SEMESO 24
Seminário de Engenharia de Software

Introduction to Distributed and Embedded Multi-Agent Systems

Nilson Mori Lazarin

Centro Federal de Educação Tecnológica (CEFET/RJ)

Universidade Federal Fluminense (UFF), Brasil

https://semeso.github.io/2024/minicursos/

UDESC
UNIVERSIDADE DO ESTADO DE SANTA CATARINA

ALTO VALE
CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ

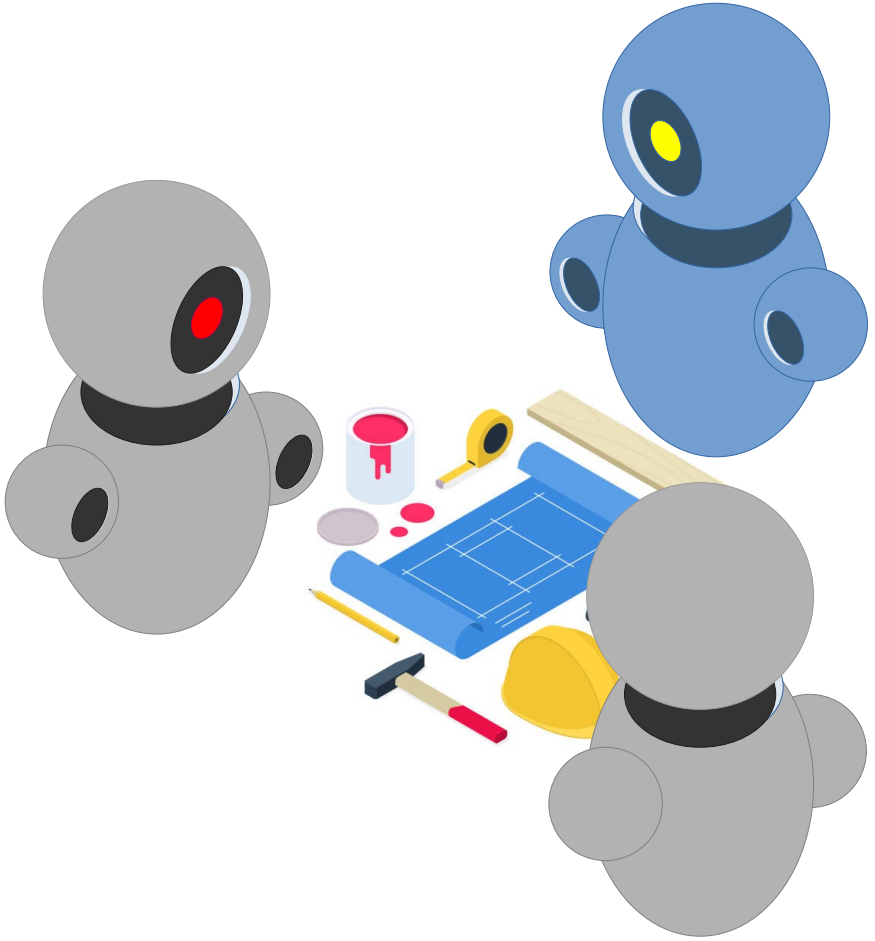
fapesc
Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina

CEFET/RJ
Uned Nova Friburgo

Instituto de Computação

September 2024

INTRODUCTION



Who I am?

Prof. Me. Nilson Lazarin
Cefet/RJ - UFF

Cognitive Hardware on Networks (Chon) Research Group
https://chon.group

Where we have been?

This workshop is part of:

- **Undergratuate course**
  - CEFET/RJ Maria da Graça and Nova Friburgo
- **Graduate course**
  - M.Sc. and Ph.D. at UFF

It was **presented at:**

IFSP, UFPel, UFRJ, UFSC, UnB, UTFPR and Estácio;

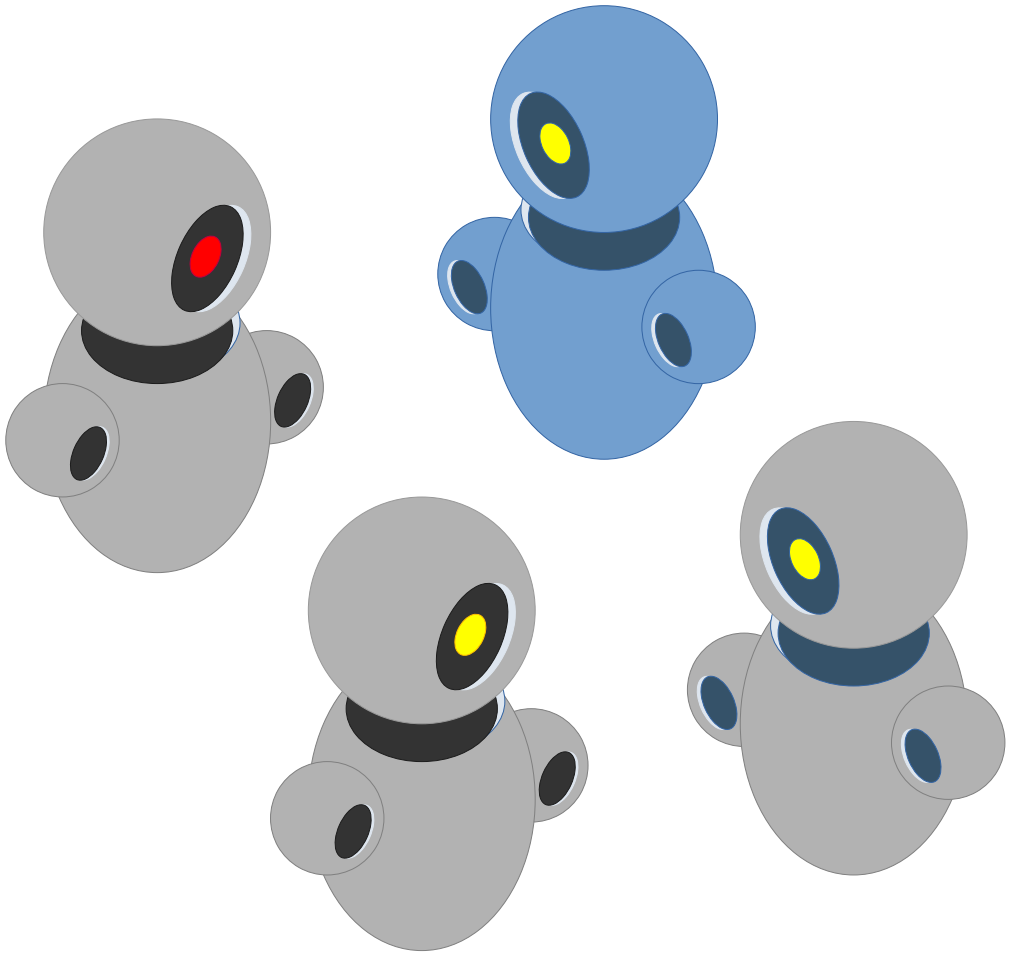
https://workshops.chon.group

Goals

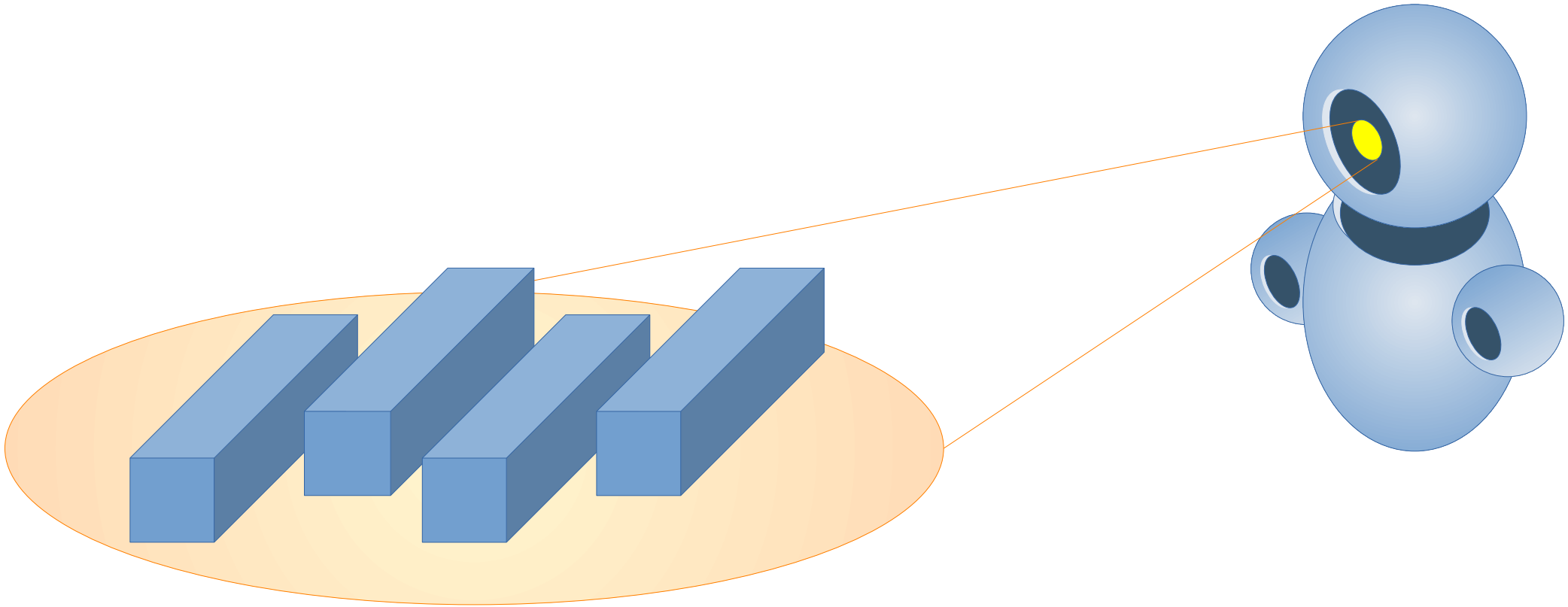
This workshop goals are:

- **To experience** the development process of **Embedded MAS** using BDI-agents and Jason.
- to learn how to **collaborate** using IoT in **distributed** scenarios.

THE AGENT ORIENTED APPROACH

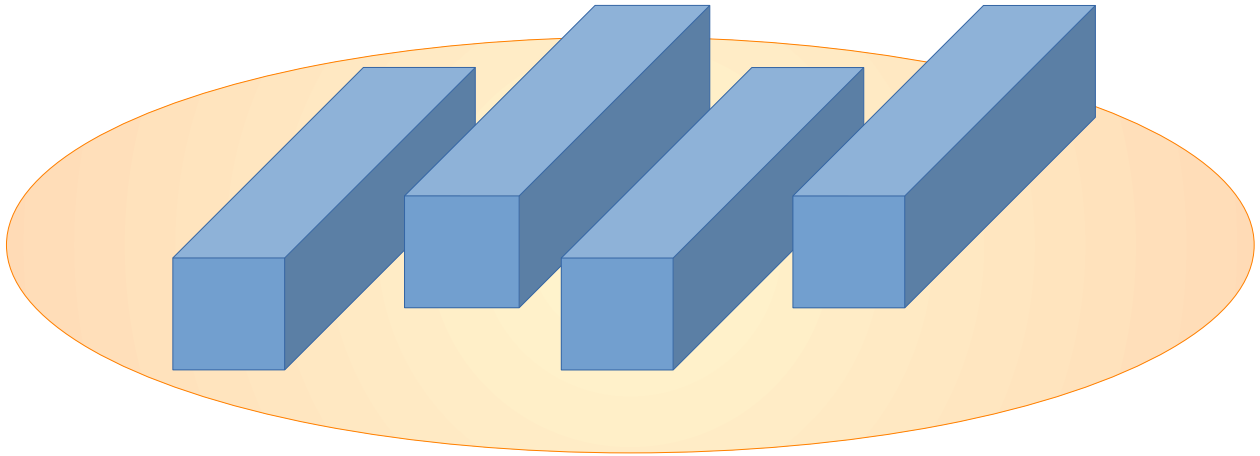


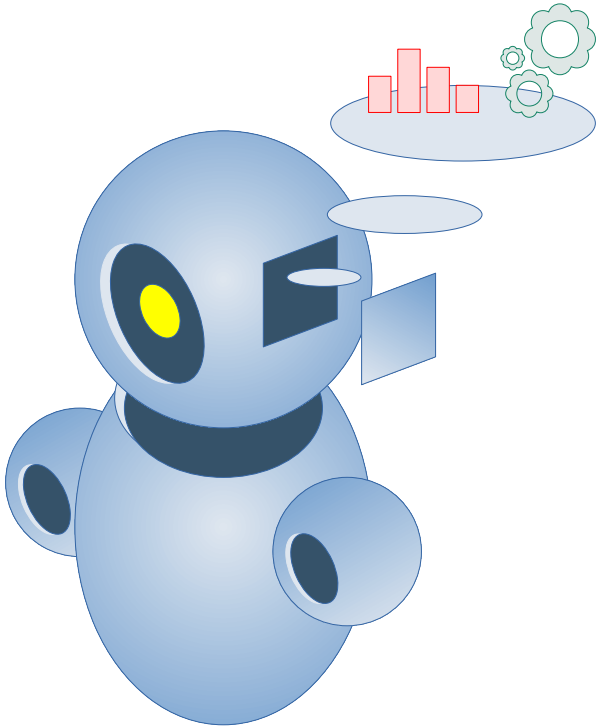
Agent



WOOLDRIDGE, Michael. Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1999. p. 27–77
MICHEL, Fabien; FERBER, Jacques; DROGOUL, Alexis. Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. 2009

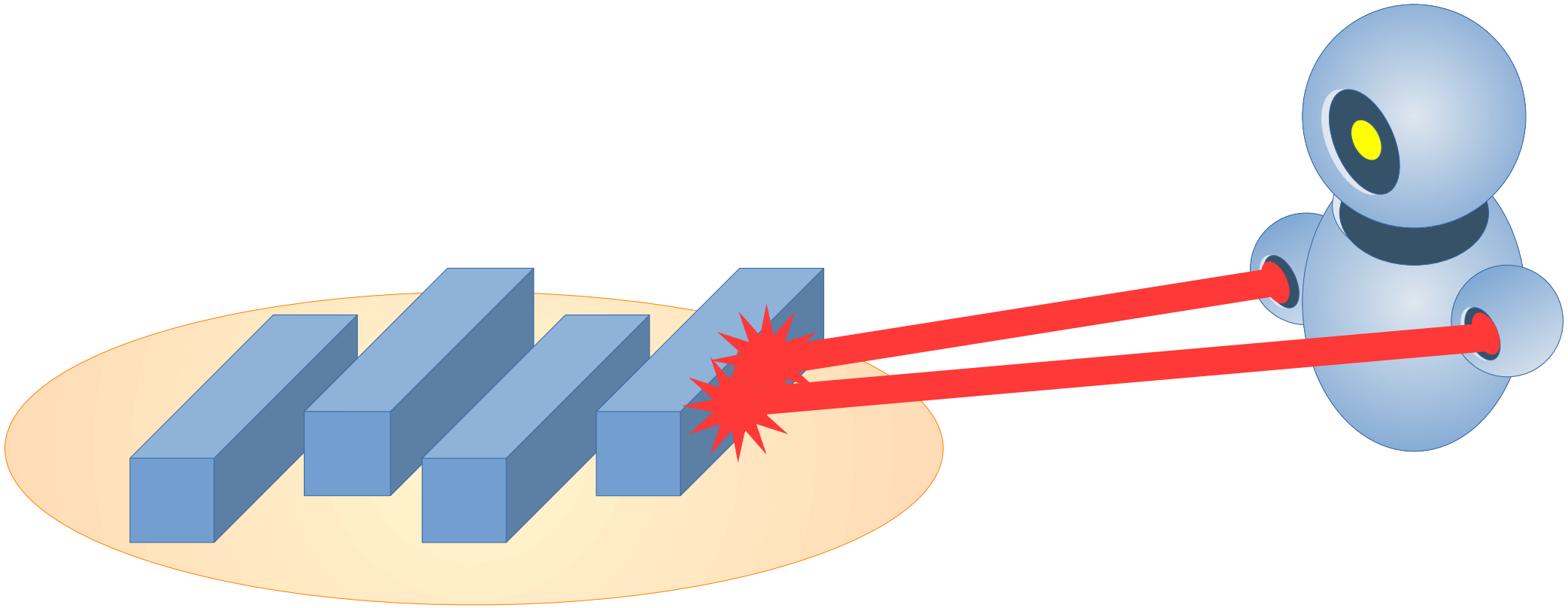
Agent





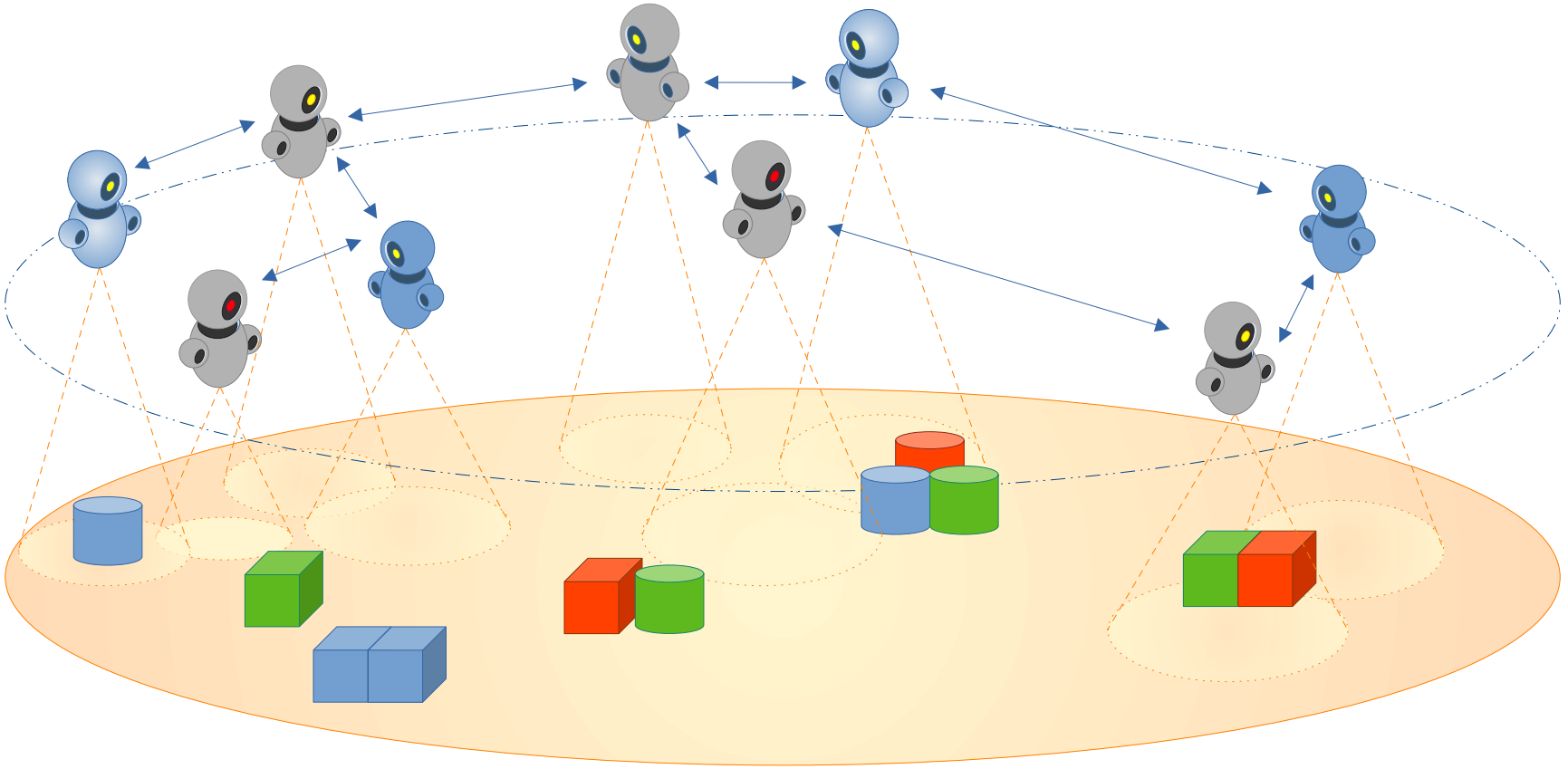
WOOLDRIDGE, Michael. Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1999. p. 27–77
MICHEL, Fabien; FERBER, Jacques; DROGOUL, Alexis. Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. 2009

Agent



WOOLDRIDGE, Michael. Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1999. p. 27–77
MICHEL, Fabien; FERBER, Jacques; DROGOUL, Alexis. Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. 2009

Multi-agent System

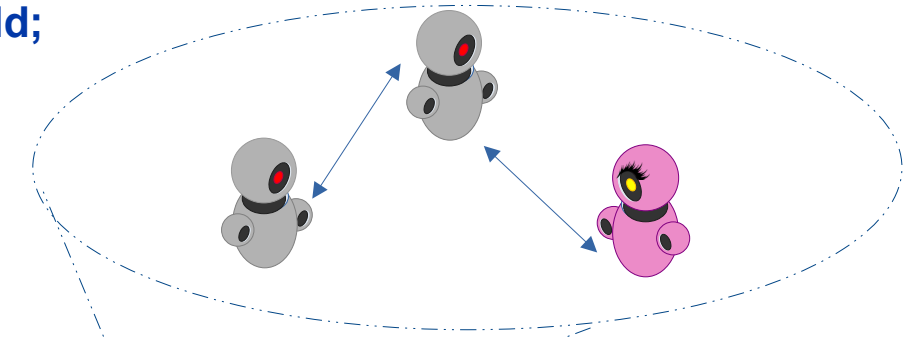


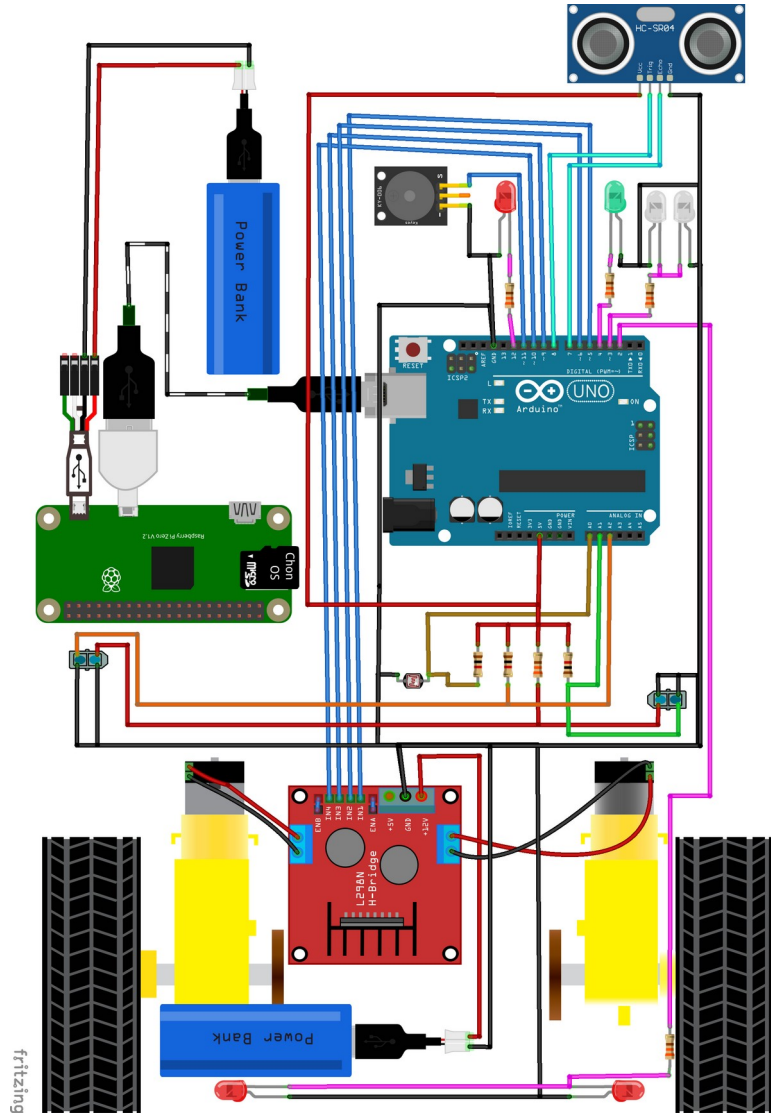
WOOLDRIDGE, Michael. Intelligent Agents. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA: MIT Press, 1999. p. 27–77

Embedded Multi-agent System

Attributes:

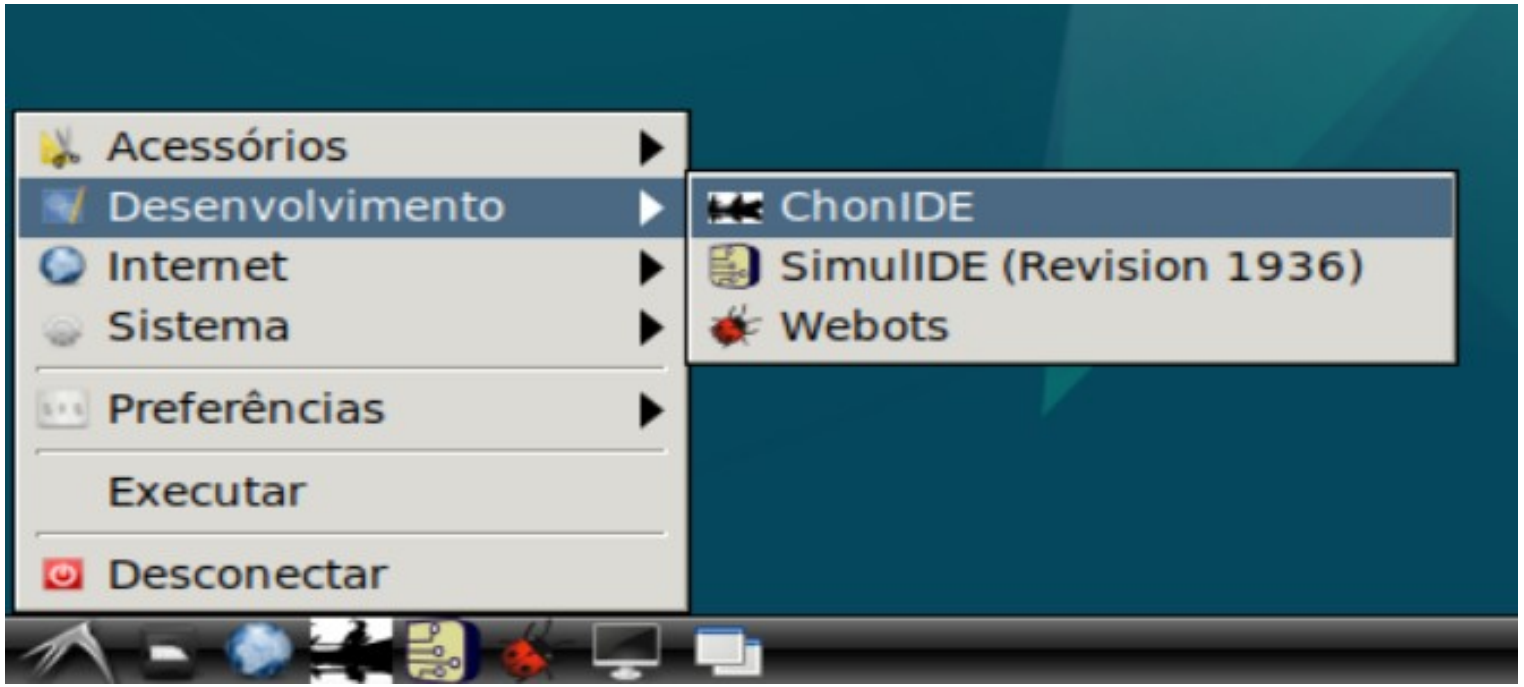
1) Hosted in a cyber-physical system with all sensors and actuator;
2) Acting and perceiving in the real-world;
3) Without remote control;
4) Without external processing;





