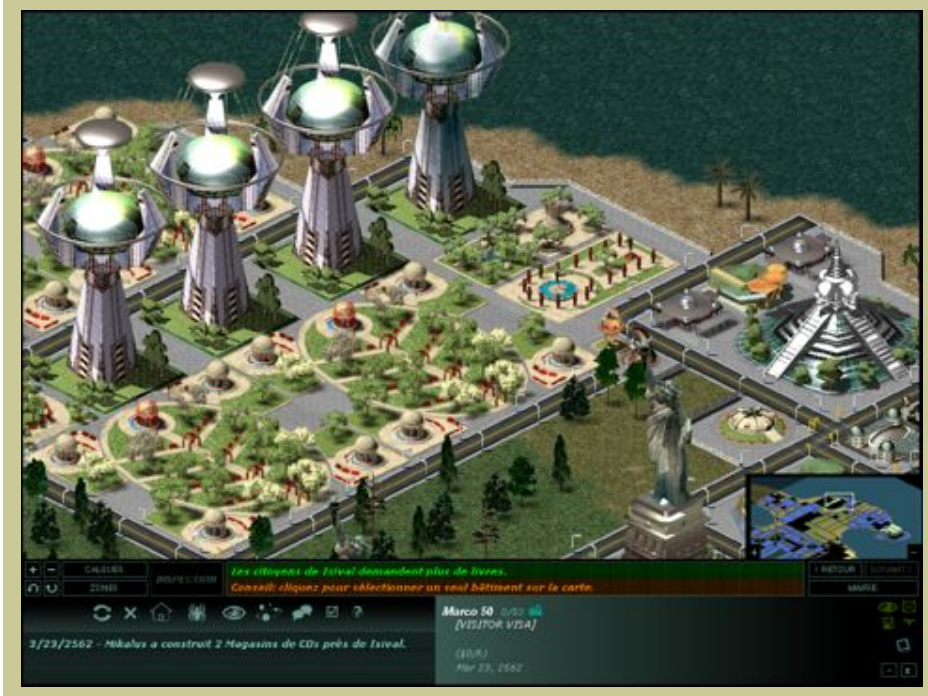
Postmortem: Oceanus Communications' *Legacy Online*

By Marco Cultrera,Hipolito Iroel Perez

Legacy Online's story started more than six years ago in an unlikely place: Cuba. It was here that a group of Cuban students from the University of Santa Clara (not to be confused with Santa Clara University in California) embarked upon a crazy dream. While on blasting away at each other playing games on a LAN, they decided to put their technical skills to use and try to make a living developing videogames.

From the beginning, the goal was to create a MMOG, something only a handful of companies in the world had managed to do at the time. Since Cuba's Internet connection then consisted of 56k modem for the entire island, you can appreciate how ambitious this project was.

While the initial development work was underway (working on a virtual Internet simulated through the LAN of their university), the group managed to pitch their idea to a Canadian working with a professor at their university and Oceanus Communications was born. The team started traveling back and forth from Cuba to Canada, testing the fruits of their work on the real Internet, until a stable contingent of developers was established in Ottawa and the real development work could start. The Canadian government's Pilot Visa Project for skilled IT workers greatly facilitated this process.



Legacy Online* was originally called *Star Peace.

Originally called *Star Peace*, the game was an ambitious undertaking. The idea was to bring together two of the most successful genres in the videogame industry: city building and strategy games. The goal was to create an all-encompassing experience that pit hundreds of players against each other in a unique battlefield.

The starting point was to create a realistic economic simulator in which players would participate as interstellar tycoons, growing their business by building factories, retail outlets, warehouses other buildings necessary to develop and sustain a thriving society on new planets. These planets would start off empty, and would slowly be populated by simulated citizens immigrating from Earth.

To foster player interaction and a realistic environment, a political component was added to the mix. Players could be elected as Mayors of cities and Presidents of planets. This also provided players with another goal: apart from becoming immensely rich and developing a planet, they could choose to leave behind a legacy of providing good leadership, government and city planning.

