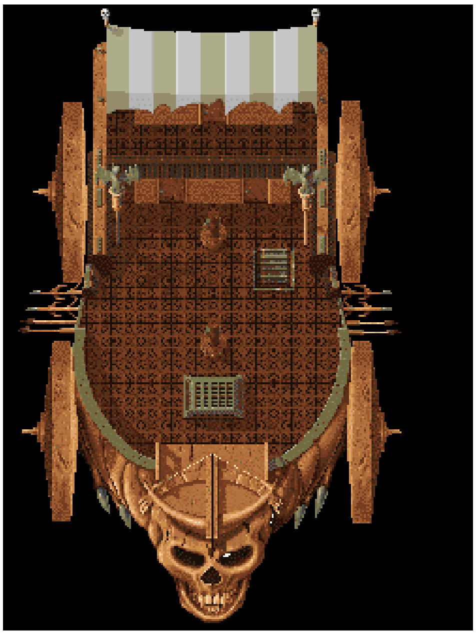
GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: SSI's *Dark Sun Online: Crimson Sands*

By Andre Vrignaud

The initial design goals for DARK SUN ONLINE (also known as DSO) were quite simple. We were tasked to take the DARK SUN II single-player code base and turn it into a large-scale multiplayer online game for AT&T's now-defunct Interchange network. The game was initially planned around a client/server architecture (although it didn't end up that way) to protect against hacking, to support thousands of simultaneous players, and to have plenty of quests and global events to keep players entertained. However, while designing the project, we had several considerations to keep in mind:



First, our resources were quite limited, especially when it came to art and audio. We had one part-time artist available to create the few odd objects that we absolutely had to have; otherwise, we had to make do with art we could reuse and steal from other internal projects. We had the same limitations with sound and music resources and often had to go to extraordinary lengths to find quality material.

Second, we knew we had to use the existing code base and couldn't rewrite the game from scratch. This obviously tied us down with a great deal of baggage, not the least of which was that we were using a dated game engine, both graphically and in terms of the game's code. Because of this we chose not to focus on the graphical aspects of the game, which would have been futile, but dedicated all of our resources toward creating a multiplayer game with strong game play elements.

Finally, we had to coordinate all of our efforts with an external programming group and contract art group. As you'll see, there were definite challenges with meshing the internal game design and the external programmers. In addition, we had to go through many iterations of art creation for some of the new characters and monsters. This proved to be an entertaining experience in its own right.

The origins of DARK SUN ONLINE can be traced back to the summer of 1994 when representatives of AT&T's Interchange network approached SSI to do a new ADVANCED DUNGEONS AND DRAGONS online game. At that time, AT&T was in the process of planning its own proprietary online service, and believed that having a strong game presence would be of great help in launching the network. Having seen the success of NEVERWINTER NIGHTS on America OnLine, the company wished to fund the development of a sequel based on the DARK SUN engine for its exclusive online use.

HUMAN
Centuries of abusive magic have marred the purity of humanity, resulting in a wide variety of humankind. Some might call this an unfortunate side effect, others might call it retribution.

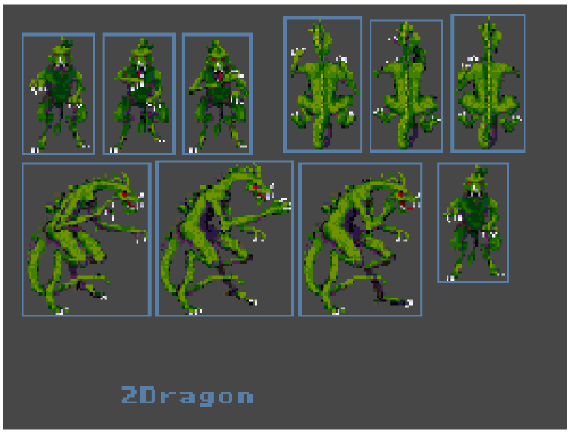
Our first challenges became apparent in October of that year as the contract was signed and work began. Due to some overly aggressive estimates as to the ease of porting DARK SUN II to an online environment, the project was basically underbid from the very beginning. This had many ramifications, beginning with a lack of art and sound resources to draw upon, and continuing with difficulties keeping external team members' morale up.

