
blunderDB

Release 0.10.1

Kevin UNGER <blunderdb@proton.me>

Mar 01, 2026

CONTENTS

1	Version history	3
2	Table of contents	5
2.1	Download and Installation	5
2.2	Manual	6
2.3	User Guide	9
2.4	List of commands	15
2.5	Keyboard shortcuts	19
2.6	Frequently Asked Questions (FAQ)	21
2.7	Annex: Advanced Filter Usage	23
2.8	Windows Annex: False Detection of blunderDB as Malware	25
2.9	Mac Annex: Possible Blocking of blunderDB	27
2.10	Annex: Database Schema	35
3	Contact	39
4	Donate	41
5	Acknowledgments	43

blunderDB is a software for creating backgammon position databases. Its main strength is to provide a single place where a player can aggregate the positions he or she has encountered (online, in tournaments) and be able to review these positions by filtering them with various filters that can be arbitrarily combined. blunderDB can also be used to create catalogs of reference positions.

The present documentation is structured as follows:

- the **download and installation** section explains how to obtain and run blunderDB.
- the **manual** describes the general functioning of blunderDB and how it should be used.
- the **user guide** is a practical introduction for quickly using blunderDB.
- the list of **commands** as well as the list of **keyboard shortcuts** enables efficient use of blunderDB.
- The **FAQ** provides answers to the most frequently asked questions.

VERSION HISTORY

Version	Date	Cause and/or nature of changes
0.1.0	December 31, 2024	Beta version
0.2.0	January 6, 2025	Various bug fixes. Addition of match/TP/GV tables. Added search filters (moves, cube decisions, date). Added metadata for positions. Import/export functionality between blunderDB instances. Added metadata functionality for databases. Introduction of version numbers (database and application).
0.3.0	January 27, 2025	Various bug fixes. Automatically saves the window size. Imports any comments from XG.
0.4.0	February 3, 2025	Various bug fixes. Adding an icon for blunderDB. Filter corrections. Adding macOS support.
0.5.0	February 4, 2025	Addition of new filters (mirror, non-contact, jan blot, outfield blot).
0.6.0	February 13, 2025	Add filter library. Displaying the database version in the metadata.
0.7.0	February 16, 2025	Support Japanese and German XG exports.
0.8.0	May 3, 2025	New button to hide pipcount. Button to load random position.
0.9.0	November 02, 2025	Fix filter library bug. Database import/export. Display arrows for selected moves. Keyboard shortcuts for import/export.
0.10.0	25/02/2026	Match import from eXtreme Gammon (XG), GNUbg (SGF), Jellyfish (MAT/TXT) and BGBlitz (BGF/TXT). MATCH mode: navigate through moves of an imported match, with played move highlighting. Match panel: list, sort, inline editing, player swap, tournament assignment. Recursive folder import and drag & drop import. EPC (Effective Pip Count) calculator with built-in GNUbg bearoff database. Collections: custom position grouping with dedicated COLLECTION mode. Tournaments: group matches by event. Save and restore session state (last search, current position). Automatic database schema migration. Multi-engine display in analysis. Player 1 error/blunder filter in searches. Database export with granular selection (matches, collections, tournaments, played moves). Toggle button for MATCH mode. Pipcount display in MATCH mode.

TABLE OF CONTENTS

2.1 Download and Installation

blunderDB is a single executable that requires no installation.

The latest version of blunderDB is available under the MIT license:

- for Windows: <https://github.com/keving/blunderDB/releases/latest/download/blunderDB-windows-0.10.1.exe>
- for Linux: <https://github.com/keving/blunderDB/releases/latest/download/blunderDB-linux-0.10.1>
- for Mac: <https://github.com/keving/blunderDB/releases/latest/download/blunderDB-macos-0.10.1.zip>

Note

blunderDB uses Webview2 for rendering the graphical interface. It is likely that Webview2 is already present natively on your operating system. If not, the first execution of blunderDB will offer to download and install it. No user action is required.

Note

On Linux, if blunderDB is not executable after downloading, run the command `chmod +x ./blunderDB-linux-x.y.z` in a terminal, where x, y, z corresponds to the downloaded version.

Note

On Linux, if you get the error `libwebkit2gtk-4.0.so.37: cannot open shared object file, your distribution uses webkit2gtk-4.1 instead of webkit2gtk-4.0`. Download the dedicated build: <https://github.com/keving/blunderDB/releases/latest/download/blunderDB-linux-webkit2gtk-4.1-0.10.1>

Warning

On Windows, it is possible that it may hesitate to run blunderDB. See: [Section 2.8](#) for an explanation of why and how to bypass any potential blocks.

Warning

On Mac, it is possible that it may have reservations about running blunderDB. See [Section 2.9](#) to understand why and bypass any potential blocks.

2.2 Manual

2.2.1 Introduction

blunderDB is software for creating backgammon position databases. Its main strength is to provide a single place to aggregate positions that a player has encountered (online, in tournaments) and to be able to re-study these positions by filtering them according to various arbitrarily combinable filters. blunderDB can also be used to create catalogs of reference positions.

Positions are stored in a database represented by a *.db* file.

2.2.2 Main Interactions

The main interactions possible with blunderDB are:

- add a new position,
- modify an existing position,
- delete an existing position,
- search for one or more positions,
- import matches from various sources (XG, GNUbg, BGBlitz, Jellyfish),
- navigate through the moves of an imported match,
- organize positions into collections,
- organize matches into tournaments.

2.2.3 Modes

To do this, the user switches to dedicated modes for:

- navigation and viewing positions (NORMAL mode),
- editing positions (EDIT mode),
- editing a query to filter positions (COMMAND mode or search window),
- navigating through the moves of an imported match (MATCH mode),
- computing the EPC (Effective Pip Count) from a position (EPC mode),
- browsing a collection of positions (COLLECTION mode).

The user can freely tag positions and annotate them with comments.

In the following sections of the manual, the graphical interface and the main modes of blunderDB are described.

2.2.4 Description of the interface

The interface of blunderDB is composed, from top to bottom, of:

- [top] the toolbar, which gathers all the main operations that can be performed on the database,
- [in the middle] the main display area, which allows for displaying or editing backgammon positions,
- [at the bottom] the status bar, which provides various information about the database or the current position.

Panels can be displayed to:

- display the analysis data associated with the current position from eXtreme Gammon (XG), GNUbg, or BGBlitz,
- display, add, or modify comments,
- display the list of imported matches (match panel),
- display and manage position collections (collections panel),
- display and manage tournaments (tournaments panel),
- display the filter library,
- display the search history.

Modal windows can be displayed to:

- [EDIT mode only] set search filters,
- display the blunderDB help,
- configure database export settings,
- display the database metadata.

The main display area provides the user with:

- a board to display or edit a backgammon position,
- the level and owner of the cube,
- the pip count of each player,
- the score of each player,
- the dice to play. If no values are displayed on the dice, the position of the dice indicates which player has the turn and that the position is a cube decision.

The status bar is structured from left to right with the following information:

- the current mode (NORMAL, EDIT, COMMAND),
- an informational message related to an operation performed by the user,
- the index of the current position, followed by the number of positions in the current library.

Note

In the case of positions resulting from a user search, the number of positions indicated in the status bar corresponds to the number of filtered positions.

2.2.5 NORMAL mode

NORMAL mode is the default mode of blunderDB. It is used for:

- scrolling through the different positions in the current library,
- displaying the analysis information associated with a position.
- displaying, adding, and modifying comments on a position.

Tip

Refer to *Keyboard shortcuts* for available shortcuts.

2.2.6 EDIT mode

EDIT mode allows you to edit a position with the option to either add it to the database or define the type of position to search for. EDIT mode is activated by pressing the *TAB* key. The distribution of checkers, the cube, the score, and the turn can be modified using the mouse (see *Edit a position*).

Tip

Refer to *Keyboard shortcuts* for available shortcuts.

2.2.7 COMMAND mode

COMMAND mode allows you to perform all the functionalities of blunderDB available in the graphical interface: general operations on the database, position navigation, displaying analysis and/or comments, searching for positions based on filters. ... After getting familiar with the interface, it is recommended to gradually use this mode for a powerful and smooth use of blunderDB, especially for position search functionalities.

To switch to COMMAND mode from any other mode, press the *SPACE* key. To submit a query and exit COMMAND mode, press the *ENTER* key.

blunderDB executes the queries sent by the user as long as they are valid and immediately modifies the state of the database if necessary. There are no explicit save actions required from the user.

Tip

Refer to *Section 2.4* for the list of available commands in COMMAND mode.

2.2.8 MATCH mode

MATCH mode allows you to navigate through the moves of an imported match. It is activated by pressing *CTRL-TAB* or by executing the *m* command.

In this mode, the user can:

- browse through the moves of a match using the *LEFT* and *RIGHT* keys,
- switch between games using the *PageUp* and *PageDown* keys,
- display the move analysis (checker and cube) by pressing *CTRL-L*,
- toggle between checker move and cube analysis with the *d* key,
- see the actually played move highlighted in the analysis.

When the user enters MATCH mode, the last visited match is automatically loaded. The last visited position in each match is saved and restored.

Tip

Refer to *Keyboard shortcuts* for available shortcuts.

2.2.9 EPC mode

EPC mode allows you to calculate the EPC (Effective Pip Count) of a bearoff position. It is activated by executing the `epc` command or by clicking the corresponding button in the toolbar.

In this mode, the user edits the checker positions in the home board (last 6 points) and the following information is computed in real time for each player:

- the EPC (Effective Pip Count),
- the average number of rolls needed (Mean Rolls),
- the standard deviation,
- the pip count,
- the wastage (difference between the EPC and the pip count).

Note

The computation relies on the built-in 6-point bearoff database from GNUbg.