The potential audience was huge. We felt many of the fans of city-building games would love the concept and embrace the opportunity to test their skills against real players. Another key innovation was the idea of a MMOG that wasn't a role-playing game (at the time that was the only MMOG genre). Interestingly, even today 90% of the MMOGs on the market are RPGs.

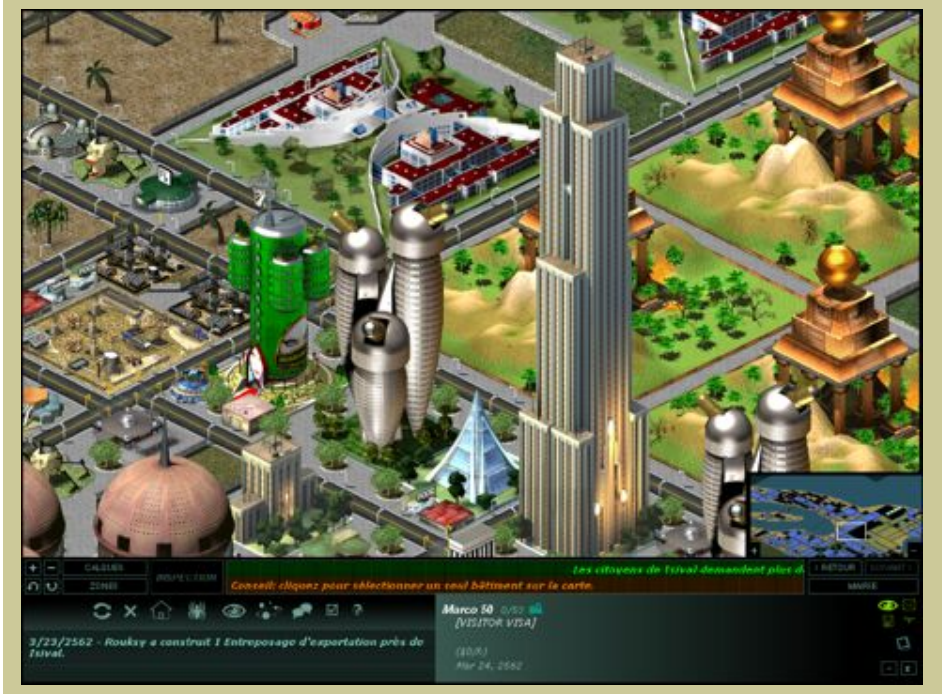
Legacy Online launched on June 16th this year, but as an online game, it is constantly a work in progress. We're continually building add-ons and tweaking the game, but the bulk of the work is complete and we can take a look at the challenges we've faced so far.

What Went Right

1. Choosing to use a web interface embedded in the native code and lightweight client. In designing the game's user interface and client, we had two specific goals in mind: create a lightweight client that could be easily downloaded, and allow it to access game data on the website in the same format as players see it in the game. We ultimately chose to use a skeleton of native code embedded within web-based components constructed from HTML and ASP.

In a data-heavy game like *Legacy Online*, it's important for players to access key information easily, so that an unexpected turn of the economy doesn't bring down your accounts. To facilitate this, at any moment a player can go to www.legacyonline.net, and with just a few clicks, monitor their account and make sure everything is OK without actually connecting to the game. The player profile page displayed on the website is exactly the same as a player sees in the real game.

This solution gives players an instant snapshot of how they are performing in the game. All planet rankings, covering every possible aspect of the economy, are computed in real time. On the website you can review rankings planet by planet, or consult the general ranking across all of the planets. Not many MMOGs can, at any given time, show who is the "best player" and fuel competition in such a direct fashion.



A modern capitalist empire competing against others on the planet -- and on other planets.