Thanks to these resource constraints, we had to go to great lengths to get the material needed to create the game. To begin with, we raided the source code archives and reused art from both DARK SUN I and DARK SUN II. This turned out to be more difficult than it sounds, since there was a slight perspective shift from the first game to the second, and much of the early art simply didn't look right in the new DARK SUN II perspective. We also borrowed a great deal of art from an SSI title called AL-QADIM, which had an Arabian Nights theme. Although much of the art was unusable for the same perspective reasons, some fit perfectly. Sand is sand, after all, no matter what angle you view it from!

For the entire duration of the project, we only had a half-time artist to do some miscellaneous objects, interim cut-scenes, and cinematics. Since there was always a huge crunch going on with other internal projects, we also had to deal with losing that artist at the oddest (and most frustrating) times - especially since there was enough work to keep several artists occupied for months. One of the biggest effects that this situation had upon the original game's design was the loss of all of the game's cinematics, apart from the introduction. (Ironically, even this introductory cinematic was only available for a short time in the DARK SUN ONLINE 1.0 retail version. Later versions of the game were never re-released at the retail level, and the introductory cinematic scene was too large to be included in the downloadable versions.)

Additional art work that needed to be accomplished for this project included the customization of hundreds of player icons, some perspective tweaks on DARK SUN I monsters, and the creation of several new monsters. Since we effectively had no internal resources to create this art, we turned to an external art house.

The character portraits went relatively well, with only a few redos on dental-floss bikini bottoms and overly-endowed female characters. The Dark Sun I monsters were also tweaked with little problem. Unfortunately, the new monsters turned out to be far more challenging for the external artists. Two monsters in particular - the Dragon and the Nightmare Beast - were somewhat less than fearsome. As you can see from the sample art, the Dragon (even after some rework) looked like a gecko, and the Nightmare Beast… well, the Nightmare Beast just looked like Barney. Perhaps luckily, the Dragon encounter ended up being cut from the game, and since we never could get Barney looking right, his appearance was cut, too. (There were those who thought keeping Barney in could be quite a cathartic experience to certain twisted people, but fortunately saner minds prevailed.)



The "gecko-like" Dragon

We did much better as far as sound effects went. Digging in the archives, we were able to draw upon the sound effects from Dark Sun I and II, which pretty much covered most of what we needed.

However, we were still a little short in some areas and cast about trying to find a few more additional effects. Lo-and-behold, the sound department coincidentally sent out a large archive of .WAV files from the soon-to-be released THUNDERSCAPE title for use as Windows event sounds. Much to the chagrin of some on the THUNDERSCAPE team, those effects quickly found their way into DSO.

Using Contracted Developers

Beyond all of the internal resource issues, DARK SUN ONLINE was also an interesting project due to its use of external programming talent for the online coding and porting of the game. At the time, SSI had no internal online programming talent available to code a multiplayer version of DSO. So the online coding was subcontracted out to an external programming group. Now, working with external groups is pretty common in the industry, and normally isn't that big of a problem. Unfortunately, due to some political and financial issues, the contract was finally signed (over a few objections) with the external group getting paid on a strict milestone basis, with no royalty points or interest in the finished title. As you can imagine, this became a bad morale issue for the group.