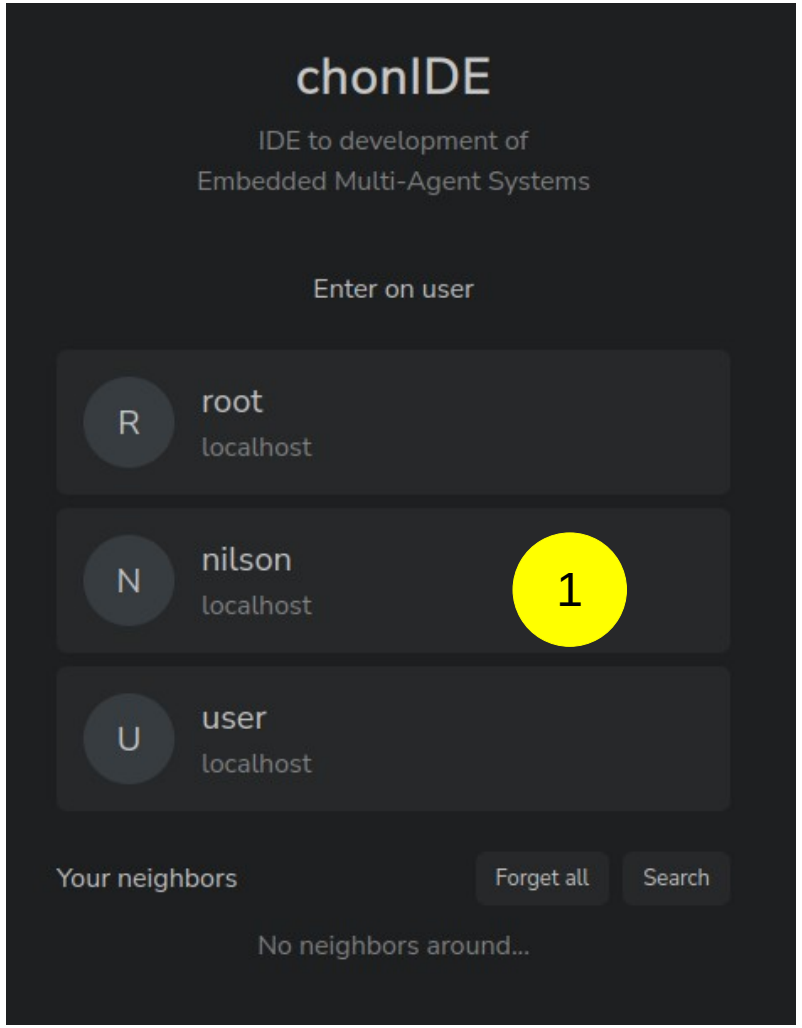
Low-cost 2WD prototype model for Embedded Multi-agent Systems https://bots.chon.group

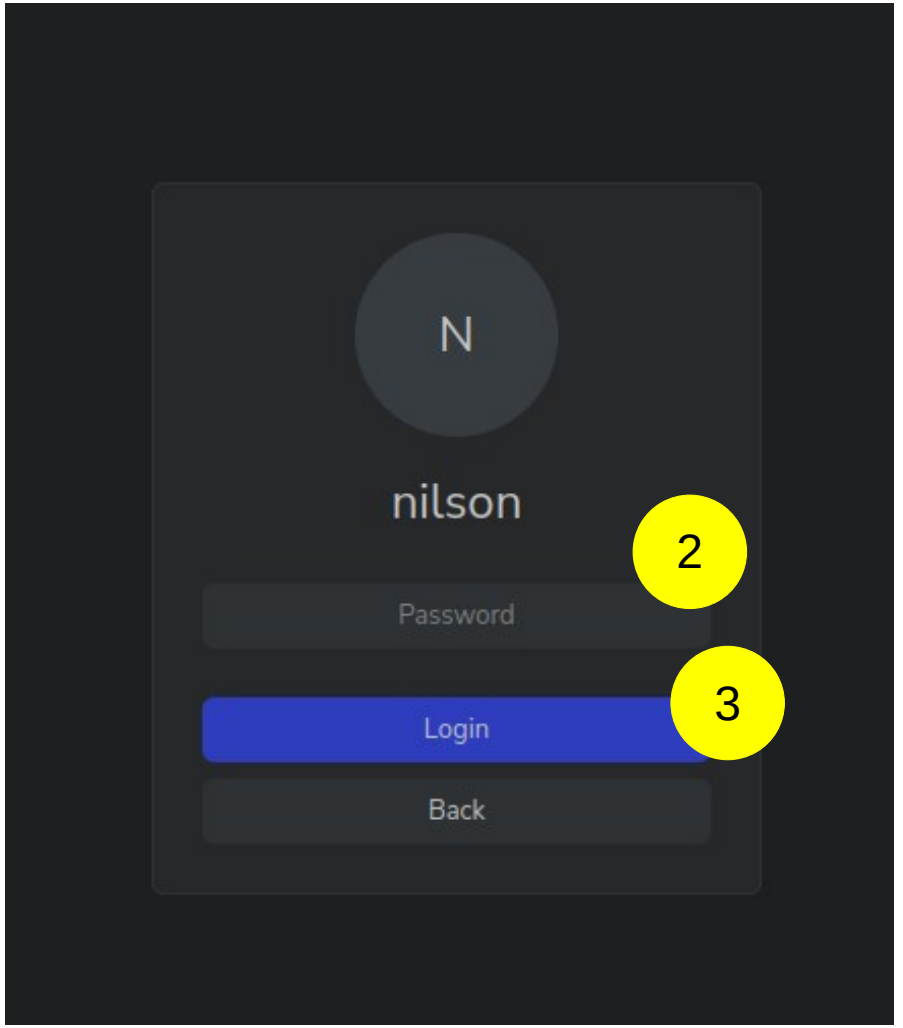
THE DEVELOPMENT TOOLS



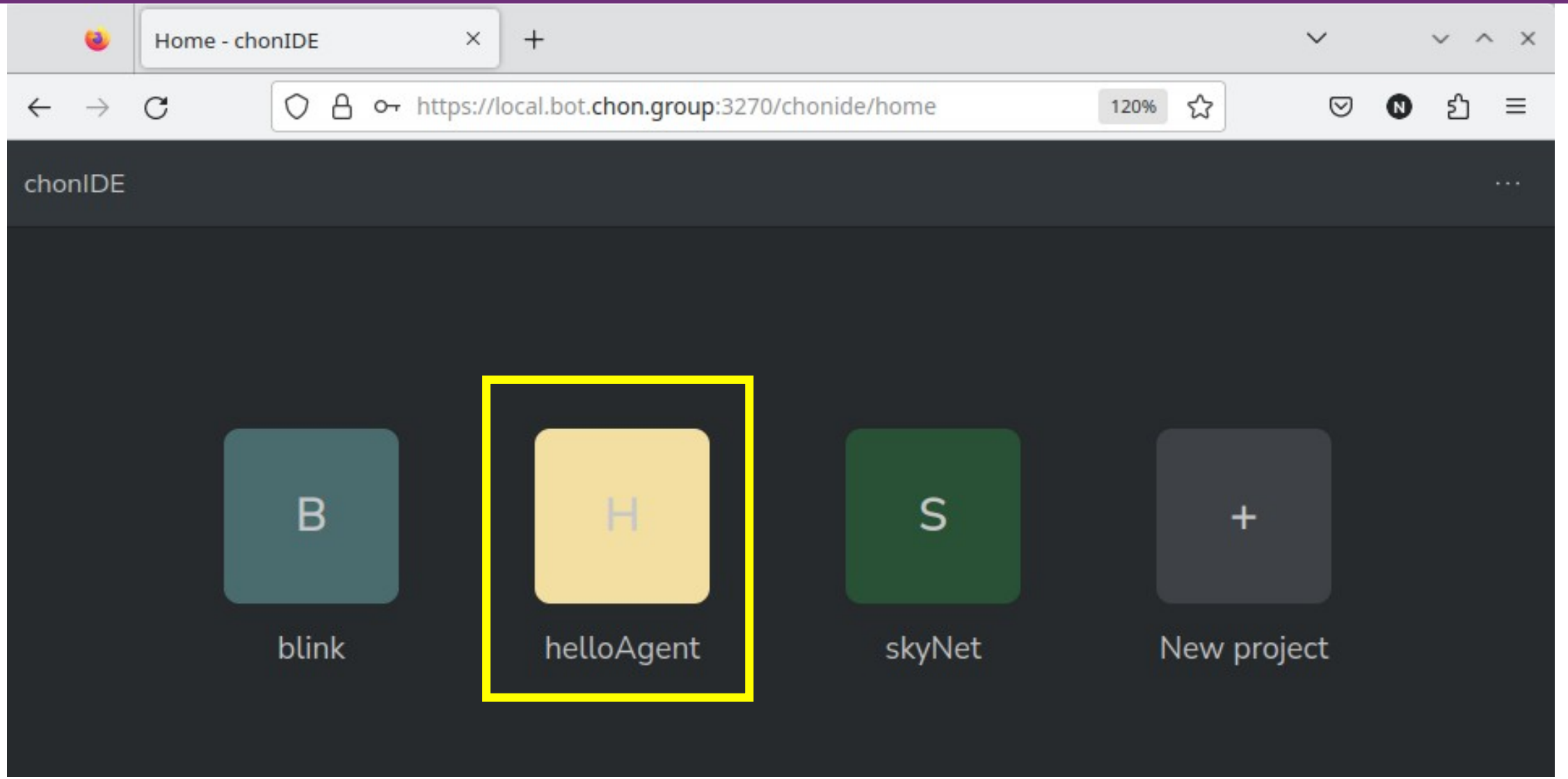
How to install?
https://github.com/chon-group/chonIDE/wiki