Notwithstanding some early criticisms from beta testers about slow loading web pages (prior to launch, testers reported that slower Internet connections caused some web page display problems), this strategy has proven to be a winning one. Relying on a proven platform like HTML and ASP has simplified code maintenance and debugging. Add the increased speed and reliability of the Internet, and this approach has contributed to making *Legacy* a practically flawless gaming experience -- even with a slow connection. Moreover, updating the web-based part of the client is very easy, and large changes can be made and new features added without the need to deploy new code to the users.

From a business perspective, *Legacy* can take advantage of online distribution due to its lightweight client. Nobody likes to wait hours (even on a broadband connection) to try a game.

2. The Directory server. One of the main components of a MMOG is the database that stores the game information. When we were developing *Legacy Online* we found that using traditional database applications (with a set of tables and store procedures with fixed functionality to access the database) was not a good idea. We needed the ability to access and modify data in our application, but those requirements couldn't be met at the time using "Views" and "Stored Procedures" on a fixed data model.

The idea was to make the game as extensible as possible, and ultimately we achieved that. Had we taken a traditional approach, we would have had to constantly make changes to our database application, which we wanted to avoid. Instead, we came up with the idea of implementing the database application as a huge registry file, not very different from the Windows registry.

The registry is a very flexible way to store information. In Windows it cannot grow too much because there is a space limit to contend with. But by implementing a registry-like layer over a SQL Server database, we could store practically unlimited amounts of information. There is

another factor prompted this decision: we did not plan to do database-like queries on the game information. Information in our game is accessed using an absolute path in the directory. For example to find a user profile, we would locate it using a path instead of using a key in the database -- in other words, table.route/users/p/peter would be the key to find all the values related to Peter's profile.

The Directory Server also plays a major role in the game's server farm -- it is the center all the servers because it handles all the servers' configuration data. The Directory Server also stores a lot of information that is shared by the game's other servers. This information can be access concurrently. *Legacy Online* servers depend on each other, and it's the Directory Server that coordinates their interactions. Another nice thing about this approach is that we can move a server from one computer to another, or replace a server experiencing problems. Before a server is taken offline, it can store its entire configuration on other servers until it's up and running on the network again. That permits smooth hardware modifications.

Thanks to this approach, we could connect the *Legacy Online* website directly to the Directory server, saving us a lot of time maintaining and developing the website. The game information that is of interest to the player community is rendered to the Directory and the website pulls that information in automatically. This data includes player profiles, game rankings (across all the categories), the different worlds' status and stats, in-game news, and so on.

3. Low bandwidth requirements and good response with high latency users. For an MMOG, one of the major issues to deal with is the bandwidth requirement per player. Once there are thousands of players connected to the game, it can crumble if players use more bandwidth than the game can support. Another issue to resolve was what to do with users with poor connections, and those with very high latencies.

To implement *Legacy Online*, we carefully decided how information would flow between the client and the servers, and among the servers. We studied many of the available distributed-object computing standards at the time, and while many were good candidates (for example DCOM and CORBA), in the end we decided to create our own object request broker (ORB) because we wanted to control every byte that was transmitted.

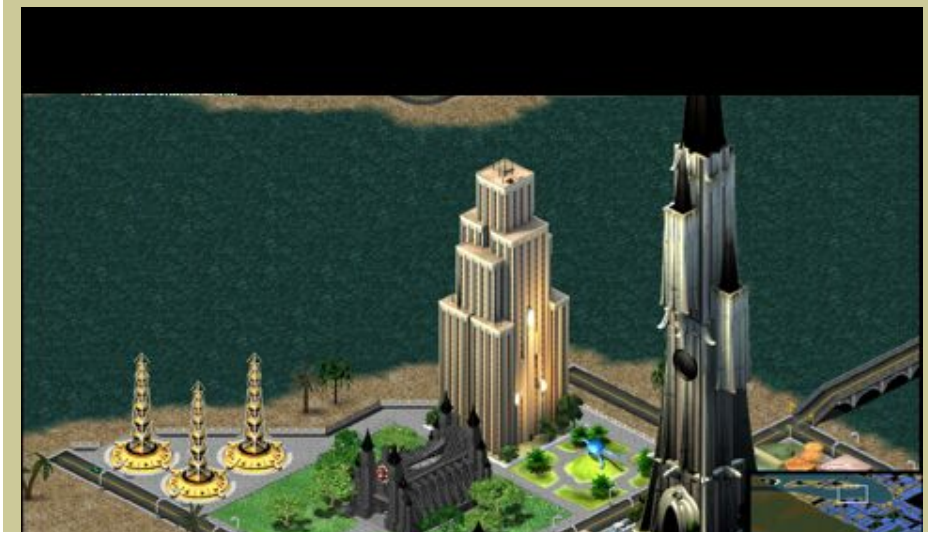
In particular, we needed to have special control over disconnections, timeouts and error handling. We created an ORB named RemDO (Remote Delphi Object). This protocol was specifically designed for Internet applications developed with Borland Delphi, because that was our main development platform. RemDO is also available through OLE automation. Having our homemade protocol enabled us to optimize our servers' functionality, interoperation and communication with game clients.

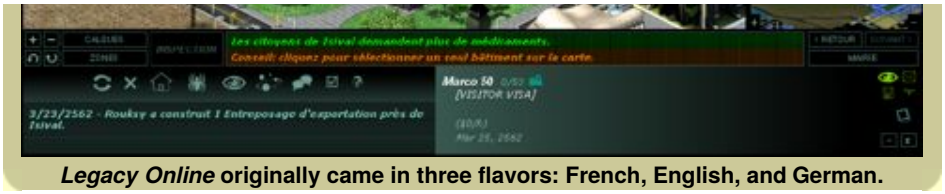
Another technique we used to minimize the transmission of information was to implement several layers of cache, in the servers and in the game client. The game client is in part web-based, so we were very careful to exploit the use of the Internet Explorer cache. We do not abuse the use of images, but the images we have are immediately cached after visiting the game's different interface pages. Nearly 60% of the game's interface is implemented using simple web pages (ASP pages), effectively reducing traffic.

Legacy Online follows a classic client/server approach so users with very high latency do not affect the functionality of other players. The game has proven to be playable even on computers with very slow connections. Our current lowest Internet connection requirement for players is a 33Kbps modem.

Another factor that works in favor of players with high latencies is the game design. Because the game is a massively multiplayer strategy game, there is no shooting or actions that require an immediate response. So latency is not a big problem. Most of a player's actions take effect a few seconds later, so the game still behaves well even with latencies as high as two seconds.

4. Multilanguage support. When *Legacy* was first released as *Star Peace* by Montecristo Games, part of the publishing agreement entailed providing the game in three different languages: French, English and German.





***Legacy Online* originally came in three flavors: French, English, and German.**

In a very text-oriented game like *Legacy*, achieving this wasn't straightforward. We proceeded to strip the embedded English terms shown in the game from all the code and develop a special component that would automatically find any string in the game and save it separately into a text file. We divided the language files into three categories: Server, Client and Web pages, and we created a tool that put all the text to be translated into a single file that was given to the translator. The same tool would reverse the process, deploying the translated text back into the game.

When the agreement with Montecristo came to an end, we continued running and updating the game, but only in English, as we were unable to provide support for French and German players. Later we added Spanish (the native tongue of many of our developers) and Italian (since for a brief period the *Legacy* client was offered for free with the Italian version of *PC Gamer* magazine).

Adding support for Spanish and Italian was made easy by the same translation utility that we originally built to support French and German. But a new challenge came up a few months ago when we had to support French again, to take full advantage of the Telefilm Canada New Media Fund that *Legacy* qualified for. (Telefilm Canada is a Canadian cultural agency that supports the development and promotion of Canadian new media industries.) Since the moment we dropped French from the game, the text hadn't been updated. English strings were used creating a hybrid version of the client, mostly in French but with a significant part of English strings popping up here and there. We got around this problem by updating the tool we were using to integrate the missing French.

