ROSY

An elegant language to teach the pure reactive nature of robot programming

Hugo Pacheco (FCUP & INESC TEC) and Nuno Macedo (FEUP & INESC TEC)
IRC'20

Motivation

Robotics for novice programmers

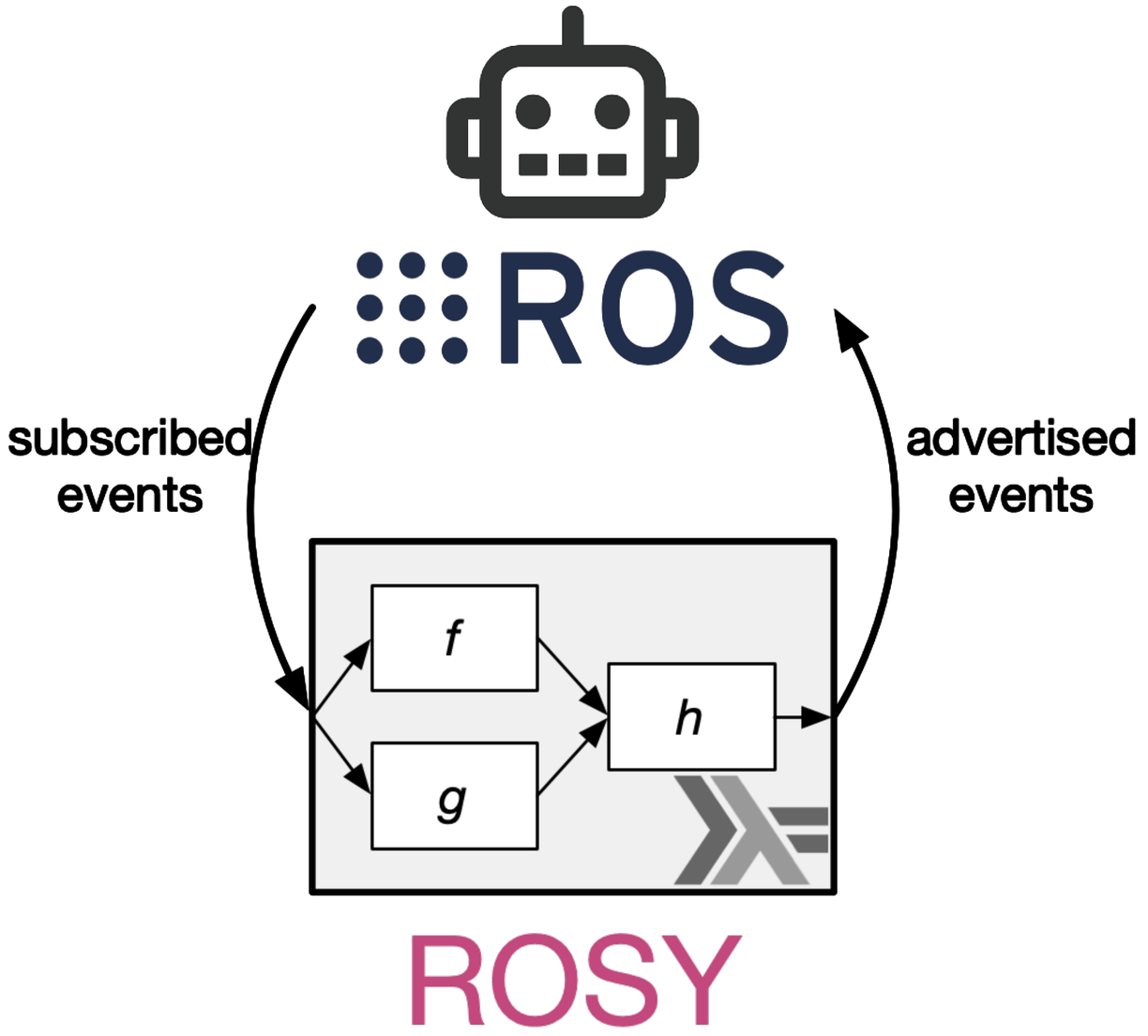
- Robotic software
  - appeals to programming novices
  - requires reasoning about relevant concepts (modularization, communication, …)
- But this richness is accompanied by similar complexity
  - clocks, sampling rates, synchronization, ...

