



MMbg PC3 – Backgammon für MMBasic

Handbuch · Version 1.10.22 · Stand 19. Oktober 2026

© 2026 Matthias Hollatz · Freie Software unter der GNU GPL v3 oder neuer

Wichtig: Diese Version läuft **nur auf dem Pico Computer 3 (PC3)** mit der Firmware **PicoMitePC3 V7.00.00 ab RC2** (Stand heute; Grafik mit Bildpuffer, Ton und alle Tests auf RC2). Auf anderen PicoMite-Boards und mit älterer Firmware läuft sie nicht. Für den PicoCalc gibt es eine eigene Ausgabe (MMbg PiCa).

Inhalt

1. Was ist MMbg?
2. Voraussetzungen
3. Installation
4. Der Bildschirm und das Terminal
5. Ein Match spielen
6. Undo, Speichern und Fortsetzen
7. Aufzeichnung, Spielanalyse und Bildschirmfotos
8. Engines und Spielstärke
9. Die neuronalen Netze
10. Spielen gegen gnubg über WLAN
11. Lange Testläufe (Long Runs)
12. Optionen
13. Befehlsübersicht
14. Dateien
15. Hinweise und Grenzen
16. Lizenz und Dank
17. Ein Flug durch den Quellcode

1. Was ist MMbg?

MMbg ist ein vollständiges Backgammon-Programm in MMBasic. Es spielt **Matches** bis 1 bis 15 Punkte, mit Doppler, Crawford-Regel und allem, was dazugehört.

- **Spielstärke:** Mit der Engine *big* spielt MMbg etwa auf dem Niveau von GNU Backgammon „Expert“ (0-ply). Gegen gnuBg „World class“ (2-ply) gewinnt es rund **47 % der 5-Punkte-Matches** (577 von 1236 gemessenen Matches).
- **Grafik und Terminal:** Das Brett erscheint farbig über HDMI (640 × 480) mit flüssig animierten Zügen. Gleichzeitig zeigt ein serielles Terminal ein ASCII-Brett mit Zugliste, Bewertungen und Log.
- **Bedienung:** Gespielt wird mit Tastatur oder USB-Maus. Auf Wunsch gibt es Geräusche für Würfeln und Ziehen.
- **Komfort:** Mehrstufiges Undo, automatisches Speichern und Fortsetzen. Matches werden als `.mat`-Datei für die Analyse in gnuBg aufgezeichnet. Positionen lassen sich über **GNUbg IDs** austauschen.
- **Für Tüftler:** Über WLAN kann gnuBg auf dem PC als Gegner dienen. Dazu kommen Testläufe über Nacht, Duplicate-Matches und ausführliche Logs.

2. Voraussetzungen

Was	Anforderung
Rechner	Pico Computer 3 (PC3): RP2350B mit PSRAM, HDMI, USB
Firmware	PicoMitePC3 V7.00.00 ab RC2. Mit <code>?MM.VER</code> prüfen: mindestens 7.000052
Bildschirm	HDMI, MODE 3 (640 × 480, 16 Farben)
Tastatur	USB-Tastatur am PC3 und/oder serielles Terminal
Terminal (empfohlen)	VT100-fähig (z. B. Tera Term, PuTTY), 80 × 40 Zeichen , Farben an
Maus (optional)	USB-Maus
Ton (optional)	<code>OPTION AUDIO</code> eingerichtet (beim PC3 I²S)
WLAN (optional)	Nur für gnuBg als Gegner (Kapitel 10)
SD-Karte (optional)	Laufwerk B: für Programm, Logs und <code>.mat</code> -Dateien

Das große Netz braucht rund 600 KB Arbeitsspeicher, die Animation weitere 150 KB für einen Bildpuffer. Der PC3 hat dafür genug PSRAM, deshalb läuft diese Ausgabe nur dort.

3. Installation

3.1 Dateien

Alle Dateien gehören **in denselben Ordner** wie das Programm, entweder auf das Flash-Laufwerk **A:** oder in einen Ordner auf der SD-Karte **B:** (Kapitel 3.3).

Datei	Größe	Zweck	Fehlt sie ...
mmbg_pc3.bas	ca. 190 KB	das Programm	–
bgnet345.dat	ca. 60 KB	kleine Netze 3, 4, 5 (gnubg)	nur Engines 0–2
bgnet02.dat	Text	Ausgabe-Gewichte der großen Netze	keine Engines 4 und 5
bgnet0.bin, bgnet2.bin	je 256 000 Byte	große Netze Kontakt und crashed	keine Engines 4 und 5
bgesc.bin	klein	Escape-Tabellen (gnubg)	keine Engines 4 und 5
bgbo15.dat	ca. 1,6 MB	Bear-off-Datenbank bis 15 Steine	Rennen nur mit Netz 5

Die Einstellungs- und Spielstandsdateien (`mmbg_pc3.ini`, `bgext.ini`, `mmbg_pc3.sav`) legt MMbg selbst im Programmordner an.

3.2 Übertragen und starten

Schritt 1: Jede Datei mit `XMODEM RECEIVE "name"` vom PC auf den PC3 übertragen oder über die SD-Karte kopieren.

Schritt 2: Das Programm laden und starten:

```
LOAD "mmbg_pc3.bas", C
RUN
```

Mit `, C` lässt MMBasic die Kommentare weg. Die veröffentlichte Fassung hat nur wenige Kommentare und lädt auch ohne `, C`.

Schritt 3: Beim Start erscheinen Logo, Lizenz und die Ladezeilen für Netze, Datenbank und WLAN. Danach wartet MMbg im Startbild auf **N** (neues Match).

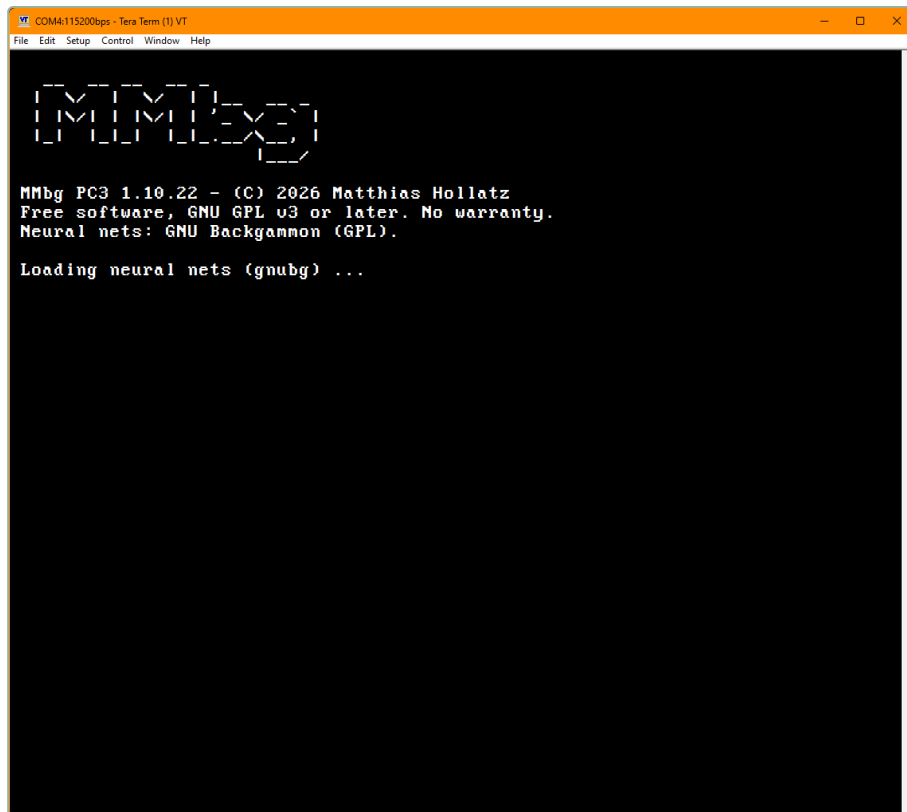


Bild 1: Start im Terminal – Logo, Lizenz, die Netze werden geladen.

