

[Home](#) \* [Engines](#) \* **Mr. Turk**



[Gaughan's](#) reconstructed Turk <sup>[2]</sup>

**Mr. Turk**,  
an early chess program written in [Fortran](#) by [Gary Boos](#) and [James Mundstock](#) from the [University of Minnesota](#) on a [CDC 6600](#), which participated in the [ACM's Second North American Computer-Chess Championship](#) 1971. It did not use [alpha-beta](#), but a search based on a *Multipurpose, Theorem-Proving Heuristic Program* as described by [James R. Slagle](#) and [Philip Bursky](#) in 1968 <sup>[1]</sup>.

## Table of Contents

[Description](#)

[See also](#)

[External Links](#)

[References](#)

[What links here?](#)

## Description

by [Gary J. Boos](#) from [Ben Mittman's](#) 1971 Panel <sup>[3]</sup>:

Since late 1967 [James Mundstock](#), myself, and others, have been working on our chess program, *Mr. Turk*. *Mr. Turk* was developed at the University of Minnesota on a CDC 6600. At almost all times everyone working on the program was both a chess player and a reasonably good FORTRAN programmer. Our main goal has been to produce a program that could win as many chess games as possible. The methods used in striving for this goal have varied from group to group.

Based upon our chess experience (Mundstock is a class B player, and I am an experienced, class A player), we knew that to obtain a winning

position, one normally has to outplay the opponent in both the opening and the middle game. Consequently, we concentrated our initial efforts on writing good [evaluation](#) routines for the [opening](#) and [middlegame](#), plus producing routines that supplied [legal moves](#), location of [pinned pieces](#), which squares are attacked by which pieces, etc. The result was a program that would produce reasonable moves 80 to 90% of the time, with (in effect) a 1-ply level lookahead.

The next major task was to write a lookahead routine and incorporate it into the rest of the program. Existing lookahead routines were not impressive. They tended to have a vast number of limbs in their move tree, and very little evaluation was spent on the positions examined. An experienced human chess player selects variations with an efficiency at least 10 times greater than any pre-1970 program. Mundstock and I felt that any attempt to approach a master's thought process should help in shaping and improving the move tree. The most noticeable difference in previous lookahead routines and a master's analysis is that the master analyzes one and only one line at a time. He seems to try to insure that that line is the best for both sides, and he attempts to analyze it as deeply as his vision and time permit.

Mundstock noticed an article by [Slagle](#) and [Bursky](#) in the Journal of the ACM, that pointed toward an algorithm that seemed better than alpha-beta pruning. Based upon this article, and guided by Mundstock, I wrote a lookahead routine whose main theme is that the best line is analyzed until it is shown that it is no longer the best line.

This process eliminates many common problems that accompany a fixed depth search, one of which is the [Ostrich Effect](#) which existed in even [Northwestern University's Chess 3.0](#). Tests showed that in a small set of positions, Mr. Turk could find the main variation on the first try. We believe that the basic theme of our lookahead routine is better than alpha-beta pruning.

Full scale tests of the program revealed serious shortcomings. Mr. Turk had only a [fixed width](#) (5 moves) move tree, and when all of the top 5 moves were bad (often twice a game), the program was forced to play the best of those 5. That is to say, we had no fall-back routine; no panic button to push.

Other weaknesses were: a) the inability to include sacrificial moves in the move tree, b) little endgame capability, and c) only a small opening book. Nevertheless, we challenged five other programs to postal matches. Only Northwestern University was capable of playing. The match was started in late 1970, and Chess 3.2 is presently winning the two game match. Our team has been working on the above weaknesses

since September 1970, and also performing a major overhaul on the existing code. We hope to be able to include tactical moves in the move tree, provide a fall-back algorithm, enlarge and improve our [opening book](#) and still keep the necessary storage at under 54k.

It is our opinion that existing chess programs have many weaknesses, and that their play is far too often superficial. Almost all programs find it very difficult to win an endgame if positional play is of the essence. Most programs have opening books, but I seriously doubt that any can handle [transpositions](#). I have never seen a program sacrifice material unless either checkmate or a net win of material was within its view. Also (and this is probably the hardest task of all) no program has been able to develop a logical plan of play in a general position. With the aid of other chess players in Minnesota, we have developed a secret weapon for the [Second ACM tournament](#), and will attempt to exploit one of these weaknesses.

The Second ACM tournament will be far stronger than the [1970 championship](#) (how much stronger will be indicated by where Chess 3.5 finishes). The tournament will provide very interesting games, and the panel discussions between chess programmers from across the nation will bring forth new ideas. We must learn all the lessons we can, for next year, the Russians are coming.

## See also

- [Iron Fish](#)
- [Kempelen](#)
- [The Baron](#)
- [The Turk](#)

## External Links

- [The Turk from Wikipedia](#), the historic fake chess-playing machine
- [Ein Türke in Paderborn](#) from [ChessBase Nachrichten](#) (German)

## References

1. <sup>^</sup> [James R. Slagle](#), [Philip Bursky](#) (1968). *Experiments With a Multipurpose, Theorem-Proving Heuristic Program*. [Journal of the ACM](#), Vol. 15, No. 1
2. <sup>^</sup> Source image with [L. Stephen Coles](#): <http://www.grg.org/images/TurkSColes.jpg> This is a derivative, released under the [GFDL](#) and [CC](#), [Wikimedia Commons](#)

3. <sup>^</sup> [Ben Mittman](#) (1971). [Computer Chess Programs \(Panel\)](#). pdf from [The Computer History Museum](#)

## What links here?

Page	Date Edited
<a href="#">ACM 1971</a>	Jul 19, 2016
<a href="#">CDC 6600</a>	Oct 8, 2014
<a href="#">Engines</a>	Mar 10, 2018
<a href="#">Gary Boos</a>	Apr 10, 2013
<a href="#">Horizon Effect</a>	Sep 10, 2017
<a href="#">Iron Fish</a>	Sep 8, 2016
<a href="#">James Mundstock</a>	Apr 10, 2013
<a href="#">Mr. Turk</a>	Feb 21, 2015
<a href="#">Raspberry Turk</a>	Sep 7, 2017
<a href="#">The Turk</a>	Apr 1, 2017
<a href="#">The Turk (TR)</a>	Apr 1, 2017

[Up one level](#)