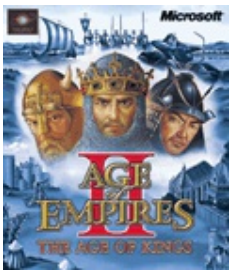
GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: Ensemble Studio's Age of Empires II: Age of Kings

By Matthew Pritchard



For anyone who has ever worked on a PC game and poured their heart and soul into their work, they may have imagined in an optimistic moment, “If this game sells a million copies…” Maybe it was spoken out loud, or carefully whispered so that no one else would hear. It’s the expression of the dreams and promise of success that drives so many of us. But recently I found myself somewhere I never anticipated as I listened to this ironic ending to that very statement: “… I am going to be so disappointed if that’s all it sells.” And you know what? I had to agree.

Catching Up

Two years ago in a previous postmortem published in *Game Developer,* I told you the story of Ensemble Studios, a scrappy upstart that overcame challenges to create the game *Age of Empires* (*AoE*). Since its release two years ago in the great real-time strategy (RTS) wars of 1997, approximately three million copies of *AoE* have been sold worldwide, along with almost a million copes of the *Rise of Rome* (*RoR*) expansion pack. The totals don’t give the whole story, though. *AoE* proved to be a consistent seller, hanging around the top of the PC Data charts, and even re-entered the top ten a year-and-a-half after its release. The demographics of the buyers were another surprise. Sure we had the sales to the 14- to 28-year-old male hard-core players, but we also had significant sales to older players, women of all ages, and casual game players of all sorts. That is to say we had a crossover hit on our hands. If you have ever watched the VH1 show *Behind the Music*, then you know the story of the upstart band that finds itself suddenly on top of the world — things change, and not always for the better. I wouldn’t go so far as to say that we sank into a wild orgy of sex, fast cars, and money — despite the wishes of a couple of our guys — but this change along with the benefits of success brought us a whole new set of challenges, making our next game no easier than the first.

Designing a Sequel

It was a surprise to no one that Ensemble Studios’ next game would be a sequel to *AoE*, although most people probably didn’t know that we had a contract with our publisher for a sequel long before the original game was finished. Given our historically-based themes and time periods in *AoE*, the chosen time period for *Age of Empires II: The Age of Kings* (*AoK*), the Middle Ages, practically picked itself. That was the only easy part, however. Like a band going back into the studio after a hit record, there were differing opinions of what direction to take next. Do we play it safe and stick tightly to the *AoE* formula, or do we get bold and daring and take the whole game genre in new directions? This is the million-dollar question every successful game is faced with when the topic of a follow-up is raised. But the successful band I’m using as an analogy is fortunate. They don’t have to contend with the unbelievably rapid pace of evolution in PC hardware and games.

Improvements to the game in every area from graphics to user interface are expected in this business as a matter of fact. Expectations can be a bitch sometimes. Take the vast demographics of *AoE* players that I mentioned earlier — they are the largest group of people most likely to buy the sequel — and everyone is concerned about making sure that this huge and diverse group will like the next game so much they will run out and buy it. We’ll just do more of what we did right in *AoE*, we said. That sounds great, but it’s almost impossible to quantify in a meaningful, detailed way. The game business is brutal to those who fail to move forward with the times, but it’s also equally brutal to those who experiment too much and stray from the expectations of the players.

A shot from a very early version of the game. Most everything shown would be revised before the game shipped.

When we started work on *AoK*, we thought that we could make use of our existing code and tools, and that this would make the sequel easier to create than the original. Filled with these optimistic thoughts, we concluded that we could develop *AoK* in a single year. This was also going to be our opportunity to add all those dream features and make our magnum opus of computer games. So we set about to do just that. To make enhancements for *AoK*, we had pulled together a giant wish list of features and ideas from inside and outside sources. To the game design we added all sorts of neat new features such as off-map trade, renewable resources, combat facings, sophisticated diplomacy and systems of religion, and so on. Of course, the art, sound, and game content were also going to be bigger and better and bolder and brighter and...well...you get the idea.