It's now very easy for us to add a new language to *Legacy* and we can use the same tool to offer multilanguage support on any other game Oceanus will develop in the future. In addition, thanks to this built-in multilanguage structure, *Legacy* can also easily be converted into non-western script with a minimal amount of development work.

5. Easy modification of the game to add new features and sub-games. In any online game, a significant part of the appeal is the constant flow of new expansions and features the user can expect from the developers. In a game like *Legacy Online*, given the amount of details and depth, it was natural to set up a system in which small adds-on and tweaking of the economic parameters could be easily implemented.

In the four-year history of *Legacy*, a constant flux of changes and new features have been added, varying from very small tune-ups to full-fledged new lines of business and buildings types. Every time this has happened, players experienced minimal downloading wait or none at all, if the changes were related to the web interface of the game.

In terms of full-fledged sub-games, we have some currently in development that will dramatically expand the *Legacy Online* universe -- so much so that they could be stand-alone games. Once again, these additions merely entail the deployment of a small update that our client automatically downloads.

Given the nature of *Legacy*, where players sometimes have to wait to accumulate money to advance to the next level, we've decided to add the possibility of adding independent sub-games with which players could kill time. At this moment no games have been added, but the supporting code is already in place and we are working with Sega.com about the addition of these games. (There's nothing like playing a good-old 2D side-scrolling Sonic game while waiting for those dollars to pile up!)

What Went Wrong

1. Premature release of the game under of the pressure of the first publisher (Montecristo Games).

When Montecristo games agreed to publish *Legacy* (then known as *Star Peace*) in 2000, we immediately started the process of transforming the game from beta to gold. From the beginning it was clear that there was substantial work to be done, and even though we provided Montecristo with estimates of the time it would take, the release date was nevertheless pushed up on us.

Eager to launch the game as soon as possible, a very ambitious launch date was set, preceded by an open beta that only lasted one month. We warned the publisher that more time was necessary for extensive testing, but they decided to go forward with releasing it anyway.

Naturally, the launch was a disaster. The game suffered prolonged periods of downtime, with our programmers struggling behind the scenes to patch problems overnight. In hindsight one could say the launch was doomed; we were inexperienced when it came to dealing with a publisher, and Montecristo was inexperienced in the online space (*Star Peace* was the first online game they launched). Let's just say it wasn't a match made in heaven.

After a few months of mayhem, our collaboration with Montecristo ended. We brought the servers back to Ottawa (they had been in London, England) and we took over the game. The resulting improvement in service was immediate. Players refer to that period as the "Dark Ages".

2. The use of a cache system based on files instead of a database. As a strategy game, the main duty of *Legacy Online*'s servers is to simulate the game's virtual worlds. The CPU is a sacred resource that the game simulation servers have to make the most of. But what

happens when the players want to see what is going on? If the server running the virtual world had to respond to every single request made by players, the simulation would slow down far too much.

To deal with this issue it was necessary to implement a cache system on the backend of the server application to hold copies of most of the objects in the simulation (not all of them -- just those of interest for game play). This cache is only updated when it becomes too old as compared to the one on the server. We used the well-known idea of the TTL--Time To Leave.

The problem was that we implemented the cache system using the Windows File System. Windows File System is fast and somewhat reliable, but for the type of caching we do, a database was the better option. However, at the time databases were too slow so we had to be innovative.

We have since found that databases are faster than the file and cache system. If we were to start over, we would probably have implemented the cache using a database, possibly embedded in the same database as the game's directory server for performance reasons. Yet with constant optimizations, we have made the current cache system more than adequate for the game's needs.

3. Choosing Delphi 5 for client/server programming. One of the most unusual things about *Legacy Online* is the language we used to develop the servers and the client. In the game's early design and implementation stages we spent a lot of time discussing which language and platform to use. At the time, Java was getting lots of hype, and while we were very fond of the language, after some initial testing we decided it was not the best choice.