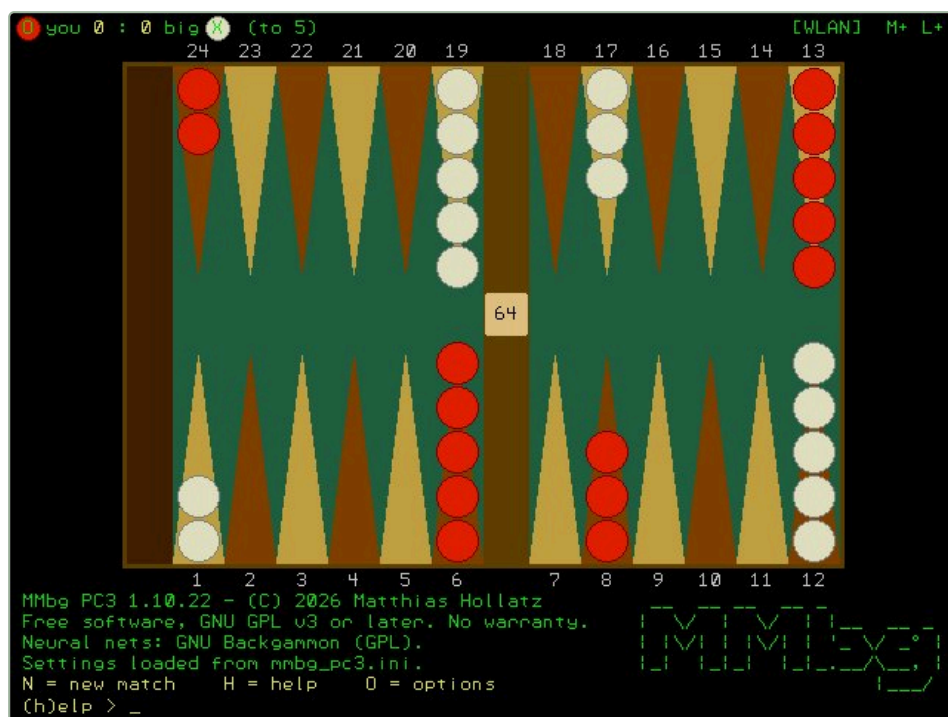


Bild 2: Startbild auf dem HDMI-Bildschirm. Unten links Version, Lizenz und Meldungen, rechts das Logo, darunter die wichtigsten Tasten.

Fehlermeldungen mit Zeilennummer beziehen sich beim Laden mit `, C` auf das Programm ohne Kommentarzeilen.

3.3 Alles in einem Ordner auf der SD-Karte

Programm, Netze und Datenbank können auch zusammen in einem Ordner auf **B:** liegen. Das Flash-Laufwerk A: bleibt dann frei für anderes. Empfohlen:

Ordner	Inhalt
B:/mmbg	Programm, Netze, Datenbank, ini-Dateien, gespeichertes Match
B:/mmbg/log	Logs, .mat -Dateien und Bildschirmfotos

Einrichten (einmalig):

```
MKDIR "B:/mmbg"  
MKDIR "B:/mmbg/log"
```

Dann alle Dateien aus Kapitel 3.1 nach B:/mmbg kopieren. Gestartet wird aus diesem Ordner, damit MMbg seine Dateien dort findet und die ini-Dateien dort anlegt:

```
CHDIR "B:/mmbg"  
LOAD "mmbg_pc3.bas", C  
RUN
```

Zum Schluss den Log-Ordner einstellen: **O** → **7**, dann **L** und B:/mmbg/log eingeben, mit **S** speichern. Der Ordner für Logs, .mat und Bildschirmfotos ist also unabhängig vom Programmordner und kann auch ganz woanders liegen. Er muss vorher angelegt sein.

4. Der Bildschirm und das Terminal

4.1 HDMI-Bildschirm

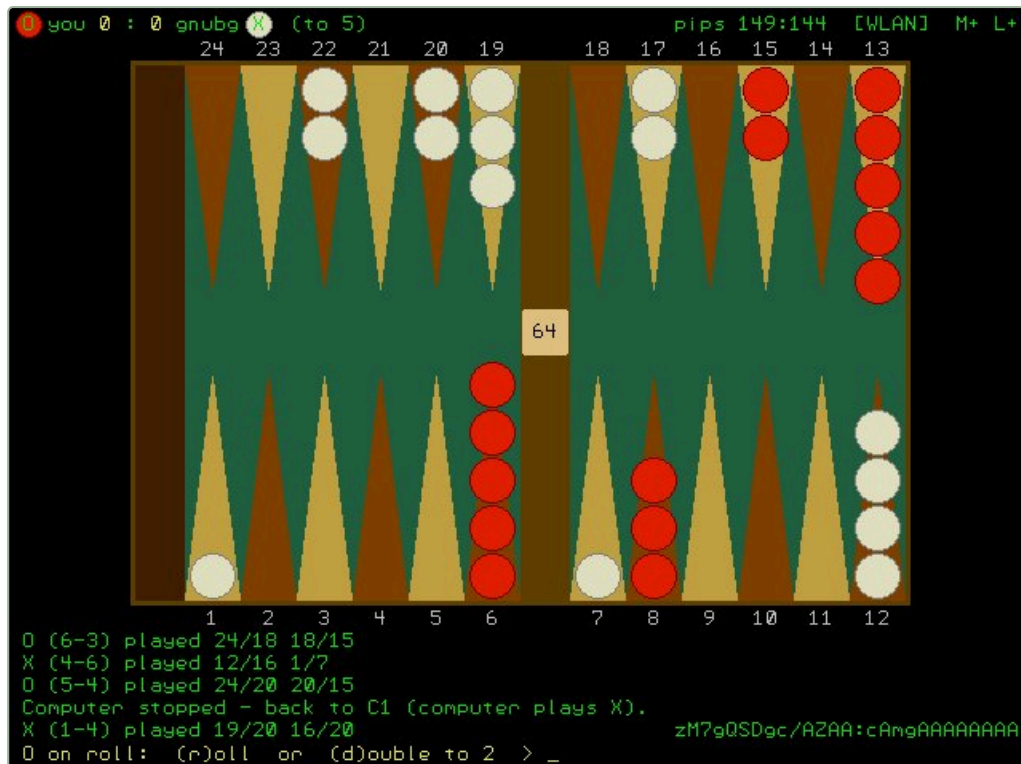


Bild 3: Laufendes Spiel. Titel mit Spielern, Stand, Pips und Schaltern, darunter das Brett mit dem Doppler in der Bar, das Log mit fünf Zeilen, rechts unten die GNUbg ID und ganz unten der Prompt (hier die Frage vor dem Wurf).

- **Titelzeile**
- O und X erscheinen als kleine Steine in der Spielerfarbe, daneben der Spieler: `you` für einen Menschen, sonst der Name der Engine.
- Der **Spielstand ist gelb**, dahinter steht die Matchlänge, z. B. `(to 5)`.
- `M 4:2` zählt die gewonnenen Matches der Sitzung. Er erscheint bei Computerläufen und sobald ein Match zu Ende ist.
- In der Mitte steht Crawford oder Post-Crawford.
- Rechts folgen die Pips (Option), `[WLAN]` und `M±` für die `.mat`-Aufzeichnung sowie `L±` für das Log.
- **Brett:** 24 Punkte, Bar, Ablage. Die Würfel liegen immer **rechts vom Spieler** aus gesehen, also O-Würfel rechts und X-Würfel links.
- Der Doppler steht in der Mitte der Bar oder beim Besitzer. Im Crawford-Spiel ist er grau mit „64“; Doppeln ist dann verboten.
- Ein **angebotenes Doppel** erscheint als einzelner Doppler im Spielfeld auf der Seite dessen, der entscheiden muss.
- **Log:** Die letzten fünf Meldungen, Züge und Hinweise stehen über dem Prompt.
- **GNUbg ID:** Die ID der aktuellen Stellung steht grün rechts in der untersten Log-Zeile (Kapitel 7.4).
- **Farben:** Text ist grün, Tasten, Eingaben und veränderliche Werte sind gelb.