Our vision

Robotics for novice programmers

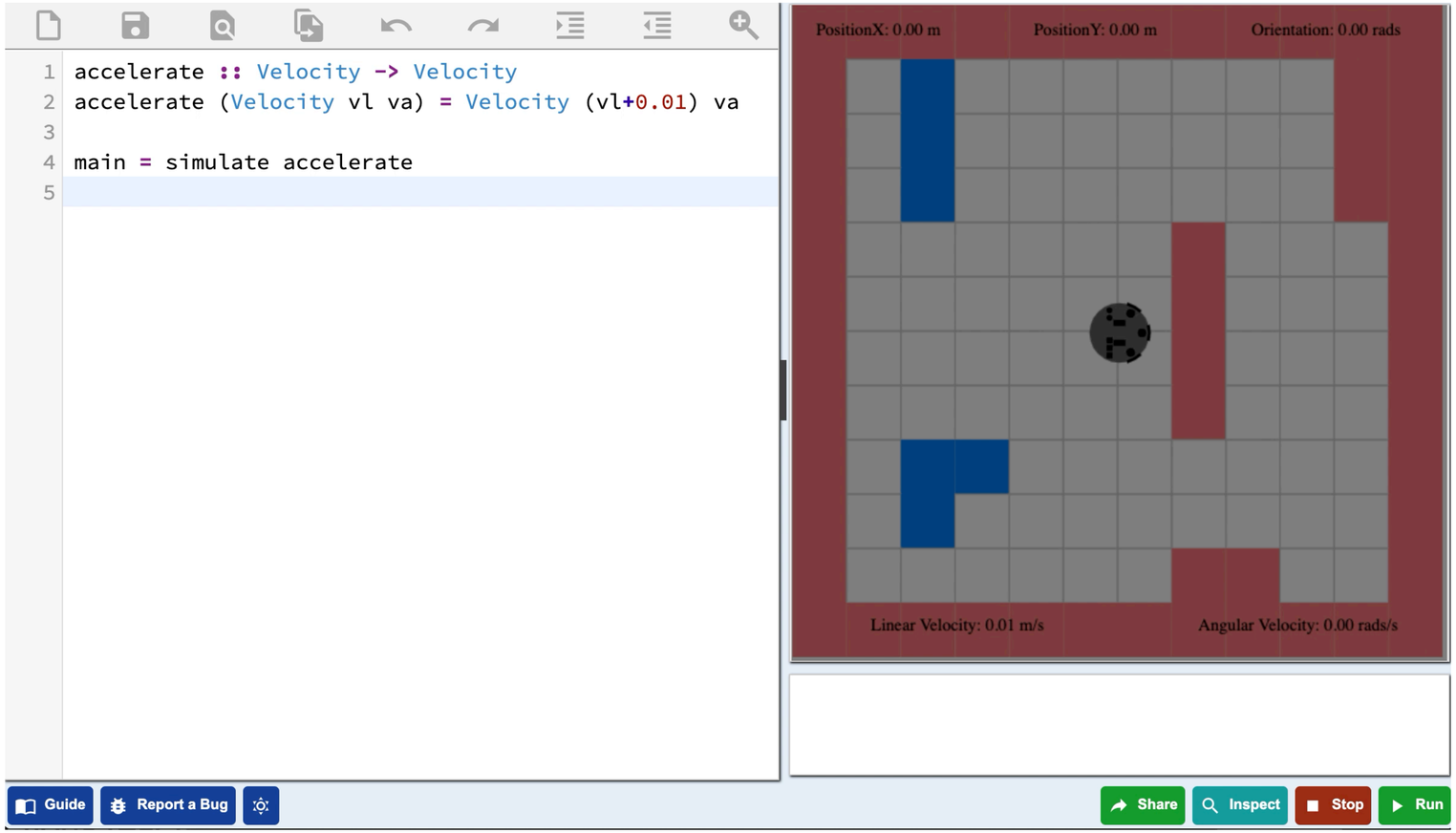
Introductory robot programming languages should:

- be compatible with robotic standards
- adopt a simple declarative programming style with a cause-and-effect interface
- support compositional component design
- rely on general-purpose programming languages



