Salomé Lahmar
16201438

AI for Games & Puzzles
Assignment 1

I tested both of the alpha beta variants on 10 big trees with the following parameters :

- b = 4
- h = 8
- i = 4
- s = 4

I assigned the true value 42 to the top node.

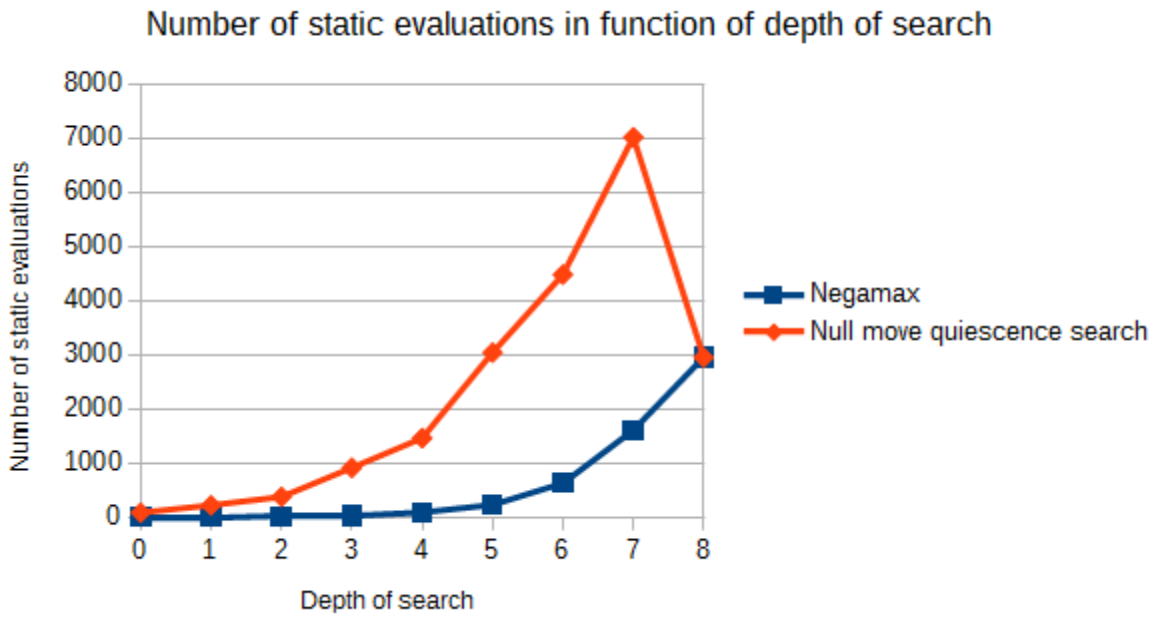
1. Number of static evaluations

	Depth 0	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5	Depth 6	Depth 7	Depth 8
Tree 1	1	4	15	24	68	145	503	1158	2350
Tree 2	1	4	10	39	74	267	690	1692	3092
Tree 3	1	4	13	42	133	289	737	1980	3493
Tree 4	1	4	11	37	119	251	747	2205	3260
Tree 5	1	4	9	25	41	225	456	1022	2205
Tree 6	1	4	12	35	97	285	765	1696	3065
Tree 7	1	4	13	24	47	147	399	1313	2142
Tree 8	1	4	11	45	158	288	895	2069	4517
Tree 9	1	4	12	28	70	172	427	1353	2851
Tree 10	1	4	11	40	102	219	675	1577	2532
Mean	**1**	**4**	**11,7**	**33,9**	**90,9**	**228,8**	**629,4**	**1606,5**	**2950,7**

Negamax

	Depth 0	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5	Depth 6	Depth 7	Depth 8
Tree 1	14	133	156	420	905	2112	3595	5580	2350
Tree 2	9	132	107	819	1644	3467	5293	6480	3092
Tree 3	5	226	571	1420	1892	3545	5058	9525	3493
Tree 4	9	94	508	988	2054	3220	5205	9285	3260
Tree 5	190	300	493	833	679	3301	3744	4905	2205
Tree 6	59	461	352	1169	901	3418	5373	7505	3065
Tree 7	9	32	261	684	911	2261	3000	4880	2142
Tree 8	249	256	549	1290	2184	3664	6159	8940	4517
Tree 9	5	261	295	613	1473	2423	3068	6260	2851
Tree 10	317	322	476	912	1987	2897	4321	6790	2532
Mean	**86,6**	**221,7**	**376,8**	**914,8**	**1463**	**3040,8**	**4481,6**	**7015**	**2950,7**

Null-move quiescence search



As we can see on the graph above, the null move quiescence search variant of alpha beta performs many more evaluations than the negamax variant. Indeed, as soon as the algorithm reaches the depth provided, it tries to evaluate the node by searching all the nodes under it, whereas the negamax variant only evaluates it by negating its estimated value.
The number of static evaluations is the same when the depth of search is equal to the height of the tree, as both of the algorithms only evaluate the same portion of leaf nodes.