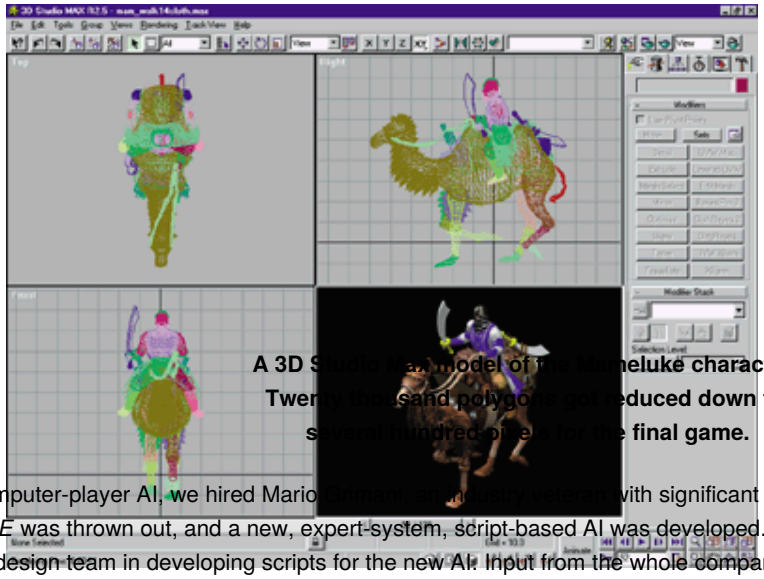
Several months down the road, reality reared its ugly head in big way: we had bitten off more than we could chew and the game's design was losing focus. Instead of sticking to the core of what makes an RTS game great, we had gone off in many contradictory directions. Along with that came the realization that there was no way that we were going to finish *AoK* in a single year and have it anywhere close to the quality of *AoE*. This was a sobering time for Ensemble Studios staff and our publisher, Microsoft. While the Ensemble Studios crew adjusted quickly, it caused a few problems for some of the people at Microsoft: "Uh, guys, we've already gone ahead and committed to our bosses that we would have another Age of Empires game this year," is probably a good way to paraphrase it. From this situation, a contingency plan was born. We were going to take another year to finish *AoK*, giving us time to get the game back on track and to create the ambitious content for it. We also had a plan to help our publisher out: we would create an in-house expansion pack for *AoE*. It would be a significant addition to the game, yet require only a small amount of our resources, and most importantly, it would be ready in time for Christmas 1998, taking the slot originally planned for *AoK*. Thus was born the *RoR* expansion pack. *RoR* helped, but it didn't take all the pressure off us. Unlike the latitude we had with *AoE*, which had also come out a year late, our new deadlines for *AoK* were very firm and hung over us the entire time. The pressure was very much on.

What Went Right

We did what it took to make *AoK* a triple-A game. While the decisions to take an extra year and reset the units to an *AoE* baseline were tough in the short term, they were the right decisions to make. The commitment of Ensemble Studios to exceed the quality of its prior games never wavered. To realize our goals, we added the additional programmers, artists, and designers that we needed. When we needed to stop, take a hard assessment of what we were doing, and kill our own children if need be, we did just that. We pushed ourselves hard and we came together as a team.

1. Addressed the major criticisms of the first game

Despite *AoE*'s success and generally glowing reviews, there were two things about the game that were repeatedly criticized: the artificial intelligence of the computer players and the pathfinding and movement of units. And to be honest, they were right. Because these issues got so much press, we knew going in that if we didn't address them in a visible and obvious way, *AoK* was going to be raked over the coals by reviewers and users. It didn't matter that other popular RTS games had pathfinding that was just as bad, or that our AIs didn't cheat and theirs did — we weren't going be judged against them, but rather against ourselves.



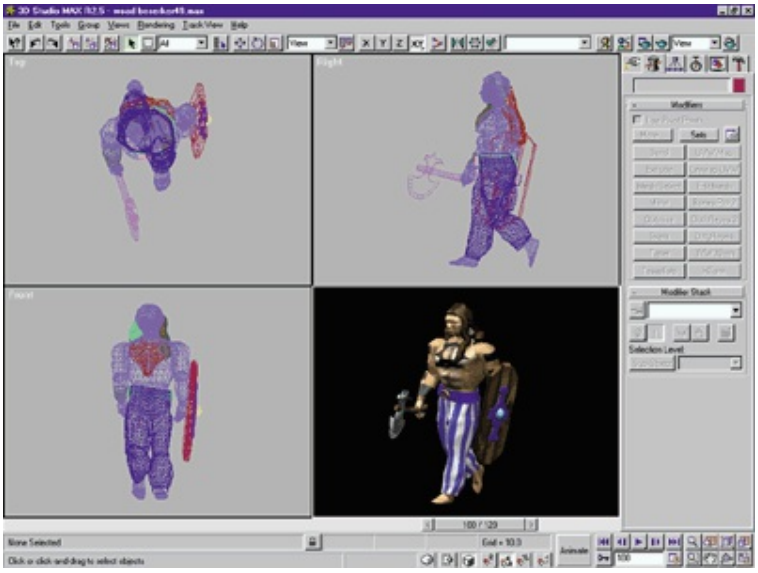
A 3D Studio Max model of the Mameluke character. Twenty thousand polygons got reduced down to several hundred pixels for the final game.

