

## Postmortem: Stardock's Galactic Civilizations

By Brad Wardell

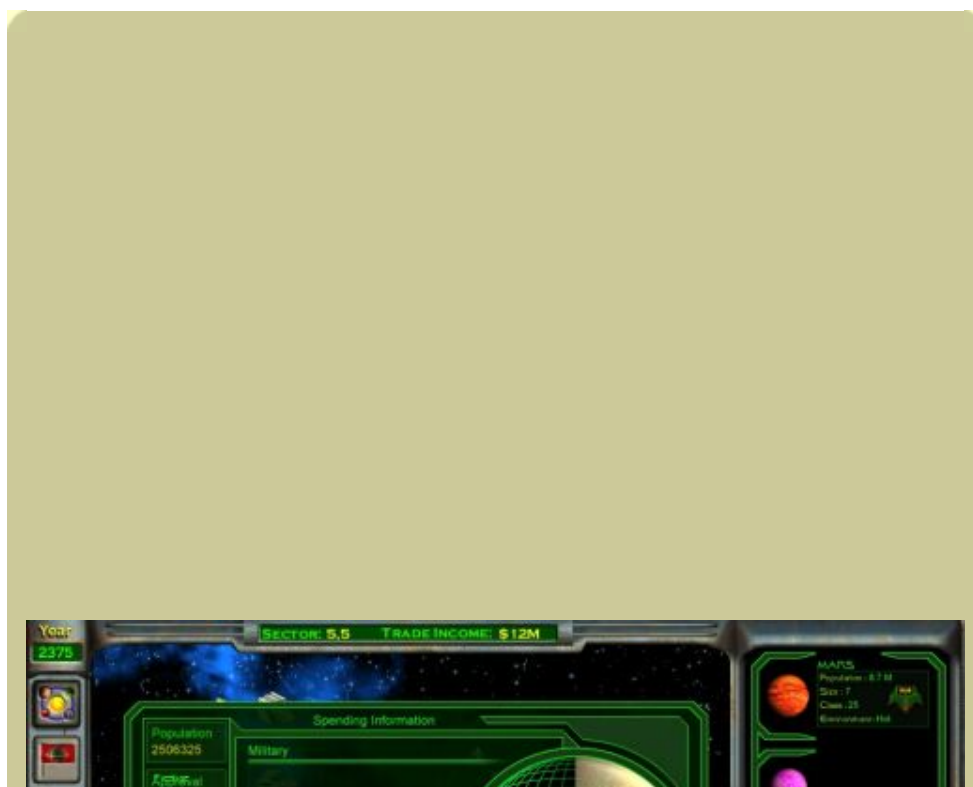
Our company, Stardock, was founded in 1993. We had our first real hit in 1994 with the release of *Galactic Civilizations* for OS/2, a space-based strategy game set near the dawn of the 23rd century. The game developed a cult following when we first released it, and most people who had heard of the game knew of its reputation for good AI. (Coincidentally, the third issue of *Game Developer* magazine featured an article I wrote about the game's multithreaded AI).

For years I'd wanted to go back and redo *Galactic Civilizations* with a significant budget. The original OS/2 version had been developed largely in my dorm room at Western Michigan University. Since then, Stardock has developed into a real company. A pretty impressive game development team had been formed during the development of *Entrepreneur*, *LightWeight Ninja* and *The Corporate Machine*. While those games were relatively small releases, they enabled us to accumulate talent and experience that we could apply towards a remake of *Galactic Civilizations*. Essentially, the *Galactic Civilizations* team was a combination of the teams that developed *LightWeight Ninja* and *The Corporate Machine*.

In the Fall of 2001, we began working on *Galactic Civilizations*. The core team included three developers and two graphics designers. We were also able to occasionally "borrow" a few developers from the Object Desktop team -- a Windows utility that our company also develops.