4.2 Terminal

Das Terminal (80 × 40, VT100) zeigt dasselbe Spiel als Text:

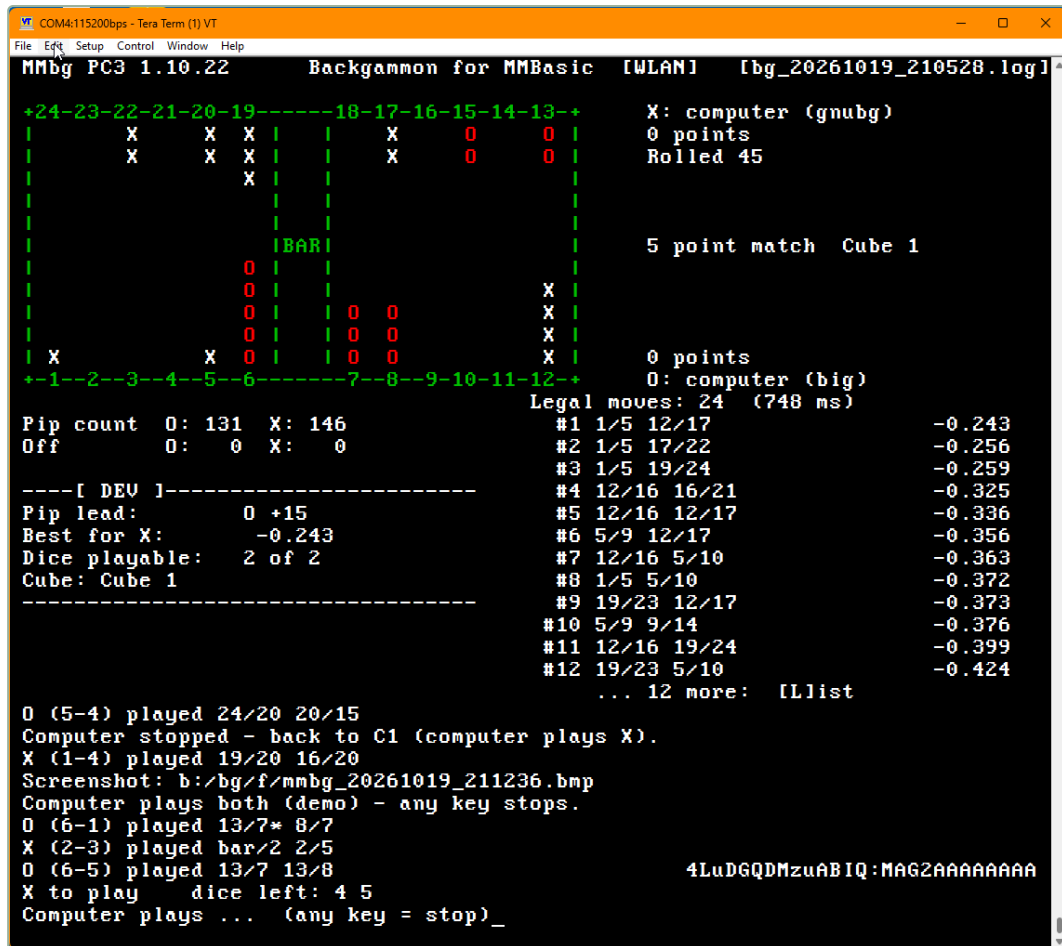


Bild 4: Das Terminal während eines Computerlaufs (big gegen gnuhg): ASCII-Brett, rechts Spieler und Würfel, darunter Pips, DEV-Spalte, Zugliste mit Bewertungen und das Log mit der GNUhg ID.

- **ASCII-Brett:** Der Rahmen ist grün, O-Steine sind weiß, X-Steine rot. Die Option *swap* tauscht die Farben, die Option *flip* spiegelt das Brett wie in der Grafik.
- **Rechts neben dem Brett:** Spieler, Punkte, Matchlänge, Doppler und die gewürfelten Augen.
- **Darunter:** Pips, abgetragene Steine und die Bar. Die **DEV-Spalte** (Taste P) zeigt Pip-Vorsprung, den besten Wert, die spielbaren Würfel und den Doppler.
- **Zugliste rechts:** die 12 besten Züge mit Bewertung (Equity), den Rest mit L. Vor dem Wurf erscheint stattdessen die Doppler-Analyse: Gewinnchancen im Match bei No double, Double/take und Double/pass sowie die Empfehlung des Computers.
- **Unten:** die letzten **8 Log-Zeilen** und die **GNUbg ID** rechts. Von dort lässt sie sich bequem mit der Maus kopieren.

5. Ein Match spielen

5.1 Start

- **N** startet ein neues Match mit der eingestellten Länge.
- **M7** startet ein Match bis 7 Punkte. Möglich sind 1 bis 15 Punkte.
- **C1** (Standard): Der Mensch spielt O (unten), der Computer spielt X (oben).
- Der Eröffnungswurf bestimmt, wer beginnt.

5.2 Ziehen

Eingabe	Bedeutung
13/7 8/7	Züge in gnuhg-Schreibweise, mehrere Teilzüge durch Leerzeichen getrennt
bar/22, 6/off	von der Bar, abtragen (kurz: b, o; auch - statt /)
13 24	nur die Startpunkte: der große Würfel zuerst
b 6	von der Bar und von der 6
#3	Zug Nr. 3 der Zugliste
E	Zug beenden, falls kein Würfel mehr spielbar ist
U	Undo (Kapitel 6)

Ein Teilzug ist erst gültig, wenn danach noch die größtmögliche Zahl Würfel spielbar bleibt. Kann ein Spieler gar nicht ziehen, passt MMbg automatisch und zeigt die Würfel kurz an.

Mit der Maus:

Aktion	Bedeutung
Links auf einen eigenen Stein (oder die Bar)	mit dem großen Würfel ziehen. Geht das nicht, mit dem kleinen
Rechts auf einen eigenen Stein	mit dem kleinen Würfel ziehen
Klick auf die eigenen Würfel	Undo
Vor dem Wurf: Klick auf den Doppler	doppeln
Vor dem Wurf: Klick aufs Brett	würfeln
Klick auf den angebotenen Doppler	annehmen. Aufgeben nur mit der Taste P

5.3 Doppler

Vor jedem Wurf fragt MMbg den Spieler am Wurf, sofern er doppelnd darf: (r)oll or (d)ouble. Bietet der Gegner ein Doppel an, lautet die Antwort (t)ake oder (p)ass.

- Mit der Option *hint* zeigt das Log die Doppler-Empfehlung und die Match-Gewinnchancen.
- Im **Crawford-Spiel** ist Doppeln verboten. Danach folgt Post-Crawford.
- Gespielt wird nach Match-Regeln. Geldspiel-Regeln wie Jacoby oder Beaver gibt es nicht.

5.4 Spiel- und Matchende

Gewinnt ein Spieler, zählen ein Punkt (einfach), zwei (Gammon) oder drei (Backgammon), jeweils mal Doppler. Das nächste Spiel beginnt nach drei Sekunden, ein neues Match mit ENTER.

6. Undo, Speichern und Fortsetzen

6.1 Undo

- **U** im laufenden Zug nimmt die eigenen Teilzüge zurück.
- **U** ohne gezogenen Stein springt zum **letzten eigenen Zuganfang** zurück, mit denselben Würfeln. Das geht auch, wenn der Computer schon gezogen hat, zum Beispiel nach einem Verklicken.
- **Weiteres U** geht noch einen eigenen Zug zurück und so weiter, bis zum Anfang des Spiels.
- Nach einem Undo würfelt der Computer neu, damit niemand auf dessen Würfel reagieren kann. Undo wirkt nur innerhalb des laufenden Spiels.

Grundlage ist der **Zugverlauf**: Zu jedem Zuganfang merkt sich MMbg die GNUbg ID, also Brett, Stand, Doppler und Würfel.

6.2 Speichern und Fortsetzen

- MMbg speichert das laufende Match **automatisch** in `mmbg_pc3.sav`:
- bei jedem Zuganfang eines Menschen,
- an den Fragen roll/double und take/pass,
- bei Computerläufen C2 und C3 in jedem Zug. C4 speichert nicht.
- **SG** speichert sofort, **LG** lädt das gespeicherte Match.
- Beim **Start** fragt MMbg: „*Saved match found – continue it? (y/n)*“.
- Gespeichert werden der ganze Zugverlauf (Undo geht nach dem Laden weiter), Computer-Modus, Engines und die laufende `.mat`-Aufzeichnung.
- Am Matchende wird die Datei gelöscht.