To handle the computer-player AI, we hired Mario Grimani, an industry veteran with significant AI experience under his belt. The computer-player AI from *AoE* was thrown out, and a new, expert-system, script-based AI was developed. While Grimani was doing the coding, Sandy Petersen led the design team in developing scripts for the new AI. Input from the whole company was encouraged, and various people contributed scripts that were pitted against each other in an evolutionary fashion to develop a computer player that could race a human player up through the ages and react to his tactics.

For the pathfinding problems, nothing less than an all-out blitz was ordered up. The game engine's movement system was redesigned and no fewer than three separate pathfinding and two obstruction systems were developed, requiring five different people working on them at various times. A high-level pathfinder computes general routes across the world map, ignoring such trivial things as people walking, which were handled by lower-level pathfinders that could thread a path through a closely packed group of units. In the end, we were so successful in ridding the movement problems that hampered *AoE* that reviewers and players couldn't help but take notice and acknowledge the improvement.

2. We innovated within the genre

While in the end *AoK* stayed much closer to its*AoE* roots than we had initially envisioned, we pushed the RTS gaming experience forward with a host of improvements. Some of these were interface-only improvements, such as the "Find Next Idle Villager" command, completely customizable hot-keys, and the extensive rollover help. Other improvements changed the game play itself, such as the Town Bell (ring it and all your villagers run inside the Town Center to defend it), in-game technology tree, and of course, Automatic Formations. One of the most praised features, Automatic Formations, caused a group of selected units to automatically arrange themselves logically by putting the strongest units up front and the ones needing protection in the rear. They stay in formation while traveling, replacing the "random horde" that players had become accustomed to in RTS games. Programmer Dave Pottinger originally set out to create a formation system incorporating characteristics of a turn-based war game's formation system, but as the game progressed and our understanding and vision for the game matured, a complicated formation system gave way to a simpler system that better served the game.



A 3D Studio Max model for a male villager.
For *AoK*, female villagers were also added.

When I wrote the graphics engine for *AoE*, I used a 166MHz Pentium at work and tested on my 486 at home. A 2MB video card was my target, but the game would run with only 1MB of video memory. Today I have a 32MB TNT2 card in my 500MHz Pentium III system. These changes in the typical game player's system are mirrored by the increase in player expectations for a great visual experience. In *AoK*, I'm proud to say, we met and exceeded game players expectations. The first thing that you notice upon playing *AoK* is the scale. The units in the game are about the same size, but the buildings and trees are no longer iconic. They are large structures with a scale that looks as if the units could comfortably reside inside them. Castles and Wonders are now gigantic, imposing structures that fill the screen. And the art itself is just so much better. Our entire art staff gained a great deal of experience and skill with *AoE* and *RoR*, and *AoK* became a showcase for their improved talent. It wasn't just the units and buildings, though. In *AoE*, the terrain had something of an Astroturf feel to it and the need to make transition tiles by hand limited the game to four terrain textures. For *AoK*, a whole new terrain system was developed, allowing us to mix terrains together, shade elevation in 3D, greatly increase the number of textures, and even alpha-blend textures such as water. The highest compliments came at the 1999 E3 show when we were unable to convince people from some of our biggest competitors that *AoK* was still a 256-color game.

