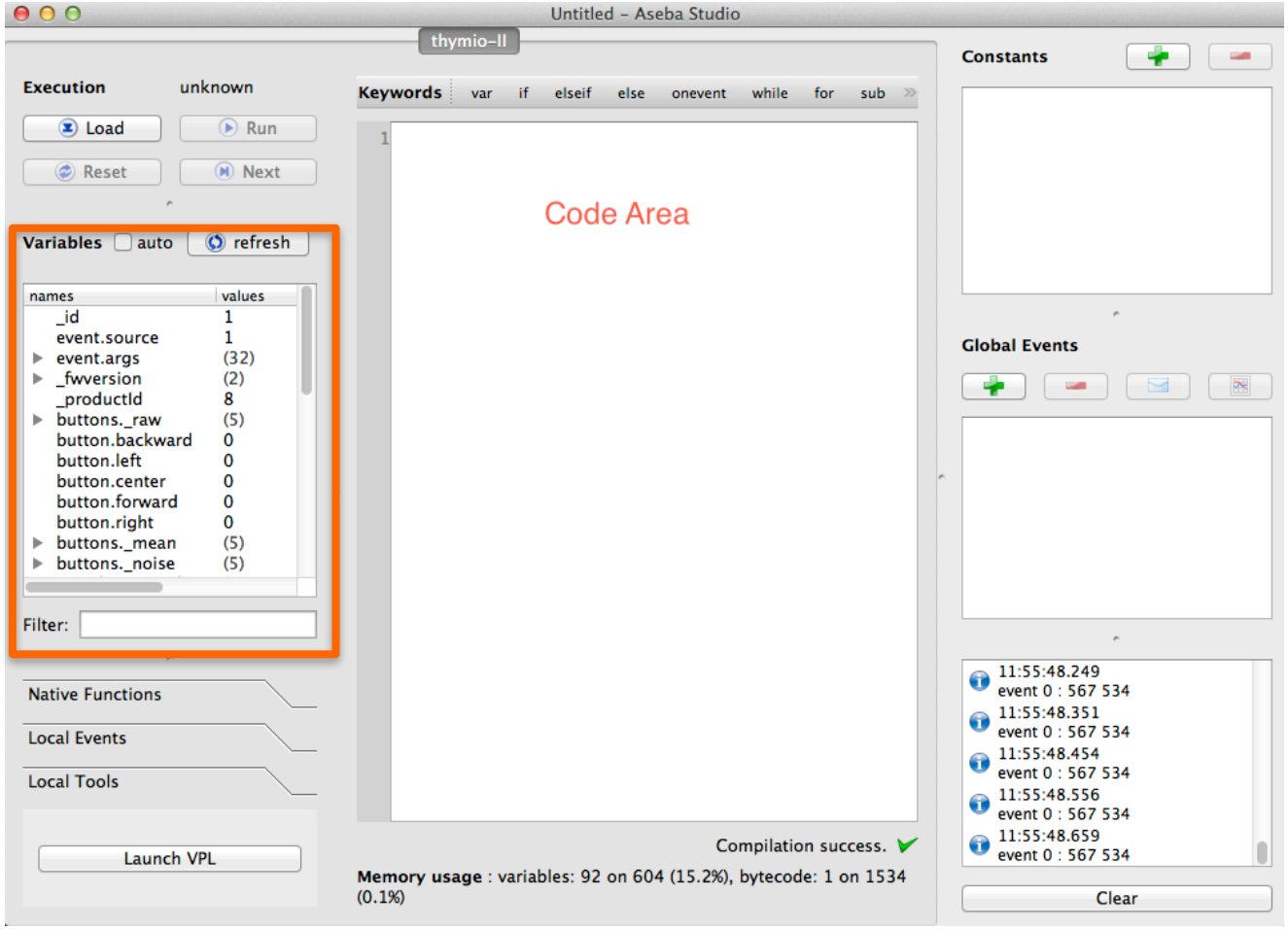
CSCI 1108
Introduction to
Experimental Robotics

Software environments for robotics
and
Simulator

Aseba Studio



https://aseba.wikidot.com/en:thymioapi

Android™ Based Robotics

Nicolas Oros, Jeffrey Kritchmar

https://youtu.be/2czndpV6pWw

Robotics Software Environments

- Ubuntu (Unix OS)

- ROS (Robot Operating System)
  www.ros.org
  Subscription architecture, wide range of services, many robots have ROS nodes