Was in `mmbg_pc3.sav` steht: Die Datei ist reiner Text, eine Angabe je Zeile. Das Herzstück sind die `turn=-`-Zeilen: Für **jeden Zuganfang** des Spiels steht dort die **GNUbg ID** der Stellung (Brett, Stand, Doppler, Würfel), dazu die Zugnummer und wie viele `.mat`-Einträge bis dahin geschrieben waren. Aus der letzten ID baut MMbg beim Laden die Stellung wieder auf; die früheren IDs machen Undo nach dem Laden möglich.

```

' MMbg PC3 saved match
version=1.10.22
date=19-10-2026 20:26:53
computer=1                Computer-Modus (C1)
engine_o=0                Engines 0 und X
engine_x=4
game=7                    Spielnummer der Sitzung
turn=4HPwATDgc/ABMA:cAmgAAAAAAAA 0 0    Zuganfang: GNUbg ID, Zug, .mat-Stand
turn=4HPwATDgc/ABMA:QNrABAAGAAA 1 1
...
matfile=B:/mmbg/log/mmbg_20261019_202011.mat
mgame=3                  Spielnummer im Match (.mat)
act=2|52: 13/8 24/22     bisher aufgezeichnete Aktionen des Spiels
now=4HPwATDgc/ABMA:cAmgAAAAAAAA         Stellung vor dem Wurf (an roll/double)

```

Jede ID lässt sich auch einzeln in gnuBG einfügen, um eine Stellung aus einem gespeicherten Match anzusehen. Die Erklärungen rechts stehen nur hier im Handbuch, nicht in der Datei.

7. Aufzeichnung, Spielanalyse und Bildschirmfotos

7.1 Matches als `.mat` für gnuBG

Mit der Option **mat** (M+ im Titel) schreibt MMbg jedes Match in eine eigene Datei:

```
[Log-Ordner]mmbg_JJJJMMTT_HHMMSS.mat
```

- Das Format ist dasselbe, das gnuBG mit `export match mat` schreibt.
- Der Kopf wird beim Matchstart geschrieben, jedes Spiel am Spielende angehängt.
- Neue Matches überschreiben alte Dateien nie.
- Was mit Undo zurückgenommen wurde, erscheint nicht in der Datei.
- Ein fortgesetztes Match schreibt in seine alte Datei weiter.
- X steht in gnuBG als Spieler 0 (oben), O als Spieler 1 (unten).

Analyse in gnuBG:

1. Die Datei zum PC übertragen, z. B. per XMODEM oder SD-Karte. XMODEM füllt das Dateiende mit Nullbytes auf, das stört gnuBG nicht.
2. In gnuBG: *Datei* → *Importieren* → `.mat` oder `import mat datei.mat`.
3. *Analyse* → *Match analysieren* oder `analyse match`, dann die Statistik mit `show statistics match`.

7.2 Log-Datei

Mit **G** an oder aus (L± im Titel). Die Datei `bg_JJJJMMTT_HHMMSS.log` im Log-Ordner enthält:

- jeden Zug mit Zeit, Würfeln, Platz in der Zugliste und Bewertung,
- die Rechenzeit, die Pips und die Antwortzeit von gnuBG,
- Doppler-Entscheidungen mit Gewinnchancen sowie Spiel- und Matchergebnisse.

Ab etwa 1 MB beginnt eine neue Datei. Mit **GN** beginnt sofort eine neue.

Eine Zugzeile sieht so aus (im Log steht sie in einer Zeile):

```
2105.3s G12 T 14 X 6-4 big 24/18 13/9 #1/18 +0.123 best +0.123
gen 512ms pip 0 140 X 150 ext 70ms
```

Das heißt: Sekunden seit dem Start, Spiel 12, Zug 14, X hat 6-4 gewürfelt, Engine big, der gespielte Zug, sein **Platz in der Zugliste** (1 von 18 legalen Zügen), seine Bewertung und die beste Bewertung, die Rechenzeit des Zuggenerators, die Pips beider Seiten und bei gnuBG die Antwortzeit. Dazu kommen Zeilen mit `---` (Doppler, Engines, Einstellungen, Undo), `***` (Spielende), `###` (Matchende und Gesamtstand) und `!!!` (Fehler).

7.3 .mat oder Log - was wofür?

	.mat -Aufzeichnung	Log-Datei
Zweck	das Match nachspielen und analysieren	die Engines beobachten, vor allem bei Testläufen
Format	Standardformat, liest gnuBG (und andere Programme)	eigenes Textformat für Mensch und Auswerteskript
Inhalt	nur das Spiel: Würfe, Züge, Doppler, Ergebnisse	zusätzlich Platz in der Zugliste, Bewertungen, Rechenzeiten, Pips, gnuBG-Antwortzeiten, Fehler, Einstellungen
Umfang	eine Datei je Match, klein	eine laufende Datei, ab ca. 1 MB eine neue
typisch	eigene Partien gegen den Computer	Long Runs über Nacht (Kapitel 11)

Beides kann gleichzeitig an sein. Wer nur spielt und seine Partien in gnuBG analysieren will, braucht nur `.mat`.

7.4 GNUBG ID

Die ID beschreibt eine Stellung vollständig: Brett, Matchlänge, Stand, Doppler, Crawford, Spieler am Wurf und Würfel. Ein Beispiel:

```
4HPwATDgc/ABMA:cAmgAAAAAAAA
```

- **MMbg → gnuBG:** Die ID steht rechts unten. Im Terminal kopieren und in gnuBG in das Feld *GNUBG ID* einfügen oder `set gnuBGid ...` eingeben. So lässt sich jede Stellung sofort in gnuBG untersuchen.
- **gnuBG → MMbg:** Eine ID einfach am Prompt einfügen, auch an `roll/double` oder `take/pass`. MMbg fragt „Take over the position...? (y/n)“ und baut die Stellung auf. Ein Geldspiel, ein Match über 15 Punkte und ein Doppler über 64 werden abgelehnt.
- Die IDs sind mit gnuBG 1.08.003 gegengeprüft. Ein angebotenes Doppel enthält die ID nicht, weil gnuBG es beim Einlesen verwirft.

7.5 Tipp-Option

Mit **hint** (Option 3) schreibt MMbg den besten Zug ins Log, wenn ein Mensch am Zug ist, und an den Doppler-Fragen die Empfehlung. Das eignet sich gut zum Lernen.

7.6 Bildschirmfoto (Strg+P)

An jeder Eingabe speichert **Strg+P** den ganzen HDMI-Bildschirm als BMP-Datei (24 Bit, ca. 900 KB) in den Log-Ordner:

```
[Log-Ordner]mmbg_JJJJMMTT_HHMMSS.bmp
```

Das Log meldet den Dateinamen. Die Bilder in diesem Handbuch sind so entstanden. (Der Bildbetrachter im Dateimanager des PC3 zeigt die Farben falsch an, am PC sind sie richtig.)

8. Engines und Spielstärke

8.1 Die Engines

Nr.	Name	Arbeitsweise	Stärke (grob)
0	base	Bewertungsformel: Pips, Blots, Punkte, Bar	Anfänger
1	book	base plus Eröffnungsbuch	Anfänger+
2	tuned	im Labor abgestimmte Gewichte	Gelegenheitsspieler
3	net	kleine gnubg-Netze (0-ply)	guter Klubspieler
4	big	große gnubg-Netze, Vorfilter mit den kleinen Netzen	≈ gnubg „Expert“ (0-ply)
5	deep	big mit 1 Zug Vorausschau (1-ply)	stärker als big, ca. 18-mal langsamer
E	gnubg	gnubg auf dem PC über WLAN	so stark wie gnubg eingestellt ist

Gewählt wird mit **A45**: Engine für O ist 4, für X ist 5. Für jede Seite gibt es ein Zeichen aus 0 bis 5 oder E.

Messwerte für big:

Messung	Ergebnis
Matches gegen gnubg 2-ply (bis 5)	577 von 1236 = 46,7 %
Fehler je Zug, gemessen an gnubg 2-ply	big 2,6 mEq, gnubg 0-ply 2,4 mEq
bester Zug getroffen	big 82 %, gnubg 0-ply 83 %

- Rund 90 % des Rückstands auf gnubg 2-ply kommen aus der Zugwahl, rund 10 % aus dem Doppler.
- Die Rechenzeit liegt im Mittel bei etwa 0,5 s je Zug.

8.2 Doppler-Modell

Für den Doppler rechnet MMbg wie gnubg mit 0-ply: Aus den Gewinn-, Gammon- und Backgammon-Chancen des Netzes entstehen über die **Match-Gewinntabelle Kazaross-XG2** die Chancen für No double, Double/take und Double/pass.

