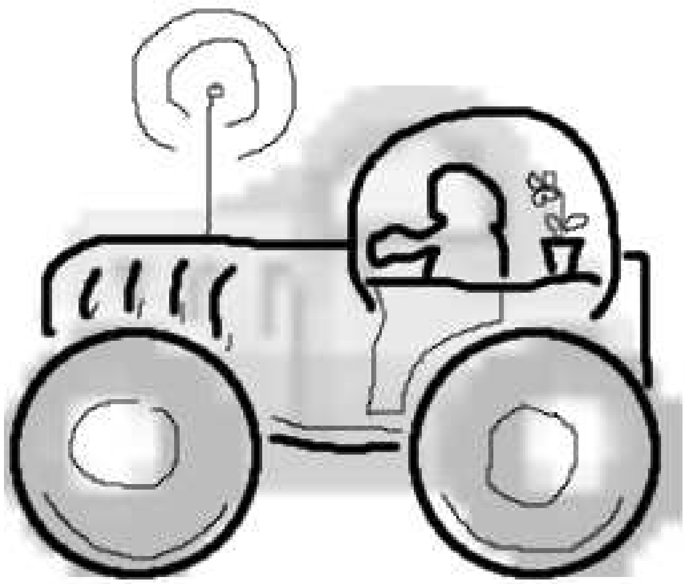
Moon-Buggy Usage and Maintenance

27 December 2004
Edition 1.0.51+esd



by Jochen Voß

1 Introduction

Moon-buggy is a simple character graphics game, where you drive some kind of car across the moon's surface. Unfortunately there are dangerous craters there. Fortunately your car can jump over them!

Moon-Buggy comes with no warranty, to the extent permitted by law. You may redistribute copies of Moon-Buggy under the terms of the GNU General Public License. For more information about these matters, read the file '`COPYING`' of the source code distribution or press ⟨c⟩ at moon-buggy's title screen.

Please mail any suggestions and bug reports to `voss@seehuhn.de`. Your message should include the moon-buggy version number, as obtained by the command '`moon-buggy -V`'.

New versions of Moon-buggy may be found on the sunsite ftp server or on the moon-buggy download page (see Chapter 6 [References], page 3).

2 Moon-buggy Invocation

Normal usage is to start '`moon-buggy`' without any options. Nevertheless there are some. The long options (starting with '`--`') are only supported on some system types.

'`-h`'
'`--help`' shows a short usage message.

'`-m`'
'`--mesg`' disable write access to your terminal by other users, so your boss can't disturb your game play with the help of "write" or "finger".

'`-n`'
'`--no-title`'
skips the title screen.

'`-s`'
'`--show-scores`'
shows the current highscore list and exits.

'`-V`'
'`--version`'
prints the program's version to standard output and exits.

3 Playing the Game

The game is controlled by a couple of keys, which are described near the bottom of the screen. Most important are the following keys.

⟨SPC⟩
⟨j⟩ makes your car jump. The length of a jump is fixed and you can only start a new jump while the wheels have contact to the ground.

⟨a⟩
⟨l⟩ fires the mining laser. The laser is used to remove stones, which may block your way.

⟨q⟩ quits the game prematurely. But you still keep your score and may enter the highscore list.

Moon-buggy can handle a wide range of screen sizes. But it runs best if the screen has at least 80 columns. If you run it on a screen with significantly less columns it may look strange and could be hard to play.

4 Shared Score Files

Moon-buggy has two concepts of score files: the *global score file* is located somewhere in the installation directories and is shared between all players. The exact place in the file system can be controlled via `configure`'s '`--sharedstatedir`' option. This highscore file is used whenever Moon-buggy has the permissions to access it. Otherwise the *local score file* is used. This file is located in the user's home directory, the file name is '`.mbscore`'. Thus, the local score file is not shared between different accounts.