ChonIDE: login

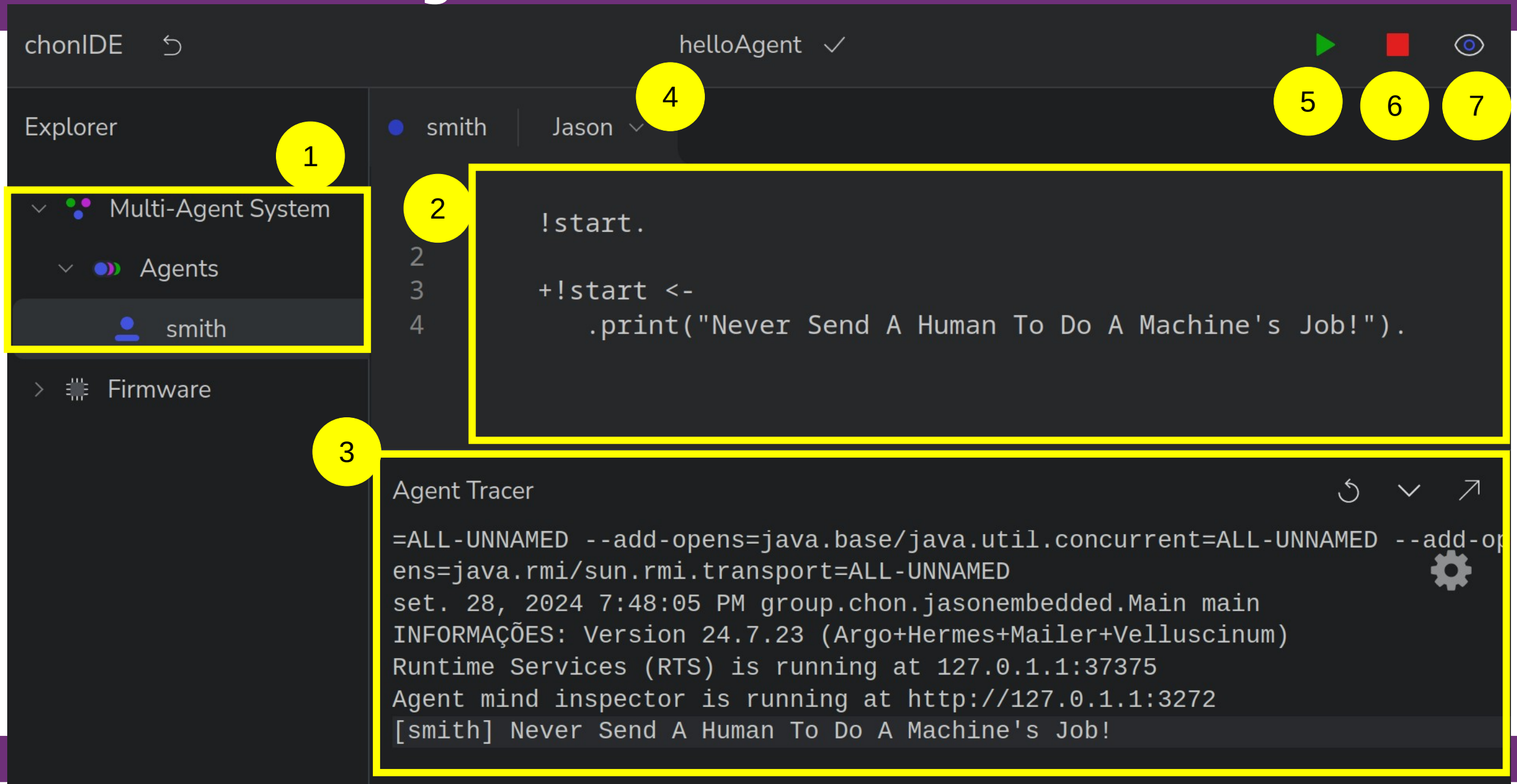




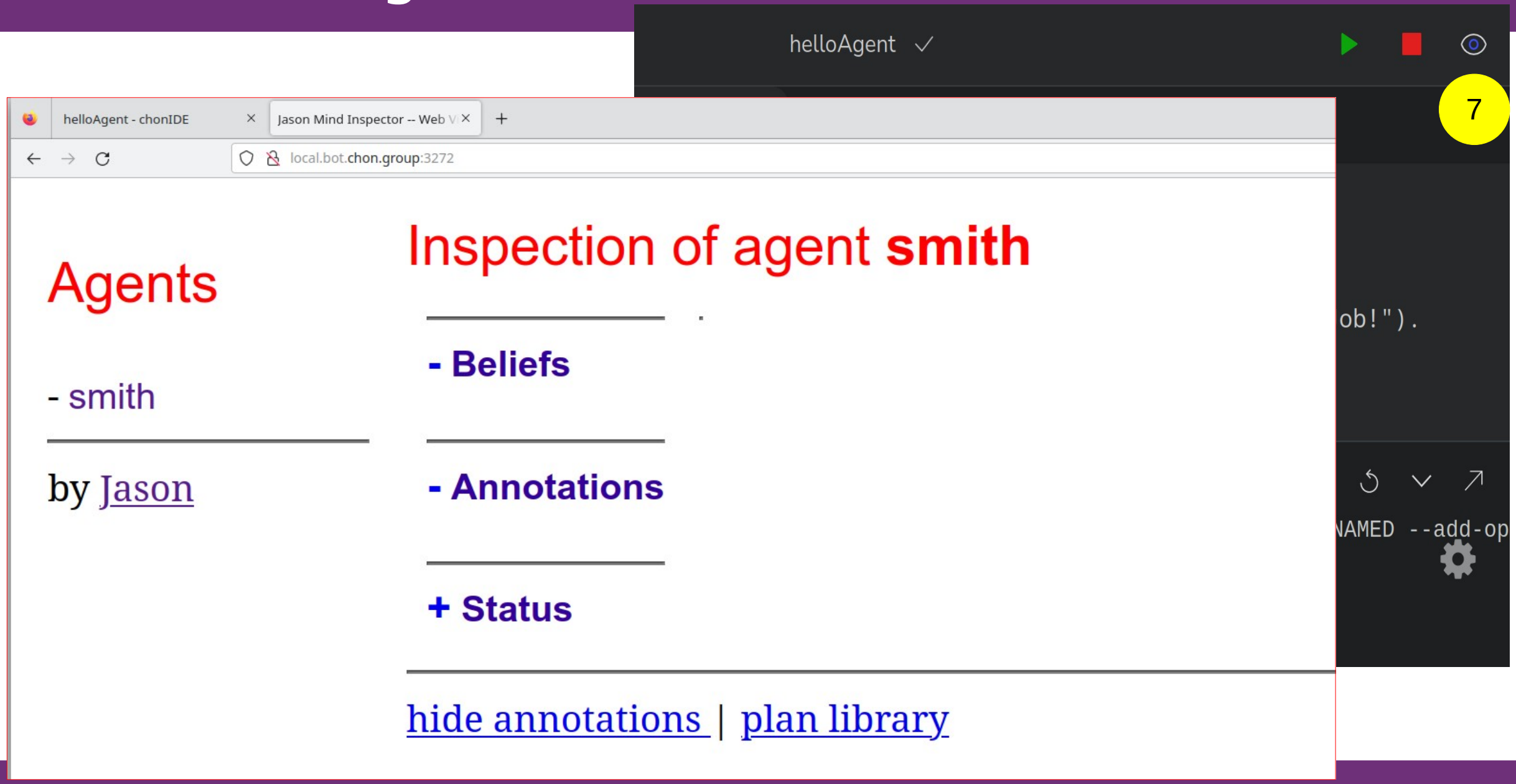
ChonIDE: helloAgent



ChonIDE: helloAgent

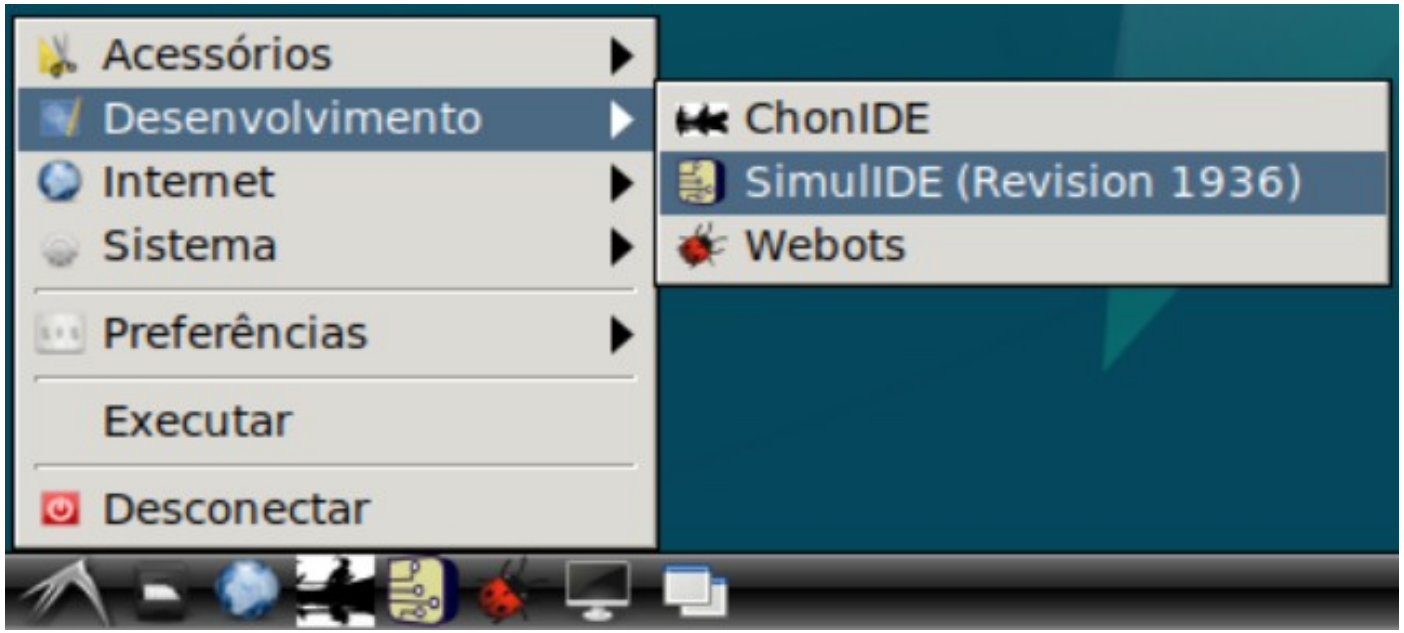


ChonIDE: helloAgent



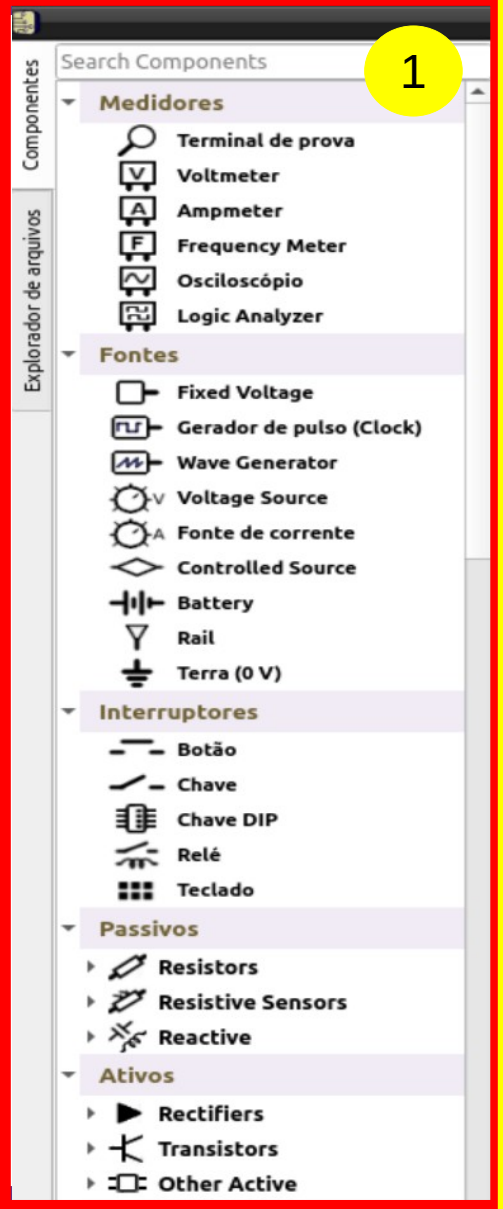
SimulIDE Circuit Simulator

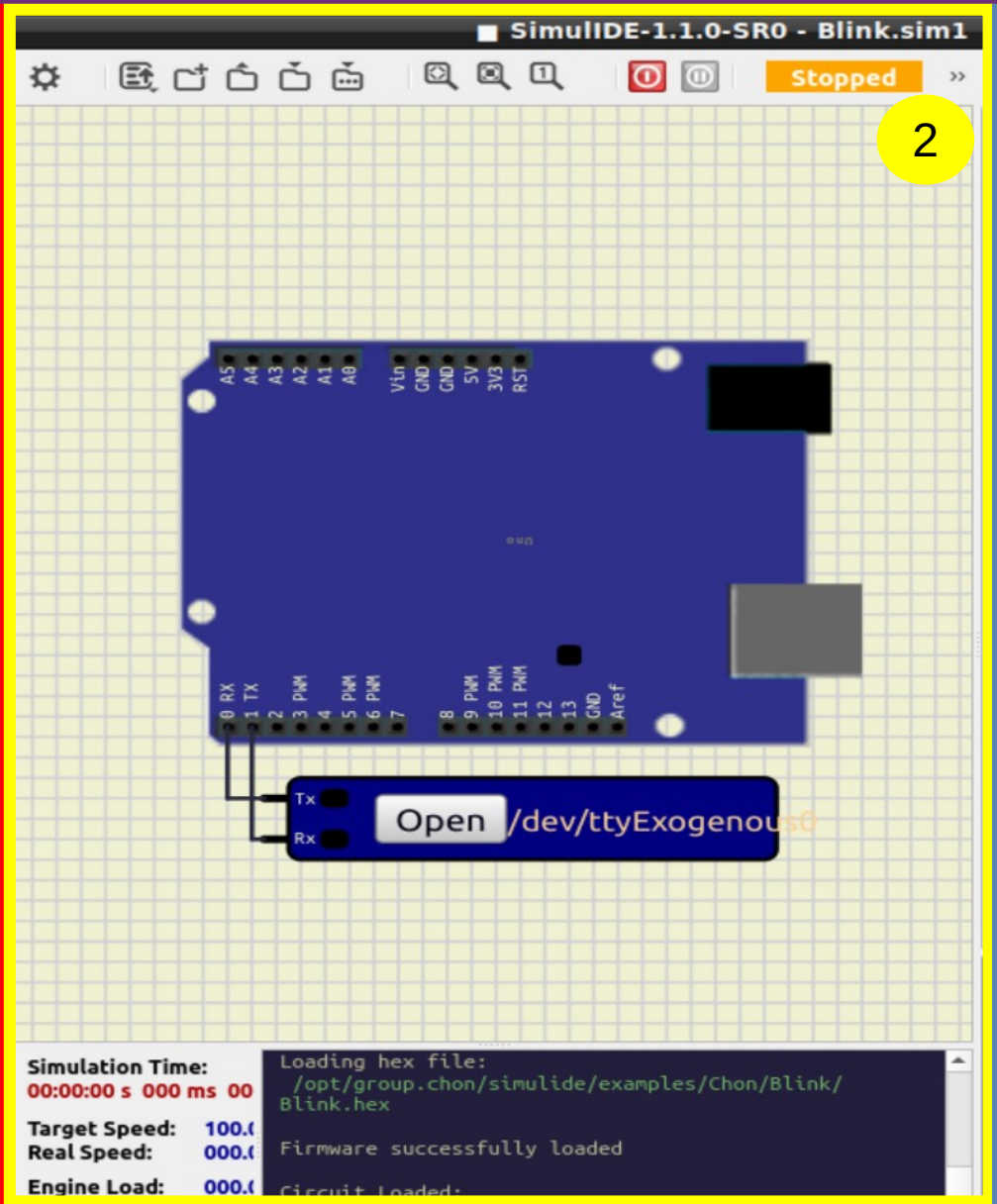
by Chon Group

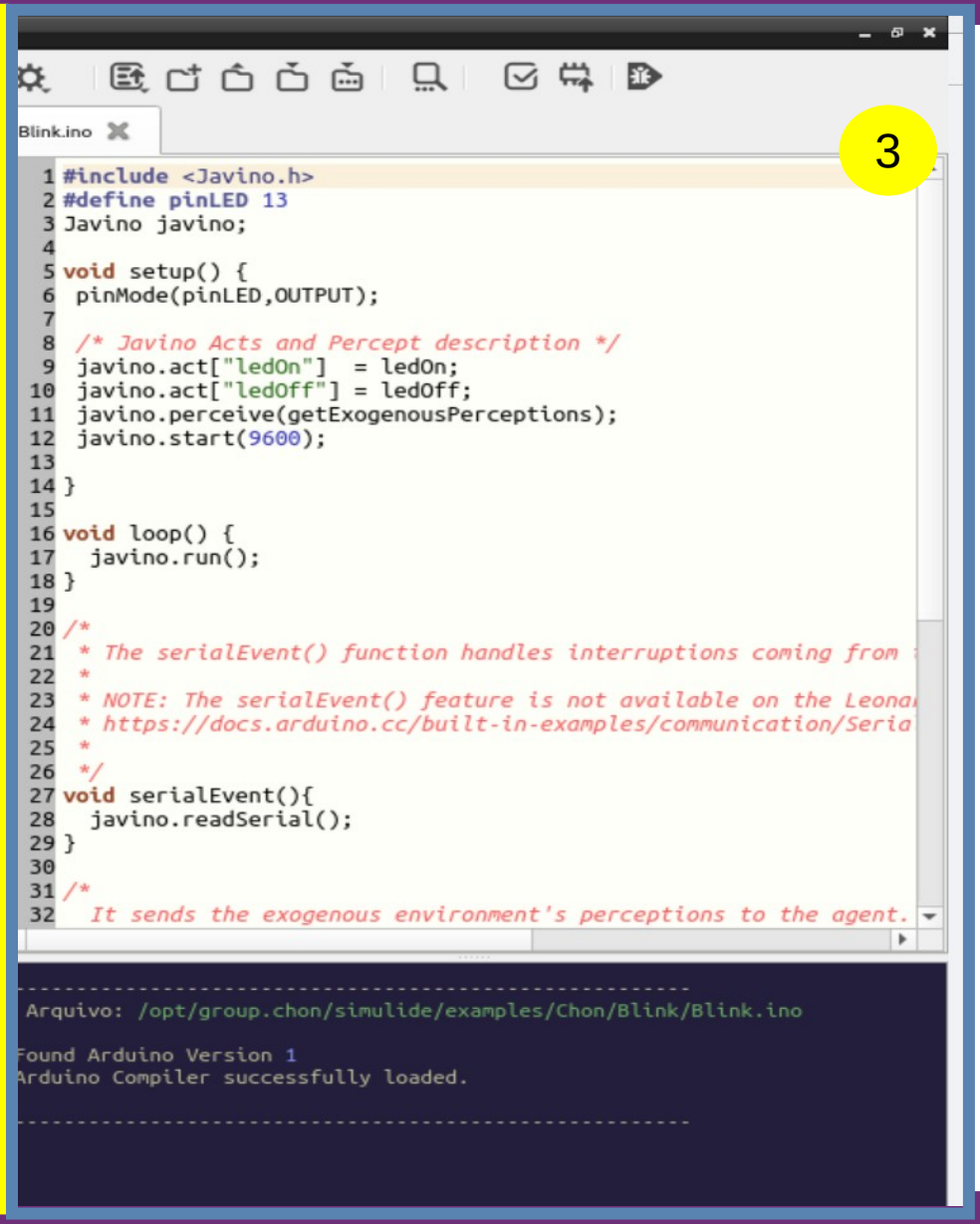


How to install?
https://github.com/chon-group/dpkg-simulide

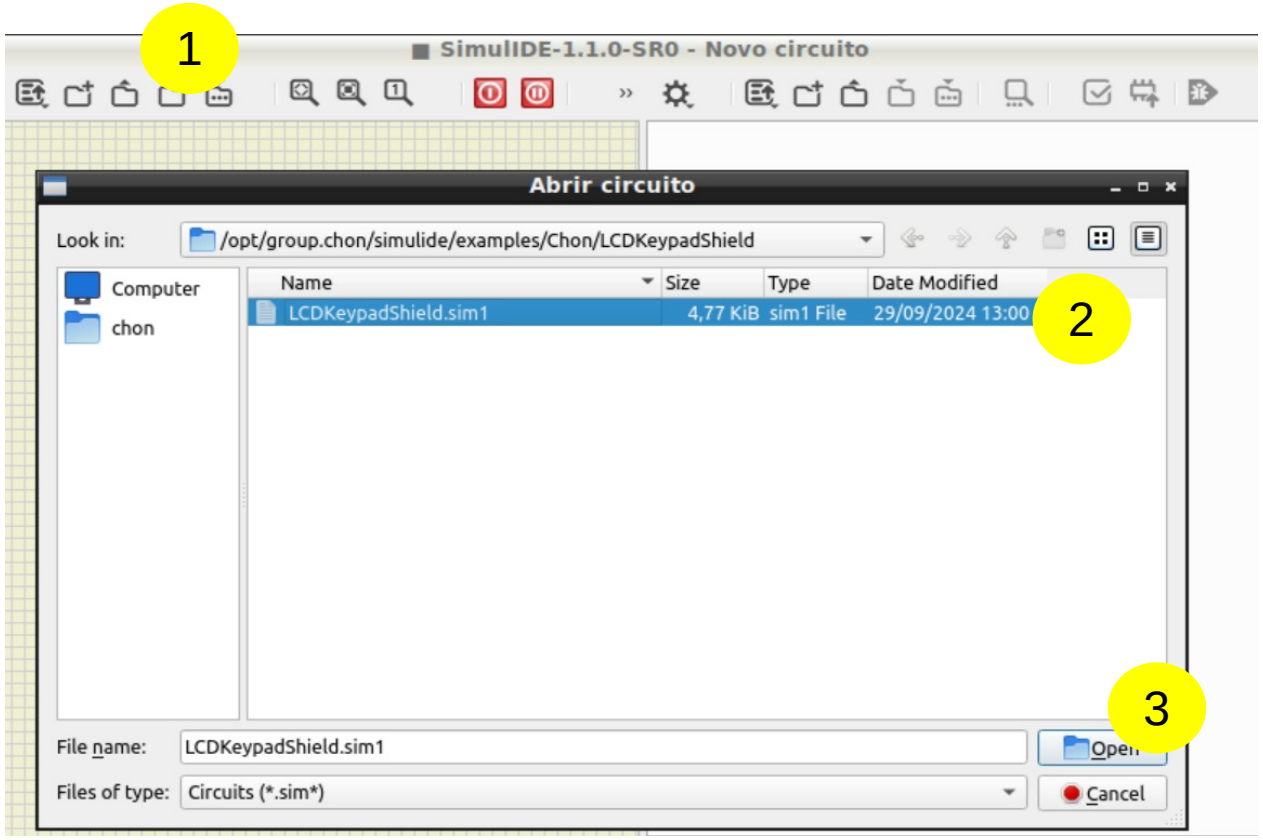
SimulIDE

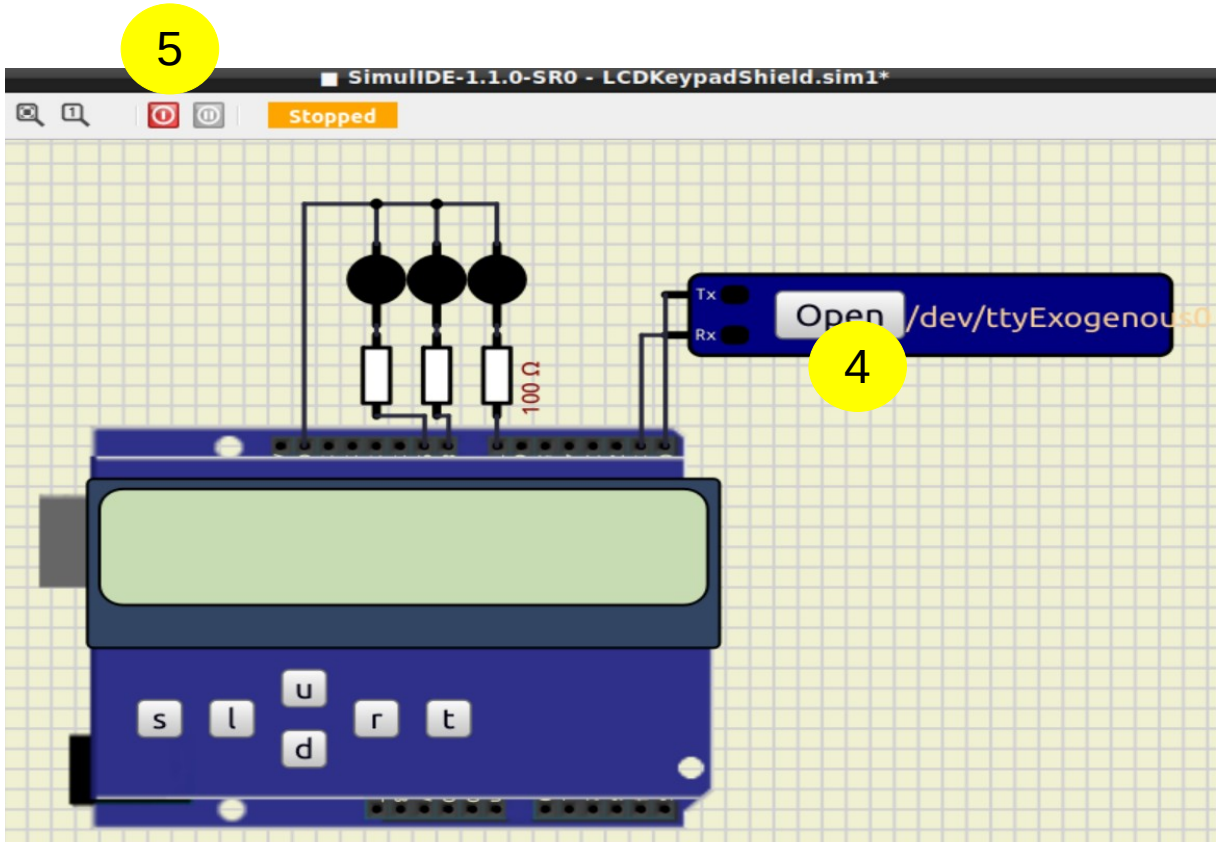

Componentes
Explorador de arquivos
Search Components
1
Medidores
Terminal de prova
Voltmeter
Ampmeter
Frequency Meter
Osciloscópio
Logic Analyzer
Fontes
Fixed Voltage
Gerador de pulso (Clock)
Wave Generator
Voltage Source
Fonte de corrente
Controlled Source
Battery
Rail
Terra (0 V)
Interruptores
Botão
Chave
Chave DIP
Relé
Teclado
Passivos
Resistors
Resistive Sensors
Reactive
Ativos
Rectifiers
Transistors
Other Active


SimulIDE-1.1.0-SR0 - Blink.sim1
Stopped
2
Tx
Rx
Open
/dev/ttyExogenous0
Simulation Time:
00:00:00 s 000 ms 00
Target Speed: 100.(
Real Speed: 000.(
Engine Load: 000.(
Loading hex file:
/opt/group.chon/simulide/examples/Chon/Blink/
Blink.hex
Firmware successfully loaded
Circuit Loaded:

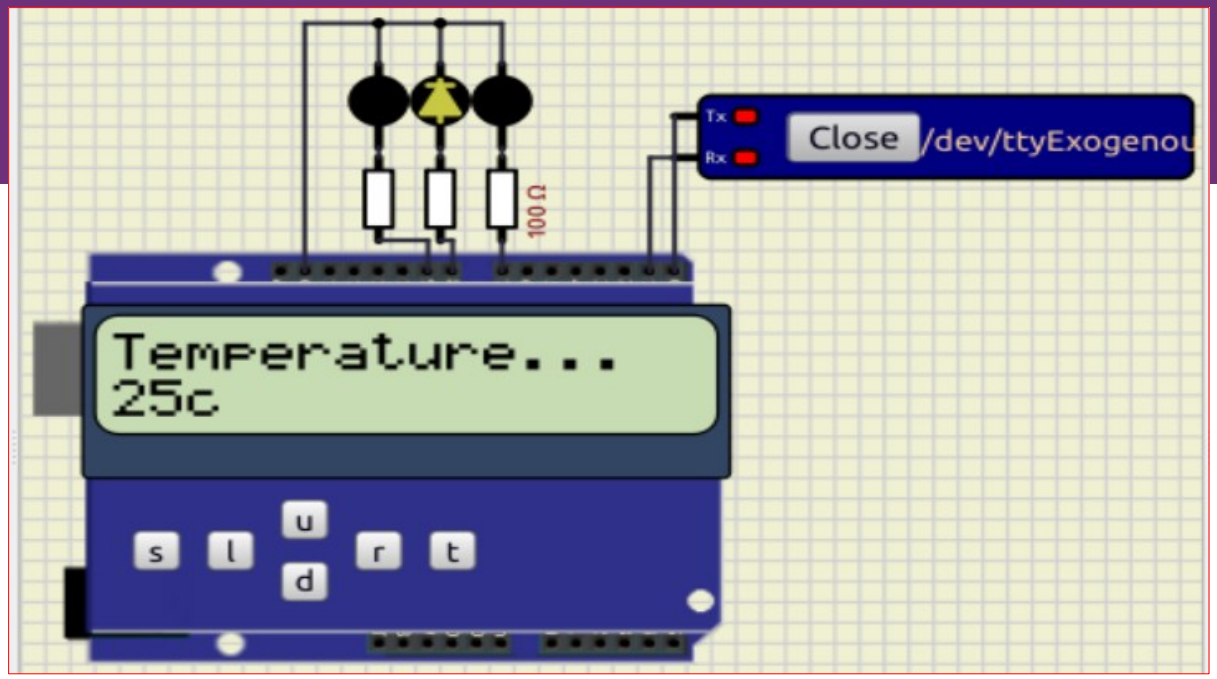

Blink.ino
3
#include <Javino.h>
#define pinLED 13
Javino javino;
void setup() {
pinMode(pinLED,OUTPUT);
/* Javino Acts and Percept description */
javino.act["ledOn"] = ledOn;
javino.act["ledOff"] = ledOff;
javino.perceive(getExogenousPerceptions);
javino.start(9600);
}
void loop() {
javino.run();
}
/*
* The serialEvent() function handles interruptions coming from
*
* NOTE: The serialEvent() feature is not available on the Leona
* https://docs.arduino.cc/built-in-examples/communication/Seria
*
*/
void serialEvent(){
javino.readSerial();
}
/*
It sends the exogenous environment's perceptions to the agent.
Arquivo: /opt/group.chon/simulide/examples/Chon/Blink/Blink.ino
Found Arduino Version 1
Arduino Compiler successfully loaded.

SimulIDE


1
SimulIDE-1.1.0-SR0 - Novo circuito
Abrir circuito
Look in:
/opt/group.chon/simulide/examples/Chon/LCDKeypadShield
Computer
chon
Name
Size
Type
Date Modified
LCDKeypadShield.sim1
4,77 KiB
sim1 File
29/09/2024 13:00
2
3
File name:
LCDKeypadShield.sim1
Open
Files of type:
Circuits (*.sim*)
Cancel

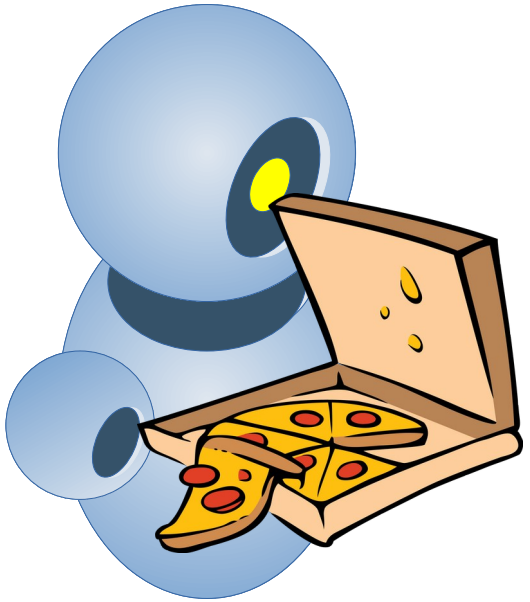

5
SimulIDE-1.1.0-SR0 - LCDKeypadShield.sim1*
Stopped
Tx
Rx
Open
/dev/ttyExogenou
4
100 Ω
s
l
u
d
r
t

SimulIDE

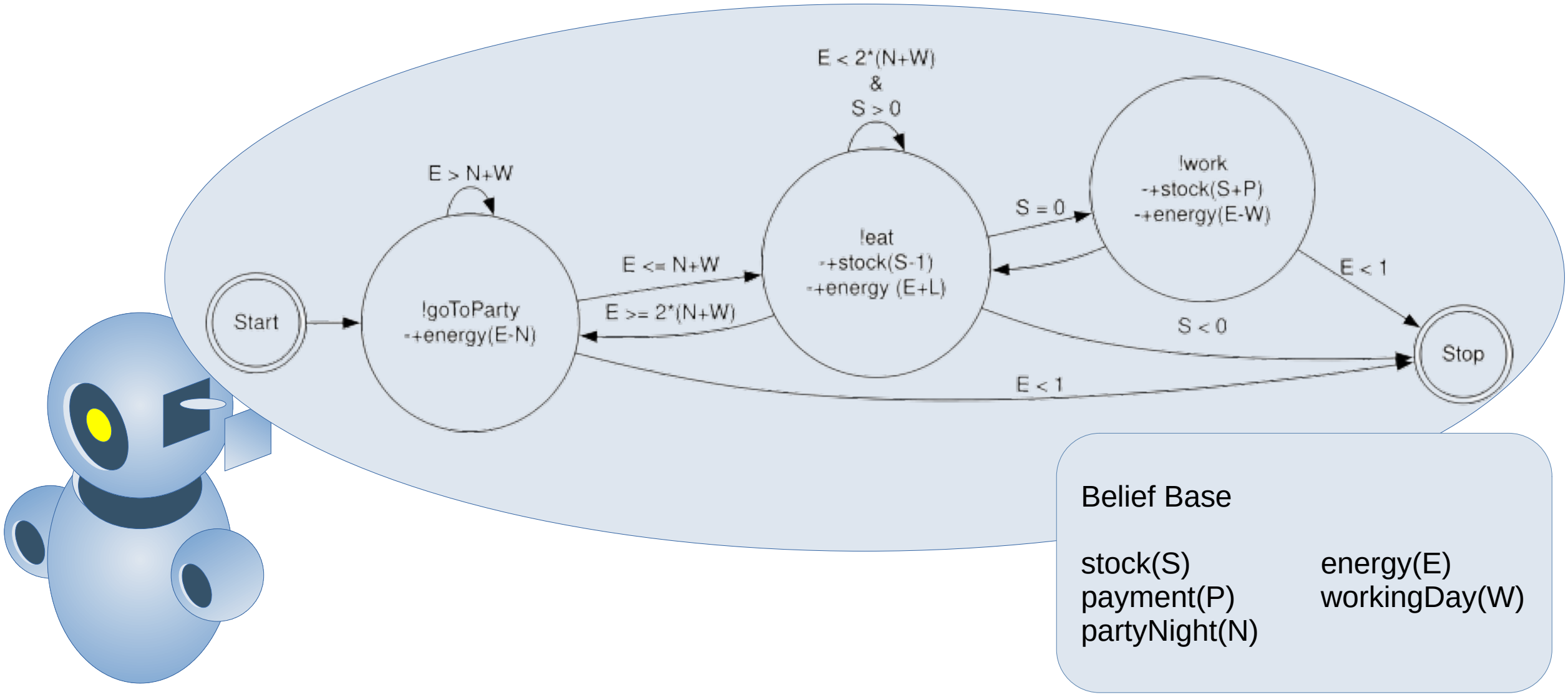



```
chon@chonide: ~
Arquivo  Editar  Abas  Ajuda
chon@chonide:~$ javino /dev/ttyEmulatedPort0
javino@ttyEmulatedPort0$ write yellowAlert
javino@ttyEmulatedPort0$ request getPercepts
temperature(25);rainLast24hrs(0);humidity(50);
javino@ttyEmulatedPort0$
```

MY FIRST SINGLE AGENT



My First Agent



Belief Base

stock(S)
energy(E)
payment(P)
workingDay(W)
partyNight(N)

My First Agent: Code

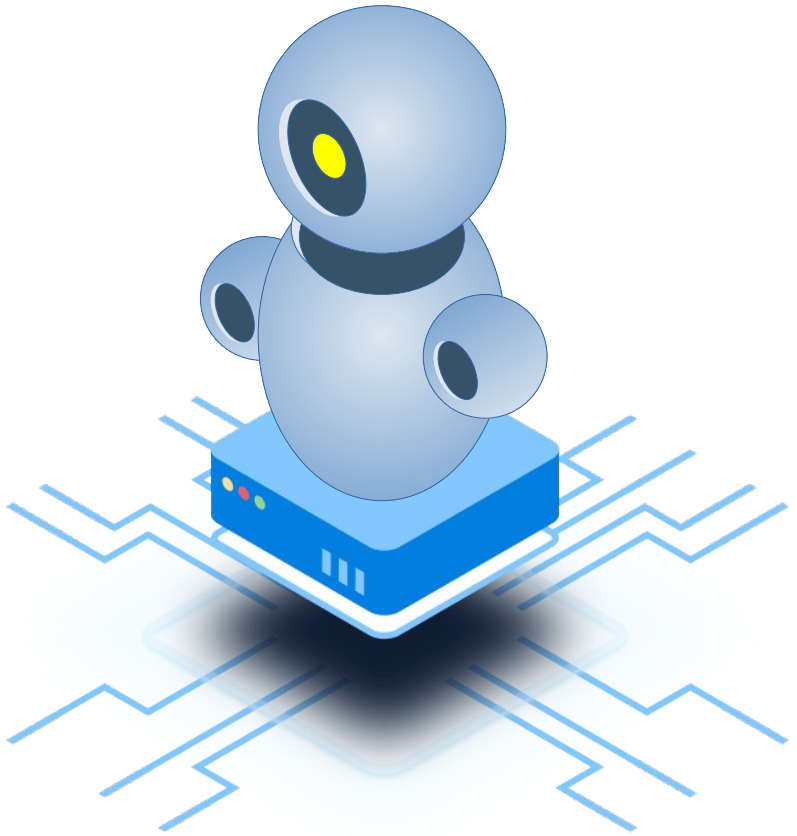
```
/* Initial beliefs and rules */
stock(0).       // This is a Belief that represents the pizza stock.
payment(+2).    // ... represents how many pizzas does Giacomo get by a working day?
energy(5).      // ... represents the Giacomo life energy.
workingDay(-1). // ... represents how much energy does Giacomo lose in a working day?
partyNight(-2). // ... represents how much energy does Giacomo lose on a party night?
lifeParameters(S,E,P,W,N) :- stock(S) &  energy(E) & payment(P) & workingDay(W) & partyNight(N). //this is a RULE!

/* Initial goals */
!startGiacomoLife.

/* Plans */
+!startGiacomoLife <- !startClock; !goToParty.
+!goToParty: lifeParameters(S,E,P,W,N) & (E+(N+W)>1) <- -+lastParty(0); -+energy(E+N); .wait(1000); !goToParty.
+!goToParty: lifeParameters(S,E,P,W,N) & (E+(N+W)<=1)<- !eat.
+!eat: lifeParameters(S,E,P,W,N) & (E<2*(-1*(N+W))) & S>0  <- -+stock(S-1); -+energy(E+1); .wait(1000); !eat.
+!eat: lifeParameters(S,E,P,W,N) & (E>=2*(-1*(N+W))) <- !goToParty.
+!eat: lifeParameters(S,E,P,W,N) & S=0  <- !work.
+!work: lifeParameters(S,E,P,W,N) <- -+stock(S+P); -+energy(E+W); !eat.
+!startClock <- +lastParty(0); !!tickingClock.
+!tickingClock <- .wait(100); ?lastParty(Time); -+lastParty(Time+100); !tickingClock.

+energy(E): E < 1 <- .print("Giacomo died of hunger!"); .stopMAS.
+stock(S): S < 0 <- .print("Without food!"); .stopMAS.
+lastParty(Time): Time > 10000 <- .print("Giacomo died of sadness!"); .stopMAS.
```

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/course/06-MyFirstAgent/giacomoAgent.chon

My First Agent: Code

initial beliefs -->	energy(5). day(sunny). myPhone(“555-111111”).	belief plan -->	+day(D) <- .print(“Today is ”,D).
a rule -->	happy :- day(D) & D=sunny & money(M) & M=veryMuch.	belief plan --> mental notation search in mind (?) new goal	+day(D): D=saturday & time > 18 <- +hungry; ?pizzaPhone(Number); !callTo(Number).
initial goals -->	!start. !callTo(“555-123321”).	contantion plan ->	-!start <- .print(“without belief myPhone(T) in mind”).
achievement plan -->	+!start <- .print(“hello world”).	JASON stdlib → https://jason-lang.github.io/api/jason/stdlib/package-summary.html	
achievement plan -->	+!start: myPhone(T) <- .print(“Call me:”,T).		

EMBEDDED MAS



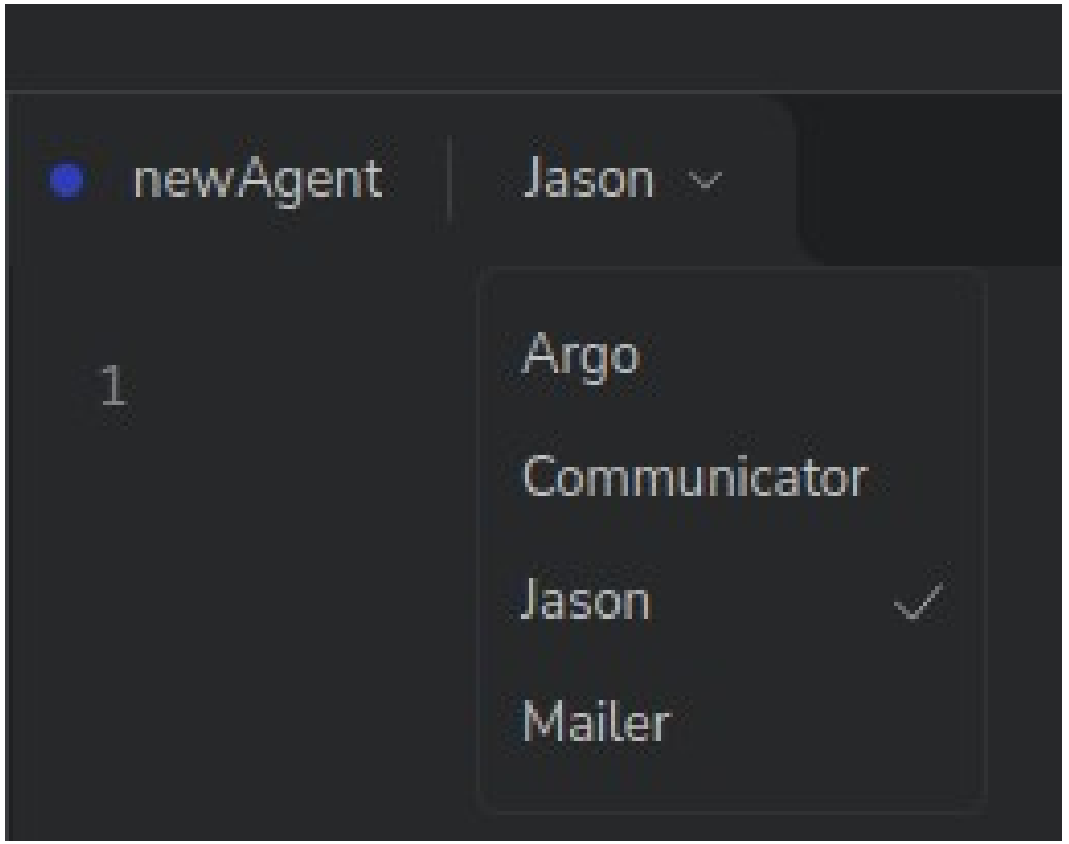
Jason Embedded

Jason Embedded is a spin-off Version of Jason for IoT and Embedded MAS:

- **Autonomy** as a characteristic an Embedded MAS provides to a device that does not need external architectures to control and manage it.
- **Communicability** is the agent's ability to communicate with other agents in its MAS or another MAS.
- **Mobility** as the agents' ability to move from one MAS to another.
- **Fault Tolerance and Adaptability** are defined as the ability of an agent to identify if a physical resource is absent or not answering commands and to overcome this situation by modifying its goals.
- **Distributed Ledger Technologies** technologies to facilitate the agreement between agents.

Jason Embedded

- **Jason Embedded** also provides **new types of agents** to control hardware devices and to communicate with agents hosted in other MAS.



Jason Embedded

- **Jason Embedded** also provides **new types of agents** to control hardware devices and to communicate with agents hosted in other MAS.