8.3 Computer-Modi

Befehl	Modus
C0	zwei Menschen
C1	Computer spielt X (Standard)
C2	Computer spielt beide Seiten (zum Zuschauen, mit Pausen)
C3	beide, ohne Pausen und ohne Animation
C4	Turbo: beide, ohne Bildschirm, endlos Matches (Testläufe)

In C2 bis C4 stoppt **jede Taste** den Computer.

9. Die neuronalen Netze

9.1 Wie arbeitet so ein Netz? (ganz einfach erklärt)

Ein neuronales Netz ist im Kern eine **große Rechenvorschrift mit vielen festen Zahlen**. Es bekommt eine Stellung und gibt zurück, wie gut sie ist. Mehr nicht – es denkt nicht voraus und kennt keine Regeln.

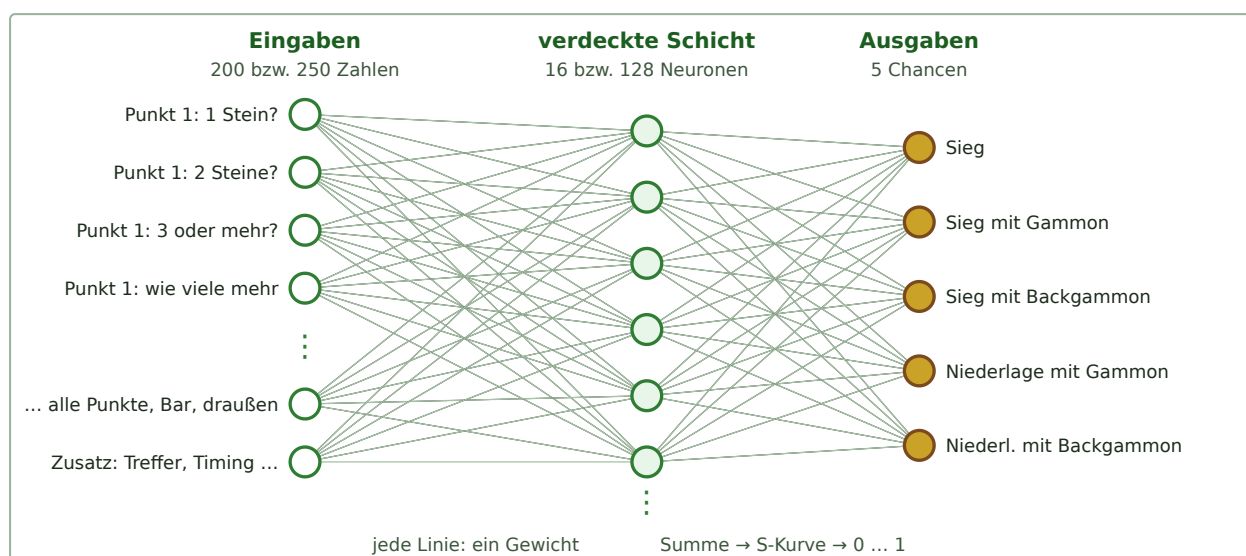


Bild 11: Aufbau eines gnubg-Netzes (verkleinert dargestellt). Links die Stellung als Zahlen, in der Mitte die verdeckte Schicht, rechts die fünf Chancen.

1. Die Stellung wird zu Zahlen. Für jeden Punkt und jede Seite gibt es vier Eingaben: Liegt dort genau 1 Stein (ein Blot)? Genau 2? 3 oder mehr? Und wie viele darüber hinaus? Dazu kommen Bar und abgetragene Steine. So entstehen 200 Zahlen, meist 0 oder 1. Das große Netz bekommt noch 50 Zusatzmerkmale, die ein Mensch „sehen“ würde: Wie oft kann ich getroffen werden? Wie leicht kommen hintere Steine heraus? Wie viel Zeit bleibt mir?

2. Die verdeckte Schicht. Jedes Neuron in der Mitte bildet eine **gewichtete Summe** aller Eingaben: jede Eingabe mal ihr eigenes Gewicht, alles zusammengezählt. Die Summe läuft durch eine **S-Kurve**, die sie auf einen Wert zwischen 0 und 1 drückt. Jedes Neuron wird so zu einem kleinen „Detektor“ für ein Muster, etwa „gutes Heimfeld“ oder „gefährdeter Blot“ – welches Muster, hat das Netz selbst gelernt.

3. Die Ausgaben. Die fünf Ausgabe-Neuronen machen dasselbe noch einmal mit den Werten der Mitte. Heraus kommen fünf Chancen zwischen 0 und 1: Sieg, Sieg mit Gammon, Sieg mit Backgammon, Niederlage mit Gammon, Niederlage mit Backgammon. Daraus wird ein einziger Wert, die **Equity**:

$$\text{Equity} = 2 \times \text{Sieg} - 1 + \text{Gammon-Siege} + \text{Backgammon-Siege} - \text{Gammon-Niederlagen} - \text{Backgammon-Niederlagen}$$

4. Woher kommen die Gewichte? Sie sind das eigentliche Wissen. Die gnubg-Autoren haben sie mit sehr vielen ausgespielten Stellungen trainiert: Das Netz schätzt, vergleicht mit dem tatsächlichen Ergebnis und verschiebt die Gewichte ein klein wenig in die richtige Richtung – millionenfach. MMbg rechnet nur mit den fertigen Zahlen, es lernt nicht selbst.

5. Wie wird daraus ein Zug? Der Zuggenerator findet alle legalen Züge (oft 20 bis 50, bei Paschen auch einige hundert). Für jeden Zug wird die **Stellung danach** durch das Netz geschickt. Gespielt wird der Zug mit der besten Equity. Engine *big* bewertet erst alle Züge mit dem kleinen Netz und rechnet nur die besten 12 mit dem großen nach; *deep* schaut zusätzlich auf alle 21 möglichen Würfe des Gegners voraus.

Das große Netz braucht für eine Stellung rund 32 000 Multiplikationen ($250 \times 128 + 128 \times 5$). In MMBasic dauert das auf dem PC3 etwa 19 ms.

9.2 Die Netze in MMbg

MMbg verwendet die **Netzugewichte von GNU Backgammon** (gnubg 1.08, GPL). MMbg rechnet die Netze selbst in MMBasic. Die Eingabekodierung ist dieselbe wie in gnubg.

Netz	Aufbau	Einsatz
3 Kontakt (klein)	200 Eingaben, 16 verdeckte Neuronen, 5 Ausgaben	Engine net, Vorfilter für big
4 crashed (klein)	200-16-5	Stellungen mit zusammengebrochenem Heimfeld
5 Rennen (klein)	200-8-5	Rennen, wenn kein Kontakt mehr besteht
0 Kontakt (groß)	250 Eingaben (200 + 50 Zusatzmerkmale), 128 Neuronen, 5 Ausgaben	Engine big und deep
2 crashed (groß)	250-128-5	Engine big und deep

- **Die 5 Ausgaben** sind: Sieg, Sieg mit Gammon, Sieg mit Backgammon, Niederlage mit Gammon, Niederlage mit Backgammon.
- **Die 50 Zusatzmerkmale** des großen Netzes, etwa Treffer-Chancen, Escapes, Containment und Timing, berechnet MMbg nach dem Vorbild von gnubg.
- **Bear-off-Datenbank bgbo15.dat** : Sie ist selbst berechnet und enthält keine gnubg-Daten. Sie liefert exakte Werte im Endspiel: eine Seite, 6 Punkte, bis 15 Steine, 54 264 Stellungen, mit Gammon-Anteilen.

Welche Netze welche Engine braucht:

Engine	benötigt
0 bis 2	nichts, nur das Programm
3 net	bgnet345.dat
4 big, 5 deep	zusätzlich bgnet02.dat, bgnet0.bin, bgnet2.bin, bgesc.bin und genug Speicher (PSRAM)
alle	bgbo15.dat macht das Endspiel exakt

10. Spielen gegen gnuhg über WLAN

Der PC3 kann **GNU Backgammon auf einem PC** im Heimnetz als Gegner nutzen. MMhg stellt dabei die Fragen, gnuhg rechnet.

