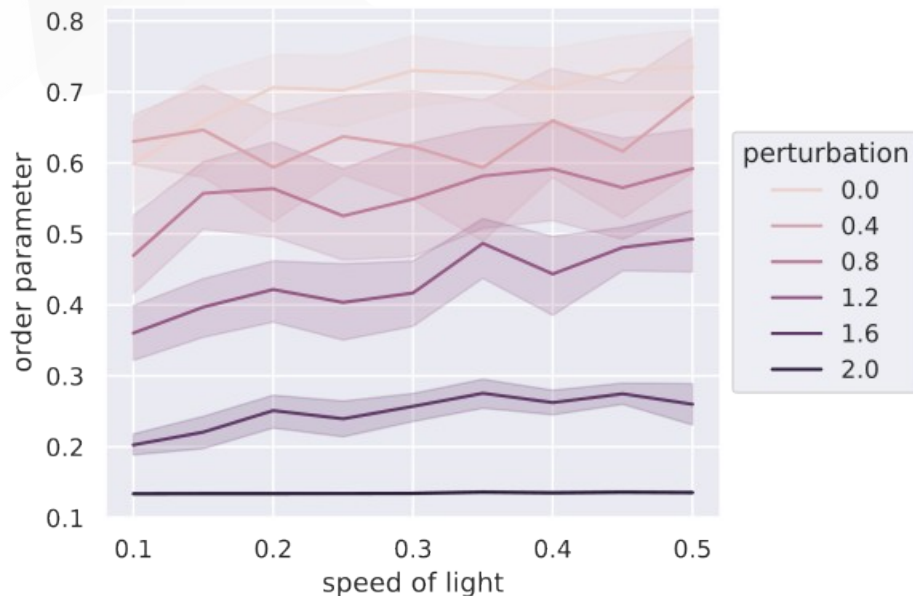
Credit: Uri Wilensky, CC BY-NC-SA 3.0

Interstellar flocking model

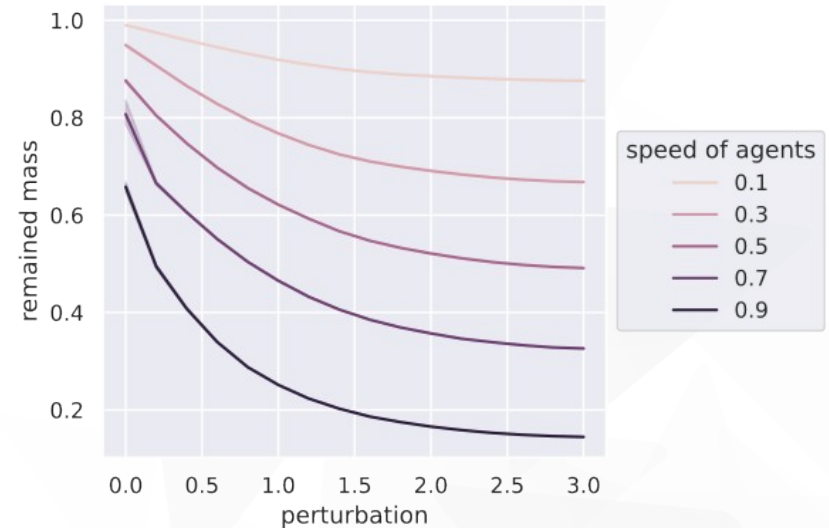
- ▼ Hypothetical scenario: what if some kinds of flocks (spaceship?) are trying to align with each other (with a perturbation) in the space
- ▼ Propelled by photon rocket: consume mass to accelerate
- ▼ Order parameter: how well the flocks align
- ▼ <https://github.com/Adriankhl/relativitization-model-flocking>

Interstellar flocking model

- ▼ Speed of light is slower → takes longer to communicate → flock are less ordered

