GNU Backgammon

version 0.02
February, 2000

Gary Wong

1 Introduction

GNU Backgammon (`gnubg`) plays and analyses backgammon games and matches.

FIXME update this section (latest version is on web page).

It is currently a work-in-progress. So far it is able to play cubeless games and tournament matches, evaluate and roll out positions, tune its own evaluation functions using either TD or supervised training, maintain databases of positions for training and other purposes, and more.

It is driven by a command-line interface, and displays an ASCII rendition of a board on text-only terminals, but also allows the user to play games and manipulate positions with an X11 board window where available.

Plans for the future include completing its ability to play and analyse cubeful games and matches, and making it extensible on platforms where Guile is available, so that the user can program it to answer sophisticated questions like "How many chequers must Black have borne off in positions like this one to be able to accept a double?" and automate complicated rollout tasks.

It currently plays at about the level of a strong intermediate flight tournament player (rated in the mid 1700s on FIBS, the First Internet Backgammon Server — around the 90th percentile) and is gradually improving; it should be somewhat stronger than this when released. Since almost all of the CPU time required during supervised training is spent performing rollouts, and rollouts can easily be performed in parallel, it is hoped that users will be able to pool rollout results and collectively train it to a level stronger than any individual could obtain.

If you can tolerate the rough edges, periodic snapshots of `gnubg` and anonymous CVS access to the pre-release source are being made available. You can also compete against recent versions of `gnubg` on FIBS; it plays there under the names '`gnu`' and '`mgnutest`'.

2 How to Play Backgammon

The rules presented in this chapter were written by Tom Keith for the Backgammon Galore web site, and are included here with his permission.

2.1 Rules of Backgammon

2.1.1 Setup

Backgammon is a game for two players, played on a board consisting of twenty-four narrow triangles called *points*. The triangles alternate in color and are grouped into four quadrants of six triangles each. The quadrants are referred to as a player's *home board* and *outer board*, and the opponent's home board and outer board. The home and outer boards are separated from each other by a ridge down the center of the board called the *bar*.

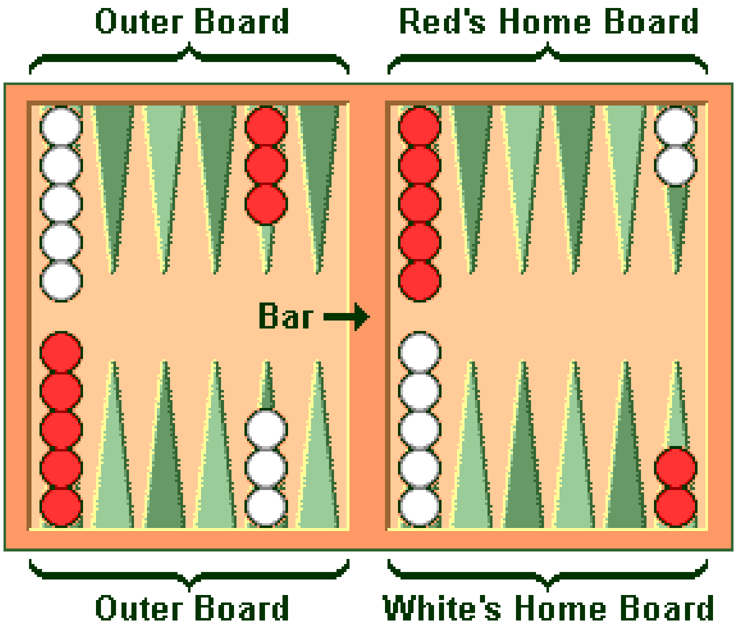

Figure 1. A board with the checkers in their initial position.

An alternate arrangement is the reverse of the one shown here, with the home board on the left and the outer board on the right.

The points are numbered for either player starting in that player's home board. The outermost point is the twenty-four point, which is also the opponent's one point. Each player has fifteen checkers of his own color. The initial arrangement of checkers is: two on each player's twenty-four point, five on each player's thirteen point, three on each player's eight point, and five on each player's six point.

Both players have their own pair of dice and a dice cup used for shaking. A *doubling cube*, with the numerals 2, 4, 8, 16, 32, and 64 on its faces, is used to keep track of the current stake of the game.

2.1.2 Object of the Game

The object of the game is for a player to move all of his checkers into his own home board and then bear them off. The first player to bear off all of his checkers wins the game.

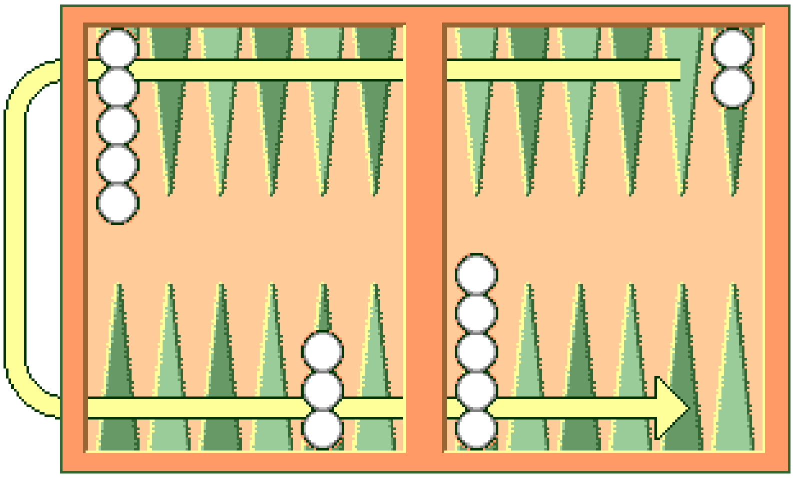

Figure 2. Direction of movement of White's checkers. Red's checkers move in the opposite direction.