### Building A Virtual Community

One early development decision we made was to leverage Stardock's information technology strengths. You see, even though Stardock started its commercial existence as a game developer, we are best known for Object Desktop (<http://www.objectdesktop.com/>), a suite of desktop enhancements. To support this product, we run a site called WinCustomize.com (<http://www.wincustomize.com/>), a site for downloading Windows skins, themes and icons. To create WinCustomize, we had to develop an immense infrastructure to handle hundreds of thousands of user accounts, tracking of what they had submitted, their statistics, access levels, and so forth. Each day the site receives tens of thousands of unique visitors who can individually manage their skins and themes and compare their popularity to the other thousands of users who are doing the same thing.





**Early User Interface concept.**

In short, one of Stardock's biggest advantages that it could bring to *Galactic Civilizations* was the company's ability to build and manage large virtual communities. If we could find a way to bring this to *Galactic Civilizations*, we could make up for our lack of mainstream awareness with a rewarding gaming community for strategy gamers. This is where Stardock Central came into play (<http://www.galciv.com/sdcentral.html>). Stardock Central, which ships with *Galactic Civilizations*, integrates into our company's customer database and can see which products a the user has access to and allows them to download updates. We developed this system because we update our software products very often (multiple times per week), and we needed a tool to get these updates to users seamlessly.

#### The Word-of-Mouth Strategy

In the game industry, you sell lots of copies of your game in one of two ways. The first way is used by the "big guys": get millions of copies of your game on every retail shelf in the first 30 days. Within 90 days, those titles are often gone. We at Stardock knew that we didn't have the kind of clout to go that route, so we took the other path. We recognized that as a smaller developer, we would likely get little pre-release coverage. But we felt that if we created a virtual community and provided extensive free updates to the game after release, we could make up for the marketing deficiency with strong word-of-mouth marketing from players. This could help the game stay on store shelves longer, which in turn would increase overall sales at retail.





Early concept of the Altarians

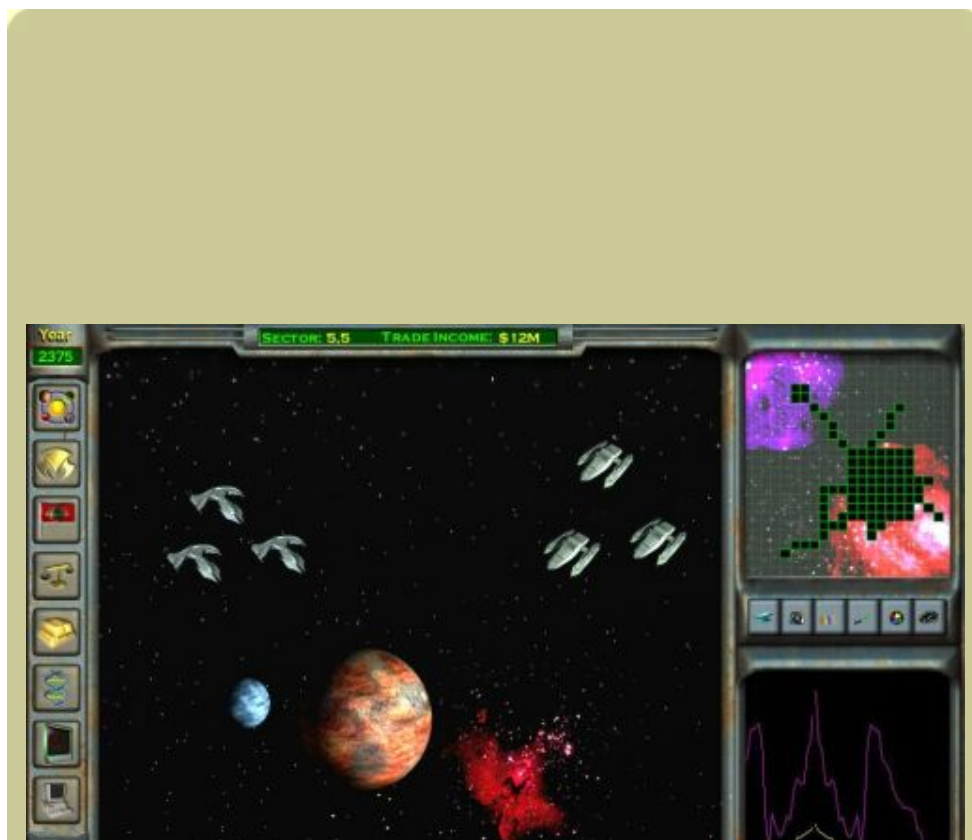
Since its release, *Galactic Civilizations* has received favorable reviews, but only has 1/12th the buy-in of a similar space strategy game that was released in the previous month. So our challenge is to keep the game "fresh and new" for as long as we can so that the game remains at retail as long as possible.

