Postmortem: Paradox Development Studio's Stellaris

gamasutra.com/view/news/274018/Postmortem_Paradox_Development_Studios_Stellaris.php

Henrik Fåhraeus is the Game Director of Stellaris*. He is a veteran game designer and currently the head of the design department at Paradox Development Studio. Over the years, he has worked as both a designer and programmer on most titles the studio has released.*

Rikard Åslund is the Project Lead of Stellaris*. He has been with Paradox Development Studio since 2011, working on games such as* Crusader Kings II *and* Europa Universalis IV*. Before he became Project Lead on* Stellaris*, he worked as a Senior Programmer.*

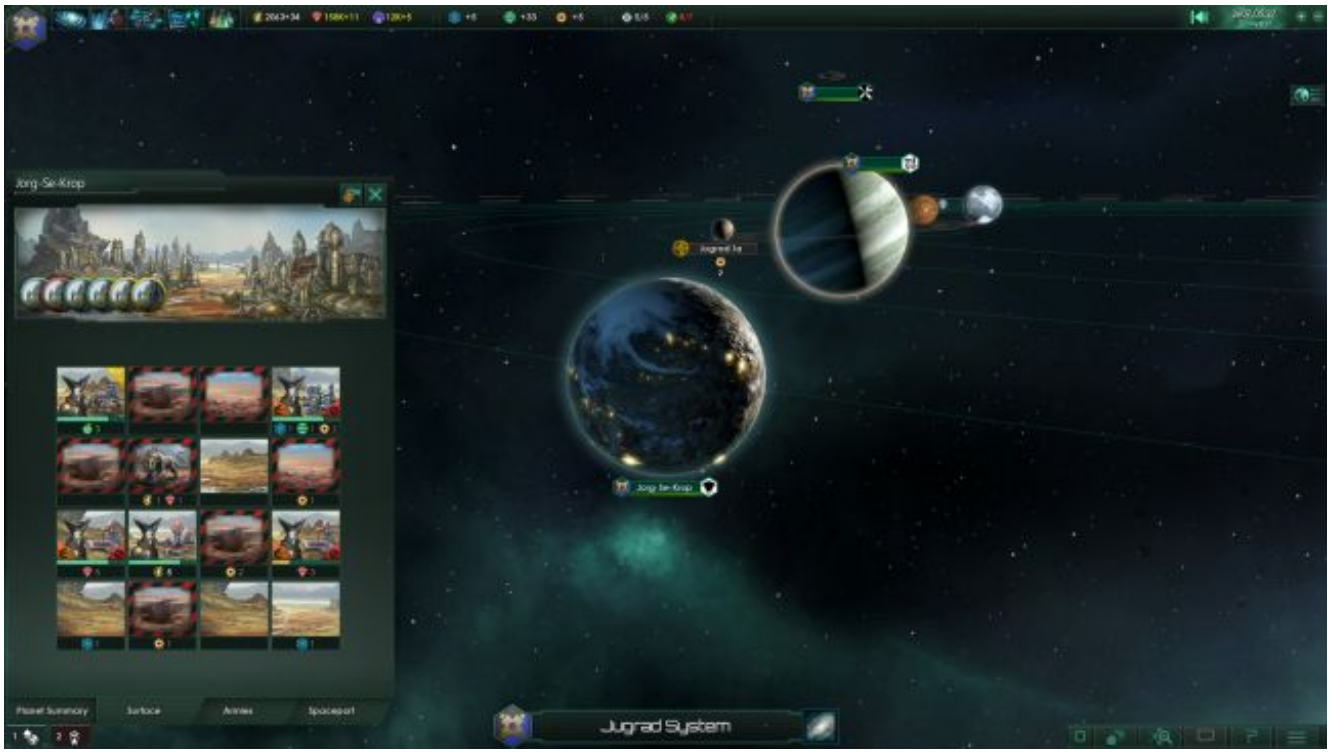
INTRODUCTION

Henrik: Paradox Development Studio is known for making its own particular brand of strategy games; very historical and quite hardcore. Our designers, especially, are absolute history nuts (me included, of course!).

However, after having made a few games like this, we started dreaming of applying our knowledge to something different, a project where we could let our imagination run wild where we wouldn't be shackled by historical events and processes. I think the first time a space game was seriously mentioned in the office must have been way back in 2005 or so. However, those times were still tough for the studio and we were better served sticking to safer ground.