2.1.3 Movement of the Checkers

To start the game, each player throws a single die. This determines both the player to go first and the numbers to be played. If equal numbers come up, then both players roll again until they roll different numbers. The player throwing the higher number now moves his checkers according to the numbers showing on both dice. After the first roll, the players throw two dice and alternate turns.

The roll of the dice indicates how many points, or *pips*, the player is to move his checkers. The checkers are always moved forward, to a lower-numbered point. The following rules apply:

1. A checker may be moved only to an *open point*, one that is not occupied by two or more opposing checkers.
2. The numbers on the two dice constitute separate moves. For example, if a player rolls 5 and 3, he may move one checker five spaces to an open point and another checker three spaces to an open point, or he may move the one checker a total of eight spaces to an open point, but only if the intermediate point (either three or five spaces from the starting point) is also open.

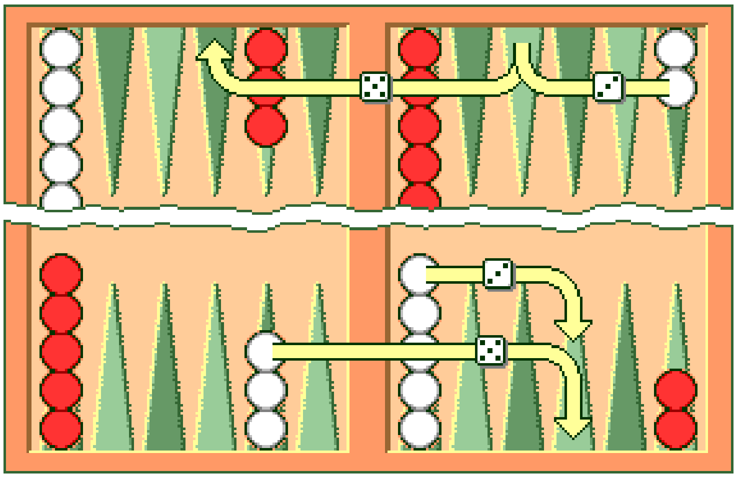

Figure 3. Two ways that White can play a roll of 53.

3. A player who rolls doubles plays the numbers shown on the dice twice. A roll of 6 and

6 means that the player has four sixes to use, and he may move any combination of checkers he feels appropriate to complete this requirement.

4. A player must use both numbers of a roll if this is legally possible (or all four numbers of a double). When only one number can be played, the player must play that number. Or if either number can be played but not both, the player must play the larger one. When neither number can be used, the player loses his turn. In the case of doubles, when all four numbers cannot be played, the player must play as many numbers as he can.

2.1.4 Hitting and Entering

A point occupied by a single checker of either color is called a *blot*. If an opposing checker lands on a blot, the blot is *hit* and placed on the bar.

Any time a player has one or more checkers on the bar, his first obligation is to *enter* those checker(s) into the opposing home board. A checker is entered by moving it to an open point corresponding to one of the numbers on the rolled dice.

For example, if a player rolls 4 and 6, he may enter a checker onto either the opponent's four point or six point, so long as the prospective point is not occupied by two or more of the opponent's checkers.

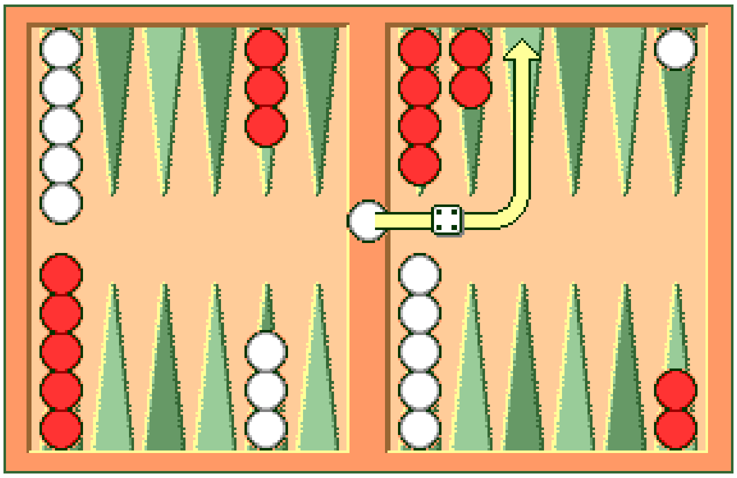

Figure 4. If White rolls 64 with a checker on the bar, he must enter the checker onto Red's four point since Red's six point is not open.

If neither of the points is open, the player loses his turn. If a player is able to enter some but not all of his checkers, he must enter as many as he can and then forfeit the remainder of his turn.

