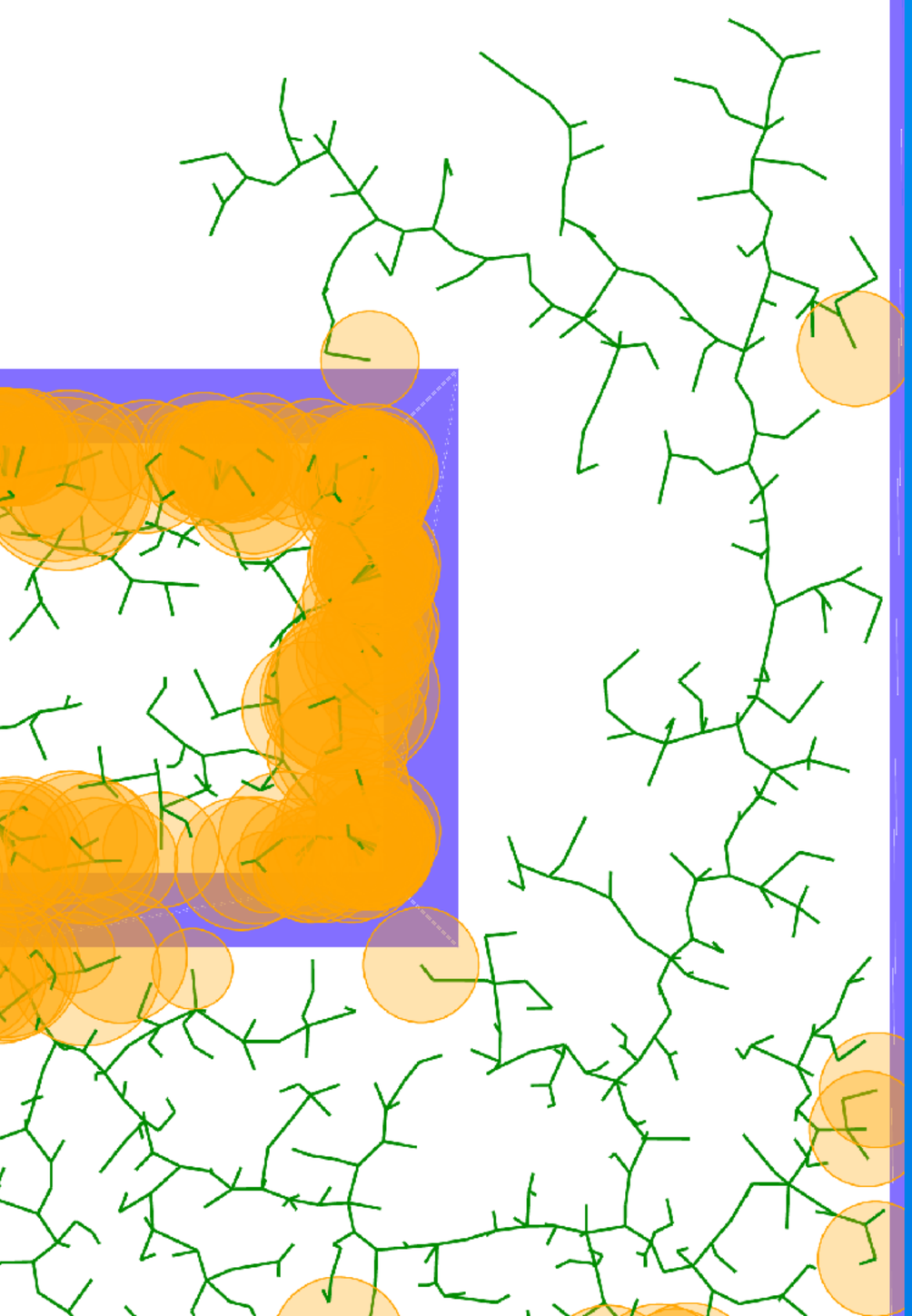


MOTION PLANNING USING OMPL

Sumeet Gyanchandani
Thomas Tiotto
Bart van der Vecht

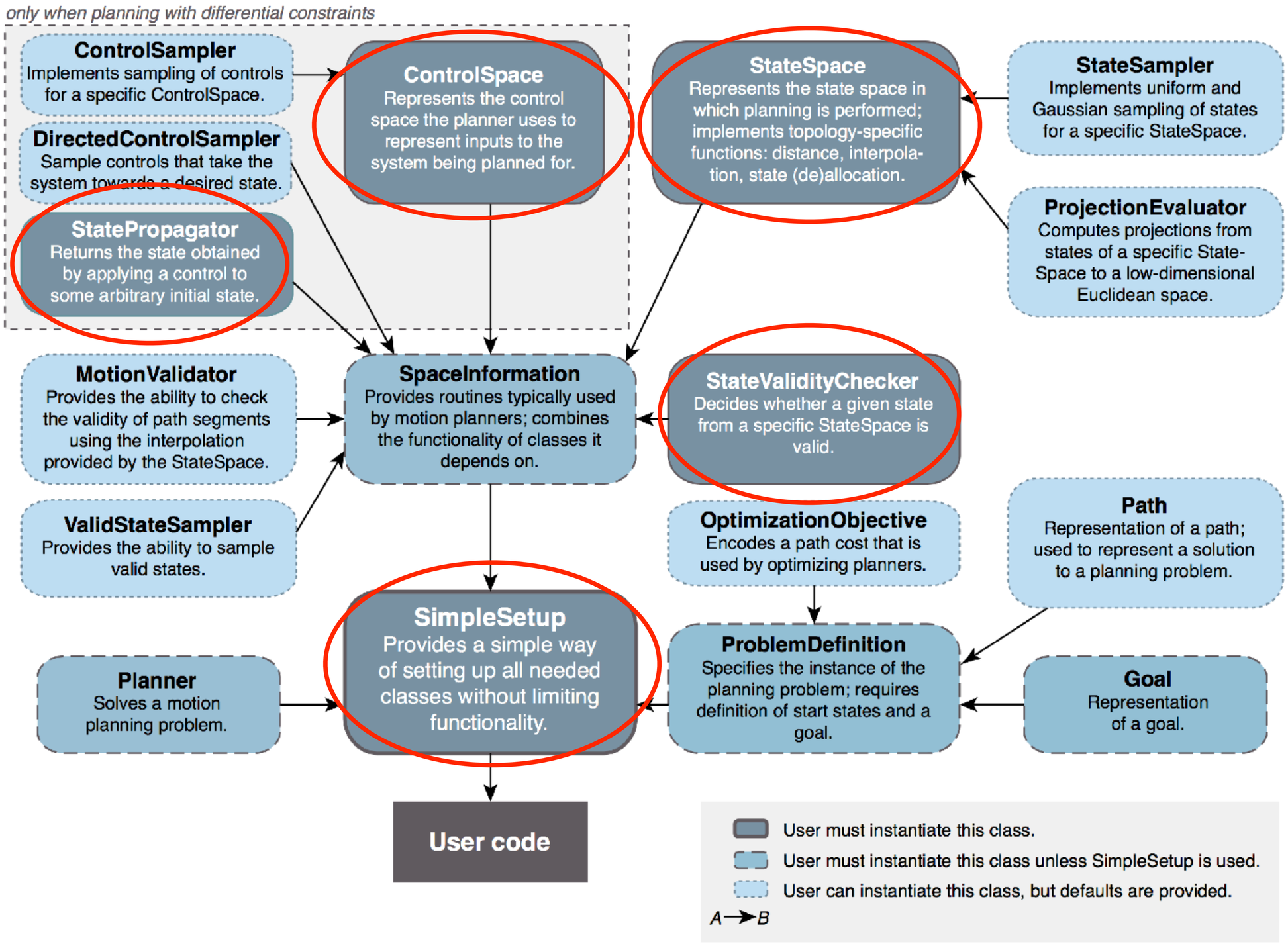
INTRODUCTION

BROKEN BICYCLE
MIDTERM PROBLEM 1.1

BROKEN THYMIO

The OMPL API attempts to abstract the functionality by supplying various wrapper classes, with the most high-level being SimpleSetup.

We were only required to supply the state space, state propagator, control pace, state validity checker and start and goal states.



setControlSpaceBounds

The bounds on the control space are where the kinematic models for the bicycle and differential drive robots are implemented.

Bicycle:

0° and +30° steering lock

Thymio:

±10 for the right wheel and between −10 and 0 for the left wheel.

manyObstacles and cornerObstacle

They must return false whenever the sampler selects a state that violates one of the obstacle constraints.