As the game is much more fun, when players can compete for the highest scores, the global score file should be writable for the moon-buggy program (but not for the user's text editor). Here is a list of possible policies:

1. The preferred solution is to make moon-buggy a setgid program. This means the following: you choose a group ID, which cannot be used by ordinary users, and allow only members of this group to write the score file. Many systems have a group '`games`' for this purpose (Check your '`/etc/group`' file). You make this group own the '`moon-buggy`' executable and set the file's setgid bit. This bit gives moon-buggy its file groups's access permissions and so the program is able to access the score file.

   You can use `configure`'s '`--with-setgid`' option (see Chapter 5 [Installation], page 2) to properly set up access permissions for this model.
2. You could make moon-buggy a setuid program but for security reasons this is not recommended. The file '`SECURITY`' from the bsd-games distribution (see Chapter 6 [References], page 3) has some informations about this topic.
3. Another solution is to make the score file world writable. But then you have to trust your players: every player may open the score file with his favourite text editor and change it freely. So this is not recommended, too.
4. Finally you could do nothing. Then every account uses its local score file.

5 Installation

Moon-buggy requires the curses library as a prerequisite. If curses is not installed on your system, you may use the free ncurses package (see Chapter 6 [References], page 3). The game does not work with BSD curses, thus on BSD systems you will need the ncurses package.

When you unpack the moon-buggy distribution, you get one directory, which holds all the source files and the documentation. Generic installation instructions are in the file '`INSTALL`'. There are some points of interest:

- By default, the program will be installed as '`/usr/local/bin/moon-buggy`'. You can specify an installation prefix other than '`/usr/local`' by giving '`configure`' the '`--prefix=`*`path`*' option.
- You can choose the score file's location via `configure`'s '`--sharedstatedir=`*`dir`*' option. The score file then will be '*`dir`*`/moonbuggy/mbscore`'. On GNU/Linux systems you should use

  ```
  --sharedstatedir=/var/games
  ```

  to comply with the Filesystem Hierarchy Standard (see Chapter 6 [References], page 3).
- Moon-buggy supports shared score files. For details, see Chapter 4 [Shared Score Files], page 2. As explained there, you may want to make moon-buggy a setgid program. This can be done with the '`--with-setgid`' option. If you use

  ```
  --with-setgid=games
  ```

  then the installation process arranges everything for setgid usage.

  Caution: this introduces potential security risks. I tried to minimise these, but nevertheless be careful with this. And remember: moon-buggy comes with no warranty.
- If your version of the curses library is not autodetected, you have to use some of the '`--with-curses-includedir`', '`--with-curses-header`', and '`--with-curses-libs`' options. For example you should use

  ```
  --with-curses-includedir=/usr/pkg/include
  ```

  if your curses header files are in '`/usr/pkg/include/`'. You may use

  ```
  --with-curses-header="<mycurses.h>"
  ```

  if your curses header is called '`mycurses.h`'. And you may use

  ```
  --with-curses-libs="-L/usr/pkg/lib -lncurses"
  ```

  if your curses library is called '`ncurses.a`' and is located in '`/usr/pkg/lib/`'.

Example: On GNU/Linux systems you probably can use the following commands. For the last one you need root user permissions.

```
./configure --sharedstatedir=/var/games --with-setgid=games
make
make install
```

6 References

This chapter gives some references to resources mentioned in the manual. As the internet changes incessantly, the addresses below may be outdated. If you find this to happen, please report it as a bug.

The latest version of **Moon-buggy** may be found on the sunsite archive in the directory `ftp://sunsite.unc.edu/pub/linux/games/arcade/`.

I provide a **moon-buggy download page**, which is located at `http://seehuhn.de/comp/moon-buggy.html`.

Version 2.0 of the **Filesystem Hierarchy Standard** (FHS 2.0) may be found on the internet at `ftp://sunsite.unc.edu/pub/linux/docs/fhs/fhs-2.0.tar.gz`.

The **ncurses** package can be found on the GNU project's ftp server and its mirrors. The original site is `ftp://prep.ai.mit.edu/pub/gnu/ncurses/ncurses-4.2.tar.gz`.

Hints about **security concerns** may be found in the file ‘`SECURITY`’ from the bsd-games package. One location where to find it is `ftp://sunsite.unc.edu/pub/linux/games/bsd-games-2.7.tar.gz`.