After the last of a player's checkers has been entered, any unused numbers on the dice must be played, by moving either the checker that was entered or a different checker.

2.1.5 Bearing Off

Once a player has moved all of his fifteen checkers into his home board, he may commence *bearing off*. A player bears off a checker by rolling a number that corresponds to the point on which the checker resides, and then removing that checker from the board. Thus, rolling a 6 permits the player to remove a checker from the six point.

If there is no checker on the point indicated by the roll, the player must make a legal move using a checker on a higher-numbered point. If there are no checkers on higher-numbered points, the player is permitted (and required) to remove a checker from the highest point on which one of his checkers resides. A player is under no obligation to bear off if he can make an otherwise legal move.

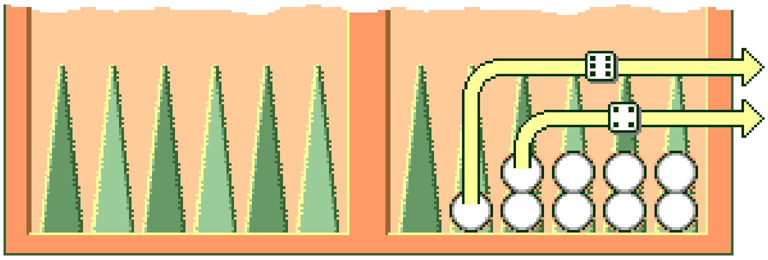

Figure 5. White rolls 64 and bears off two checkers.

A player must have all of his active checkers in his home board in order to bear off. If a checker is hit during the bear-off process, the player must bring that checker back to his home board before continuing to bear off. The first player to bear off all fifteen checkers wins the game.

2.1.6 Doubling

Backgammon is played for an agreed stake per point. Each game starts at one point. During the course of the game, a player who feels he has a sufficient advantage may propose doubling the stakes. He may do this only at the start of his own turn and before he has rolled the dice.

A player who is offered a double may *refuse*, in which case he concedes the game and pays one point. Otherwise, he must *accept* the double and play on for the new higher stakes. A player who accepts a double becomes the *owner of the cube* and only he may make the next double.

Subsequent doubles in the same game are called *redoubles*. If a player refuses a redouble, he must pay the number of points that were at stake prior to the redouble. Otherwise, he becomes the new owner of the cube and the game continues at twice the previous stakes. There is no limit to the number of redoubles in a game.

2.1.7 Gammons and Backgammons

At the end of the game, if the losing player has borne off at least one checker, he loses only the value showing on the doubling cube (one point, if there have been no doubles). However, if the loser has *not* borne off any of his checkers, he is *gammoned* and loses *twice* the value of the doubling cube. Or, worse, if the loser has not borne off any of his checkers and still has a checker on the bar or in the winner's home board, he is *backgammoned* and loses *three times* the value of the doubling cube.

2.1.8 Optional Rules

The following optional rules are in widespread use.

Automatic doubles
If identical numbers are thrown on the first roll, the stakes are doubled. The doubling cube is turned to 2 and remains in the middle. Players usually agree to limit the number of automatic doubles to one per game.

Beavers
When a player is doubled, he may immediately redouble (beaver) while retaining possession of the cube. The original doubler has the option of accepting or refusing as with a normal double.

The Jacoby Rule
Gammons and backgammons count only as a single game if neither player has offered a double during the course of the game. This rule speeds up play by eliminating situations where a player avoids doubling so he can play on for a gammon.

2.1.9 Irregularities

1. The dice must be rolled together and land flat on the surface of the right-hand section of the board. The player must reroll both dice if a die lands outside the right-hand board, or lands on a checker, or does not land flat.
2. A turn is completed when the player picks up his dice. If the play is incomplete or otherwise illegal, the opponent has the option of accepting the play as made or of requiring the player to make a legal play. A play is deemed to have been accepted as made when the opponent rolls his dice or offers a double to start his own turn.
3. If a player rolls before his opponent has completed his turn by picking up the dice, the player's roll is voided. This rule is generally waived any time a play is forced or when there is no further contact between the opposing forces.

2.2 Rules for Match Play

When backgammon tournaments are held to determine an overall winner, the usual style of competition is *match play*. Competitors are paired off, and each pair plays a series of games to decide which player progresses to the next round of the tournament. This series of games is called a *match*.

Matches are played to a specified number of points. The first player to accumulate the required points wins the match. Points are awarded in the usual manner: one for a single game, two for a gammon, and three for a backgammon. The doubling cube is used, so the winner receives the value of the game multiplied by the final value of the doubling cube.

Matches are normally played using the *Crawford rule*. The Crawford rule states that if one player reaches a score one point short of the match, neither player may offer a double in the immediately following game. This one game without doubling is called the Crawford game. Once the Crawford game has been played, if the match has not yet been decided, the doubling cube is active again.

Match to 5		White	Black	
Game 1:	White wins 2 points	2	0	Doubling Allowed
Game 2:	Black wins 1 point	2	1	
Game 3:	White wins 2 points	4	1	
Game 4:	Black wins 1 point	4	2	Crawford Game
Game 5:	Black wins 2 points	4	4	Doubling Allowed
Game 6:	White wins 2 points	6	4	

In this example, White and Black are playing a 5-point match. After three games White has 4 points, which is just one point short of what he needs. That triggers the Crawford rule which says there can be no doubling in next game, Game 4.

