

# **FOREST, user's guide**

**Othello® game**  
**by Olivier Casile**  
[forest\\_othello@orange.fr](mailto:forest_othello@orange.fr)

Version 9.3  
April 2024

## **Introduction**

FOREST is an Othello program which dates back to 1978. Initially, it was written in an obscure teaching language and was laboriously running on the mini-computer at my school, featuring a 16 bits CPU and 4 kb of RAM. The program's name, FOREST, is due to the large number of trees it conceals. The program evolved a lot in the late 90's, the golden age of Othello programs, and its version 3.7 came with lots of functionalities that made it successful among the community of Othello players. Then it entered a long pause, my professional activities having taken over that hobby.

But in 2017, with the resurgence of Artificial Intelligence, I thought Othello would be an excellent opportunity to apply *Deep Learning* techniques and tools. I started revamping FOREST, simplifying its over-complicated user interface, removing obsolete or un-necessary features, modernizing the alpha-beta and PVS algorithms, and tuning its old evaluation function while developing a new one based on a neural network. That long renovation work lasted about 18 months and led to the versions 4.0 to 4.5 of the program.

The development of the neural network took much longer than expected, because there is no rule to obtain a good compromise between accuracy, sophistication, and performance. All options must be tried and it takes time. The neural network is developed with the Keras and TensorFlow® deep learning frameworks, and then re-coded in C and C++ in the program to maximize performances at run-time. The supervised training phase is made on a library of more than one million games and 5 million positions of which the exact result is known. The version of FOREST that you installed uses a relatively sophisticated with a specialized loss function, combined with an alpha-beta search of depth 14, which provides strength similar to what traditional programs achieve with a 26 plies alpha-beta search. FOREST, which was an average player in its previous versions, has now caught up with strong programs such as Zebra, Edax, Ntest, or the recent Egaroucid.

Another area of progress is the opening library, which is essential to allow the program to reach the mid-game phase in a balanced position, and then make the difference before the end-game phase which is deterministic. Until FOREST 4.3, the opening library was based on the Thor database, which contains 100.000+ games played by human players and computers since the 80'. But the Thor database contains lots of sub-optimal games played by average human players or programs, which makes it difficult to extract a good opening library from it. Now, I am using a new Monte-Carlo algorithm with FOREST playing against itself.

The main functions offered by FOREST are:

- 6 playing levels, from beginner to master;
- Choice of classical openings;
- Choice of XOT openings;
- Large library of pre-computed positions in openings and end-games;
- Computer hints and display of the principal variation;
- Navigating through the moves of a game;
- Exchanging players roles;
- Changing level during a game;
- Import and export games;
- Simulation of moves and full evaluation of positions;

- Games rules and strategy;
- English and French languages.

## **Compatibility**

FOREST is a **64 bits** program requiring a PC with an Intel or AMD CPU, 4 Gb of RAM or more, and a Microsoft® Windows™ x64 operating system version 10 or more recent.

- Since its version 4.5, FOREST internally represents game board as Bitboards, et uses the **Popcnt** machine instruction to count discs.
- Since the version 5.2, it also uses the vector instruction set **AVX2**. The consequence of that evolution is that FOREST cannot anymore execute on processors not supporting those instructions (CPU's anterior to the Intel Haswell generation of 2012, and AMD processors anterior to the Excavator generation of 2015). FOREST is testing the support for the Popcnt and AVX2 instructions when it starts and will tell you if they are absent.
- At last, in its latest version, FOREST uses multi-threading to accelerate computations and augment its depth of analysis. It then requires a processor with at least 4 logical cores to execute.

Should your computer miss the above characteristics, you can still fall back to one of the a prior versions of FOREST, which run are available on my Web page <http://ocasile.pagesperso-orange.fr/forest.htm> .

## **Installation**

FOREST is a « portable » program not requiring any installation utility. It is generally available as a Zip file that you just need to expand in a folder of your choice. From that folder you can then directly execute the FOREST.exe file, or even define a shortcut to that executable file that you will put on your Windows desktop or pin to your task bar. FOREST uses the Windows registry to remember its position on the screen and the settings you will define.

Although I do not intentionally put any virus or adware in the FOREST executable, you should anyway double-check with your own anti-virus.