### What Went Right

**1. Reserving funds.** This was probably the riskiest part of our project strategy. We reserved funds to continue work on the game after it was released. This was particularly challenging in our case because Stardock funded the game entirely on its own. When we took the concept for the game to publishers, they universally passed on it since it was often perceived that turn-based strategy games were on the way out. So the game ended up self-funded with part of the budget reserved for after-release.

In order to do that, you have to remain on schedule. You can't release a game that isn't finished because the game magazines review what's in the box, not what the company puts into subsequent game patches, and you'll suffer at the hands of reviewers. On the other hand, if your development schedule falls behind, you eat into your after-release budget. Fortunately, our risk paid off big time -- the reviews have been favorable.

The second part of the risk in reserving funds was seeing how people would react to the idea of getting a bunch of rapid and significant updates after release. Would they view it as "they didn't finish the game?" Or that the game was "buggy"? So far, this has turned out well for us.





Another early concept of the user interface.

All of these updates, which were based on player feedback and released shortly after we received players' suggestions, have helped build a loyal player community. These updates aren't trivial, either: about 15,000 lines of new code have been added to *Galactic Civilizations* since it was released, along with several megabytes of additional graphics. To put this another way, a considerable amount of money was spent after the game's release even though the game had already won two Editor's Choice Awards from *Computer Games Magazine* and *Computer Gaming World*, and *PC Gamer* said it's better than all the other turn-based strategy games currently out combined. I apologize if this sounds like hubris, but it was just such a massive risk for us. Imagine if the game had received lukewarm reviews, or even been panned outright? All of these updates would have looked like we were just finishing the game, rather than adding significant free support to it.

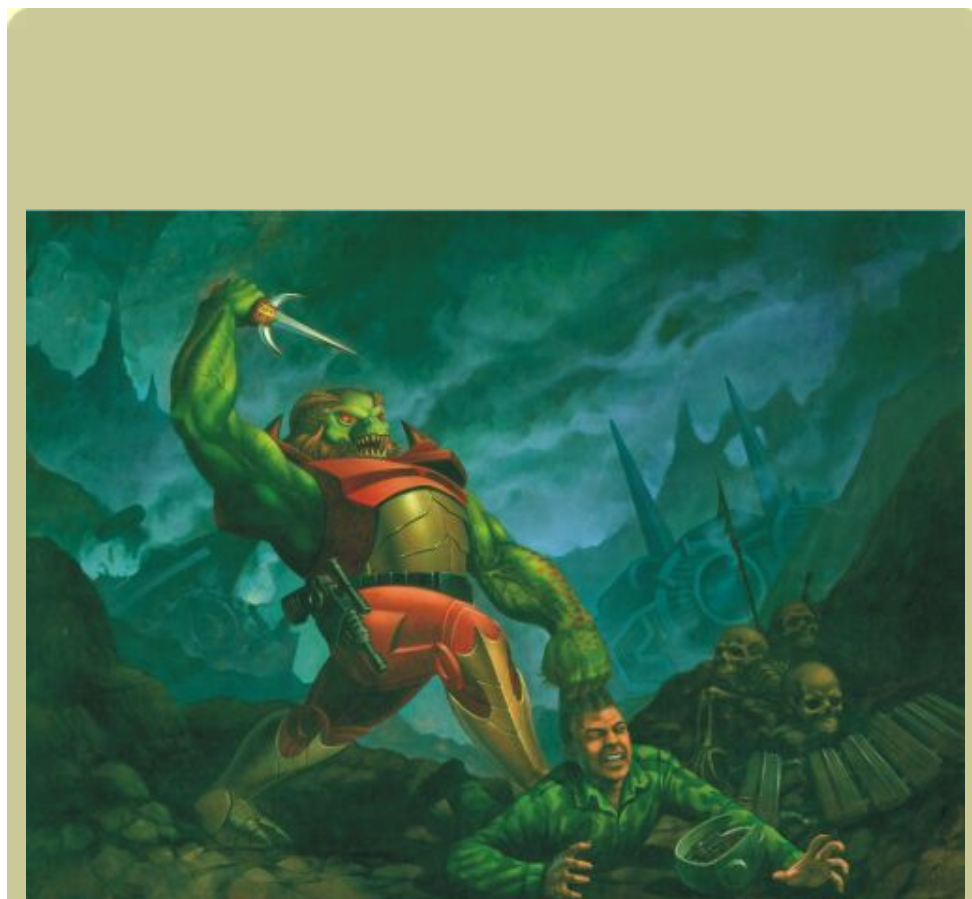
Our strategy really rested on the assumption that the game, out of the box, would be a complete, successful and solid experience so that our updates would be seen as enhancements, not bug fixes. Only then would the word of mouth from happy customers likely reach the level to make up for our generally under-the-radar marketing (it was hard to get significant coverage of our space-based, turn-based strategy game, with *Master of Orion III* taking much of that limelight already).