There is no bonus for winning more than the required number of points in match play. The sole goal is to win the match, and the size of the victory doesn't matter.

Automatic doubles, beavers, and the Jacoby rule are not used in match play.

3 Sample Game

Although GNU Backgammon has many commands and options, only a few of them are used during routine play. This chapter demonstrates some of those most commonly used.

If your `gnubg` installation is complete, you should be able to start it and see output similar to the following:

```
GNU Backgammon 0.02  Copyright 1999 Gary Wong.
GNU Backgammon is free software, covered by the GNU General Public
License, and you are welcome to change it and/or distribute copies
of it under certain conditions.  Type "show copying" to see the
conditions.
There is absolutely no warranty for GNU Backgammon.  Type "show
warranty" for details.
(No game)
```

When you start a game, `gnubg` will (by default) show which player's move it is; it is displaying '`(No game)`' at the moment to indicate that it isn't anybody's turn.

GNU Backgammon uses a pseudo-random number generator to produce dice rolls. It will normally initialise the state of the generator to some value based on the current time (and perhaps other system information, depending what features are available on your platform). But for purposes of this example, it will be better to start the generator from a known state, so that the rolls it generates will be repeatable. Enter the following command to change the generator state ("seed"):

```
(No game) set seed 15
Seed set to 15.
(No game)
```

The default settings allow you to play X (the black pieces, if you have a graphical board window) and the computer will play O (red). It will play with no lookahead by default; it is very fast at choosing moves in this mode (instantaneous, on most hardware) though it does not play at its full strength. Let's start a game.

```
(No game) new game
O rolls 1, X rolls 3.
 GNU Backgammon  Position ID: 4HPwATDgc/ABMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    |   | O              X |
 | X           O    |   | O              X |
 | X           O    |   | O                |
 | X                |   | O                |
 | X                |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   | X                |
 | O                |   | X                |
 | O           X    |   | X                |
 | O           X    |   | X              O |
 | O           X    |   | X              O |     Rolled 13
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X

(X)
```

This is the output you will see if a graphical board is unavailable or disabled; if your environment supports a graphical board, you will see a better representation of a backgammon board than this. Exactly the same information is available, either way.

You have just won the opening roll (you have a 3, and `gnubg` has a 1) — the dice roll is in the bottom right hand corner for the TTY board, or shown as graphical dice on the right hand side of the board window. The position ID (the '`4HPwATDgc/ABMA`' in this example) is an encoded representation of the current position. It comes in useful when using some of `gnubg`'s advanced commands, but you can safely ignore it for now. Notice that the prompt has changed to '`(X)`', to indicate that it is X's (your) turn.

If you are familiar with backgammon, you will know that the best play for this roll is to make your 5 point. The command to do that is:

```
(X) 8 5 6 5
```

in the terminal window (i.e. move one chequer from your 8 point to your 5 point, and move another from your 6 to your 5 point); or by dragging each chequer to the destination point with mouse button 1 in the board window. Click mouse button 1 on the dice to indicate you have finished your move (just as you would pick up your dice to end your move if you were playing on a real board).

Once you have done that, `gnubg` will take its turn, rolling 63 and playing 24/15. It is now your move again:

```
 GNU Backgammon  Position ID: 4HPwCSCwZ/ABMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    |   | O              X |
 | X           O    |   | O              X |
 | X           O    |   | O                |
 | X                |   | O                |
 | X                |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   |                  |
 | O                |   | X                |
 | O                |   | X                |
 | O           X    |   | X  X             |
 | O     O     X    |   | X  X           O |     On roll
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X
(X)
```

Since it is no longer the opening roll, you have the option of doubling before rolling your dice. To roll without doubling, just enter the command ***roll*** (or ***r*** for short):

```
(X) roll
```

```
 GNU Backgammon  Position ID: 4HPwCSCwZ/ABMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    |   | O              X |
 | X           O    |   | O              X |
 | X           O    |   | O                |
 | X                |   | O                |
 | X                |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   |                  |
 | O                |   | X                |
 | O                |   | X                |
 | O           X    |   | X  X             |
 | O     O     X    |   | X  X           O |     Rolled 35
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X
(X)
```

(You can also roll by clicking the dice just underneath the board window.)

Let's play this roll by hitting twice: 13/10*, 6/1*. You can enter this move with *13 10 6 1* — note that you don't need to specify anything special to hit; `gnubg` will automatically move blots that are hit to the bar. You can also make this move by dragging chequers in the board window, of course.

```
(X) 13 10 6 1
 GNU Backgammon  Position ID: 4HPwAWBhZ+IBMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    | O | O              X |
 | X           O    | O | O              X |
 | X           O    |   | O                |
 | X                |   | O                |
 |                  |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   |                  |
 | O                |   |                  |
 | O                |   | X                |
 | O           X    |   | X  X             |
 | O     X     X    |   | X  X           X |     On roll
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X
(X)
```