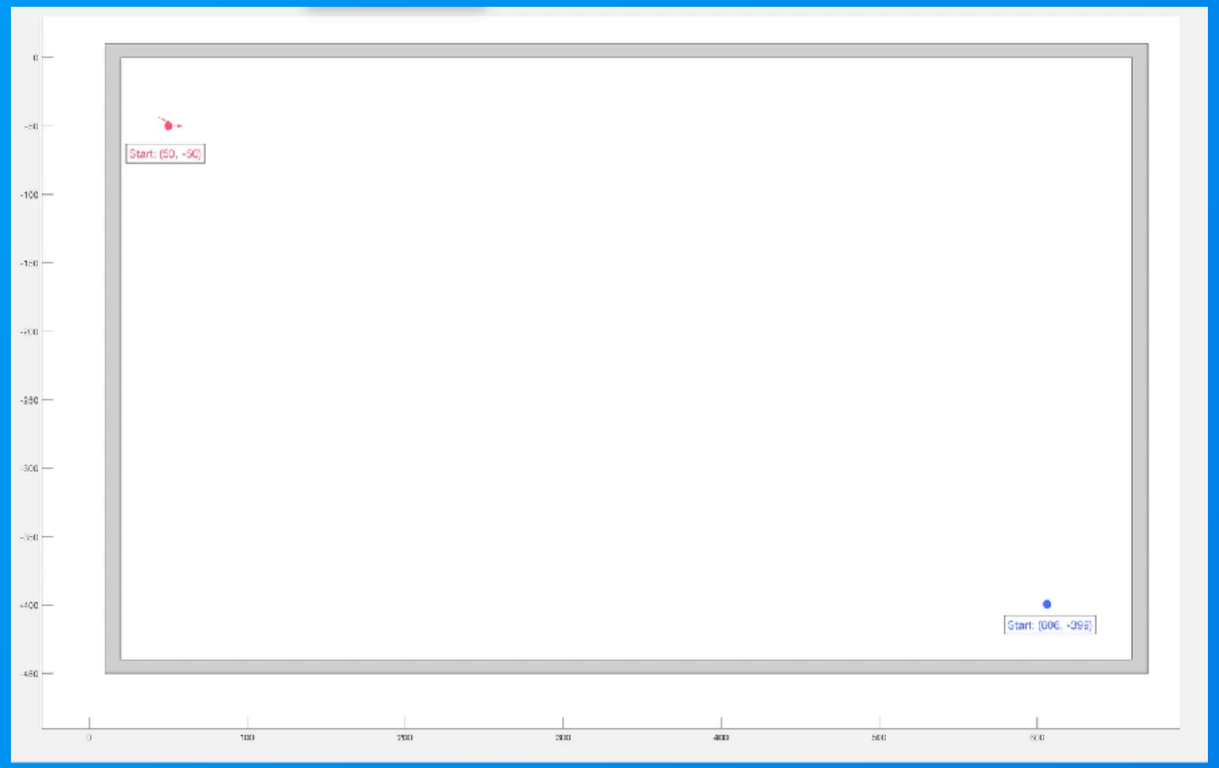
bicycleMovements

This function defines the control model for our bicycle robot. The control space consists of 2 dimensions: one for linear speed, and one for the steering angle.