2.2.10 COLLECTION mode

COLLECTION mode allows you to browse the positions of a collection. It is activated by double-clicking on a collection in the collections panel (*CTRL-K*).

In this mode, the user can navigate through the positions of the collection using the *LEFT* and *RIGHT* keys. To exit COLLECTION mode and return to NORMAL mode, press *TAB*.

2.3 User Guide

This guide is a practical introduction to pick up quickly blunderDB.

2.3.1 Create a new database

To create a new database, click on the “New Database” button in the toolbar. Choose a path to save the database, as well as a name, and click “Save”.

Note

The file extension for blunderDB databases is *.db*.

Tip

Keyboard shortcuts: *CTRL-N*. Command: `n`

2.3.2 Open an existing database

To load an existing database, click on the “Open Database” button in the toolbar. Navigate to the path where the database is located, select the *.db* file, and click “Open.”

Tip

Keyboard shortcuts: *CTRL-O*. Command: *o*

2.3.3 Import and merge a database

To import and merge another blunderDB database into the currently open database, click on the “Import Database” button in the toolbar. Select the *.db* file to import and click “Open.”

blunderDB will intelligently merge the two databases:

- Positions that do not exist in the current database will be added with their analyses and comments.
- Positions that already exist will be updated: analyses will be completed if missing, and comments will be merged (adding new comments without duplicating existing ones).
- A message will summarize the number of positions added, merged, and ignored.

Note

The import requires that both databases have compatible schema versions. It is possible to import a database with a lower or equal version into a database with a higher version.

Caution

The import operation immediately modifies the currently open database. It is recommended to make a backup copy before importing another database.

2.3.4 Edit a position

To edit a position, switch to EDIT mode by pressing the *TAB* key. Edit the position with the mouse:

- Click on the points to add checkers. A left-click assigns checkers to player 1. A right-click assigns checkers to player 2. To insert a prime, click on the starting point, hold down the button, and release on the endpoint. Click on the bar to place checkers in the bar.
- To clear the position, double-click on an empty area outside the board or press the *BACKSPACE* key.
- To send the cube to Player 1, left-click on the cube. To send the cube to Player 2, right-click on the cube.
- To indicate the player who is to move, click on the designated dice area.
- To edit the dice, left-click to increase the value of a die, right-click to decrease the value of a die. If the dice faces are empty, it means the position is a cube decision.
- To edit the players’ score, left-click to increase the score, right-click to decrease the score.

Tip

The input of the position with the mouse for the checkers is done in the same way as in XG.

2.3.5 Add a position to the database

After editing the previous position, blunderDB is in EDIT mode.

To save the previously obtained position, press *CTRL-S* or click the “Save Position” button in the toolbar.

Tip

From EDIT mode, switch to COMMAND mode and execute: *w*

2.3.6 Tag a position

To add a tag *toto* to the current position, switch to COMMAND mode by pressing *SPACE*, type *#toto*, and confirm the command by pressing *ENTER*.

2.3.7 Delete a position

To delete the current position from the database, press *Del* or click the “Delete Position” button in the toolbar.

Tip

In COMMAND mode, execute *d*.

Caution

The deletion of the position is final and does not require any user confirmation.

2.3.8 Import a position from XG

To import a position directly from XG,

1. display the position to import in XG and press *CTRL-C*,
2. open blunderDB and press *CTRL-V*.

Note

The automatic paste detects the source format (XG, GNUbg, BGBlitz).

2.3.9 Import a match

blunderDB can import matches from various sources.

Supported formats:

- eXtreme Gammon (XG): *.xg* files
- GNUbg: *.sgf* files
- Jellyfish: *.mat* and *.txt* files
- BGBlitz: *.bgf* and *.txt* files

To import one or more match files:

1. Press *CTRL-I* or click the “Import” button in the toolbar.
2. Select one or more files to import.
3. blunderDB automatically detects the format and imports the match.
4. A progress window shows the number of imported, failed, and skipped (duplicate) files.

Tip

Command: *i*

Note

blunderDB automatically detects duplicates and prevents importing a match already present in the database.

2.3.10 Import a folder of matches

To recursively import all match files contained in a folder and its subfolders:

1. Press *CTRL-SHIFT-F* or click the corresponding button in the toolbar.
2. Select the folder containing the match files.
3. blunderDB automatically collects and imports all recognized files (*.xg*, *.sgf*, *.mat*, *.txt*, *.bgf*).

2.3.11 Drag and drop

blunderDB supports drag and drop. You can drag and drop onto the blunderDB window:

- match or position files (*.xg*, *.sgf*, *.mat*, *.txt*, *.bgf*) to import them,
- database files (*.db*) to open or merge them with the current database,
- folders to recursively import all the files they contain.

2.3.12 Navigate through a match

To navigate through an imported match:

1. Open the match panel with *CTRL-T*.
2. Double-click on a match to enter MATCH mode.
3. Use the *LEFT / RIGHT* keys to browse through moves.
4. Use *PageUp / PageDown* to switch between games.
5. Press *CTRL-L* to display the analysis.
6. Press *d* to toggle between checker move analysis and cube analysis.

Tip

Shortcut: *CTRL-TAB* to toggle MATCH mode / exit MATCH mode. Command: *m*

Note

blunderDB remembers the last visited position in each match. When returning to a match, the last visited position is automatically restored.

2.3.13 Manage the match panel

The match panel (*CTRL-T*) allows you to:

- list all imported matches (sorted from most recent to oldest),
- sort matches by column (player 1, player 2, date, match length, tournament),
- edit player names or date by double-clicking on the fields,
- swap player 1 and player 2 using the swap button,
- assign a match to a tournament,
- delete a match using the *Del* key.

2.3.14 Manage collections

Collections allow you to organize positions into custom groups. To access the collections panel, press *CTRL-K*.

Create a collection:

1. Open the collections panel (*CTRL-K*).
2. Enter the name of the new collection and click “Add”.

Add positions to a collection:

1. Select the desired positions.
2. Add them to the collection from the collections panel.

Browse a collection:

- Double-click on a collection to enter COLLECTION mode and browse its positions.

Tip

Command: coll

2.3.15 Manage tournaments

Tournaments allow you to organize imported matches by event. To access the tournaments panel, press *CTRL-Y*.

Create a tournament:

1. Open the tournaments panel (*CTRL-Y*).
2. Click “Add” and enter the tournament name.

Assign a match to a tournament:

- From the match panel (*CTRL-T*), use the dropdown menu in the tournament column to assign a match.

2.3.16 Calculate the EPC

The EPC (Effective Pip Count) calculator allows you to compute the bearoff statistics of a position.

1. Execute the `epc` command or click the corresponding button in the toolbar.
2. Edit the checker positions in the home board (last 6 points).
3. Results are displayed in real time: EPC, average number of rolls, standard deviation, pip count, and wastage.

Note

The calculator works for both players simultaneously.

2.3.17 Display the analysis of a position imported from XG

If a position analyzed by XG, GNUbg, or BGBlitz has been imported into blunderDB, the analysis can be displayed by pressing *CTRL-L*.

If the position corresponds to a checker decision, the five best moves are displayed on separate lines. For each line, the information provided is in this order: the associated checker move, the normalized equity, the error in equity of the move, the winning chances, gammon, and backgammon for the player, and the winning chances, gammon, and backgammon for the opponent, along with the level of analysis.

If the position corresponds to a cube decision, the cost of each decision is displayed along with the winning chances of the position.

When multiple analysis engines are present for the same position (for example XG and GNUbg), an additional column indicates the source engine of each analysis.

In *MATCH* mode, the actually played move is highlighted in the move list. In *NORMAL* mode, if the position was encountered in multiple matches, all played moves are indicated.

Tip

By clicking on a move in the analysis panel, the corresponding arrows are displayed on the board.

2.3.18 Export a position to XG

To export a position from blunderDB to XG,

1. display the position to export in blunderDB and press *CTRL-C*,
2. open XG and press *CTRL-V*.

2.3.19 View the different positions

To view the different positions in the current library, use the *LEFT* and *RIGHT* keys. The *HOME* key allows you to go to the first position, and the *END* key lets you go to the last position.

To display the bearoff on the left, press *CTRL-LEFT*. To display the bearoff on the right, press *CTRL-RIGHT*.

2.3.20 Search for positions based on criteria

To search for types of positions,

- switch to *EDIT* mode by pressing *TAB*,

- edit the structure of the position to search for. blunderDB will filter positions that have at least the entered checker structure. If unsure, to maximize results, clear the position by pressing the *BACKSPACE* key. Edit the cube position and the score if necessary.

Method 1 (simple):

- Open the search window (*CTRL-F*)
- Add and set the search filters
- Confirm by clicking on “Search”.

Method 2 (advanced):

- switch to COMMAND mode by pressing *SPACE*,
- type *s*, and add any additional filters (for example, *cube* or *score* to consider the cube and score, respectively. See [Section 2.4.4](#) for a comprehensive list of available filters).
- confirm the request by pressing *ENTER*.

The displayed positions are those from the database that meet the search criteria entered by the user.

2.4 List of commands

2.4.1 Global operations

Command	Action
new, ne, n	Create a new database.
open, op, o	Open an existing database.
import_db, idb	Import and merge another database.
export_db, edb	Export the current selection to a new database.
quit, q	Close blunderDB.
help, he, h	Open blunderDB help.
meta	Display database metadata.
epc	Open the EPC (Effective Pip Count) calculator.
met	Open the Kazaross-XG2 match equity table.
tp2	Open the takepoint table with a 2-cube.
tp2_live	Open the takepoint table with a 2-cube for long race positions.
tp2_last	Open the takepoint table with a 2-cube for last roll positions.
tp4	Open the takepoint table with a 4-cube.
tp4_live	Open the takepoint table with a 4-cube for long race positions.
tp4_last	Open the takepoint table with a 4-cube for last roll positions.
gv1	Open the gammon value table with a 1-cube.
gv2	Open the gammon value table with a 2-cube.
gv4	Open the gammon value table with a 4-cube.