- OpenCV (Open Computer Vision)
  opencv.org

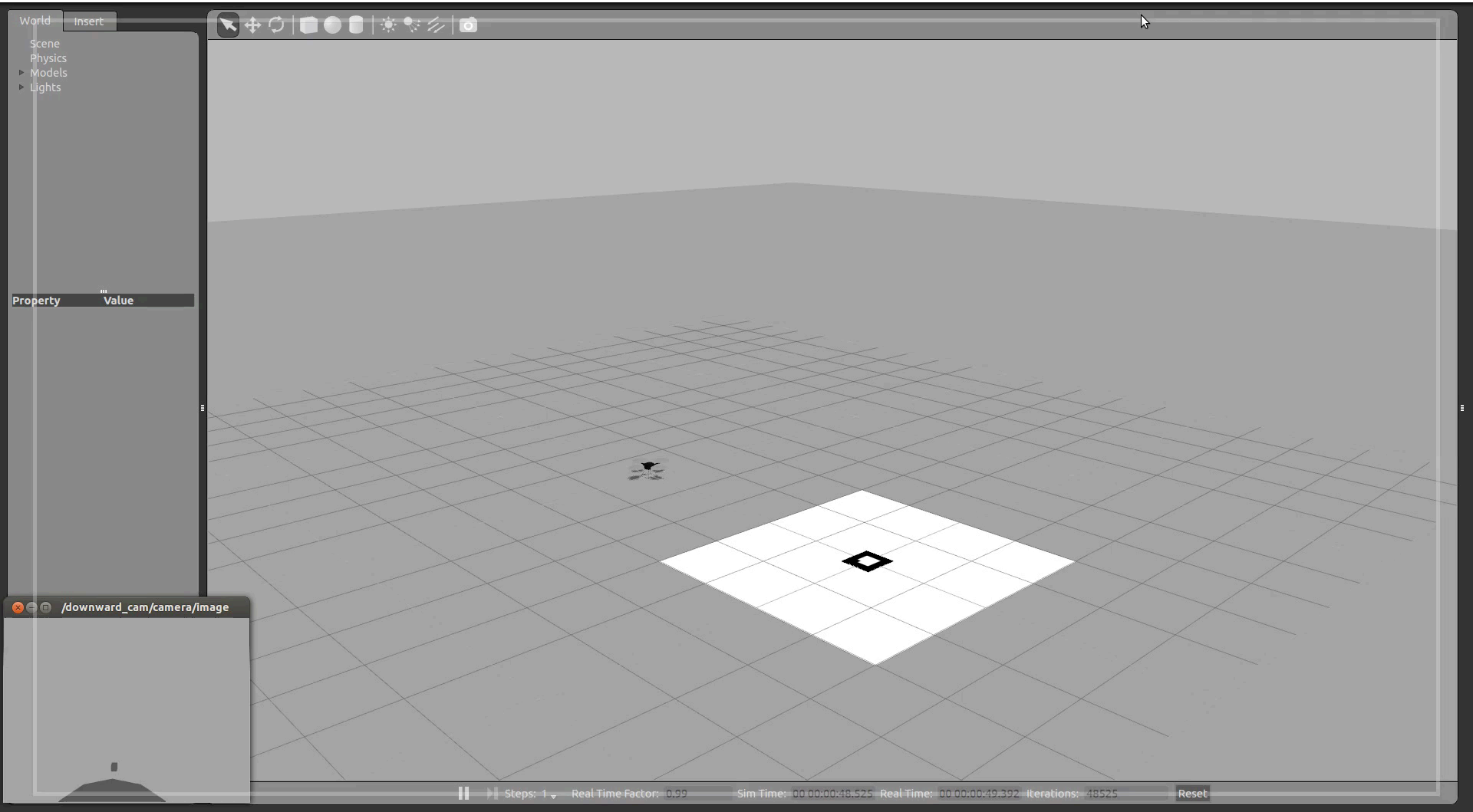
  Most commonly computer vision package

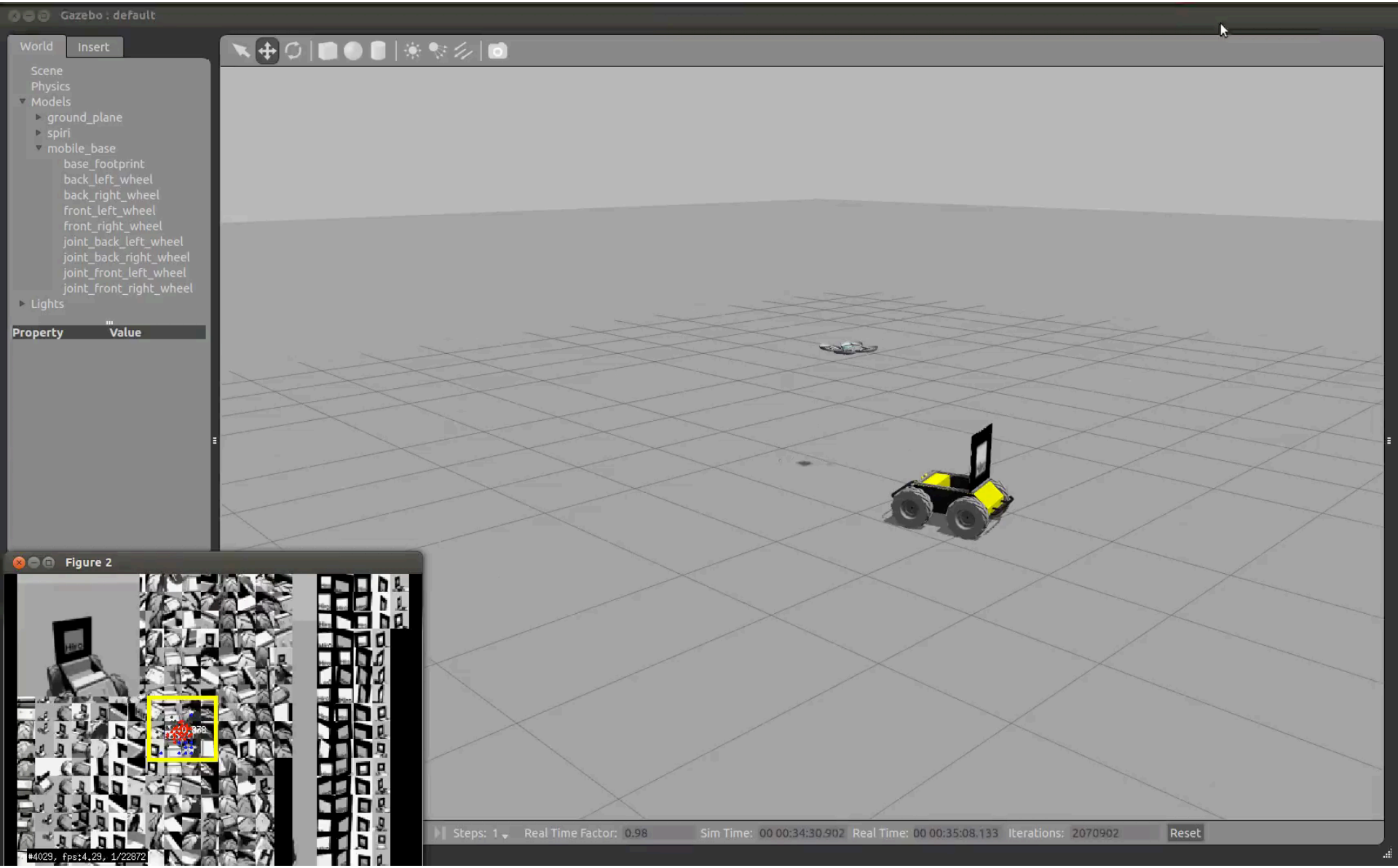
Robotics Simulators

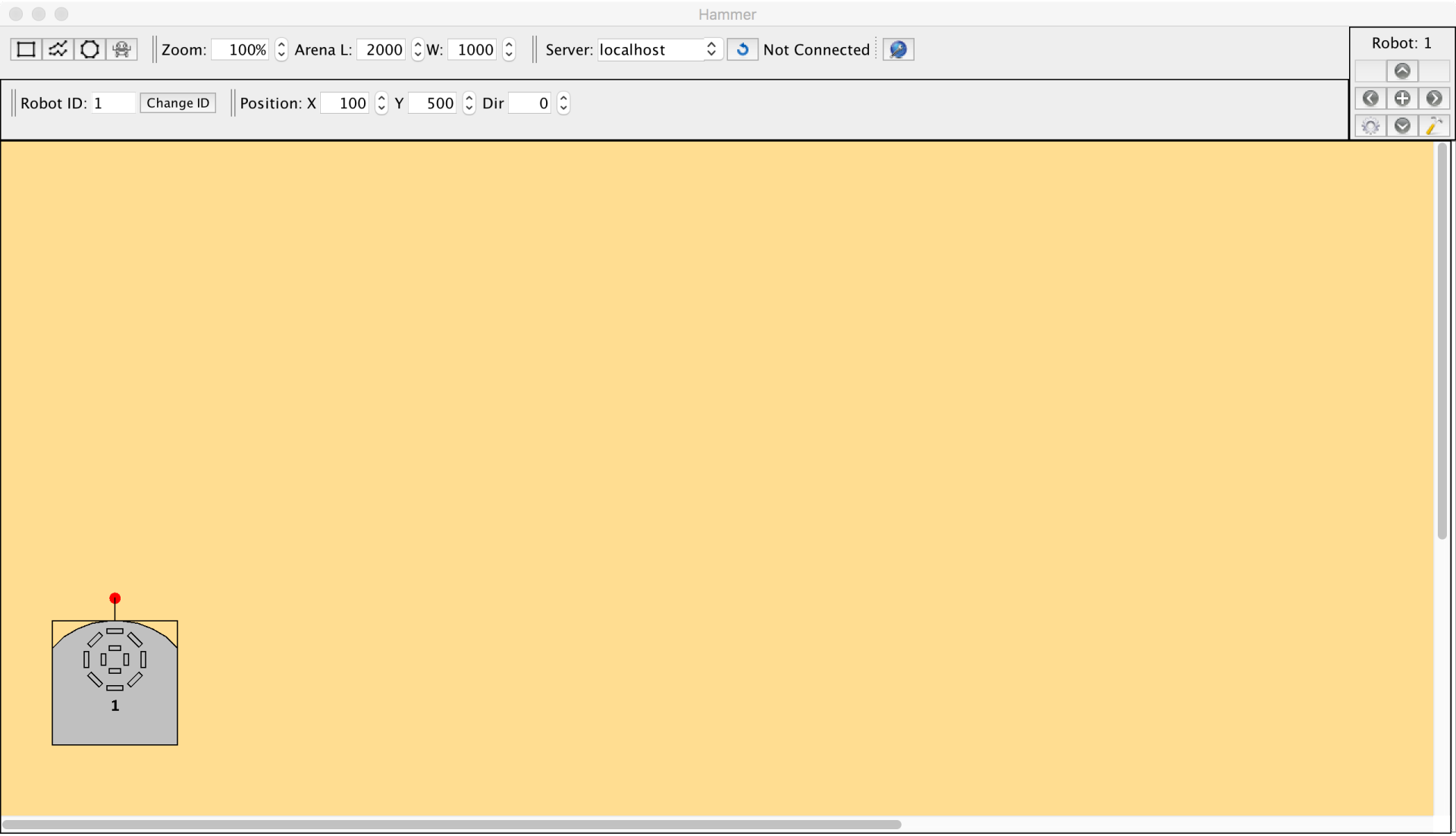
- Physical robots have traditionally been expensive and often require careful handling (safety)
- A common way in robotics is therefore to use programs that simulate the physical behaviour of a robot.
- Simulators are useful for initial development, but roboticists always stress their limitations

Example of open source simulator

GAZEBO




Gazebo : default
World
Insert
Scene
Physics
Models
ground_plane
spiri
mobile_base
base_footprint
back_left_wheel
back_right_wheel
front_left_wheel
front_right_wheel
joint_back_left_wheel
joint_back_right_wheel
joint_front_left_wheel
joint_front_right_wheel
Lights
Property
Value
Figure 2
#4029, fps:4.29, 1/22872
Steps: 1
Real Time Factor: 0.98
Sim Time: 00 00:34:30.902
Real Time: 00 00:35:08.133
Iterations: 2070902
Reset


Hammer
Zoom: 100%
Arena L: 2000
W: 1000
Server: localhost
Not Connected
Robot: 1
Robot ID: 1
Change ID
Position: X 100
Y 500
Dir 0
1