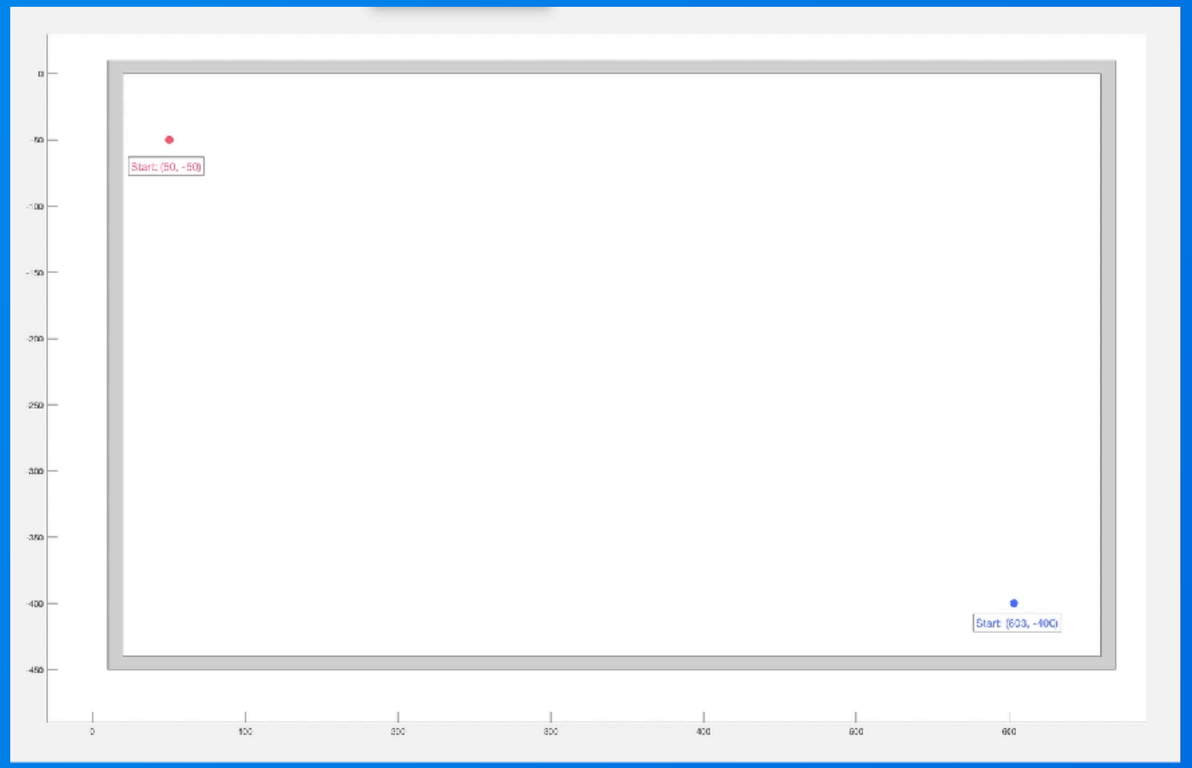
thymioMovements

This method defines the control model for the thymio robot. The control space consists of 2 dimensions, one which defines the linear speed for the left wheel, and one dimension for the other.

