AmyBoard

COLLABORATORS			
	TITLE : AmyBoard		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		June 24, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AmyBoard

1.1 AmyBoard.guide

Introduction
****

This manual documents how to run and install AmyBoard and how to report bugs.

Index of concepts and symbol names.

1.2 AmyBoard.guide/Copying

GNU GENERAL PUBLIC LICENSE
****

Version 2, June 1991

Preamble
========

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program’s name and an idea of what it does.
Copyright (C) 19yy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type ‘show w’.  This is free software, and you are welcome
```

to redistribute it under certain conditions; type ‘show c’
for details.

The hypothetical commands show w and show c should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than show w and show c; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright
interest in the program ‘Gnomovision’
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

1.3 AmyBoard.guide/Installation

How to configure, compile and install AmyBoard.
****

AmyBoard requires OS 2.04, 1 MB of RAM and MUI 2.0 or later. To play against the Amiga you need GNU Chess installed. (Aminet, game/think directory) To play on the ICS you need a telnet program (AmiTCP/IP, AS225 or whatever else) reading from stdin and writing to stdout or a modem and a terminal program using stdin/stdout as well.

After unpacking the binary archive to RAM:, you should run the installation script by doubleclicking the icon Install. This icon assumes that you have the Commodore Installer in your workbench path. If you still don’t have the Installer, you can get it from Aminet, util/misc directory.

Installation will be done in one directory, say WORK:, a subdirectory AmyBoard will be created. The binary and some icons will be in WORK:AmyBoard, config files in WORK:AmyBoard/lib. (1) If you have a recent version of AmyBoard in the same directory, the old config files and icons will be copied to WORK:AmyBoard/obsolete, so that you can reuse them.

Once the installation script is done, you should start AmyBoard with the Work:AmyBoard/AmyBoard icon and the MUI prefs. Change the colors and size of the board, the location of the window until you like it, then save these settings with the MUI prefs.

The installation script will create a preferences file AmyBoard.prefs for you. This file is assumed preliminary (though perfectly usable), it is a good idea to edit and modify it using your favourite text editor. Things you might wish to change are:

* The colors; this is done by changing the pen numbers whitePiecePen, blackPiecePen, lightSquarePen and darkSquarePen. Registered MUI users should keep the values -1 which indicate to use the default MUI colors Shinepen, Shadowpen, Halfshinepen and Halfshadowpen, respectively, and use the MUI prefs to change the colors.

* Path and memory requirements of the GNU Chess binary: More memory improves the strength of GNU Chess. (The installation script already did this for you.) The memory requirements are changed with the arguments -C and -T:

  -C 1001 -T 4001
  about 1 MByte

  -C 3001 -T 20001
  about 2 MByte

  -C 4001 -T 30001
  about 2.5 MByte

  -C 6001 -T 40001
  To tell AmyBoard that the GNU Chess binary is in Fun:gnuchess-4.0.pl74 and you want it to allocate about 1 MByte RAM you would enter

  ```
  firstChessProgram=Fun:gnuchess-4.0.pl74/gnuchessx -C 1001 -T 4001
  secondChessProgram=Fun:gnuchess-4.0.pl74/gnuchessx -C 1001 -T 4001
  ```

  It is important to use the complete path here, as the AmigaDOS function LoadSeg ignores your path settings. See

  GNU Chess options
  .

* The default time settings of GNU Chess: Variables movesPerSesssion and timeControl. See
  GNU Chess options
  .

* Your favourite ICS host (icsHost) and port number (icsPort). (You probably won't need to change the defaults, as the installation script can create ICS icons for different hosts, which override these settings.) See
  ICS options
  .

* The name of your telnet or terminal program. (Note, that this must use stdin and stdout!) The installation script lets you select one. (Again, note that you have to use the complete path.) See

  ICS options
  . See

Serial ICS connection
.

For the full set of AmyBoard's options see See
Invocation
.

Be sure, that the programs stack is no less than 10000 bytes. This is especially important, when running the program from Toolmanager, as the default value of Toolmanagers Exec Objects are 4096 bytes only.

If you would like to recompile AmyBoard, you currently need either Dice or gcc. Edit the Makefile, remove the comments from your compilers switches and comment the others in. DMake or Make, respectively, will do the compilation for you.

It shouldn't be a serious problem to use SAS/C, as I tried to keep the code compatible. However, I cannot test it, as I don't have this compiler. (Gifts welcome. :-) By using a Unix compatible front end like dev/c/cc.lha from Aminet, you should even be able to use the same Makefile.

---------- Footnotes ----------

(1) This has the advantage that deinstalling AmyBoard is rather easy: Just delete the AmyBoard drawer. However, you won't do this, won't you? ;-)

1.4 AmyBoard.guide/Invocation

Command options supported by AmyBoard.
****

On Unix it is usual behaviour to implement command options with names like -v (verbose) or -h help. If a program like AmyBoard has lots of possible options, it makes sense to drop these short option names and instead use option names like -verbose or -help. This is almost like on the Amiga where one would use verbose or help.

To be both compatible to XBoard and have a real Amiga program I decided to allow both: With AmyBoard you may use either -debug or debug, whatever you want. Thus in the following words beginning with a hyphen (-) are option names, but you may omit the hyphen.

AmyBoard always runs in one of 4 possible modes:

amyboard [options]
    starts AmyBoard as a GNU chess interface. As an interface to GNU Chess, AmyBoard lets you play a game against the machine, set up arbitrary positions, force variations, or watch a game between two machines.

amyboard -ics [options]
    lets AmyBoard act as an interface to the Internet Chess Server

(ICS). You can play against other ICS users, observe games they are playing, or review games that have recently finished. See ICS .

amyboard -ncp [options]

allows to use AmyBoard as a simple chessboard to play through games. It will read and write game files and allow you to play through variations manually. You can use it to browse games off the net or review GNU Chess, ICS, and correspondence games you have saved. These features are available at all times; even if you do not have access to GNU Chess or the ICS, you can use them.

cmail

will use AmyBoard as an interface to electronic mail correspondence chess. (This possibility is currently not given on the Amiga: cmail is a Perl script which assumes a full mail setup, both usually not given on the Amiga. Probably someone would be interested in converting cmail into an ARexx script?

Most of the AmyBoard options have both a long name and a short name. You have three possibilities to set the options:

1. By editing the preferences file PROGDIR:lib/AmyBoard.prefs, where PROGDIR: indicates the directory of the AmyBoard binary. The lines of the file are interpreted as if they were toolkit options, empty lines or lines beginning with a semicolon are treated as comments.

2. By editing an icons toolkit options.

3. By supplying arguments with the CLI command line.

Of course toolkit options or CLI arguments override settings of the prefs file.

The argument parsing behaves in general much like ReadArgs(), except for boolean arguments which are a mixture of the /S and /T switches: In general, they behave like /S arguments in the CLI, but like /T in the toolkit options or in the prefs file. To be precise, you may use something like