Some more programming with Aseba

If you are starting to program, please follow the first tutorial carefully and please ask questions if things are not clear. While this is not a programming course itself, it is a nice way to start programming with a simple programming language and the basic concepts of programing should be easy to learn (still means you might have to spend some time with it as it is different to a natural language).

Hint: The learning centre should be a good place to get individual help, and of course do not hesitate to ask your TAs and Instructors.

Loops in Aseba

A loop is a basic operation in a programming language to allow repeated execution of some code. In Aseba it has the form (syntax)

for *name* in *first*:*last* do
    code
end

When the *interpreter* comes to the “for” keyword it invokes the following “microprogram” (internal set of instructions):

1a. Set running variable *name* to the value specified in *first. (only first time)*
1b. Increment name by one (*name=name+1*) (if not first time)
2. Check value of variable *name.*
    2a. If *name<=last* then execute code and go to 1b
    2b. If *name>last* then leave loop and execute code after end statement.

Conditional statements in Aseba

A conditional statement is a specific code to dierect an operation in different directions depending on logical statements such as ”if a variable is larger than a certain value then do *something* else do *something-else*. In Aseba it has the form (syntax)

if *name* > *value* then
    code
end

When the *interpreter* comes to the “if” keyword it evaluates the logical statement (here if the value in variable *name* is larger than the value of variable *value)*. If it is true it executes the code. If not it continues after the end statement.

Data in Aseba

- Variables

  `var name` Regular placeholders for numbers (basic data type)

  `var list[]` Lists: If we want to refer to specific values in a collection of data, we need basic construct for data collections. One of the most basic constructs is a list. Later you learn more about advanced data structures like arrays, trees, etc

- Constants

  Constants are special variables that should not Change during the execution of a program

```
var i = 0
var a[3] = [11,22,33]


for i in 1:3 do
      a[i]=a[i-1]+i
end
```