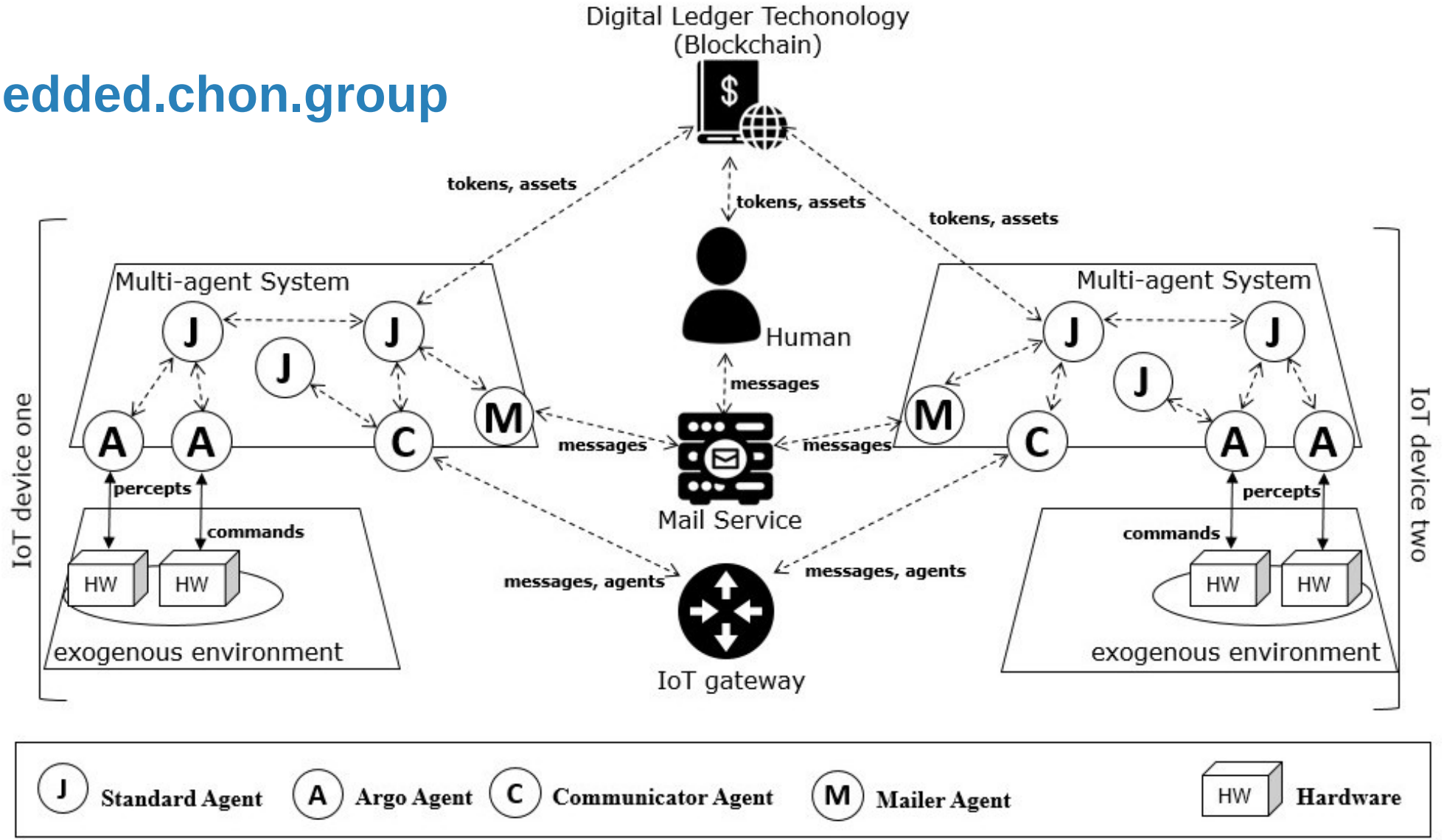
  - **ARGO**: is a customized agent architecture built on the Jason framework that extends Jason's standard agents by adding the **ability to control microcontrollers** [Pantoja et al., 2016].
  - **Hermes (Communicator)** : is another customized agent architecture built on Jason's Standard agent by adding the **ability to communicate** with agents from **other MAS**, which also have Communicator agents [Pantoja et al., 2018].
  - **Mailer** : agent architecture based on the Jason framework to enable secure and simplified **human-agent interaction** [Lazarin et al., 2024].

Jason Embedded

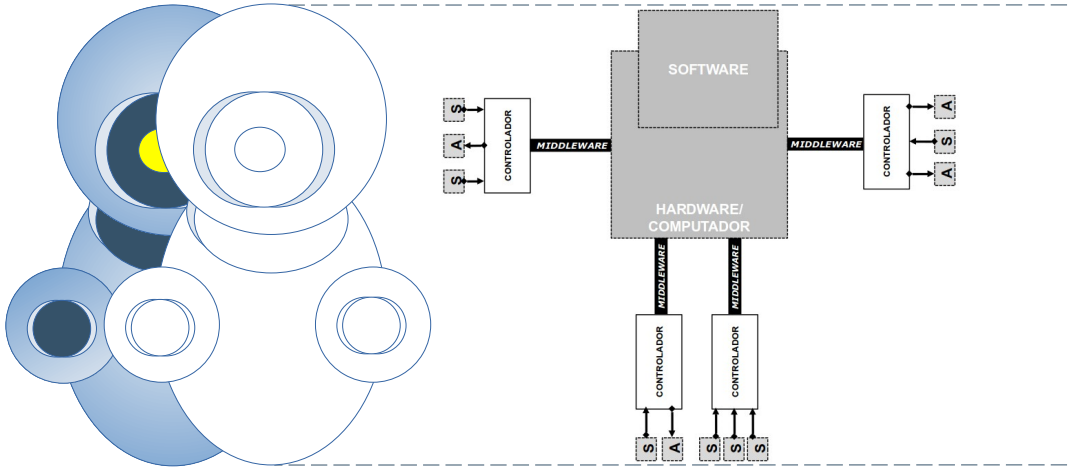
- **Mobility** is based on **bio-inspired protocols** [Jesus et al., 2021].
  - The protocols simulate natural behaviors that could be explored in collaborative tasks using IoT devices.
  - One or more agents or even the whole MAS could move from one device to another to take control of the destination device (**Predation**) or to use it as a temporary non-hazardous relationship (**Mutualism** and **Inquilinism**) if both parties accept the protocol.
- **Middleware** that allows cognitive agents to manipulate Tokens and NFT in blockchain network [Lazarin et al., 2023].

Jason Embedded

https://jasonembedded.chon.group



ARGO AGENT

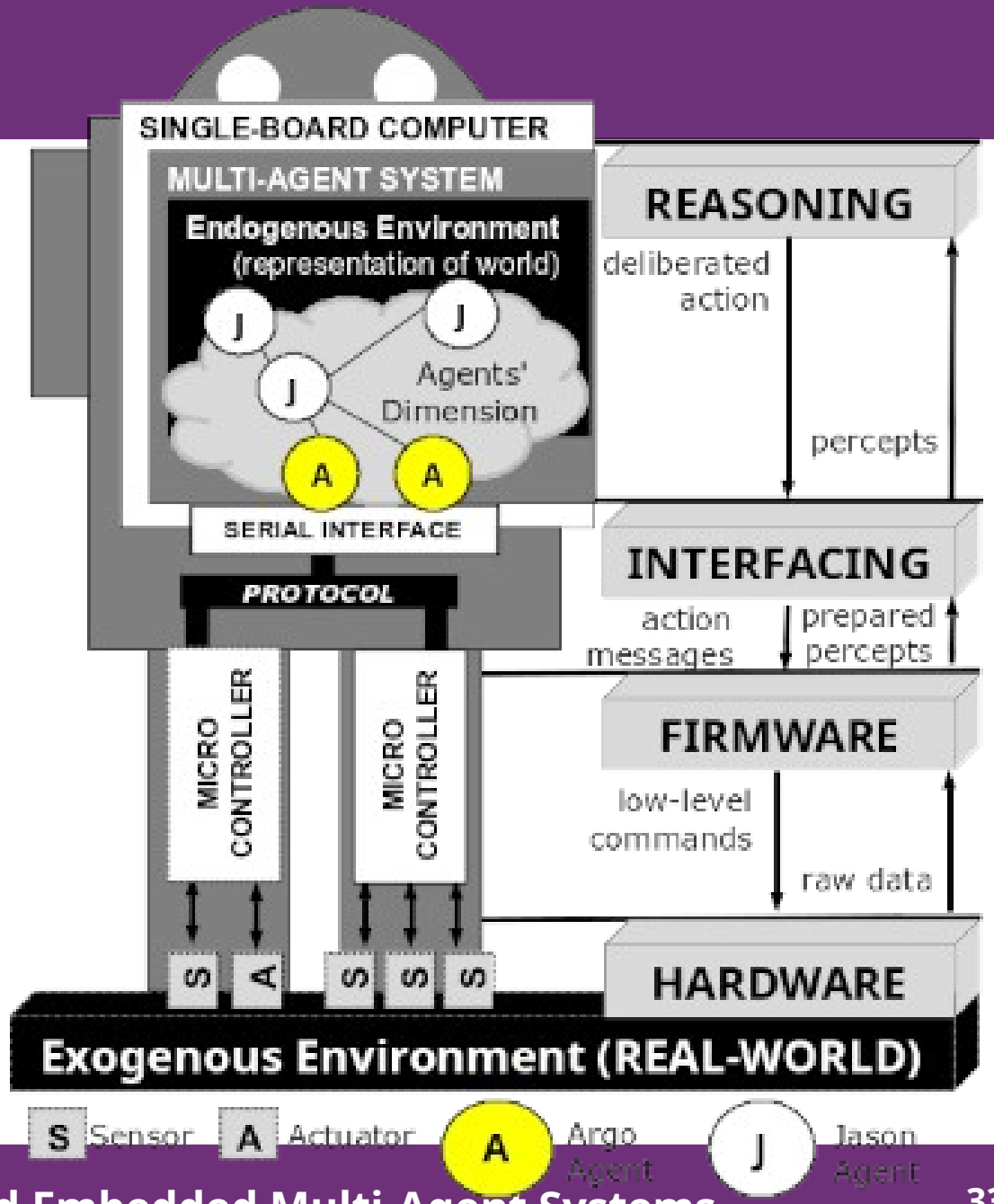


https://argo.chon.group

Argo for Jason

ARGO is a customized agent architecture that **interfaces** sensors and hardware actuators.

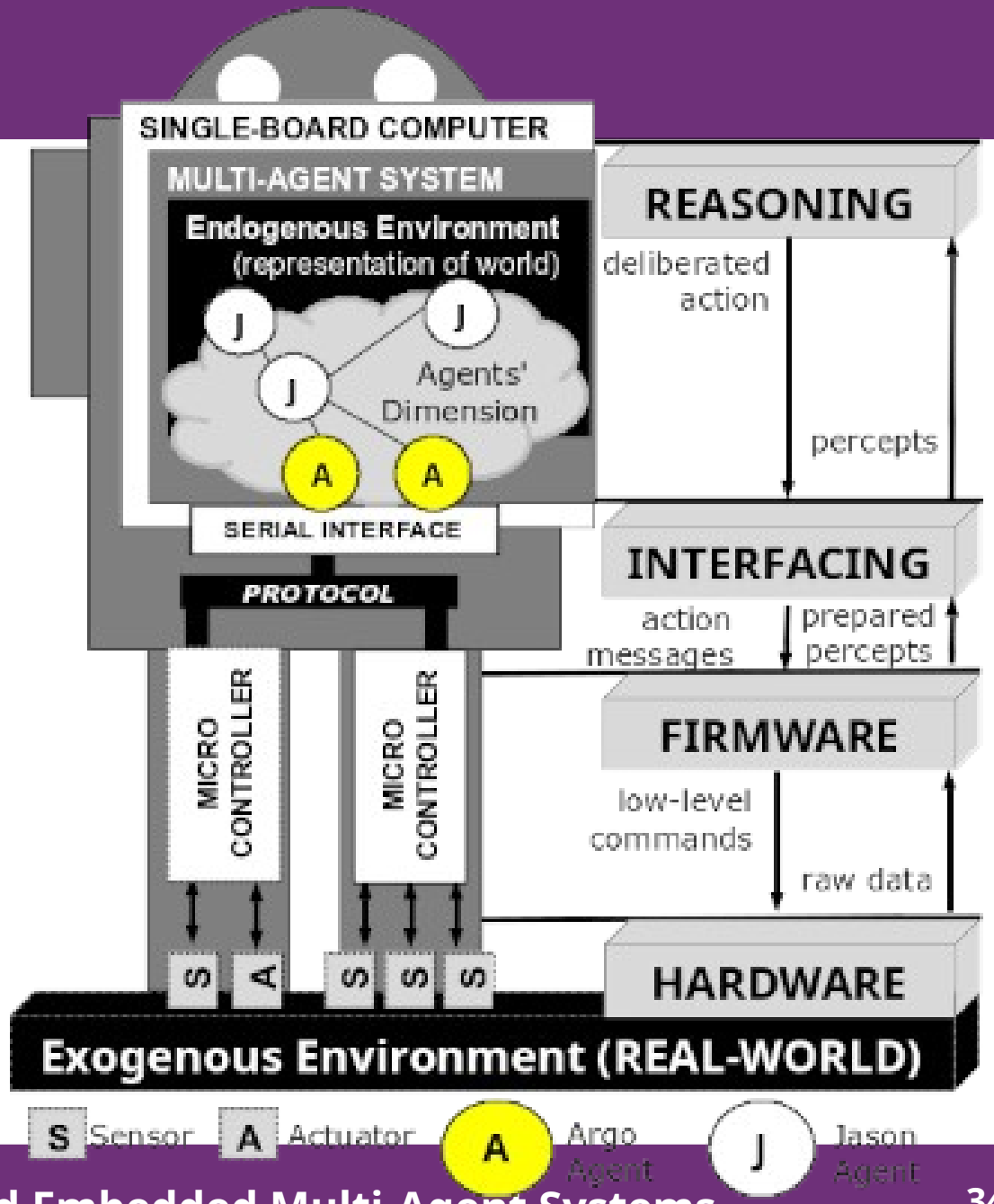
ARGO aims to be a practical architecture for **programming embedded BDI agents** using **Jason** and microcontroller boards.



Argo for Jason

The **ARGO** allows to:

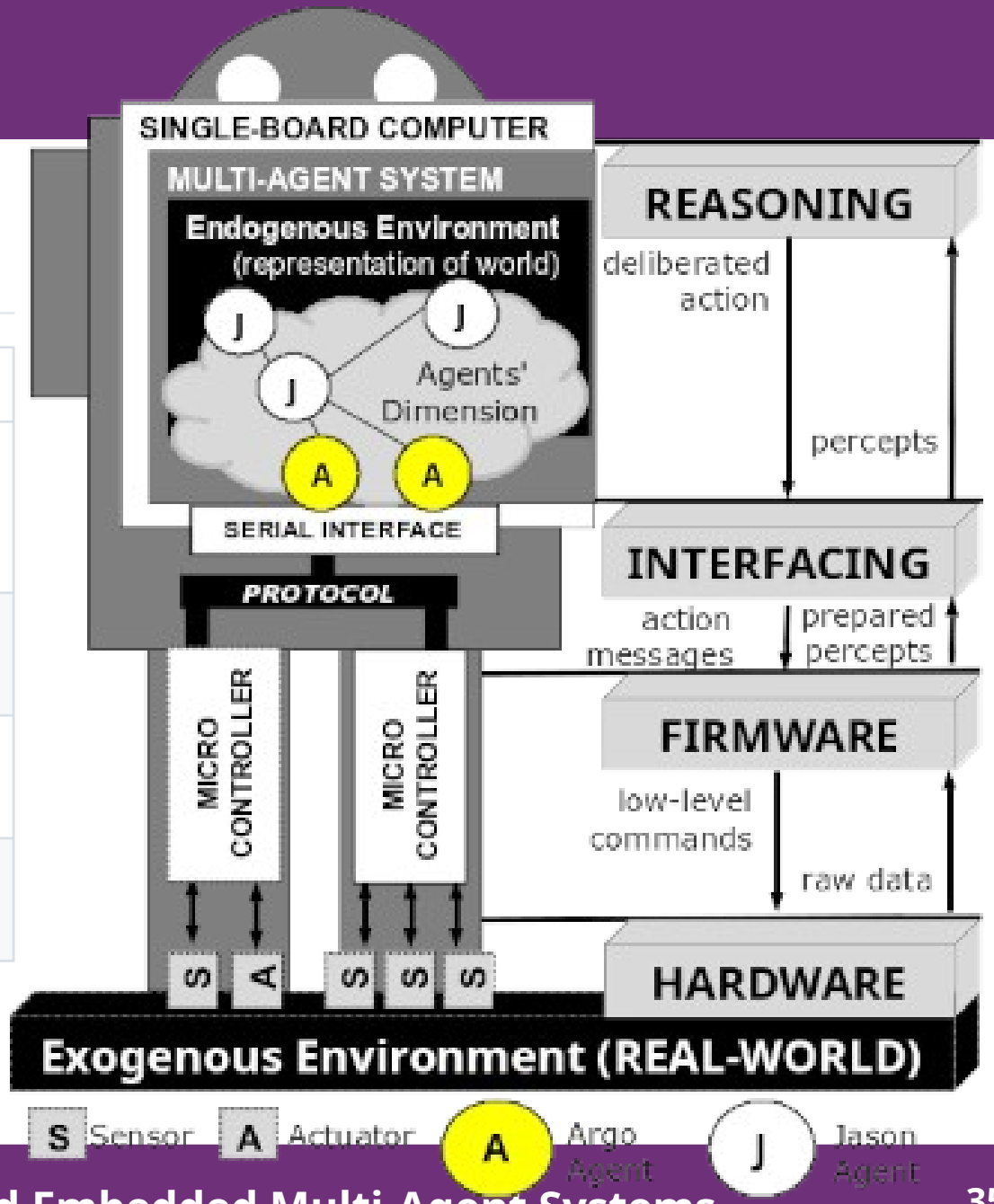
1. to direct control actuators at runtime;
2. to receive sensor perceptions within a predefined period automatically;
3. to change which devices are being accessed at runtime;
4. to communicate with other agents in Jason;
5. whether or not to perceive the real world at runtime;
6. to change perception filters at runtime.



Argo for Jason

ARGO Internal Actions

Action	Description
.argo.port(*P*)	defines which serial port should be used by the agent, where *P* is a literal representing the port identification(e.g., COM8, ttyUSB0).
.argo.limit(*M*)	defines the sensing interval, where *M* is a value in milliseconds.
.argo.percepts(*open*|*close*)	decides whether or not to perceive the exogenous environment (physical world).
.argo.act(*A*)	sends to the hardware an action, represented by literal *A*, to be executed by a microcontroller;



Example: blink

chon-group / Argo

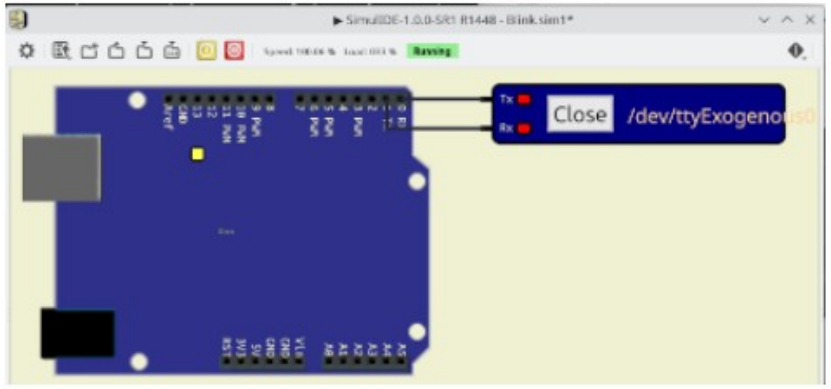
Type / to search

Code | Issues 1 | Pull requests | Wiki | Security | Insights | Settings

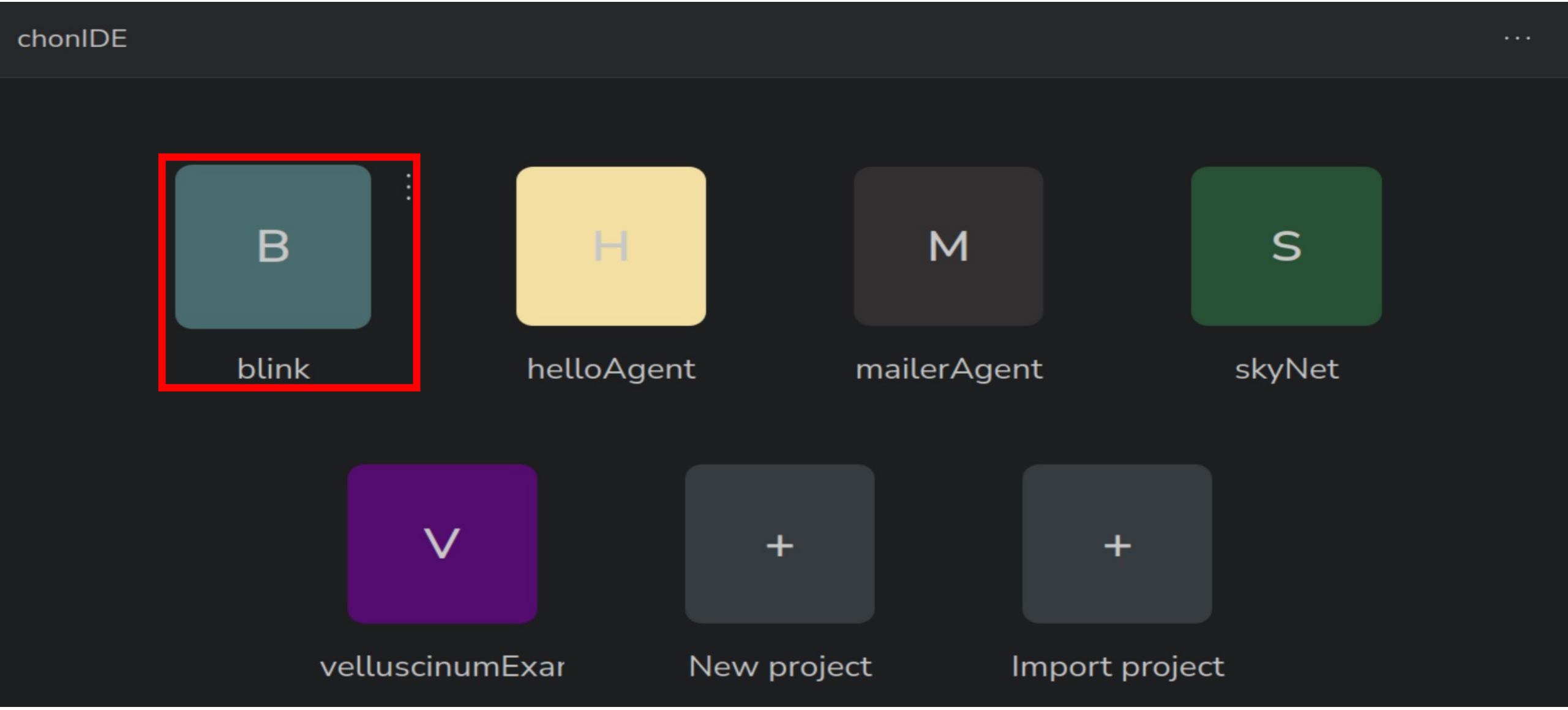
Tools

Nilson Lazarin edited this page 19 hours ago · 1 revision

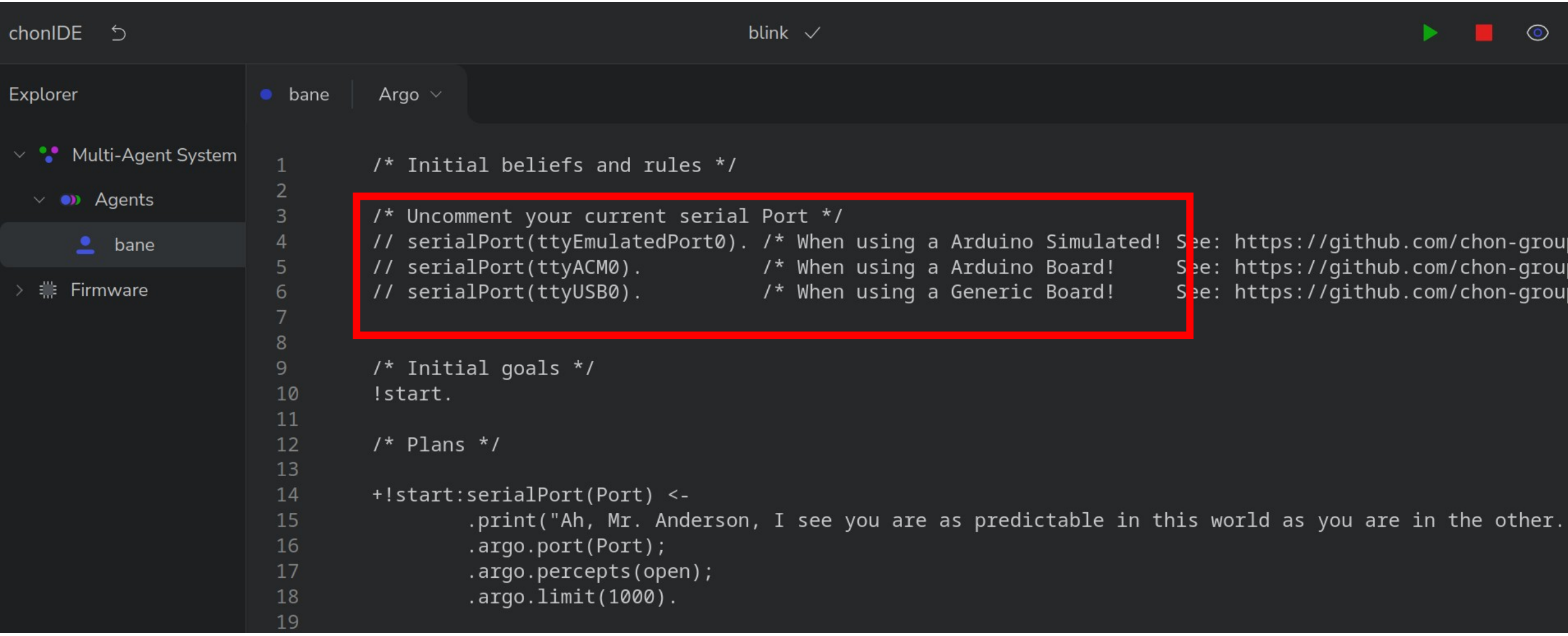
Examples | Tools | Wiki

Arduino	SimulIDE
	
Blink tutorial with Arduino Board	Blink tutorial with SimulIDE

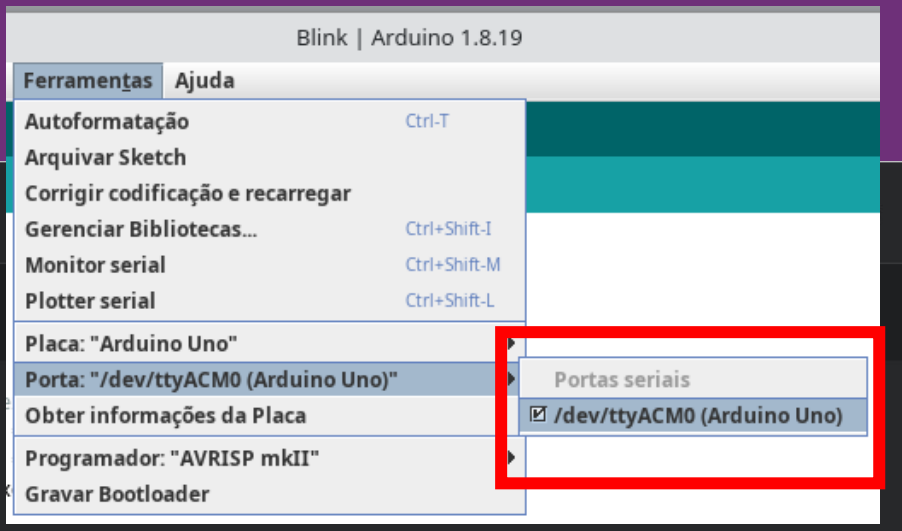
Example: blink



Example: blink



Example: blink



chonIDE blink

Explorer

- Multi-Agent System
  - Agents
    - bane
  - Firmware

bane | Argo

```
1       /* Initial beliefs and rules */
2
3       /* Uncomment your current serial Port */
4       // serialPort(ttyEmulatedPort0). /* When using a Arduino Simulated! See: https://github.com/chon-grou
5       serialPort(ttyACM0).             /* When using a Arduino Board!     See: https://github.com/chon-grou
6       // serialPort(ttyUSB0).          /* When using a Generic Board!     See: https://github.com/chon-grou
7
```

Agent Tracer

```
[Javino] Using version 1.6.4
[bane] Ah, Mr. Anderson, I see you are as predictable in this world as you are in the other.
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
[bane] Turning ON  the Led in Arduino!
[bane] Turning OFF the Led in Arduino!
```

Exercise: LCD KeyPad

chon-group / Argo

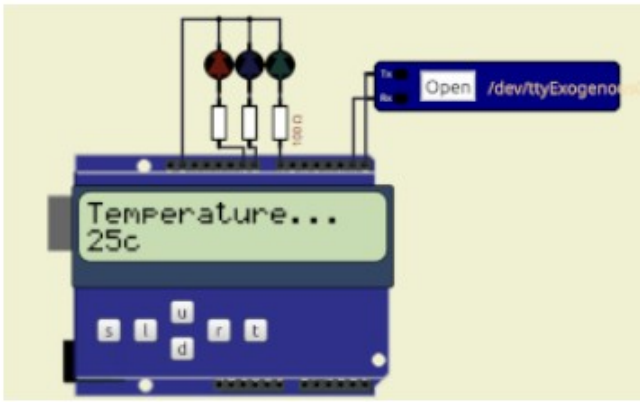
Type /

Code | Issues 1 | Pull requests | Wiki | Security | Insights | Settings

Arduino LCD Keypad Project

Nilson Lazarin edited this page 4 minutes ago · 6 revisions

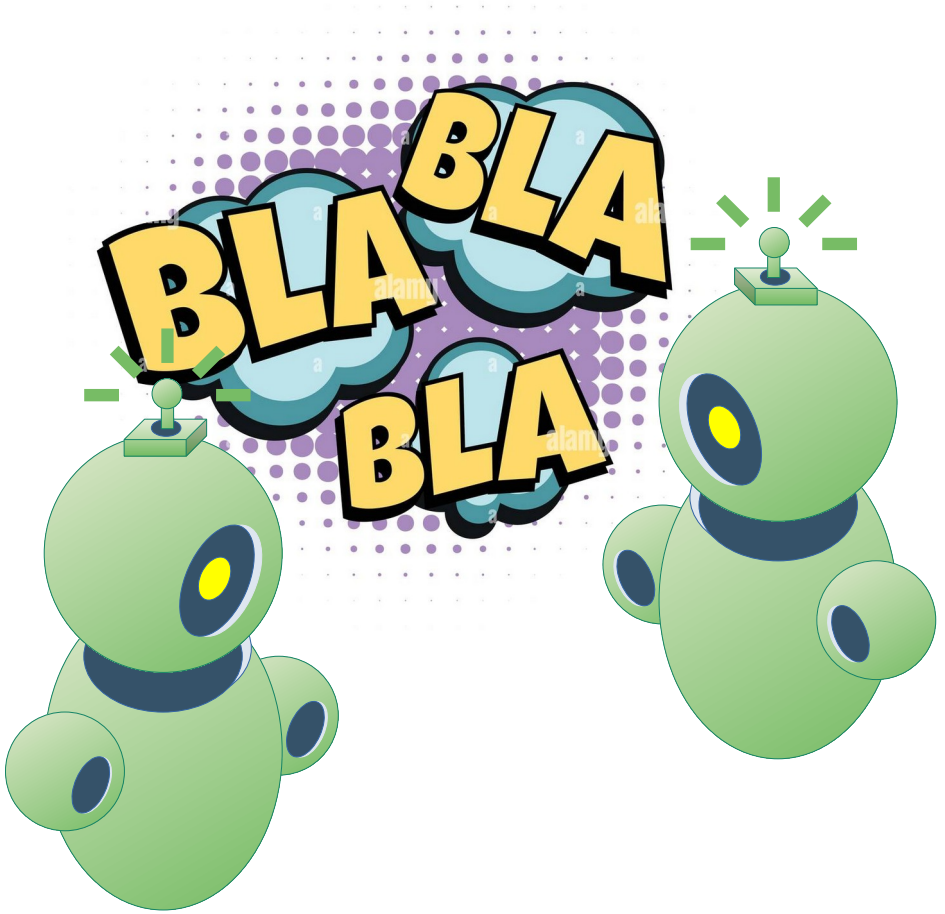
Home | Examples | Tools



Adjust the ARGO agent to:

- turn on the **green LED** if the rain intensity less than 30mm;
- turn on the **yellow LED** if the rain intensity exceeds 50mm;
- turn on the **red LED** if the the rain intensity exceeds 80mm;

COMMUNICABILITY



Jason Framework: Communication

Based on the theory of Speech Acts [1] and KQML [2] (Knowledge Query and Manipulation Language), a communication protocol for knowledge-based systems.