**2. Bonus development time.** We finished the game on January 20, 2003, but our release date changed at the last second when our publisher, Strategy First, made a great deal with Infogrames. Infogrames' *Master of Orion III* had been delayed and rescheduled to come out at the same time as *Galactic Civilizations*, so we agreed to push our date back one month -- a deal that helped both of our games. In addition, Infogrames was kind enough to put a flier for *Galactic Civilizations* into every box of *MOO3*.

This additional month proved to be crucial for us. Our art team was able to add in a series of "mini-cut scenes" that added a lot of flavor to the game, and the additional time allowed us to polish other parts of the game. It also allowed us to have our "BonusPak" come out on the day of the game's release, rather than a month after release.

**3. Our unproven art team.** About 99% of the artwork in *Galactic Civilizations* was done by two people. That includes about ten minutes of cut scenes (other than the opening intro) and the sound effects for the cut scenes.

What's really amazing is that our main animator in *Galactic Civilizations* started working on the game without any prior 3D animation experience. He was a recent graduate from Sheridan, and while very talented, he had to learn how to model and animate using 3ds max within a 16-month time frame. I think anyone who sees the cut scenes and other individual pieces of artwork will be pretty impressed, even by AAA standards - and it's almost all done by two guys.





**4. The AI.** The computer AI in *Galactic Civilizations* has been singled out repeatedly in game reviews as exceptional. That's a key indicator that things went right is because strategy games that are rushed out the door usually suffer from inadequate AI.

Things went a lot better in *Galactic Civilizations'* AI than we had thought they would. The OS/2 version of the game was written in C, which made it much harder to have multiple AI personalities - there was a because you was a lot of copying and pasting of code which made it a hassle to introduce basic improvements to the AI.

The AI in the new *Galactic Civilizations* was written in C++ and using object-oriented methods. Inheritance saved a lot of time when we coded the core functions and allowed us to focus on creating the six different AI engines in the game without having to rewrite things like `CalculateEnemyMilitaryStrength()` half a dozen times. The result was an AI that is much more sophisticated than we originally expected.

But here's where things really get interesting about the AI. In the Fall of 2002, just months before release, users suggested having Starbases and galactic resources in the game - two concepts that were not in the design document nor in the development schedule. The beta testers' starbase concept was so good that we decided to put it in. That meant that the AI would need to be programmed to handle building, upgrading, and placing starbases with only a month or so of development time. While the AI has room for improvement that we've been adding to since the game's release, it still came together pretty well. I think years from now when people ask talk about the most important feature of *Galactic Civilizations*, they will cite the artificial intelligence.

**5. Choosing not to include multiplayer functionality.** We knew when we began development that it was going to be hard to compete in today's game market. So we relied on our beta testers heavily to fashion a game that would meet the expectations of strategy gamers.