2.4.2 NORMAL Mode

Command	Action
import, i	Import one or more positions/matches from file (xg, sgf, mat, txt, bgf).
delete, del, d	Delete the current position.
[number]	Go to the specified index position.
list, l	Show the analysis of the current position.
comment, co	Show/write comments.
filter, fl	Show/hide the filter library.
history, hi	Show/hide the search history.
match, ma	Show/hide the match panel.
collection, coll	Show/hide the collections panel.
#tag1 tag2 ...	Tag the current position.
e	Load all positions from the database.
m	Toggle MATCH mode (last visited match).

2.4.3 EDIT Mode

Command	Action
write, wr, w	Save the current position.
write!, wr!, w!	Update the current position.
s	Search for positions with filters.

2.4.4 Search Filters

The filters below must be juxtaposed during a search, i.e., after the start of the s command.

Warning

In the position search, by default, blunderDB takes into account the current checker structure, ignoring the position of the cube, the score, and the dice. To consider the position of the cube, the score, and the dice, it must be explicitly mentioned in the search.

Note

blunderDB considers that a backchecker is a checker located between point 24 and point 19.

Note

blunderDB considers that the number of checkers in the zone is the number of checkers located between point 12 and point 1.

Note

blunderDB considers the outfield to extend between point 18 and point 7.

Note

blunderDB considers the jan to extend between point 1 and point 6.

Tip

The parameters for filtering positions can be combined arbitrarily.

Query	Action
cube, cub, cu, c	The position checks the cube configuration.
score, sco, sc, s	The position checks the score.
d	The position checks the dice or the cube decision.
D	The position checks the dice roll.
nc	The position has no contact.
M	The position or the mirror one meets the filters.
p>x	The player has at least x pips behind in the race.
p<x	The player has at most x pips behind in the race.
px,y	The player has between x and y pips behind in the race.
P>x	The player has a race of at least x pips.
P<x	The player has a race of at most x pips.
Px,y	The player has a race between x and y pips.
e>x	The equity (in millipoints) of the position is greater than x.
e<x	The equity (in millipoints) of the position is less than x.
ex,y	The equity (in millipoints) of the position is between x and y.
E>x	The error of the move played by player 1 (in millipoints) is greater than x.
E<x	The error of the move played by player 1 (in millipoints) is less than x.
Ex,y	The error of the move played by player 1 (in millipoints) is between x and y.
w>x	The player has winning chances greater than x%.
w<x	The player has winning chances less than x%.
wx,y	The player has winning chances between x% and y%.
g>x	The player has gammon chances greater than x%.
g<x	The player has gammon chances less than x%.
gx,y	The player has gammon chances between x% and y%.
b>x	The player has backgammon chances greater than x%.
b<x	The player has backgammon chances less than x%.
bx,y	The player has backgammon chances between x% and y%.
W>x	The opponent has winning chances greater than x%.
W<x	The opponent has winning chances less than x%.
Wx,y	The opponent has winning chances between x% and y%.
G>x	The opponent has gammon chances greater than x%.
G<x	The opponent has gammon chances less than x%.
Gx,y	The opponent has gammon chances between x% and y%.
B>x	The opponent has backgammon chances greater than x%.
B<x	The opponent has backgammon chances less than x%.
Bx,y	The opponent has backgammon chances between x% and y%.
o>x	The player has at least x checkers off.
o<x	The player has at most x checkers off.
ox,y	The player has between x and y checkers off.
O>x	The opponent has at least x checkers off.

continues on next page

Table 1 – continued from previous page

Query	Action
O<x	The opponent has at most x checkers off.
Ox,y	The opponent has between x and y checkers off.
k>x	The player has at least x backcheckers.
k<x	The player has at most x backcheckers.
kx,y	The player has between x and y backcheckers.
K>x	The opponent has at least x backcheckers.
K<x	The opponent has at most x backcheckers.
Kx,y	The opponent has between x and y backcheckers.
z>x	The player has at least x checkers in the zone.
z<x	The player has at most x checkers in the zone.
zx,y	The player has between x and y checkers in the zone.
Z>x	The opponent has at least x checkers in the zone.
Z<x	The opponent has at most x checkers in the zone.
Zx,y	The opponent has between x and y checkers in the zone.
bo>x	The player has at least x blots in the outfield.
bo<x	The player has at most x blots in the outfield.
box,y	The player has between x and y blots in the outfield.
BO>x	The opponent has at least x blots in the outfield.
BO<x	The opponent has at most x blots in the outfield.
BOx,y	The opponent has between x and y blots in the outfield.
bj>x	The player has at least x blots in the jan.
bj<x	The player has at most x blots in the jan.
bjx,y	The player has between x and y blots in the jan.
BJ>x	The opponent has at least x blots in the jan.
BJ<x	The opponent has at most x blots in the jan.
BJx,y	The opponent has between x and y blots in the jan.
t'word1;word2;...'	The position comments contain at least one of the words.
m'pattern1,pattern2,...'	The best checker moves containing at least one of the patterns.
m'ND,DT,DP,...'	The best cube decisions for No Double/Take, Double Take, Double Pass.
T>x	Date of position addition after x (YYYY/MM/DD).
T<x	Date of position addition before x (YYYY/MM/DD).
Tx,y	Date of position addition between x and y (YYYY/MM/DD).

Note

Filtering positions based on the dice roll (*D*) necessarily implies filtering positions based on the type of decision (*d*).

Note

For the relative difference filter in the race ($p>x$, $p<x$, px,y), the player is behind in the race compared to the opponent if $x>0$ and ahead if $x<0$. For example: $p<-10$: the player is at least 10 pips ahead in the race. $p50,70$: the player is between 50 and 70 pips behind in the race.

For example, the command `s s c p-20,-5 w>60 z>10 K2,3` filters all positions taking into account the checker structure, the score, and the cube of the edited position where the player has between 20 and 5 pips ahead in the race, with at least 60% winning chances, at least 10 checkers in the zone, and the opponent has between 2 and 3 backcheckers.

2.4.5 Various commands

Command	Action
clear, cl	Clear the command history.

Note

Database migrations are now performed automatically when opening a database.

2.5 Keyboard shortcuts

2.5.1 Database

Shortcut	Action
CTRL-N	Create a new database.
CTRL-O	Open an existing database.
CTRL-SHIFT-I	Import a database.
CTRL-SHIFT-S	Export the database.
CTRL-Q	Close blunderDB.
CTRL-M	Edit database metadata.

2.5.2 Position

Shortcut	Action
CTRL-I	Import one or more positions/matches from file (xg, sgf, mat, txt, bgf).
CTRL-SHIFT-F	Recursively import a folder of match/position files.
CTRL-C	Copy a position to the clipboard.
CTRL-V	Paste a position from the clipboard (automatic format detection).
CTRL-S	Save a position.
CTRL-U	Update a position.
Del	Delete the current position.
BACKSPACE	Reset board, cube, score, and dice.
CTRL-G	Show the position metadata.

2.5.3 Navigation

Shortcut	Action
CTRL-R	Reload all the positions from the database.
PageUp, h	First position / Previous game (MATCH mode).
LEFT, k	Previous position.
RIGHT, j	Next position.
UP, k	Previous move (when a move is selected in the analysis).
DOWN, j	Next move (when a move is selected in the analysis).
PageDown, l	Last position / Next game (MATCH mode).
CTRL-K	Go to a specific position.
r	Load a random position.

2.5.4 Display

Shortcut	Action
CTRL-LEFT	Board orientation to the left.
CTRL-RIGHT	Board orientation to the right.
p	Show/hide pipcount.

2.5.5 Modes

Shortcut	Action
TAB	Switch to EDIT mode / Exit MATCH mode to NORMAL.
CTRL-TAB	Toggle MATCH mode (enter/exit).
SPACE	Switch to COMMAND mode.

2.5.6 Tools

Shortcut	Action
CTRL-L	Show/hide the analysis.
CTRL-P	Show/hide the comments.
CTRL-B	Show/hide the filter library.
CTRL-F	Search for a position.
CTRL-H	Show/hide the search history.
CTRL-T	Show/hide the match panel.
CTRL-K	Show/hide the collections panel.
CTRL-Y	Show/hide the tournaments panel.
?	Show/hide the help.

2.5.7 Command Line

Shortcut	Action
UP	Browse command history up.
DOWN	Browse command history down.

2.5.8 Search History Panel

Shortcut	Action
Click	Select/deselect a search (show position).
Double-click	Execute search and close panel.
UP, k	Select previous search (younger, above).
DOWN, j	Select next search (older, below).
Esc	Deselect the search. If no search selected, close the panel.

2.5.9 Filter Library Panel

Shortcut	Action
Click	Select/deselect a filter (show position).
Double-click	Execute the filter search.
UP, k	Select previous filter (when a filter is selected).
DOWN, j	Select next filter (when a filter is selected).
Esc	Deselect filter. If no filter selected, close the panel.

2.5.10 Analysis Panel

Shortcut	Action
Click	Select/deselect a move (show/hide arrows).
UP, k	Select previous move (when a move is selected).
DOWN, j	Select next move (when a move is selected).
d	Toggle between checker and cube analysis (MATCH mode only).
Esc	Deselect move. If no move selected, close the panel.

2.5.11 Match Panel

Shortcut	Action
Click	Select a match.
Double-click	Enter MATCH mode.
UP, k	Select previous match.
DOWN, j	Select next match.
ENTER	Load the selected match.
Del	Delete the selected match.
Esc	Deselect/close the panel.