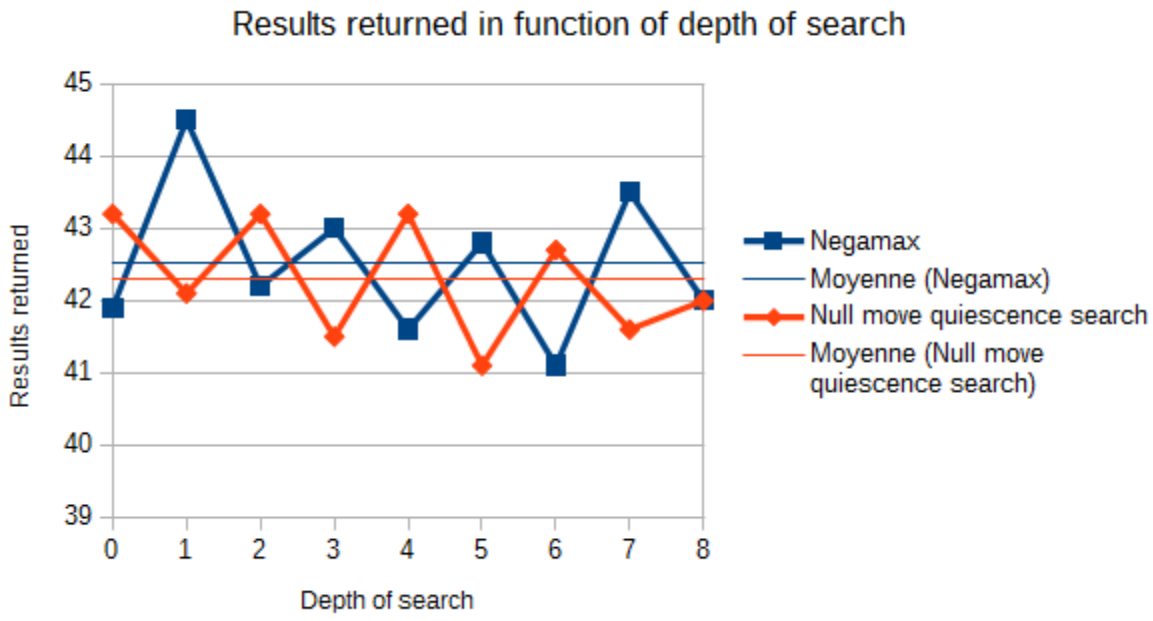
2. Results returned by searches

	Depth 0	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5	Depth 6	Depth 7	Depth 8	**Mean**
Tree 1	44	46	43	43	42	44	43	45	42	**43,56**
Tree 2	42	43	45	44	41	43	41	43	42	**42,67**
Tree 3	46	43	42	44	41	42	41	43	42	**42,67**
Tree 4	44	45	45	42	40	44	42	45	42	**43,22**
Tree 5	42	46	41	44	44	43	41	44	42	**43**
Tree 6	42	46	38	42	41	41	39	43	42	**41,56**
Tree 7	38	42	43	44	43	45	41	43	42	**42,33**
Tree 8	38	46	40	41	41	42	40	43	42	**41,44**
Tree 9	44	44	39	43	40	43	42	43	42	**42,22**
Tree 10	39	44	46	43	43	41	41	43	42	**42,44**
Mean	**41,9**	**44,5**	**42,2**	**43**	**41,6**	**42,8**	**41,1**	**43,5**	**42**	**42,51**

Negamax

	Depth 0	Depth 1	Depth 2	Depth 3	Depth 4	Depth 5	Depth 6	Depth 7	Depth 8	**Mean**
Tree 1	44	43	43	43	45	42	43	42	42	**43**
Tree 2	43	43	45	43	43	41	42	42	42	**42,67**
Tree 3	46	41	42	42	43	40	43	41	42	**42,22**
Tree 4	45	45	45	41	42	42	43	42	42	**43**
Tree 5	44	44	44	42	44	41	43	42	42	**42,89**
Tree 6	42	41	41	39	42	41	42	41	42	**41,22**
Tree 7	42	42	44	43	45	41	43	42	42	**42,67**
Tree 8	42	42	42	41	42	40	42	42	42	**41,67**
Tree 9	44	40	40	40	42	42	42	41	42	**41,44**
Tree 10	40	40	46	41	44	41	44	41	42	**42,11**
Mean	**43,2**	**42,1**	**43,2**	**41,5**	**43,2**	**41,1**	**42,7**	**41,6**	**42**	**42,29**

Null move quiescence search



The graph above shows that the null move quiescence search is more precise than the negamax algorithm : the mean value of the results returned is closer to the true value of the top node (42). Moreover, the standard deviation of the NMQS is smaller than the negamax's (around 1,4 and 1,8) : the negamax algorithms returns more outliers.

3. Conclusion

Both of the implemented algorithms have their advantages and disadvantages : indeed, the negamax variant will be quicker to return a result, but it will not be always the right one. On the other hand, the NMQS variant will be slower, as it evaluates more nodes, but the result is closer to the true value. Let us note that, sometimes, executing the negamax on the whole tree can be quicker than NMQS on a smaller depth.

This assignment allowed me to practise the programming in Python, a language that I find concise and clear for this type of algorithms. I did the assignment step by step, checking after every function if what I did was right before beginning to implement another.