NO OBSTACLES

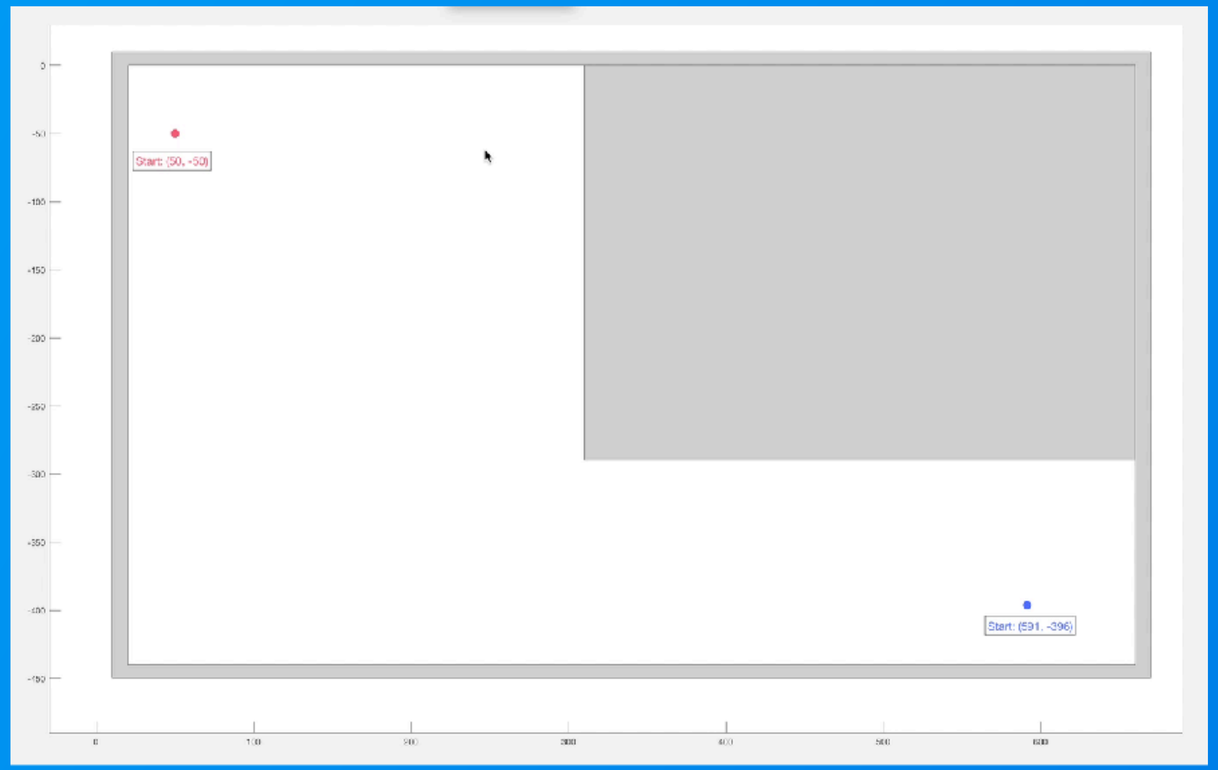


Bicycle

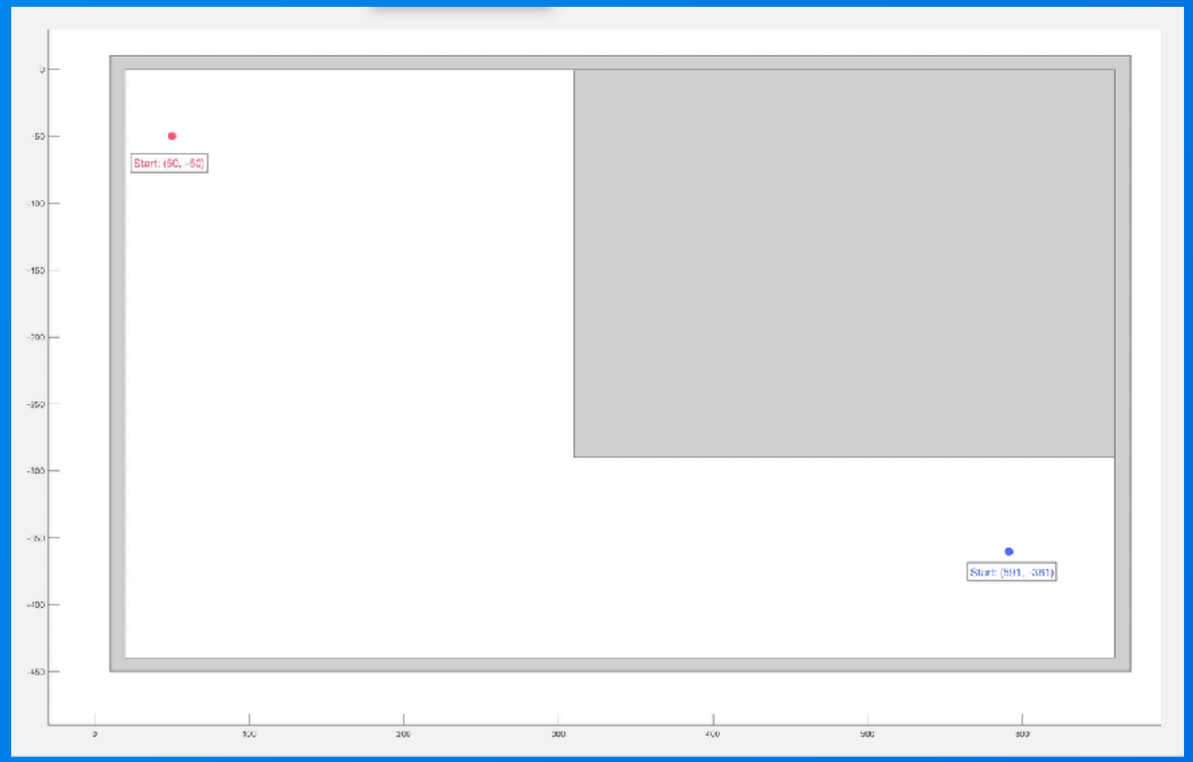


Thymio

ONE HUGE OBSTACLE IN THE CORNER

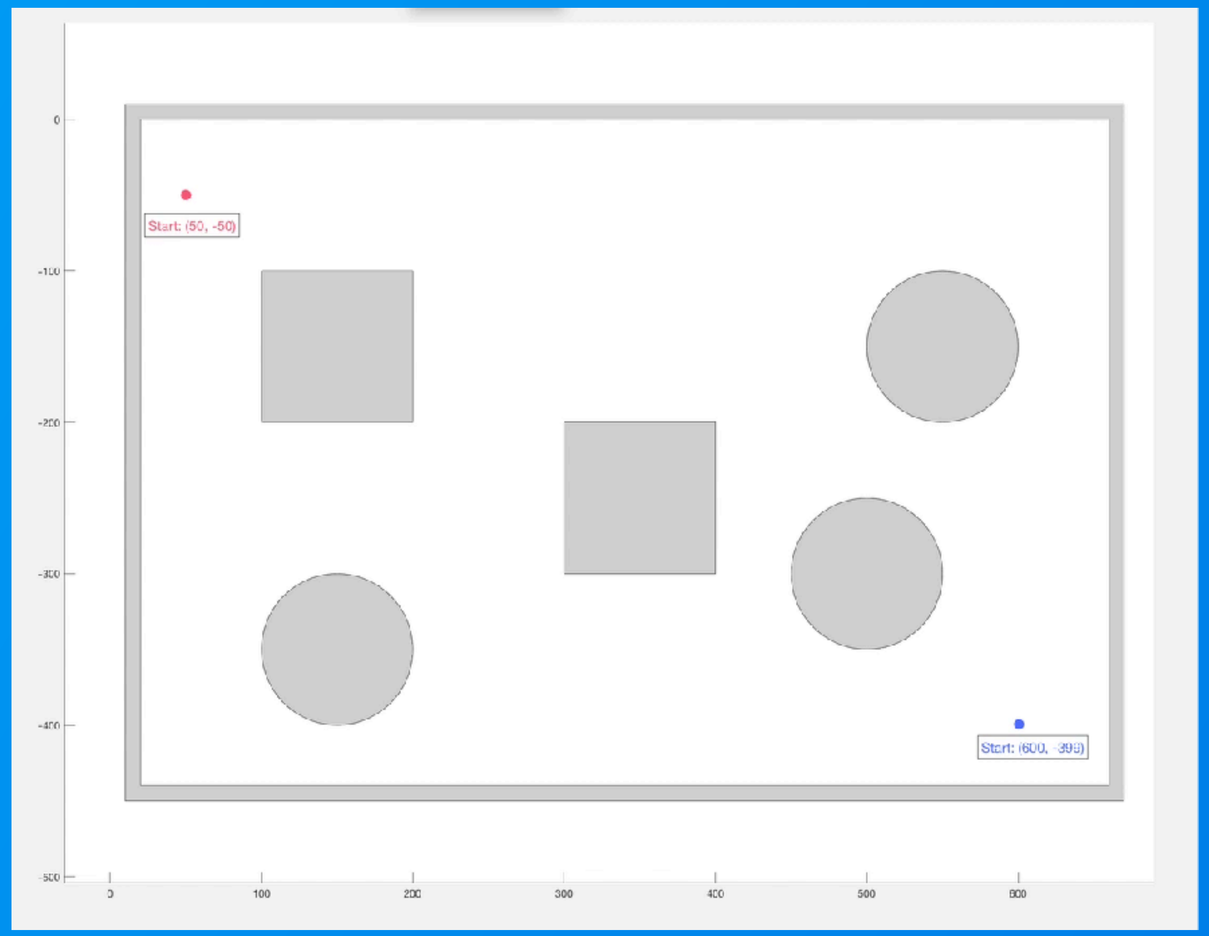


Bicycle