GNU Backgammon has rolled 65 and been unable to move. Let's take another roll:

```
(X) roll
```

```
 GNU Backgammon  Position ID: 4HPwAWBhZ+IBMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    | O | O              X |
 | X           O    | O | O              X |
 | X           O    |   | O                |
 | X                |   | O                |
 |                  |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   |                  |
 | O                |   |                  |
 | O                |   | X                |
 | O           X    |   | X  X             |
 | O     X     X    |   | X  X           X |     Rolled 25
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X
(X)
```

Play this roll as *13 11 6 1*. gnubg will now roll 16 — and still can't move!

```
 GNU Backgammon  Position ID: 4HPwAWDDZsoBMA
 +13-14-15-16-17-18------19-20-21-22-23-24-+     O
 | X           O    | O | O              X |
 | X           O    | O | O              X |
 | X           O    |   | O                |
 |                  |   | O                |
 |                  |   | O                |
v|                  |BAR|                  |     (Cube: 1)
 | O                |   |                  |
 | O                |   |                  |
 | O                |   |                  |
 | O           X    |   | X  X           X |
 | O  X  X     X    |   | X  X           X |     On roll
 +12-11-10--9--8--7-------6--5--4--3--2--1-+     X
(X)
```

Our position is now strong. Let's try doubling — this can be done by entering the command *double* at the prompt, or by clicking the doubling cube in the board window (FIXME not implemented yet). When you are doubled, you should use the command *take* or *drop* to accept or refuse the cube. GNU Backgammon will drop this double (FIXME actually it doesn't, but it will once it knows how):

```
(X) double
O refuses the cube and gives up 1 point.
X wins a single game and 1 points.
The score (after 1 game) is: O 0, X 1.
```

That concludes the example game. You have seen most of the commands normally used while playing; others you will need include *bar* when re-entering a hit chequer (e.g. *bar 20 24 20*) and *off* to remove a chequer from the board when bearing off (e.g. *6 off 5 off*).

Type *help* from within gnubg for a summary of all commands, or browse the rest of this manual to learn more. Have fun playing GNU Backgammon!

4 Starting and Leaving gnubg

This chapter describes how to run `gnubg`, and how to end it when you are finished.

4.1 Invoking gnubg

Start GNU Backgammon by running the program *gnubg*. Once started, it will process commands from the terminal.

You can also instruct `gnubg` to read an existing match from a file, to play or analyse. (FIXME actually you can't yet, but you should be able to.) To read the match from *filename*, specify *gnubg filename*.

`gnubg` will give you a short summary of how to invoke it if given the option '`--help`' ('`-h`' for short), and report its version with '`--version`' ('`-v`').

GNU Backgammon will normally use a graphical board window under the X Window System, if possible. To prevent this behaviour and use standard terminal output instead, you can specify the '`--tty`' option (which can be abbreviated to '`-t`').

4.2 Leaving gnubg

`exit`
`quit` To exit `gnubg`, use the *exit* or *quit* commands (they are synonymous). If you are currently playing a game, the match in progress will be lost, so `gnubg` will ask you if you are sure you want to exit in this case. (FIXME it doesn't yet, but it should.)

`gnubg` will also exit upon end-of-file; you can therefore exit by typing an end-of-file character (usually *C-d*, i.e. hold the ⟨CTRL⟩ key and press D).

If you send an interrupt sequence (often by pressing *C-c*), the current `gnubg` command (if any) will be terminated. This can be useful if you do not want to wait for a slow command to complete.

4.3 Shell Commands

You can execute shell commands from `gnubg` without leaving or suspending it.

`!`*command*
Invoke a subshell to execute *command*. The environment variable `SHELL` determines which shell to use (`gnubg` defaults to '`/bin/sh`' if `$SHELL` is not set). If *command* is not specified, the shell will be invoked. FIXME not implemented yet.

5 Basic Commands

There are a few simple commands you can use within GNU Backgammon to display information you might need to know:

`help` *command*
`?` *command*
: Describe the commands that `gnubg` understands. `help` by itself lists the main commands with a one-line description of each; when *command* is specified, that command is explained in more detail, and if it has any subcommands then they are listed in brief.

  `?` is a synonym for `help`.

`show copying`
`show warranty`
: These commands are used to show the conditions GNU Backgammon is distributed under. FIXME summarise GPL.

6 Playing Backgammon with `gnubg`

FIXME introduction

6.1 Commands Used During the Game

`new game`
: This command is used to start a game within a match or session. (Note that all games are played within matches or sessions; see Section 6.4 [Matches and Sessions], page 17. To play a single game, use either a 1 point match or a session lasting for one game, depending whether you want gammons and the doubling cube to be active.) `new game` will set up the board in the starting position, and roll one die for each player for the opening move.

  If you are in the middle of a game, `new game` will ask you if you want to abort the game in progress. If you do, a new game will replace the current one (i.e. the partially completed game will have no effect on the score). If you want the current game to be scored for either player, you should use the `resign` command instead.

`move` *from to ...*
from to ...
: The `move` command allows you to make chequer plays when it is your turn. In its normal form, you should specify pairs of point numbers indicating the points you want to move a chequer from and to. Specify one pair for each chequer you want to move. (For instance, on an opening roll of 31, you might use `move 8 5 6 5` to move two chequers — one from your 8 point to your 5 point, and the other from your 6 point to your 5 point.) For several example moves, see Chapter 3 [Sample Game], page 8.