```
amyboard ncp
```

in the CLI (indicating that the value of ncp should be toggled), but not as a toolkit option or as a prefs file entry: In that case you have to explicitly state something like

```
NCP true
```

(For obvious reasons this version conflicts with the above.) However, you may always write

```
NCP=false
```

As usual, option names are case-insensitive.

GNU Chess options
Controlling GNU Chess.

```
                 ICS options
                                  Connecting to and using ICS.

                 I-O options
                                  Load and Save options.

                 User interface options
                       Look and Feel.

                 Other options
                                Miscellaneous
```

1.5 AmyBoard.guide/GNU Chess options

```
Controlling GNU Chess
=====================

-tc or -timeControl minutes[:seconds]
-mps or -movesPerSession moves
     Each player begins with the specified amount of time on his clock.
     If a player makes the specified number of moves before his clock
     runs out, AmyBoard adds the specified amount of time to his clock.
     Default: 40 moves in 5 minutes.

-st or -searchTime minutes[:seconds]
     Tells GNU Chess to spend at most the given amount of time
     searching for each of its moves. Without this option, GNU Chess
     chooses its search time based on the number of moves and amount of
     time remaining until the next time control.

-sd or -searchDepth number
     Tells GNU Chess to look ahead at most the given number of moves
     when searching for a move to make. Without this option, GNU Chess
     chooses its search depth based on the number of moves and amount
     of time remaining until the next time control.

-thinking or -showThinking
     If this option is set, GNU Chess's notion of the score and best
     line of play from the current position is displayed as it is
     thinking. The score indicates how many pawns ahead (or if nega-
     tive, behind) GNU Chess thinks it is. In matches between two
     machines, the score is prefixed by W or B to indicate whether it
     is showing White's thinking or Black's.

-mm or -matchMode
     Automatically runs a game between two chess programs.  If the
     loadGameFile or loadPositionFile option is set, AmyBoard starts
     the game with the given opening moves or the given position;
     otherwise, the game starts with the standard initial chess
     position.  If the saveGameFile option is set, a move record for the
     match is appended to the specified file. If the savePositionFile
     option is set, the final position reached in the match is appended
     to the specified file. When the match is over, AmyBoard exits.
```

Default: false.

-fcp or -firstChessProgram program

Name of first chess program. In matches between two machines, this program plays Black. Default: gnuchessx.

-scp or -secondChessProgram program

Name of second chess program, if needed. In matches between two machines, this program plays White; otherwise it is not started. Default: gnuchessx.

-initString string

The string that is sent to initialize the chess program. Default:

