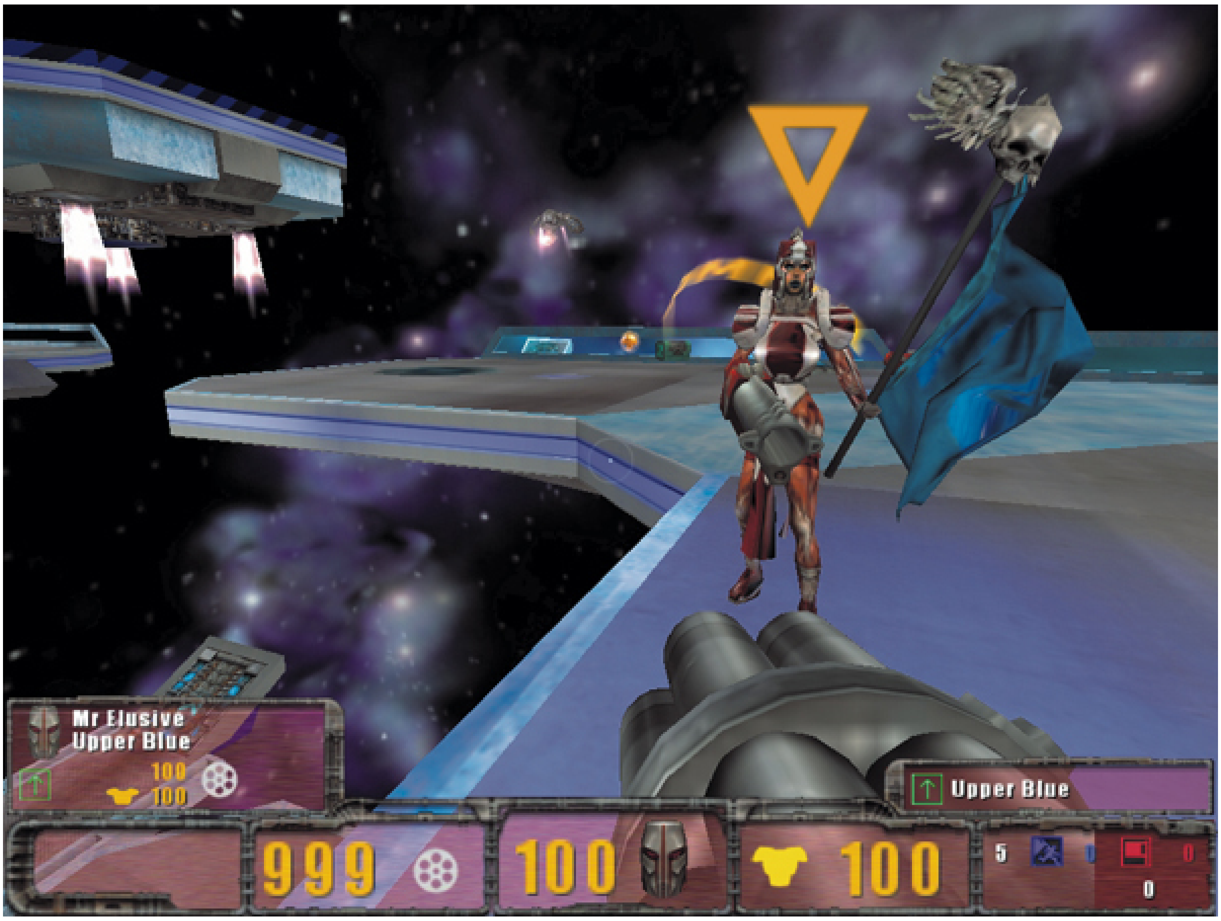
Soldiers for the virtual battlefield

Even at secondary school, Jan Paul van Waveren spent more time on computer games than on his homework, though he doesn't seem to have suffered from it.



Screen image from Quake III Arena, a so-called first-person shoot-'em-up (FPS) game in which the player sees the world from the first person perspective.

christian jongeneel

Even at secondary school, Jan Paul van Waveren spent more time on computer games than on his homework, though he doesn't seem to have suffered from it. At university he spent some of his time linking up personal computers with his friends, trying to blast each other to virtual smithereens in a game of Doom. Van Waveren is now on the verge of being awarded his Master of Science degree for a thesis on the artificial intelligence he developed for the Quake III Arena game. He has already started working for the American game company, id Software.