ROSY by example

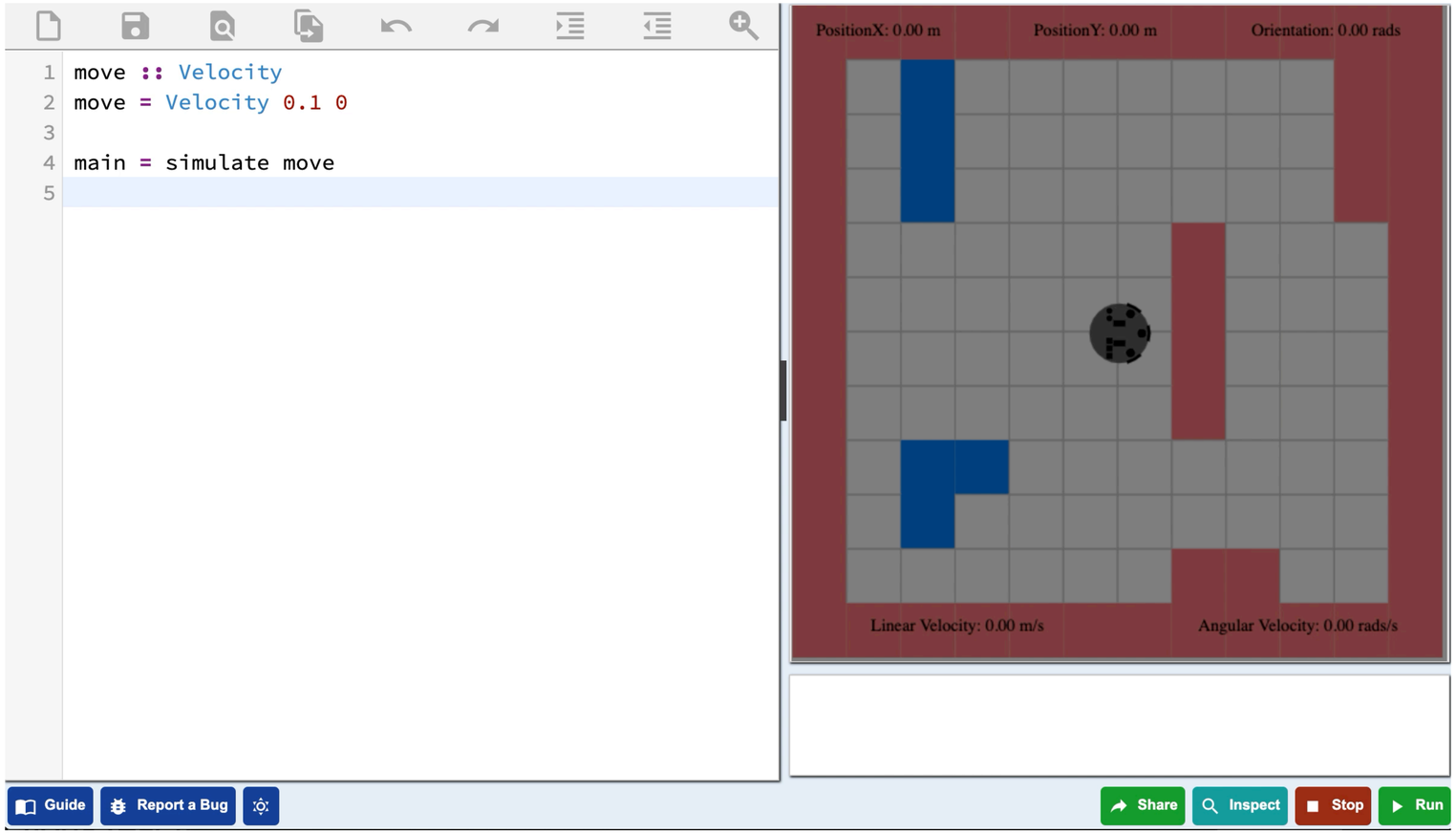
Basic reactive control



Pure functions reacting to and producing events, may be wrappers to ROS topics

ROSY by example

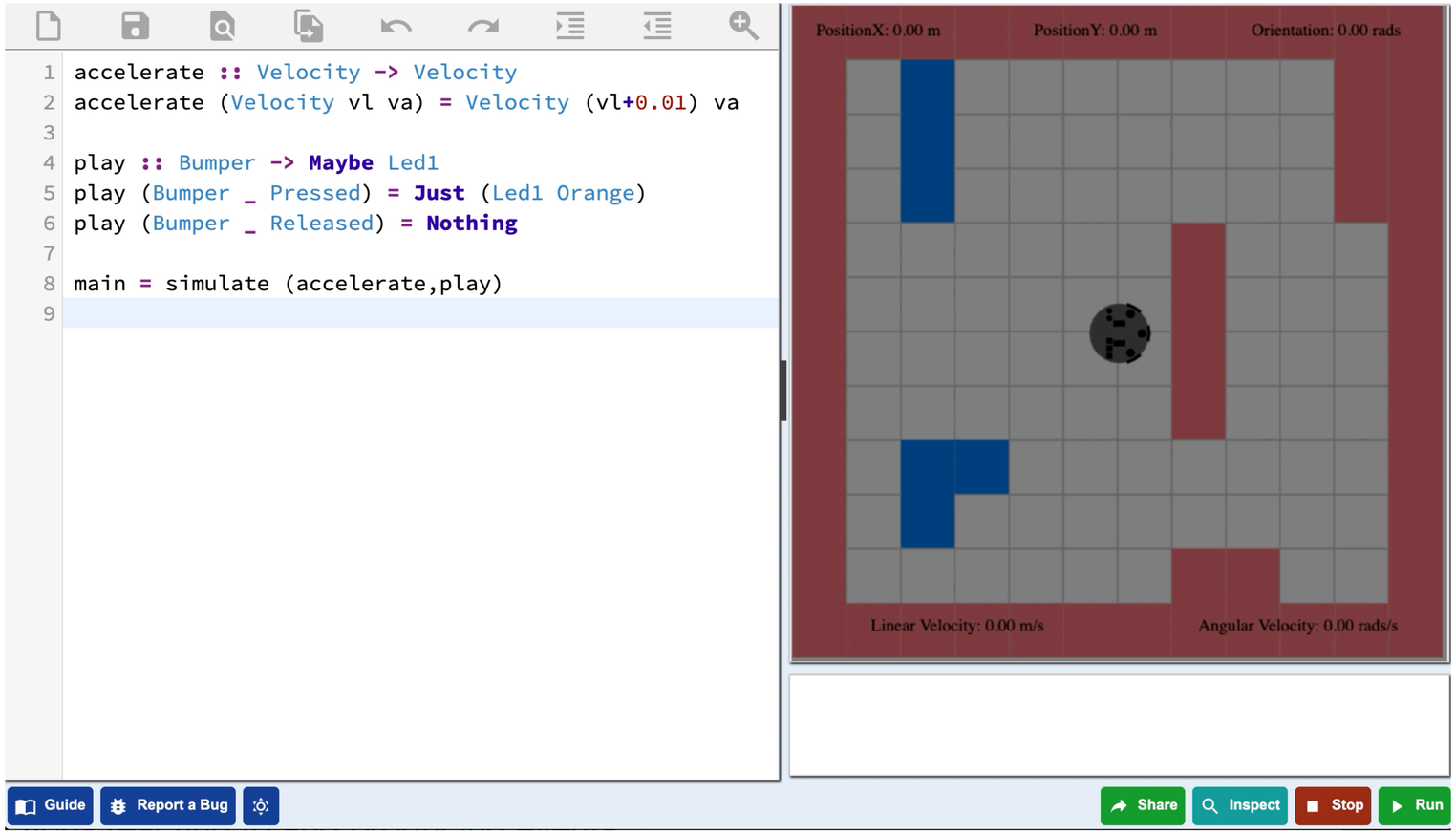