10.1 Einrichten

Schritt 1 - WLAN am PC3 einmalig einrichten und neu starten:

```
OPTION WIFI "MeinNetz", "Passwort"
```

Schritt 2 - gnuhg auf dem PC starten (z. B. `gnuhg-cli`) und eingeben:

```
external :41000
```

gnuhg wartet dann auf Port 41000. Die Spielstärke wird in gnuhg selbst eingestellt, z. B. 0-ply oder 2-ply.

```

GNU Backgammon 1.08.003-mingw 32-Bit 20240428

Copyright (C) 1999-2004 Gary Wong.
Copyright (C) 2004-2024 the AUTHORS; for details type 'show version'.

This program comes with ABSOLUTELY NO WARRANTY; for details type 'show warranty'.
This is free software, and you are welcome to redistribute it under certain conditions; type 'show copying' for details.

'eval' and 'hint' chequerplay will use 0 ply evaluation.
'eval' and 'hint' cube decisions will use 0 ply evaluation.
'eval' and 'hint' will use:
  Chequer play:
    Neural net evaluation:
      0-ply evaluation.
      Using pruning neural nets.      Cubeful evaluations.
      Noiseless evaluations.
  Move filters:
    Move filter for 1 ply:
      keep the first 0 0-ply moves and up to 8 more moves within equity 0,16
    Move filter for 2 ply:
      keep the first 0 0-ply moves and up to 8 more moves within equity 0,16
      Skip pruning for 1-ply moves.
    Move filter for 3 ply:
      keep the first 0 0-ply moves and up to 8 more moves within equity 0,16
      Skip pruning for 1-ply moves.
      keep the first 0 2-ply moves and up to 2 more moves within equity 0,04
    Move filter for 4 ply:
      keep the first 0 0-ply moves and up to 8 more moves within equity 0,16
      Skip pruning for 1-ply moves.
      keep the first 0 2-ply moves and up to 2 more moves within equity 0,04

```

Bild 5: gnuhg 1.08.003 auf dem PC (Kommandozeile), eingestellt auf 0-ply. Danach folgt `external :41000`.

Schritt 3 - in MMhg die Optionsseite öffnen (**O** → **7**) und einstellen:

- **H** host: die IP-Adresse des PCs
- **P** port: z. B. 41000
- **W** timeout in Sekunden
- **F** fallback: die Engine, die einspringt, wenn gnuhg nicht antwortet (z. B. 4 big)

- **T** testet die Verbindung (Version und 10 Pings)
- **S** speichert alles in die ini-Dateien

Steht **[WLAN]** im Titel, ist das WLAN da.

```
MMbg PC3 1.10.22 - OPTIONS (C) 2026 Matthias Hollat
key + ENTER = change ENTER alone = back to the game
-----
BOARD 1 animation + 4 flip board + home board left
      2 pips in title + 5 swap colours + 0 red, X light
      3 hint - 8 mouse + USB mouse found
      0 sound - dice, moves, hits (PLAY BBC SOUND)
-----
RECORD 9 match file .mat + G log file +
      L folder b:/bg/f/ for log and .mat files, must exist
      now: log bg_20261019_210528.log match -
-----
GNUBG external engine E = gnuBG on a PC over WLAN (TCP)
on the PC start gnuBG and type: external :41000
H host 192.168.1.132 P port 41000
W timeout 30 s F fallback 4 big
D debug log - C check gnuBG's moves + (slower)
V level (text for the log) 2-ply
T test connection (version + 10 x ping)
-----
TEST RUNS U duplicate - each match again, same dice, sides swapped
          R dice seed 0 = from the clock (now 1717527)
-----
STATUS WLAN: connected, IP 192.168.1.150 gnuBG at 192.168.1.132:41000
        S save all settings (mmbg_pc3.ini + bgext.ini)
-----

Connected in 7 ms. gnuBG 1.08.003-mingw 32-Bit Answer: Interface: 2!RFBF:
0!Engine: 1.01!Software: 1.08.003-mingw 32-Bit Ping 10x: 10 ok, avg 29 ms,
max 102 ms

gnuBG level for the log (e.g. 0-ply, 2-ply) > 0-ply_
```

Bild 6: Verbindungstest mit **T** geglückt: gnuBG-Version, 10 Pings mit Antwortzeiten. Danach fragt MMbg nach dem Text für das Log (**V level**).

```
MMbg PC3 1.10.22 - OPTIONS (C) 2026 Matthias Hollat
key + ENTER = change ENTER alone = back to the game
-----
BOARD 1 animation + 4 flip board + home board left
      2 pips in title + 5 swap colours + 0 red, X light
      3 hint - 8 mouse + USB mouse found
      0 sound - dice, moves, hits (PLAY BBC SOUND)
-----
RECORD 9 match file .mat + G log file +
      L folder b:/bg/f/ for log and .mat files, must exist
      now: log bg_20261019_210528.log match -
-----
GNUBG external engine E = gnuBG on a PC over WLAN (TCP)
on the PC start gnuBG and type: external :41000
H host 192.168.1.132 P port 41000
W timeout 30 s F fallback 4 big
D debug log - C check gnuBG's moves + (slower)
V level (text for the log) 2-ply
T test connection (version + 10 x ping)
-----
TEST RUNS U duplicate - each match again, same dice, sides swapped
          R dice seed 0 = from the clock (now 1717527)
-----
STATUS WLAN: connected, IP 192.168.1.150 gnuBG at 192.168.1.132:41000
        S save all settings (mmbg_pc3.ini + bgext.ini)
-----

Connect failed: connect 192.168.1.132:41000 failed: No response from
client Is gnuBG waiting? (gnuBG: external :41000)

connect 192.168.1.132:41000 failed: No response from client

option > _
```

Bild 7: Verbindungstest gescheitert: gnuBG läuft nicht oder wartet noch nicht (**external :41000** vergessen), oder Host/Port stimmen nicht.

10.2 Spielen

Befehl	Wirkung
A4E	O = big, X = gnuhg
C1	der Mensch spielt O, gnuhg spielt X: ein Spiel gegen gnuhg
C2 oder C3	big gegen gnuhg zum Zuschauen
C4E	Engines O 4, X gnuhg und Turbo

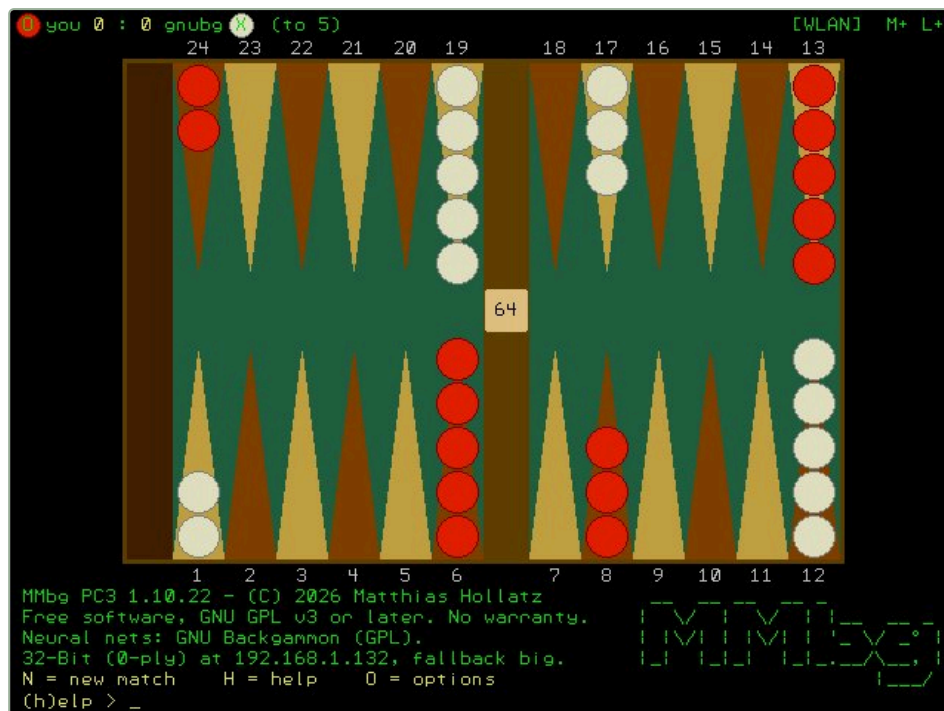


Bild 8: X ist gnuhg. Das Log meldet Version, Stufe (0-ply), Adresse und die Ersatz-Engine (fallback big).

