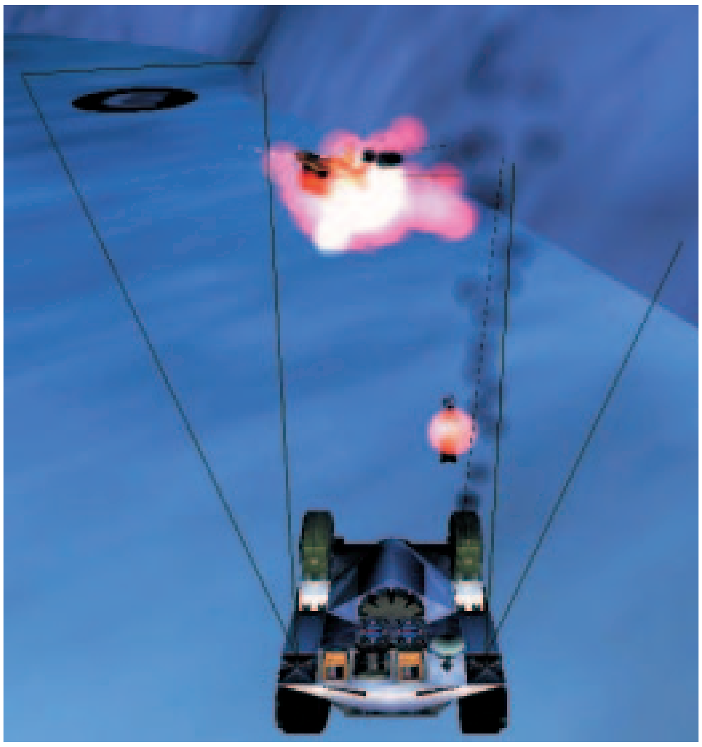
Linux Games are here! MINDROVER

FIONN BEHRENS



When I was about 16, during my blessed Amiga days, a game called C-Robots fell into my hands. This involved using the programming language C (in simplified form) to program a robot, which then had to defend itself, with the aid of this program, independently against other robots. This game introduced and clarified the C programming language for me, and I have always wondered why nobody is still making such a game in a more modern form – till now. And what lies before me now has definitely got a high addiction factor.

There has certainly been some progress in terms of form in the past 10 years: Now the robots no longer race around as red and white dots on a whitish grey chequered square, but are shown looking smart and full of detail in 3D, racing through arenas full of obstacles and traps, equipped with an enormous selection of accessories and aids. But I digress – let's start from the beginning.

The Mindrover game has been around for some time for an unimportant operating system. But now the maker *Cognitoy* has decided to convert the new and heavily revised version of this program just for Linux. The advantage of this is that, as a rank beginner one can immediately access a large collection of completed robots, forums and add-ons on the Web, which provide help and examples if you have a problem. In practical terms the new version is fully compatible with the old one. The old robots just have to be recompiled.

First, the game has to be installed. While it is common nowadays to see games gobbling up disk space by the gigabyte, at just over 60 MB, Mindrover will strike you as pretty laughable. As usual, installation goes smoothly with the Loki installer (which also brings with it the necessary icons for the Gnome and KDE desktops). As already mentioned, the game requires a 3D graphics card with OpenGL support; Apart from this the demands made on the computer are only slight. On older computers the game start does take a bit longer, but then even with a P200 everything should go smoothly.

After the start Mindrover welcomes you with a log-in screen, so that several users can use the one program with different settings and robots. After

