**Instructional Days:** 19-23

**Topic Description:** Build, program and present a rescue robot.

**Objectives:**

Students will be able to:

* Build and program a robot that uses input and output devices to count simulated people by following a black line and counting “people” on the path.

**Outline of the Lesson:**

* Explanation of project guidelines and floor (15 minutes)
* Design, build, program robot (195 minutes)
* Rescue Robot challenge (50 minutes)
* Reflection and cleanup (15 minutes)

**Student Activities:**

* Brainstorm how to build and program the robot.
* Build the robot.
* Write a program in Robot Educator software.
* Test the robot frequently and refine program and hardware.
* Participate in rescue challenge.
* Complete project reflection. Take robots apart and put materials away.

**Teaching/Learning Strategies:**

* Hand out requirements and rubric. Explain guidelines and answer questions. Show students the arena with the victims laid out. Explain that they must use sensors so that the robot will follow the black line and will sense when it has encountered a victim or a gap.
* Circulate the room and make sure students are on task; answer questions as needed.
* During the rescue challenge, assign one student as timekeeper and one to keep track of victims found. Collect each group’s program as they compete and immediately assess the robot using the rubric, while the next group gets set up.
* At the end of the challenge, have each student complete the project reflection and submit it, then clean up the robots.

**Resources:**

* Rescue Robot Activity
* Rescue Robot Sample Rubric   
  Exploring Computer Science—Unit 6: Robotics 283
* Project Reflection
* Official RoboCup Jr Rescue Competition Rules (2008): http://rcj.robocup.org/rescue.html
* Instructions for building modules are available at   
  http://rcj.sci.brooklyn.cuny.edu/rcj2010/rescue\_suggestedbuildinginstructions.pdf . Alternatively use white butcher paper on the floor with black electrical tape as a path. Use green electrical tape to indicate victims.

**Rescue Robot Activity**

The rescue robot assignment is based on the second level of RoboCupJunior, an international competition. More information about RoboCupJunior is available at http://rcj.robocup.org. This robot simulates robots sent to rescue people during natural disasters. It must find “victims” along the path through each “room” and avoid obstacles. The goal is to program a robot that uses sensors to respond to different stimuli.

**Task:**

Build a robot that follows a black line on a white background, counts green or metallic “people” and avoids obstacles.

**Requirements:**

* The robot must follow the black line and attempt to complete the course through the entire arena. The robot will begin at the starting location in the doorway of the first “room”.
* The robot should stop and flash a light for at least two seconds to indicate it has found a victim. For extra credit, count the number of victims and display the count.
* The robot should be able to avoid items of debris blocking the black line.
* If a robot has been stuck or lost the black line for more than 20 seconds, the teacher may pick it up and put it back onto the black line a little beyond where it ran into problems. The 20-second rule allows it to try to find its way back to the line without intervention. A team may decide to quit if the robot is faulty or repeatedly loses the line.
* Robots must be controlled autonomously except for being started by a member of the team.
* The robot will have 10 minutes to complete the course and identify all victims.
* Each team must print out its program and hand it in at the same time that they compete.
* Fair play is an important part of the RoboCup challenge. Teams are expected to help other teams as needed and not deliberately interfere with or damage other teams’ work. All students are expected to respectfully watch all other teams compete.

**Process:**

6. Brainstorm ideas about how your robot should work: what sensors will you need? What motors and lights? What programming constructs will you need?

7. Start building your robot.

8. Build a program that controls the robot

9. Test frequently and revise the program. Make sure it correctly detects victims and that it can follow the line. Check if it can navigate gaps.

You will have three and a half class periods to build and program the robot; then you will present it in class.

Official Rules available http://rcj.robocup.org/rescue.html

**Official RoboCupJunior Rescue Challenge**

**5.1. Victims**:

5.1.1. Ten (10) points are awarded for each victim located by the robot. The robot indicates that it has found a victim by stopping and flashing a lamp for at least two (2) seconds.

5.1.2. Extra points are NOT awarded for the same victim being located more than once.

**5.2. Gaps in the black line:**

5.2.1. Ten (10) points are awarded for each gap in the black line that the robot successfully negotiates (i.e. recovers the line on the far side of the gap).

**5.3. Debris blocking the black line:**

5.3.1. Ten (10) points are awarded for each item of debris blocking the black line that the robot successfully avoids (i.e. moves around the debris and recovers the line).

**5.4. Rooms:**

5.4.1. Ten (10) points are awarded for each room that the robot navigates successfully (i.e. enters through one doorway and exits through the other doorway).

**5.5. Ramp:**

5.5.1. Thirty (30) points are awarded for the robot successfully negotiating a ramp without any assistance.

**5.6. Penalties:**

5.6.1. Two (2) points are deducted for each false victim identification (i.e. whenever a robot indicates that it has found a victim at a location where there isn't one).

5.6.2. Five (5) points are deducted for each lack of progress (i.e. whenever human intervention is required to enable a robot to resume progress along the black line).

Official Rules available

http://rcj.robocup.org/rcj2008/china-rescue-rules-page.pdf (Note: This references the RoboCupJunior 2008 Rescue rules. The committee members were Ashley Green, Maverick Luk, Eli Kolberg and Bill Freitas. You may wish to work with the most up to date version.)

**Rescue Robot Rubric**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Extra Credit | A | B | C | F |
| Victims | Found victims are counted and count is displayed | All victims correctly identified | Most victims correctly identified | Some victims correctly identified | No victims correctly identified |
| Gaps |  | All gaps navigated correctly | Most gaps navigated correctly | Some gaps navigated correctly | No gaps navigated correctly |
| Debris |  | Robot avoided all debris | Robot avoided most debris | Robot avoided some debris | Robot unable to avoid debris |
| Rooms |  | Robot entered all rooms through one door and exited through the other | Robot entered most rooms through one door and exited through the other | Robot entered one room and was unable to exit | Robot did not enter the first room |
| Constructions | Robot constructed using advanced gearing or other advanced construction techniques. Robot demonstrates extraordinary creativity. | Robot is of sound construction: nothing falls off, robot works as intended. | Parts of robot fall off.  Very simple construction – mechanics not used well. | Robot does not work as intended, but does move. Robot falls apart. Robot is unable to navigate due to construction | Robot falls apart or does not move at all. Construction appears careless or haphazard. |
| Programming | Program uses advanced techniques including blocks from the complete palette, flow blocks, etc. | Program is straightforward and efficient, using loops and parallel sequences as necessary. Program uses sensors and strong logic to navigate challenges and find victims. | Program is straightforward and easy to understand. Program uses inefficient logic to navigate challenges and find victims. | Program is poorly written or difficult to understand. Program has unused parts or does not correctly control robot. Program does not correctly use sensors to control motion. | Program does not work. |
| Cooperating | Student(s) helped other groups. Managed own role & helped group members. | Student worked well with group. Student participated actively in all parts of project. | Student worked somewhat well with group. Student participated in most parts of project. | Student had trouble working with group. Student participated in few parts of project. | Student did not participate in project. Student sabotaged others’ work. Made it difficult for group to work. |