Optional Visualization

Bookmark this page

Visualizing Robots

Note: This part is optional. It is cool and very easy to do, and may also be useful for debugging. Be sure to comment out all visualization parts of your code before submitting.

We've provided some code to generate animations of your robots as they go about cleaning a room. These animations can also help you debug your simulation by helping you to visually determine when things are going wrong.

Here's how to run the visualization:

1. In your simulation, at the beginning of a trial, insert the following code to start an animation:

```
anim = ps2_visualize.RobotVisualization(num_robots, width, height)
```

   (Pass in parameters appropriate to the trial, of course.) This will open a new window to display the animation and draw a picture of the room.

2. Then, during *each time-step*, before the robot(s) move, insert the following code to draw a new frame of the animation:

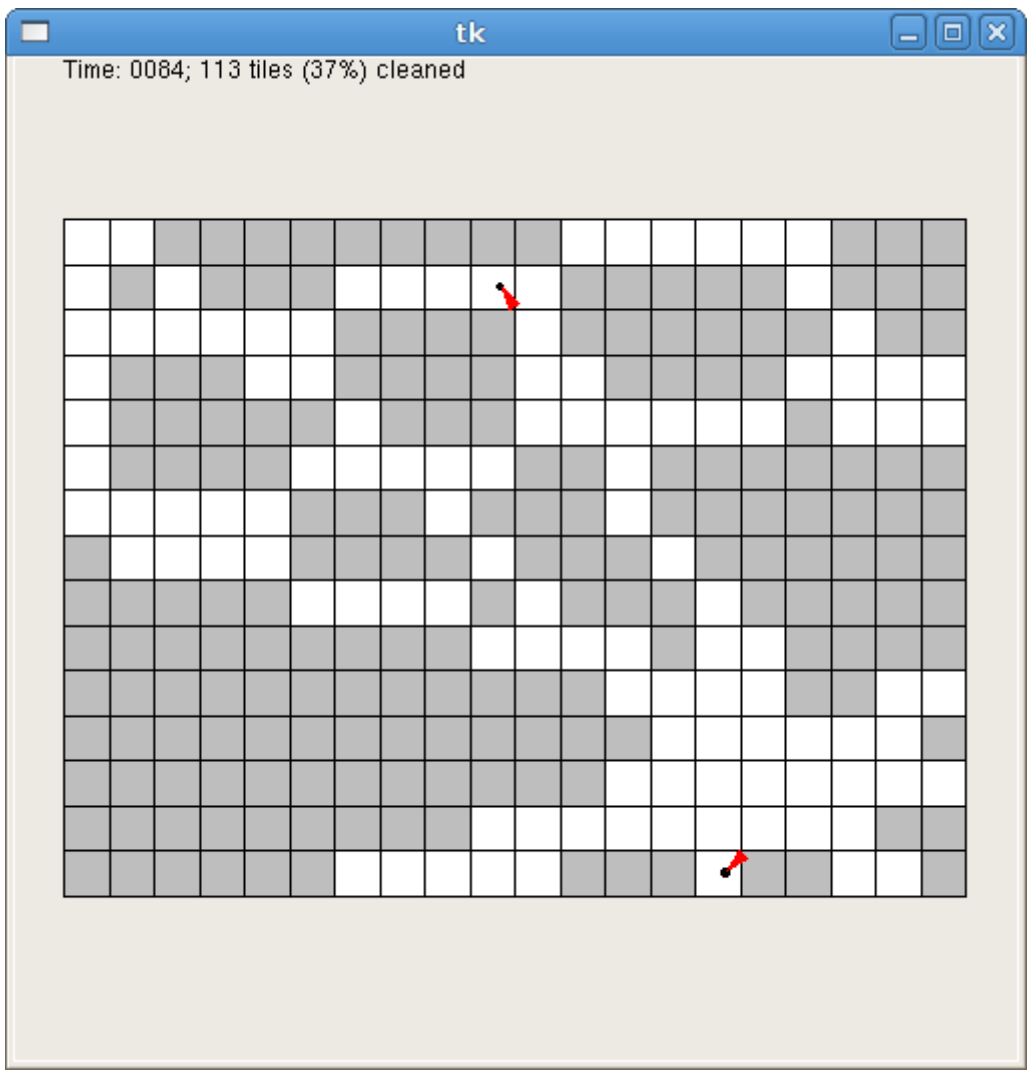
```
anim.update(room, robots)
```

   where `room` is a `RectangularRoom` object and `robots` is a list of `Robot` objects representing the current state of the room and the robots in the room.

3. When the trial is over, call the following method:

```
anim.done()
```

The resulting animation will look like this:



The visualization code slows down your simulation so that the animation doesn't zip b ult, it shows 5 time-steps every second). Naturally, you will want to avoid running the anima e trying

to run many tri

‹ Previous p Next ›

For purposes of debugging your simulation, you can slow down the animation even further. You can do this by changing the call to `RobotVisualization`, as follows:

```
anim = ps2_visualize.RobotVisualization(num_robots, width, height, delay)
```