[1] M. Bierwisch, “Semantic Structure and Illocutionary Force”, em Speech Act Theory and Pragmatics, J. R. Searle, F. Kiefer, e M. Bierwisch, Orgs., Dordrecht: Springer Netherlands, 1980, p. 1–35. doi: 10.1007/978-94-009-8964-1_1.
[2] T. Finin, R. Fritzson, D. McKay, e R. McEntire, “KQML as an agent communication language”, em Proceedings of the third international conference on Information and knowledge management - CIKM ’94, Gaithersburg, Maryland, United States: ACM Press, 1994, p. 456–463. doi: 10.1145/191246.191322.

Jason Framework: Performatives

The performatives that are currently available for agent communication in Jason are largely inspired by KQML. We also include some new performatives, related to plan exchange rather than communication about propositions. The available performatives are briefly described below, where s denotes the agent that sends the message, and r denotes the agent that receives the message [1].

PERFORMATIVE	DESCRIPTION
tell	s intends r to believe (that s believes) the sentence in the message's content to be true;
untell	s intends r not to believe (that s believes) the sentence in the message's content to be true;
achieve	s requests that r try to achieve a state of the world where the message content is true;
unachieve	s requests that r try to drop the intention of achieving a state of the world where the message content is true;
tellHow	s informs r of a plan;
untellHow	s requests that r disregard a certain plan (i.e., delete that plan from its plan library);
askIf	s wants to know if the content of the message is true for r;
askAll	s wants all of r's answers to a question;
askHow	s wants all of r's plans for a triggering event;

[1] R. H. Bordini e J. F. Hübner, "BDI Agent Programming in AgentSpeak Using Jason", em Computational Logic in Multi-Agent Systems, F. Toni e P. Torroni, Orgs., em Lecture Notes in Computer Science, vol. 3900. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, p. 143–164. doi: 10.1007/11750734_9.

Jason Framework: Message Structure

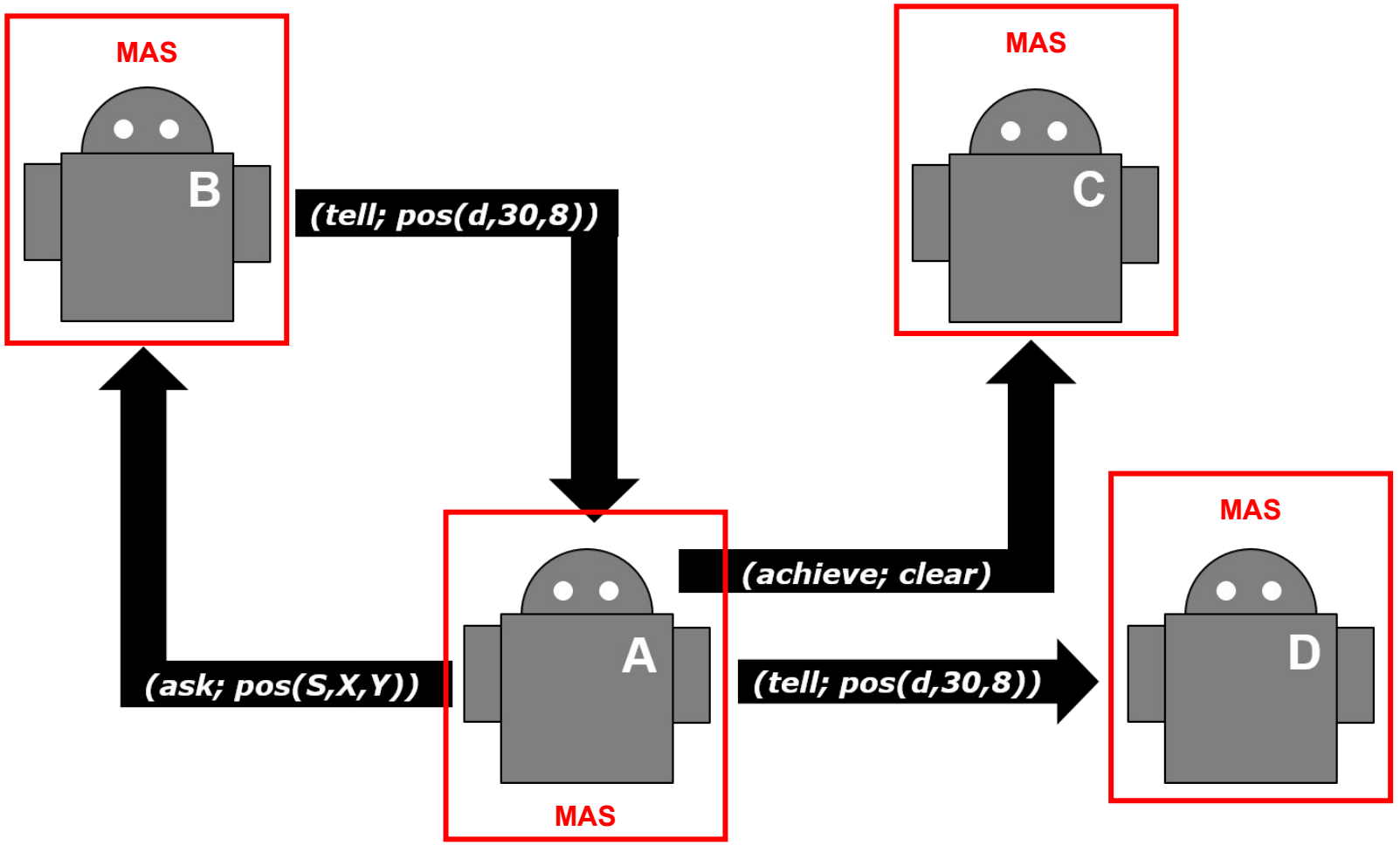
.send(receiver, ilf, message, answer, timeout)
.broadcast(ilf, message)

- **receiver**
  Name of the agent receiving the message (or list of recipients)
- **ilf**
  Illocutionary force of the speech act (KQML)
- **message**
  Message content
- **asnwer**
  A term that will store the response (optional field)
- **timeout**
  Timeout in milliseconds to receive a response (optional field)

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/SEMESO/2024/projects/chiefGlutton.chon

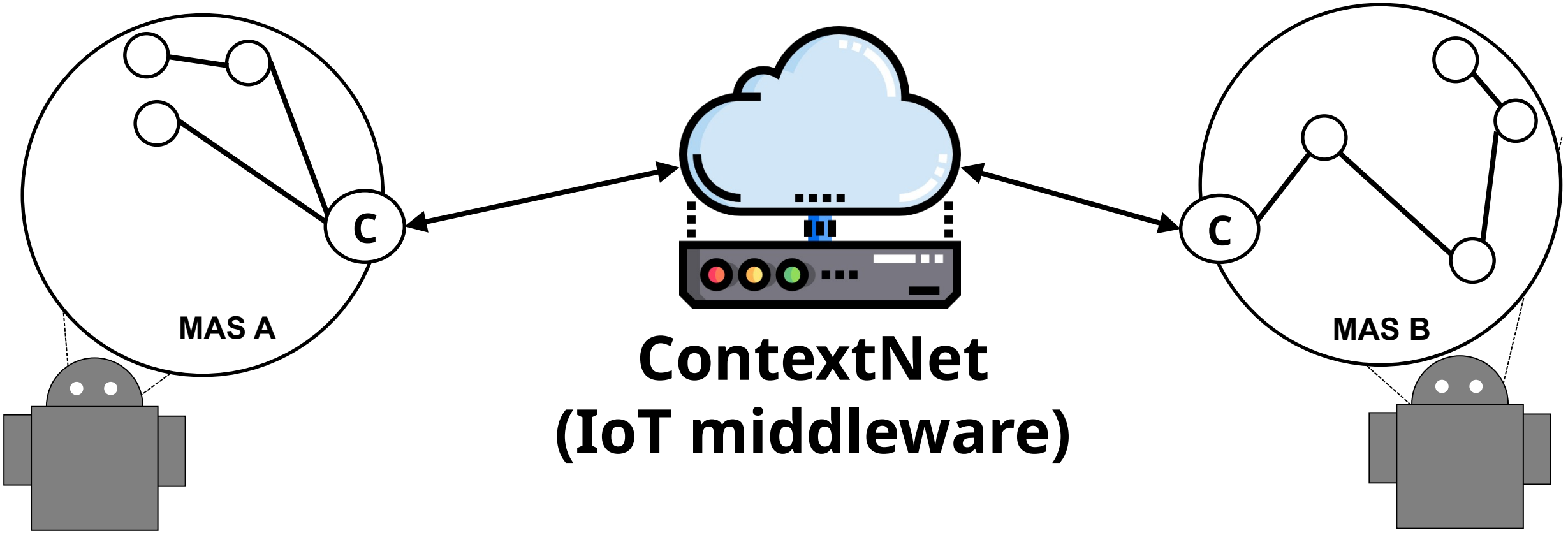
Image credits:
https://smurfs.fandom.com/wiki/Chef_Smurf
https://smurfs.fandom.com/pt/wiki/Fominha

Collaborative MultiAgent System



Hermes Agents: Server Communication

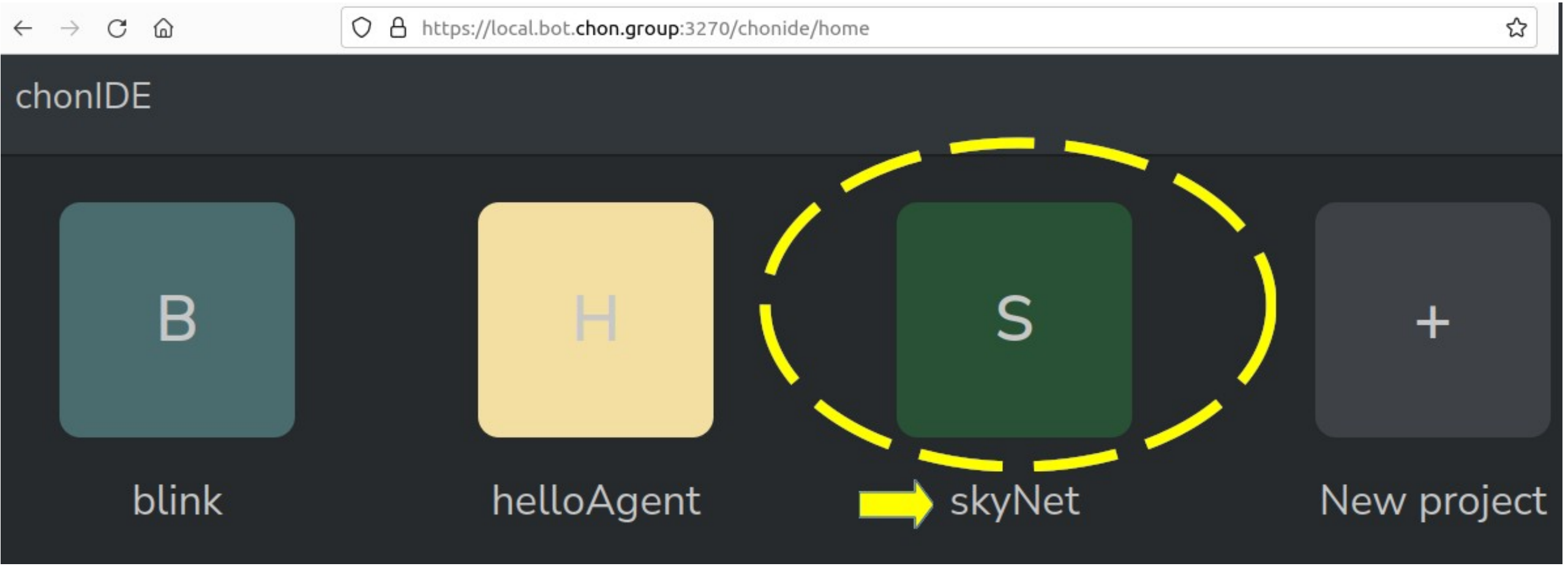
Public Server: skynet.chon.group - UDP Port: 5500



Hermes Agents: Internal Actions

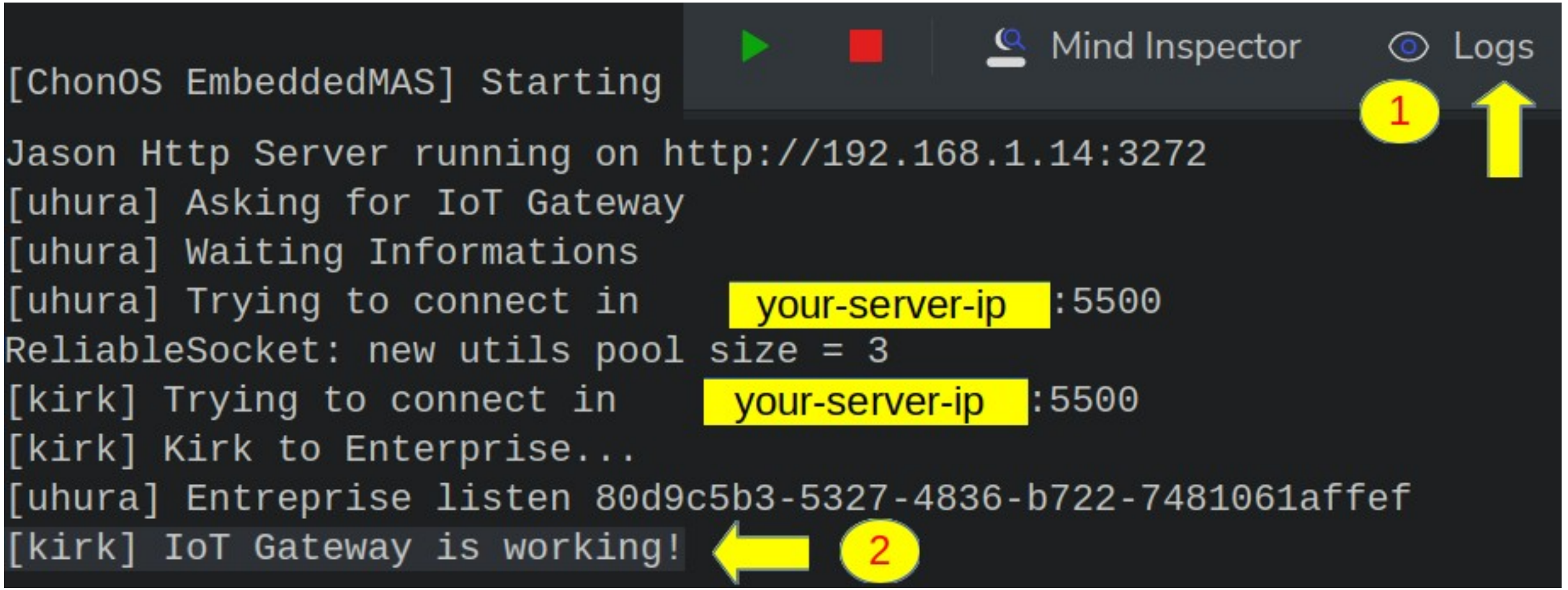
Hermes	.hermes.configureContextNetConnection("***X***",***G***,***E***,***U***)	Configures an ContextNet network. Where ***X*** is a string that represents an network name; ***G*** is a literal that represents the FQDN or the network address of an gateway; ***E*** is a number that represents the network port of an gateway; ***U*** is a literal that represents the identification of the device in the network.
	.hermes.connect("***X***")	Joins at "***X***" ContextNet network.
	.hermes.sendOut(***D***,***f***,***M***)	Dispatches a message to another MAS. Where ***D*** is a literal that represents the identification of the recipient MAS; ***f*** is a illocutionary force (*tell* | *untell* | *achieve* | *unachieve*); ***M*** is a literal that represents the message.
	.hermes.moveOut(***D***,***b***,***A***)	Carries over the agents to other MAS. Where ***D*** is a literal that represents the identification of the recipient MAS; ***b*** is a bio-inspired protocol (*inquilinism* | *mutualism* | *predation*); ***A*** is one, all, or a set of agents (i.e., all, agent or [$agent_1$, $agent_2$, $agent_n$])
	.hermes.disconnect("***X***");	Disconnects at "***X***" ContextNet network.

Hello SkyNet



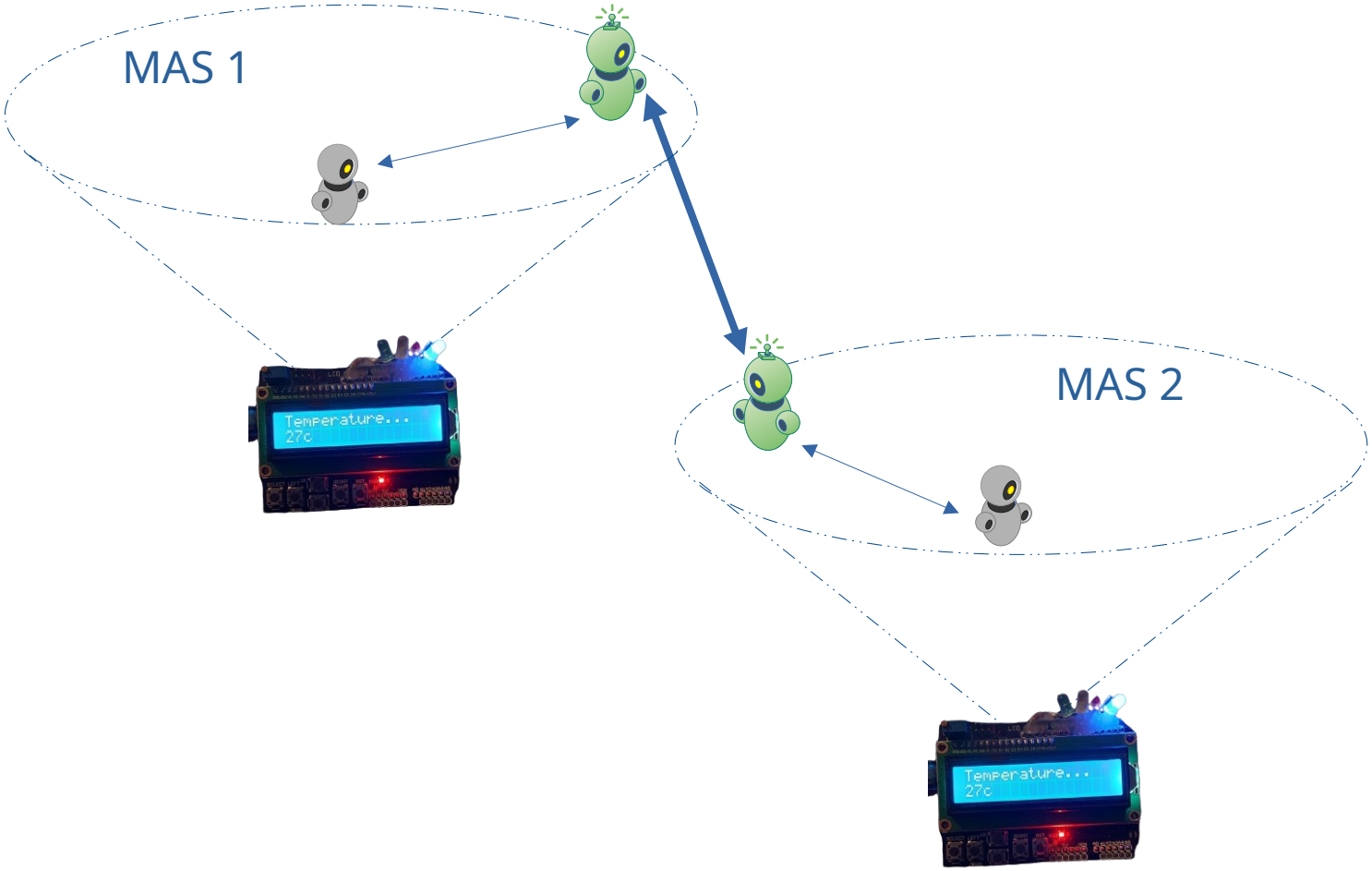
Hello SkyNet

Hello SkyNet



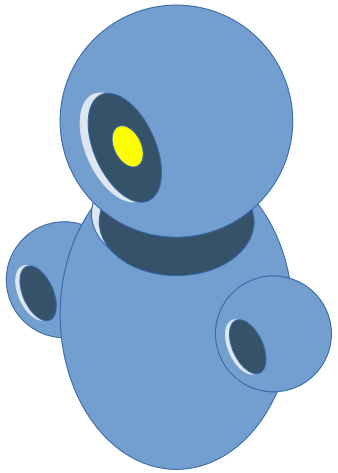
Example: MAS Communication

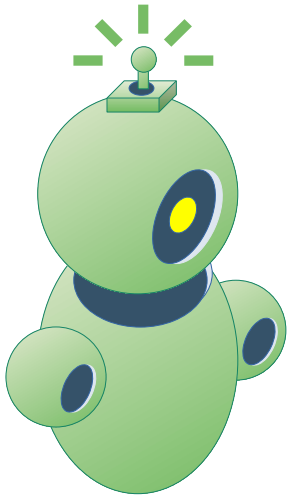
https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/SEMESO/2024/projects/hermesAGENT.chon



Example: MAS 1

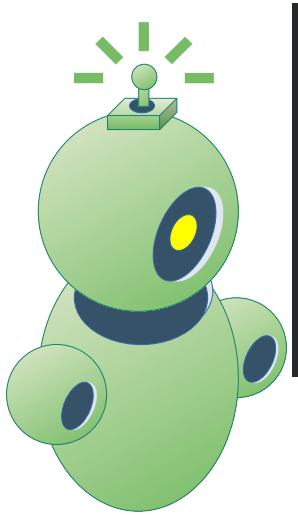
```
+rainLast24hrs(R): R > 50 <-
        .print("New perception-> rainLevel: ",R," mm");
        .concat("Atention rain level is ",R," mm", RainLevelMessage);
        .send(agentHERMES,achieve,forward(RainLevelMessage)).
```





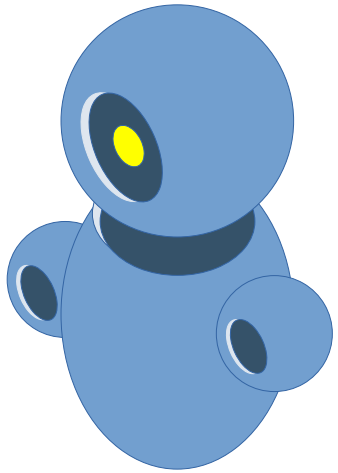
```
+!forward(Message) <-
        ?neighborUUID(N);
        .print("Forwarding a message to ", N);
        .sendOut(N, achieve, showMessage(Message)).
```

Example: MAS 2



```
+!showMessage(Message)[source(Device)] <-
        .print("I received the following message: ", Message);
        .send(agentARGO, achieve, infoLCD(Message)).
```

```
+!infoLCD(Message) <-
        .random(R); .wait(2000*R);
        .argo.act(Message).
```



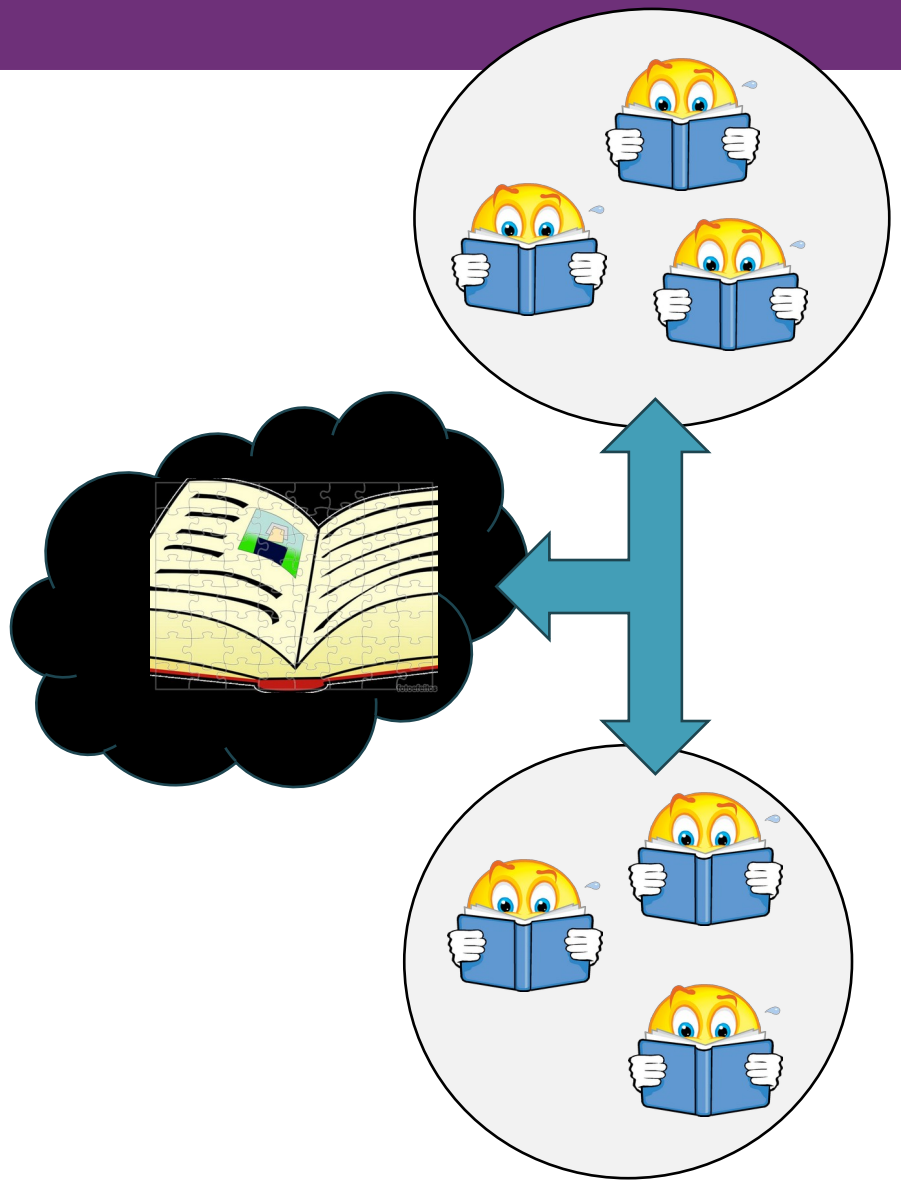
Simple Exercise

Adjust to:

- turn on the **green LED** in **MAS 2** if the rain intensity less than 30mm in **MAS 1**;
- turn on the **yellow LED** in **MAS 2** if the rain intensity exceeds 50mm in **MAS 1**;
- turn on the **red LED** in **MAS 2** if the the rain intensity exceeds 80mm in **MAS 1;**

VELLUSCINUM

https://github.com/chon-group/Velluscinum/wiki



MAS + DLT

- Can facilitate the agreement between agents

  Taking what is registered in the Ledger as accurate;

- Can be helpful to:

  Manage trust relationships;

  Open MAS;

  Distributed scenarios.