```
new
beep
random
easy
```

Setting this option from the command line is tricky, because you must type in real newline characters. In most shells you can do this by entering a \ character followed by a newline. It is easier to set the option from your .Xdefaults file; in that case you can include the character sequence \n in the string, and it will be converted to a newline.

If you change this option, don’t remove the new and beep commands. You can remove the random command if you like; including it causes GNU Chess to randomize its move selection slightly so that it doesn’t play the same moves in every game. (Even without random, GNU Chess randomizes its choice of moves from its opening book.) You can also remove easy if you like; including it toggles easy mode off, causing GNU Chess to think on your time. That is, if easy is included in the initString, GNU Chess thinks on your time; if not, it does not. (Yes, this does seem backwards, doesn’t it!) You can also try adding other commands to the initString; see the GNU Chess documentation for details.

-whiteString string
-blackString string

These options control what is sent when the Machine White and Machine Black buttons are selected. This is mostly for compatibility with obsolete versions of GNU Chess.

-childPriority
-childStack

set the respective settings of the process to be created for GNU Chess. Note, that the GNU Chess task requires a relatively big stack, compared to true Amiga programs: Do not use less than 50000 bytes. By increasing GNU Chess’s memory requirements you probably increase it’s stack requirements too: Thus you might need to change the default of 100000 bytes.(1)

---------- Footnotes ----------

(1) It would have been possible to compile GNU Chess with dynamic stack checking. However, this would give overhead with every function call and result in a drastical loss of playing strength, so I decided

not to use this feature.

1.6 AmyBoard.guide/ICS options

Connecting to and using ICS

-ics or -internetChessServerMode

Connect with an Internet Chess Server to play chess against its other users, observe games they are playing, or review games that have recently finished. Default: false. To use AmyBoard in ICS mode, run it in the foreground, and use the terminal you started it from to type commands and receive text responses from the chess server. See ICS . See File Menu . See Step Menu . See Mode Menu .

-icslogon or -internetChessServerLogonScript file-name

Whenever AmyBoard connects to the Internet Chess Server, if it finds a file with the name given in this option, it feeds the file's contents to the ICS as commands. The default file name is PROGDIR:lib/.icsrc. Usually the first two lines of the file should be your ICS user name and password. The file name can be either absolute or relative to PROGDIR: or ENV:. Due to a problem which I did not figure out yet, it may be needed to add a empty line at the top of .icsrc.

-autocomm or -autoComment

If autoComment is True, any remarks made on ICS while you are observing or playing a game are recorded as a comment on the current move. This includes remarks made with the ICS commands say, tell, whisper, and kibitz. Limitation: remarks that you type yourself are not recognized; AmyBoard scans only the output from ICS, not the input you type to it. Default: False.

-autoflag or -autoCallFlag

If autoCallFlag is true and your opponent runs out of time before you do, AmyBoard will automatically call his flag, claiming a win on time (or a draw if you do not have mating material). Default: false.

-autobs or -autoObserve

If autoObserve is true and you add a player to your gnotify list on ICS, AmyBoard will automatically observe all of that player's games, unless you are doing something else (such as observing or playing a game of your own) when one starts. Default: false.

-quiet or -quietPlay

If this option is true, AmyBoard will automatically issue a

```
set shout 0
```

command whenever you start an ICS game and a

```
set shout 1
```

command whenever you finish one. Default: false.

-icshost or -internetChessServerHost host

The Internet host name or address of the chess server to connect to when in ICS mode. Default: chess.lm.com. See the file ics-addresses in the AmyBoard distribution for a list of other addresses to try. See also the output of the command

```
finger chess@ics.onenet.net
```

If your site doesn't have a working Internet name server, try specifying the host address in numeric form. The address of chess.lm.com is 192.231.221.16; that of ics.onenet.net is 164.58.253.10.

-icsport or -internetChessServerPort port-number

The port number to use when connecting to a chess server in ICS mode. Default: 5000.

-telnetProgram prog-name

This option gives the name of the telnet program to be used with the gateway and useTelnet options. The default is telnet. The telnet program is invoked with the value of internetChessServerHost as its first argument and the value of internetChessServerPort as its second argument.

Of course you are not forced to use a certain telnet program, in fact the only requirement is, that the program reads input from stdin and writes to stdout. For example, you might wish to use a terminal program, if you don't have a SLIP or PPP connection. See

Serial ICS connection

.

-icsWindow

is a specification of a window to open for ICS input/output. This is especially useful, if you start AmyBoard from Workbench: The CLI window is used otherwise. Note that setting this option forces a separate window, even if you start the program from the CLI. However, AmyBoard does not detach from the CLI (except when using Run >nil: <nil:), thus you better set this option in the icon toolkits only.

1.7 AmyBoard.guide/I-O options

Load and Save options
=====================

-lgf or -loadGameFile file

```
-lgi or -loadGameIndex index
     If the loadGameFile option is set, AmyBoard loads the specified
     game file at startup. The file name - specifies the standard
     input. If there is more than one game in the file, AmyBoard pops
     up a menu of the available games, with entries based on their PGN
     tags.  If the loadGameIndex option is set to N, the menu is
     suppressed and the N th game found in the file is loaded
     immediately.  The menu is also suppressed if matchMode is enabled
     or if the game file is a pipe; in these cases the first game in
     the file is loaded immediately.

-td or -timeDelay seconds
     Time delay between moves during Load Game. Fractional seconds are
     allowed; try -td 0.4. A time delay value of -1 tells AmyBoard not
     to step through game files automatically. Default: 1 second.

-sgf or -saveGameFile file
     If this option is set, AmyBoard appends a record of every game
     played to the specified file. The file name - specifies the
     standard output.

-autosave or -autoSaveGames
     If this option is true, at the end of every game AmyBoard prompts
     you for a file name and appends a record of the game to the file
     you specify. Ignored if saveGameFile is set.

-lpf or -loadPositionFile file
-lpi or -loadPositionIndex index
     If the loadPositionFile option is set, AmyBoard loads the
     specified position file at startup. The file name - specifies the
     standard input. If the loadPositionIndex option is set to N, the
     Nth position found in the file is loaded; otherwise the first
     position is loaded.