Fast forward to late 2012. Times were good. *Crusader Kings II* had elevated the studio to a new level and we were looking to expand and try new things. So, in parallel with the development of *Europa Universalis IV* (which we regarded as a pretty safe bet), we started up a risky new project, codenamed "Nero". This was to become *Runemaster*, our ill-fated venture into the RPG genre.

In 2013, we thought we should aim even higher, and I started writing design drafts for a space game, codenamed "Augustus." I was torn between making either a kind of *Crusader Kings* in space, with a rich written lore and various interesting empires and dynasties (think Frank Herbert's "Dune"), or a more traditional 4X with special focus on exploration (which I hadn't seen done before.)



I wrote high concepts for both options, but we eventually went ahead with the latter for several reasons. We saw

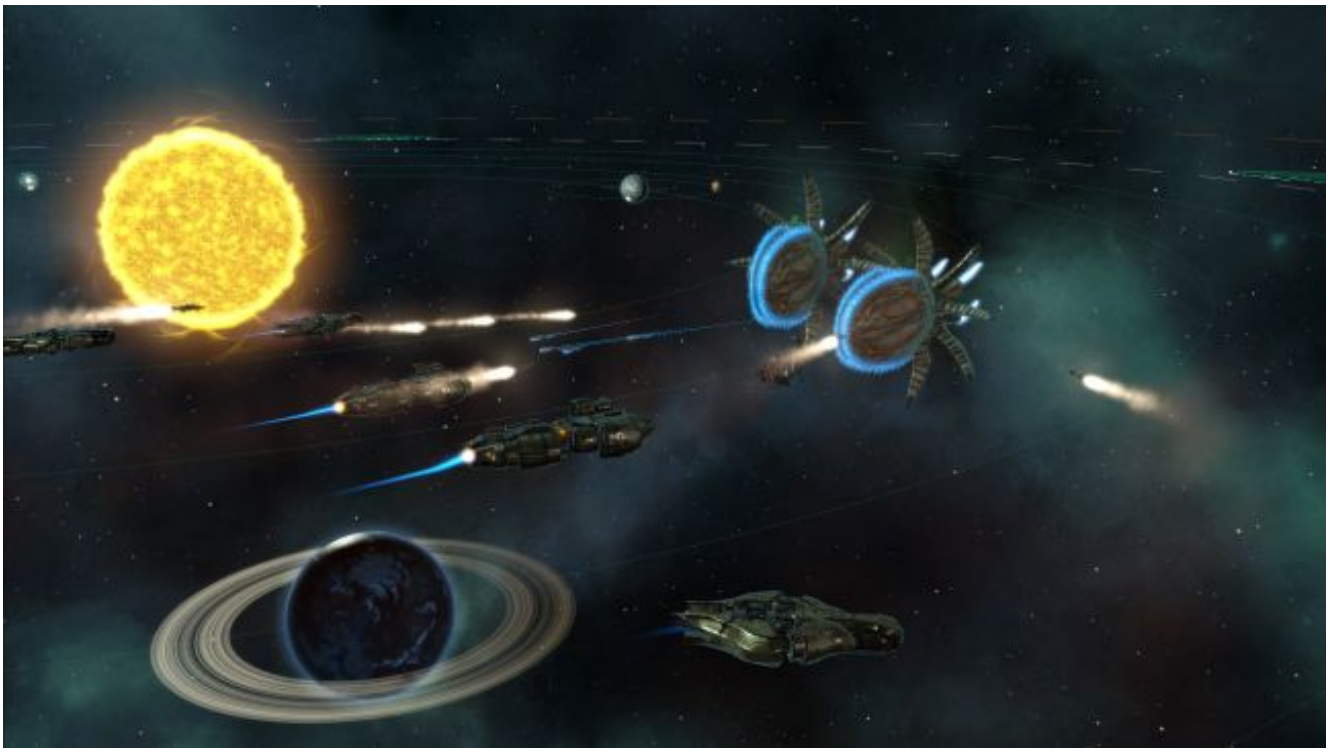
a potential opening in that market and we didn't want to compete too much with our own historical games. We also wanted to make a truly accessible and easy-to-learn game for once, and the small, symmetrical start of the standard 4X lends itself better to that purpose. So, when it was clear that *Europa Universalis IV* was another hit, production went ahead full speed on what was to become *Stellaris*!