2.6 Frequently Asked Questions (FAQ)

2.6.1 What is the purpose of blunderDB?

blunderDB allows users to create a personalized database of positions. Its strength lies in not presupposing any classification *a priori*. This gives users the freedom to query positions with great flexibility by combining various criteria (race, structure, cube, score, backcheckers, checkers in the zone, chances of winning/gammon/backgammon, etc.).

Another convenient use of blunderDB is the creation of reference position catalogs. With the ability to tag positions, users can organize all their reference positions in a structured way using a single file. I hope that blunderDB facilitates the sharing of positions between players.

2.6.2 What motivated the creation of blunderDB?

I used to store interesting positions or blunders in different folders. However, I encountered difficulties in retrieving positions based on criteria that weren't initially considered in my choice of thematic categories. For example, if the positions were sorted by type of game (race, holding game, blitz, backgame, etc.), how could I retrieve all positions at a certain score or at a given cube level? Additionally, some old positions tended to be forgotten. I wanted a tool that aggregates all my positions without presupposing thematic categories *a priori*, allowing me to ask questions of the database. This type of software is quite common in chess, like ChessBase.

2.6.3 How to save the state of the current database?

The database is updated immediately upon validation of the request. No explicit saving operation is necessary.

2.6.4 Should I create different databases for different categories of positions?

Unless there are clearly identified reasons, it is essential not to split positions into separate databases, as this could prevent you from linking them in future searches. The philosophy of blunderDB is not to assume position categories *a priori* but to allow the user to query them flexibly. When positions have been encountered under specific conditions or for particular reasons, it may be wise to store them in separate databases. For example, you could create distinct position databases for:

- reference positions,
- blunders from live tournaments,
- blunders from online play.

2.6.5 How to merge multiple databases?

If you have multiple blunderDB databases that you wish to merge, use the “Import Database” feature:

1. Open the main database (the one that will receive the imported positions)
2. Click on the “Import Database” button in the toolbar
3. Select the database to import
4. blunderDB will automatically merge the positions

During the merge, blunderDB avoids duplicates and intelligently merges analyses and comments. Identical positions will not be duplicated, but their analyses and comments will be combined.

Note

It is recommended to make a backup copy of your main database before importing another database.

2.6.6 Which match file formats are supported?

blunderDB supports the following match formats:

- **eXtreme Gammon (XG)**: *.xg* files, with complete move analysis, cube decisions, played moves, and multi-engine support.
- **GNUbg**: *.sgf* files (Smart Game Format), with analysis.
- **Jellyfish**: *.mat* and *.txt* files.
- **BGBLitz**: *.bgf* files and text positions.

Import can be done via a single file, multiple selection, recursive folder, paste from clipboard, or drag and drop.

blunderDB automatically detects duplicates and prevents importing a match already present in the database.

2.6.7 What is a collection?

A collection is a custom grouping of positions. Unlike a filter search which is dynamic, a collection is a fixed set of positions manually chosen by the user. Collections allow, for example, grouping reference positions for a particular topic.

2.6.8 What is EPC?

EPC (Effective Pip Count) is a more accurate measure than the simple pip count for evaluating bearoff positions. The blunderDB EPC calculator uses the GNUbg 6-point bearoff database and computes in real time the EPC, average number of rolls, standard deviation, pip count, and wastage.

2.6.9 Does blunderDB have a command-line interface?

Yes, blunderDB has a command-line interface (CLI) that allows performing without a graphical interface operations such as creating databases, importing matches, exporting, searching positions, displaying statistics, etc. Refer to the CLI documentation for more details.

2.6.10 Can I modify, copy, share blunderDB?

Yes, absolutely. blunderDB is licensed under the MIT license.

2.6.11 What data format does blunderDB use?

The database is a simple SQLite file. In the absence of blunderDB, it can thus be opened with any SQLite file editor.

2.6.12 What were the design principles of blunderDB?

The modal operation of blunderDB (NORMAL, EDIT, COMMAND) is inspired by the very powerful text editor [Vim](#). I wanted blunderDB to be lightweight, standalone, installation-free, and available on multiple platforms, which led to my choice of the Go programming language and the Svelte library. For database serialization, the file format needed to be cross-platform and suitable for containing a database. The SQLite file format seemed like the perfect choice.

2.6.13 What is the software architecture of blunderDB?

- The backend is coded in [Go](#). It is responsible for all operations on the SQLite database that stores the positions.
- The frontend is coded in [Svelte](#). It is responsible for rendering the graphical interface and the Backgammon board.
- The application is encapsulated with [Wails](#), enabling the production of native desktop applications that can run on both Windows and Linux.
- The database is managed by [SQLite](#).

For more information, see the [blunderDB GitHub repository](#).

2.6.14 On which platforms does blunderDB run?

blunderDB runs on Windows, Linux and Mac.

2.6.15 Where does the blunderDB icon come from?

The blunderDB icon is the “goggling” emoticon from the [SMirC series](#).

2.7 Annex: Advanced Filter Usage

Warning

This section is for advanced users of blunderDB who wish to fully leverage the position search features.

Filters are at the heart of position analysis in blunderDB. Their use allows for searching specific positions with great precision. In this section, the use of filters through the COMMAND mode is detailed. The COMMAND mode can be accessed by pressing the SPACE key. It allows users to quickly combine filters and use the filter library with ease.

2.7.1 Command-line search for positions

To perform a search using filters,

1. Switch to EDIT mode using the TAB key.
2. Edit the current position.
3. Switch to COMMAND mode using the SPACE key.
4. Use the command `s` followed, optionally, by filters.
5. Start the search with the ENTER key.

Warning

Don't forget to clear the current position before starting a search (BACKSPACE key), if it isn't the one you want, to avoid excessively filtering checker structures.

Note

The list of available filters in COMMAND mode is provided in [Section 2.4.4](#).

2.7.2 Filter Library

The filter library allows the user to save search commands to facilitate their thematic studies.

To add a filter to the library,

1. Switch to EDIT mode.
2. Open the filter library by pressing CTRL-B.
3. Edit the current position.
4. Give a name to the filter.
5. Edit the search command.
6. Save the search command using the "Add" button.

Tip

While editing the command, you can use the UP and DOWN arrow keys to navigate through the command history.

To use a filter saved in the library,

1. Switch to NORMAL mode.

2. Open the filter library by pressing CTRL-B.
3. Search for the desired filter.
4. Double-click on the filter to start the search.

2.7.3 Examples

Here are some examples of using filters in COMMAND mode:

Type of position	Checker structure	Command
Races		s nc
Short races		s nc P<70
Hit to point 1		s m"6/1*"
Backgame 1-4	points 24, 21	s p>35
Take/Pass decision at 2-away 4-away	empty dice for top player, score 2a-4a	s s d
Too Good doubles	empty dice for bottom player	s d e>1000
Blitz with at least 20% gammon	point in jan, checkers on the bar	s g>20
Player 1 errors greater than 40 millipoints		s E>40
Positions from Aachen2024 tournament		s t"Aachen2024"
One backchecker to bring back		s k1,1
Leave the 20-point anchor	point 20	s m"20/"
Prime vs prime	Set the primes	s
Ace-point bear-off	point 1 for the opponent	s P<60
Double with at least 20pip ahead	empty dice for bottom player	s d p<-20

2.8 Windows Annex: False Detection of blunderDB as Malware

Note

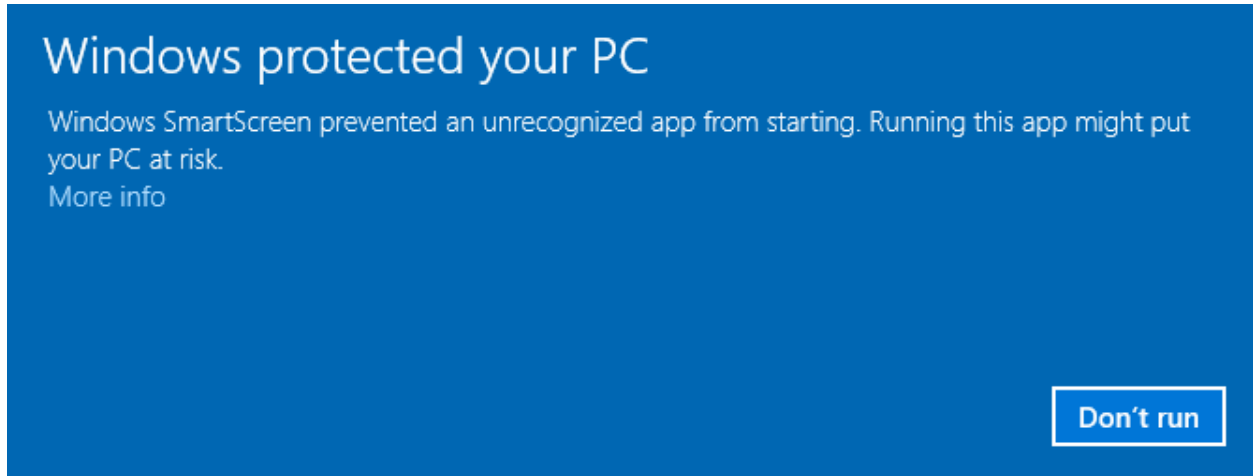
The following applies to Windows 10 and 11 operating systems.

Windows now requires software publishing companies or independent software developers to digitally certify their applications, or even distribute them via the Windows Store. It is therefore recommended to turn to external companies to obtain a digital certificate, which costs several hundred euros (see, for example, https://learn.microsoft.com/en-us/archive/blogs/ie_fr/certificats-de-signature-de-code-ev-extended-validation-et-microsoft-smartscreen).

