Learning to Fail

Sarah Moss
Dordt College, sarah.moss@dordt.edu

Follow this and additional works at: https://digitalcollections.dordt.edu/voice
Part of the Christianity Commons, and the Higher Education Commons

Recommended Citation
Available at: https://digitalcollections.dordt.edu/voice/vol63/iss2/3
LEARNING TO FAIL

Computer Science Professor Kari Sandouka wants her students to experience failure as they learn about information system design.

That was one reason why she decided to have her students build and program Dexter GoPiGo robots in her information system design course last fall.

“I wanted students to think about how they can adjust when things break down unexpectedly,” says Sandouka.

Sandouka gave her class five Raspberry Pi robot car kits and split the class into four teams of three people. Each team had the freedom to decide what they wanted their robots to do—navigate an obstacle course, take pictures of other objects, or follow a line. The students could adapt code Sandouka gave them or write their own.

“Having a working product was extra credit,” says Sandouka. “I wanted the students to take something that wasn’t theirs—that they hadn’t built entirely by hand—and adapt it to make it work.”

Failure was part of the process; three of the robots encountered coding problems or sensor issues, and one team’s robot did not work at all. Sandouka thinks that failing in the classroom will help students understand how to respond to questions or problems in the workplace.

“I tell the students that people will come to them and say, ‘This isn’t working.’ They’ll need to be the expert and determine how to figure out how to solve that problem,” says Sandouka. “There’s a customer service aspect of understanding how to troubleshoot and figure out problems without getting frustrated.”

Group work was also an important part of the learning process for the students.

“A few students would have liked to just do the work on their own,” says Sandouka. “But a lot of businesses now have a bull pit area rather than separate cubicles for workers. People are collaborating in the workplace, so students need to understand how to work with all kinds of people.”

Sandouka had each team designate one person to handle coding, one to deal with documentation, and one to lead the final presentation. Having to take a robot from defunct to functioning and talk about that process with their classmates sometimes pushed students outside their comfort zones.

Sandouka’s students appreciated the challenge. One team programmed their robot to go through an obstacle course. Dylan Vander Berg, a junior computer science and actuarial science double major, wrote most of the robot’s code from scratch. It took his team nearly eight hours to build and program their robot.

“All my programming experience has been straight software, so it was cool to program a robot and see it do things in the real world,” says Vander Berg.

“I wish we could have had a little more time to see all we could do with the robots,” says Dan Kelly, a sophomore computer science major who led the team presentation. “You can do some pretty cool stuff with them.”

Sandouka plans to have her students spend even more time with the robots in the future.

“Often coders just want to code,” says Sandouka. “They don’t think about the full life cycle of something. With this project they had to think about all the steps from start to finish.”

SARAH MOSS (10)