Most of the time, Jan Paul van Waveren is plain Jan Paul van Waveren, but sometimes he lurks in dark dungeons carrying an outsize gun and shooting anything that moves. Then he's Mr Elusive. Mr Elusive has seen his kill rate go down a bit of late. It's not just that it's about time he graduated, after six years at tu delft

reading Applied Informatics, he also couldn’t resist a job offer by
a U.S. games manufacturer. He is now busy programming the games he once played. Every programmer knows that games are among the most demanding software around. Glitches in office software may be acceptable, but in games they are anathema.

For Jan Paul, it all really started with his interest in model helicopters. From these he progressed to building radio-controlled model cars. However, their appeal proved to be limited and during his first years at secondary school, he became fascinated by hovercraft and applied himself to designing these. The first working hovercraft was developed by a British inventor Sir Christopher Cockerell (1911-1999) in 1958. Until last year, large versions maintained a regular ferry service across the Channel between Dover and Calais. Over a period of two years he built various models, using electric motors at first, then real combustion engines for model making. They were all designed down to the last detail, and everything pointed to his being destined for a future as a mechanical engineer. Then a friend introduced him to personal computers (powered by an Intel 80286 processor), computer games and programming. The games were the ones produced by his current employer, id Software: Wolfenstein and Doom. These are known as first-person games. The player looks through the eyes of a commando who has been sent into a maze
to fulfill a mission. The player’s task is to explore the maze, to find secret doors, hidden treasures, and other objects. And of course, to remove anyone standing in your way. The opponents, usually referred to as monsters, aren’t very intelligent: they are always waiting on the same spot, and their aim isn’t too accurate. Of course, there is always a chance of getting hit, so a health monitor on the screen shows you when it’s time to track down another first-aid kit. At the time, Doom was a revolutionary game. It offered better graphics than anything before, and let game freaks to create their own levels. Such computer games consist of various components. For example, there could be one component to produce the graphical representation of the maze, and another containing all the possible movements of the commando. A level is a data structure that contains the shape of the maze, the location of the treasures,
and the number of opponents. The creators of Doom decided to publish the specification of that data structure so people could design their own mazes. Hundreds of thousands of players who had completed all the mazes supplied with the game were then able to make their own versions and exchange them through the Internet, turning Doom into the «game that never ends».
Skins During my latter years at secondary school, I made a lot of levels for Doom, about one every week,’ Van Waveren says. ‘At first, the levels were distributed through a Bulletin Board System (bbs), later the Internet took over. Then came Doom’s successor, Quake 1 which was much better suited than Doom for playing through a local network or the Internet. I used to do that

a lot. Since then, the whole games world has become much bigger and more commercial. But the Quake 1 community was relatively small. I was part of a clan, a group of players who often play together. We had our own ‘skin’ and things like that.’

Now ‘skin’ is a word that requires some explaining. Each character in a computer game is constructed around a wire frame model. The faces of the model are filled with a texture, forming a pattern of colours. This pattern, the skin, determines the appearance of a character. Quake 1 allowed players to create their own skin. Monsters with the head of Microsoft boss Bill Gates soon became popular.

‘We used to come together, bringing our PCs, which by then were Pentium 133 machines,’ Van Waveren continues, ‘and spend hours wiring them up, so we would just have one hour left to play in before it was time to break up again. Later, thanks to the Internet, you could dive straight in. But even that had its drawbacks. To begin with, phone bills started to mount. Then, slow Internet connections made the game difficult to play. What’s more, you didn’t stand a chance against experienced players on the Internet unless you spent a lot of time practising. But then again, practising against the moron or lesser intelligent monsters included with Quake 1 wasn’t going to be very useful.

Van Waveren thought how nice it would be if you could practise by playing against the computer, but with more intelligent opponents. It turned out that he wasn’t the first one to think along those lines. The Internet offered a large selection of bots. Bots (the word is derived from robots) are programs that act intelligently. The programs used by search engines to roam the Internet looking for information to catalogue are also called bots. In computer games,

a bot is a program that behaves like a human opponent, i. e. with strategic insight and, to some extent, unpredictably. They make playing these games much more realistic.