Since I am offering blunderDB for free, I do not wish to pursue these costly options. Consequently, it is very likely that Windows will warn you of a potential threat or even block the execution of blunderDB entirely. The following sections explain the steps to bypass Windows' warnings.

2.8.1 Windows SmartScreen Warning

After downloading blunderDB, when you run it, Windows may display a warning such as:

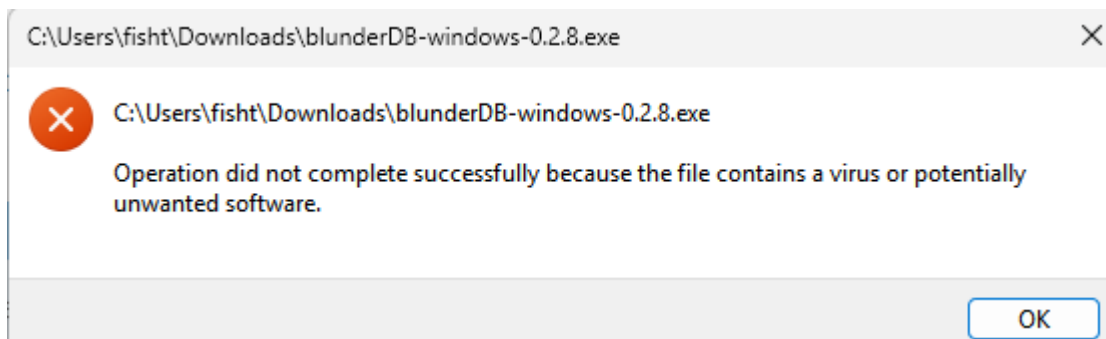


If you want to allow a specific executable blocked by SmartScreen:

1. **Try running the executable:**
 - When you attempt to launch the executable, SmartScreen may block it and display a warning.
2. **Click on “More Info”:**
 - In the SmartScreen warning window, click on **More Info**.
3. **Select “Run anyway”:**
 - If you trust the executable, click **Run anyway** to bypass the SmartScreen warning for this instance.

2.8.2 Windows Defender Blocking

For certain security settings in Windows, even after bypassing SmartScreen (see the previous section), Windows Defender may prevent the execution of blunderDB with messages such as:

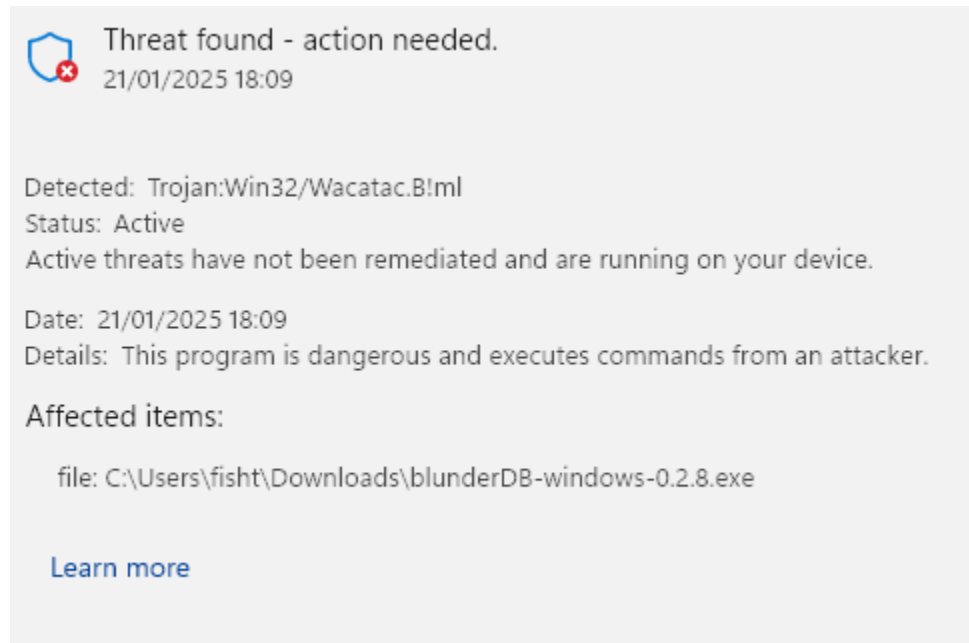


or even:

or even place it in quarantine.

Windows Defender is known to trigger false positives. This issue is explicitly mentioned in the FAQ on the official Golang website (<https://go.dev/doc/faq#virus>) or in GitHub tickets for some projects programmed in Go (<https://github.com/golang/vscode-go/issues/3182>).

If you want to prevent Windows Security from scanning blunderDB:



1. **Open Windows Security:**

- Go to **Start** and type **Windows Security**.

2. **Go to “Virus & Threat Protection”:**

- Click on **Virus & Threat Protection**.

3. **Manage Settings:**

- Scroll down and click on **Manage settings** under Virus & Threat Protection settings.

4. **Add or remove exclusions:**

- Scroll down to the **Exclusions** section and click on **Add or remove exclusions**.

5. **Add an exclusion:**

- Click on **Add an exclusion** and select **File**. Then, navigate to the executable you want to exclude and select it.

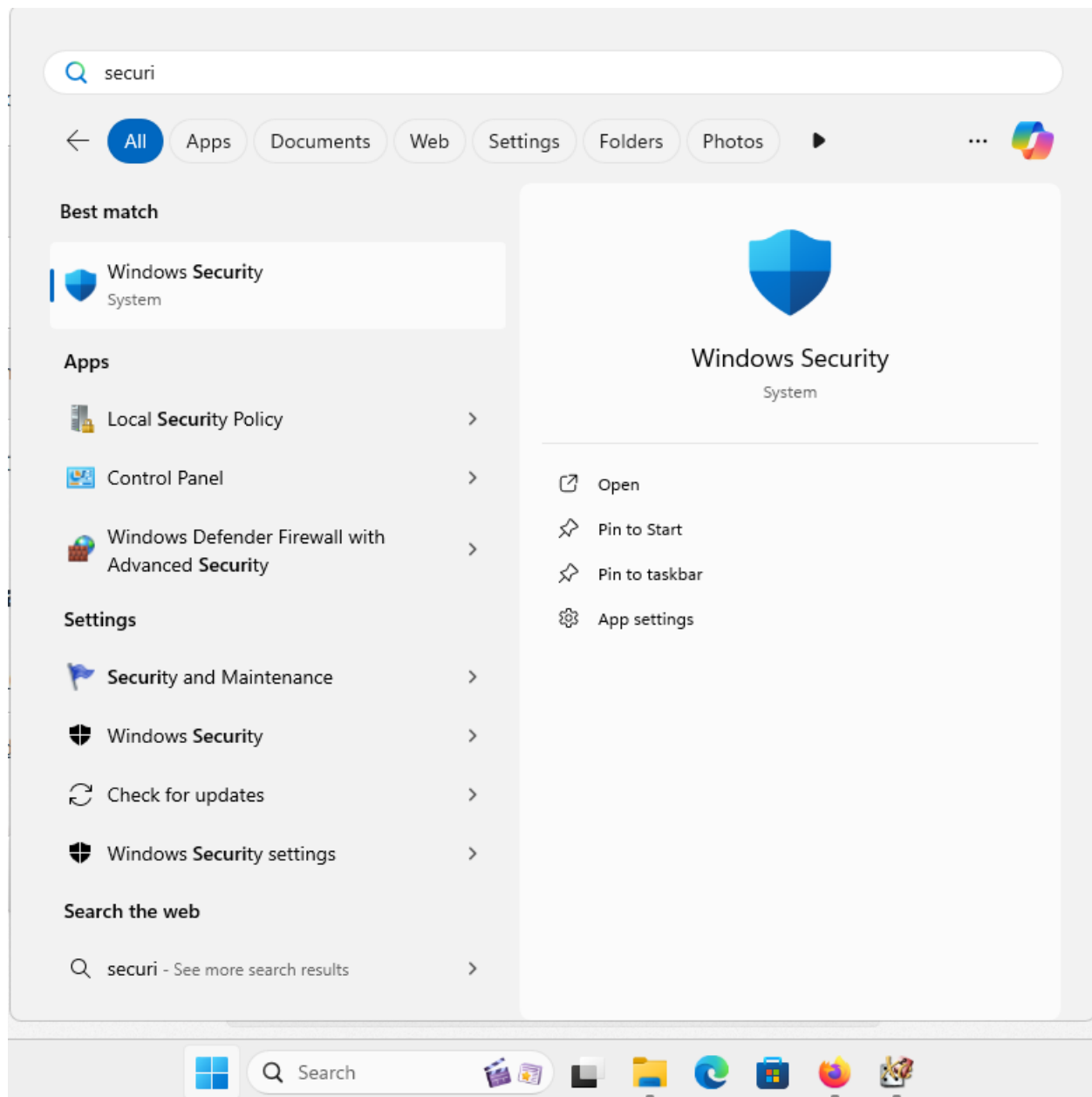
2.9 Mac Annex: Possible Blocking of blunderDB