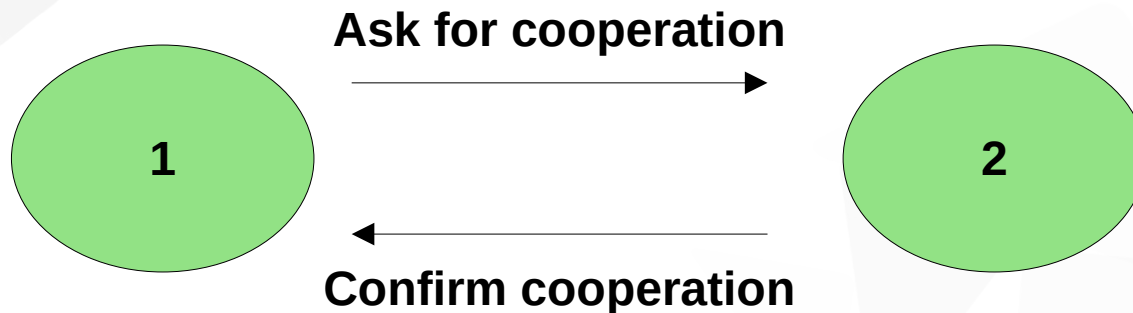


- ▼ Stronger perturbation → photon rocket consume more mass to align with each other → less remained mass



Interstellar knowledge dynamics

- ▼ Cooperation stimulates innovation
- ▼ Takes time to seek cooperation → affected by time delay imposed by special relativity



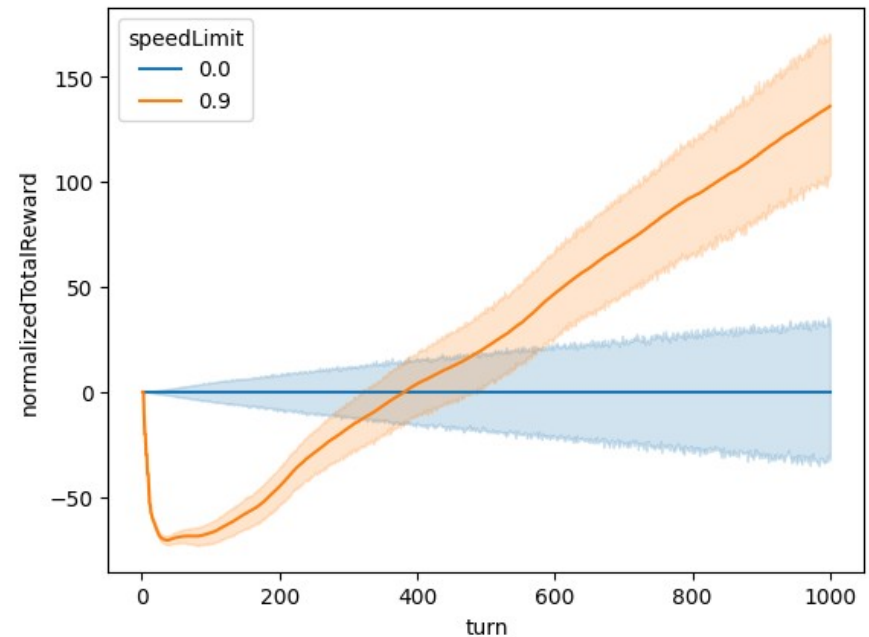
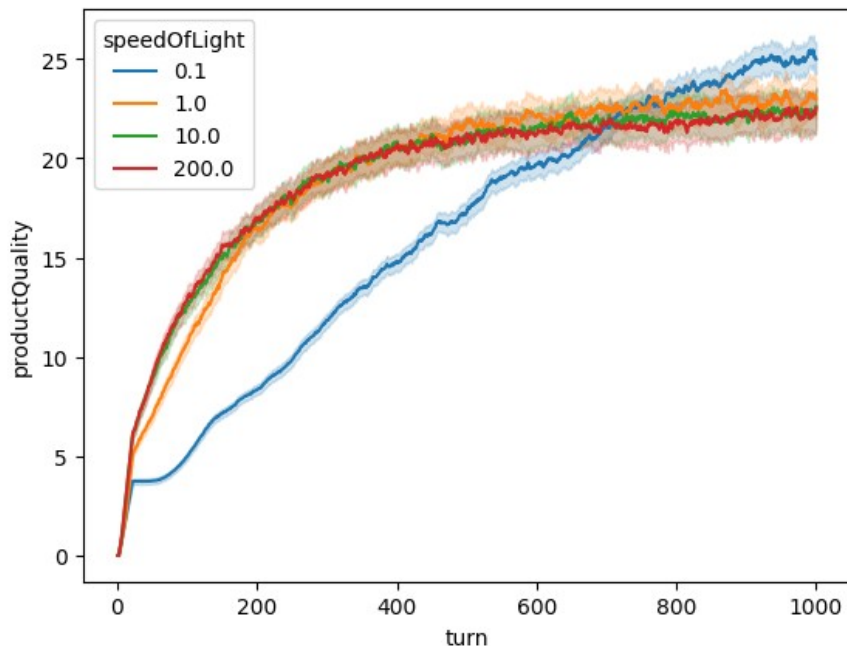
Interstellar knowledge dynamics