The original team that began work on *Star Peace* had extensive experience with Object Pascal (the language used in Borland Delphi 3.0), which at the time was superior in terms of features when compared to most programming languages on the market -- including C++ and Java.

The other good thing about Borland Delphi as a platform was its easy and fast development cycle, which was great for prototyping applications. Delphi initially facilitated the development of our homemade ORB, RemDO.

At that time, to achieve the same results in C++ would have likely taken the team three times as long. Also in Delphi's favor was the fact that the game used language features that C++ does not even offer today, like RTTI (Run Time Typing Information) and Meta Classes. However, the effort to adapt the design to a C++ implementation would have permitted us to port the game server to other platforms like Unix or Linux (it currently runs on Windows NT/2000). That would open up our hosting choices for the game, considering that many of these operating systems do a better job of using the server's system resources.

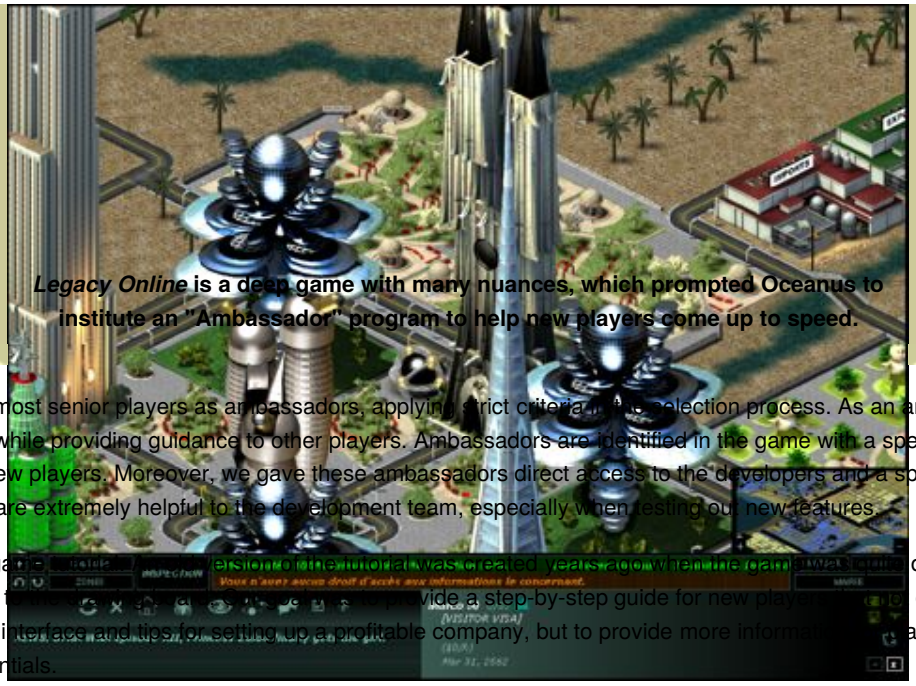
Using C++ also would have allowed us optimize more code for the servers and provide a faster simulation. Borland lacks experience in terms of code generation and optimization when compared to its competitors in the market. When considering the innovative things we've seen in other languages, if we had the chance to do it all again, we'd use another language. No offense, Borland!

4. Learning curve for players was higher than expected. Given the fact that *Legacy Online* is a very deep game with many nuances, it takes new players substantial time to become experts. From the beginning, the steep learning curve became evident, and even with an easy and intuitive graphic interface, we lost a lot of players who were scared off by the game's complexity.

Potential publishers were often concerned that the game's complexity would negatively impact newbies. Consequently, we created different tools to help new players understand the game. For instance, it's made clear to all newbies before they start that they should expect to go bankrupt more than once as they begin to understand the economy in the game -- and that those bankruptcies would not penalize them moving forward.

Another element instrumental in helping solve this problem is the player community itself, particular our older players. These experienced players go out of their way to help new players get started. Every time someone new joined the game, they could always count on someone helping them out.