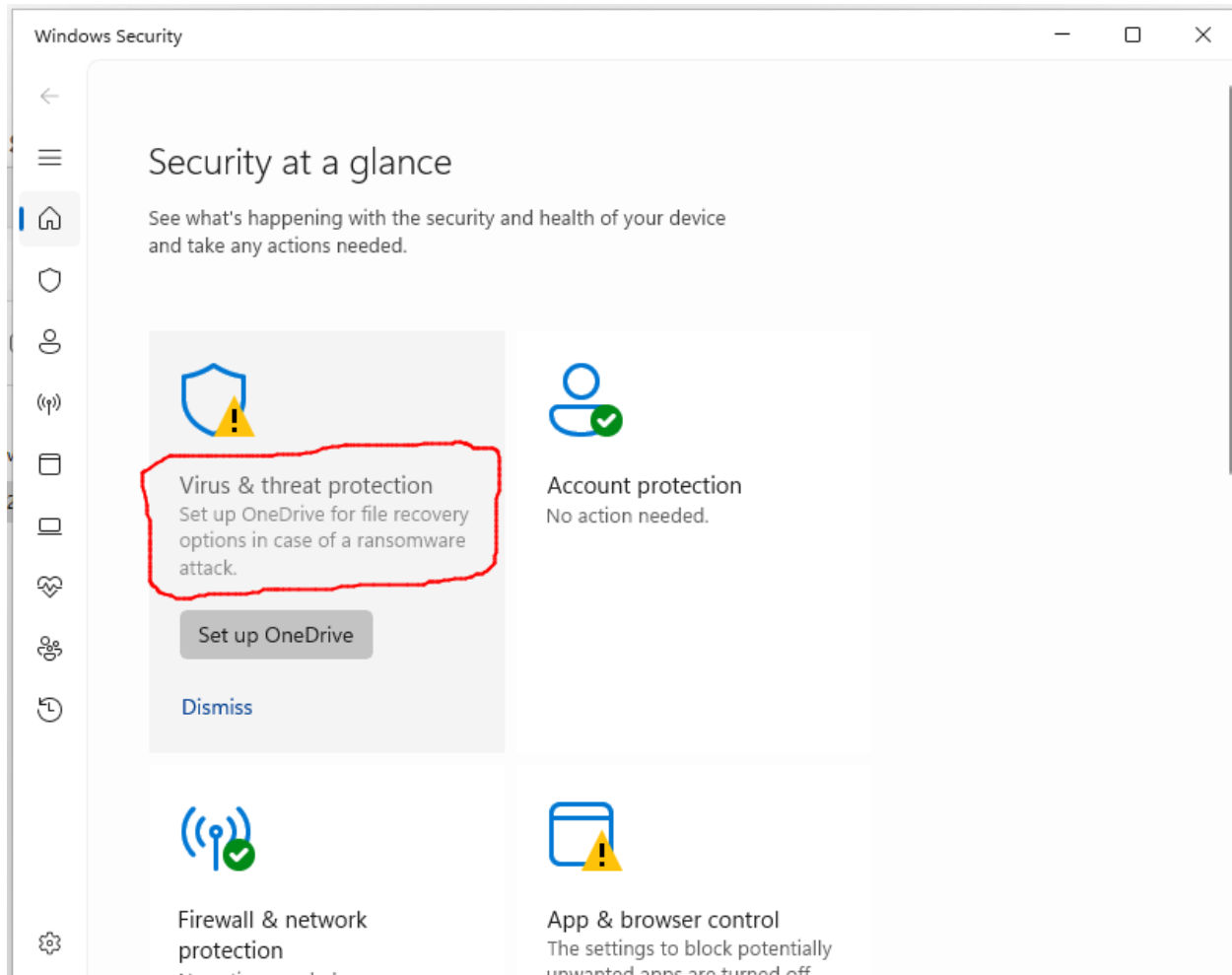
Note

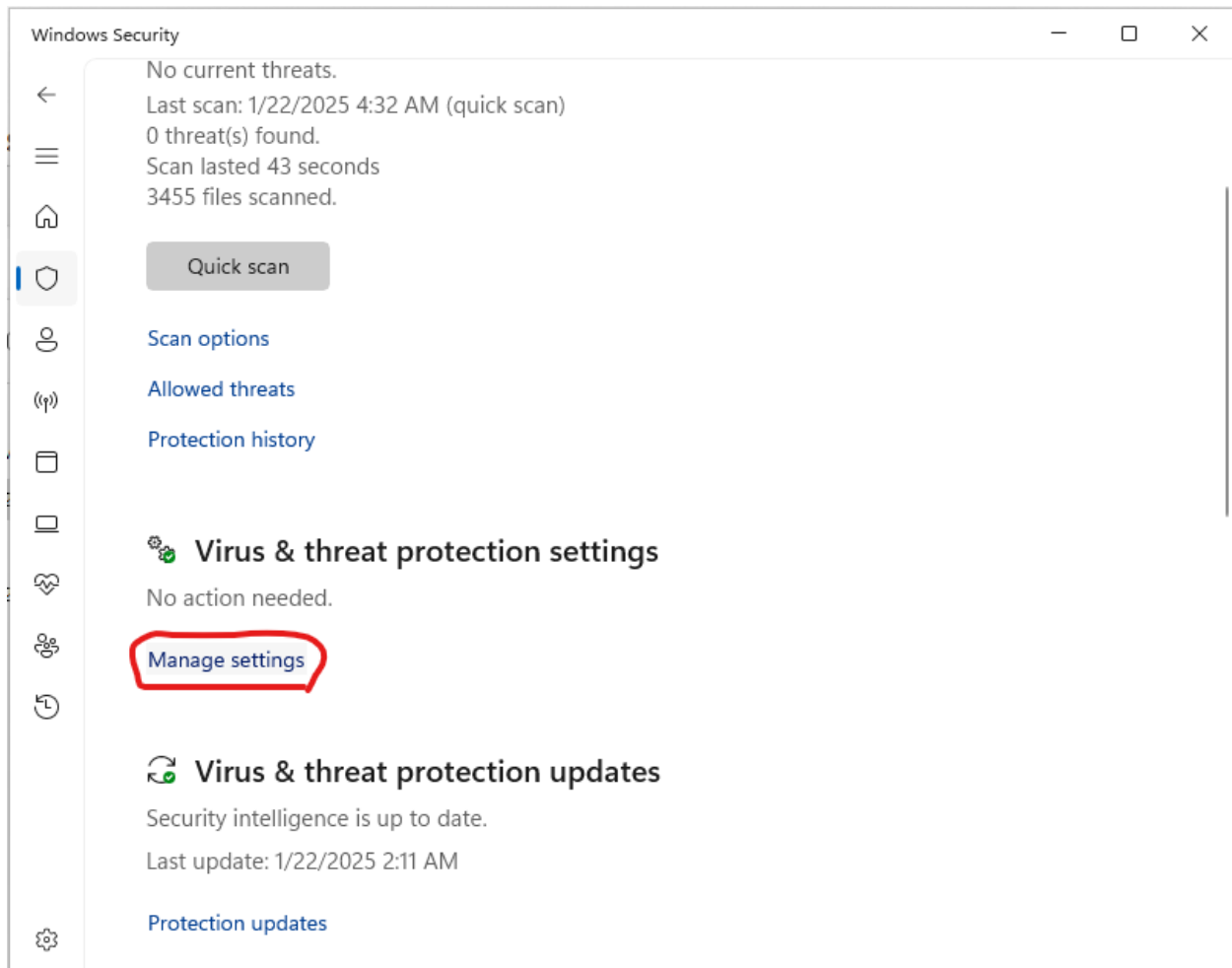
The following concerns the macOS operating system.

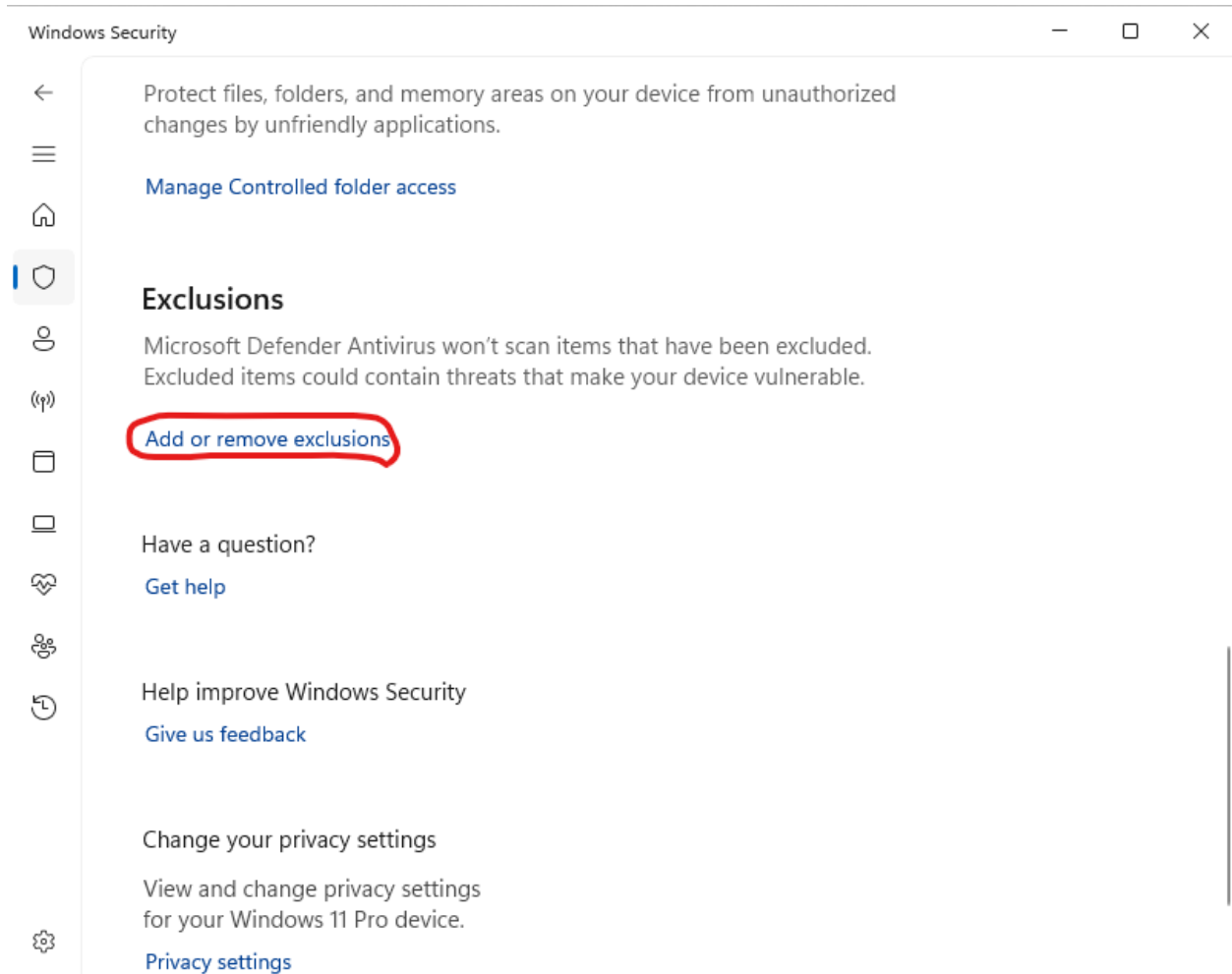
Mac requires software publishing companies or independent software developers to digitally certify their applications. The developer must enroll in the Apple Developer Program by paying an annual membership fee (<https://developer.apple.com/support/compare-memberships/>).