-spf or -savePositionFile file
     If this option is set, AmyBoard appends the final position reached
     in every game played to the specified file. The file name -
     specifies the standard output.

-oldsave or -oldSaveStyle
     If this option is false (the default), AmyBoard saves games in PGN
     (portable game notation) and positions in FEN (Forsythe-Edwards
     notation).  If the option is true, a save style that is compatible
     with older versions of AmyBoard is used instead.
```

1.8 AmyBoard.guide/User interface options

```
                Look and Feel
=============

-bell or -ringBellAfterMoves
     If this option is true, AmyBoard alerts you by ringing the
     terminal bell after each of your opponent’s moves (or after every
     move if you are observing a game on the Internet Chess Server).
```

The bell is not rung after moves you make or moves read from a saved game file. Default: false.

If you turn on this option when using AmyBoard with the Internet Chess Server, you will probably want to give the

```
set bell 0
```

command to the ICS, since otherwise the ICS will ring the bell itself after every move (not just yours). (The .icsrc file is a good place for this, see

ICS options

)

-queen or -alwaysPromoteToQueen

If this option is false (the default), AmyBoard brings up a dialog box whenever you move a pawn to the last rank, asking what piece you want to promote it to. If the option is true, your pawns are always promoted to queens. (Your opponent can still underpromote, however.)

-size or -boardSize (Huge | Large | Medium | Small | Tiny)

Determines how large the board will be and what built-in piece bitmaps will be used. On a huge board (the default), pieces are 80x80 pixels, on a large board 64x64 pixels, and on a medium board 40x40 pixels. Small (44x22) and tiny (44x19) boards are for PAL and NTSC HighRes displays, respectively.

Note, that you better adjust this by simply resizing the window and using the MUI prefs to fix a certain window size. That way you can determine the windows size and position and not only the chessboards resolution.

-coords or -showCoords

If this option is true, AmyBoard displays algebraic coordinates along the board's left and bottom edges. The default is false. MUI's tiny font, which you can select with the MUI prefs, will be used.

-flip or -flipView

If you are playing a game on the ICS, the board is always oriented at the start of the game so that your pawns move from the bottom of the window towards the top. Otherwise, the starting is determined by the flipView option; if it is false (the default), White's pawns move from bottom to top at the start of each game; if it is true, Black's pawns move from bottom to top.

-bm or -bitmapDirectory

By default, AmyBoard uses a set of compiled-in bitmaps for its pieces. If the bitmapDirectory option is set at runtime, piece bitmaps are taken from files in the specified directory instead. The first character of a bitmap name gives the piece it represents (p, n, b, r, q, or k); the next characters give the size in pixels (80, 64, or 40), and the following character indicates whether the piece is solid or outline (s or o). Bitmap file names have the extension .bm. The outline bitmaps are used only in monochrome mode.

```
     Two sets of bitmaps are distributed with AmyBoard. Those in the
     directory bitmaps are normally compiled in as the default.  Those
     in the bitmaps.xchess directory can be selected at runtime with
     the bitmapDirectory option. If you want to compile in the latter
     set as the default, rename the bitmaps directory to bitmaps.fselch
     and the bitmaps.xchess directory to bitmaps; then recompile
     AmyBoard.

-wpp or -whitePiecePen
-bpp or -blackPiecePen
-lsp or -lightSquareColor
-dsp or -darkSquareColor
     Pen specifications, defaults are -1, in which case MUI's Shine-,
     Shadow-, Halfshine- and Halfshadowpen, respectively, will be used.
     Thus you'd better use the MUI prefs to fix these settings.
```

1.9 AmyBoard.guide/Other options

```
Miscellaneous
=============

-ncp or -noChessProgram
     If this option is true, AmyBoard acts as a passive chessboard; it
     does not start a chess program at all. Turning on this option also
     turns off clockMode. Default: false.

-debug or -debugMode
     Turns on debugging printout.
```

1.10 AmyBoard.guide/Menus

```
                Menus, Buttons and Keys
***********************

   All AmyBoard commands are available on menus. The most frequently
used commands also have shortcut keys or on-screen buttons.



                 File Menu
                        Accessing external games and positions.

                 Mode Menu
                        Selecting AmyBoard's mode.

                 Action Menu
                      Talking to GNU Chess or ICS opponents.

                 Step Menu
```

```
                        Controlling the Game.

                 Options Menu
                     User preferences

                 Help Menu
                        Getting help from GNU Chess.
```

1.11 AmyBoard.guide/File Menu

```
                Accessing external games and positions
=======================================

Reset
     Resets AmyBoard and GNU Chess to the beginning of a new chess
     game. The r key is a keyboard equivalent. In Internet Chess Server
     mode, clears the current state of AmyBoard, then resynchronizes
     with ICS by sending a refresh command. If you want to stop
     playing, observing, or examining a game on ICS, use an appropriate
     command from the Action menu, not Reset. See
                Action Menu
                .

Load Game
     Plays a game from a record file. The g key is a keyboard
     equivalent.  An ASL requester prompts you for the file name. If
     the file contains more than one game, a window displays a list of
     games (with information drawn from their PGN tags, if any), and
     you can select the one you want. Alternatively, you can load the
     Nth game in the file directly, by typing the number N after the
     file name, separated by the , character.