- ▼ Cooperation stimulates innovation
- ▼ Takes time to seek cooperation → affected by time delay imposed by special relativity
- ▼ Smaller time delay → more competitive
- ▼ Research also takes time → affected by time dilation if agents move

Interstellar knowledge dynamics

- ▼ Slower speed of light → worse innovation initially, but becomes better finally because it is less competitive

- ▼ Move closer to neighbor → fewer innovations initially because of the time dilation, but it pays off later



Criticism

- ▼ This is too hypothetical
- ▼ Not interesting to some people
- ▼ So let's make a game to make it more interesting

Why make a game?

- ▼ I like simulation game
- ▼ Turn-based strategy/simulation game is essentially a big agent-based model
- ▼ I have many ideas, but I don't want to be too (academically) rigorous
- ▼ One of / the only sensible way to get something from the framework



The game

Game mechanics

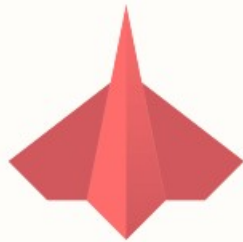
- ▼ How the economy should look like given the huge time delay and the possibility of time dilation? Credit-based economy vs barter economy? Fuel rest mass as money?
- ▼ Political / administrative system? Does democracy work?
- ▼ How can scientific research across stellar systems be optimized?

Game mechanics

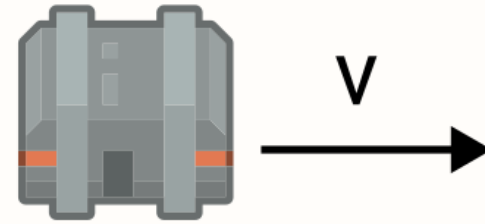
- ▼ Population dynamics
- ▼ Politics
- ▼ Diplomacy
- ▼ Economics
- ▼ War
- ▼ Science and technology
- ▼ ...



