

## Table of Contents

[Pattern Matching](#)

[Evaluation](#)

[Partial](#)

[Global](#)

[Search](#)

[See also](#)

[Publications](#)

[Forum Posts](#)

[References](#)

[What links here?](#)

[Home](#) \* [Engines](#) \* **PawnKing**

### **PawnKing,**

a [knowledge](#) based chess program by [Helmut Horacek](#) for the domain of [pawn endgames](#). PawnKing was written in [Pascal](#) and subject of research in Horacek's Ph.D. thesis, co-supervised by [Wilhelm Barth](#) from [Vienna University of Technology](#) and [Curt Christian](#) from [University of Vienna](#) <sup>[1]</sup>. Patterns serve to recognize important positional components in a given position, and the presence of several configurations of positional elements suggests application of [evaluation](#) and appropriate plans selecting useful moves to be [searched](#).

## Pattern Matching

A pattern is composed of three boards, a **fixed-board**, a **not-board**, and a **variable-board**. In the pawn endgame, a [quad-bitboard](#) for pawns and kings of both sides is sufficient. For a pattern match, the fixed-board must be subset of the current board, the intersection of the current board with the not-board must be empty, with the variable-board not empty.

```
bool QBB::match(const QBB & fix, const QBB & not, const QBB & var)
```

```
const {
  if ((*this & fix) != *this) return false;
  if ((*this & not) != QBB(0)) return false;
  if ((*this & var) == QBB(0)) return false;
  return true;
}
```

The pattern match generally is performed for both sides (with appropriate changes for the black side) as well as various parts of the board (shifting the pattern by some ranks and files). Finally, some important pawns and key squares of a successfully matched pattern are given identities for further reference by piece paths and evaluation. A square can receive several identities, e.g. a pawn can be both [passed](#) and [protected](#). However, instead of such generalized pattern matching, it seems appropriate to predetermine set-wise instances of [passed pawns](#), [candidates](#), [backward pawns](#) etc. with the help of [pawn-](#) and [attack spans](#).

## Evaluation

### Partial

The knowledge discovered by patterns alone is not sufficient to obtain a value which accurately enough represents a certain positional element. Classifications are performed for all relevant positional elements, such as [passed pawns](#), [candidate pawns](#), [weak pawns](#) and [levers](#) inside a **partial** evaluation. The basic elements building up a partial evaluation are [distances](#) between pairs of squares which are referenced by the identities created by the pattern match. A well defined set of distance types is available, [rank](#), [file](#), diagonal, [pawn path](#) and [king path](#) distance. A so called **sensitive value** expresses to what extent the most important distance values may change without hurting the successful check of the conditions. A **goal distance** represents the necessary number of moves to accomplish the goal, which is the aim of the partial evaluation. The best hierarchical values per classifications, a tuple of the conditions along with its sensitive value and goal distance, are saved for the **global** evaluation.

### Global

In contradiction to conventional evaluation, PawnKing's value range consists of a few discrete values: decisive, big and slight advantage for White or Black respectively, as well as equality. To satisfy attributes like *unclear*, *with compensation*, and the dynamics not yet evaluable by static means, **two** values of that kind are included in the global value, an optimistic and a pessimistic, with a special defined set of relational operators considering average of optimistic and pessimistic value and further best hierarchical values.

## Search

PawnKing uses a [depth-first alpha-beta](#) minimax search without [iterations](#). Except core restrictions, a [depth](#) limit is not applied. [Move ordering](#) considers [promotions](#), [captures](#), moves protecting a hung pawn, or an evasion with such a pawn, as well as a classification, sensitivity value and goal distance which where calculated from the square the plan of the move belongs to. All created positions including their value, as well as the sensibility of the key squares all over the board, are saved in a [table](#), so when it arises again due to a [transposition](#), its value is taken directly from the table. If only the same pawn structure occurs, it is checked whether the king locations does not hurt the sensibility of any key square.

## See also

- [Chunker](#)
- [King Pattern](#)
- [Pawn Pattern and Properties](#)
- [Pattern Recognition](#)
- [Peasant](#)

## Publications

- [Helmut Horacek](#) (1982). *Ein Modell für Schachbauernendspiele mit menschlichen Problemlösungstechniken*. Ph.D. thesis, Department of Computer Science, [University of Vienna](#), supervisors [Curt Christian](#) (University of Vienna) and [Wilhelm Barth](#) ([Vienna University of Technology](#)).
- [Helmut Horacek](#) (1982). *Construction of Planned Move Trees for Chess Pawn Endings*. Sigart Newsletter 80, pp. 163-168.
- [Helmut Horacek](#) (1983). *Knowledge-Based Move Selection and Evaluation to Guide the Search in Chess Pawn Endings*. [ICCA Journal](#), Vol. 6, No. 3
- [David E. Welsh](#), [Boris Baczynskyj](#) (1985). [Computer Chess II](#). William C Brown Publications, ISBN-13: 978-0697099112 <sup>[2]</sup> <sup>[3]</sup>

## Forum Posts

- [PAWNKING](#) by José de Jesús García Ruvalcaba, [CCC](#), August 14, 1998

## References

1. <sup>^</sup> [Helmut Horacek](#) (1982). *Ein Modell für Schachbauernendspiele mit menschlichen Problemlösungstechniken*. Ph.D. thesis, Department of Computer Science, [University of Vienna](#), supervisors [Curt Christian](#) (University of Vienna) and [Wilhelm Barth](#) ([Vienna University of Technology](#))
2. <sup>^</sup> [PAWNKING](#) by José de Jesús García Ruvalcaba, [CCC](#), August 14, 1998
3. <sup>^</sup> [An interesting book with some insights on Bob's Cray Blitz](#) by [Julien Marcel](#), [CCC](#), June 23, 2011

## What links here?

| Page  | Date Edited  |
|---|--------------|
| <a href="#">Chunker</a>                         | Jul 12, 2016 |
| <a href="#">Engines</a>                         | Mar 10, 2018 |
| <a href="#">Helmut Horacek</a>                  | Oct 31, 2016 |
| <a href="#">Mathematician</a>                   | Apr 9, 2018  |
| <a href="#">Pawn Endgame</a>                    | Oct 11, 2017 |
| <a href="#">Pawn Pattern and Properties</a>     | Nov 11, 2017 |
| <a href="#">PawnKing</a>                        | Sep 5, 2014  |
| <a href="#">Peasant</a>                         | May 21, 2015 |
| <a href="#">Vienna University of Technology</a> | Dec 7, 2017  |

[Up one Level](#)