‘Of the available artificial players, one stood out from the crowd,’ Van Waveren says, ‘but even so I started to notice peculiarities after a while. At the hardest level, the bot just couldn’t lose. True, the bot showed intelligent behaviour, but the moment it showed itself, it scored a kill. At lower levels, the bot had a less accurate aim, but then it also did less intelligent things.’

Van Waveren wanted to change the bot but he didn’t have the source code. The source code is the original, human-readable text that makes up a program before it is turned into the ones and zeroes the computer can understand.

‘So I started to make small changes to the game. One of the first changes was an anti-camp modification. Some players will simply hang around in a favourable location, waiting for others to walk into their trap. They are called campers and their behaviour is considered unsporting. My little program issued a warning if you stayed in one place for too long. If you didn’t leave, you were simply teleported to a different location, or a little thundercloud would appear over your head and follow you around wherever you went. All of which definitely didn’t improve your chances of survival.’

Omicron In 1997 Van Waveren had completed a bot that could reliably simulate a human player. Omicron, as the bot was called, became highly popular in no time at all. Within a few hours of the first release almost 10,000 people had downloaded it from the Internet! After two weeks, it had almost made it to the top-ranking most popular Quake bot. Van Waveren’s programming talent didn’t go unnoticed, and he was offered a job by an American software company. But he didn’t take it, as he wanted to graduate first.
‘When I started, I didn’t know a lot about artificial intelligence,’ he admits.
‘I simply browsed through the Quake source code, read a little here and there, and used my own intelligence.’ His intelligence made him decide to use waypoints, the digital version of Ariadne’s thread. A waypoint is a spot in the maze which the artificial player can use to get its bearings so it doesn’t get lost in the maze. This is only one of the many arts a bot has to master. Another is to be able to detect an opponent and deliver accurate fire. Of course, all this information about positions and opponents is known to the Quake program itself, but the bot isn’t supposed to use that information, or he would gain too much of an advantage over his human counterpart, and in a sense it would be cheating. Therefore, the bot is programmed to calculate whether a human player would be able to see an opponent from its current position, before it is allowed to open fire.
‘The waypoints method is not very advanced,’ Van Waveren admits. ‘You either have to program all the waypoints for the bot, which is a lot of work, or you have to let him explore the maze by himself, in which case it will behave like a novice player over and over again. An additional problem with Quake 1 was that the waypoints were difficult to program into the data structures of the special programming language used to modify Quake 1.’

The arrival of Quake 2 changed all this. Now the rules and artificial intelligence were programmed in the programming language C, which meant that all restrictions were lifted. On top of that, id Software, as they had done with Quake 1, published part of the program source code, enabling enthusiasts to adapt parts of the program to their own taste, which is how Van Waveren came to write the Gladiator bot for Quake 2. This bot offered many more options than its predecessor, allowing opponents to be programmed in subtle ways. For example, human players can indicate whether their artificial opponent is to show a preference for certain weapons, whether it has a defensive nature, or takes the offensive. The different settings are applied using fuzzy logic, for one. The result is that the artificial player’s properties vary, making it less predictable. Like a human player, the bot may have its preferences, but there is nothing to stop it taking another route and displaying different behaviour. The opponents were also given a vocabulary to communicate both with the human player and with each other. They could also understand simple sentences entered through the keyboard.

Capture the flag The Gladiator bot appeared on the Internet from December 1998 onwards. There was also a ctf (Capture The Flag) variant of the Quake 2 game on offer. Normally speaking, the object of the game is to get to a certain destination, usually a door to the next level. In ctf there are two different teams, and the object of the game is to capture the flag of the other team. Van Waveren programmed the bots so they also understood this type of game.
Again his activities didn't pass unnoticed: 'In June 1999 I got an e-mail message from Graeme Devine, project leader at id Software. They were working on Quake III Arena. They wanted to incorporate a bot, but they didn't have much experience with that kind of thing, whereas I had been making bots for two or three years. He asked if I was willing to come to Dallas, Texas, for a talk and to swap some ideas, and they sent me a ticket. The last thing on my mind when I went there was to land a job. It just seemed like a nice trip. When I got to Dallas, they first showed me how their bots were put together, then I showed them how mine worked. I had programmed my bots in such a way that they could be easily integrate into another game. That gave us the idea to put the theory to the test. It only took us a few days to adapt my bot for Quake 3. It worked so well that they thought that perhaps it would be a good idea to acquire the bot or parts of it. I sold them the source code. Then they hired me to help with further integration and extending the bot code. At first I spent most of my time on artificial intelligence, but now I'm also working on a number of other things, like parts for a new version of Doom.'