Objective

Using digital assets in the relationships between cognitive agents:

the transfer of funds;

registration of ownership of artifacts;

declaration of promises or agreements.

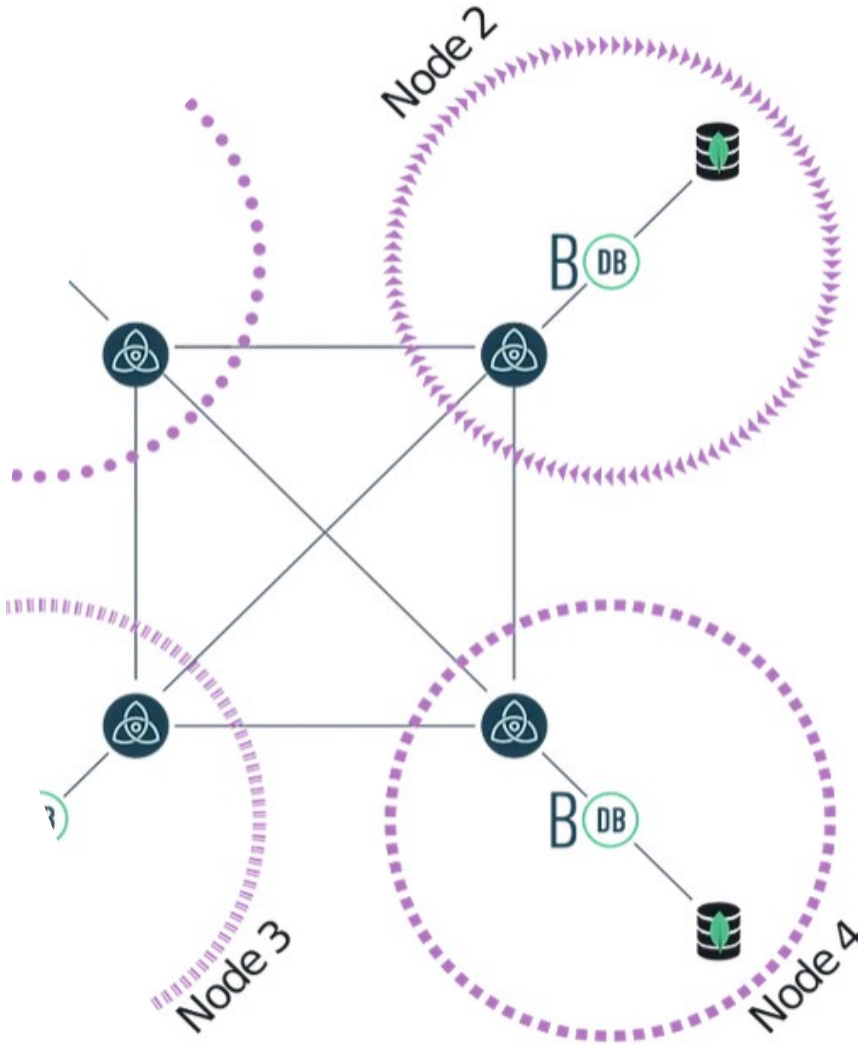
Allowing the agents to manipulate assets and wallets directly in the DLT.

Theoretical foundation

- A DLT can be considered an **append-only decentralized database** because it provides a storage mechanism.

However, compared with a traditional database, its **performance is much lower** because it has a **low download rate** and a **high latency** [13].

New approaches **based on distributed databases** have been used to improve the performance of permissioned DLT.



13. McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., Mc-Mullen, G., Henderson, R., Bellemare, S., Granzotto, A.: Bigchaindb: a scalable blockchain database. white paper, BigChainDB (2016)

Theoretical foundation

- When considering the number of transactions that a DLT can successfully execute per second:

  DLT permissionless (e.g., Bitcoin or Ethereum)

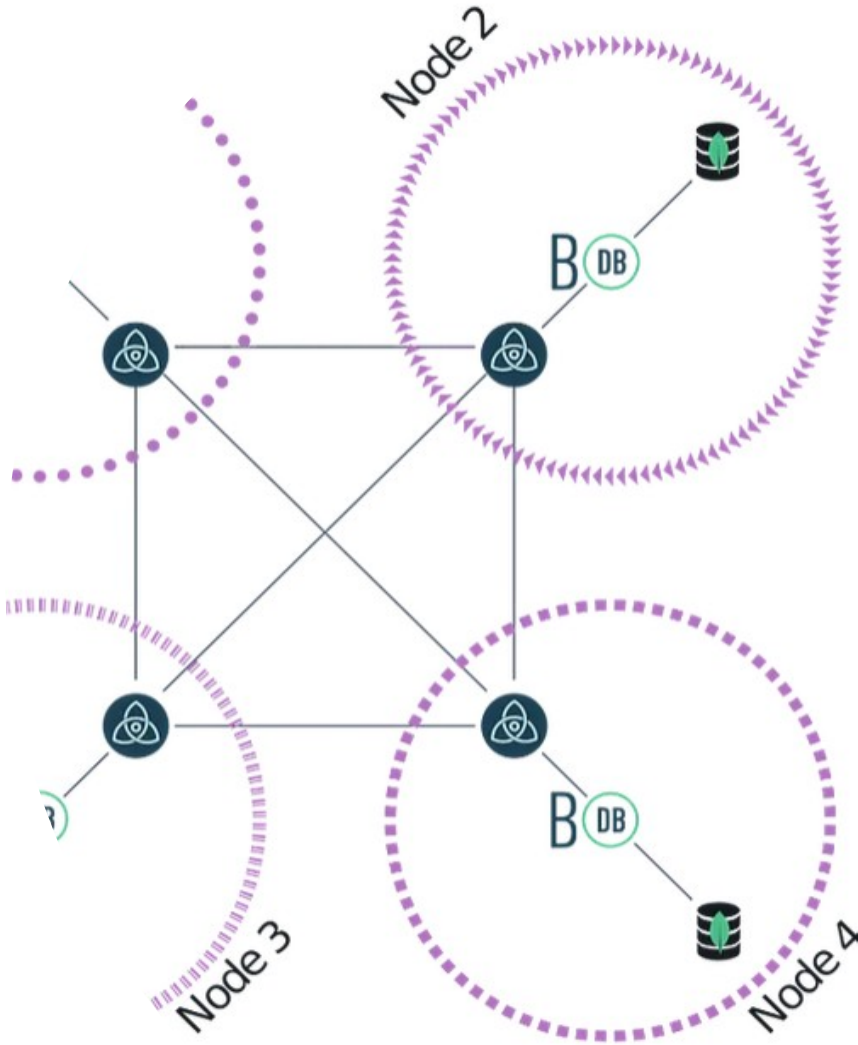
  - *ranges between 3.45 and 4.69 TPS [24,15];*

  DLT permissioned (e.g., HyperLedger Fabric)

  - *vary between 4.28 and 10.51 TPS [7, 15];*

  DLT permissioned distribution database-based (e.g., BigchainDB)

  - *varies between 50.60 and 175 TPS [7, 10].*



7. Contreras, A.F.: Benchmarking of blockchain technologies used in a decentralized data marketplace. Grado en Ingeniería Informática, E.T.S de Ingenieros Informáticos (UPM), Madrid, España (Jun 2019), https://oa.upm.es/55775/

10. Ge, Z., Loghin, D., Ooi, B.C., Ruan, P., Wang, T.: Hybrid blockchain database systems: Design and performance. Proc. VLDB Endow. 15(5), 1092–1104 (may 2022). https://doi.org/10.14778/3510397.3510406

15. Nasir, Q., Qasse, I.A., Abu Talib, M., Nassif, A.B.: Performance Analysis of Hyperledger Fabric Platforms. Security and Communication Networks 2018, 1–14 (2018). https://doi.org/10.1155/2018/3976093

24. Tikhomirov, S.: Ethereum: State of knowledge and research perspectives. In: Imine, A., Fernandez, J.M., Marion, J.Y., Logrippo, L., Garcia-Alfaro, J. (eds.) Foundations and Practice of Security. pp. 206–221. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-75650-9_14

Proposed Approach

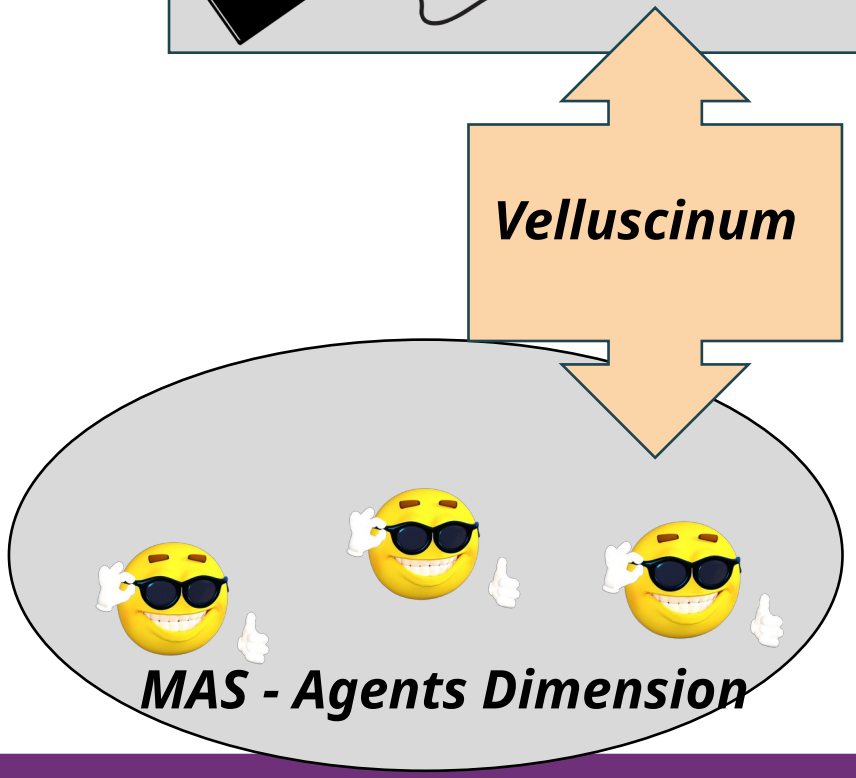
Use of digital assets to supporting the BDI agents' relationship:

Indivisible assets

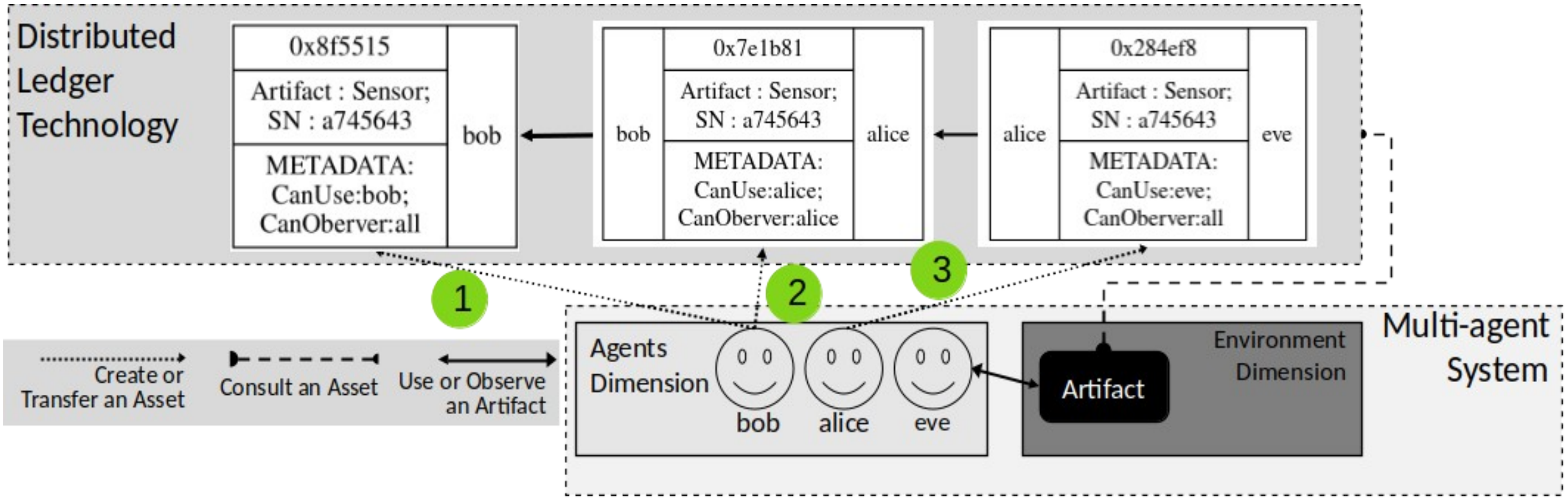
- as property record
- as promise or agreement

Divisible assets

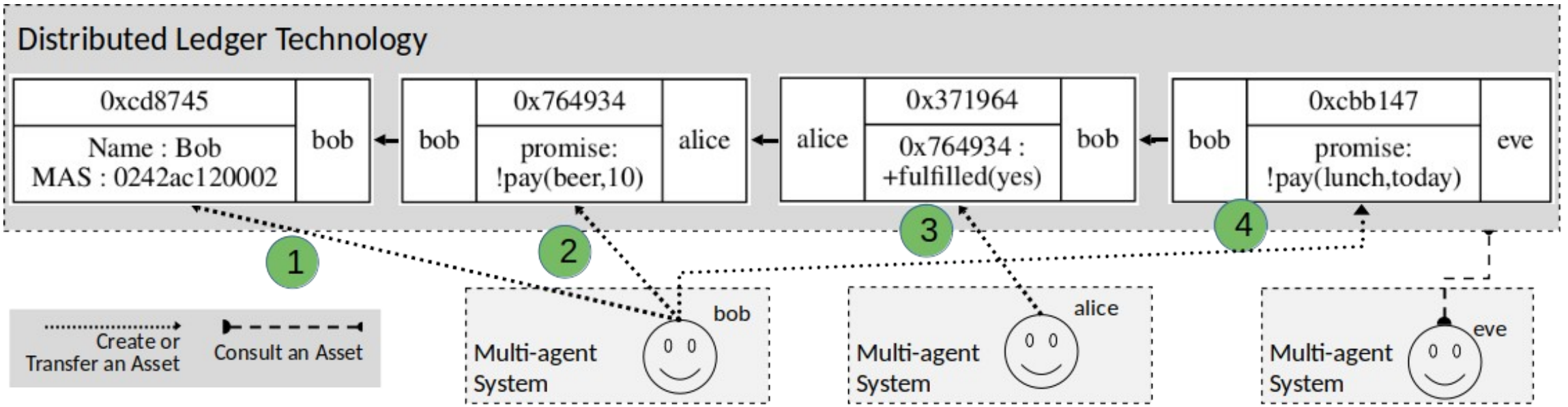
- as cryptocurrencies



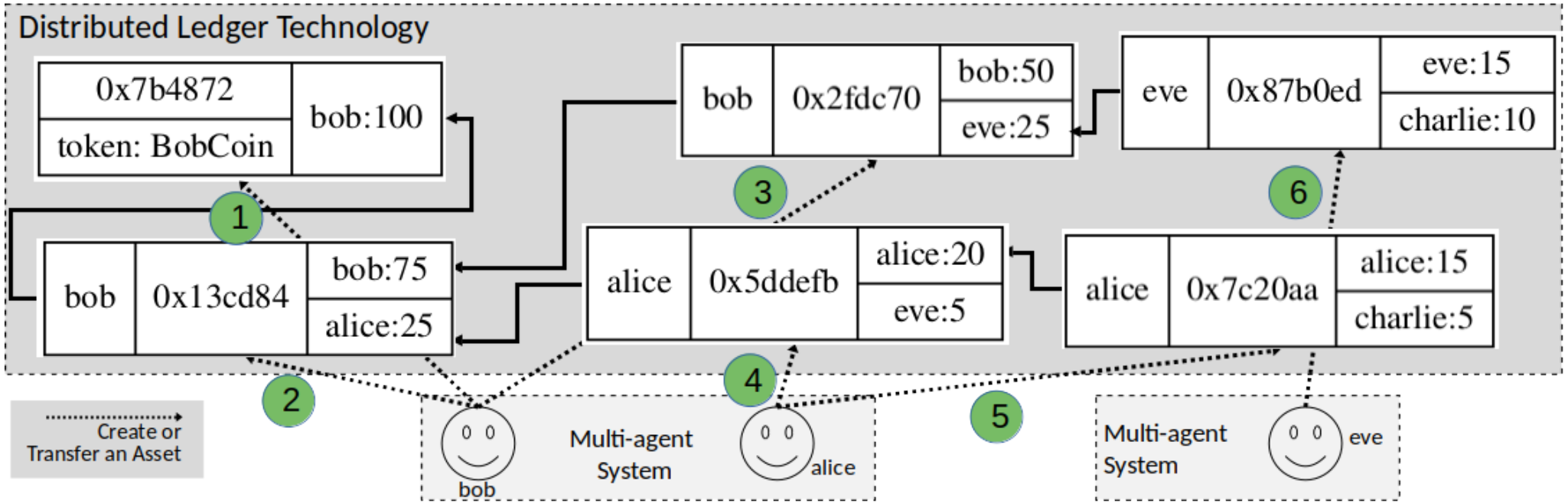
Indivisible asset such as property record



Indivisible asset such as promise or agreement



Divisible asset such as cryptocurrency



https://github.com/chon-group/Velluscinum/wiki/

The Internal Actions provided by the Velluscinum

Internal Action	Description
buildWallet	Generates a digital wallet and returns a belief.
loadWallet	Loads (or creates if not exists) the agent's key pair into the project folder, that represents a digital wallet, and returns a belief.
storeWallet	Persists a belief that represents the agent's digital wallet as a key pair into the project folder.
walletContent	Retrieves the content of a Wallet and return a belief with a list of assets.
tokenBalance	Check *units* of a specific token into a wallet and return a belief.
tokenInfo	Retrieves the content of a Token and return a belief with a list inside.
deployNFT	Registers an asset and returns a belief.
transferNFT	Transfers a Non-Fungible-Token and returns a belief.
deployToken	Creates *n* units from an asset and returns a belief.
transferToken	Transfer V units of C and returns +b(T);
stampTransaction	Stamps a transaction.

https://github.com/chon-group/distributedAndEmbeddedAI/blob/main/workshops/SEMESO/2024/projects/chiefGlutonVellus.chon

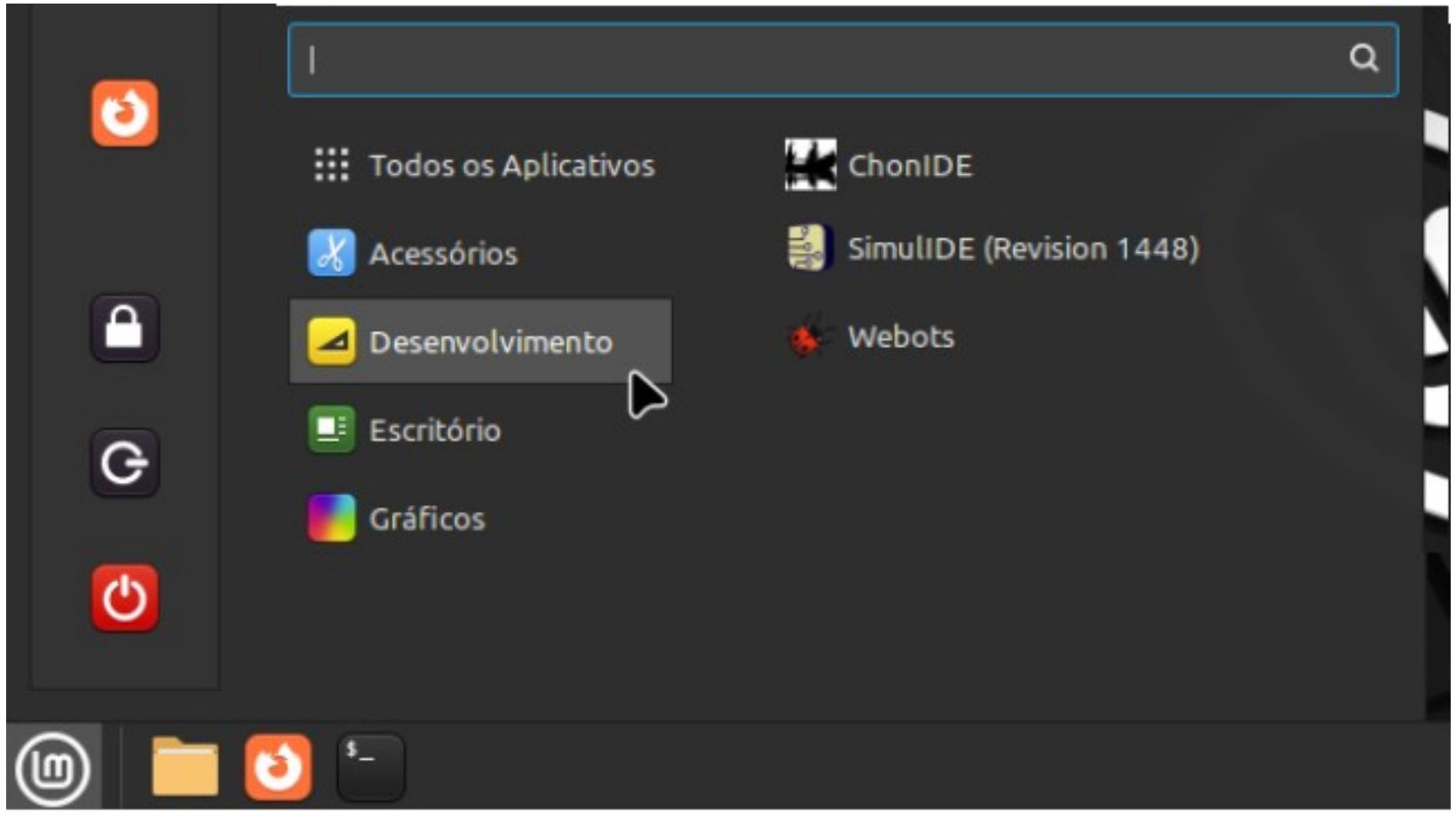
Image credits:
https://smurfs.fandom.com/wiki/Chef_Smurf
https://smurfs.fandom.com/pt/wiki/Fominha
https://smurfs.fandom.com/pt-br/wiki/%C3%81rvore_de_Dinheiro

OTHER TOOLS

Webots

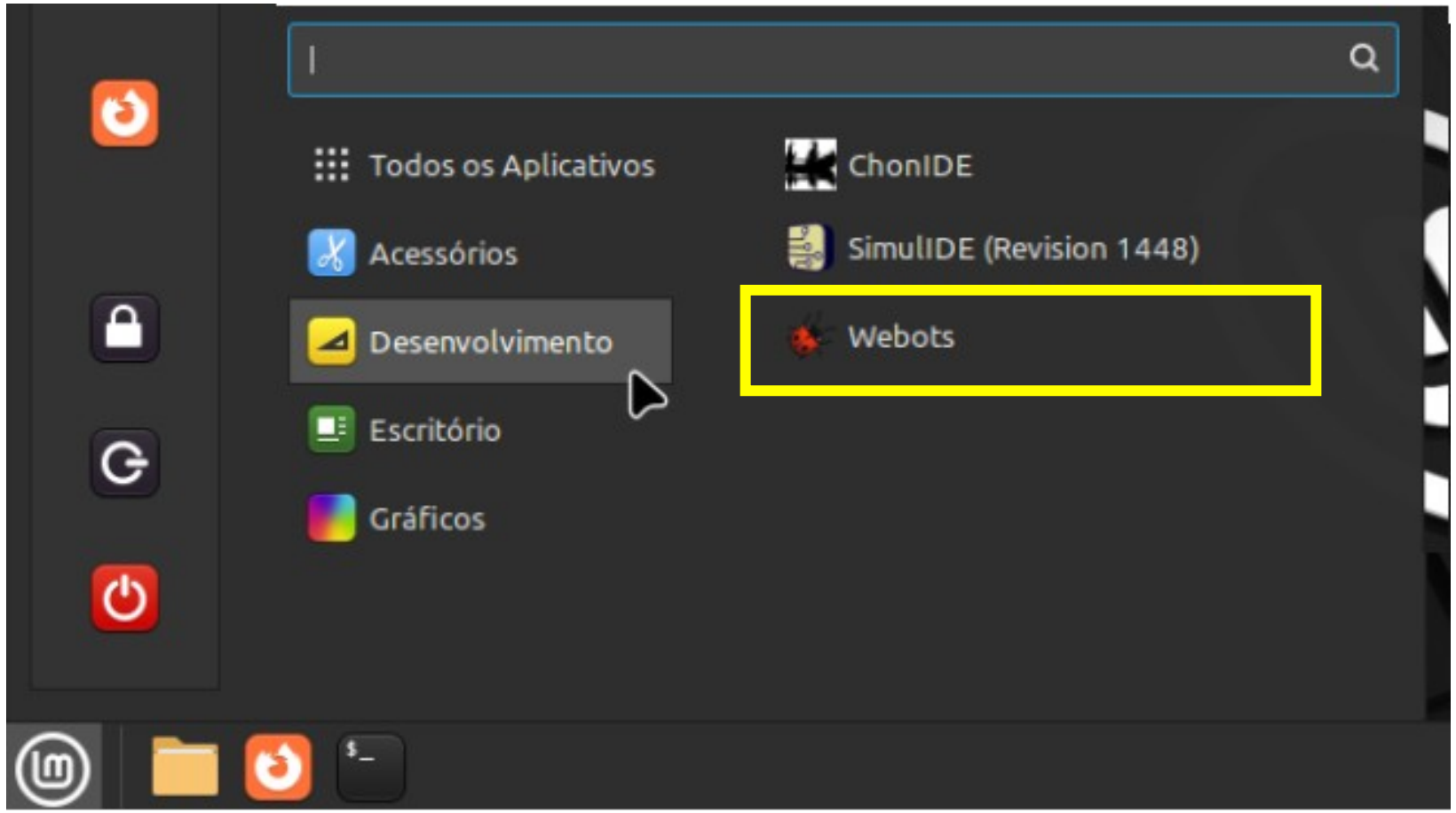
Michel, O. (1998). Webots: Symbiosis Between Virtual and Real Mobile Robots. In: Heudin, JC. (eds) Virtual Worlds. VW 1998. Lecture Notes in Computer Science(), vol 1434. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-68686-X_24