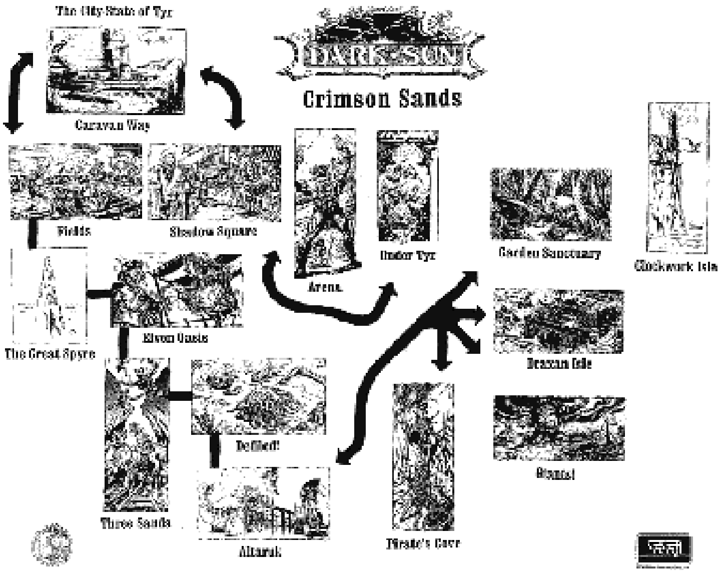
Much of the art from DSO was reused from earlier games

This morale problem reared its ugly head quite quickly as the external programmers and internal scriptors began to interact. Earlier DARK SUN games (other than the art work) were created as a joint effort between the programmers and the scriptors (also known as the designers). The programmers would work on the game engine, adding features such as spells, new monster effects, and new GPL (Game Programming Language) commands for the scriptors' use. Meanwhile, the scriptors would use GPL to code the actual adventure parts of the game - the events, quests, and NPCs (nonplayer characters). The key to this arrangement working harmoniously was that most of the GPL-interpreting code (in the engine itself) was already done, allowing the scriptors to work relatively autonomously. Unfortunately, this wasn't the case with DSO.

The big problem with this project was that the external programmers had to take a code base for a game that was designed as a single-player game and make it a multiplayer game. Unfortunately, the original programmers (on DARK SUN I AND II) made assumptions about the way certain GPL commands would work, in order to save time on coding error-checking routines. For example, the original programmers never had to deal with the possibility of two different people trying to talk to the same NPC at the same time; it simply wasn't possible in a single-player game. However, this situation is extremely common in the online version. The result of this was that the external programming group had to recode the way GPL

"Barney," the Nightmare Beast

worked - many, many times as the game's development progressed and we learned more about how the engine would really work.



Sysop tools were added that allowed the DSO world to expand beyond its original size and let players discover new challenges.

Rich Donnelly, the lead scriptor for DARK SUN ONLINE says, "Oftentimes, I would write script for the engine, and the engine would change halfway through, in such a manner that the script would no longer function. Even worse, there were times when crucial information was lost in the associating chain, such as new GPL functions or commands. This caused a lot of strife and recoding on everyone's part."

To sum up, since the external programming group had no buy-in or incentive to make the project the best it could be, they generally took the easier path whenever possible. The most grievous example of this was the networking change from the planned client/server architecture to a peer-to-peer system. Due to low morale and compensation, the external group insisted on coding the networking architecture as a peer-to-peer network, using the local clients to run most of the game logic. Although we raised the red flag multiple times about problems inherent with this scheme, our team was overruled in the interests of keeping costs low. To be fair, the peer-to-peer coding was done competently for what it was. Unfortunately, though, this type of networking code wasn't the best for the game at large, and we had to deal with the hacking issues that it raised for the life of the project.

The Debugging Process

A few months passed, and eventually the programming group got the engine to the point where the scriptors could do some GPL coding without having to restart every month. At this point, the scriptors dug in and started coding the underlying global routines for the game, and the external programmers started modifying the game engine to work over an IPX network, which would to allow us to test and develop internally. Within a few weeks, we had that IPX-capable version of the engine to work with - which raised new issues.