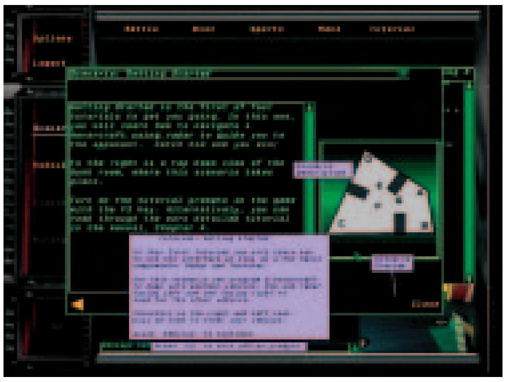
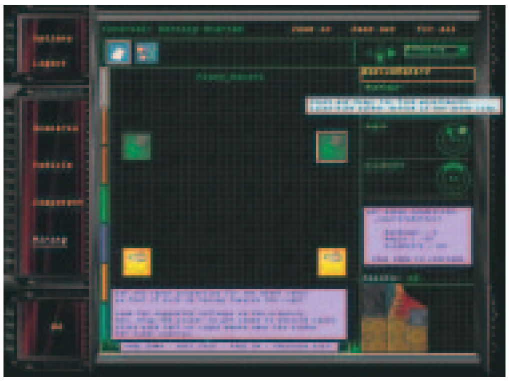


Fig. 1: Well-made online help and context-oriented tutorial



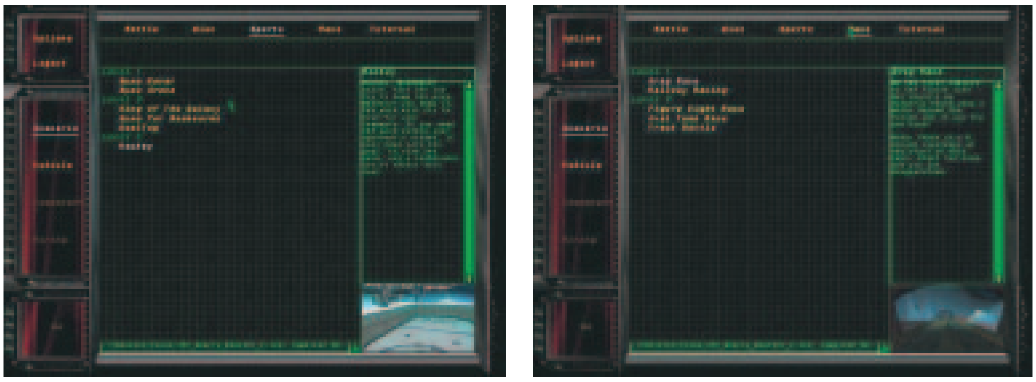


Fig. 2: Enormous varieties of scenarios allow for a multiplicity of solutions

entering your name, you come to the console, where you have to decide on a scenario. Mindrover waits here with a very large and effective selection of task settings for the ambitious robot constructor. The scenarios are divided into several sub-groups such as Fight, Sport, Race, Various or Training. With regard to training: practically the entire content of the well-made manual can also be found within the game, as well as occasionally context-sensitive online help within an extensive tutorial, competently introduces you to the first steps towards your own robot.

Of course, old favourites such as Rocket Arena are among the scenarios, but also others such as Sumo Wrestling, in which the aim is to avoid getting caught. There are some where you have to find and collect as many valuable items as possible, and various types of races. Many other challenging and original situations are provided by the game.

To fulfil the multitude of tasks the robot can be issued with an equally large number of sensors and devices. These are then linked with each other using the built-in, graphical programmer interface and must be made to co-operate. To do this, the programmer has the use of logic operators, counters, registers, status models and much more in graphical form, all well explained and very simple to use. Two components are always linked through a vector by mouse, which then defines the type of interaction. That is very simple in principle, yet one very soon reaches a degree of complexity in the wiring, making it hard to keep an overview. But it appears that even users with no programming experience cope very well with this system. However, it is advisable to assign each component and each vector a meaningful name right at the start.

If you have finally created something, and think it really ought to work, you can pit it against another robot. Neatly enough, Cognitoy has also thought of options for issuing status messages and similar in the form of text and coloured lights or even fireworks because your own creations do not always behave as intended. To be more precise, they hardly ever do. So the beginner rapidly learns that robotics really is a complex field of activity. Unfortunately the occasionally clumsy and hard to influence camera direction makes it difficult in complex scenarios to observe close up and from a good angle. There is some room for improvement here.

If at some point the perfect robot is completed, it can be swapped with friends, with or without the graphical source code. Your creation can then be set against their robot. This is where the true challenge of the game lies as the robots installed in the game may be very good but not exactly works of genius.