Webots



Manual de instalação
https://cyberbotics.com/doc/guide/installation-procedure#installing-the-debian-package-with-the-advanced-packaging-tool-apt

Webots



Manual de instalação
https://cyberbotics.com/doc/guide/installation-procedure#installing-the-debian-package-with-the-advanced-packaging-tool-apt

Webots

distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / Examples /

nilsonLazarin Car4WD Simulated World developed by @bptfreitas

Name	Last commit r
..	
Blink	developmer
Car4WD	Car4WD Sim
Blink.zip	developmer
Car4WD.zip	Car4WD Sim

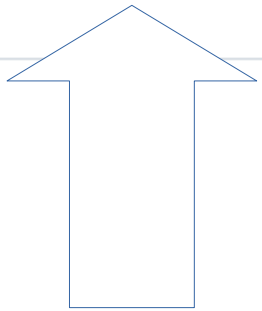
https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Car4WD.zip

Webots

distributedAndEmbeddedAI / course / 05-TheDevelopmentTool / Examples /

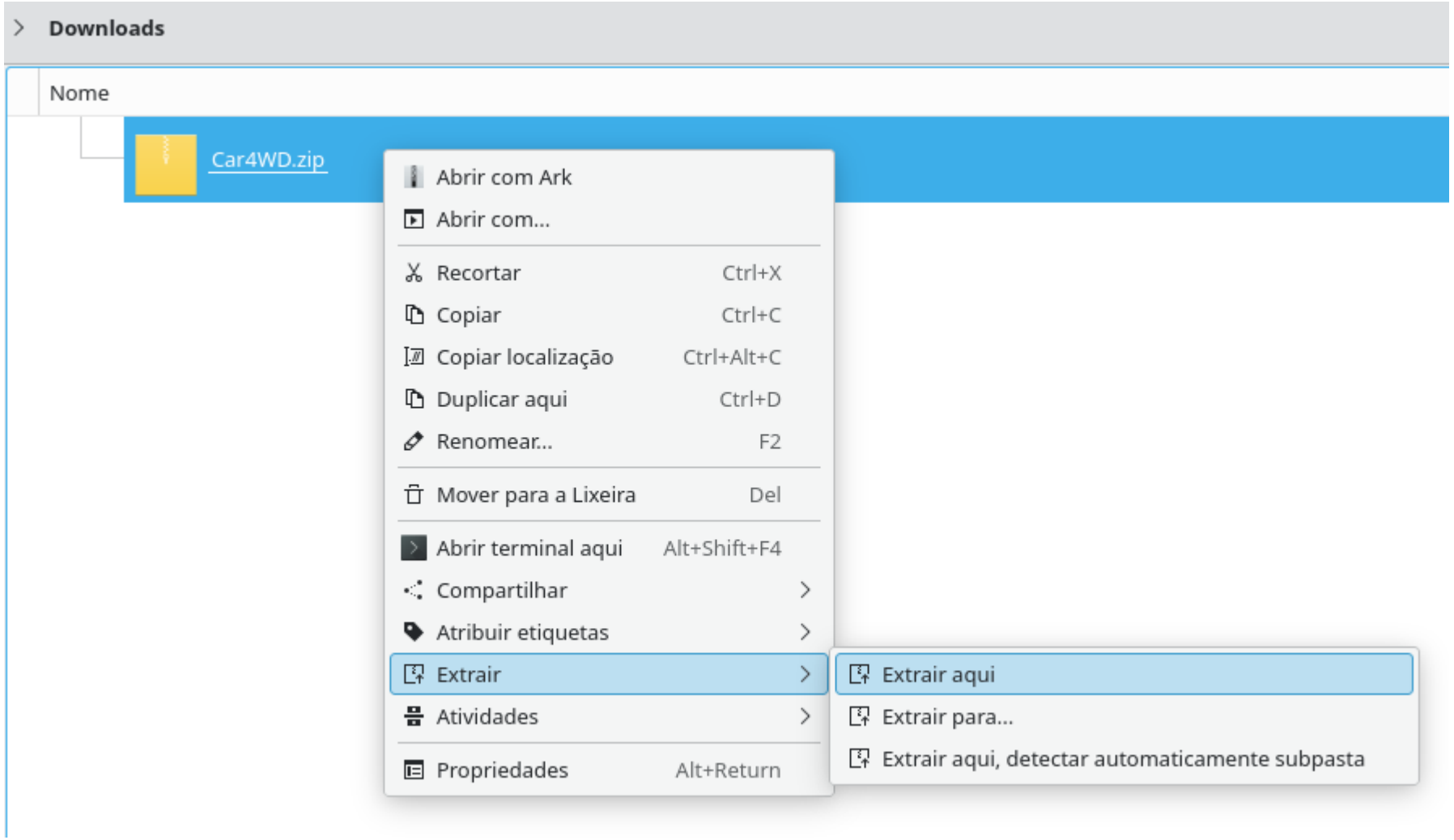
nilsonLazarin Car4WD Simulated World developed by @bptfreitas

Name	Last commit
..	
Blink	developmer
Car4WD	Car4WD Sim
Blink.zip	developmer
Car4WD.zip	Car4WD Sim

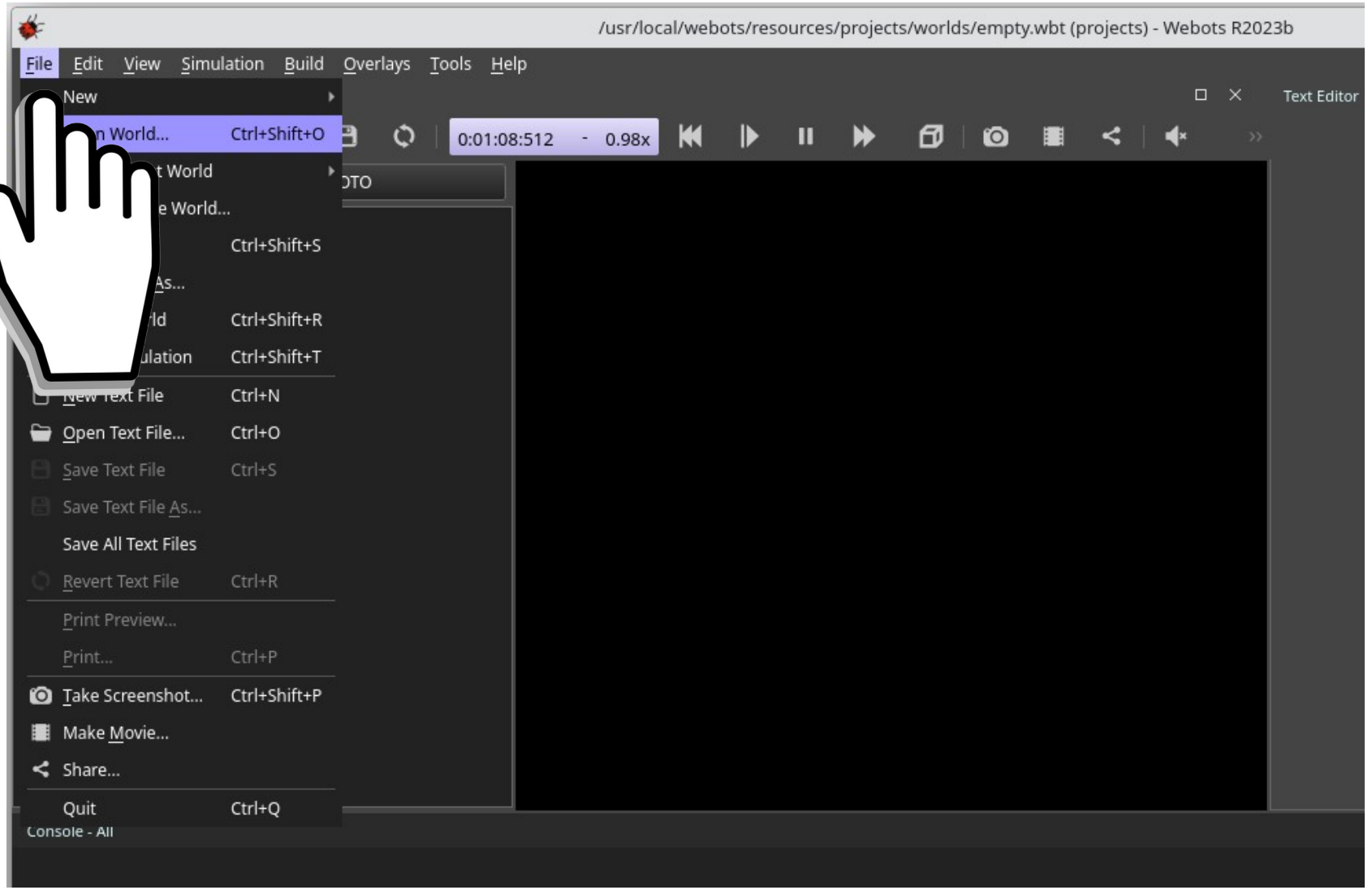


https://github.com/chon-group/distributedAndEmbeddedAI/raw/main/course/05-TheDevelopmentTool/Examples/Car4WD.zip

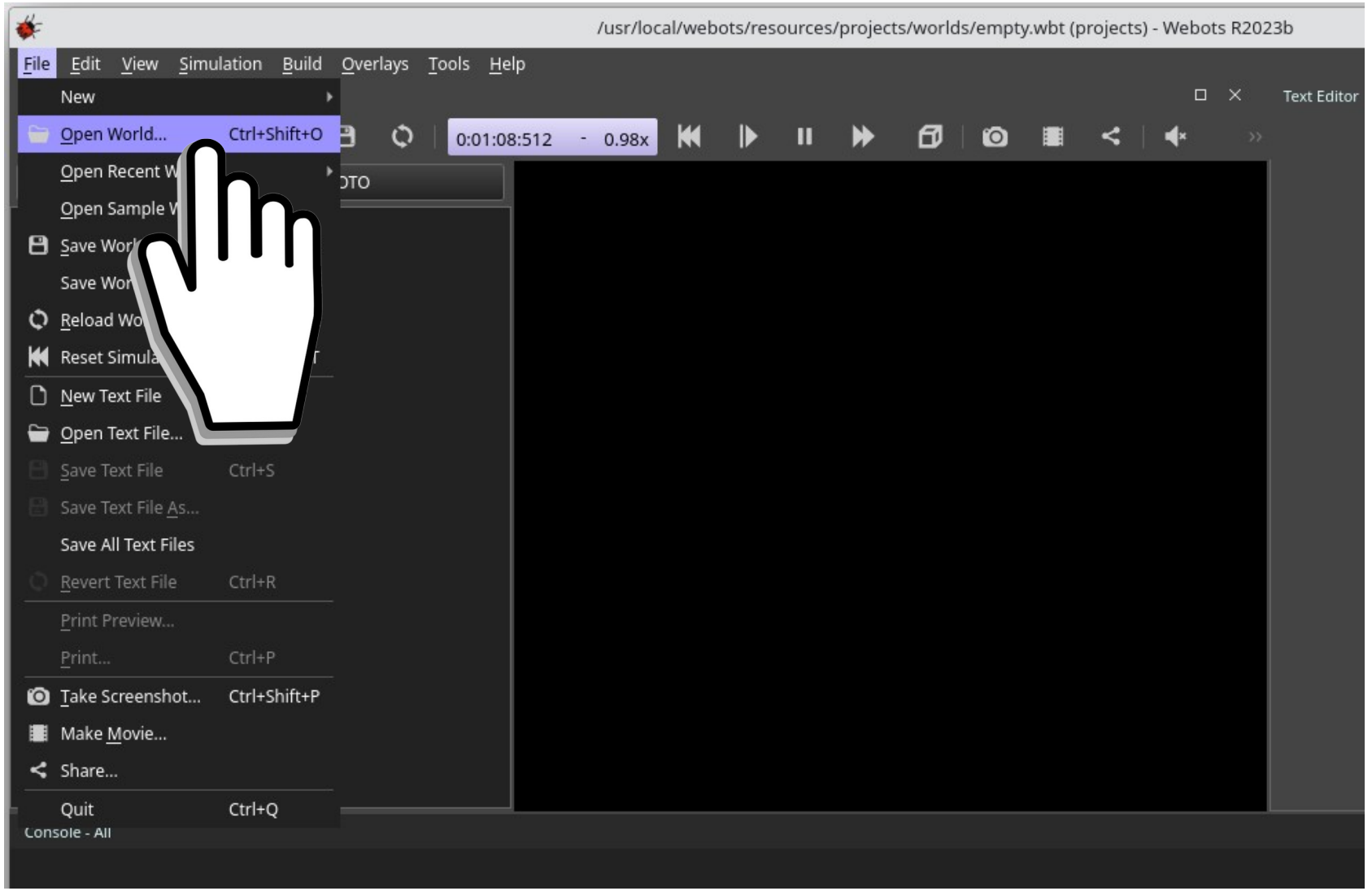
Webots


Downloads
Nome
Car4WD.zip
Abrir com Ark
Abrir com...
Recortar Ctrl+X
Copiar Ctrl+C
Copiar localização Ctrl+Alt+C
Duplicar aqui Ctrl+D
Renomear... F2
Mover para a Lixeira Del
Abrir terminal aqui Alt+Shift+F4
Compartilhar
Atribuir etiquetas
Extrair
Atividades
Propriedades Alt+Return
Extrair aqui
Extrair para...
Extrair aqui, detectar automaticamente subpasta

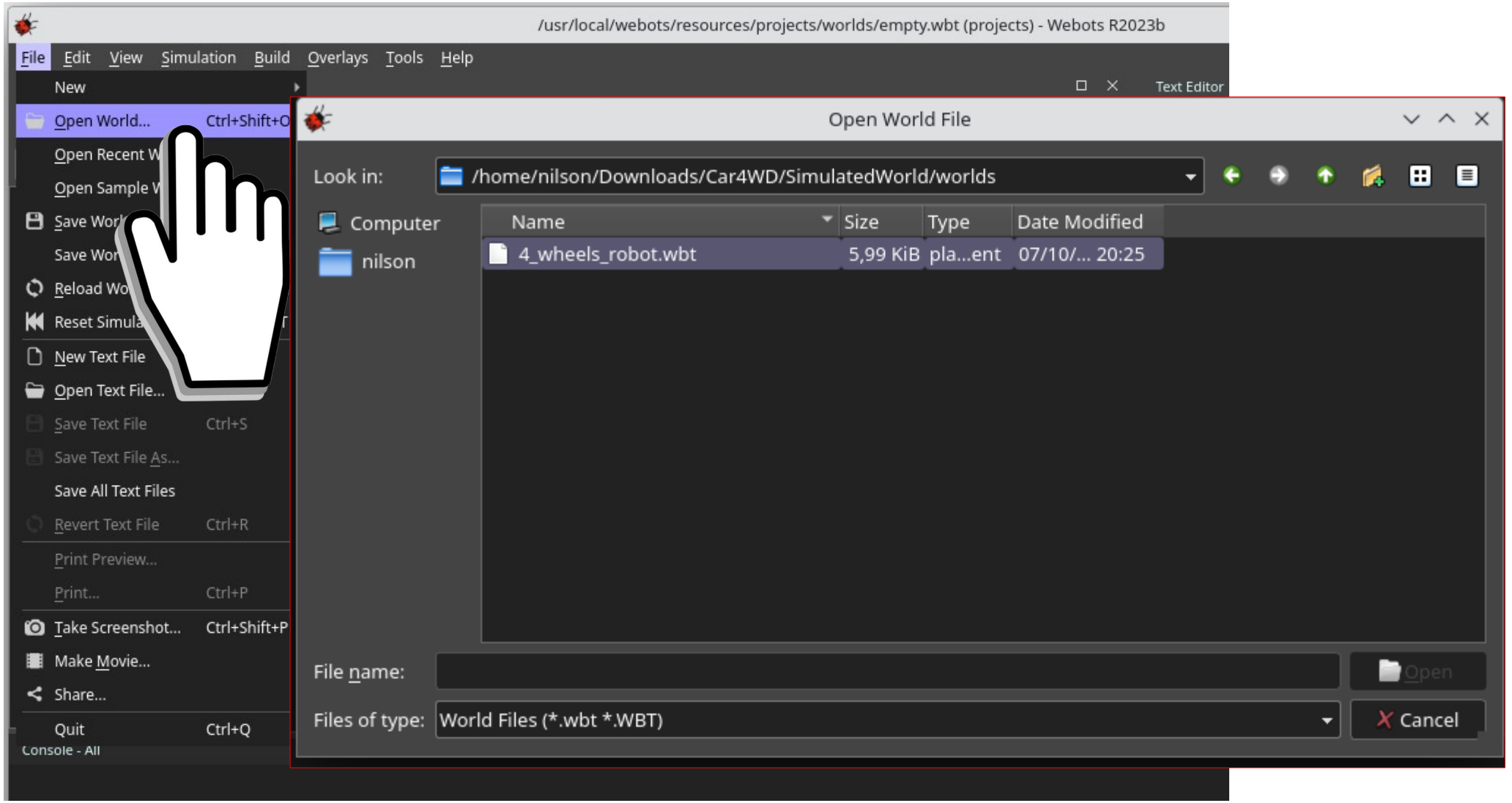
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Ctrl+Shift+O
Ctrl+Shift+S
Ctrl+Shift+R
Ctrl+Shift+T
Ctrl+N
Open Text File... Ctrl+O
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
0:01:08:512 - 0.98x
Text Editor
Console - All

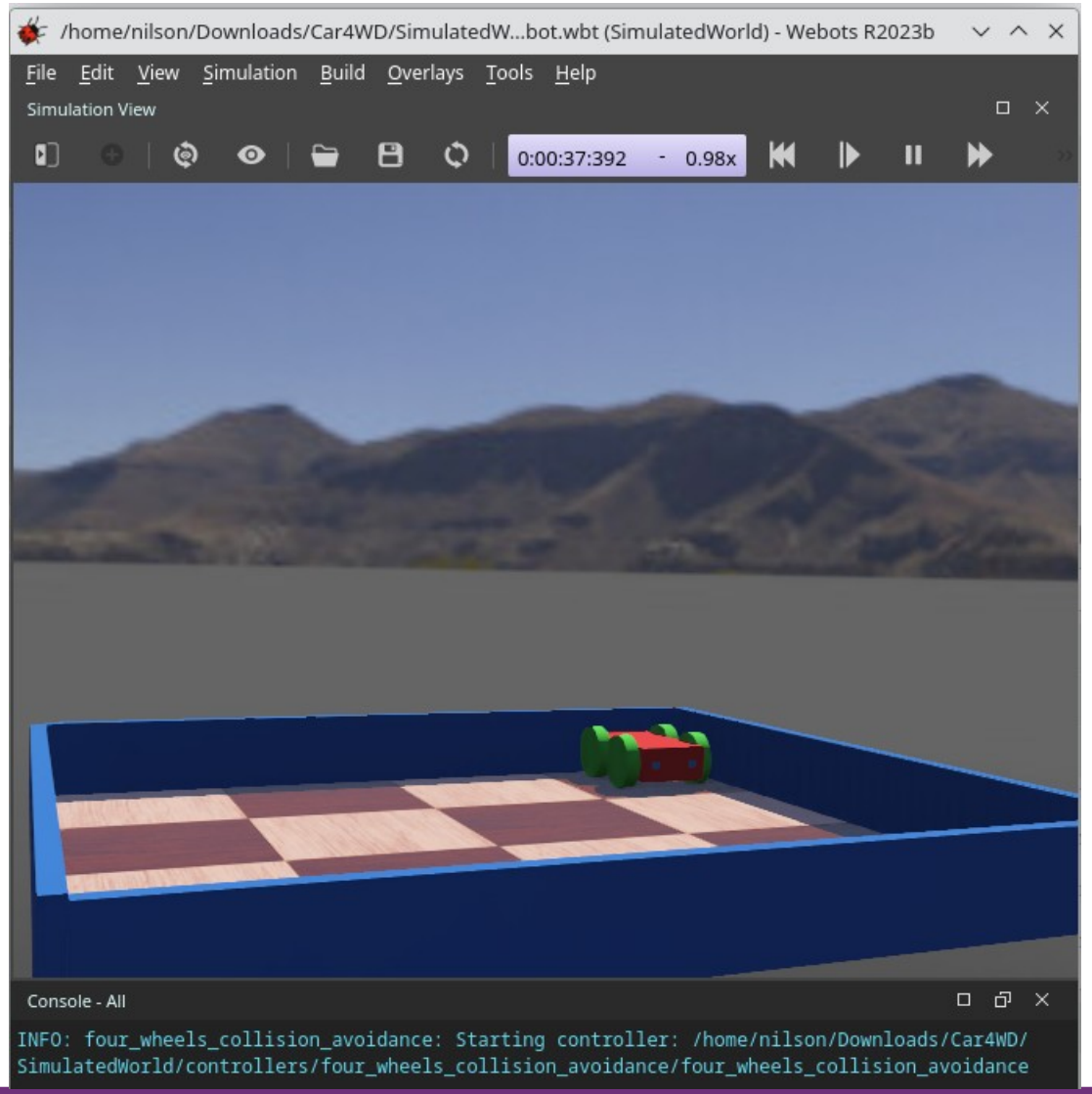
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Text Editor
0:01:08:512 - 0.98x
New
Open World... Ctrl+Shift+O
Open Recent W
Open Sample W
Save Wor
Save Wor
Reload Wo
Reset Simula
New Text File
Open Text File...
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
Console - All

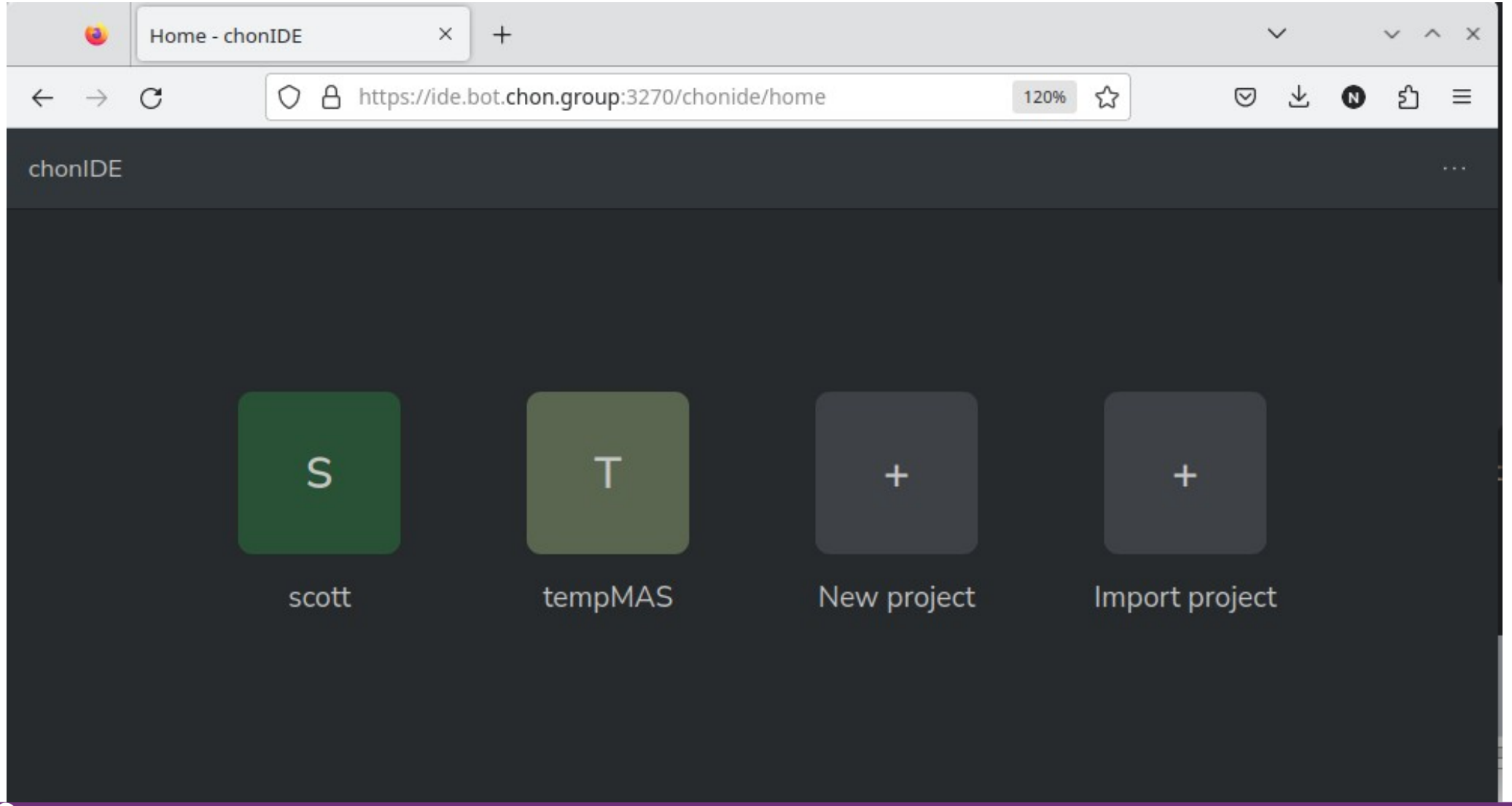
Webots


/usr/local/webots/resources/projects/worlds/empty.wbt (projects) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
New
Open World... Ctrl+Shift+O
Open Recent W
Open Sample W
Save Wor
Save Wor
Reload Wo
Reset Simula
New Text File
Open Text File...
Save Text File Ctrl+S
Save Text File As...
Save All Text Files
Revert Text File Ctrl+R
Print Preview...
Print... Ctrl+P
Take Screenshot... Ctrl+Shift+P
Make Movie...
Share...
Quit Ctrl+Q
Console - All
Text Editor
Open World File
Look in: /home/nilson/Downloads/Car4WD/SimulatedWorld/worlds
Computer
nilson
Name Size Type Date Modified
4_wheels_robot.wbt 5,99 KiB pla...ent 07/10/... 20:25
File name:
Open
Files of type: World Files (*.wbt *.WBT)
Cancel

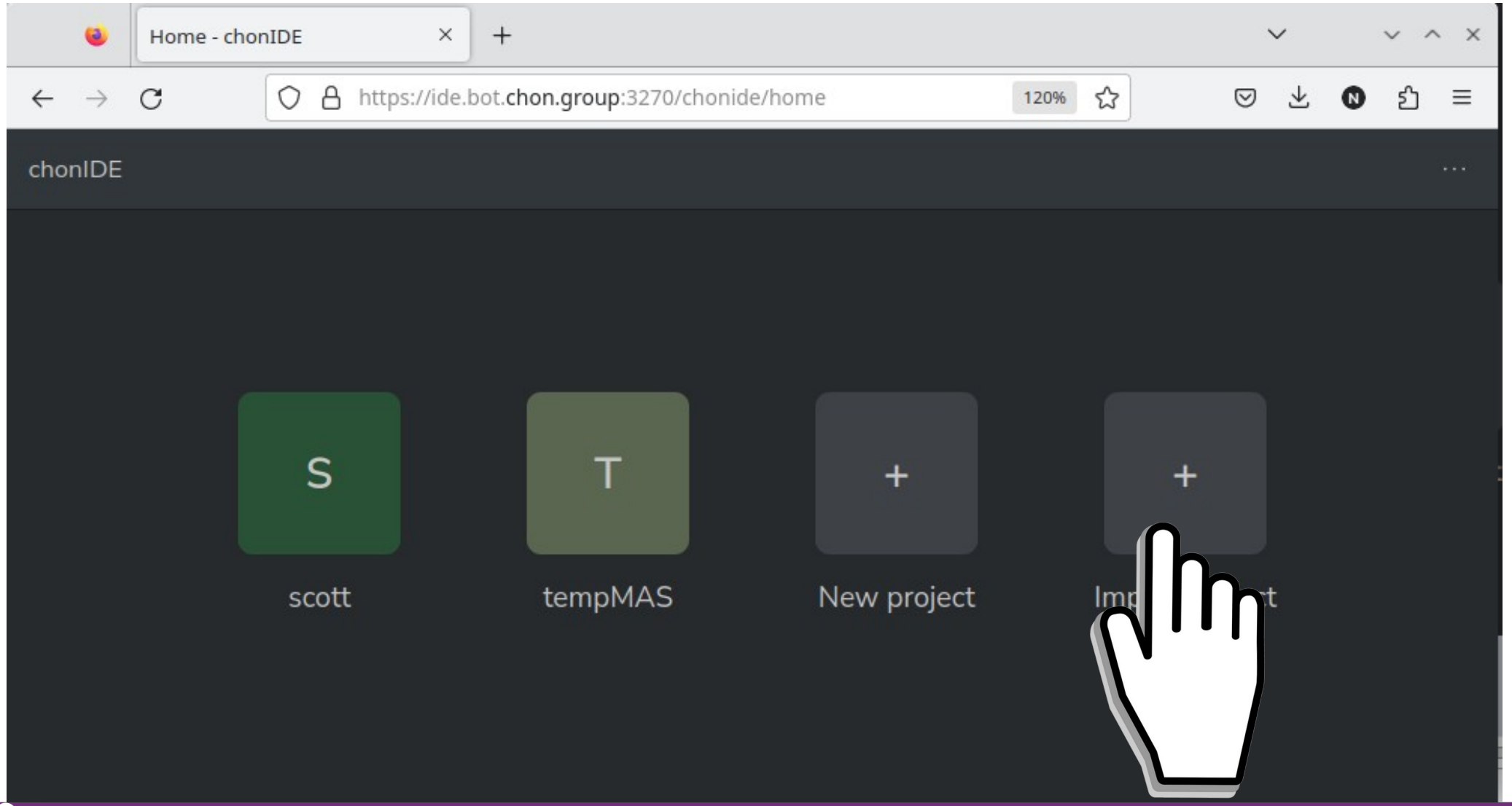
Webots


/home/nilson/Downloads/Car4WD/SimulatedW...bot.wbt (SimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:00:37:392 - 0.98x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/Downloads/Car4WD/SimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