Chronologically, this was soon after DOOM was released, and if you'll recall, the early versions of DOOM had problems with packet flooding, which brought down quite a few corporate networks. This was quickly fixed in subsequent releases, but the internal management at SSI was still a little gun-shy, and asked us to set up a completely separate network for our development work. This in turn meant we needed a UNIX box to run the game's server code, which meant we needed to set up and configure a Linux box (Linux is a free variant of UNIX) to act as the host. To make a long story short, configuring a UNIX machine isn't an easy task, and by the time we got the hardware, built the network, and configured the host, we had lost a few weeks of development time.

The greatest hit that our original timeline took resulted from the initial plan to run DARK SUN ONLINE as a DOS application in a DOS box, but communicate with a Windows-based TCP/IP stack. AT&T's Interchange online service was a Windows 3.1 application and ran over a proprietary TCP/IP stack that AT&T had licensed. Since the original DARK SUN games were DOS applications, and since Windows 95 hadn't yet penetrated the market sufficiently, it was thought best to keep DARK SUN ONLINE as a DOS application. We would depend on a "shim" of sorts to allow the game to communicate with the Windows protocol stack. We spent a couple of painful months trying to get this going, and eventually had something that worked, but not well. The shim would occasionally lock up for seemingly no reason, and we were unable to get the source code for the protocol stack to try to fix the bug. Eventually, we scrapped that effort and ported DSO to a true Windows application. In retrospect, we probably should have bit the bullet early on and moved the game to a Win32 application from the start.

After this initial setback, work progressed for a few months without any particular disasters to note. Art from the external group was slowly coming together, and we resolved most of our sound and music issues. The external programming group was hitting its milestones, and all in all, everything was quiet. Yes, as many of you just guessed, too quiet.

In early November of 1995, I received a call from the head of Interchange's game group. It turned out that AT&T had pulled one of their classic turn-arounds and decided that it didn't actually want to be in the online network business after all. In short, AT&T's Interchange network was canceled, along with all of the projects being funded for it - including DARK SUN ONLINE.

At first, it didn't look too bad. Although we didn't have a network to launch the game on, we had gotten a great deal of the project funded, and through a nice quirk in the contract, all rights to the game reverted back to SSI. This allowed us to shop the title around to different networks and possibly negotiate a better royalty deal. Unfortunately, this process took several months, and since SSI's internal management wasn't confident that they'd be able to find somewhere to place the title, the project was placed on a back burner. A couple of the scriptors and I continued development, along with the external programmers, but little other work was done. However, a few months later, it was decided that TEN was the appropriate network for the title, and work began again in earnest in January of 1996 - with one other little bump.

In mid-January, one of the three internal scriptors decided to move on to new opportunities. This normally wouldn't have been that big a deal - things were moving along well, and the two remaining scriptors would simply have had to pick up the remaining work and take care of it in the extra months we'd allocate for them. However, the scriptor that left was responsible for a great deal of the global game code upon which the game, along with much of the other scriptors' work, depended. Sadly enough, it turned out that much of that work had either not been done, or had been done so poorly that the code couldn't be depended upon. It fell to Rich Donnelly to pick up the pieces.

"It was quite horrific for me to find that the game was missing some serious pieces of code. These were pieces that were vital to an online environment, such as global region transfer code that handled moving entire parties instead of individuals, or monster generation routines for general combat encounters that could incorporate several different players in the same combat. Suddenly, I found myself with little time to correct these problems. I worked heavily for about a month, and managed to finish getting everything implemented and working."

The Beta Test

The extra months added to our schedule after AT&T folder their service were quite useful, as they gave us some extra time to prepare for the external beta test. Although we had a core group of internal testers on the project, we knew that we'd need to run a large-scale external test to really shake out all of the bugs. To that end, we prepared a web page that gave instructions on how to sign-up for the beta test - and were quickly inundated with responses.

We originally planned to solicit beta testers and mail out a CD-ROM to everyone who responded. Unfortunately, so many people responded that we couldn't follow through and were only able to send out around 500 discs; however, the rest of the testers were able to download a 30MB PKZipped version of the game.

