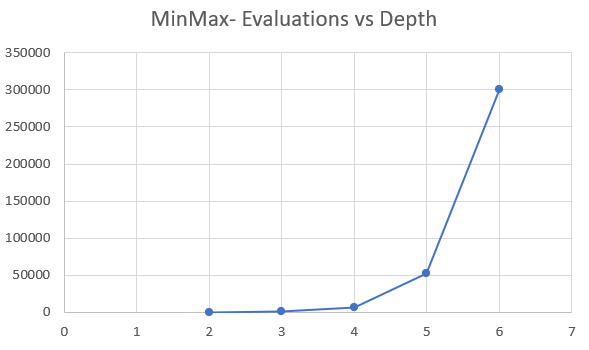
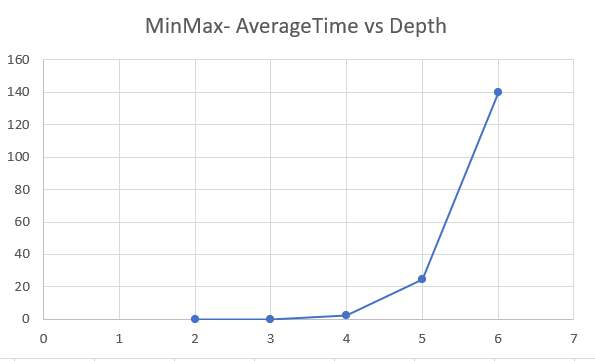
AverageTime vs Depth

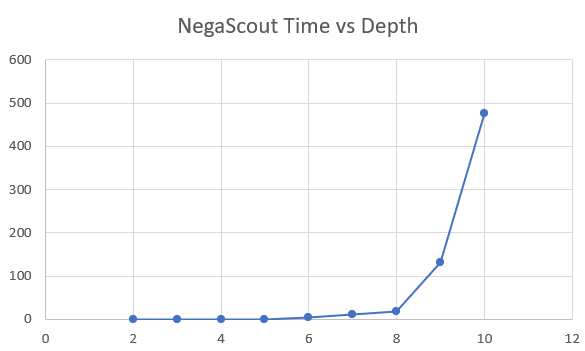
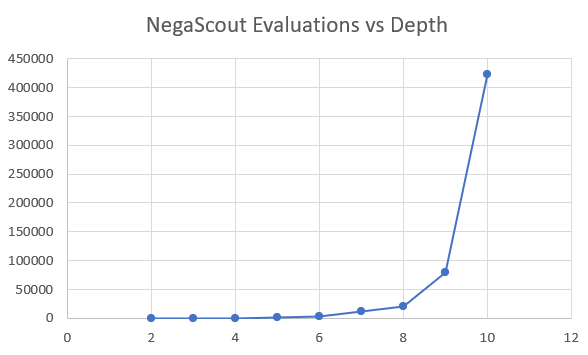
**X axis**: depth in algorithm applied **Y axis**: Average Time Taken by AI player per single move

Evaluation vs Depth

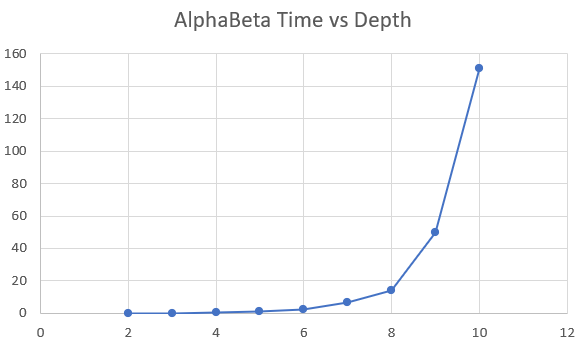
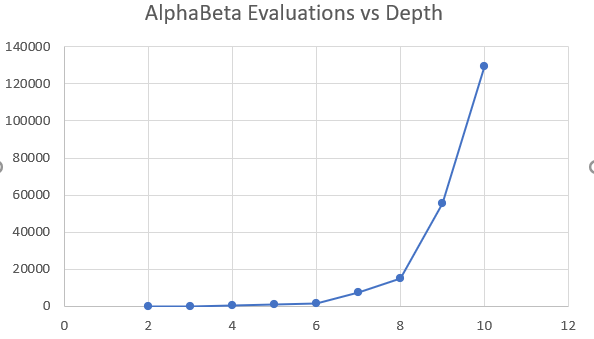
**X axis**: depth in algorithm applied **Y axis**: Average number of evaulations performed by AI player per single move



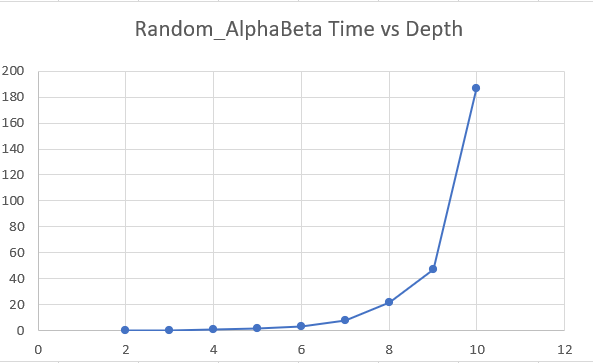
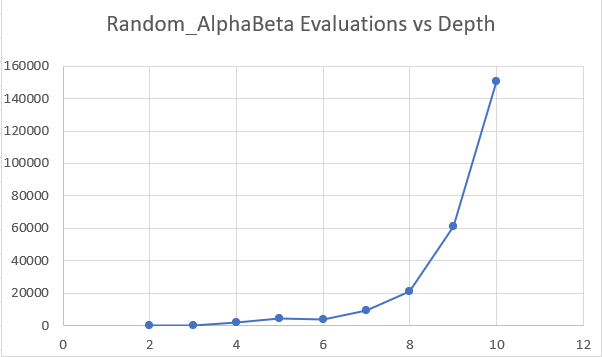
These graphs shows the exponential curve function of Time and Computations performed with respect to depth of the modal. It can be seen from the above graphs that after depth of four minmax AI player is of no use for implementation purpose as there is a lot of time involved for a single move,



High rise in efficiency can be visualized by implementing NegaScout. This algorithm is able to perform better in the terms of time as it responds within decent time slot till the depth of 8.



AlphaBeta also shows much greater efficient as compared from MinMax Algorithm. It performs much better than MinMax and bit better than NegaScout (scale-y axis of NegaScout algorithm is smaller than AlphaBeta)



Random Alpha Beta’s performance is similar to Alpha Beta’s performance in the terms of time and computation processes.

AI’s against each other:

Based on the experimental analysis all the AIs performs equilaterally well in competition with each other. MinMax, AlphaBeta, NegaScout are very competitive on the same level as they all are based on Brute Search sequential algorithm. Random AlphaBeta AI’s performance also proved to be very good in competing with other AI’s.