## **User's guide**

If you don't know anything about the rules of Othello, which are very simple, you should first have a look at its explanations on <http://www.britishothello.org.uk/rules.html>

If, however, this game is already familiar to you, just move the mouse cursor to the square of a legal move and click it. FOREST will flip the appropriate discs, play its turn, and wait for your next move. The game continues in this manner until its end, when FOREST will announce the final score. Please note that FOREST points out your legal moves by changing the cursor to a cross when it goes over the corresponding squares. Moreover, the color of this cross cursor indicates the color (black or white) of the disc you will place on the board. If, during the course of the game, a player must pass, FOREST will inform you in a pop-up dialog box.

After your first games at the default level, you will certainly start exploring the different functionalities of FOREST:

### **The menu bar**

- **Game** allows to start a new game or to quit the program.
- **Edit** allows to un-do the last round (the last move of the computer and your previous move) in order to re-play; to copy the game to the clipboard (in a text format starting with « F5D6... »); to import a game from the clipboard; to change the game level; or to modify the program settings.

- **View** allows adding or removing elements of the game display, such as the licit moves on the current game or the value of these moves in the library of pre-computed positions.
- **Information** can display a summary of the content of the library of pre-computed positions, detailed information on the neural network that evaluates positions and on the number of positions used for training, or general information on the program.

The tool bar proposes the following buttons:

- Starting a **new game** with a choice of options.
- **Inversion of players**, the computer plays immediately on the current board position, then it's your turn.
- **Centrale** symmetry of the board game.
- **Symmetry** of the board game relatively to the **downward diagonal**.
- **Symmetry** of the board game relatively to the **upward diagonal**.
- **Move back** to the **beginning** of the game.
- Move **one move back** in the game.
- **Move back one complete turn** (the last move of the computer and your own last move) so you play again on your previous position.
- Move **one move forward** within moves already played in the current game.
- **Move to the last move** already played in the current game.
- **Rapid evaluation** of available moves.

### Game options

- Choice of the **starting player**, yourself or FOREST.
- **Game level**: beginner, beginner +, amateur (default choice), amateur +, expert, and master.
- **Choice of opening**, i.e. choice of the sequence of the first moves: any (random), XOT (random balanced initial position), or one of the classic openings designated by its name.

### Program settings

- **Advise the best move** to the user, signaled by a small red circle around the center of the corresponding empty square.
- **Anticipation** of the opponent's move, which allows FOREST to start computing its next move while its opponent is thinking.
- Display **information on the moves** played by FOREST, such as the probability to win or its expected score.
- **Use or not**, by FOREST, of its **opening library** at the beginning of games, which significantly improves its strength.

### Standard openings

Since there are Othello tournaments, and as well as for other board games such as Chess, experienced players have learnt that certain openings (sequences of the initial moves in a game) are better than others, leading to more favorable positions. When looking at the history of games played in tournaments, we clearly see that the most favorable openings are the most used, and there exists a nomenclature, more or less official, of well-known openings, that you will find here: <http://samsoft.org.uk/reversi/openings.htm>

FOREST proposes about forty of these openings, amongst the most played, in the new game options. It will follow the sequence of moves of the selected opening as long as you will follow it, then will start playing using its normal evaluation algorithm.

### XOT openings

Good players having the tendency to always play the same good openings, sometimes up to the 20th move or more, tournaments begin being boring. It was therefore established, with the help of two excellent Othello programs (Edax and Ntest), a list of about 3600 openings that lead to a balanced position after 8 moves, i.e. leading to a position where neither of the players have an advantage. That list of openings, known as the « XOT » list, can be found at the address: <http://xot.xmav.eu>

FOREST includes that list and will load randomly any of these openings at your demand, then will let you play, either with the Black or White discs.

### Simulation and evaluation of moves

On the game position currently displayed by the program, a left click of the mouse allows you to play and to trigger automatically the next move of the computer. But a **right click on the mouse** is also possible to play a licit move without letting the computer play, which allows to replay known games, to force moves to study them, or to explore openings.