Weitere Schalter auf der Optionsseite:

- **D debug log:** Anfrage und Antwort von gnuhg stehen im Log.
- **C check:** MMbg bewertet auch die gnuhg-Züge mit. Das ist langsamer, aber gut für Vergleiche.
- **V level:** nur ein Text für das Log, z. B. „2-ply“. MMbg kann die Spielstärke von gnuhg **nicht abfragen**: Sie wird in gnuhg selbst eingestellt (z. B. `set evaluation chequerplay evaluation plies 2`). Wer im Log festhalten will, gegen welche Stufe gespielt wurde, trägt sie hier passend von Hand ein und speichert mit S.

gnubg 1.08.003 vertauscht im `external`-Modus den Spielstand bei Zug- und Doppel-Anfragen. MMbg gleicht das aus. Der Fehler ist an die gnuhg-Entwickler gemeldet.

11. Lange Testläufe (Long Runs)

Für Stärkevergleiche spielt der Computer beliebig viele Matches gegen sich selbst oder gegen gnuBG.

1. Engines wählen, z. B. A4E (big gegen gnuBG) oder A45 (big gegen deep).
2. Modus **C4** (Turbo) starten: kein Bildschirm, endlos Matches, eine Statuszeile mit Zwischenständen. **C3** läuft mit Bildschirm, aber ohne Pausen.
3. Ist **Log** an, steht jeder Zug in der Log-Datei. Für lange Läufe sollte der Log-Ordner auf der SD-Karte liegen, z. B. B:/mmbg/log (Kapitel 3.3).
4. Mit **jeder Taste** stoppen.

Hilfen für faire Vergleiche (Optionsseite):

- **U duplicate:** Jedes Match wird mit denselben Würfeln und getauschten Seiten noch einmal gespielt. Das gleicht Würfelglück aus.
- **R dice seed:** ein fester Startwert für reproduzierbare Würfel. 0 nimmt den Startwert aus der Uhr.

Auswertung:

- Am Matchende schreibt das Log `### MATCH` und `### total: games ... matches ...`.
- Der Titel zeigt den Matchzähler `M a:b`.
- Beispiel: Ein Nachtlauf über 13 Stunden ergab 117 Matches big gegen gnuBG 2-ply, 46,2 %, im Mittel 6,5 Minuten je Match.

12. Optionen

12.1 Schnellmenü (Taste O)

```
1 anim+ 2 pips+ 3 hint- 4 flip+ 5 swap+ 8 mouse+ 9 mat+ 0 sound- 6 save 7 more
```

Eine Taste schaltet die Option um; **+** heißt an, **-** aus.

Taste	Option
1	Animation der Züge
2	Pips im Titel
3	Tipp (bester Zug, Doppler-Empfehlung im Log)
4	Brett spiegeln (Heimfeld links)
5	Farben tauschen (O rot, X hell)
8	Maus an/aus
9	<code>.mat</code> -Aufzeichnung (wirkt ab dem nächsten Match)
0	Geräusche: Würfeln, Ziehen, Aufsetzen, Schlagen (nur mit Animation, braucht <code>OPTION AUDIO</code>)
6	Einstellungen speichern (<code>mmbg_pc3.ini</code>)
7	Optionsseite

12.2 Optionsseite (O → 7)

Die Seite zeigt **alle** Einstellungen mit ihrem Wert. Eine Taste plus ENTER ändert eine Einstellung, ENTER allein führt zurück ins Spiel.

```
MMbg PC3 1.10.22 - OPTIONS (C) 2026 Matthias Hollar
key + ENTER = change      ENTER alone = back to the game
-----
BOARD  1 animation      +      4 flip board      +      home board left
        2 pips in title  +      5 swap colours   +      0 red, X light
        3 hint          -      8 mouse          +      USB mouse found
        0 sound         -      dice, moves, hits (PLAY BBC SOUND)
-----
RECORD  9 match file .mat +      G log file      +
        L folder b:/bg/f/ for log and .mat files, must exist
        now: log bg_20261019_210528.log match -
-----
GNUBG   external engine E = gnubg on a PC over WLAN (TCP)
        on the PC start gnubg and type: external :41000
        H host 192.168.1.132 P port 41000
        W timeout 30 s F fallback 4 big
        D debug log - C check gnubg's moves + (slower)
        V level (text for the log) 2-ply
        T test connection (version + 10 x ping)
-----
TEST RUNS U duplicate - each match again, same dice, sides swapped
          R dice seed 0 = from the clock (now 1717527)
-----
STATUS  WLAN: connected, IP 192.168.1.150 gnubg at 192.168.1.132:41000
        S save all settings (mmbg_pc3.ini + bgext.ini)
-----
option > _
```

Bild 9: Die Optionsseite. Hier liegen Logs und `.mat` in einem eigenen Ordner auf der SD-Karte, gnubg läuft auf dem PC 192.168.1.132.

Bereich	Tasten
BOARD	1-5, 8, 0 wie im Schnellmenü
RECORD	9 <code>.mat</code> , G Log an/aus, L Ordner für Log, <code>.mat</code> und Bildschirmfotos
GNUBG	H host, P port, W timeout, F fallback, D debug log, C check, V level, T test
TEST RUNS	U duplicate, R Würfel-Startwert
STATUS	WLAN-Zustand; S speichert alles (<code>mmbg_pc3.ini</code> und <code>bgext.ini</code>)

13. Befehlsübersicht

Die Taste **H** zeigt alle Befehle und darunter unter ACTIVE, was gerade eingestellt ist:

```
COM4:115200bps - Tera Term (1) VT
File Edit Setup Control Window Help
MMbg PC3 1.10.22 - Backgammon for MMBasic (C) 2026 Matthias Hollatz
Free software, GNU GPL v3 or later. Neural nets: GNU Backgammon (GPL).
=====
MOVE      13/7 8/7 bar/22 6/off      gnuhg notation (b = bar, o = off)
          13 24 b 6                  start points only, big die first
          #3                          move 3 of the move list
          E end turn    U undo        U again: back to earlier turns
=====
MOUSE     left / right on own checker move it with big / small die
          own dice cube board         undo / double / roll
          offered cube                 take (pass only with key P)
=====
CUBE      R roll    D double          asked before each roll
          T take    P pass            when the opponent doubles
=====
MATCH     M new     M7 to 7   K cube   new match, length, cube on/off
          SG save   LG load         it is also saved by itself
=====
GNUBG ID  position:match             paste at a prompt = set up (y/n)
                                     current ID: bottom right + terminal
=====
COMPUTER  C0 C1 C2 C3 C4             off, plays X, both, fast, turbo
          A45   C4E                 engines 0 = 4, X = 5 / + turbo
          engines: 0 base 1 book 2 tuned 3 net 4 big 5 deep E gnuhg
=====
OTHER     L D P Ctrl+P               list, dump, DEV column, screenshot
          O G GN H Q                 options, log, new log, help, quit
=====
ACTIVE
Match     to 5 points, 0 0 : X 0, no game yet   Cube 1
Players   0 you   X big   computer C1 plays X
Options   anim+ pips+ hint- flip+ swap+ mouse+ mat+ log+ sound- (key 0)
Files     log bg_20261019_210528.log match -
Saved     -
Nets      small + big loaded (engines 3, 4, 5)
Bear-off  Bear-off database: bgbo15.dat (up to 15 checkers, with gammons).
WLAN      WLAN: connected, IP 192.168.1.150   gnuhg at 192.168.1.132:41000
gnuhg     level 0-ply, fallback big, check on, timeout 30 s
=====
any key = back
```

Bild 10: Die Hilfeseite (hier im Terminal, auf dem HDMI-Bildschirm sieht sie gleich aus).

