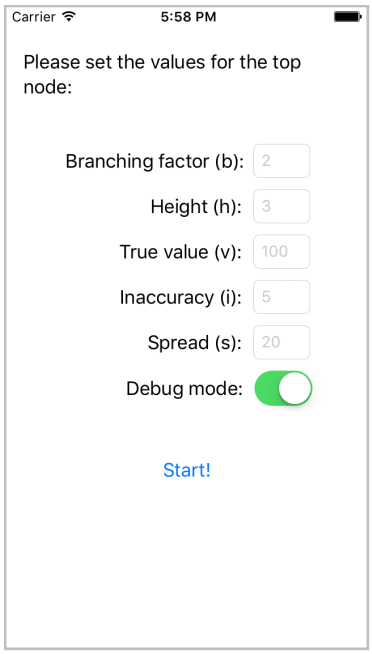
Samuel Pineda

COMP30260: Assignment 1
Null Move Quiescence Search

For this assignment I decided to use SWIFT 3 the programming language used in Xcode 8 to develop applications for iOS devices. I designed a basic user interface for the iPhone that allows the user to set the values for the top node and activate or deactivate the debug mode —which prints informative messages whenever a leaf node is evaluated and whenever an interior node is calculated—.



However, due to the small size of an iPhone's screen all the informative messages are printed in the console of Xcode. For instance, the following screenshot is an extract of the output generated for a small tree when the debug mode is on.

```
------------------------------------------------------------------------
 Simple Alpha-Beta (depth = 1):
------------------------------------------------------------------------

{estValue: -82, h: 0, ab: (-∞, ∞), eval: -82, method: "Static (regular)"}
{estValue: -95, h: 0, ab: (-∞, -82), eval: -95, method: "Static (regular)"}
        {estValue: 104, h: 1, ab: (95, ∞), eval: 95, method: "Search"}

-> RESULT: value = 95, static evaluations = 2

------------------------------------------------------------------------
 A-B with Null Move Quiescence (depth = 1):
------------------------------------------------------------------------

{estValue: 90, depth: -1, lower: -∞, upper: 82, best: 90, method: "Static (NMQuiesce)"}
{estValue: 85, depth: -1, lower: -∞, upper: 82, best: 85, method: "Static (NMQuiesce)"}
        {estValue: -82, depth: 0, lower: -∞, upper: ∞, best: -82, method: "Static (NMQuiesce)"}
{estValue: 96, depth: -1, lower: -∞, upper: 95, best: 96, method: "Static (NMQuiesce)"}
{estValue: 119, depth: -1, lower: -∞, upper: 95, best: 119, method: "Static (NMQuiesce)"}
        {estValue: -95, depth: 0, lower: -∞, upper: ∞, best: -95, method: "Static (NMQuiesce)"}
                {estValue: 104, h: 1, ab: (95, ∞), eval: 95, method: "Search"}

-> RESULT: value = 95, static evaluations = 6
```

To compare the results between the null move quiescence search and the pure alpha-beta algorithm I created 10 trees with the following parameters:

	Value
Branching factor	5
Height	10
True value	100
Inaccuracy	5
Spread	20

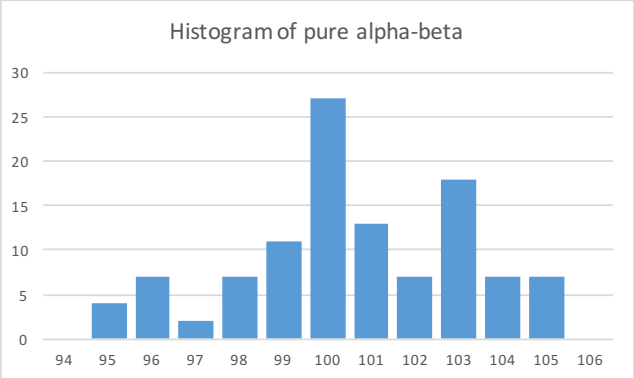
Then, I applied both algorithms to the same ten trees —at all search depths from 0 to 10— and for every tree I recorded the results in a table like the following one:

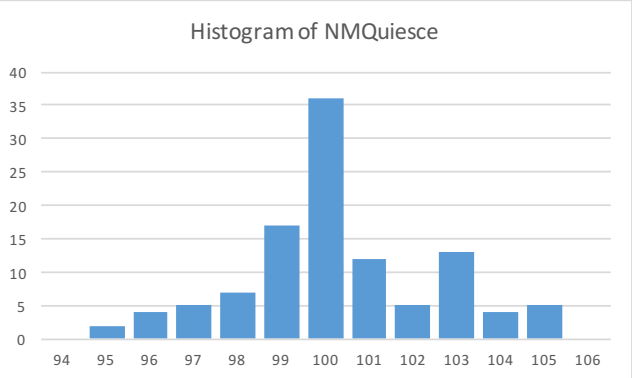
Search depth	AB - value	AB - static evaluations	NMQuiesce - value	NMQuiesce - static evaluations
0	96	1	99	104
1	103	5	99	143
2	99	17	99	272
3	101	60	99	970
4	100	180	102	3540
5	100	633	100	10327
6	102	1602	104	23811
7	96	5122	96	63568
8	100	13970	101	113138
9	100	39455	98	231276
10	100	98734	100	98734

After recording the results for every tree I tried to summarise, organise and describe the data with some basic statistical concepts. For instance, the following table summarises the data through five features that I considered important: percentage of times that the correct value —100— was found, the average difference between the calculated value and the true value, the average number of static evaluations performed, the average of all the calculated values, and the standard deviation of all the calculated values.