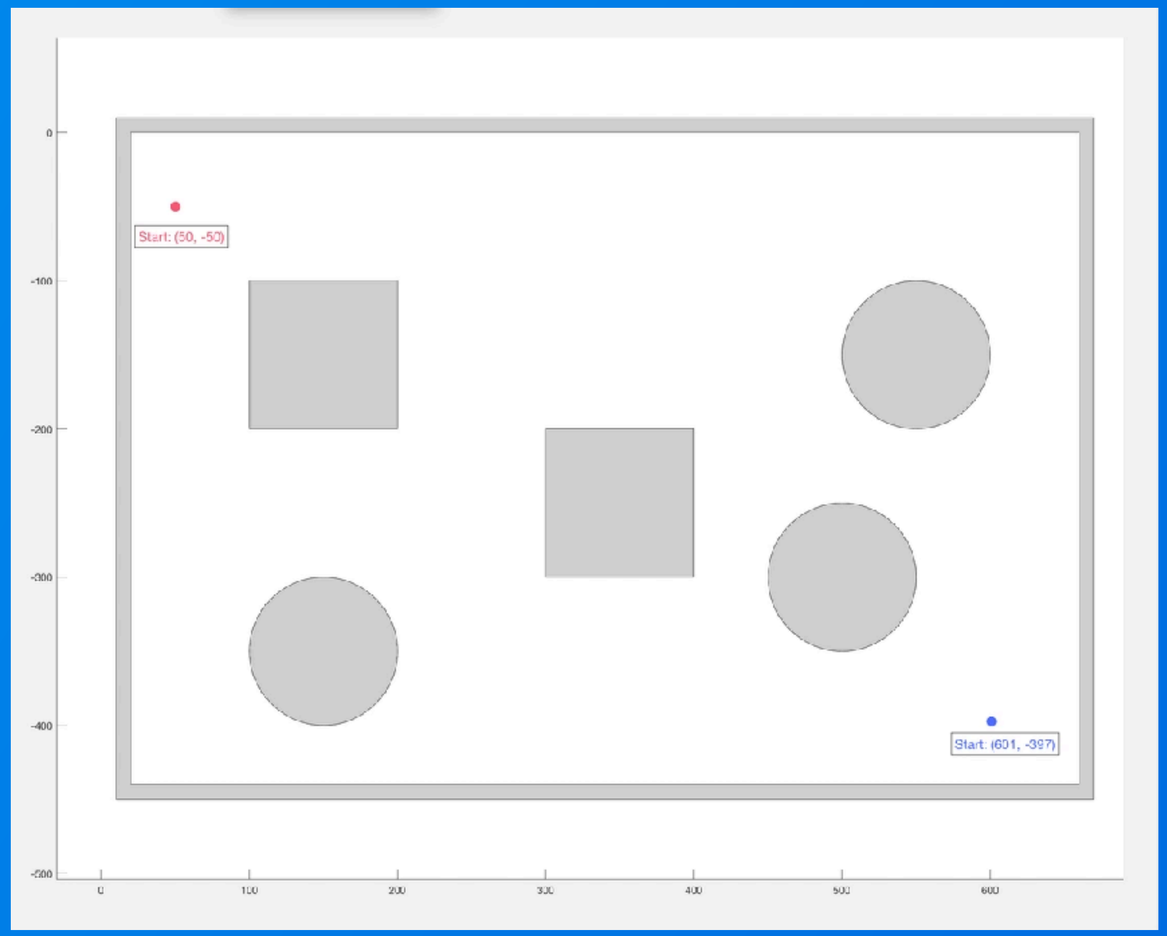


Thymio

MANY SMALL OBSTACLES | RRT

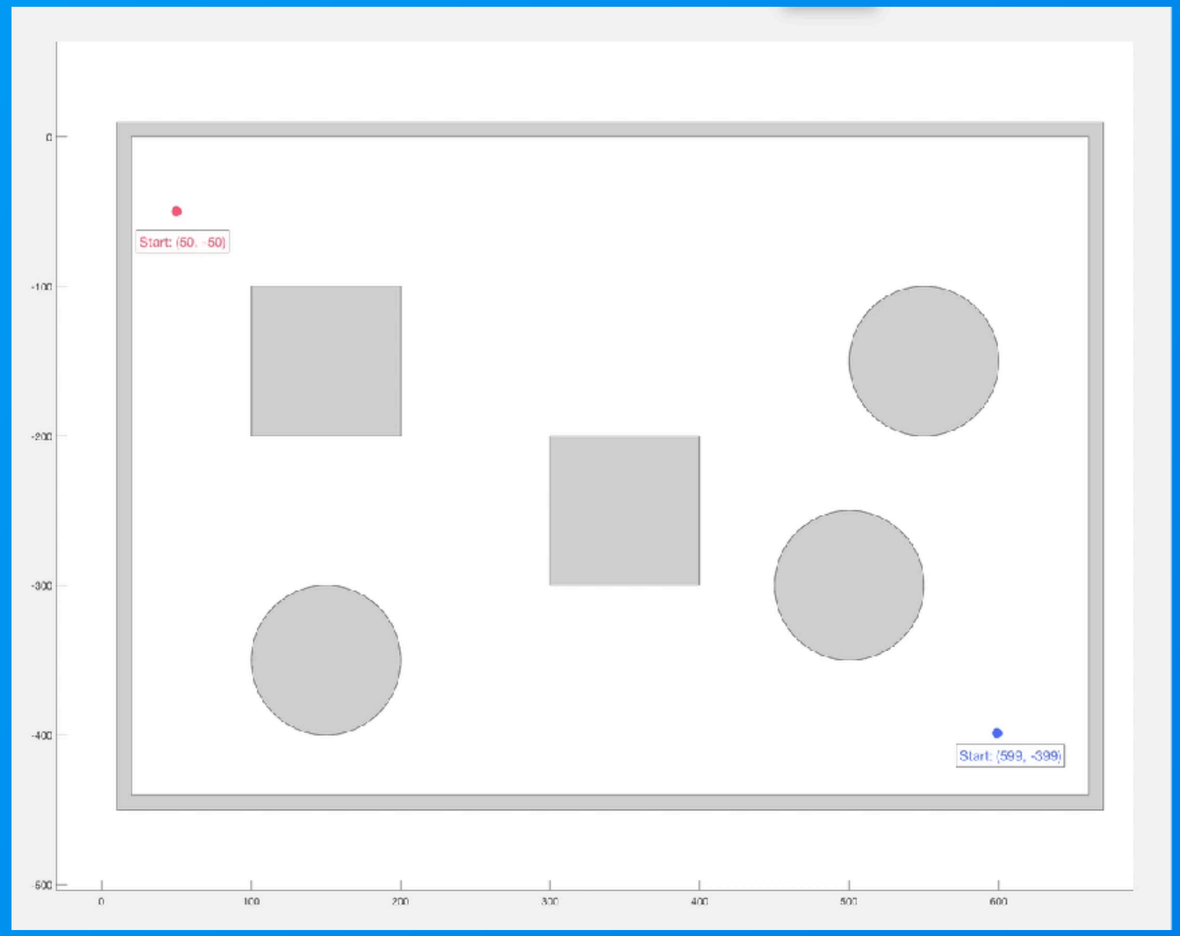


Bicycle

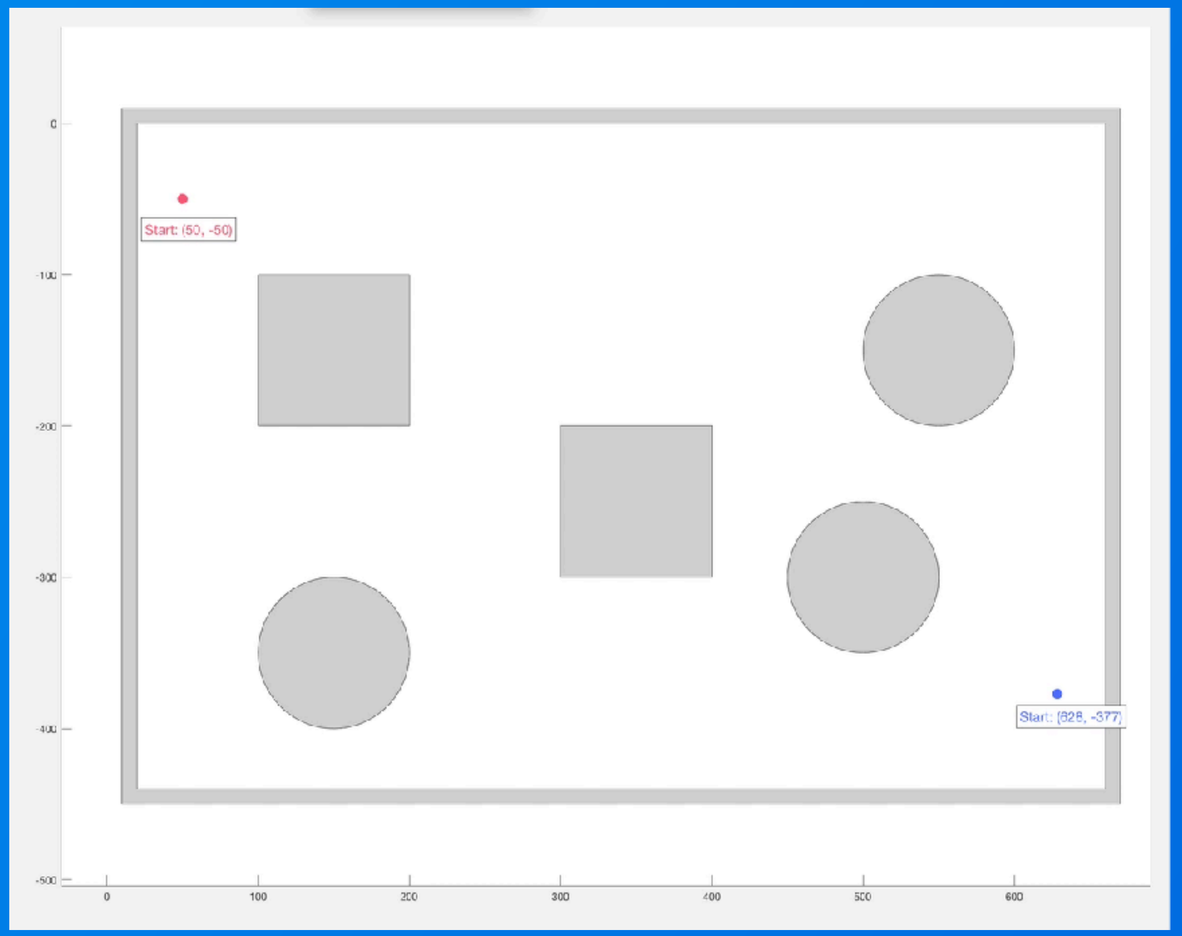


Thymio

MANY SMALL OBSTACLES I K-PIECE



Bicycle



Thymio

ISSUES

TUNING THE PARAMETERS

- resolution
- timestep
- planning time

MOST CONSTRAINED BICYCLE DOESN'T REACH GOAL

CUTTING CORNERS

some instances of our testing showed that the path sometimes cut a corner off an obstacle

CONCLUSIONS

PROS

- very powerful framework
- highly modular

CONS

- insufficient documentation of the API
- parameter tuning is time consuming
- hard to work without visual references