```
function drawCar(state)

    % process inputs to function
    x           = state(1);       % inertial x position (cm)
    y           = state(2);       % inertial y position (cm)
    theta       = state(3);       % heading angle (rad)
    vel         = state(4);       % forward velocity (cm/sec)
    theta_dot   = state(5);       % turn rate (rad/sec)
    t           = state(6);       % time (s)

    % define persistent variables
    persistent car_handle;  % figure handle for car
    persistent lidar_handle; %figure for lidar
    persistent Vertices
    persistent Faces
    persistent facecolors


    % first time function is called, initialize plot and persistent vars
    if t==0
        figure(1), clf
        [Vertices,Faces,facecolors] = defineCarBody;
        car_handle = drawBody(Vertices,Faces,facecolors,...
                                   x, y, theta,...
                                   []);
        title('Rover Course')
        xlabel('x (cm)')
        ylabel('y (cm)')

        axis([x-100,x+100,y-100,y+100]);
        grid on

    % at every other time step, redraw quadrotor and target
    else
        drawBody(Vertices,Faces,facecolors,...
                     x, y, theta,...
                     car_handle);

        % move axes with car
        set(car_handle.Parent, 'XLim',[x-100,x+100])
        set(car_handle.Parent, 'YLim',[y-100,y+100])
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function handle = drawBody(V,F,colors,...
                               x, y, theta,...
                               handle)
  V = rotate(V, theta);  % rotate rigid body
  V = translate(V, x, y);  % translate after rotation
  if isempty(handle)
    handle = patch('Vertices', V', 'Faces', F,...
                 'FaceVertexCData',colors,...
                 'FaceColor','flat');
    %hold on;
    %lidar_handle = rectangle('Position', lidar, 'Curvature', [1 1]);
  else
    set(handle,'Vertices',V','Faces',F);
    %set(lidar_handle, 'Position', lidar);
    drawnow
  end

end

%%%%%%%%%%%%%%%%%%%%%%%
function pts=rotate(pts,theta)

%TODO
  % define a rotation matrix as an SO(2) matrix that is able to rotate
  % non-homogenous points found in the pts 2xn matrix by an angle of theta.
  % and return the pts matrix rotated by the SO(2) matrix from this function.

  rotMat = [cos(theta), -sin(theta); sin(theta), cos(theta)];
  disp(size(pts, 2));
  for i = 1:size(pts, 2)
      temp = [pts(1, i); pts(2, i)];
      temp = rotMat * temp;
      pts(1, i) = temp(1, 1);
      pts (2, i) = temp(2, 1);
  end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% translate vertices by x, y
function pts = translate(pts,x, y)

  pts = pts + repmat([x;y],1,size(pts,2));

end

% end translate


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define aircraft vertices and faces
function [V,F,colors] = defineCarBody

% parameters for drawing aircraft
  % scale = 20  %only for scale drawing purposes.
  scale = 1;
  chasis_width = 18;
  chasis_forward = 20;
  chasis_backward = -2;
  wheel_base = 21;
  wheel_width = 2.5;
  wheel_radius = 4.5;
  lidar_forward = 16*scale;
  lidar_radius = 3.5*scale;

  %define points
  b_lf = [chasis_forward chasis_width/2]';
  b_rf = [chasis_forward -chasis_width/2]';
  b_lr = [chasis_backward chasis_width/2]';
  b_rr = [chasis_backward -chasis_width/2]';
  li_lf = [lidar_forward+lidar_radius lidar_radius]';
  li_rf = [lidar_forward+lidar_radius -lidar_radius]';
  li_rr = [lidar_forward-lidar_radius -lidar_radius]';
  li_lr = [lidar_forward-lidar_radius lidar_radius]';

  %TODO define the points associated with the boundary of the tires
  rw_lf = [wheel_radius chasis_width/2+wheel_width]';
  rw_rf = [wheel_radius chasis_width/2]';
  rw_lr = [-wheel_radius chasis_width/2+wheel_width]';
  rw_rr = [-wheel_radius chasis_width/2]';

  lw_lf = [wheel_radius -chasis_width/2]';
  lw_rf = [wheel_radius -(chasis_width/2+wheel_width)]';
  lw_lr = [-wheel_radius -chasis_width/2]';
  lw_rr = [-wheel_radius -(chasis_width/2+wheel_width)]';

  %define faces
  body = [b_lf, b_rf, b_rr, b_lr];
  lidar = [li_lf, li_rf, li_rr, li_lr];  %lidar square here
  rWheel = [rw_lf, rw_rf, rw_lr, rw_rr];
  lWheel = [lw_lf, lw_rf, lw_lr, lw_rr];

  %TODO define the faces associated with the tires for plotting purposes

  % colors
  red     = [1, 0, 0];
  green   = [0, 1, 0];
  blue    = [0, 0, 1];
  yellow  = [1,1,0];
  magenta = [0, 1, 1];
  black   = [0, 0, 0];

%TODO add the tires to the V and F matricies
V = [body, lidar, rWheel, lWheel];

F = [...
      1, 2, 3, 4;... %body
      5, 6, 7, 8;... %lidar square
      9, 10, 11, 12;... %rWheel
      13, 14, 15, 16;... %lWheel
];

%Add the wheel colors to the colors matrix
colors = [...
        blue;... % body
        black;... %lidar
        black;... %rWheel
        black;... %lWheel
  ];

  V = scale*V;   % rescale vertices
  end
```