3. Better use of bug tracking software and crunch-time management

During the development of *AoE*, we had a single machine in the office that would connect up to RAID, a remote bug database in Redmond, Wash., via an ISDN modem. This was used to handle bugs found by testers at Microsoft. Every so often someone would fire the connection up and, if the machine at the other end was in a good mood, make hard copies of new bug reports to pass around to people. We also had a different software package for communicating bugs and issues among ourselves, but there were not enough users on the license for everyone. Suffice it to say, this system left something to be desired, but it was all we had. During the development of *AoK*, ThinRAID was made available to us, allowing everyone to access the bug database directly from their web browser. Having only one system on everyone's desktop, available whenever needed, that was always up-to-date made a huge improvement in our ability to track bugs, stay on top of things, avoid redundancies, and just plain save time.

A bird's-eye view of the *AoK* game world. Compared to *AoE*, *AoK* has bigger worlds with more objects and richer graphics.

The last six months of development on *AoE* were pretty much one continuous blur of people working nonstop. This took a heavy toll on people, sometimes even straining their health or marriages. As a company, we vowed to not let things get that bad again. To further underscore the need, the composition of Ensemble Studios had shifted dramatically away from being mostly young, single men (with presumably no life) to being dominated by married men with a growing number of children and babies on the way. To protect ourselves, we scheduled crunch time well in advance at multiple points in the development process. The hours were 10 a.m. to midnight, Monday through Friday, with Wednesday nights ending at 7 p.m. so we could go home to our families. We had weekends off and meals were provided during the week. For the most part this worked very well, although having a "family night" where family members could join us for dinner once a week proved to be more of a distraction than we would have liked. Producer Harter Ryan deserves much credit for making crunch time so much easier on *AoK*.

4. Better use of tools and automated testing

After *AoE* was finished, we developed several in-house and in-game tools to make the job of development easier. The most-used tool was ArtDesk, a multi-purpose program that converted graphics from standard formats to our proprietary formats, which allowed us to view and analyze the content of our graphics data, and generated many of the custom data files for the game. This easy-to-use GUI-based program replaced several antiquated DOS command-line utilities and automated many tasks, saving a huge amount of time over the development span. In an effort led by Herb Marselas, programming tools such as Lint, BoundsChecker, and TrueTime were used to a degree never approached during *AoE*'s development and proved invaluable in improving the quality of our code. Finally, in-game utilities such the Unit Combat Comparison simulator allowed the designers to balance the game in a more scientific way. Every effort made in the tools area was rewarded with either time saved or significant improvements in the product. The only glaring omission in all this was the lack of an art asset

management tool.

5. We met our system requirements

A game that's expected to sell in the millions needs to be able to run on most of the computers it will encounter. Requiring cutting-edge systems or specific video cards won't work. With *AoK* being an 8-bit 2D game, meeting video card requirements wasn't going to be very difficult. But memory and processor-speed targets were another story. All the new systems in *AoK* would put their demands on the computer. Optimization issues were worked on hard for the last several months of development. The eleventh-hour addition of some clever tricks and a variable graphics-detail switch allowed us to hit our CPU target of a Pentium 166MHz, MMX-supported CPU. The minimum memory requirement of 32MB was also met, but with some reservations. Large multiplayer games on huge maps would need an extra eight or 16MB to be really playable. All in all though, the minimum system requirements for *AoK* are some of the lowest for games released in the Christmas 1999 season, widening the game's potential audience.

What Went Wrong

I'd like to say that we had fewer problems developing *AoK* than we did for *AoE*, but it didn't turn out that way. Some problems listed in the *AoE* postmortem were addressed in*AoK* and others weren't. And like that band going back into the studio to record a follow up to a hit album, we encountered a whole slew of brand new problems, many of which we found we were just as unprepared for as we were the first time around. I've tried to include some of the issues that became more important due to the fact that we were making a sequel to a successful game.

1.We still don't have a patch process

This was a problem area from the *AoE* Postmortem, and as of this writing it still has not been addressed. I outlined the reasons we needed a process to issue patches for our game in a timely manner in the *AoE* Postmortem. Additionally, a new reason reared its ugly head: cheating in multiplayer games. At first people found bugs in *AoE* and exploited them to win unfairly. Then it got even worse. Programs called "trainers" were developed that would actually modify the game's code while it was running to allow players to cheat.

The wireframe model for the trebuchet unit (right) and the fully textured and skinned version (left). The trebuchet proved to be one of the most popular units in the game.