     The game file parser will accept PGN (portable game notation), or
     in fact almost any file that contains moves in algebraic notation.
     If the file includes a PGN position (FEN tag), or an AmyBoard
     position diagram bracketed by [- and -] before the first move, the
     game starts from that position. Text enclosed in parentheses,
     square brackets, or curly braces is assumed to be commentary and
     is displayed in a pop-up window. Any other text in the file is
     ignored. PGN variations (enclosed in parentheses) are treated as
     comments; AmyBoard is not able to walk variation trees.

Load Next Game
     Loads the next game from the last game record file you loaded.
     The shifted N key is a keyboard equivalent.  Not available if the
     last game was loaded from a pipe.

Load Previous Game
     Loads the previous game from the last game record file you loaded.
     The shifted P key is a keyboard equivalent.  Not available if the
     last game was loaded from a pipe.

Reload Same Game
```

```
     Reloads the last game you loaded.

Load Position
     Sets up a position from a position file.  A popup dialog prompts
     you for the file name. If the file contains more than one saved
     position, and you want to load the Nth one, type the number N
     after the file name, separated by a space. Position files must be
     in FEN (Forsythe-Edwards notation), or in the format that the Save
     Position command writes when oldSaveStyle is turned on.

Save Game
     Appends a record of the current game to a file.  An ASL requester
     prompts you for the file name. If the game did not begin with the
     standard starting position, the game file includes the starting
     position used. Games are saved in the PGN (portable game notation)
     format, unless the oldSaveStyle option is true, in which case they
     are saved in an older format that is specific to AmyBoard. Both
     formats are human-readable, and both can be read back by the Load
     Game command.

Save Position
     Appends a diagram of the current position to a file.  An ASL
     requester prompts you for the file name. Positions are saved in
     FEN (Forsythe-Edwards notation) format unless the oldSaveStyle
     option is true, in which case they are saved in an older,
     human-readable format that is specific to AmyBoard. Both formats
     can be read back by the Load Position command.

Mail Move
Reload CMail Message
     See the manual page for cmail(6).

About
     Shows the current AmyBoard version number.

Exit
     Exits from AmyBoard. The q key is a keyboard equivalent.
```

1.12 AmyBoard.guide/Mode Menu

```
Selecting AmyBoard's mode
=========================

Machine White
     Forces GNU Chess to play White. GNU Chess mode only.

Machine Black
     Forces GNU Chess to play Black. GNU Chess mode only.

Two Machines
     Plays a game between two computer programs. GNU Chess mode only.

ICS Client
```

ICS mode only. Takes AmyBoard out of the Edit Game or Edit Position state.

While you are examining a game on the ICS, you can issue the ICS position-editing commands with the mouse. (Do this with ICS Client selected on the Mode menu, not Edit Position; the latter edits only your local copy of the position.)

To drop a new piece on a square, press the left mouse button on the square, while holding down the Shift key. This brings up a menu of white and black pieces. Additional menu choices let you empty the square or clear the board. You cannot set the side to play or drag pieces to arbitrary squares while examining on ICS, however; the ICS permits only legal moves in this mode.

Edit Game

Allows you to make moves for both Black and White, and to change moves after backing up with the Backward command. The clocks do not run.

In GNU Chess mode, GNU chess continues to check moves for legality but does not participate in the game. You can bring GNU Chess back into the game by selecting Machine White, Machine Black, or Two Machines.

In ICS mode, the moves are not sent to the ICS: Edit Game takes AmyBoard out of ICS Client mode and lets you edit games locally. f you want to edit games on ICS in a way that other ICS users can see, use the ICS examine command or start an ICS match against yourself.

Edit Position

Lets you set up an arbitrary board position. Use the left mouse button to drag pieces to new squares, or to delete a piece by dragging it off the board or dragging an empty square on top of it. To drop a new piece on a square, press the left mouse button while holding down the shift key. This brings up a menu of white and black pieces. Additional menu choices let you empty the square, clear the board or set the side to play next. Selecting Edit Position causes AmyBoard to discard all remembered moves in the current game.

In ICS mode, changes made to the position by Edit Position are not sent to the ICS: Edit Position takes AmyBoard out of ICS Client mode and lets you edit positions locally. If you want to edit positions on ICS in a way that other ICS users can see, use the ICS examine command, or start an ICS match against yourself. (See also the ICS Client topic above.)

Show Game List

Shows or hides the list of games generated by the last Load Game command.

Edit Tags

Lets you edit the PGN (portable game notation) tags for the current game. After editing, the tags must still conform to the PGN tag syntax:

```
<tag-section> ::= <tag-pair> <tag-section>
      <empty>
<tag-pair> ::= [ <tag-name> <tag-value> ]
<tag-name> ::= <identifier>
<tag-value> ::= <string>
```

See the PGN Standard for full details. Here is an example:

```
[Event "Portoroz Interzonal"]
[Site "Portoroz, Yugoslavia"]
[Date "1958.08.16"]
[Round "8"]
[White "Robert J. Fischer"]
[Black "Bent Larsen"]
[Result "1-0"]
```

Any characters that do not match this syntax are silently ignored. Note that the PGN standard requires all games to have at least the seven tags shown above. Any that you omit will be filled in by AmyBoard with ? (unknown value), or - (inapplicable value).

Edit Comment

Adds or modifies a comment on the current position. Comments are saved by Save Game and are displayed by Load Game, Forward, and Backward.

Pause

Pauses updates to the board, and if you are playing against GNU Chess, also pauses your clock. To continue, select Pause again, and the display will automatically update to the latest position. The P button and keyboard p key are equivalents.

If you select Pause when you are playing against GNU Chess and it is not your move, GNU Chess's clock will continue to run and it will eventually make a move, at which point both clocks will stop. Since board updates are paused, however, you will not see the move until you exit from Pause mode (or select Forward). This behavior is meant to simulate adjournment with a sealed move.

If you select Pause while you are in examine mode on ICS, you can step backward and forward in the current history of the examined game without affecting the other observers and examiners. Select Pause again to reconnect yourself to the current state of the game on ICS.

If you select Pause while you are loading a game, the game stops loading. You can load more moves manually by selecting Forward, or resume automatic loading by selecting Pause again.

1.13 AmyBoard.guide/Action Menu

Talking to GNU chess or ICS opponents
=====================================

Accept

Accepts a pending match offer. If there is more than one offer pending, you will have to type in a more specific command instead of using this menu choice. (ICS mode only)

Decline

Declines a pending offer (match, draw, adjourn, etc.). If there is more than one offer pending, you will have to type in a more specific command instead of using this menu choice. (ICS mode only)

Call Flag

Calls your opponent's flag, claiming a win on time, or claiming a draw if you are both out of time. You can also call your opponent's flag by clicking on his clock or by pressing the keyboard t key.

Draw

Offers a draw to your opponent, accepts a pending draw offer from your opponent, or claims a draw by repetition or the 50-move rule, as appropriate. The d key is a keyboard equivalent. (Not available in GNU Chess mode.)

Adjourn

Asks your opponent to agree to adjourning the current game, or agrees to a pending adjournment offer from your opponent. (ICS mode only)

Abort

Asks your opponent to agree to aborting the current game, or agrees to a pending abort offer from your opponent. An aborted game ends immediately without affecting either player's rating.

Resign

Resigns the game to your opponent. The shifted R key is a keyboard equivalent.

Stop Observing

Ends your participation in observing a game, by issuing the ICS observe command with no arguments. (ICS mode only)

Stop Examining

Ends your participation in examining a game, by issuing the ICS unexamine command. (ICS mode only)

1.14 AmyBoard.guide/Step Menu

Controlling the game
====================

Backward

Steps backward through a series of remembered moves. The [<] button and the "b" key are equivalents.

In most modes, Backward only lets you look back at old positions; it does not retract moves. This is the case if you are playing against GNU Chess, playing or observing a game on the ICS, or loading a game. If you select Backward in any of these situations, you will not be allowed to make a different move. Use Retract Move or Edit Game if you want to change past moves.

If you are examining a game on ICS, the behavior of Backward depends on whether AmyBoard is in Pause mode. If Pause mode is off, Backward issues the ICS backward command, which backs up everyone's view of the game and allows you to make a different move. If Pause mode is on, Backward only backs up your local view.

Forward

Steps forward through a series of remembered moves (undoing the effect of Backward) or forward through a game file. The [>] button and the f key are equivalents.

If you are examining a game on ICS, the behavior of Forward depends on whether AmyBoard is in Pause mode. If Pause mode is off, Forward issues the ICS forward command, which moves everyone's view of the game forward along the current line. If Pause mode is on, Forward only moves your local view forward, and it will not go past the position that the game was in when you paused.

Back to Start

Jumps backward to the first remembered position in the game. The [<<] button and the shifted B key are equivalents.

In most modes, Back to Start only lets you look back at old positions; it does not retract moves. This is the case if you are playing against GNU chess, playing or observing a game on the ICS, or loading a game. If you select Back to Start in any of these situations, you will not be allowed to make different moves. Use Retract Move or Edit Game if you want to change past moves; or use Reset to start a new game.

If you are examining a game on ICS, the behavior of Back to Start depends on whether AmyBoard is in Pause mode. If Pause mode is off, Back to Start issues the ICS backward 999999 command, which backs up everyone's view of the game to the start and allows you to make different moves. If Pause mode is on, Back to Start only backs up your local view.

Forward to End

Jumps forward to the last remembered position in the game. The [>>] button and the shifted F key are equivalents.

If you are examining a game on ICS, the behavior of Forward to End depends on whether AmyBoard is in Pause mode. If Pause mode is off, Forward to End issues the ICS forward 999999 command, which moves everyone's view of the game forward to the end of the current line. If Pause mode is on, Forward to End only moves your local view forward, and it will not go past the position that the game was in when you paused.

Revert

If you are examining a game on ICS and Pause mode is off, issues

```
     the ICS command revert.

Truncate Game
     Discards all remembered moves of the game beyond the current
     position. Puts AmyBoard into Edit Game mode if it was not there
     already.