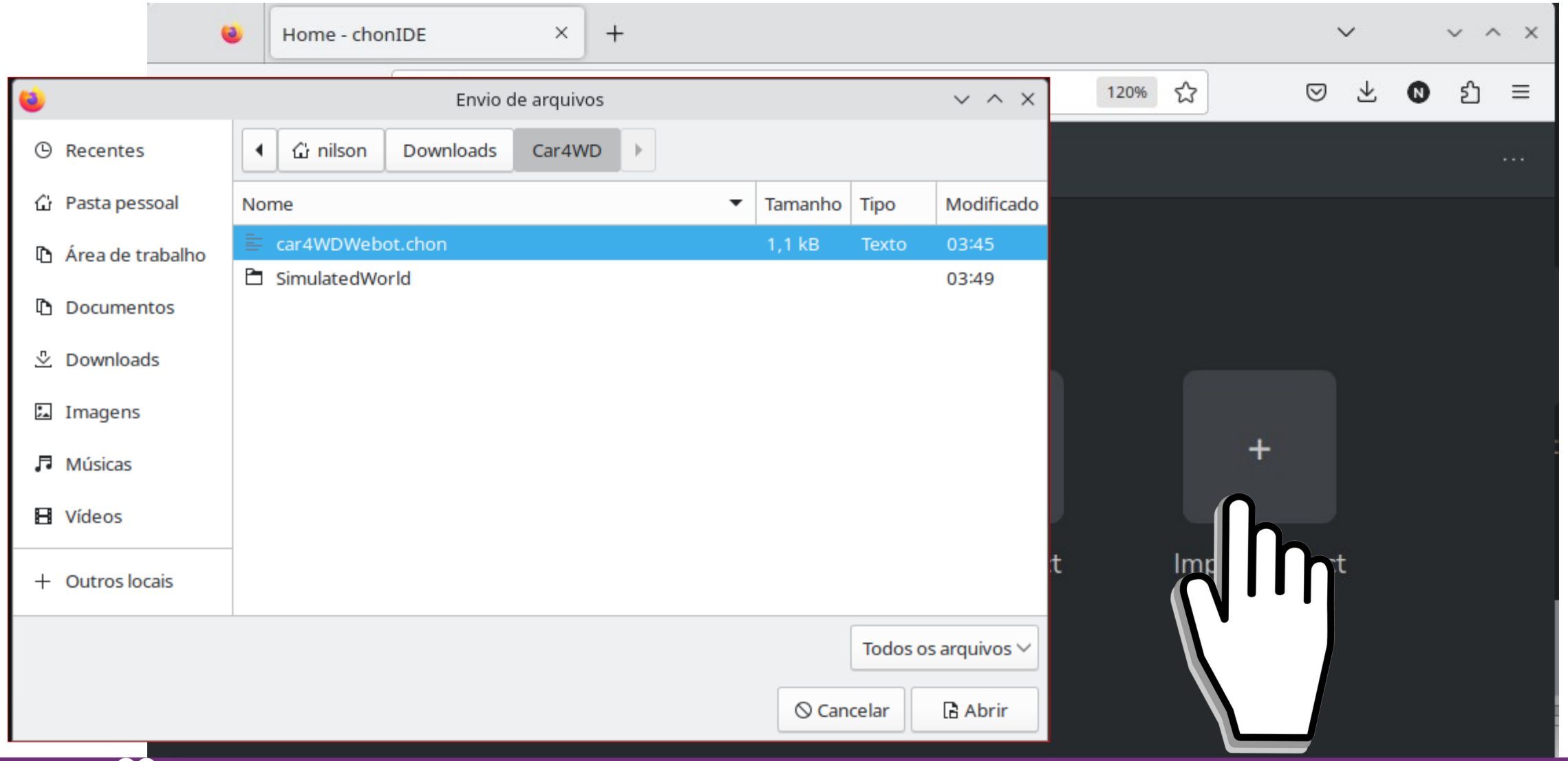
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
T
+
+
scott
tempMAS
New project
Import project

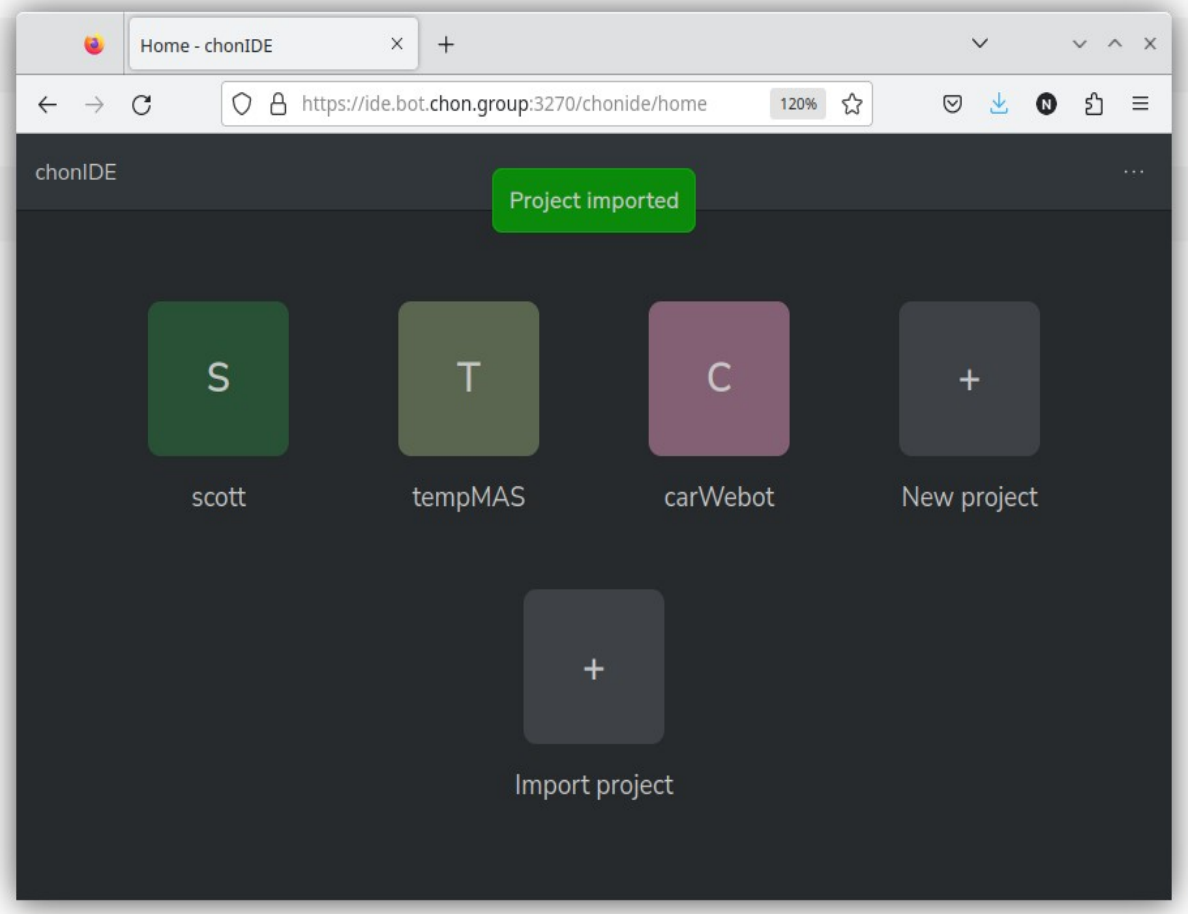
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
T
+
+
scott
tempMAS
New project

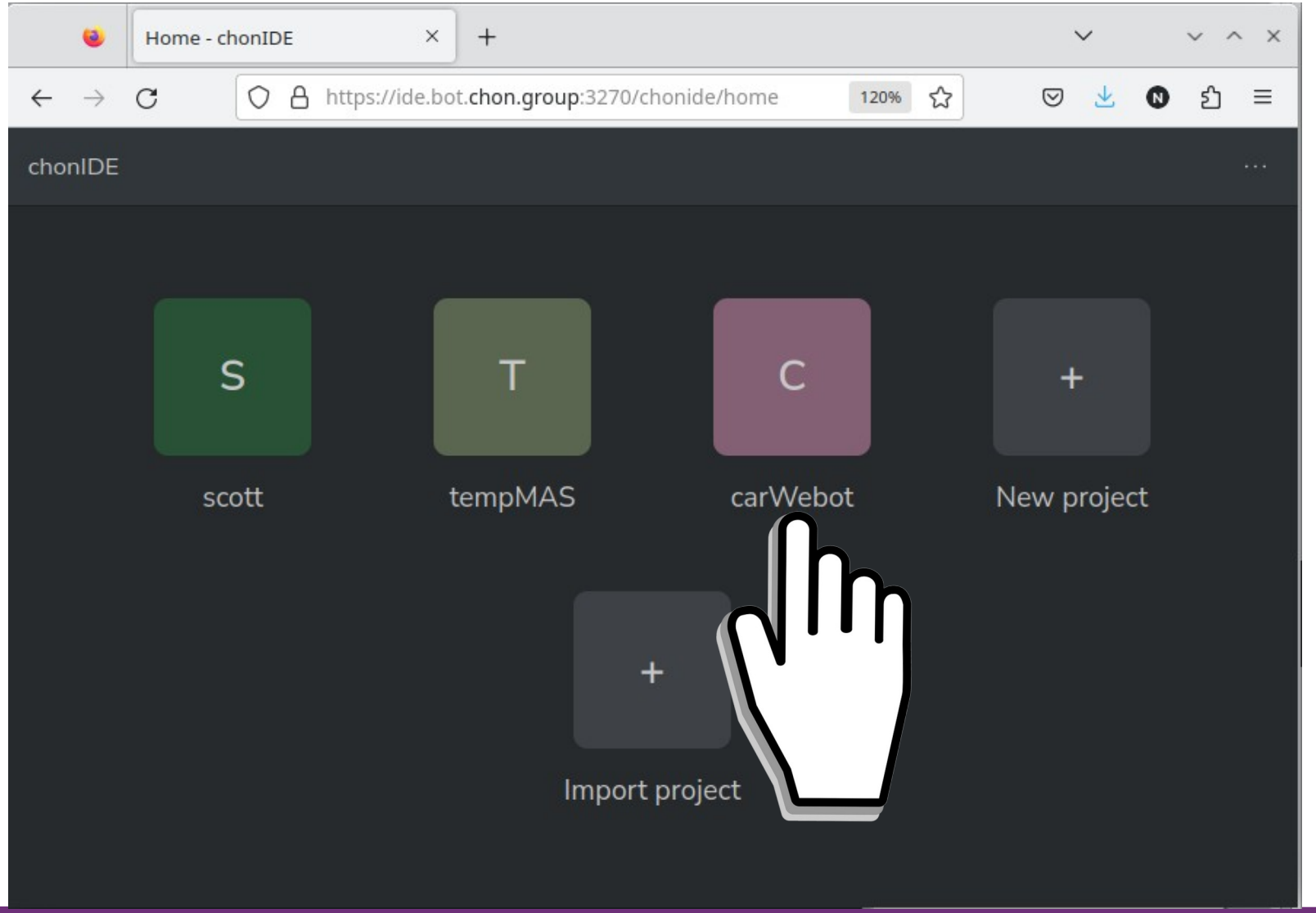
Webots


Home - chonIDE
120%
Envio de arquivos
Recentes
Pasta pessoal
Área de trabalho
Documentos
Downloads
Imagens
Músicas
Vídeos
Outros locais
nilson
Downloads
Car4WD
Nome
Tamanho
Tipo
Modificado
car4WDWebot.chon
1,1 kB
Texto
03:45
SimulatedWorld
03:49
Todos os arquivos
Cancelar
Abrir

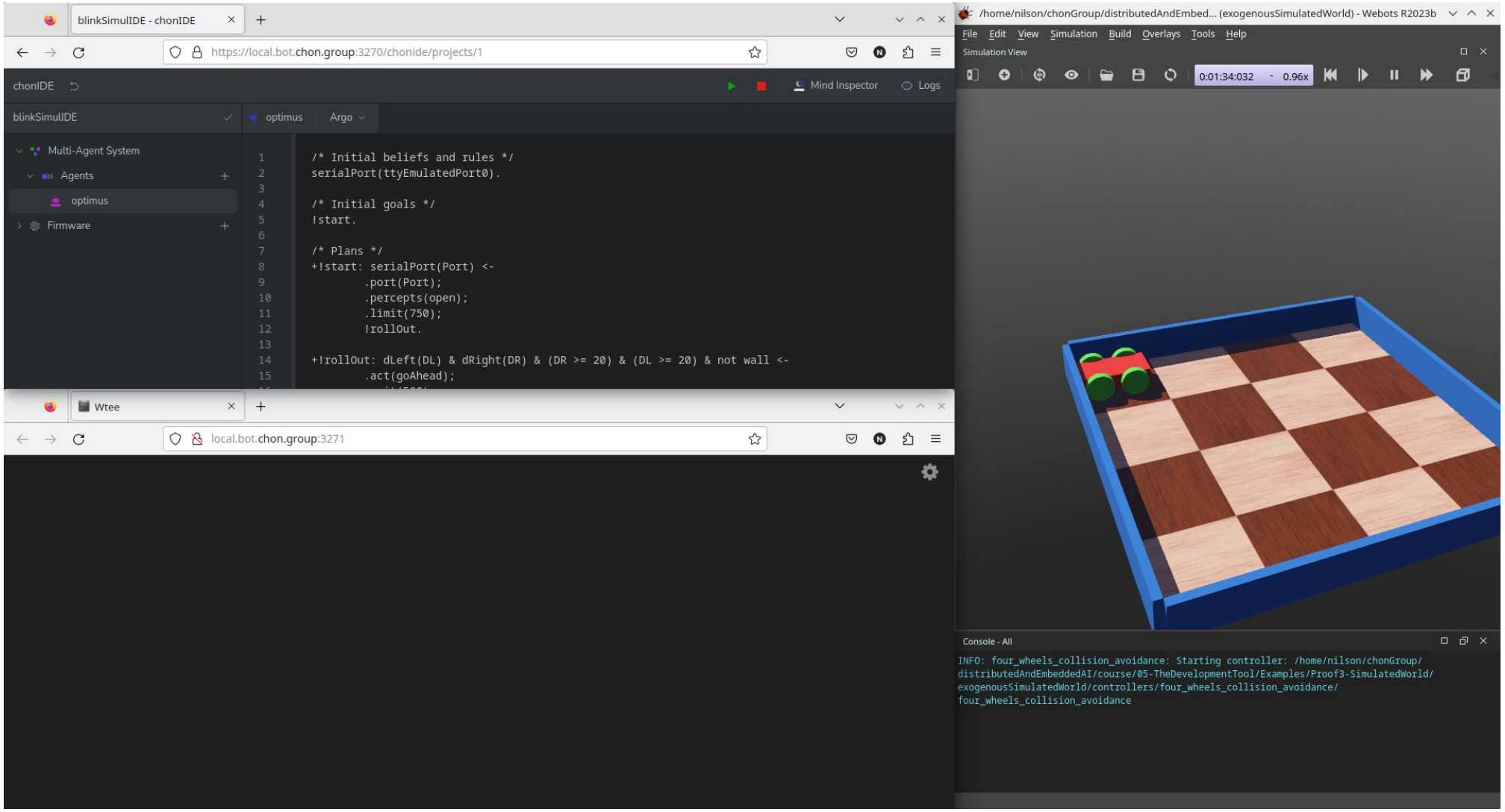
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
Project imported
S
T
C
+
scott
tempMAS
carWebot
New project
+
Import project

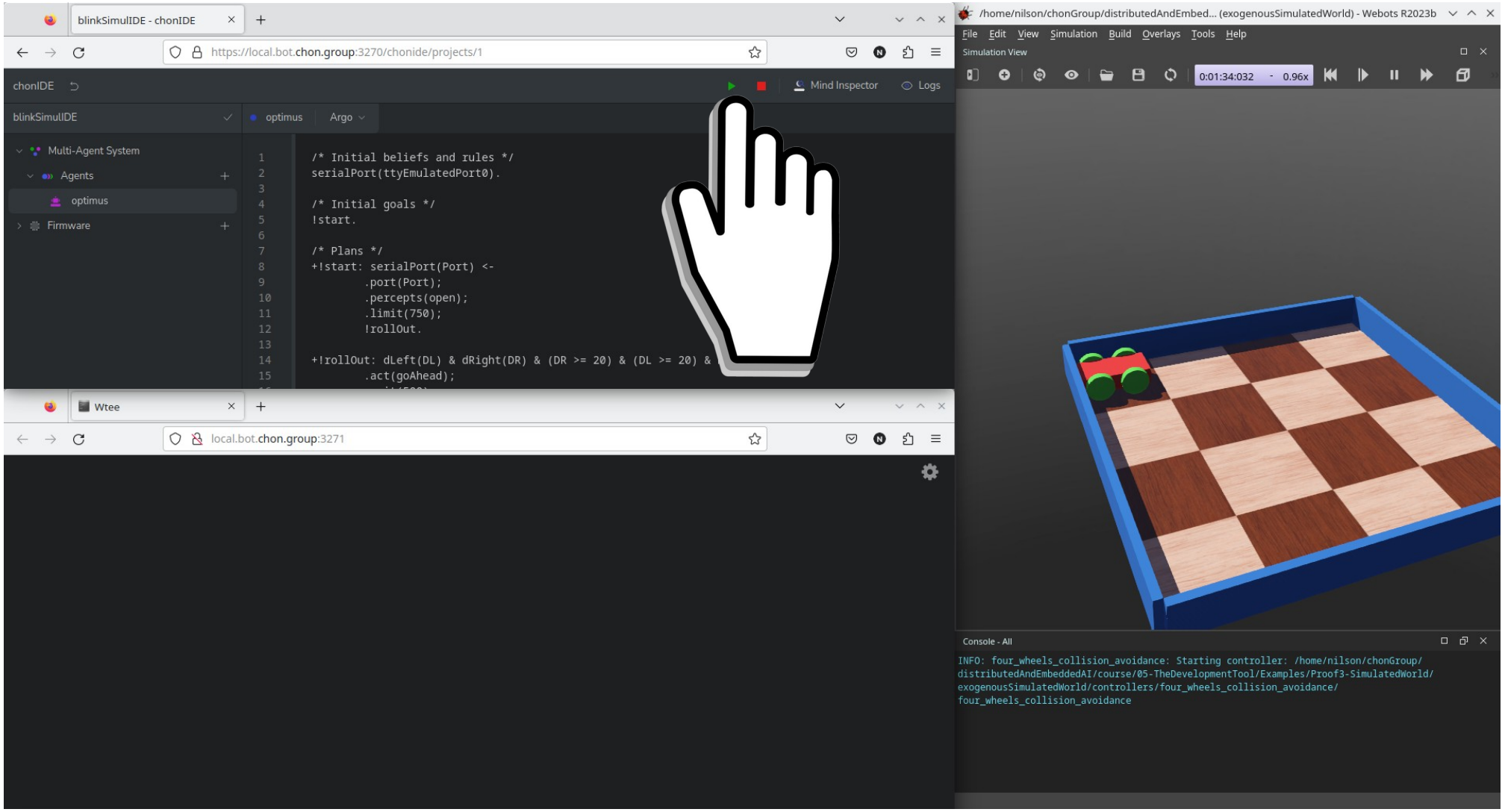
Webots


Home - chonIDE
https://ide.bot.chon.group:3270/chonide/home
120%
chonIDE
S
scott
T
tempMAS
C
carWebot
+
New project
+
Import project

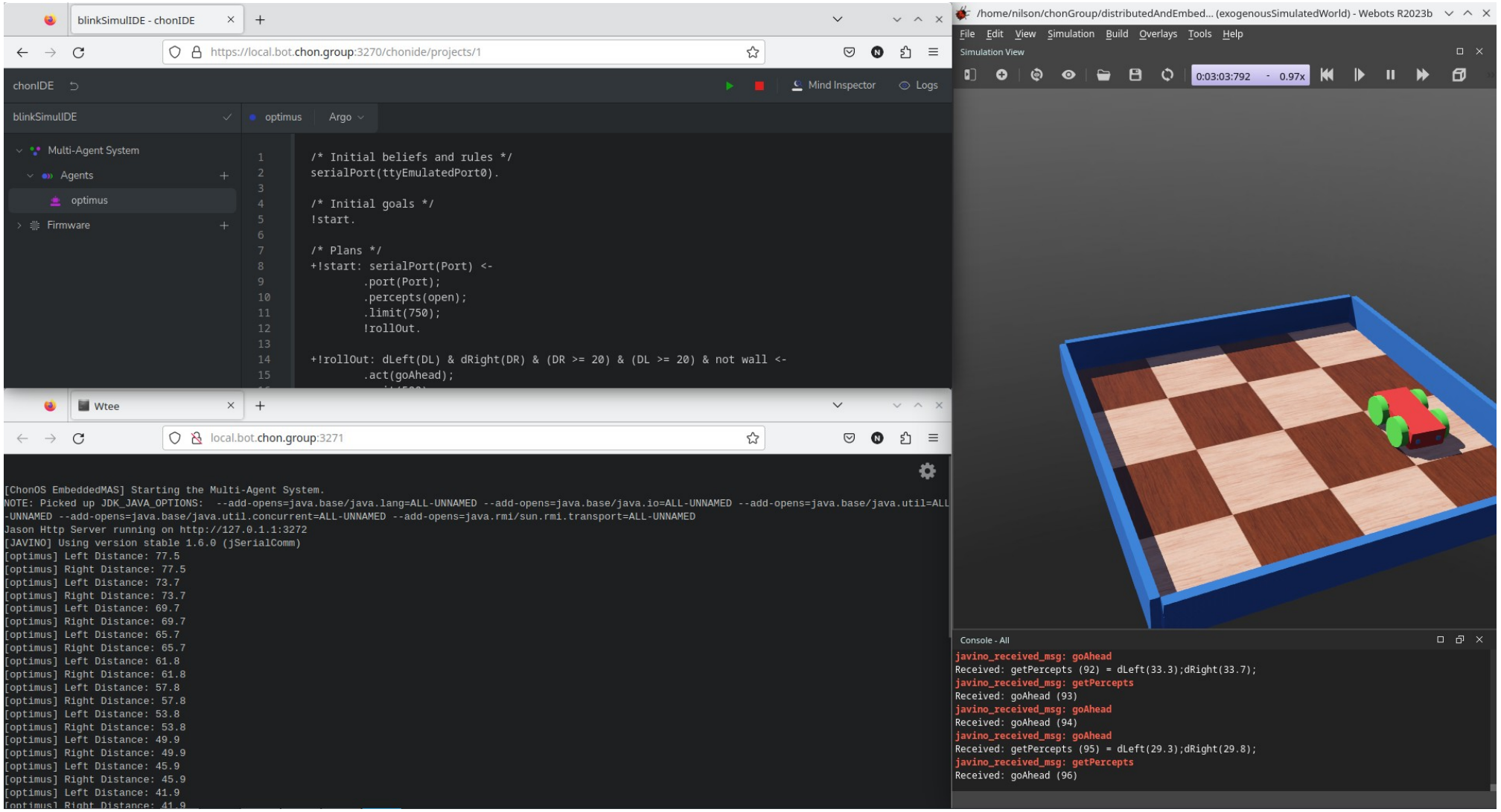
Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) & not wall <-
.act(goAhead);
Wtee
local.bot.chon.group:3271
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:01:34:032 - 0.96x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/chonGroup/distributedAndEmbeddedAI/course/05-TheDevelopmentTool/Examples/Proof3-SimulatedWorld/exogenousSimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) &
.act(goAhead);
Wtee
local.bot.chon.group:3271
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:01:34:032 - 0.96x
Console - All
INFO: four_wheels_collision_avoidance: Starting controller: /home/nilson/chonGroup/distributedAndEmbeddedAI/course/05-TheDevelopmentTool/Examples/Proof3-SimulatedWorld/exogenousSimulatedWorld/controllers/four_wheels_collision_avoidance/four_wheels_collision_avoidance

Webots


blinkSimulIDE - chonIDE
https://local.bot.chon.group:3270/chonide/projects/1
chonIDE
Mind Inspector
Logs
blinkSimulIDE
optimus
Argo
Multi-Agent System
Agents
optimus
Firmware
/* Initial beliefs and rules */
serialPort(ttyEmulatedPort0).
/* Initial goals */
!start.
/* Plans */
+!start: serialPort(Port) <-
.port(Port);
.percepts(open);
.limit(750);
!rollOut.
+!rollOut: dLeft(DL) & dRight(DR) & (DR >= 20) & (DL >= 20) & not wall <-
.act(goAhead);
Wtee
local.bot.chon.group:3271
[ChonOS EmbeddedMAS] Starting the Multi-Agent System.
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Jason Http Server running on http://127.0.1.1:3272
[JAVINO] Using version stable 1.6.0 (jSerialComm)
[optimus] Left Distance: 77.5
[optimus] Right Distance: 77.5
[optimus] Left Distance: 73.7
[optimus] Right Distance: 73.7
[optimus] Left Distance: 69.7
[optimus] Right Distance: 69.7
[optimus] Left Distance: 65.7
[optimus] Right Distance: 65.7
[optimus] Left Distance: 61.8
[optimus] Right Distance: 61.8
[optimus] Left Distance: 57.8
[optimus] Right Distance: 57.8
[optimus] Left Distance: 53.8
[optimus] Right Distance: 53.8
[optimus] Left Distance: 49.9
[optimus] Right Distance: 49.9
[optimus] Left Distance: 45.9
[optimus] Right Distance: 45.9
[optimus] Left Distance: 41.9
/home/nilson/chonGroup/distributedAndEmbed... (exogenousSimulatedWorld) - Webots R2023b
File Edit View Simulation Build Overlays Tools Help
Simulation View
0:03:03:792 - 0.97x
Console - All
javino_received_msg: goAhead
Received: getPercepts (92) = dLeft(33.3);dRight(33.7);
javino_received_msg: getPercepts
Received: goAhead (93)
javino_received_msg: goAhead
Received: goAhead (94)
javino_received_msg: goAhead
Received: getPercepts (95) = dLeft(29.3);dRight(29.8);
javino_received_msg: getPercepts
Received: goAhead (96)

Acknowledgements

THANK YOU!

nilson.lazarin@cefet-rj.br

UDESC
UNIVERSIDADE DO ESTADO DE SANTA CATARINA

ALTO VALE
CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ

fapesc
Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina

CEFET/RJ
Uned Nova Friburgo

Instituto de Computação