For humans, locating car keys or navigating a shopping mall is a relatively straightforward undertaking. Yet programming an autonomous machine to perform these tasks can be very challenging. This project focused upon developing a standard algorithm for a “digital” rover to navigate and map a field of obstacles (voids), and create a simulation in Visual C++ to test the logic’s performance against various scenarios. In theory the rover looks around its immediate area, recognizes a pattern matching what it “sees”, and takes action based on the rule assigned for the pattern. After manually testing various simulation field sizes, void shapes and configurations, a cascading decision tree approach using a series of nested if statements was deemed the best option to code the 256 possible decision options.

The simulation graphically displays on a console window movements the rover chose as it navigates the simulation field around obstacles in real-time. The simulation results showed the decision algorithm could easily navigate the rover across the field and map single voids, multiple voids and many simple shapes. Some void shapes proved more challenging, causing cells to be missed, the rover to backtrack, or trapping the rover. The simulation program demonstrated the rover movement decision algorithm could not accurately solve all scenarios. Overall the simulation program itself worked well interacting with the user to test the algorithm strategy. This exercise reinforces how difficult it is for a machine to navigate an unknown area without more interaction from outside sources or more information about the surroundings before the rover can plan the most efficient path to completion.