Move Now
     Forces GNU Chess to move immediately. (GNU Chess mode only)

Retract Move
     Retracts your last move. In GNU Chess mode, you can do this only
     after GNU Chess has replied to your move; if GNU Chess is still
     thinking, use Move Now first. In ICS mode, Retract Move issues the
     command takeback 1 or takeback 2 depending on whether it is your
     opponent's move or yours.
```

1.15 AmyBoard.guide/Options Menu

```
                User Preferences
================

Always Queen
     Toggles the alwaysPromoteToQueen option.  See

                User interface options
                .

Auto Comment
     Toggles the autoComment option.

Auto Flag
     Toggles the autoCallFlag option: AmyBoard will automatically claim
     a win for you, if your opponents flag falls on ICS.  See

                ICS options
                .

Auto Observe
     Toggles the autoObserve option. See
                ICS options
                .

Auto Save
     Toggles the autoSaveGames option. Disabled if the saveGameFile
     option is set, as in that case all games are saved to the
     specified file.

Bell
     Toggles the ringBellAfterMoves option. See
                User interface options
                .

Flip View
```

Inverts your view of the chess board for the duration of the current game. Starting a new game returns the board to normal.

If you are playing a game on the ICS, the board is always oriented at the start of the game so that your pawns move from the bottom of the window towards the top. Otherwise, the starting orientation is determined by the flipView command line option; if it is false (the default), White’s pawns move from bottom to top at the start of each game; if it is true, Black’s pawns move from bottom to top. See
User interface options
.

Old Save Style
Toggles the oldSaveStyle option. See
I-O options
.

Quiet Play
Toggles the quietPlay option. See
ICS options
.

Show Coords
Toggles the showCoords option. See
User interface options
.

Show Thinking
Toggles the showThinking option. See
GNU Chess options
.

1.16 AmyBoard.guide/Help Menu

Getting help from GNU Chess
===========================

Hint
Displays a move hint from GNU Chess. GNU Chess mode only.

Book
Displays a list of possible moves from GNU Chess’s opening book. The first column gives moves, the second column gives one possible response for each move, and the third column shows the number of lines in the book that include the move from the first column. If you select this option and nothing happens, GNU Chess is out of its book. GNU Chess mode only.

About Game
Displays information about the current game, in the form of PGN (portable game notation) tags.

1.17 AmyBoard.guide/ICS

```
                Using AmyBoard as an ICS client
*******************************
```

What is this ICS thing, you probably ask? ICS (Internet chess server, also known as ICC, Internet chess club, which is the commercial version) is an internet service very similar to irc: People use a client program like AmyBoard to connect to a server program on a remote host, the so-called server. There they meet to play chess, watch other people's games or just chat: In fact, the atmoshpere is full of virtual shouts, much like a usual chess club.

ICS is getting more and more important for chess players: Currently (March 1995) there are some thousand registered users on the different ICS hosts and it is not unusual to meet 200-300 of them on ICC in the evening hours. The number is increasing rapidly.

Usually you are using telnet (1) to connect to the chess server. (N{No Value For "AME"} does this for you.) However, telnet assumes a SLIP or PPP connection or something similar to the Internet. AmyBoard is not bound to this, however: You can replace telnet by any program reading from stdin and writing to stdout, for example a simple terminal program.

```
                 Serial ICS connection
                    Using a terminal program to connect to ICS.
```

When using AmyBoard as an ICS client, the program seems to be split into two parts: The first one is the chessboard, as usual. Additionally you are using the terminal window as a possibility to type in commands and read information not available on the chessboard.

The first time you are using the terminal is to enter your login name and password, if you are a registered player. (You don't need to do this manually, the icsLogon option may do this for you. see

ICS options

) If not, enter any name. If someone has already registered under that name, you'll be asked for a password; just hit return and try again.

You may use the terminal window to enter commands. Useful ICS commands include

help <topic>
to get help on the given <topic>. (To get a list of possible topics type help without topic.) Obviously this is a very important command for new users. :-) It is recommended netiquette to use it before asking other people.

  For example help register tells you how to become a registered ICS

```
     player.

who <flags>
     to see a list of people displayed who are are logged on.
     Administrators (people you should ask first, if you have something
     to ask) are marked with the character *, an asterisk. The <flags>
     allow you to display only selected players: For example, who of
     shows a list of players which are interested in playing, but do
     not have an opponent.

games
     to see what games are being played

match <player> [<mins>] [<inc>]
     to challenge another player to a game. Both opponents get <mins>
     minutes for the game, <inc> seconds will be added after each move.
     If another player challenges you, the server asks, if you want to
     accept the challenge: Use the accept or decline commands to answer.

accept
decline
     to accept or decline another players offer; this is both used to
     start a game (if the player challenges you) or to terminate the
     game, for example, if your opponent offers you a draw, to adjourn
     or abort the game. See
                Action Menu
                .

