Rocket to Mars

Lesson Date: 5/4/2023 Teacher(s): Joshua Miller

Duration: 20 minutes per activity Age range: 9-13

Lesson Objectives:

1. First Objective: Teach basic programming skills
2. Second Objective: Understanding computers only do what you tell them to do

Lesson Materials:

1. 4x Red arrows, printed and cut out RedArrowsDoc.docx - Google Docs
2. 4x Blue arrows, printed and cut out (if group size allows) BlueArrowsDoc.docx - Google Docs
3. Planets, printed and cut out, one page per group Planets.docx - Google Docs
4. 1 8 x 10 grid

Lesson Introduction:

5. Familiarize students with robots they already know
6. Explain how the fundamentals of programing work
7. Explain rules of the activity, being you need to go from earth

Lesson Activity:

1. First, place the earth in the corner of the board, one team's earth in opposite corners of the grid
2. Then, place the Mars planet anywhere on the grid, depending on how difficult you want it to be.
3. Have the students divide into groups, one be the rocket, one be the tester, and the rest be programmers.
4. The programmers, by themselves, will plan the route of the rocket using the arrows. It is recommended they do not do it on the grid, but can keep the grid within eyesight. The up

arrow means the rocket person moves forward one square, the left and right arrows mean to turn left or right, but do not move in that direction.

5. When the programmers are done, the rocket will move to the earth square, and the tester will read out the code. “Forward, forward, turn left, forward, turn right….”. The rocket will execute what the tester says.
6. If there are two teams these teams will execute their code at the same time, alternating reading directions. Hopefully they do not crash…

Lesson Conclusion:

1. The students were able to debug any issues in their code, which is a skill programmers do too
2. Helped recognize patterns
3. Help think the way a computer or a robot would think, so that they can code their own if they would like

Lesson Reflection:

1. What went well was our rotating through being the tester, rocket, and programmer, each student got an opportunity to do the work and do the fun part.
2. With only one programmer, most bugs get missed, causing frustration during execution. Having multiple programmers would help fix this. Additionally, make sure the students arrange their arrows in a line, so that it is more readable, and can fix any syntax mistakes (i.e. if they had their directions in a square, and a forward arrow was supposed to be a left arrow, can be very confusing in a square)
3. Having a larger group of people