In the meantime, he also wants to graduate on the strength of his work for Quake 3. This means mostly writing, since the programming has already been done. The bot Van Waveren came up with for Quake 3 is special because not only can it manoeuvre through the maze on its own, it can also collaborate with other bots, like a military unit.

Basic skills 'The bot for Quake 3 uses four layers,' Van Waveren explains. 'The bottom layer contains the base: the bot's input and output, in other words the information the bot can acquire, and the actions it can take. The input includes a representation of the environment, and the output includes all the possible moves. Moves can include walking and jumping. I use areas to represent the environment. An area is a continuous section of the environment. The artificial player knows in which area it is located, and in which part of it. Based on that information it can quickly determine where it can go. When walking, this is simple: it can go to the adjacent areas. When jumping however, it can also reach non-adjacent areas. All this information has to be calculated and stored.'

So, the bottom layer determines the basic skills the artificial player has at its disposal. It also controls the general supply of information, but it doesn't do anything with it yet. The next layer mainly handles short-term objectives.

For example, if the bot cannot carry more than one object at a time, and it is already carrying a first-aid kit, it is unwise to try to pick up a second kit, unless it is badly hurt, of course.

'So, the bot has to decide how badly it wants the object,' Van Waveren says.

'I use fuzzy logic for this, letting it weigh the pros and cons, and draw its own conclusion whether or not to pick up the object.'

The third layer takes care of the real intelligence, i.e. setting out the long-term goals and the survival strategy. The bot does this by running through a number of state dependent procedures. It starts with its long-term goals, for example the elimination of a certain opponent. From this goal, it deduces its short-term objectives, taking into account the current conditions. This can result in it switching to a puzzle procedure to solve a problem.

A simple example: if the bot spots a desirable object that it can't reach (e.g. because there is a barred door blocking its way), it has to decide on a new interim objective that must be fulfilled in order to reach the object. The bot follows a chain of reasoning to solve the problem. If it encounters an opponent, it switches to the battle state. In such an event, the artificial player's properties, stored in the second layer, will determine whether it prefers to jump or duck to evade bullets. If the opponent runs off, the bot has to decide on the best course of action in the given circumstances. Should it follow in pursuit, or revert to achieving its long-term targets? If it decides to go after its opponent, it will have to reconsider after a few seconds to see if pursuit still is the best option. During combat, the bot can also decide to retreat if it is in a bad state.

Lieutenant The underlying technology is that of an expert system: lots of if-then rules, which are constantly being evaluated. The intelligence follows from the interaction of all those rules. Unlike with many traditional expert systems, there is a limiting factor: time. A smooth gaming experience is essential. There is little point in working through the entire set of rules every tenth of a second (which is the unit of time the Quake 3 bots use for their thinking process). If an artificial player has just scanned its surroundings for desirable objects, there is no need to do the same again after taking half a step. Staying on the lookout for opponents does remain crucial however. If you like, a team of bots can do everything for you. Human players can gather a number of bots around them, and command them like a general.

Van Waveren: 'You can do the same when you're playing with humans, but bots tend to be better at obeying orders'.

You can even go one step further and appoint a lieutenant and send him out with a couple of recruits to defend an important position. The intelligence for this kind of action is in the fourth layer. In addition to determining his own objectives, a leader has to think for his subordinates. Commands are issued through a text interface, and the bots don’t care whether they come from a human player or another bot. If you wanted to, you could even have armies of bots fighting each other, and then sit and watch the screen to see who comes out on top. But that’s not what so-called shoot- ‘em-up games were invented for. The name of the game is survival, working off aggression, problem-solving, and most of all, airing one’s feelings of superiority. And of course, there is the ever-smarter technology, the foundation of an industry worth billions and rapidly overtaking Hollywood in volume. n

For more information, please contact Jan Paul van Waveren, phone +31 172 588 264, e-mail mrelusive@idsoftware.com or Dr. Leon Rothkrantz, phone +31 15 278 7504, e-mail l.j.m.rothkrantz@its.tudelft.nl

Za P P W aR K