The beta period in particular proved to be quite an educational experience, and we definitely learned some lessons from it. In fact, it's probably worth taking a bit of space here to list some of them.

LESSON ONE. You can never allocate too much time for beta testing and debugging an online title. The sheer number of bugs found by the external testers was absolutely amazing - and the potential for each of those bugs to cause havoc was greatly magnified by the interactive nature of the game. Allocate significantly more time than you first think for your testing period - and double it. Seriously.

LESSON TWO. Be sure to have robust facilities to deal with the flood of bugs. I highly recommend setting up some method for external testers to enter their bugs on a web page, and a method to import those bugs into whatever bug database you use. We had the bug submission page, but no way of automatically moving those reports into our database. In fact, due to the kludged nature of our bug-tracking system, we ended up having each and every bug sent to my office e-mail account. I would then log in every morning and run a mailbox filter, forwarding all of the bugs to my lead tester to sort, compile, and enter into our internal bug system. I don't think he's forgiven me to this day.

LESSON THREE. Have a FAQ. Period. Every morning I would spend hours reading and answering hundreds of messages. Some were suggestions, some were flames, but most were simple questions - and we quickly realized that we were seeing a great deal of them over and over. However, once I got the FAQ written and published, the flood relented… a little. Beyond that, though, the FAQ turned out to be a great promotional and educational tool; it even staved off our marketing department a few times by proactively giving them whatever information they were looking for. Really, now - can you ask for a better reason to do a FAQ than that?

LESSON FOUR. Don't hype the product until it's ready. We purposely kept quiet about the title until very late in the project and were thus mostly able to handle peoples' expectations. (I say "mostly" because there are some people you'll never be able to satisfy.) We wince now when we see the hype that's been built around ULTIMA ONLINE. Although UO is likely to be an impressive and groundbreaking product once it's finished, it's also unlikely to ever satisfy the heightened expectations that have been built up around it.

LESSON FIVE. Make it extremely clear that it's a beta-test. Then do it again. And again… and yes, yet again. Even so, I guarantee you people will threaten, flame, and otherwise get extremely mad at you when changes are made. As Donnelly says, "Players are not concerned with the fact that the game may not be in its final state - if you change anything, there will be complaints, regardless. During our beta test, we altered the way the environment worked several times and wiped the host quite often to insure that the test was started afresh, with no corruption. Naturally, players screamed at us every time this happened. In addition, watch what you show the players during testing periods, as people aren't thinking about what the project will look like in the future; they care about what it looks like now. I recall that ULTIMA ONLINE got quite a negative reputation initially when Origin showed their alpha, as players immediately assumed that was what the game was going to be like. Beta testing includes more marketing than most people realize."

LESSON SIX. Hackers will exploit the slightest loophole, and it's often that exploitation that ruins the game for all of the other players. DSO was particularly hackable due to its odd peer-to-peer architecture - an unfortunate legacy of the external programming group doing things the easier way for them. Again, if you're doing to do a large-scale RPG on the 'net, don't consider building it around anything but a client/server architecture - and make sure the server is the arbitrator of all key game logic.

After a long and painful beta period, DARK SUN ONLINE was finally launched live on TEN - and as dictated by Murphy, the inevitable bugs popped up. DSO version 1.1 was released a few months later to address most of those bugs, and DSO version 1.5 came along a few months later as a game expansion. Discussions are underway as to the possibility of doing a DSO 2.0 - and as the player base continues to grow, I'm sure we'll see the game expand and evolve even more in the future.