My basic idea was to marry scripted, branching storytelling with the 4X genre and our own Grand Strategy formula. I love the little stories in the game *FTL* (and, naturally, *Crusader Kings II*), so I thought it should be possible…. and it really seems it was, because so far *Stellaris* has broken all records for the studio!

Rikard: *Stellaris* has in many ways been a dream project of mine. Sci-fi games have always had a certain attraction to me that few other games ever had. I love dreaming myself away to an alien world and the feeling of exploring something new and unknown. *Stellaris* gives me precisely that sensation. I feel extremely proud of what the team has achieved.

DATA BOX

Developer: Paradox Development Studio
Publisher: Paradox Interactive
Release Date: 2016-05-09
Platforms: Windows, Mac and Linux
Number of Developers: 8-15
Length of Development: ca 32 months



WHAT WENT RIGHT

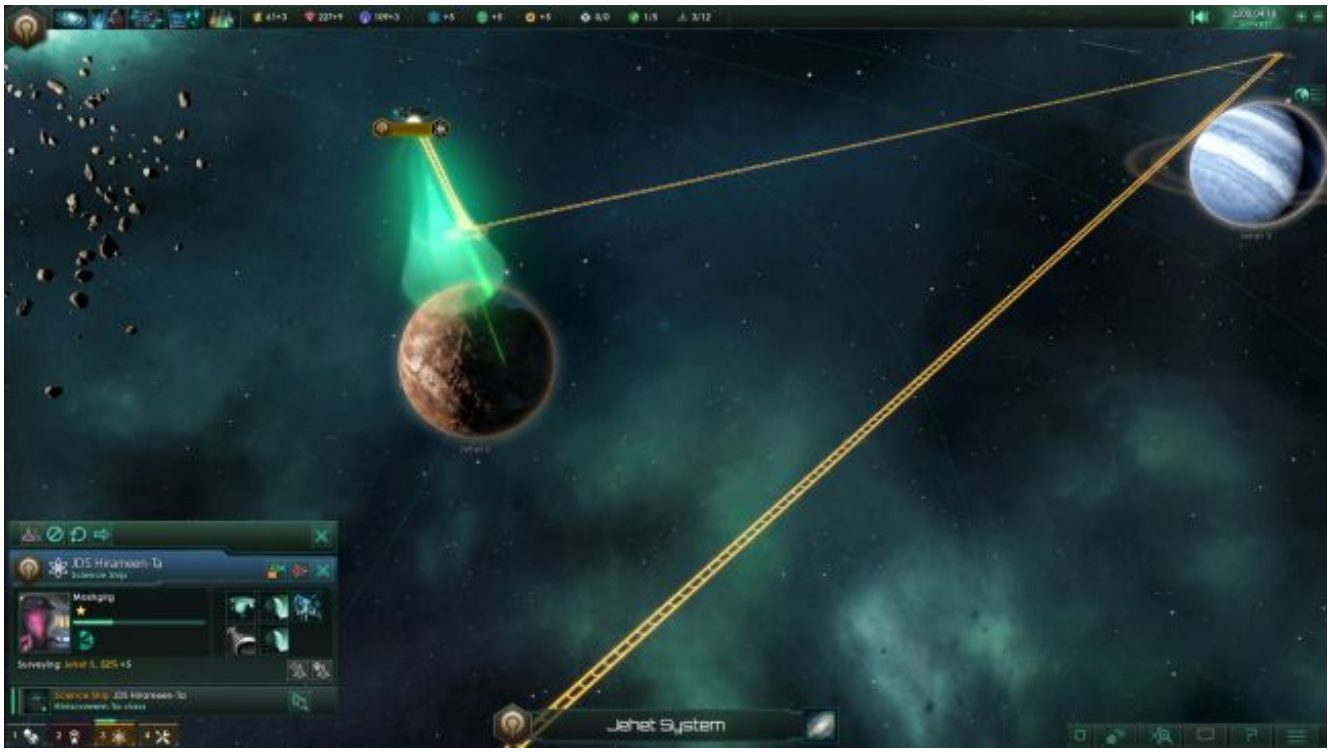
1. The Early Game

Henrik: We spent a lot of time working on the early game and the player's initial experience; surveying planets, working the surface tiles of your homeworld, discovering anomalies and making first contact with alien empires. I think we pretty much nailed this (although we still need more weird objects to find and study on the actual map.) Of course, the whole team tends to spend the most time playing the early game, so it naturally gets more love. Most critically, perhaps, it was the part of the game that I felt would make *Stellaris* stand out the most from other games in the genre, so I tended to obsess more over it. For example, I really wanted unknown aliens to be, well,