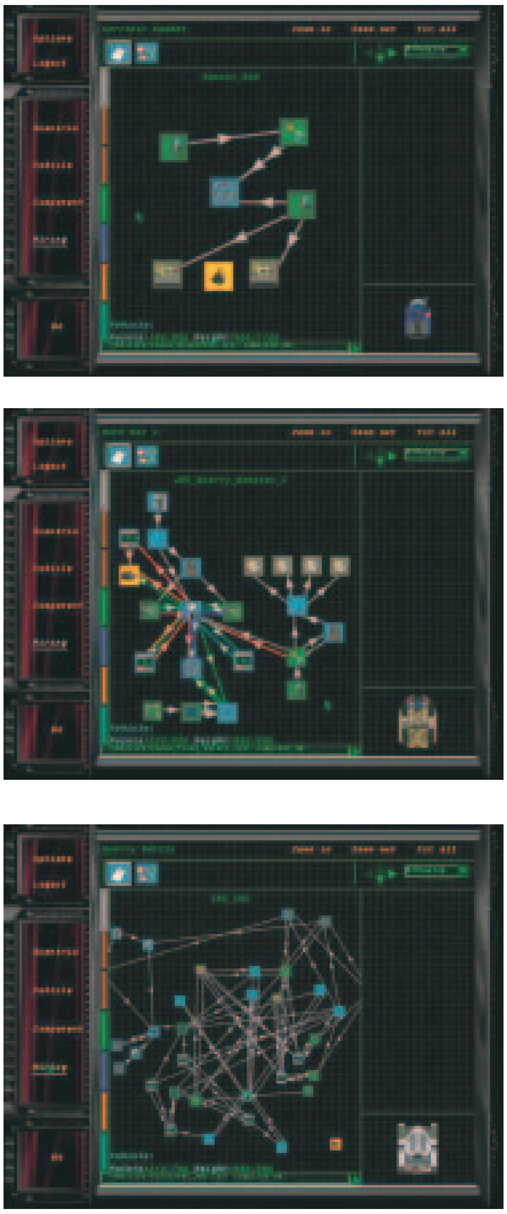
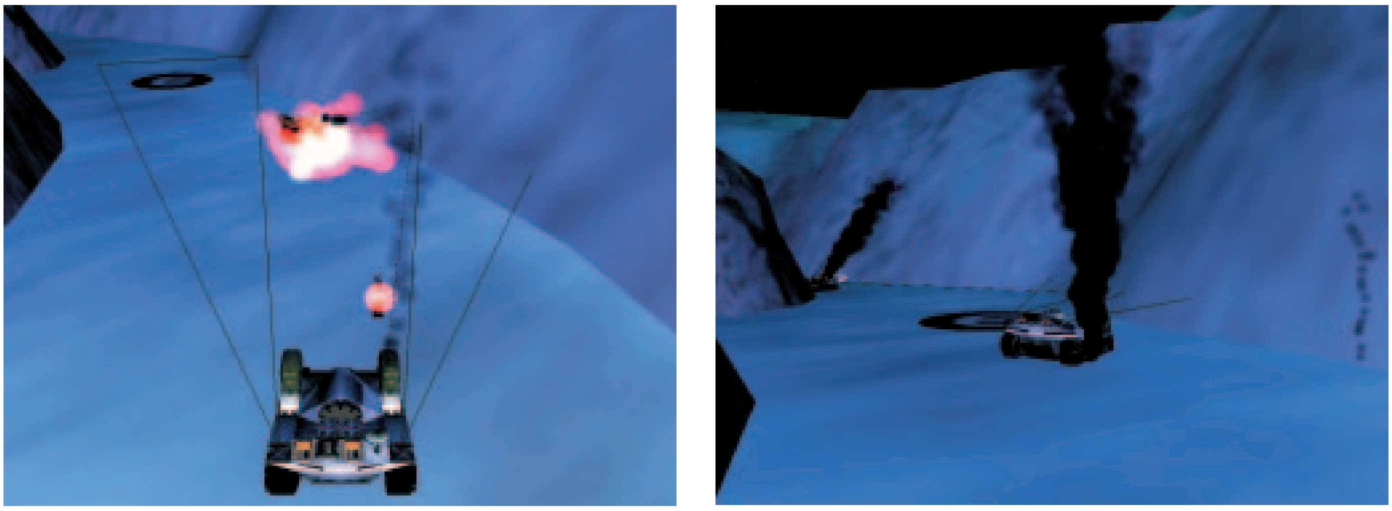
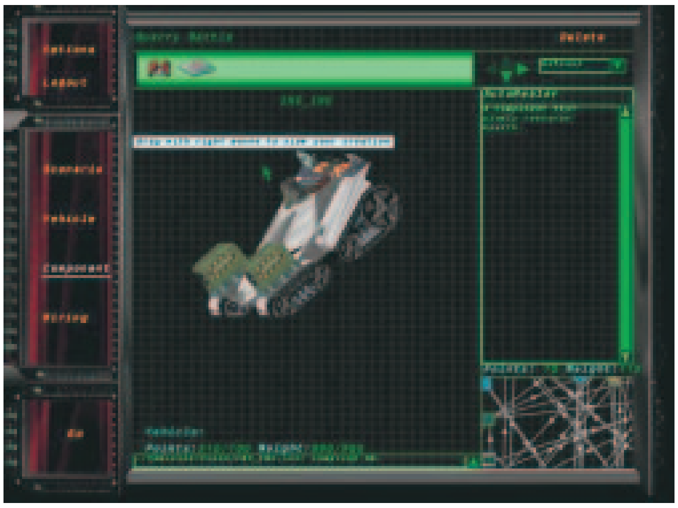