  You should use the words `bar` and `off` when moving a chequer from the bar or bearing it off the board, e.g. `move bar 20` or `move 3 off`. These words can be abbreviated to `b` and `o` respectively.

  If there is only one legal move you can make, then the command `move` by itself will make it for you without requiring you to specify it in full. Similarly, if there is no play available at all, then `move` will end your turn without moving.

  As long as you specify at least one pair of points, then the word `move` is optional — the command `bar 20 24 20` means exactly the same thing as `move bar 20 24 20`, for instance.

  If you are using the X Window System, you can also move chequers using the board window. One way to do this is to use mouse button 1 to drag a chequer (that is, press the button when the pointer is over the chequer you wish to move; move the pointer to the point you wish to play it to, and then release the button).

  An alternative is to click buttons 2 or 3 on the chequer; button 2 will move it by the number of pips showing on the left die, and button 3 will move it according to the right die. If you don't like the order the dice are displayed in, pressing either button 2 or 3 on the dice will swap their positions.

Whichever method you use to move the chequers, once you have made a legal move you can end your turn by clicking mouse button 1 on the dice.

FIXME reference set automove and set autobearoff.

`resign` [*type*]
`agree`
`accept`
`decline`
`reject`

The `resign` command is used to give up a game without playing it to completion. It is often useful during endgame play when the game reaches a position where it is impossible for one player to win. If you do not specify a value *type*, then the player whose turn it is offers to give up one game (at the current cube value) to the opponent; you can also specify `1`, `2` or `3` to resign a single, double or triple game. Specifying `normal`, `gammon` or `backgammon` is also legal, and is identical to expressing the number of points as a digit. `single` is yet another synonym for one game.

The opponent may accept the resignation with either the `agree` or `accept` commands, but is not obliged to. To ignore the resignation and continue play normally, use either the `decline` or `reject` commands. (`accept` and `reject` are also legal commands in response to a double; see Section 6.5 [Cube], page 18.)

`set board` *id*

If you wish to directly set the chequers to a different position, you can use the `set board` command. You need to know the *position ID* of the chequer arrangement you want; position IDs are always displayed when the board is shown. (If you are playing on a text terminal, the position ID is in the upper right hand corner of the board diagram; when using the X board window, the ID is shown below the board on the right hand side.) For instance, to set the chequers to the starting position, use the command `set board 4HPwATDgc/ABMA`.

FIXME add a mode to set the board to an arbitrary position when the user doesn't know the ID, in the X version at least.

`show board` [*id*]

The `show board` command is used to display a particular position — normally the board of the current game, but it is also possible to view unrelated layouts. When specified without the optional *id* parameter, the current position is displayed. (When using a text terminal, this is useful if subsequent output has caused the board diagram to scroll off the screen. In the X board window, the command can be used to undo erroneous chequer plays by resetting the window to the position at the start of the turn.)

When a position ID *id* is given, the chequers are arranged into the position specified and that board is displayed. Note that this command affects the display only; the current game is unchanged. Use the `show board` command with no parameter if you want to see the current game again.

`hint`

At any time during the game when the dice have been rolled, you can use the `hint` command to see the moves `gnubg` recommends. The output is of the following form:

```
Win    W(g)   W(bg)  L(g)   L(bg)  Equity       Move
0.542  0.142  0.008  0.113  0.008  (+0.114)     6/5 8/5
0.505  0.120  0.008  0.122  0.007  (+0.009)     24/23 23/20
0.498  0.126  0.008  0.123  0.007  (+0.000)     24/23 13/10
0.499  0.113  0.008  0.121  0.007  (-0.011)     24/23 24/21
0.486  0.125  0.008  0.120  0.009  (-0.024)     13/10 10/9
0.481  0.116  0.008  0.129  0.008  (-0.051)     6/5 24/21
0.472  0.122  0.008  0.129  0.009  (-0.064)     6/5 13/10
```

The moves are listed in descending order of preference, so in this case, `gnubg` recommends the move '`6/5 8/5`'. The first five columns are its estimates of the probability of the player on roll winning ('`Win`'), winning a gammon ('`W(g)`'), winning a backgammon ('`W(bg)`'), losing a gammon ('`L(g)`'), and losing a backgammon ('`L(bg)`') if the game is played to completion without use of the doubling cube, after the candidate move in that row is made[1]. The sixth column, '`Equity`', is the estimated cubeless equity following the move — this is the expected number of points per game won by the player on roll.

`gnubg` will 'look ahead' a certain number of moves when evaluating the probabilities, according to the search depth set by the `set plies` command (see Chapter 7 [Analysis], page 19).

`show pipcount`

Use `show pipcount` to automatically count the number of 'pips' each player needs to bear off. Depending on the position, the output will look something like:

```
The pip counts are: X 103, O 112.
```

`show player`
`set player` *name* `human`
`set player` *old-name* `name` *new-name*

Both players have a certain amount of configuration information; use the command `show player` to summarise these settings. By default, `gnubg` will play for player 0, whose name is initially '`O`'. Player 1 defaults to a human (i.e. `gnubg` will prompt the user for a move when it is player 1's turn) whose name is '`X`'.

