

# iCreate Roomba Search Algorithm

Jacob Wilson

December 15, 2016

## Setup

### C++ Implementation

Setup the libcreate library on your machine, the github page has adequate instructions.

libcreate Library: <https://github.com/AutonomyLab/libcreate>

Ensure the the serial cable is connected, check that the device is listed as ttyUSB0 (run `ls /dev` on linux), and force permissions with `sudo chmod 777 /dev/ttyUSB0`. For Windows, ensure the appropriate drivers are installed.

The iCreate should be now be connected to your laptop device, this can be tested through the example programs provided by the library. Function calls aren't well documented, however the functions can be found in the example code. "drive" sets the linear velocity in meters per second and the angular velocity in radians per second (see libcreate-master/examples for more usage).

Other libraries exist for the iCreate to connect with C++, however the libcreate library seemed to be the most organized and updated.

### Python Implementation

The iCreate website suggested the BreezyCreate2 API for connecting with Python, follow the link below for setup.

BreezyCreate2 Library: <https://github.com/simondlevy/BreezyCreate2>

## Usage

### C++ Implementation

`g++ file.cpp -o find_route` – to compile manually

`./find_route points_file adjacencyMatrix_file search_type start_point end_point` – to run manually

`make basic` – to run BFS, DFS, and uniform on the basic path

`make adv` – to run BFS, DFS, and uniform on the advanced path

### Python Implementation

`./Build.sh` – to make the python code executable

`./find_route points_file adjacencyMatrix_file search_type start_point end_point` – to compile and run manually

`./find_route_run` – to run BFS, DFS, and uniform on the basic path

`./find_route_run 1` – to run BFS, DFS, and uniform on the advanced path

## Description

`points_file` details the label for each node and its associated X-Y (real-world) coordinate locations. `adjacencyMatrix_file` details the unweighted adjacency matrix of the problem, whether an edge connection exists between the points/nodes of the problem. The 0th row and column label which node is being specified. The file doesn't add weight because the weights are calculated using the distance formula with the X-Y coordinates.

`searchType` details the type of search to perform with the given information. The options are

breadth-first search (BFS as the argument), depth-first search (DFS as the argument), or uniform search (UNIFORM as the argument).

start\_point is the argument which sets the point to start the search

end\_point is the argument which sets the point to end the search

If the user types incorrectly the program will display to the user the proper formatting and options, for both C++ and Python.

## Code

See github repo: [https://github.com/jacwilso/AI\\_IndependentStudy](https://github.com/jacwilso/AI_IndependentStudy)

## Further Expansion

### Problems

1. The current iCreate has cheap motor parts which makes knowing/setting the heading and speed with certainty problematic. The motor slips thus cannot be programmed with precision. The speed at which the Roomba moves/rotates is also depended on the flooring the Roomba is on.
2. Roomba is laptop dependent, ie requires 3rd party library to be installed on each laptop. The Roomba also needs to be connected to laptop at all times.

### Possible Solutions

1. This could be included as part of the project, but some initial thoughts:
  - Have an overhead camera and use computer vision to track the actual location and heading of the robot.
  - Have an established surface of testing and allow for some uncertainty (would need to be fairly large range).
  - Replace the Roomba motors.
  - Change the approach from setting the linear/angular velocities to detecting a collision with an object at the nodes
2. The iCreate website has usefule tutorials, specific solutions can be found at the bottom of the page (<http://www.irobot.com/About-iRobot/STEM/Create-2/Projects.aspx>).
  - Purchase an Arduino or Raspberry Pi (preferably the Pi) and install within the Roomba frame. Downfall of this option, the micro controller will need to be accessible to connect to a mouse/keyboard/monitor.
  - Purchase a bluetooth circuit and install within the Roomba frame. Downfall of this option, the 3rd party libraries still need to be installed on each machine.
3. Sound/ ring when the Roomba reaches goal. The Roomba has sound affects when it performs certain actions like charging/ powering on/off/ etc.
4. The Roomba is surrounded by sensors, none of which get used in the current model, the connection layout could be altered to be modeled with walls around the edges