

[Home](#) * Getting Started

Just **getting started** with a new engine? Congratulations! You are in for a lot of fun, but there is a considerable amount of work ahead of you.

The foundation of a chess engine is the [board representation](#). This is the "back end" for the chess engine, which controls how it keeps track of the board and the rules of the game. The **very first step** to writing a chess engine is to write a complete, [bug](#) free board representation that knows **every** rule of chess. While this can sometimes be a pain, especially implementing the more complicated rules such as [castling](#) and [repetition draws](#), it is the backbone of the chess engine, and your engine will not get far without it.

When writing an engine, it is **extremely** important to write bug free code. The best strategy when starting a new engine is to create a debugging framework under it so that every single piece of code gets tested, no matter how simple it looks. Many experienced engine authors have ended up rewriting their engines because they have become unmanageable due to bugs.

Once you have a nice solid foundation ready, you are ready to start learning about the fun stuff: [Search](#) and [Evaluation](#). These are the "brains" behind a chess engine; what allows it to pick a good move.

You'll probably also want to connect your program to a [GUI](#) (Graphical User Interface). You are in luck, though; you don't have to write your own. It is only necessary to understand some basic text commands in order to communicate with the many available GUIs today. Most engines use either the [Chess Engine Communication Protocol](#) or [Universal Chess Interface](#) for this communication.

If you want ideas and see how other programmers have done things, take a look at some of the [Open Source Engines](#). These can be very helpful when translating rather vague algorithms into specific data structures and code. Just be careful, and don't copy the code and say it is your own! [Clones](#) are frowned upon by the computer chess community as a whole.

It is also a very good idea to join some of these [Computer Chess Forums](#). The chess programming community is very friendly and will help you out with personalized advice. We are always happy to accept new members!

Some other good computer chess references can be found in [Recommended Reading](#).

[Up one level](#)