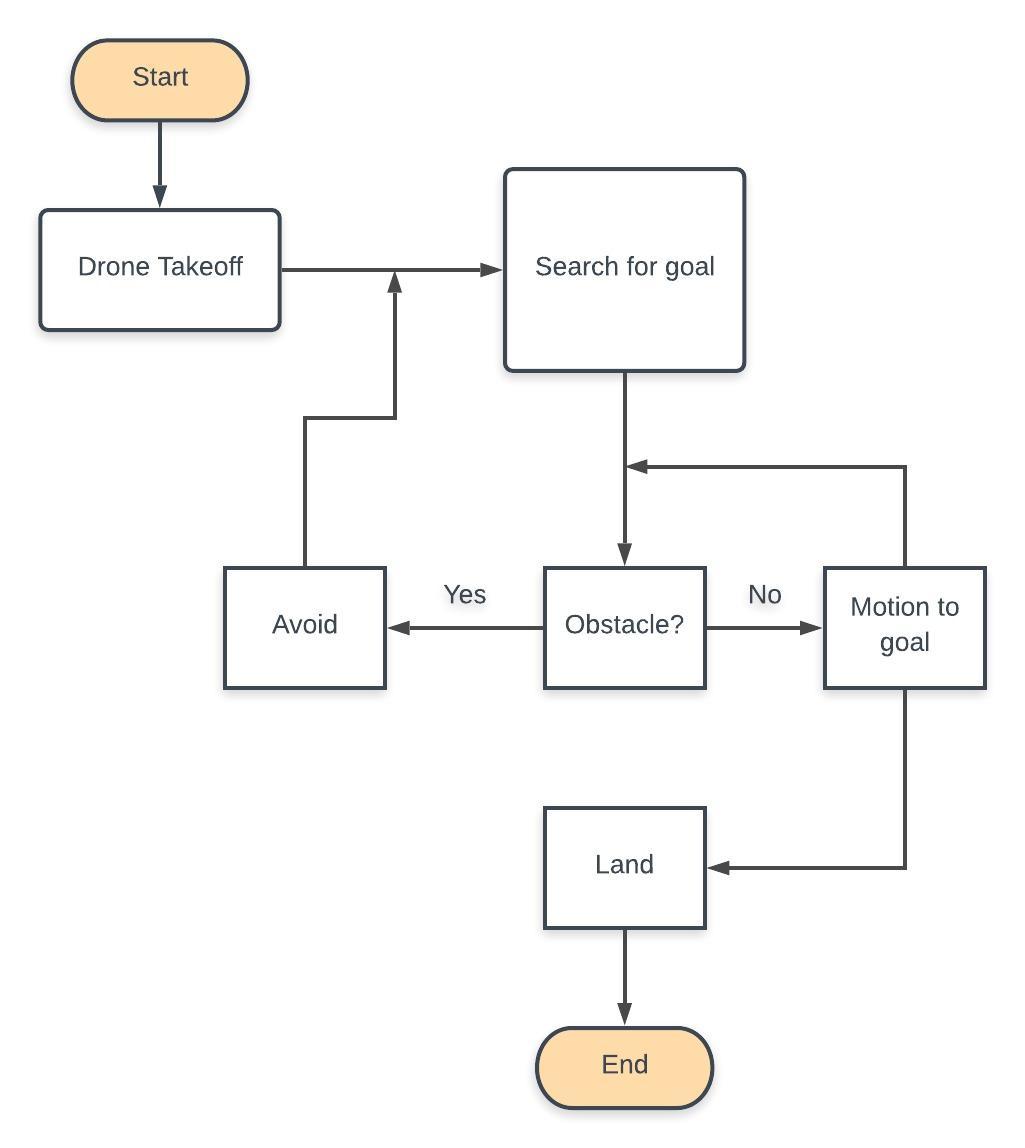
The fly.py program automates an AR Drone so that it can fly toward a goal while avoiding obstacles through a bug algorithm. It uses software from ROS to control the drone. ROS is effective software that allows the control of various kinds of robots. As of currently ROS is only capable of working on the Ubuntu Linus system and therefore the entire fly.py program was built using ROS on an Linux.

The algorithm is as follows:



The programs starts the drone by using the takeoff function. The drone searches for the goal using the front camera which transfers images over to the computer in OpenCV, a computer vision software. The goal is a fixed size object. The drone will rotate about the z axis until it is centered in the image. It will travel toward the goal so long as there are no obstacles in the way. If there is an obstacle blocking the path to the goal, the drone will strafe either right or left depending on the quicker route while also rotating the opposite way to face the goal simultaneously. An object is a fixed size object also. With both the goal and obstacles, distance is measured as an area on an image. So the closer the object is to the drone camera the larger its area will appear. This is the reason each obstacle size needs to be fixed. After the drone corrects its orientation, it will move toward the goal. If the goal is reached, the drone will automatically land and the program will terminate otherwise it will move toward the goal and avoid other obstacles.

For additional help to get started visit the wiki.ros.org website for installation instructions.

|  |
| --- |
| Using ROS we operate an ARDrone to fly toward a goal, while avoiding obstacles. |

|  |
| --- |
|  |

|  |
| --- |
| There are a few things we must do before testing: |

|  |
| --- |
|  |

|  |
| --- |
| 1. Use Linux Ubuntu 16.04 |

|  |
| --- |
| - you can either dual install it along side your own operating system or use a computer with it already installed (must have Wireless Internet capability) |

|  |
| --- |
|  |

|  |
| --- |
| 2. Install ROS for Ubuntu (http://wiki.ros.org/kinetic/Installation) |

|  |
| --- |
| - follow all steps |

|  |
| --- |
|  |

|  |
| --- |
| 3. Download all files used for the program (drone.launch and fly.py) |

|  |
| --- |
|  |

|  |
| --- |
| 4. Connect to the ROS node from computer (possibly not necessary) |

|  |
| --- |
| - this may require you to change hostname to an IP address (e.g. 192.168.1.2) |

|  |
| --- |
|  |

|  |
| --- |
| 5. On a terminal run command 'roscore' (no quotations) |

|  |
| --- |
| - this will connect you to the ROS master node |

|  |
| --- |
|  |

|  |
| --- |
| 6. On a seperate terminal and in the workspace of drone.launch run command 'roslaunch drone.launch' |

|  |
| --- |
| - lets the drone know what packages to use |

|  |
| --- |
|  |

|  |
| --- |
| 7. On a third terminal and in the workspace of fly.py, run 'python fly.py'  - the default goal is a 3.5 x 4.5 inch bright pink paper  - default obstacles are 9 x 6.5 inch bright lime green paper  - you may change the hsv values of the desired colors for goals and obstacles |

- observe