Nathaniel Faries

Question 1

Board A	Minmax	Alpha-Beta
Depth = 3	37076	460
4	587045	1721
5	--	5053
6	--	21021
Board B	Minmax	Alpha-Beta
3	59233	451
4	836613	1278
5	--	7392
6	--	30690
Board C	Minmax	Alpha-Beta
3	49046	304
4	763501	1265
5	--	8830
6	--	54736

Did not do minmax at depth of five or six due to time...based on the results at depth 4 something's gone wrong somewhere...

Question 2: Does the number of states explored depend on the order in which you generate new states during the search? Explain your answer and justify it using results from your program.

The number of states explored does not depend on the order new states are generated for minimax search as it will generate all states within the specified depth. However, for alpha-beta search it will vary as the order is changed. We can see this from the above results in how there is no apparent correlation between the number of states generated by minimax and alpha-beta as we look from table to table. To further drive home this point, I reversed the order in which states are generated and ran the algorithms again for a depth of 3.

	Minmax	Alpha-Beta
Board A	37076	220
Board B	59233	645
Board C	49046	540

Question 3: Explain the heuristic evaluation function you used and provides a clear rationale for the factors you included.

The heuristic evaluation function I used is based on the number of "sequences" that exist for both players. A "sequence" is defined as two or three friendly pieces in a line, with at most two empty spaces between them, and no enemy pieces between them. The function differentiates between sequences of two and of three, so that they are weighted differently. The heuristic value of enemy sequences is then subtracted from friendly sequences, so that reducing enemy sequences is represented in heuristic value.

The function is defined as:

$$(c_1 * S_{f2}) + (c_2 * S_{f3}) - (c_3 * S_{e2}) - (c_4 * S_{e3})$$

Where ci is a coefficient that can be tweaked, and the subscript of S represents a sequence of "**f**riendly" or "**e**nemy" and two or three pieces. In the code submitted c1 and c3 were 1, and c2 and c4 were 3, thus providing an evaluation function that favoured sequences of three. Additionally, due to the nature of the code that found these sequences, a sequence of three was counted more times than a sequence of two, thus providing a natural weighting in favour of sequences of three.

Question 4: Discuss the computational trade-offs with respect to the use of a more complex evaluation function and the depth of the game tree that can be evaluated.

The computational requirements of this heuristic are incredibly simple and as such don't take up much time. However, with a more elaborate and complex heuristic, the alpha-beta pruning process could be more efficient in finding better states and thus spending less time searching per level of depth in the tree, and by doing so can search deeper in the tree in the same amount of time. Improvements for this particular assignment would be in making the evaluation function richer and able to provide value for more features of the game board.