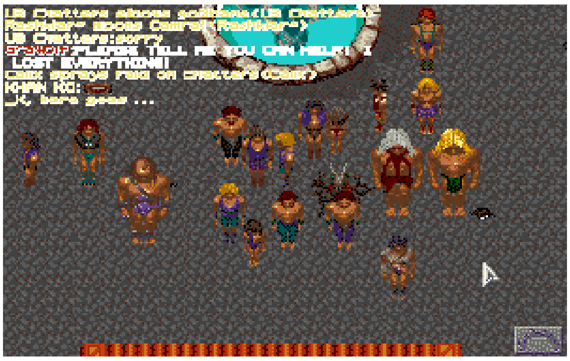
Modifications Along the Way

DARK SUN ONLINE differed from its original design goals in a number of ways. First, the peer-to-peer architecture we ended up with limited us to hundreds of simultaneous players instead of the planned thousands. That same peer-to-peer architecture also made the game extremely hackable, as much of the game logic was processed on the local machine instead of the host.

Second, a great deal of new sysop tools were added due to beta tester feedback. Those tools, in turn, helped keep the game viable by allowing staff members to run hand-made events while the scripted quests were being fixed. In addition, extra regions were added to give the players more room, and even more regions were added for version 1.5.

Third, due to the easily-hackable architecture of the game, players were able to cheat and escape routines to punish character death. Version 1.5 of DSO removed those penalties until we could move the game logic to the server - hopefully in a future version of the game.

Finally, all cinematics were cut except for the introduction for version 1.0. That introduction cinematic was also cut in post-1.0 versions, due to its being too large to realistically download. In addition, since the game was never re-released at the retail level after version 1.0, the Red Book music became unavailable. However, it's possible that the original DARK SUN I MIDI tracks might be reintroduced in future versions of the game.



UB Chatters elbows godbane<UB Chatters>
Rashklar~ kloons Camral<Rashklar~>
UB Chatters:sorry
Graywolf:PLEASE TELL ME YOU CAN HELP!! I
LOST EVERYTHING!!
Calix sprays Raid on Chatters<Calix>
KHAN KO:
_K, bare gear ...

Despite these changes, however, DARK SUN ONLINE stayed with most of its original design goals. The game play was our focus, not the graphics. Some of this was simply because of the original art that we were forced to use, but a great deal of it was due to beta testers' suggestions during the testing period.

Player interaction and communication was the focus of the entertainment, and not prescripted elements as in the original DARK SUN games. We also reused legacy art to great effect (albeit cheating like dogs the whole while to steal decent).

Finally, the online interface added a great deal of functionality, while remaining inspired by the original DARK SUN games and thus familiar to players' of those titles. The chat system in particular was quite powerful and worked out well.

I'd like to highlight a few things that went particularly well during the development of DARK SUN ONLINE… and a few things that didn't go so well. Some of them have already been touched on above, but there are a few others also worth mentioning.

What Went Right

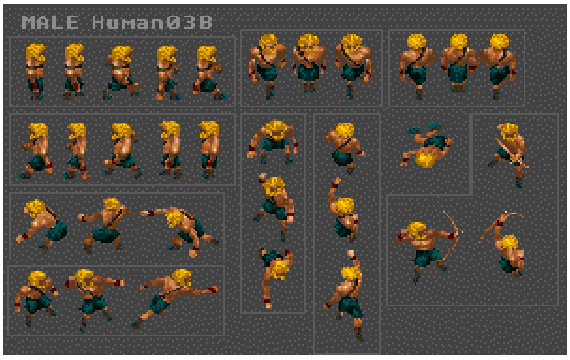
1. A close feedback loop with the external testers proved to be incredibly helpful. Other than the obvious benefit of learning about bugs and interface problems, we also got a good deal of free publicity and goodwill. That goodwill was particularly useful when we made the inevitable screw-ups. In addition, it was beta-tester feedback that spurred development of the FAQ.

2. The chat interface turned out to be far more intuitive and user-friendly than we ever thought it might. Rich Donnelly says it best. "The chat interface is probably one of the most dynamic and user-friendly chat interfaces to be seen in any online role-playing game. In almost every case, there are multiple ways to do the same task, something that almost every user enjoys. As a designer, I have found that players adopt their own style of play, regardless of how the interface or game environment is designed. Having the ability to perform tasks in a variety of different ways allows the player to find the method of play that is most comfortable to them. Look at Windows 95, for example. Some people prefer to work from their Desktop, others like it nice and clean and prefer to use the Start menu and Explorer to find the items that they're seeking. It is this versatility that people desire, and including it in your product is essential."