Being the developer — not the publisher — of *AoE*, we don't have the final decision if or when a patch is to be released. As a result, all during 1999 our reputation as developers was assaulted by fans who saw us as uncaring about the problems that were driving people away from online play of our games. The topic of cheating in multiplayer games is so extensive I hope to do an article on it in the near future. We addressed this problem with our publisher and were promised a patch process. Unfortunately, *AoK* shipped with a couple bugs that seriously needed addressing in the short term. They're not show stoppers, but if not addressed soon, the game's (and our) reputation may suffer another black eye. If a patch for *AoK* is out by the time you read this, then you can conclude that we finally established our process.

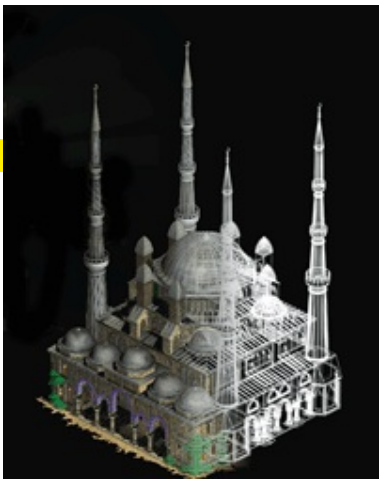
2. Unfinished versions of the game got out

This is a problem that is born of success. Prerelease versions of nearly all games wind up circulating in pirate channels known as "warez." This happened with *AoE*. Imagine our surprise at reading an entire review of the game (an alpha version) eight months before it was released. Fortunately, almost no one bothered with it until the game was properly released because nobody knew much about it. *AoK* was a completely different story. It was a highly anticipated sequel to a very successful game, and the various warez sites were tripping all over themselves to get a copy of the latest build. And get a copy they did. They were usually only one or two weeks behind our latest build. It seemed as though copies were leaking out from every imaginable source — play-testers at Microsoft, previews sent to magazines, even internal sources. Unfortunately, positively identifying and fingering the culprits was almost impossible. There were hacking attempts on our FTP server and network, though the real rub came from the pirates in Hong Kong and Singapore. They took the warez versions of *AoK*, burned them onto CDs, added some cover art, and sold the game throughout the Pacific Rim. In Korea, the CD vendors operating in front of Microsoft's headquarters had a warez version of *AoK* for sale. Warez versions were even turning up on eBay. Though we doubt bottom-line sales were hurt much, our pride certainly suffered. Any of our future games will probably require connecting to a secure server of our own design to operate, even for single-player games.

3. Play-testing had a lot of problems

This item is a catchall for several problems that we encountered. On the good news front, our new offices have a dedicated play-test area equipped with identically configured machines. The bad news is that we didn't make the best of it.

Many of our play-tests were not organized and focused enough, seriously reducing the amount of new and meaningful feedback obtained. It wasn't always clear when we were testing for specific bugs and issues, and when we were testing for "fun." We had a schedule of participants which drew upon the whole company, but schedule conflicts and lax enforcement resulted in the same people playing most of the games. We played too much multiplayer and not enough attention was given to the single-player game. And some people took it much too seriously, trash-talking other players, celebrating wins at the loser's expense and storming off when they were losing. Play-test problems weren't confined to Ensemble, though. At Microsoft, it was discovered that a play-tester had turned cheats on, playing to win not to test, in almost every game for over a month, which invalidated all the feedback from that group for the prior two months.



A Turkish mosque shows off the greater detail and improved skills of our art staff.

4. Art asset management was nonexistent

The number of individual frames of graphics in *AoK* is in the tens of thousands, and we didn't do a good job managing it. The programmers had a source-control system to help coordinate their primary output of code and the designers had the game's database system, but no such equivalent existed for the game's art assets. Artists could be working on something with no idea that anyone else was also working on it. There was no way to get a momentary snapshot of who was working on what, other than going around from office to office. Plus, there was no way to tell which files were actually live and being used and which ones were just taking up space. Also missing was a way to go back and find prior versions of art, or to guarantee that new versions wouldn't be overwritten. As we have grown as a company, this problem has grown even faster. To address this problem in the future, a source-control similar to the art asset management system is being developed for use in all future projects.