Befehl	Wirkung
13/7 8/7, 13 24, #3	ziehen (Kapitel 5.2)
E / U	Zug beenden / Undo
R D T P	würfeln, doppeln, annehmen, aufgeben (an den Doppler-Fragen)
N / M7	neues Match / Match bis 7
K	Doppler an/aus
SG / LG	Match speichern / laden
GNUbg ID einfügen	Stellung übernehmen
C0 ... C4	Computer-Modus
A45, A4E, C4E	Engines je Seite (plus Turbo)
L	ganze Zugliste
D	Brett-Dump (Terminal)
P	DEV-Spalte (Terminal)
O	Optionen
G / GN	Log an/aus / neue Log-Datei
Strg+P	Bildschirmfoto (BMP)
H	Hilfe mit ACTIVE-Übersicht
Q	beenden

14. Dateien

Datei	Ort	Wer schreibt	Inhalt
mmbg_pc3.ini	Programmordner	O → 6 oder S	Oberfläche: anim, pips, hint, flip, swap, mouse, mat, sound, match, cube, computer, Engines
bgext.ini	Programmordner	Optionsseite S	gnubg/WLAN, Log-Ordner, Duplicate, Startwert
mmbg_pc3.sav	Programmordner	automatisch, SG	laufendes Match (Zugverlauf, .mat-Stand)
bg_JJJJMMTT_HHMMSS.log	Log-Ordner	Log an	jeder Zug, Doppler, Ergebnisse
mmbg_JJJJMMTT_HHMMSS.mat	Log-Ordner	mat an	Match für gnubg
mmbg_JJJJMMTT_HHMMSS.bmp	Log-Ordner	Strg+P	Bildschirmfoto
bglog_err.txt	Programmordner	bei Problemen	Schreibfehler des Logs

15. Hinweise und Grenzen

- **Nur PC3, nur Firmware V7.00.00 ab RC2** (siehe Deckblatt).
- **Große Programme in RC2:** Ein Programm, das ohne `,C` den Programmspeicher von 256 KB sprengt, kann das Flash beschädigen („Program damaged: symbol table“ in Endlosschleife). Die veröffentlichte Fassung ist dafür klein genug; Entwicklungsfassungen mit allen Kommentaren nur mit `,C` laden.
- **Programmabbruch mit Strg+C:** Die Farbpalette bleibt verstellt, `MAP RESET` stellt sie zurück. Bei normalem Ende mit `Q` passiert das automatisch.
- **Animation:** MMbg zeichnet den Hintergrund des fliegenden Steins aus einem Bildpuffer (150 KB PSRAM). Fehlt der Puffer, zeichnet es das Feld wie früher Stück für Stück neu; das ist langsamer, sieht aber gleich aus.
- **Geräusche** gibt es nur mit Animation (C0 bis C2). Ohne `OPTION AUDIO` bleibt die Option aus und das Log meldet „no audio output“. Die Klänge (Würfeln, Ziehen, Aufsetzen, Klack) sind aus Rauschen gebaut und ein erster Wurf. Wer bessere Einstellungen findet, etwa andere Hüllkurven oder Lautstärken, meldet sie gern im TheBackShed-Forum; sie fließen dann in die nächste Version ein. Die Werte stehen im Quelltext in `GuSnd` und `GuSndOn`.
- **Nur Matchspiel** bis 15 Punkte, kein Geldspiel. Undo wirkt nur im laufenden Spiel.
- **Eine eingefügte GNUbg ID** beendet die `.mat`-Aufzeichnung für dieses Match, weil eine beliebige Stellung im `.mat`-Format nicht darstellbar ist.

16. Lizenz und Dank

MMbg ist **freie Software** unter der **GNU General Public License v3 oder neuer**: Es darf genutzt, weitergegeben und verändert werden, auch kommerziell. Wer es weitergibt, muss es wieder unter der GPL tun, mit Quelltext. Es gibt keine Garantie.

```
MMbg - Backgammon for MMBasic
Copyright (C) 2026 Matthias Hollatz
Neural network weights derived from GNU Backgammon,
(C) the GNU Backgammon authors (GPL v3 or later).
```

Dank an

- **Geoff Graham**, den Erfinder von MMBasic,
- **Peter Mather** für die PicoMite-Firmware, die PC3-Firmware und die BBC-Sound-Engine,
- die Autoren von **GNU Backgammon** für Netze, Methoden und die Prüfung über `external`,
- die Community im **TheBackShed-Forum**.

Viel Spaß beim Spielen!

17. Ein Flug durch den Quellcode

Für alle, die in `mmbg_pc3.bas` hineinschauen oder es ändern wollen: ein kurzer Rundgang. Das Programm hat rund 6 200 Zeilen und 260 Routinen. Die Namen verraten den Bereich:

Vorsilbe	Bereich
Bg...	Spiel: Regeln, Eingabe, Zuggenerator, Bewertung, Doppler, Match, Log, Speichern
Nn..., Z..., Bo...	kleine Netze, große Netze, Bear-off-Datenbank
Ex...	gnubg über WLAN (TCP), Optionsseite
Gu...	Grafik, Maus, Optionen, Geräusche
Gid..., BgId...	GNUbg ID
BgMat...	.mat -Aufzeichnung

17.1 Start

1. **Globale Variablen** (DIM ...): Brett `bd(0..25)` (+n = O-Steine, -n = X-Steine, `bd(0)` Bar von O, `bd(25)` Bar von X), Würfel `dw()`, Zuglisten `gm$()/gv()`, Netzgewichte, Bildschirmzustand `yb()`.
2. `MODE 3`, `GuPal` (Farben), `BgLogo` (Startbild).
3. **Laden:** `NnLoad` (kleine Netze), `BoOpen` (Bear-off-Datenbank), `ExIni` (gnubg-Einstellungen), dann `ZLoad` (große Netze, nur wenn der Speicher am Stück reicht).
4. `GuInit` (Bildpuffer, Farben), `GuCfgLoad` (`mmbg_pc3.ini`), `GuMsInit` (Maus), Frage nach dem gespeicherten Match (`BgLoadG`).

17.2 Die Hauptschleife

```
D0
  BgShow           Terminal und Bildschirm zeichnen (rechnet die Zugliste)
Eingabe:  Computer -> bester Zug gm$(1)   |   Mensch -> GuIn$ (Tastatur, Maus)
Befehl?   H, O, N, M7, C1, A45, SG, LG, U, GNUbg ID ... -> ausführen
Zug?      BgInput -> BgTry -> BgMove      (prüfen und ziehen)
          GuAnim -> GuMv -> GuFly        (Animation, Geräusche)
          BgNext -> BgAuto -> BgTurn      (Seitenwechsel, Doppler, Würfeln)
  Spielende? BgEndGame                      (Punkte, Log, .mat, nächstes Spiel)
LOOP
```

17.3 Wie der Computer seinen Zug findet

1. BgTurn: erst BgCubeTurn (Doppler vor dem Wurf, BgCube rechnet die Match-Gewinnchancen wie gnuBg 0-ply), dann BgRoll.
2. BgPrep ruft den **Zuggenerator** BgGen: Er probiert alle Würfelreihenfolgen und Startpunkte mit einem eigenen Stapel statt Rekursion (der PicoMite-Stapel reicht nur für wenige Ebenen). BgMx prüft vorher, wie viele Würfel höchstens spielbar sind.
3. Jeder fertige Zug landet in BgLeaf; doppelte Endstellungen werden verworfen. BgEval bewertet die Stellung: Formel (Engines 0-2) oder Netz über NnEval → NnFwd (klein), ZFwd (groß), NnBo (Bear-off-Datenbank).
4. BgSort sortiert, ZBest rechnet die besten 12 mit dem großen Netz nach, ZThink schaut bei deep einen Wurf voraus, BgBook setzt bei Bedarf den Buchzug, ExMove fragt gnuBg.
5. Platz 1 der Liste wird gespielt.

17.4 Weitere Bausteine

Routinen	Aufgabe
BgGid\$, BgIdGet, BgIdPut	GNUbg ID erzeugen, prüfen, übernehmen
BgHist, BgBack	Zugverlauf (eine ID je Zuanfang) und Undo
BgSaveG, BgLoadG	mmbg_pc3.sav schreiben und lesen
BgMatNew, BgMatTurn, BgMatGame	.mat im Format von gnuBg
BgLog, BgLogTurn	Log-Datei, eine Zeile je Zug
ExAsk\$, ExBoard\$, ExCube	Gespräch mit gnuBg über TCP (FIBS-Brettzeile)
GuFull, GuDraw, GuPt, GuDice	Bildschirm neu oder nur Geändertes zeichnen
GuFly, GuNear, GuRep	fliegender Stein, Mauszeiger, Rückfall ohne Bildpuffer
GuHit\$, GuShort\$	Mausklick und Kurzeingabe in einen Zug übersetzen
GuSnd, GuSndOn	Geräusche mit PLAY BBC SOUND
BgHelp, ExMenu, GuOpt	Hilfe, Optionsseite, Schnellmenü