Courseware   Updates & News   Calendar   Wiki   Discussion   Progress

Help

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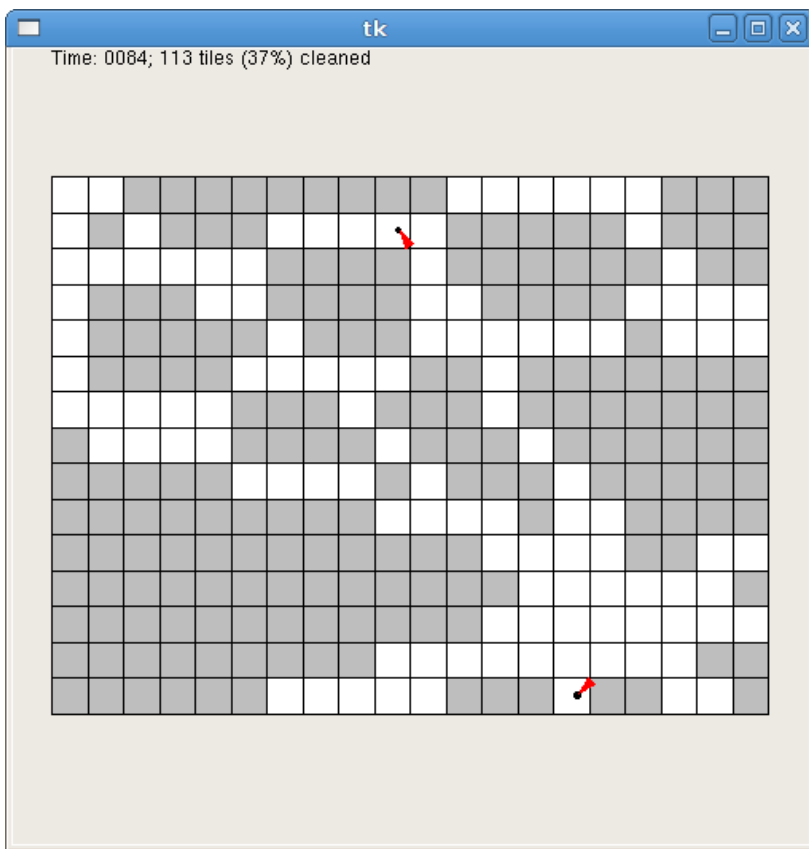
```
anim.update(room, robots)
```

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:

tk

Time: 0084; 113 tiles (37%) cleaned

The visualization code slows down your simulation so that the animation doesn't zip by too fast (by default, it shows 5 time-steps every second). Naturally, you will want to avoid running the animation code if you are trying to run many trials at once (for example, when you are running the full simulation).

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)
```

The parameter `delay` specifies how many seconds the program should pause between frames. The default is 0.2 (that is, 5 frames per second). You can increase this value to make the animation slower or decrease it (0.01 is reasonable) to see many robots cleaning the room at a faster frame rate.

For problem 5, we will make calls to `runSimulation()` to get simulation data and plot it. However, you don't want the visualization getting in the way. If you choose to do this visualization exercise, before you get started on problem 5 (*and* before you submit your code in submission boxes), **make sure to comment the visualization code out of runSimulation().**

Show Discussion

New Post

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+