One key decision we made based upon the beta testers' feedback was to nix multiplayer features, since it's much easier to add features to a game when you don't have to worry about how they'll impact issues like synchronization, latency, and game flow. Not having a multiplayer component allowed us to make *Galactic Civilizations* a much better game overall.

---

## What Went Wrong

**1. Stardock Central.** If you asked anyone who bought *Galactic Civilizations* during the first week to name the top thing didn't like about it, it probably would have been updating the game via Stardock Central. In our original plan, *Galactic Civilizations* was to ship in February, followed a month later by our "BonusPak" -- a giant, single file containing new features and fixes for any significant bugs that managed to get by us. A month after that, in late April, we had planned release Stardock Central as a way for us to provide updates to players.

But when we agreed to delay the game until the end of March, that made its release temptingly close to the Stardock Central release. We began contemplating a simultaneous release of *Galactic Civilizations* and Stardock Central, which would just require moving up the Stardock Central release date by a month. We were tempted by the thought of launching with impressive software updating technology that also included chat and discussion forums. Unfortunately, when the game shipped on March 26, Stardock Central wasn't quite ready.

Stardock Central not only had to deal with people downloading the full 500 megabyte game from the Internet, it also had to deal with people who bought the game at retail. It had to recognize what was already installed and update only what was new. It had to handle myriad date/time formats from around the world. And on top of that, we didn't take into consideration how many inexperienced computer users we'd be dealing with. Our Object Desktop customers had been using the beta version of Stardock Central for months with great results. But they are much more experienced computer users than the typical game player. Stardock Central, in short, was too complicated and too error prone upon release and caused us quite a bit of grief. We were able to respond to that by allowing people to download the game as a single big file, and to download the BonusPak as a traditional patch. But the online distribution system was more complex than we would have wanted it.





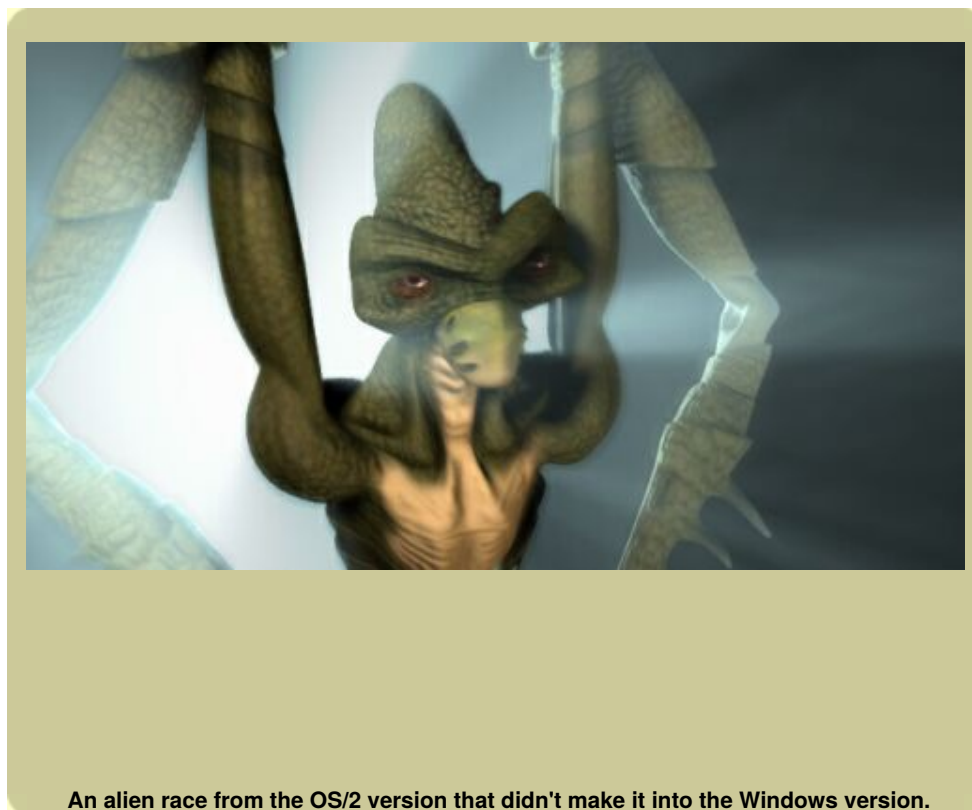
This from the diplomacy screen.