     If you have more than one pending offer (for example, if more than
     one player is challenging you, or if your opponent offers both a
     draw and to adjourn the game), you have to supply additional
     information, either by typing something like accept <player>,
     accept draw or using a command like

draw
adjourn
abort
     offers your opponent to terminate a game by mutual agreement.
     (Adjourned games may be continued later.) The server asks your
     opponent, so he can either decline your offer or accept it by
     typing the same command or accept. (Your opponent won’t be asked,
     of course, if he lacks material to win, for example.)

finger <player>
     to get informations about the given <player>. (Default: yourself.)

vars
     to get a list of personal settings

set <var> <value>
     to modify these settings

observe <player>
     to observe an ongoing game of the given <player>.

examine
oldmoves
```

to review a recently completed game

Whenever you ask to observe an ongoing game, review a completed game, or resume an adjourned game, AmyBoard retrieves and parses the list of past moves from the ICS, so you can review them with the Forward and Backward commands or save them with Save Game. Some special AmyBoard features are activated when you are in examine mode on ICS.

1.18 AmyBoard.guide/Serial ICS connection

Using a terminal program to connect to ICS.

As already mentioned, you may connect to the ICS with any terminal program, provided that it can read from stdin and write to stdout. An example (which is well suuported by the installation script) is IPDial, which you find in the comm/tcp directory on Aminet. However, you need to connect to a host with Internet access which will act as a gateway to the ICS server.

Usually you do the following:

* Create a login script for your terminal program that allows to login to your remote host. Probably you add a line like

```
          SEND "telnet chess.lm.com 5000"
```

  to the login script: This would make the remote host to create a connection to the ICS server chess.lm.com (ICC) as the very first command to execute. The installation script can create such a login script for you, to be used by IPDial (PROGDIR:lib/IcsLogin.IPDial, where PROGDIR: is the directory of the AmyBoard binary). However, you need to edit it manually, insert your remote hosts phone number, your login name and password, for example.

* Create an icon that calls your terminal program like it would do with telnet; the installation script can do this for you.

1.19 AmyBoard.guide/Limitations

Known limitations
****

There is no way for two people running copies of AmyBoard to play each other without going through the Internet Chess Server.

The game parser recognizes only algebraic notation.

The internal move legality tester does not look at the game history, so in some cases it misses illegal castling or en passant captures.

However, if you attempt such a move when using GNU Chess (or the ICS), it will reject the move, and AmyBoard will undo it and let you try another. Also FEN positions saved by AmyBoard do not include correct information about whether castling or en passant are legal.

In ICS mode AmyBoard cannot handle observing (and/or playing) more than one game at a time. It may get confused if you try to do this, thought it tries to recover gracefully. Select Reset from the File menu, if you have problems. See
File Menu
.

Your password may get echoed when you log on to the ICS.

If you are connecting to the ICS by running telnet on an Internet provider or firewall host, you may find that each line you type is echoed back an extra time after you hit Return. If your Internet provider is a Unix system, you can probably turn its echo off by typing

```
stty -echo
```

after you log in, and/or by typing ^E-Return (control-E followed by the Return key) to the telnet program after you have logged into ICS. It is a good idea to do this if you can, because the extra echo can occasionally confuse AmyBoard's parsing routines.

The .icsrc file does not work properly when you connect to ICS through a Unix gateway host with useTelnet mode. The Unix login process apparently discards type-ahead. See
ICS options
. Due to a
problemm which I did not figure out yet, it might be needed to add an empty line to the top of the .icsrc file.

Some AmyBoard functions may not work with versions of GNU Chess earlier than 4.0, patchlevel 73.

1.20 AmyBoard.guide/Bugs

Reporting Bugs
****

Report bugs in AmyBoard or GNU Chess to
wiedmann@neckar-alb.de

Please run AmyBoard from the CLI with the debug option and include the output in your message. Also tell me what machine and what version of both AmigaOS and MUI you are using.

1.21 AmyBoard.guide/Caveats

What you should keep in mind
****

- The most common problem is, that people don't see the pieces, after they have started AmyBoard the first time. The reason is simple: Their color settings in the MUI prefs don't fit AmyBoard's requirements. To fix this problem you have two possibilities
  1. Use the MUI prefs to fix appropriate colors for AmyBoard; the program uses MUI's Shinepen, Shadowpen, Halfshinepen and Halfshadowpen. Be sure, that these are all different.

  2. Enter pen settings in the prefs file of AmyBoard (usually PROGDIR:lib/AmyBoard.prefs) manually.

- Another common problem are strange crashes, especially in ICS mode. This can happen, if your stack is too low: A value of 10000 should be sufficient, but the CLI default value of 4096 definitely isn't. Note, that Toolmanager uses the same value by default: You must increase it in the Exec object, if you want to start AmyBoard from Toolmanager.

1.22 AmyBoard.guide/Contributors

Authors and Contributors to AmyBoard
****

Tim Mann (mann@pa.dec.com)
has been responsible for XBoard versions 1.3 through 3.3 and for WinBoard, a port of XBoard to Microsoft Win32 (Windows NT and Windows 95)

Jochen Wiedmann (wiedmann@neckar-alb.de)
ported XBoard to the Amiga, creating AmyBoard, and converted the documentation to texinfo.

Chris Sears and Dan Sears
wrote the original XBoard; they were responsible for versions 1.0 through 1.2.

Elmar Bartel (bartel@informatik.tu-muenchen.de)
Elmar Bartel contributed the new piece bitmaps for version 3.2.

Evan Welsh (welsh@epcc.ed.ac.uk)
wrote CMail.

Patrick Surry
helped with design, testing and documening CMail.

John Chanak
contributed the iniitial implementation of ICS mode.

Wayne Christopher
created XChess; the color scheme and the old 80x80 piece bitmaps

were taken from it.

1.23 AmyBoard.guide/Index

Index
*****