Simulate control of an iRobot Create in an interactive 3D graphical simulator.

Simulators help
Make robotics programming more accessible to introductory programmers (Fagin & Merkle, 2002) Allow students to program wherever and whenever they like But they can be complex to install and use

Make it useful
Developed an enticing 3D environment Modeled the Create’s: actuators (wheels, speaker, LEDs) sensors (cliff, bumpers, odometry, buttons) Runs programs written in Python

Make it simple to learn
Create robots are a simple and cheap, yet high quality platform Python and the pycreate library (Dodds, 2007) are easy to learn We wanted our simulator to be easy to install and use!

It helped our students
25 of 32 students (78%) this Spring found it beneficial “Very helpful when working because we didn’t need to always come to the classroom to get work done.” “I thought the simulator was excellent. I didn’t have to compete with 10 other people to use the giant black circle thing (track for line following).” “Implement realistic sensors and surfaces—they threw me off when trying to implement a line-follower code. It worked on the sim, but not with the robot.”

Capitalize on limitations
Windows-based PCs only (but simple to install) Fixed lighting (but forces issue that thresholds should be manually tuned or learned from data, rather than hardcoded) Sensors aren’t exact (so it’s good for early development only and doesn’t replace the robot) Currently only works with Python (but can be extended)

How it works
1. The student writes her program, using a library to interface to the Create (pyCreate).
2. The library opens a network socket and streams the raw opcodes to the simulator.
3. The simulator retrieves the opcodes and interprets the commands.
4. The interpreted commands are visualized through a 3D rendering.

Download it from http://www.rose-hulman.edu/~boutell/simulator