Either player can be set to a human with the command `set player` *name* `human`, where *name* is either the number of the player (0 or 1) or that player's name (initially '`O`' and '`X`'). You can also specify `both` which will set both players simultaneously. There are also options for computer players (see Section 6.3 [Computer Player], page 17).

You can change the names of the players with the `set player` *old-name* `name` *new-name* command. Again, either the player numbers or names are valid for the *old-name* parameter. Names may not contain whitespace characters, and may not be longer than 31 characters. The names '`0`', '`1`' and '`both`' are not permitted, to avoid ambiguities, and the players may not both share the same name. Names are not case sensitive.

---

[1] Note that the probability of winning includes the probability of winning a gammon or backgammon, and likewise the gammon probabilities include the backgammon probabilities.

`set turn` *player*
`show turn` FIXME

`set autobearoff`
`set automove`
FIXME

`set display`
FIXME

6.2 Controlling Dice Rolls

`set rng` *generator* `[`*seed*`]`
`show rng`
`roll`
`set autoroll`
`set dice` *pips pips*
FIXME

6.3 Having `gnubg` Make Moves

`play`
`set player` *name* `gnubg`
`set player` *name* `pubeval`
`set player` *name* `plies` *depth*
FIXME

6.4 Matches and Sessions

This section describes how to use GNU Backgammon to play series of games, whether those games are part of a match (as in tournament backgammon) or a session of independent games (conventionally called "money" play, regardless of whether any money is involved).

`new match` *length*
`new session`
`set autogame`
`set crawford`
`set postcrawford`
`set autocrawford`
`set jacoby`
`set autodoubles` *limit*
`set score` *points points*
`show crawford`
`show postcrawford`
`show jacoby`
`show score`
FIXME

6.5 The Doubling Cube

`double`
`redouble`
`beaver`
`take`
`accept`
`drop`
`reject`
`set cube centre`
`set cube owner` *player*
`set cube value` *points*

FIXME

6.6 Saving Games and Matches

`save game` *file*
`save match` *file*
`load game` *file*
`load match` *file*

FIXME

7 Using gnubg to Analyse Positions

`eval` [*id*]
`rollout` [*id*]
`set plies` *depth*
`set cache` *size*
`show cache`
FIXME

8 Position Databases

`database dump`
`database evaluate`
`database generate`
FIXME

`database train`
`train database`
FIXME pointer only

9 Modifying `gnubg`'s Neural Nets

`save weights` *file*
`load weights` *file*
`database train`
`train database`
FIXME

10 Frequently Asked Questions

10.1 Where can I get GNU Backgammon?

Pre-release snapshots of GNU Backgammon are periodically made available for FTP at `ftp://alpha.gnu.org/gnu/gnubg/`.

If you want to experiment with the very latest code, the development sources are kept in a CVS repository at `http://subversions.gnu.org/cgi-bin/cvsweb/gnubg/`; see the `gnubg` web page at `http://www.gnu.org/software/gnubg/gnubg.html` for instructions on checking out sources from the repository.

10.2 How do I play a game?

Once you are running `gnubg`, enter *`new game`* at the '`(gnubg)`' prompt to start a game against the computer opponent. You should now see a board (if an X display is available, `gnubg` will use a graphical board window; otherwise, it will display an ASCII board on your terminal). If `gnubg` won the opening roll, it will have moved; you can now type *`roll`* (or click on the dice below the board if you are using X) to roll the dice yourself. In either case, it will now be your move; you should enter the moves for each chequer as pairs of numbers. For instance, if you have rolled 3 and 1, you could type *`8 5 6 5`* to move one chequer from the 8 point to the 5 point and another from the 6 point to the 5 point. Use *`bar`* to move from the bar, and *`off`* to bear off. If you are using the X board window, you can also drag chequers around the board with mouse button 1. Click on the dice when you have finished.

10.3 I only see '`gnubg.bd: No such file or directory`'. What's wrong?

The file '`gnubg.bd`' is the bearoff database that `gnubg` uses to evaluate endgame positions. `gnubg` will look for it first in the current directory and then in the installed directory ('`/usr/local/share/gnubg/`' by default).

The `gnubg` distribution is set up to create '`gnubg.bd`' by itself during compilation, but it can be a slow process (taking half an hour or more, depending on the speed of your computer). If you would rather not wait to generate the database yourself, you can obtain a copy via FTP from `ftp://alpha.gnu.org/gnu/gnubg/gnubg.bd.gz`.

10.4 Now I get '`gnubg.weights: No such file or directory`'. What's that?

The file '`gnubg.weights`' contains the trained neural net weights for most of `gnubg`'s position evaluators. Like '`gnubg.bd`', it should be kept in either the current directory or the installed directory.

You should be able to obtain a copy from wherever you found the `gnubg` distribution. Future releases of `gnubg` will allow you to start without any existing weights at all, if you want to train your own.

10.5 The board window is shown in black and white and looks awful! What's wrong?

If you see this problem, your X server is probably using a *PseudoColor* visual, and is dynamically allocating colours to clients from a limited colourmap. On these types of displays, `gnubg` tries to be a well-behaved client by using colours from the default standard colourmap. This will allow it to share colours with other clients that use the same scheme, which will help them all to use as many colours as possible without exhausting the colourmap.