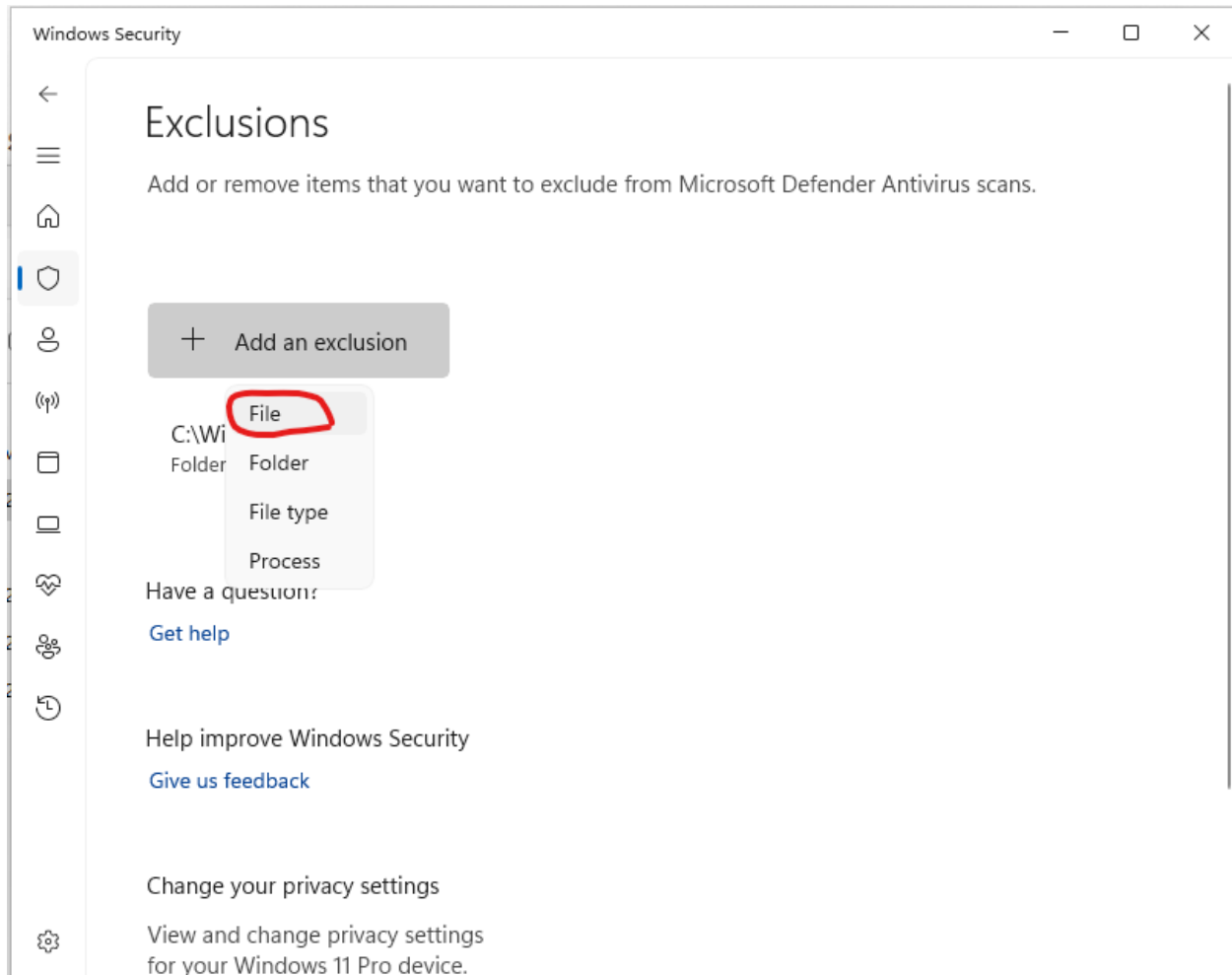
Since I am sharing blunderDB for free, I do not wish to pursue these costly options. As a result, Mac will likely warn you of a potential risk or even block the execution of blunderDB entirely. The following sections explain the steps to bypass Mac’s restrictions.