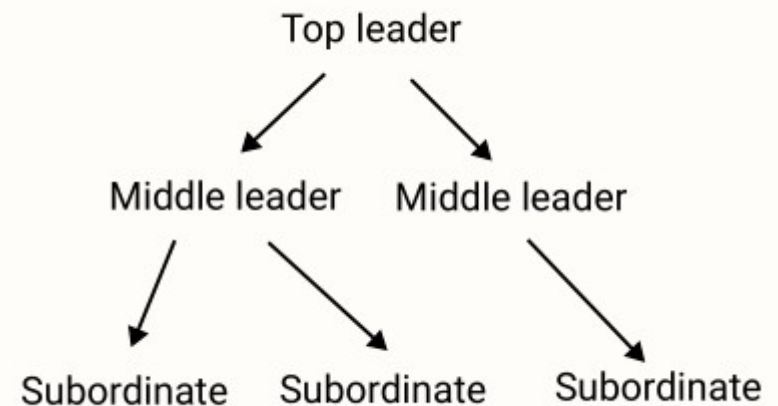
Stellar system



Spaceship



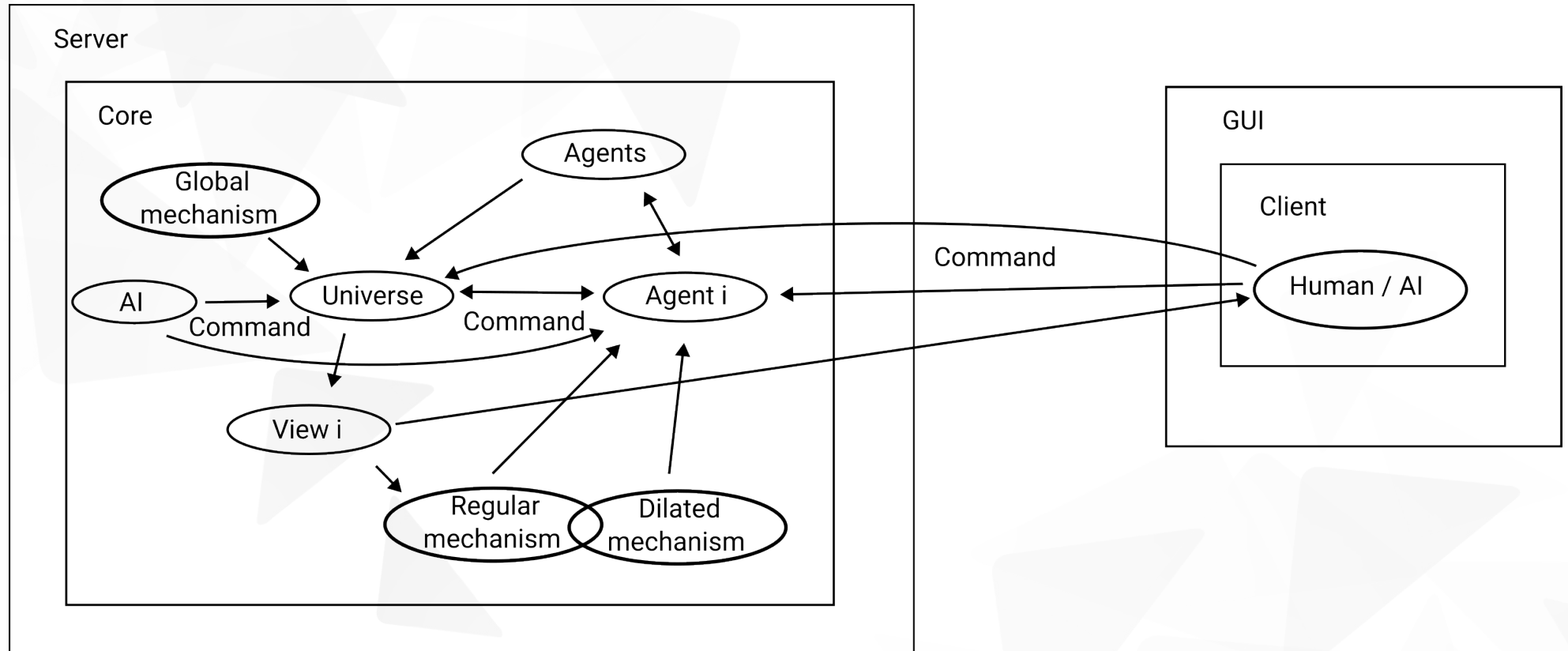
A moving factory manufactures products slower.



