Brief

Select a suitable technique (or set of techniques) from those covered during the module and write a “robot” for The Open Racing Car Simulator (TORCS - http://torcs.sourceforge.net/).

While there is no requirement to build the entire TORCS project to create a robot, it may be helpful to go through the process. A fork of the project that will build in Visual Studio 2013 (NOT 2015) is included in your assignment repository as a starting point.

You must perform the following once you have pulled the files from github:

- Run setup_win32.bat
- Run setup_win32-data-from-CVS.bat

All the files you need to be concerned about are in torcs/src/drivers/robot_base/ which includes a cpp and vcxproj file for visual studio and then some TORCS specific files in the form of robot_base.xml, car1-stock1.rg, car1-stock1.xcf and logo.rgb.

You will need to create a robot_base folder under torcs/runtime/drivers and copy the TORCS specific files across to that folder. Once you build your project, you will want to ensure that the compiled dll is also present (it should be copied automatically as part of the build, but if not, copy it yourself).

TORCS includes a number of existing robots that you may want to look at for informational purposes. The core function to be concerned with within robot_base.cpp is the drive function, think of this as a “tick” or “update” function that the game engine calls every frame.

Resources

- An in-depth tutorial on writing a TORCS robot is available at http://www.berniw.org/aboutme/publications/torcs.pdf
    - I have undertaken the work up until Chapter 2.3 for you.
- Visual Studio 2013 installer: https://www.microsoft.com/en-gb/download/details.aspx?id=44914

Submission

https://classroom.github.com/assignment-invitations/489e6160981f43933ca8191fedb77ba3

The GitHub commit made prior to or at noon on the 28th April 2017 will be used as your final submission. Any further commits to the project will be ignored for the purposes of assessment.

Your ‘master’ branch must contain a well organised project folder with all relevant files and assets and a distinct word document containing your bibliography if required.

Any other branches will be ignored.

ASSESSMENT CRITERIA	Marks
40% - 49%	
Some attempt has been made to have the robot navigate a track, but there is heavy reliance on the example code base.	
A built dll is supplied along with source code that features limited comments.	
50% - 59%	
The robot is able to navigate a number of tracks, but may become stuck or otherwise hindered in progressing smoothly.	
Source code is reasonably commented and comments describe what was intended to happen.	
60% - 69%	
The robot can successfully navigate any of the available tracks, with the ability to sensibly accelerate & brake for efficient cornering.	
70% - 100%	
The implementation is well polished, with the design of the code being well thought out and all elements of the implementations are approached efficiently.	
Grade	