3. The reuse of DARK SUN I and II, and AL-QADIM art assets proved to be an effective decision. Although many on our team would have liked to improve the existing art, or create a great deal of new art done, we were able to reuse most of the art from DS II, some from DS I, and a little from the external project AL-QADIM to good effect.

4. Our online role-playing paradigms were well thought out, and some are finding their way into other games. We learned that you can't allow a real, permanent character to die if a player was paying for it. We discouraged and punished people to keep them from cheating and skipping out of combat. DSO also allowed and encouraged player vs. player combat - an element we see being supported in more and more online role-playing games. Finally, we implemented the concept of instant communication across the online world, no matter where you were. Purists seem to disagree and dislike the lack of reality, but those purists also forget that chatting is the single most important community support tool you have. Nothing in your game design should ever discourage communication among players.

5. Although the basis of the game is player interaction, we found that having a strong random quest generator helped fill in the gaps. To quote Rich, "Having a system that can essentially generate quests on its own is something that definitely increases the entertainment value of the game. The quest generation system currently in the game is rudimentary at best, but it's definitely a solid model and a step in the right direction for what might be termed a 'story-telling engine.' As is the case with all developers, my only regret is that I didn't have the time to take this quest engine as far as we would have liked. However, the model as it stands is a serious piece of machinery, and something to be proud of."



What Went Wrong

1. We tried to make a DOS application running in a DOS window communicate to a Windows 3.1 TCP/IP stack. Instead, we should have simply ported DSO to a Win32 application and developed the game with the Windows 95 TCP/IP stack - as ended up being done months later.

2. SSI's internal resources were severely limited for the project, which in turn made us far too dependent on external resources. On top of that, the contractors should have been better incented to provide quality work. In particular, the external programming group was competent, but not financially-motivated to do the job "right." In addition, the external art group needed a great deal of hand-holding, and at the end of the day we still didn't quite get everything we wanted.

3. One of the scriptors left in the middle of the project and didn't do a great deal of the work that was necessary for the game's underlying routines. We ended up having the two remaining scriptors doing the work of three, with one of them having to redo all of the key global routines.

4. Not enough test time was allocated for debugging such a large-scale online game. This forced us to move extremely quickly which, in turn, frustrated players due to the speed and severity of the changes. Although many of these issues were unavoidable, having more time to prepare and a more flexible deadline would have allowed us to be more accommodating to testers' issues and frustrations.

5. Probably the biggest issues we had during the development of DARK SUN ONLINE were the hacking problems. These problems came as a result of taking a stand-alone game and turning it into a multiplayer game with a peer-to-peer networking system. Game logic then ran locally on the client, rather than on the host. Rich Donnelly has a little more insight on this, and

explains it thusly: "The game environment itself, having been converted from a stand-alone product to an online product, left the game with most of the logic running on the user side. The host itself merely transfers the information back and forth between all the players, keeping everyone in a huge loop of data sets. This causes some serious problems with hacking, as people could use editors on their local machines and change critical data. "Hacking thus became quite a nightmare for DARK SUN. Hacking a stand-alone product is no big deal - change the game all you like, nobody is going to complain. In an online environment, you are affecting not only yourself, but also everyone else playing the game. It's kind of like a movie theater; you purchase a ticket for one seat, but you don't buy the whole theater. When you begin screaming, the ushers escort you out, as you are disturbing others. It's the same for an online environment - it just takes one bad apple for everyone else to feel the effects."

And On the Seventh Day...

All in all, the development team was quite proud of how DARK SUN ONLINE turned out. It's one of the few large-scale graphical multiplayer online role-playing games now in existence, and currently generates tens of thousands of hours of paid use every month. However, we've also learned a great deal during the creation of the game; some have been listed above since they were key issues during development, but I'd like to take a moment to list a few more miscellaneous lessons. We hope they help you in the development of your title!