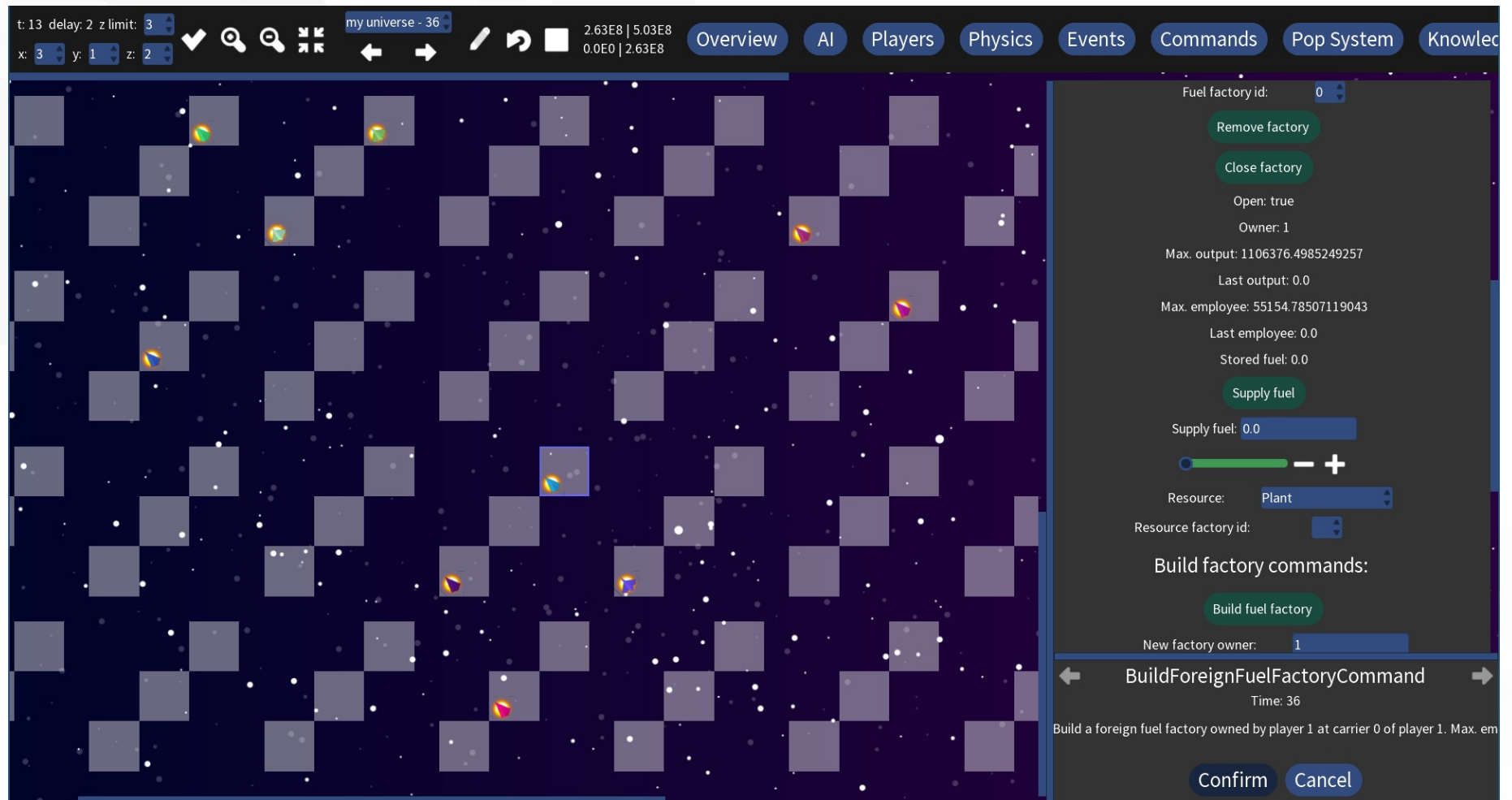
Architecture

- ▼ GUI that runs on PC and Android devices
- ▼ Server-client architecture to allow multiplayer game
- ▼ Limit what a client (human/AI) can do: receives a view of the universe, and send back a list of commands
- ▼ (Dual)-utility AI

Architecture



Screenshot



Open source dependencies

- ▼ GUI: libGDX
- ▼ Parallelization: kotlinx-coroutine
- ▼ Server-client: ktor
- ▼ Serialization: kotlinx-serialization
- ▼ IO: okio
- ▼ Simulation data organization: kotlinx-dataframe
- ▼ Unit testing: kotlin.test
- ▼ Art: creative-common licensed assets, inkscape



Discussion

The good

- ▼ I have learned a lot by contributing to open source software
- ▼ The open source ecosystem gave me plenty of building blocks for my own project
- ▼ So it is natural for me to open source my project
- ▼ I have created something interesting

The bad

- ▼ Lack of open source culture in my workplace
- ▼ As an one-man project, it is a bit too ambitious, I don't have time to polish things, e.g. GUI, documentation, fun game mechanics, translations, etc.
- ▼ I decided to go fully open source, but perhaps industry-standard closed source solutions, e.g. Unity game engine, would be better for the project
- ▼ I tried a bit to commercialize my project, i.e., sell my open source game, but I also do not have time to push this forward

Conclusion

- ▼ I have programmed a simulation framework, “Relativitization”, to ease the development of interstellar social models
- ▼ I have created some models based on the framework
- ▼ I have created a turn-based strategy game on top of the framework
- ▼ While lacking immediate practical value, I believe this is a meaningful and educational exploration
- ▼ At least I feel like I have achieved something in my PhD



Thank you

Link

- ▼ The code is licensed under GPLv3, check and star our Github repo
 - ▼ <https://github.com/Adriankhl/relativitization>