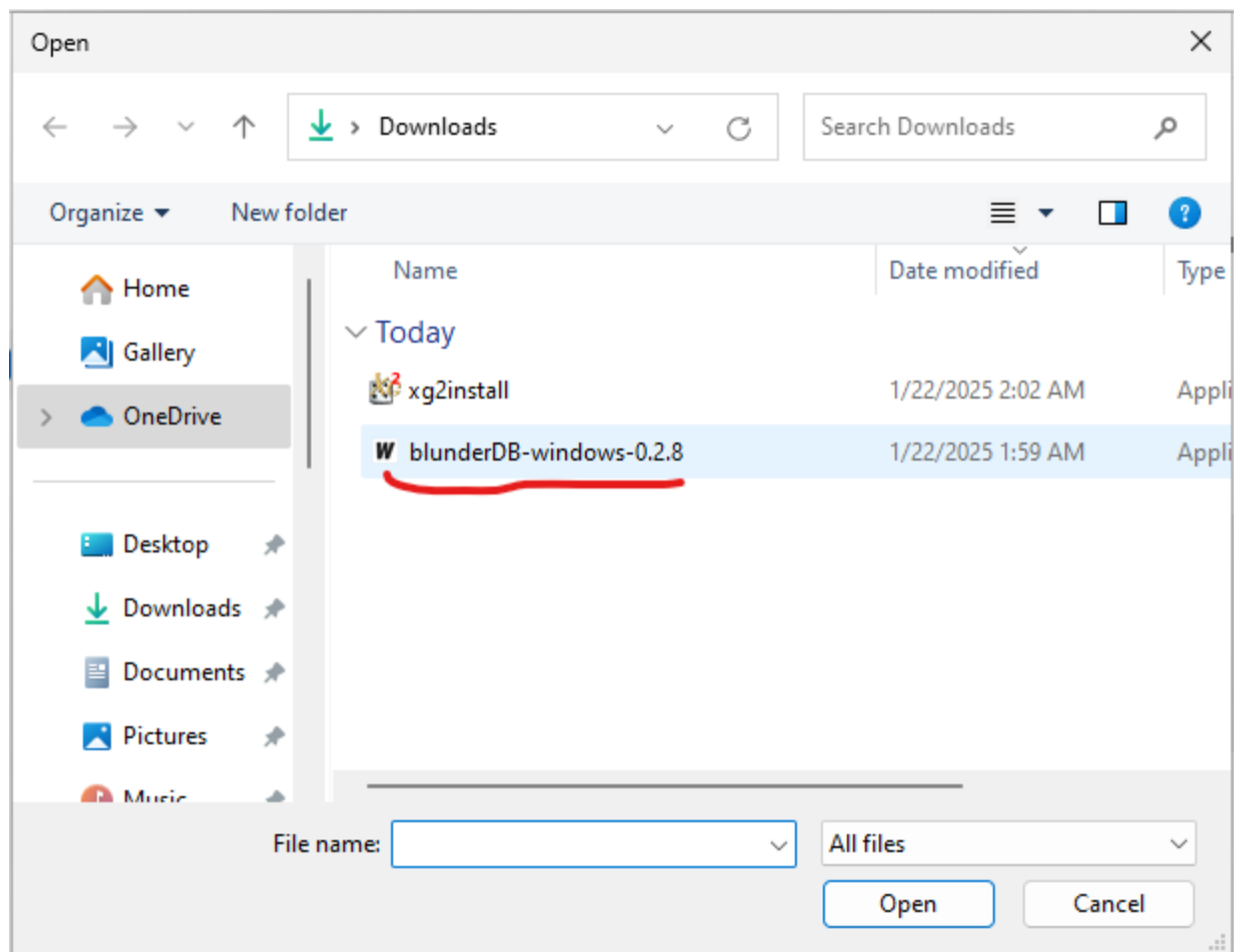


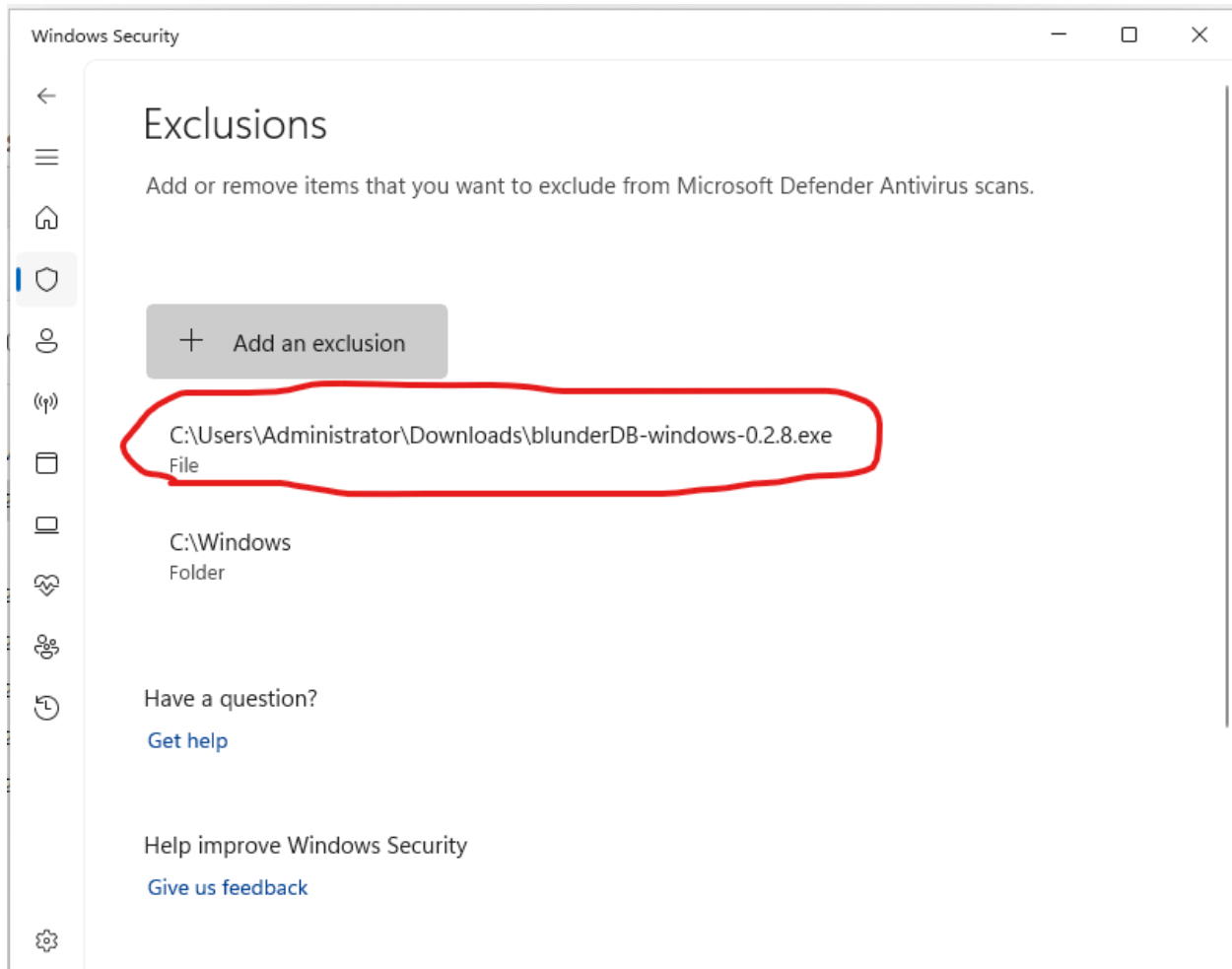












2.9.1 Installation of blunderDB

After downloading blunderDB, drag the downloaded file into the Applications section of your Finder. If you have already tried to run blunderDB and Mac warns you of a potential risk, follow the steps below.

2.9.2 Authorization to Run blunderDB

1. Open Finder and go to the Applications section.
2. Find blunderDB and right-click on it.
3. Select Open.
4. A warning window will appear. Click Open.
5. blunderDB will open, and you can start using it.

Note

You only need to perform this operation once. Afterward, you can open blunderDB without having to go through these steps.

2.10 Annex: Database Schema

Important

Always back up your .db file before performing database migrations.

2.10.1 Version 1.0.0

Version 1.0.0 of the database contains the following tables:

- **position:** Stores the positions with the columns *id* (primary key) and *state* (state of the position in JSON format).
- **analysis:** Stores the analyses of the positions with the columns *id* (primary key), *position_id* (foreign key referencing *position*), and *data* (analysis data in JSON format).
- **comment:** Stores the comments associated with the positions with the columns *id* (primary key), *position_id* (foreign key referencing *position*), and *text* (comment text).
- **metadata:** Stores the metadata of the database with the columns *key* (primary key) and *value* (value associated with the key).

2.10.2 Version 1.1.0

Version 1.1.0 of the database adds the following table:

- **command_history:** Stores the command history with the columns *id* (primary key), *command* (text of the command), and *timestamp* (date and time of command execution).

The other tables remain unchanged from version 1.0.0.

To migrate the database from version 1.0.0 to version 1.1.0, execute the command `migrate_from_1_0_to_1_1` in blunderDB.

2.10.3 Version 1.2.0

Version 1.2.0 of the database adds the following table:

- **filter_library**: Stores search filters with the columns *id* (primary key), *name* (filter name), *command* (command associated with the filter), and *edit_position* (position edited when saving the filter).

The other tables remain unchanged from version 1.1.0.

To migrate the database from version 1.1.0 to version 1.2.0, execute the command `migrate_from_1_1_to_1_2` in blunderDB.

2.10.4 Version 1.3.0

Version 1.3.0 of the database adds the following table:

- **search_history**: Stores position search history with the columns *id* (primary key), *command* (search command text), *position* (position state at the time of search), and *timestamp* (date and time of the search).

The other tables remain unchanged from version 1.2.0.

To migrate the database from version 1.2.0 to version 1.3.0, execute the command `migrate_from_1_2_to_1_3` in blunderDB.

2.10.5 Version 1.4.0

Version 1.4.0 of the database adds the following tables for match management:

- **match**: Stores imported matches with the columns *id* (primary key), *player1_name*, *player2_name*, *event*, *location*, *round*, *match_length*, *match_date*, *import_date*, *file_path*, *game_count*, and *match_hash* (hash for duplicate detection).
- **game**: Stores the games of a match with the columns *id* (primary key), *match_id* (foreign key referencing *match*), *game_number*, *initial_score_1*, *initial_score_2*, *winner*, *points_won*, and *move_count*.
- **move**: Stores the moves of a game with the columns *id* (primary key), *game_id* (foreign key referencing *game*), *move_number*, *move_type*, *position_id* (foreign key referencing *position*), *player*, *dice_1*, *dice_2*, *checker_move*, and *cube_action*.
- **move_analysis**: Stores the analysis of each move with the columns *id* (primary key), *move_id* (foreign key referencing *move*), *analysis_type*, *depth*, *equity*, *equity_error*, *win_rate*, *gammon_rate*, *backgammon_rate*, *opponent_win_rate*, *opponent_gammon_rate*, and *opponent_backgammon_rate*.

Migration from 1.3.0 to 1.4.0 is automatic when opening the database.

2.10.6 Version 1.5.0

Version 1.5.0 of the database adds the following tables for collection management:

- **collection**: Stores position collections with the columns *id* (primary key), *name*, *description*, *sort_order*, *created_at*, and *updated_at*.
- **collection_position**: Junction table that associates positions with collections with the columns *id* (primary key), *collection_id* (foreign key referencing *collection*), *position_id* (foreign key referencing *position*), *sort_order*, and *added_at*. The pair (*collection_id*, *position_id*) is unique.

Migration from 1.4.0 to 1.5.0 is automatic when opening the database.

2.10.7 Version 1.6.0

Version 1.6.0 of the database adds the following table for tournament management:

- **tournament:** Stores tournaments with the columns *id* (primary key), *name*, *date*, *location*, *sort_order*, *created_at*, and *updated_at*.
- Addition of the *tournament_id* column (foreign key referencing *tournament*) in the *match* table to assign a match to a tournament.

Migration from 1.5.0 to 1.6.0 is automatic when opening the database.

2.10.8 Version 1.7.0

Version 1.7.0 of the database adds the following column:

- Addition of the *last_visited_position* column in the *match* table to remember the last visited position in each match.

Migration from 1.6.0 to 1.7.0 is automatic when opening the database.

Note

Since version 0.10.0, all database migrations are performed automatically when opening a database file.

<https://youtu.be/Ln7XKVFqfUk>

<https://youtu.be/HkY4iXjxMeI>

CONTACT

Author: Kévin Unger <blunderdb@proton.me>. You can also find me on Heroes under the username postmanpat.

I initially developed blunderDB for my personal use to help detect patterns in my mistakes. However, it's very rewarding to receive feedback, especially after spending a lot of time on design, coding, and debugging. So feel free to reach out to share your experiences. All (constructive) feedback is welcome.

Here are several ways to discuss:

- Join the Discord server of blunderDB: <https://discord.gg/DA5PpzM9En>
- Email me at blunderdb@proton.me.
- Discuss with me if we meet in a tournament.
- On GitHub.
 - Open an issue: <https://github.com/keving/blunderDB/issues>
 - For bug fixes or improvement suggestions, create a pull request.

DONATE

If you appreciate blunderDB and want to support its past and future developments, you can

- buy me a drink if we have the pleasure of meeting!
- make a small donation via PayPal to the address blunderdb@proton.me

ACKNOWLEDGMENTS

I dedicate this little software to my wife Anne-Claire and our dear daughter Perrine. I would especially like to thank a few friends:

- *Tristan Remille*, for introducing me to backgammon with joy and kindness; for showing the way in understanding this wonderful game; and for continuing to support me despite my poor attempts to improve my play.
- *Nicolas Harmand*, a cheerful companion for over a decade in great adventures, and a fantastic sparring partner since he has caught the backgammon bug.