Basic proactive control



Constant functions for proactive nodes, abstract spin rate

ROSY by example

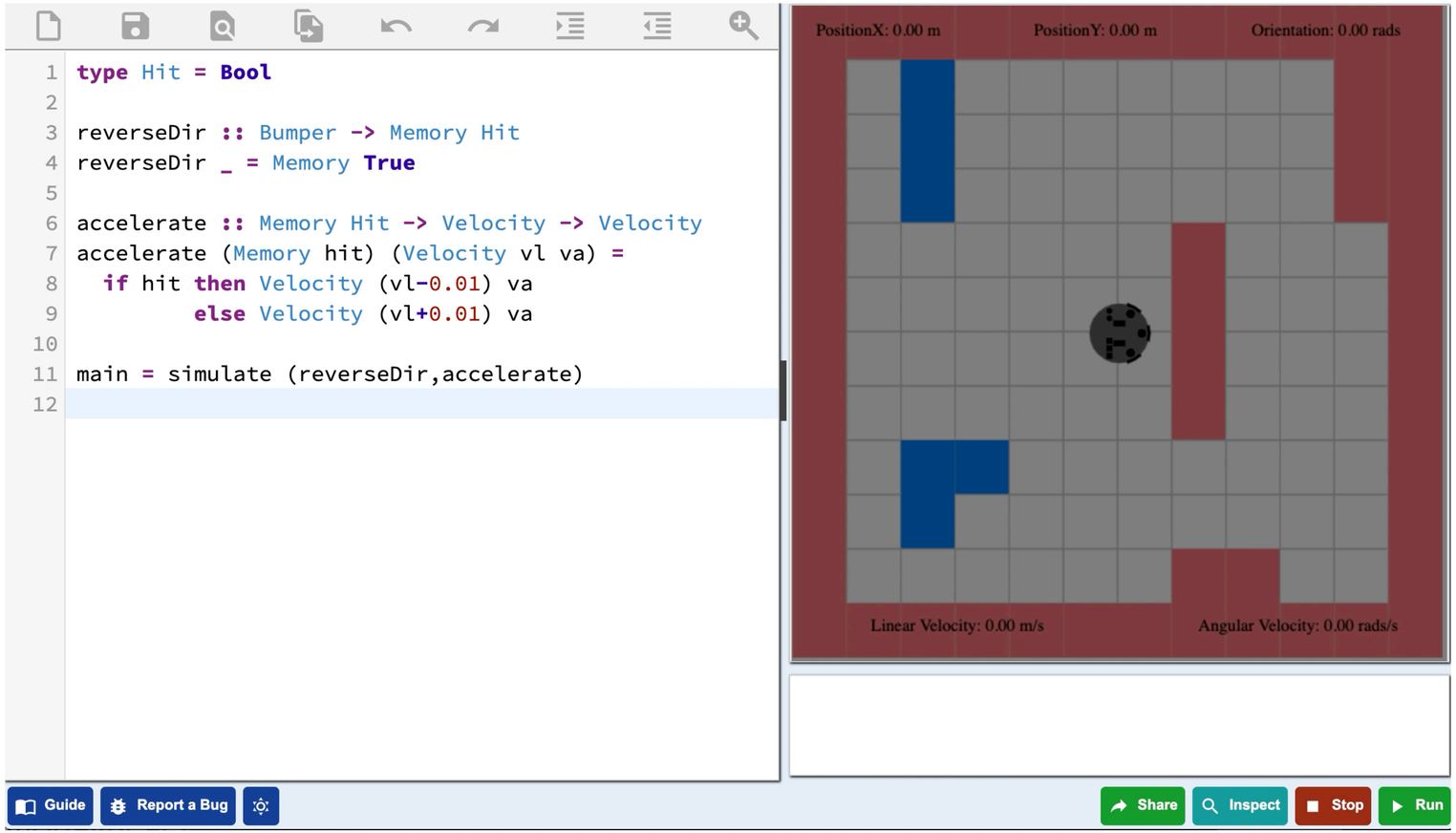
Modularity



Compositionality achieved through multiple functions, event generation may be optional