	% of correct value	Avg. difference	Avg. of static evaluations	Average Value	Std. Dev
AB	24.55	2.03	13033.81	100.59	2.57
NMQuiesce	32.11	1.60	44282.02	100.28	2.22

From this data we can observe that the null move quiescence search has a higher chance of finding the true value of the node. Also, the average difference and the standard deviation show that the calculated values are less dispersed in the null move quiescence search than in the pure alpha-beta algorithm. This behaviour can be appreciated more clearly in the following histograms:





However, this improvement of accuracy does not come for free. As expected, the null move quiescence search performs a considerable greater amount of static evaluations. The following two tables show the details of both algorithms at every depth.

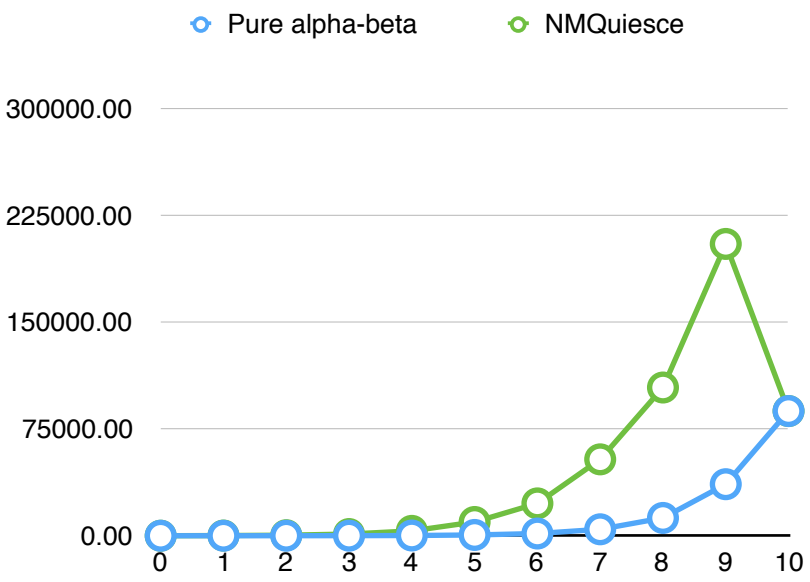
Pure alpha-beta - Depth details

Search depth	% of correct value	Avg. difference	Avg. of static evaluations	Mean	Std. Deviation
0	20.00	2.90	1.00	100.30	3.56
1	0.00	3.20	5.00	100.80	3.61
2	20.00	2.30	15.30	99.90	3.18
3	0.00	3.10	57.60	101.70	3.02
4	20.00	2.00	160.60	100.40	2.67
5	20.00	2.20	552.80	102.20	1.75
6	30.00	1.50	1558.20	99.70	2.11
7	20.00	1.80	4534.90	100.80	2.20
8	20.00	1.70	12518.80	99.90	2.23
9	20.00	1.60	36250.30	100.80	1.93
10	100.00	0.00	87717.40	100.00	0.00

NMQuiesce Statistics - Depth details

Search depth	% of correct value	Avg. difference	Avg. of static evaluations	Mean	Std. Deviation
0	20.00	2.20	21.80	101.40	2.63
1	10.00	2.40	137.20	99.00	3.09
2	30.00	1.80	384.80	101.00	2.31
3	20.00	2.10	1194.40	99.30	2.58
4	10.00	2.30	3424.70	102.10	1.73
5	40.00	0.90	9672.30	99.90	1.29
6	30.00	1.70	22746.10	101.10	2.02
7	30.00	1.50	53732.70	98.90	1.79
8	30.00	1.50	104265.50	101.30	1.64
9	40.00	1.00	205267.20	99.00	1.25
10	100.00	0.00	87717.40	100.00	0.00

The following graph shows how the number of static evaluations increased at every depth. As expected, the static evaluations performed at depth 10 are the same in both cases because there are no more nodes left to go deeper in the quiescence search.



In conclusion, this assignment allowed me to have a deeper understanding of the the negamax variant of alpha-beta and the null move quiescence search. Not only by understanding the specific series of steps required for each algorithm, but also by comparing the results obtained. Additionally, I got more acquainted with the SWIFT programming language —which I have been learning recently—.