- People want their name in lights, so the more recognition you give game players, the more they'll feed back into the system. No player wants to be the anonymous shopkeeper - but they'll be a lot happier about it if they get a big lit-up sign with their name on it! Players want recognition for their acts, be they good or evil. The more you let individuals stamp their name on the world, the more they come back - both to see their name in lights, and then to keep it there. The one caveat I'd add is that you have to be cautious when giving recognition for "evil" deeds such as player killing, thieving, and so on. Eventually some players live only for the recognition they get for harassing other players - who eventually get fed up and move on.

- Resources allocated to the project must remain accessible during the entire beta period, until the title is finally launched. In addition, if you're serious about online games, you should expect to keep at least some portion of the team on to support and expand the game in the future. I realize this latter point sounds particularly obvious, but we've seen quite a few times when this didn't happen and it had a detrimental effect on the game. (Yes, DSO was one instance of this happening.) I'm not really sure why resources get pulled at this point by companies, but I assume it has something to do with people underestimating the amount of work and support an online title requires.

- The people who designed and built the game should also be the team that supports it after its launch. These are the people who have the understanding, the tools, and most importantly, the passion. Although it is possible for an external group to take over the support for such a title, you definitely take a bit of a hit as they get up to speed. On top of that, the new support team often fails to get much support with bugs since the game company doesn't "see" them. As you may have guessed, this exact situation happened during the development of DSO. SSI was unable to dedicate resources to support the game once launched, so TEN took it over. The problem was that SSI often had no empathy for TEN when confronted with the bugs and support issues that needed attention. Sadly, SSI had become too far removed by taking themselves out of the loop.

- Checks and Balances to control player killing (Pkilling) are absolutely mandatory. Although we personally believe that allowing players to prey upon each other is a worthy and admirable goal, you have to have checks to keep the balance of power from tipping completely toward the Pkillers. Once that happens, you get into an ugly situation with Pkillers consistently preying on newbies, which in turn dramatically affects the popularity of your game. All it takes is one negative first experience to lose a potential new player forever. We dealt with this issue in several ways. The first was to simply create an arbitrary non-combat zone in the area newbies first appear in the game. These new players also got warnings (tied into a character level check) when they tried to leave that safe area. Finally, we put the highest priority on fixing bugs that allowed players to hack the game and attack people within the safe zones.

- The last lesson I'd like to leave you with is that a retail version of an online-only title is a tough sell to consumers, especially if you're also going to charge an additional fee to play online. We attempted to add a few extras to the retail version of DSO, including Red Book music, cinematics, and a printed manual. It turned out that only a few die-hards bought the retail package, and I personally suspect that many of those were people had been deeply involved with our other game, NEVERWINTER NIGHTS, and wanted a "collectable" for this sequel of sorts.

  As another data point on this, Meridian 59 was originally released as a retail product only, with a street price of around $40 (which included some amount of free online playtime). Within a couple of months 3DO changed their policy and also allowed users to download the title from their web site. Origin's ULTIMA ONLINE is a retail-only product that sells at a premium price point ($64.95), and also charges for online time. I suspect this title might be one of the few that actually pulls the retail play off, but I also wouldn't be surprised if EA does some dramatic pricing shifts a few months into the game's release.

I hope this article has given you a few things to think about, and helps you in the development of your title. I'm always interested in discussing gaming and the online industry in general, so if you have any comments or questions, please feel free to drop me a note.

André Vrignaud has worked in the computer gaming industry since 1990. Most recently, he's worked at SSI and TEN, where he produced much of their online content and direction. He wishes thank everyone involved with DSO, with special thanks to the Lead Scriptor/Designer Richard Donnelly, TEN's RPG Producers Alex Beltramo and Don Hupp, and in particular, all of the thousands of external beta-testers. He can be reached at andre@null.net.

Return to the full version of this article