Unfortunately, not all X clients use standard colourmaps. If other clients have been run before `gnubg` and allocated most of the colours, then there may not be enough left for `gnubg` to allocate a standard colourmap. If this happens, it will just take what it can get (which in the worst case might be black and white only). You can try to avoid this problem by running `gnubg` before other colour-hungry clients, or by using the '`xstdcmap`' utility to install the default standard colourmap early in your session while colours are still available. Depending on your hardware, you may be able to configure your display to allow more colours, or use a *DirectColor* or *TrueColor* visual which should resolve the problem.

Of course, if your X server can only display black and white (not even shades of grey), then there's not much you can do!

Command Index

!

! 12

?

? 13

A

accept 15, 18
agree 15

B

beaver 18

D

database dump 20
database evaluate 20
database generate 20
database train 21
decline 15
double 18
drop 18

E

eval 19
exit 12

H

help 13
hint 15

L

load game 18
load match 18
load weights 21

M

move 14

N

new game 14
new match 17
new session 17

P

play 17

Q

quit 12

R

redouble 18
reject 15, 18
resign 15
roll 17
rollout 19

S

save game 18
save match 18
save weights 21
set autobearoff 17
set autocrawford 17
set autodoubles 17
set autogame 17
set automove 17
set autoroll 17
set board 15
set cache 19
set crawford 17
set cube centre 18
set cube owner 18
set cube value 18
set dice 17
set display 17
set jacoby 17
set player gnubg 17
set player human 16
set player name 16
set player plies 17
set player pubeval 17
set plies 19
set postcrawford 17

`set rng` 17
`set score` 17
`set turn` 16
`show autodoubles` 17
`show board` 15
`show cache` 19
`show copying` 13
`show crawford` 17
`show jacoby` 17
`show pipcount` 16
`show player` 16
`show postcrawford` 17
`show rng` 17
`show score` 17
`show turn` 16
`show warranty` 13

T

`take` 18
`train database` 21

Index

A

analysis of positions 19

B

backgammon 2
backgammon rules 2
backgammons 5
bar 2
basic commands 13
bearing off 4
blot 4
board setup 2

C

cancelling commands 12
command line options 12
commands, basic 13
commands, playing 14
computer player 17
Crawford rule 6
cube owner 5
cube, doubling 18

D

databases 20
dice 17
doubling cube 2
doubling, commands 18
doubling, rules 5

E

enter 4
entering 4
errors starting `gnubg` 22
escapes, shell 12
exiting `gnubg` 12

F

FAQ 22
FIBS 1
frequently asked questions 22

G

game commands 14
games, saving 18
gammons 5
`gnubg`, making moves 17
`gnubg.bd` 22
`gnubg.weights` 22

H

hitting 4
home board 2

I

interrupting `gnubg` 12
introduction 1
invoking `gnubg` 12
irregularities 6

L

leaving `gnubg` 12

M

match rules 6
matches 17
matches, saving 18
moving chequers, commands 22
moving chequers, rules 3

N

neural nets, training 21

O

object of the game 2
obtaining `gnubg` 22
open point 3
optional rules 5
outer board 2

P

pips . . . 3
playing a game . . . 22
playing commands . . . 14
points . . . 2
position analysis . . . 19
position databases . . . 20

R

recording matches . . . 18
rolls, dice . . . 17
rules, optional . . . 5
running `gnubg` . . . 12

S

sample game . . . 8
saving matches . . . 18
sessions . . . 17
shell commands . . . 12
starting `gnubg` . . . 12

T

training . . . 21

Table of Contents

1 Introduction 1

2 How to Play Backgammon 2

- 2.1 Rules of Backgammon 2
  - 2.1.1 Setup 2
  - 2.1.2 Object of the Game 2
  - 2.1.3 Movement of the Checkers 3
  - 2.1.4 Hitting and Entering 4
  - 2.1.5 Bearing Off 4
  - 2.1.6 Doubling 5
  - 2.1.7 Gammons and Backgammons 5
  - 2.1.8 Optional Rules 5
  - 2.1.9 Irregularities 6
- 2.2 Rules for Match Play 6

3 Sample Game 8

4 Starting and Leaving `gnubg` 12

- 4.1 Invoking `gnubg` 12
- 4.2 Leaving `gnubg` 12
- 4.3 Shell Commands 12

5 Basic Commands 13

6 Playing Backgammon with `gnubg` 14

- 6.1 Commands Used During the Game 14
- 6.2 Controlling Dice Rolls 17
- 6.3 Having `gnubg` Make Moves 17
- 6.4 Matches and Sessions 17
- 6.5 The Doubling Cube 18
- 6.6 Saving Games and Matches 18

7 Using `gnubg` to Analyse Positions 19

8 Position Databases 20

9 Modifying `gnubg`'s Neural Nets 21

10 Frequently Asked Questions 22

10.1 Where can I get GNU Backgammon? 22
10.2 How do I play a game? 22
10.3 I only see '`gnubg.bd: No such file or directory`'. What's wrong? 22
10.4 Now I get '`gnubg.weights: No such file or directory`'. What's that? 22
10.5 The board window is shown in black and white and looks awful! What's wrong? 23

Command Index 24

Index 26