CRITERIA	MEETS SPECIFICATIONS
Your code should compile.	Code must compile without errors with `cmake` and `make`. Given that we've made CMakeLists.txt as general as possible, it's recommend that you do not change it unless you can guarantee that your changes will still compile on any platform. Meets Specification, compile with cmake / make.

CRITERIA	MEETS SPECIFICATIONS

The Model	Student describes their model in detail. This includes the state, actuators and update equations.
	
	State: The state is set to reference to vehicle coordinates, calculating the difference x, y and psi to car coordinates, getting a state of x=0, y=0 psi=0, and v, cte and epsi.
	
	Update Equations. Position:
	
	fg[1 + x_start + t] = x1 - (x0 + v0 * CppAD::cos(psi0) * dt);
	
	fg[1 + y_start + t] = y1 - (y0 + v0 * CppAD::sin(psi0) * dt);
	
	Orientation (modified to match simulator rotation sense) :
	
	fg[1 + psi_start + t] = psi1 - (psi0 - v0/Lf * delta0 * dt);
	
	Velocity.
	
	fg[1 + v_start + t] = v1 - (v0 + a0 * dt);
	
	Actuators:
	
	Steer value and throttle value, with the ranges of -25, 25 degrees for steering (delta), and -1, 1 for throttle (a) .

CRITERIA	MEETS SPECIFICATIONS

	Cross track error:
	
	fg[1 + cte_start + t] = cte1 - ((f0 - y0) + (v0 * CppAD::sin(epsi0) * dt));
	
	Orientation error:
	
	fg[1 + epsi_start + t] = epsi1 - ((psi0 - psides0) - v0/Lf * delta0 * dt);
	
	Cost: v*10 to get close to reference speed.
	
	for(int t = 0; t < N; t++){
	
	fg[0] += CppAD::pow(vars[cte_start + t], 2);
	
	fg[0] += CppAD::pow(vars[epsi_start + t], 2);
	
	fg[0] += 10 * CppAD::pow(vars[v_start + t] - ref_v, 2);
	
	}
	
	Minimize the use of actuators. A*50 for smooth acceleration and reduce continuum breaking.
	
	for(int t = 0; t < N - 1; t++){
	
	fg[0] += CppAD::pow(vars[delta_start + t], 2);
	
	fg[0] += 50 * CppAD::pow(vars[a_start + t], 2);
	
	}
	
	Minimize the value gap between sequential actuations. Delta*1000 to avoid suddent direction changes.
	
	for(int t = 0; t < N - 2; t++){
	
	fg[0] += 1000 * CppAD::pow(vars[delta_start + t + 1] - vars[delta_start + t], 2);
	
	fg[0] += CppAD::pow(vars[a_start + t + 1] - vars[a_start + t], 2); }

CRITERIA	MEETS SPECIFICATIONS
Timestep Length and Elapsed Duration (N & dt)	Student discusses the reasoning behind the chosen *N* (timestep length) and *dt* (elapsed duration between timesteps) values. Additionally the student details the previous values tried. N = 15, dt = 0.05, smaller dt allow better track when speed increase, as N increase, the computation requirements increase also. At higher speed a smaller dt gives a better aproximation, and in order to maintain the same look ahead distance/time N can be increased proportionally.
Polynomial Fitting and MPC Preprocessing	A polynomial is fitted to waypoints. If the student preprocesses waypoints, the vehicle state, and/or actuators prior to the MPC procedure it is described. The polynomial fit choose is a 2nd degree polynomial, show good enough for the track shape, single curve at a time, also should reduce the computations needed for a 3rd degree polynomial. Theres a conversion of the steer value is divided in to radians(25 degrees).
Model Predictive Control with Latency	The student implements Model Predictive Control that handles a 100 millisecond latency. Student provides details on how they deal with latency. recalculate the start point position x and y at time t t + latency 0.1 seconds, considering the speed and direction don’t change withing the 0.1 latency interval.

CRITERIA	MEETS SPECIFICATIONS
	px = px + v * cos(psi) * 0.1; py = py + v * sin(psi) * 0.1;

CRITERIA	MEETS SPECIFICATIONS
The vehicle must successfully drive a lap around the track.	No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle). The car can't go over the curb, but, driving on the lines before the curb is ok. Tested at different speeds, it works good with a reference speed of 40 mph, maintains an average speed of 37 mph, and a good path Link to video showing the curves after the bridge. https://youtu.be/w_k3gxsAB6A