When Sega.com came aboard, it was clear that we couldn't simply lean on the good will of some altruistic players, so we decided to start the "Ambassador" program and create a new tutorial.



***Legacy Online* is a deep game with many nuances, which prompted Oceanus to institute an "Ambassador" program to help new players come up to speed.**

We appointed 70 of the most senior players as ambassadors, applying strict criteria in the selection process. As an ambassador, a player receives in-game perks while providing guidance to other players. Ambassadors are identified in the game with a special icon so that they are easily recognizable by new players. Moreover, we gave these ambassadors direct access to the developers and a special in-game forum. Such seasoned players are extremely helpful to the development team, especially when testing out new features.

We also created a new ga[illegible]ersion of the tutorial was created years ago when the game was [illegible] different, so we decided to scratch it and go back t[illegible] provide a step-by-step guide for new players. [illegible] only presented a quick explanation of the game interface and tips for setting up a profitable company, but to provide more information [illegible] ayers that wanted to know more than the bare essentials.

The tutorial uses a web-based component, with a system of calls from the server that alerts the tutorial if the player has completed every single step. At the same time, a system of links allows a player interested in more information to easily access it.

Both solutions have proven successful: the game's learning curve is now less than half an hour, the time to complete the tutorial, and the ambassador system effectively institutionalizes the interactive aspects of the game we seek to promote. The tutorial was not only praised by many players, but noted in recent reviews as well.

5. Insufficient funding, Visas issues, other workplace problems. As noted at the beginning this article, *Legacy Online* is was made possible thanks to Canada's Pilot Visa Project for foreign IT workers. This enabled the team to set up shop in Ottawa. Ottawa is not a hub for game developers in Canada, so that presented some unique challenges (Even today, fellow Canadians are shocked to hear that we are located in Ottawa -- Ottawa is synonymous with telecommunications, not game development.)

However, once we relocated to Canada, the team faced the challenge every start-up deals with: big dreams but not much money. It hasn't been easy bringing *Legacy* to the point it is now. We needed to assign some of the team members to outside consulting work in order to generate revenue. At times, some key developers of *Star Peace* had to be deployed for long periods of time in these side jobs, because it was particularly suitable to their skills.

For eight months in 2001, only one person was managing *Star Peace*, dealing with all issues from programming new features to game mastering and troubleshooting. At the time, players knew the team was small -- but they didn't know just how small. Despite that "one man band" period in our history, the game ran reasonably well and customer service was excellent. The only evident limitation was that features couldn't be added as quickly as players would have liked.

On To New MMOGs…

Developing *Legacy Online* has been a great ride. It was the first game for all of us, so we expected to learn a great deal as we went along. Fortunately, at the end of the day we managed to develop a product that made all the struggles worthwhile.

With all that we've learned professionally and personally, we are now confident in our company's ability to develop new titles, particularly MMOGs. The MMOG space is getting more crowded, but few companies out there have the same track record and knowledge as Oceanus.

Game Data

Legacy Online
www.legacyonline.net

Number of full-time developers: 5 programmers and 3 graphic designers.
Budget: Nearly $2 million (Canadian dollars).
Length of development: 3 years.
Project size: Nearly 800,000 lines of Object Pascal (Delphi) code plus another 100,000 lines of ASP and HTML code and 10,000 lines of C++.
Launched: First released as *Star Peace* on December 1st, 2000.

Platform: The game runs on Windows 2000 servers. The client runs on Windows 9x, ME, 2000 and XP.
Developer hardware: Pentiums III and IV workstations, 250MB of RAM, 20GB of HD space. No special video cards were used.
Developer software: In the beginning the team used Windows NT Server and Windows NT 4.0. We later upgraded to Windows 2000 Server and Professional for the development and test of the game. We used Borland Delphi 3.0 and 5.0 to compile the code. The 3D modeling was done with 3D Studio Max version 3.5 and 4. The 2D processing of the images was done in Corel PhotoPaint 8 and 10.

Return to the full version of this article