If we had kept the original April 21 release date for Stardock Central, I think it would have been hailed as incredibly cool, and would have demonstrated how serious we are to releasing lots of meaningful updates to our game. Eventually we think it will be viewed that way, as it has already evolved considerably. But in hindsight, I wish we had waited to release Stardock Central. By the time people read this, Stardock Central should be fine. But that first week was, unfortunately, painful for quite a few people trying to download our updates.

**2. The ship angles.** During development, we debated whether to have the game objects be displayed at an angle, or from a top-down vantage point. We ultimately went with a top-down vantage point, but one side effect of that decision was that all of the ships in *Galactic Civilizations*, which were modeled in 3D, effectively became 2D objects.

In retrospect, we should have devised an angled view system for the game, so the beauty of 3D ships would have been more apparent. When you play *Galactic Civilizations* and see flat-looking ships, you'll know the irony that every single one of those ships is fully rendered in three dimensions. This was my decision and I wish I could take it back now.

**3. No integrated technology tree.** I actually vetoed this feature. Our development team wanted a hyperlink system for looking at all of the technologies in the game. But as a long time Civilization player, it always bugged me that in these games you could magically know how to get to a certain technology, so I axed this request during development.



An alien race from the OS/2 version that didn't make it into the Windows version.

I was wrong. I still don't like that feature, but it should be up to the individual player whether to view the technology tree or play it blindly. I won't use the feature, but I shouldn't try to prevent others from doing it if they want. So we'll be adding this feature into an update now.

**4. Lack of team discipline.** As a project manager, I tend to be hands-off. But early on in the development of the game, that approach nearly proved disastrous. Games need a single vision that is precisely executed. On our non-game projects like Object Desktop, I tend to just guide the development teams towards a general goal and let them go and add things they think are necessary. But in a game, that can be a real detriment.

Early on, we had features sneak into builds that were buggy or wrecked the game balance. For instance, a user on our forum would suggest a feature and one of our developers would throw that feature in without telling anyone on our team, and consequently no one else would know to test that feature - including QA. Fortunately we had an open beta and we quickly learned the error of our ways. But quite a bit of energy early on was spent killing features or fixing bugs in unplanned features. So we switched to a much more disciplined system.

This is one of the toughest things deal with on a game project. You want your developers to feel like they can add in features on their own without having to get them all approved, but you also want to make sure they don't throw in something that wrecks the game. Often times

someone would put in a little update and it would nearly get out the door without some nasty side effect getting caught.

**5. The user manual.** The final version of the *Galactic Civilizations* manual isn't terrible, but it isn't what we had hoped for. This the fault of anyone in particular - in fact, ultimately it's my fault. We planned to contract out the user manual to a third party, but we never seemed to get off the ground with that project. I had written an outline of the manual some months before the game was to be released.

In early December, Strategy First informed us that the manual needed to be finished within two weeks. At the time, the release date was scheduled for the end of February, so I hadn't expected such an early date on the manual. I figured either a contractor or I would write it during December and January. So instead of having two months to really flesh it out, it was written in two weeks. As a result, it's far less detailed than I would have liked. When the game was delayed until March, I was surprised to discover that the manual, as late as early February, had not yet gone to manufacturing.

This was just a series of miscommunications between us and our publisher. Our publisher didn't realize that we wanted more time to work on the manual. The manual provides a decent overview of the game, but I wanted to put in more details about how economics, influence, morale, and industry factor into the game so that users could see the relationship between them.

The correct way to do the user manual would have been to early on hand it off to someone who is not coding on the game. Find someone who is really into this type of game. After the game went gold, I sent the game on to friends I know in the game industry and it became pretty apparent based on their questions that we had missed the boat on a lot of really basic concepts. For our future games, we'll try to get someone who is very... intense to be part of the documentation from early on so that when we're done, we have a really thorough user manual.

If we had done that, then we would had something strong right away and not felt like we had to rush to put something together.

---

[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved