# Explanation into choice of approach

I chose to implement a Behaviour Tree based system to control and run an AI system for the car over all other methods. This is mainly due to the fact that it is generally more powerful than the other methods such as a Finite State Machine: although a FSM would have been just as suitable, behaviour tress are able react to all sorts of situations in games much easier. Nodes/tasks can be easily connected and flow better without the need to know what each other are, they are completely independent – with an FSM, any states that can link have to be referenced within each other and is very time consuming to add new states in as every node will need to be checked if it can link.

# Notes

*Source Code is all handled in the robot\_base driver folders, issues became apparent when trying to copy and change the robot\_base solution to my own name and directory*

# Bibliography

Opsive. (2019). *Behavior Trees or Finite State Machines | Opsive*. [online] Available at: https://opsive.com/support/documentation/behavior-designer/behavior-trees-or-finite-state-machines/ [Accessed 7 Dec. 2018].

Gamasutra.com. (2019). *Behavior trees for AI: How they work*. [online] Available at: https://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior\_trees\_for\_AI\_How\_they\_work.php [Accessed 9 Dec. 2018].

Gameaipro.com. (2019). *The Behaviour Tree Starter Kit*. [online] Available at: http://www.gameaipro.com/GameAIPro/GameAIPro\_Chapter06\_The\_Behavior\_Tree\_Starter\_Kit.pdf [Accessed 11 Dec. 2018].

Cplusplus.com. (2019). *Behaviour Tree - C++ Forum*. [online] Available at: http://www.cplusplus.com/forum/general/141582/ [Accessed 12 Dec. 2018].  
  
Wymann, B. (2019). *TORCS Robot Tutorial*. [online] Berniw.org. Available at: http://www.berniw.org/tutorials/robot/tutorial.html [Accessed 12 Dec. 2018].