Fig. 3: Robots can be programmed to be as complex as you like – but more often than not simple programs are the most effective





[top]
Fig. 4: Strike! But a few seconds later both are badly hurt...

[above, left]
Fig. 5: The finished vehicle

[above, right]
Fig. 6: Parental nightmare 2010: the children's programmable fighting machines turn the hall of your house into a theatre of war

By its nature the game has no multi-player mode or similar, and the 3D-Shooter fan will probably find this game dreadfully boring. I can only say that once you have made something by yourself and it wins, then you'll spend the whole day constantly thinking up new algorithms and dreaming up ways of making further improvements to your own baby. The construction of a new machine can easily bind you to the screen for many hours and when else could you have such fun giving yourself a splitting headache?

The only thing I really missed was an option for experienced programmers to program the machines avoiding the graphical interface in a text mode facility. This is in fact possible, because – oh glory – the graphical wiring diagram is translated by the game into source code. This in turn is converted by a compiler into an executable robot, but these sources can also be edited and compiled outside the game (and then of course the graphical representation gets lost). The game then has to start again and Mindrover is not exactly a fast starter. So the conventional debugging of a robot on a textual basis rapidly turns into a game of patience. Also, the programmer language is a completely individual development and very limited in many respects.

Conclusion

Negative: OK, something did happen, but I'm not at all sure all the improvements will really be better. It may be that I'm now on the wrong side of the fence to be able to judge if it is really useful to tempt in computer-newbies with a purely graphical programming interface. As a vi spoilt Visual-Basic hater I have in any case sometimes had a lot of trouble with what Mindrover presented to me here. Nevertheless it was more fun than bother.

Positive: Forget crossword puzzles, forget logic games. This thing doesn't even cost one fifth of the price of an ordinary Lego Mindstorm basic building set, but can be at least as challenging. Anyone who could use an 'educationally valuable' change from the 3D-Bang-You're-Dead standard brew, which at the same time also exercises the grey cells and has a true long-term fun factor, should grab it. ■

Evaluation:

Long-term game fun:	*95%*
Graphics:	*68%*
Sound:	*40%*
Control:	*55%*
Multi-player: nla	
Total score:	*80%*

The author

Fionn Behrens is a student of technical IT. But you can't study all the time... He can be contacted on the Net as Fionn at IRCnet.