ROSY by example

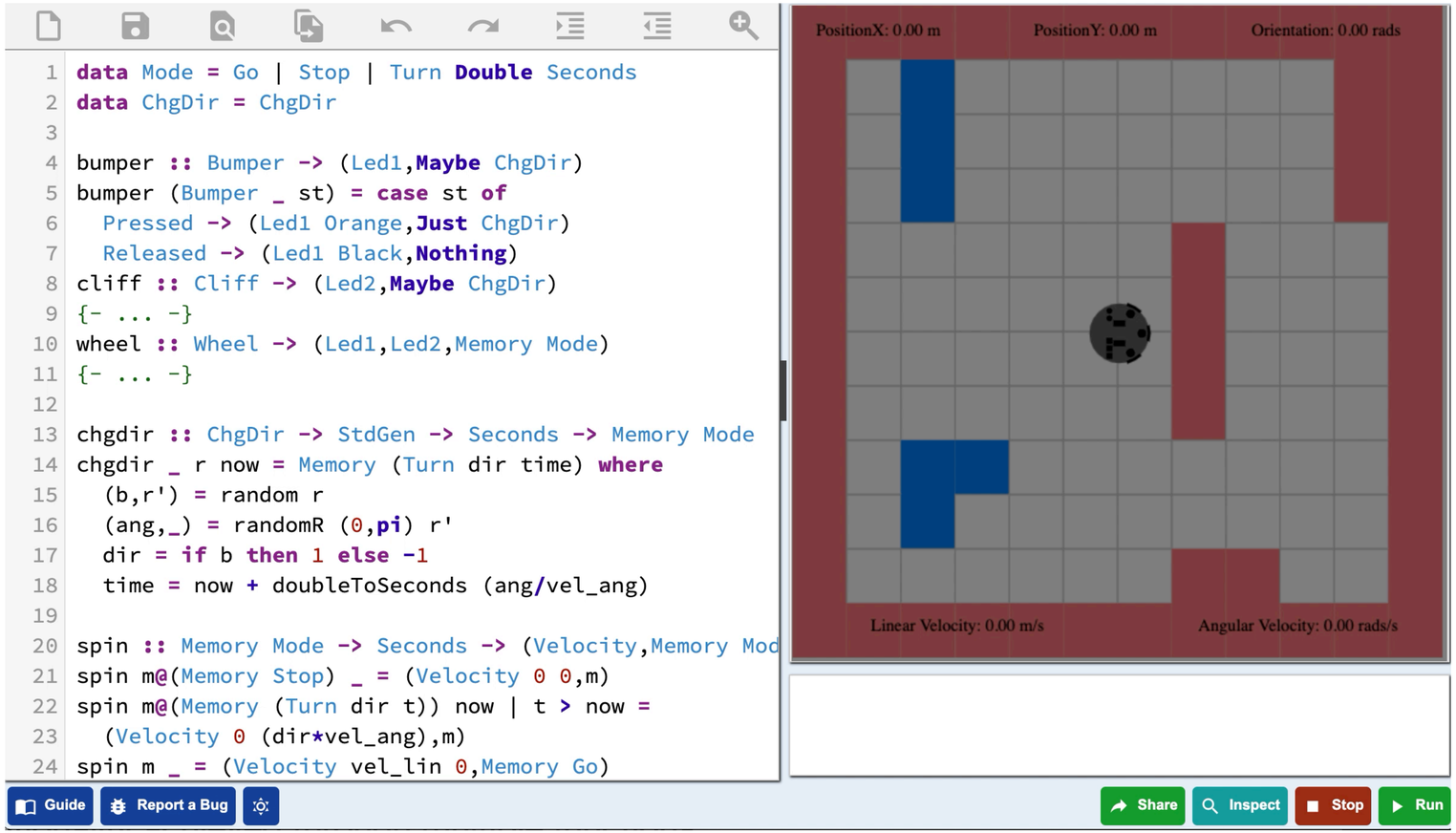
Stateful controllers



Global state explicitly modelled as events (reading and updating)

ROSY by example

A random walker



User may define events, functions may handle multiple events (implicit event merging)

ROSY by example

Multiplexing events

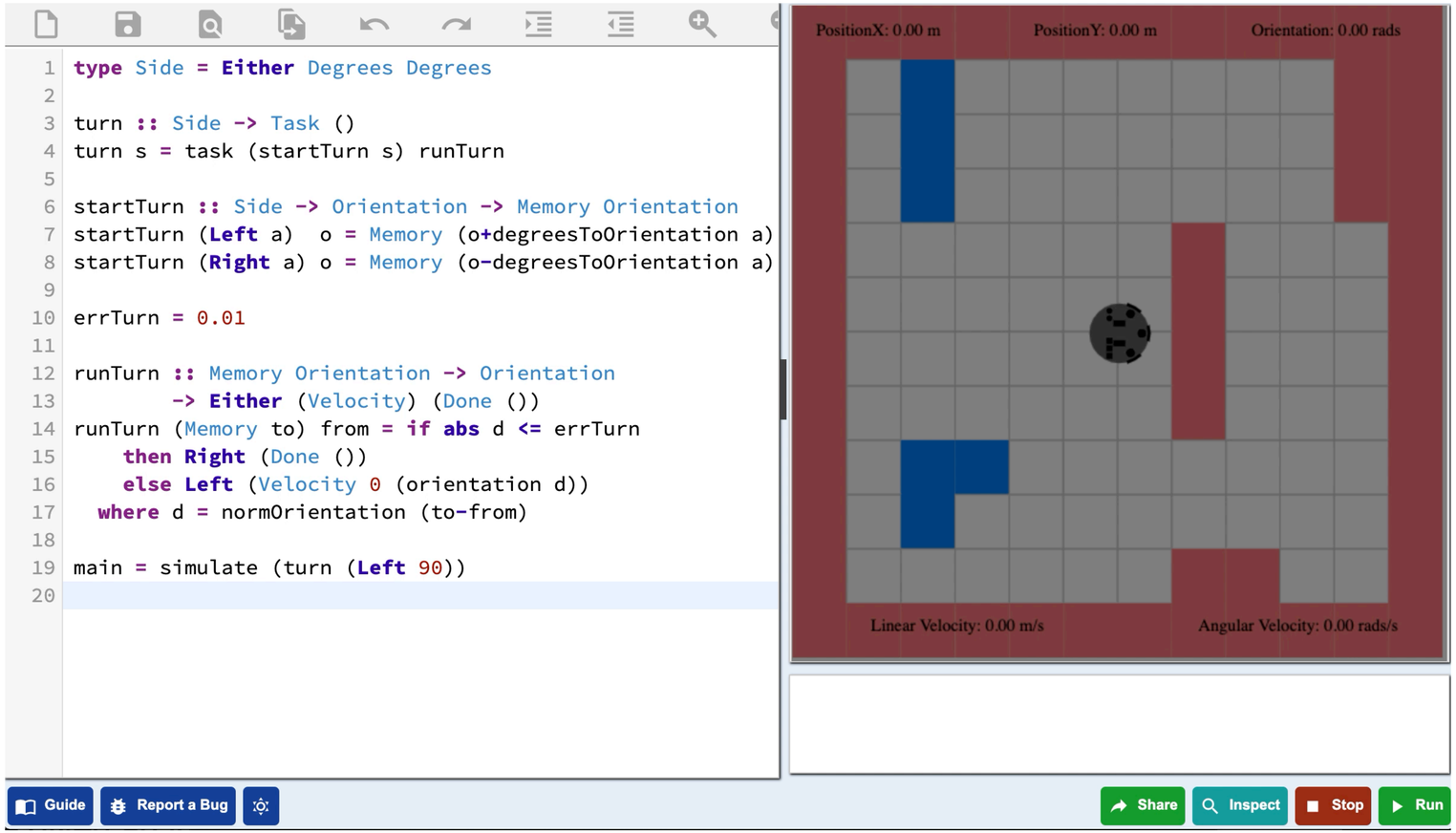
```
4  bumper :: Bumper -> (Led1,Maybe ChgDir)
5  {- ... -}
6  cliff :: Cliff -> (Led2,Maybe ChgDir)
7  {- ... -}
8  wheel :: Wheel -> (Led1,Led2,Memory Mode)
9  {- ... -}
10 chgdir :: ChgDir -> StdGen -> Seconds -> Memory Mode
11 {- ... -}
12 spin :: Memory Mode -> Seconds
13      -> (M2 Velocity,Memory Mode)
14 {- ... -}
15
16 safetyControl :: Either (Either Bumper Cliff) Wheel
17                 -> Maybe (M1 Velocity)
18 {- ... -}
19
20 data M = Start | Ignore Seconds
21 data M1 a = M1 a
22 data M2 b = M2 b
23
24 mux :: Seconds -> Memory M
25     -> Either (M1 Velocity) (M2 Velocity)
26     -> Maybe (Velocity,Memory M)
27 mux t _ (Left (M1 a)) = Just (a,Memory (Ignore (t+0.5)))
28 mux t (Memory (Ignore s)) (Right (M2 a)) | s>t = Nothing
29 mux t _ (Right (M2 a)) = Just (a,Memory Start)
```

Guide | Report a Bug | Share | Inspect | Stop | Run

Functions may also react to alternative events (implicit events merging)

ROSY by example

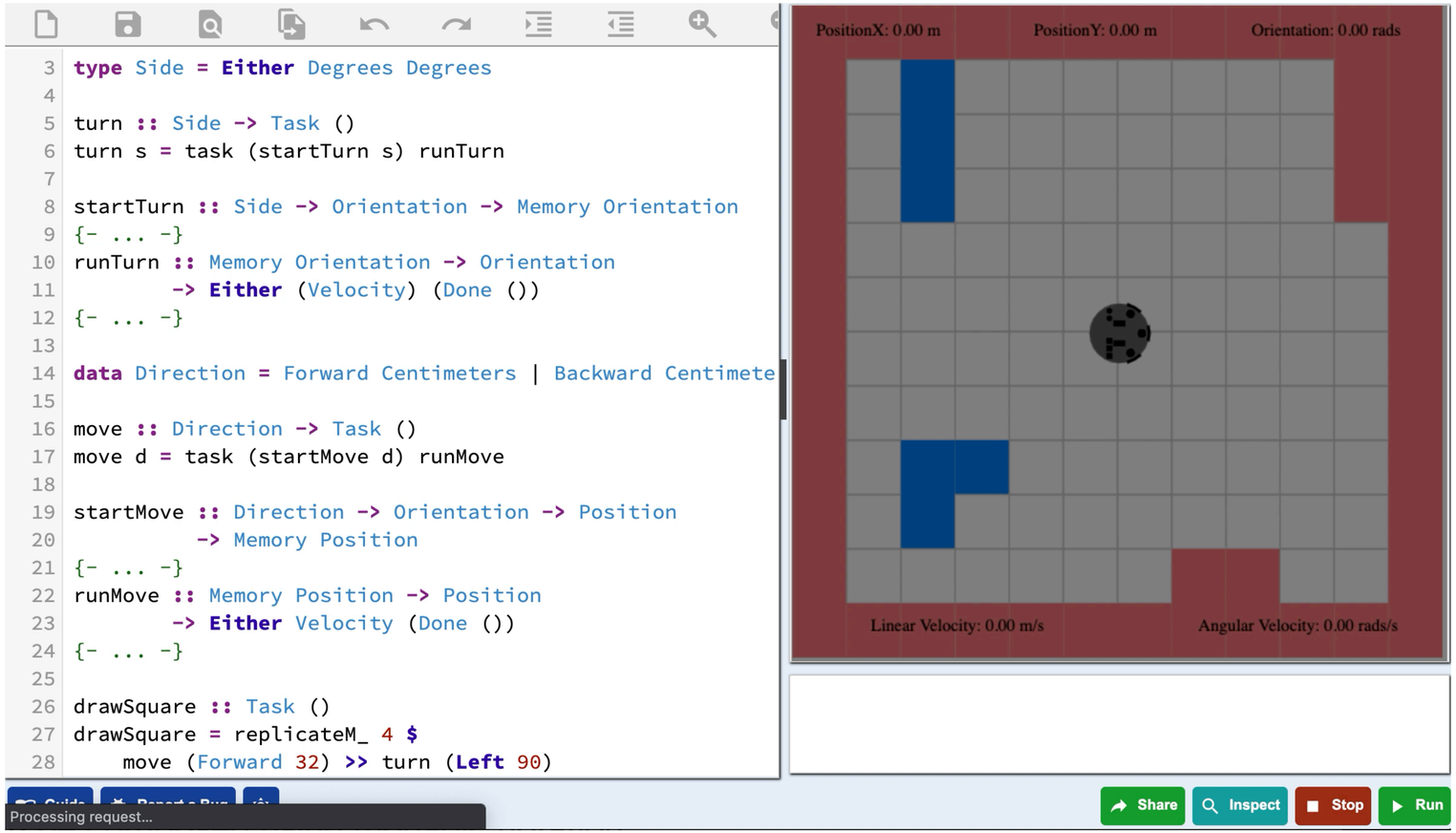
Task management



Tasks are given an initial value and a function executing until “done”