### Keyboard shortcuts

- **Ctrl-N** interrupts the current game to start a new one with the options you want.
- **Ctrl-U** un-does the last turn, i.e. the last move of the computer as well as your previous move, and allow you to re-play.
- **Ctrl-X** immediately loads a new XOT opening on the game, and lets you play. Should you want FOREST to play first, you can press the player inversion button on the tool bar.
- **Ctrl-C** copies the current game into the clipboard, in the form of a character string (format « F5D6... »), so you can import it in a text editor or into another program.
- **Ctrl-V** imports a game from the clipboard. The game must be represented as a character string (format « F5D6... »). FOREST will first check the validity of the imported character string, and you will play immediately on the imported game. Should you prefer that FOREST plays first, you can press the player inversion button on the tool bar.
- **Ctrl-S** produces a file containing the trace of the game up to the last move played.

### The trace file

At the end of every game, FOREST stores in its execution folder a trace file summarizing the game, named « Forest <date> <time>.log ». That file contains:

- The version number of FOREST.
- The date and time of the end of the game.
- The Windows name of the computer and of the user account.
- The number of logical cores of the processor, and the number of cores actually used by the program.
- The computing speed of FOREST, in moves and AI evaluations per second.
- The memory size of the transposition table, a cache of moves evaluated during iterative searches.
- The characteristics of the library of pre-computed positions.
- The AI strength level.
- The side played by FOREST, either Black (X) or White (O).
- The list of all moves played during the game, with detailed information on their evaluation by FOREST, and their principal variation.
- The final board with the number of Black and White discs.
- The character string representing the game.

Trace files are very useful to improve FOREST, and allow its author to tune FOREST, which can now learn of its successes and errors. Hence, I will greatly appreciate that you send me your trace files, in particular those were you won against FOREST at the master level, at the following e-mail address [forest\\_othello@orange.fr](mailto:forest_othello@orange.fr).

## Legal mentions

- *Othello* is a registered trademark of Kabushiki Kaisha MegaHouse
- *Reversi* is a registered trademark of Ravensburger AG
- *Windows* is a registered trademark of Microsoft Corporation
- *TensorFlow* is a trademark of Google Inc.

## License agreement

The program FOREST is developed by **Olivier Casile**. FOREST can be freely distributed but cannot be the subject of any commercial usage.

### Limited warranty and remedies:

*This program is provided "as is" without warranty of any kind, either express or implied, including, but not limited to warranties of merchantability or fitness for a particular purpose. In no event will the author be liable to you for any damages, including incidental or consequential damages, arising out of the use of the program, even if advised of the possibility of such damage.*

*You acknowledge that you have read this license, understand it and agree to be bound by its terms as the complete and exclusive statement of the agreement between us, superseding any proposal or prior agreement, oral or written, and any other communications between us relating to the subject matter of this license.*

Special thanks to:

- **Christophe Lanuit**, for his advice and technical discussions on algorithms and optimizations, its fast hashing function, for the game database that allowed me to start training FOREST, and for his elegant move generator based on the PEXT/PDEP instructions of x86 processors.
- **Matthias Berg** for his agreement to include his XOT openings list into FOREST.
- **Richard Delorme**, for his advices on parallelizing the alpha-beta algorithms, publishing and supporting the source code of his outstanding EDAX program ( <https://github.com/abulmo/edax-reversi> ), which serves as a sparring partner to FOREST.
- **Tom Kerrigan**, for his simplified version of the ABDADA algorithm (invented by **Jean-Christophe Weill**, <https://www.chessprogramming.org/ABDADA>), which largely inspired the « Lazy SMP » implementation in FOREST:  
[http://www.tckerrigan.com/Chess/Parallel\\_Search/Simplified\\_ABDADA/](http://www.tckerrigan.com/Chess/Parallel_Search/Simplified_ABDADA/).
- **Emmanuel Bougeard**, for his comparative tests of FOREST, and for his thorough study of openings.
- **Gunnar Andersson and Lars Ivansson**, for their excellent Zebra program, which also serves as a sparring partner to FOREST.
- **Chris Welty**, for his extraordinary opening book (<http://www.orbanova.com/nbook/>), that helps me validate the FOREST opening book.
- **Emmanuel Lazard**, for his agreement to distribute his « *A la découverte d'Othello* » strategy guide with FOREST.
- **Bruno de la Boiserie** for his advice on Othello strategy, for distributing FOREST in its early days, and for his comprehensive « *Anthologie des programmes d'Othello* ».

I will be very pleased to receive your comments, suggestions, bug reports, and game logs at:  
[forest\\_othello@orange.fr](mailto:forest_othello@orange.fr) .

St Laurent du Var, France, April 29<sup>th</sup> 2024