5. Problems with third-party APIs and software

Another one of the items from the *AoE* postmortem returns again. Microsoft's DirectPlay API still has a number of issues that make it less than perfect. One of its biggest problems is documentation and testing of the lesser-used portions of the API, or rather the lack thereof. Late in the development of *AoK*, our communications programmer, Paul Bettner, was able to communicate with the DirectPlay developers and an interesting scenario played out several times: Paul would attempt to solve some problem and the developers would indicate that it wouldn't work because of bugs in DirectPlay that they knew about but that were not documented.

A scene from one of the game scenarios. The updated graphics engine and building scale allowed us to create scenes much more impressive than in AoE.

DirectPlay wasn't the only problem. We decided to use DirectShow to handle our cinematics. The short version of this story is that it just didn't work. And then there was the Zone software for Microsoft's online Gaming Zone. The Zone software was developed too late in the process and had a number of problems, due to a lack of time to test and correct. Unfortunately, this means that direct TCP/IP games are more reliable than those played over the Zone, which is disappointing. This was not all the Zone's fault because we did not get our requirements to them soon enough.

The Show Goes On

One of the touchiest and most personal issues concerned letting success go to our heads. The success of *AoE* is something that a lot of people in this business have not experienced. It exceeded our wildest dreams and allowed our company to take charge of our destiny. I

remember when we got our first *AoE* royalty check — I had never held a multi-million dollar check before. That was great. We all got caught up in how good we were doing. Over time an attitude of invincibility set in. With a success like *AoE*, it's easy to forget what it was like to wonder if we were going to be in business the next year. At some of the industry events such as the Game Developers Conference and E3, some of our people behaved in ways that embarrassed us. With success comes a responsibility to behave appropriately — the game industry is a small and incestuous one, and nothing lasts forever. Behaving in an exemplary manner and being friends with the industry at large is far more important than chest-beating about our current success. Suffice it to say that people in the Ensemble Studios organization have stepped forward to address this and we have challenged ourselves to be better people.

All the early indications for *AoK* are that it's going to be a blockbuster on the order of its predecessor, and maybe even greater. The reviews from the press have been unbelievably positive. According to PC Data, *AoK* was the number-one selling game in October.

The great success of *AoE* made it possible for us to go to the next level of making great games. Though it enabled us to grow and acquire greater resources, it also raised expectations for our next game and spawned a host of new challenges. Meeting these new expectations has proved to be just as tough and rewarding a journey as creating the first game. In the end we succeeded in creating a game to be proud of, and I feel privileged to have been part of it.

Matt Pritchard is busy trying to be a modern renaissance man. When not working, he can be found with his family or playing with his collection of antique video games and computers. He can be reached at mpritchard@ensemblestudios.com.

The Age of Empires II: Age of Kings development team: Top row, from left to right: Jeff Goodsill (COO), Brad Crow (art lead), Brian Hehmann (artist), Angelo Laudon (lead programmer), Sandy Petersen (designer), Dave Pottinger (programmer), Ian Fisher (designer), Harter Ryan (producer), Duncan McKissick (artist), Trey Taylor (programmer), Mario Grimani (programmer), Paul Bettner (programmer), Chris Van Doren (artist), Jeff Dotson (artist), John Evanson (programmer), Doug Brucks (programmer), Roy Rabey (IS support), Paul Slusser (artist), Chea O'Neill (artist), Bob Wallace (strategic), Mike McCart (webmaster).

Bottom row: Rob Fermier (programmer), Nellie Sherman (logistics), Stephen Rippy (music), Herb Marselas (programmer), Mark Terrano (lead designer), Chris Rippy (sound), Herb Ellwood (artist), Thonny Namounglo (artist), Duane Santos (artist), David Lewis (programmer), Sean Wolf (artist), Bruce Shelley (lead designer), Matt Pritchard (programmer), Brian Moon (CFO), Tony Goodman (CEO), Don Gagen (artist), Greg Street (designer). Not pictured: Tim Deen (programmer) Brian Sullivan (strategic), Chad Walker (artist), Eric Walker (artist), Scott Winsett (lead artist).

Age of Empires II: Age of Kings

Ensemble Studios
Dallas, Tex.
(214) 378-6868

http://www.ensemblestudios.com

Release date: October 1999

Intended platform: Windows 95/98/NT

Project length: 24 months

Team size: 40

Critical development hardware: Pentium II 450 128MB, Dual Xeon 450 512MB

Critical development software: Visual C++, 3D Studio Max

Return to the full version of this article