ROSY by example

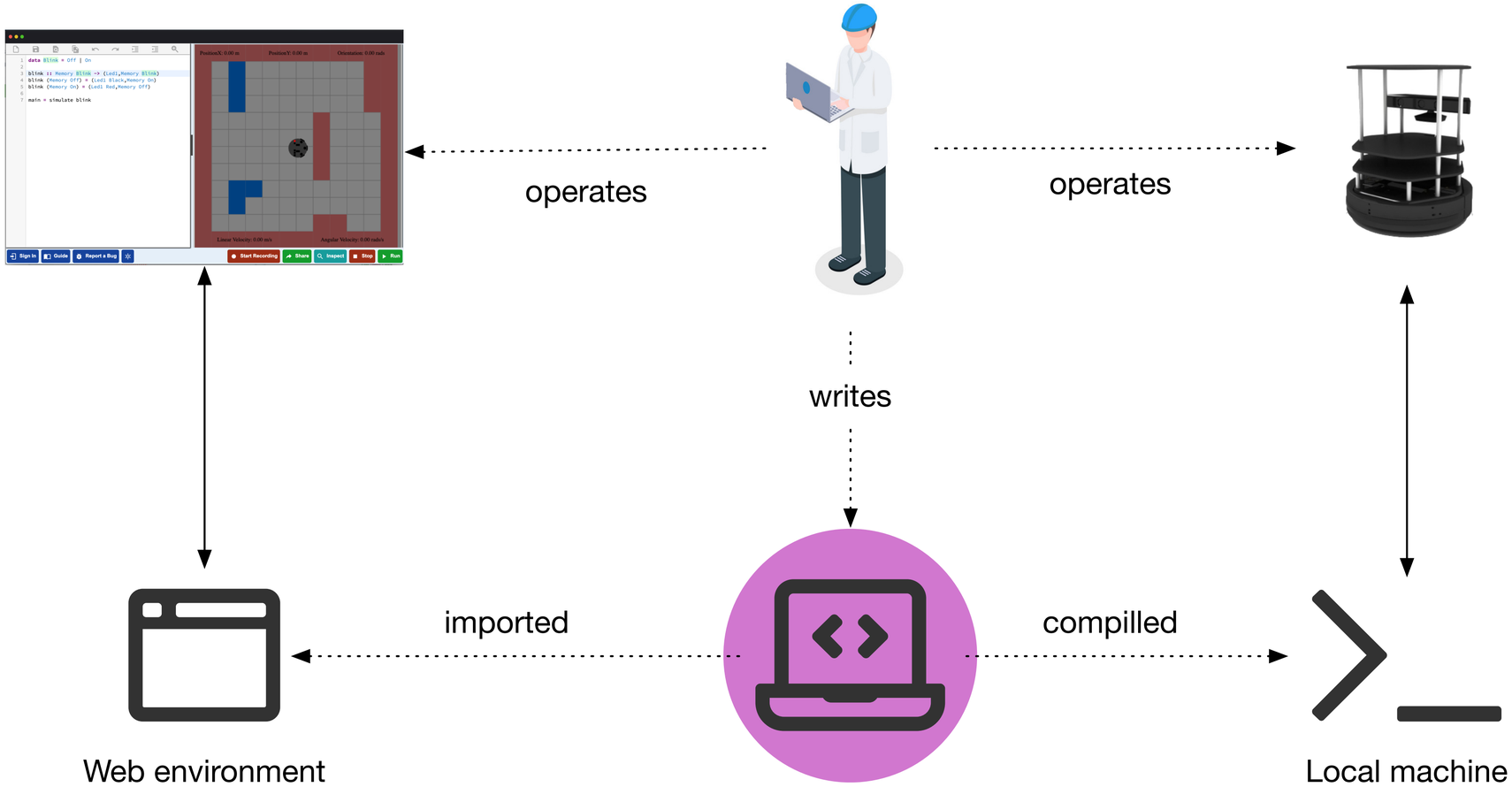
Task composition



Tasks can be composed using Haskell’s imperative notation

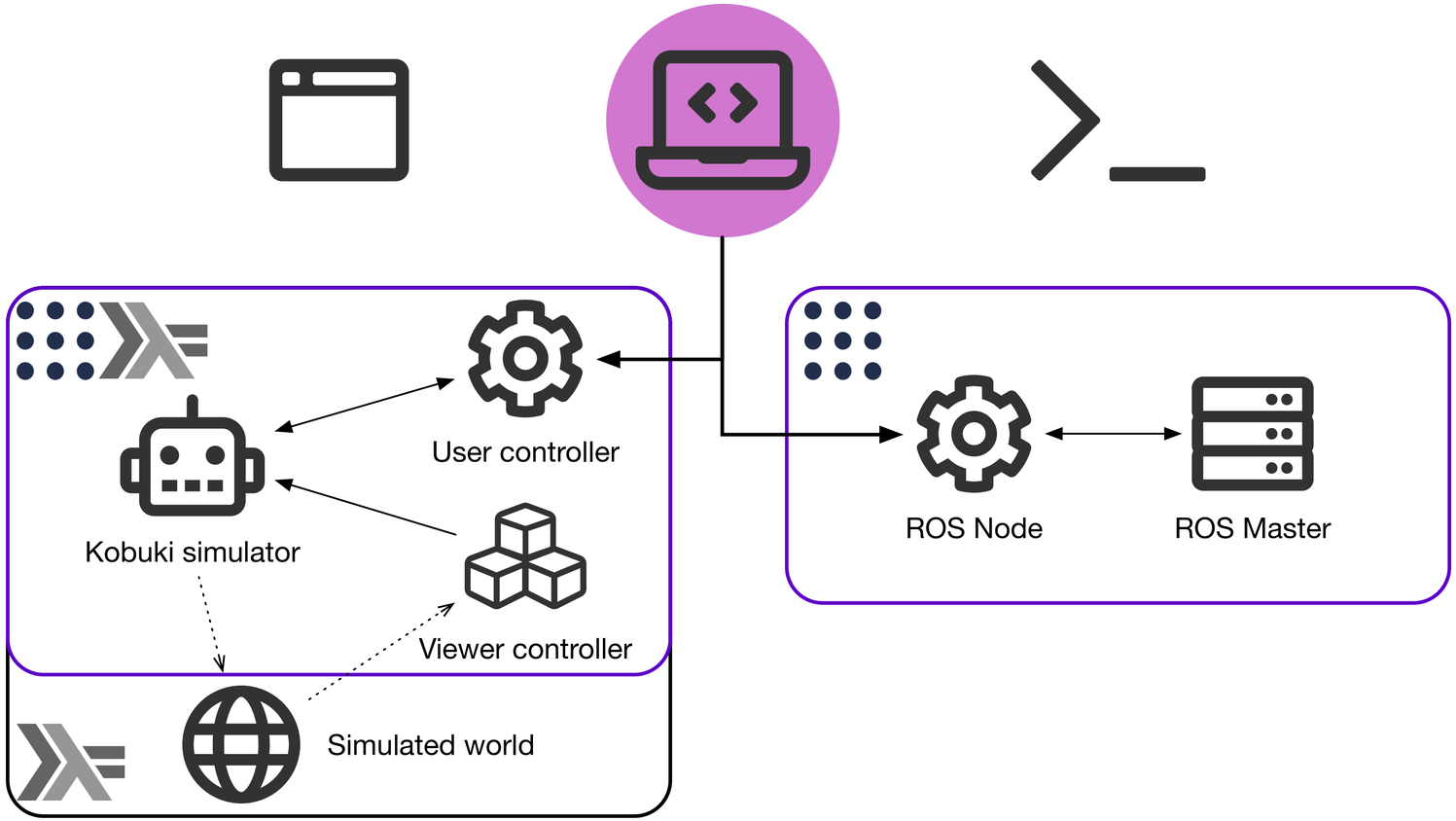
ROSY environment

Overview



ROSY environment

A peek under the hood



Final remarks

- Pedagogical robot programming language
- Sweet-spot between relevant computational concepts and simplicity
- Used in a summer camp for K12 students with no programming background
- Future work
  - Empirical evaluation
  - More advanced simulation environment
  - Explore blending with block-based languages

ROSY

An elegant language to teach the pure reactive nature of robot programming

Hugo Pacheco (FCUP & INESC TEC) and Nuno Macedo (FEUP & INESC TEC)
IRC'20