unknown. You should not be able to see the names and classes of their ships anywhere, and I wanted different graphics for unknowns in the fleet view. I even made sure that if you're playing a xenophobic empire, unknown alien fleets would look menacing in the fleet view, and be referred to with words like "threat" and "menace" (and the opposite for xenophiles). This was a fairly late addition, and should serve as an example of the kind of attention we kept giving the early stages of the game. If the entire game was as engaging as the early game, I would be satisfied (which is a rare thing for a designer).

2. Art & Sound

Henrik: We made an early misstep with the look of the graphical interface, imagining it should be super clean, minimalistic and mostly devoid of color variations. We corrected this gradually as we went, and apart from the remnants of that poor decision, the game looks and sounds beautiful. Everything in the game looks far better than I originally envisioned (though it also cost a wee bit more than originally budgeted); the ships, the planets, the aliens, everything is gorgeous. Kudos to our fantastic artists and graphics programmer! For the first time, we also had a dedicated audio director, and you really can tell from the quality of the sound effects and the voiceover; it's like a tapestry of sounds...



3. Accessibility

Henrik: This was one of our main goals with the *Stellaris* project, and I am really happy with the results. Starting on a single planet helps immensely to smooth out the learning curve since the more complex mechanics only come into play when you've grown bigger. I'm also satisfied with the "GUI hygiene" and, of course, VIR - the robotic advisor!

In our previous games, we have relied on a separate tutorial where you go through chapter after chapter learning about different mechanics. For that to work, we had to anticipate all the ways that players could break the tutorial and so we had to deactivate different parts of the interface, etc. Even more importantly, players want to play the game; they don't want to have to go through hours of tutorial lessons first. The very fact that such a tutorial exists can be intimidating to new players ("Wow, this game must be super complex… Screw it, I'm going to play Candy Crush instead.")

Admittedly, we are latecomers to this realization (or rather, to make it a priority); most other games have already taken the step to in-game advisor systems. Now, to make our own lives easier, I wanted the in-game advisor to be more reactive than proactive: It's much harder to break a tutorial like this if it keeps responding to your actions

rather than ordering you around. I'd have liked to develop the reactiveness more, and to have the advisor be more helpful in the later stages of the game as well, when players are introduced to more complex mechanics like our war and peace system. I'd also have liked the advisor to speak and behave more differently depending on your governing Ethics. Still, I am happy with the system overall, and I don't see us doing an out-of-game tutorial ever again; though we can certainly make further improvements. It does seem like we've reached a new type of player and brought them into the fold.

4. Stability & Quality

> ***"For the first time, we had a project being run by one tech-focused lead and a design-focused lead."***

Rikard: Historically our project teams have been very small (they still are in comparison to other companies) and back in our not-so-distant past the project lead acted as both tech lead, design lead and project lead. This allowed us to be extremely quick when making decisions, because the project basically was controlled by one person. With *Stellaris*, the team was a lot bigger so it was obvious that one person wasn't going to be able to do everything. Because of that, we changed the structure within the project after some time in such a way that I moved from my senior programmer/tech lead position to project lead and Henrik moved to a game director/design lead position.

Why is this relevant for the stability and quality of the game you ask? Well, because this is one of the main reasons why both stability and quality is so high in *Stellaris*. For the first time, we had a project being run by one tech-focused lead and a design-focused lead. Henrik had the final decision on what we would do and I decided when and how, this gave a really healthy constant friction between us two that forced us to constantly make compromises for what would best for the game at the given time. Balance is everything, because having a game that always crashes hides the quality that you have and on the other hand having a game that never crashes doesn't mean that it has any quality. Both Henrik and I know from experience that you need them both to succeed and *Stellaris* is living proof of this.

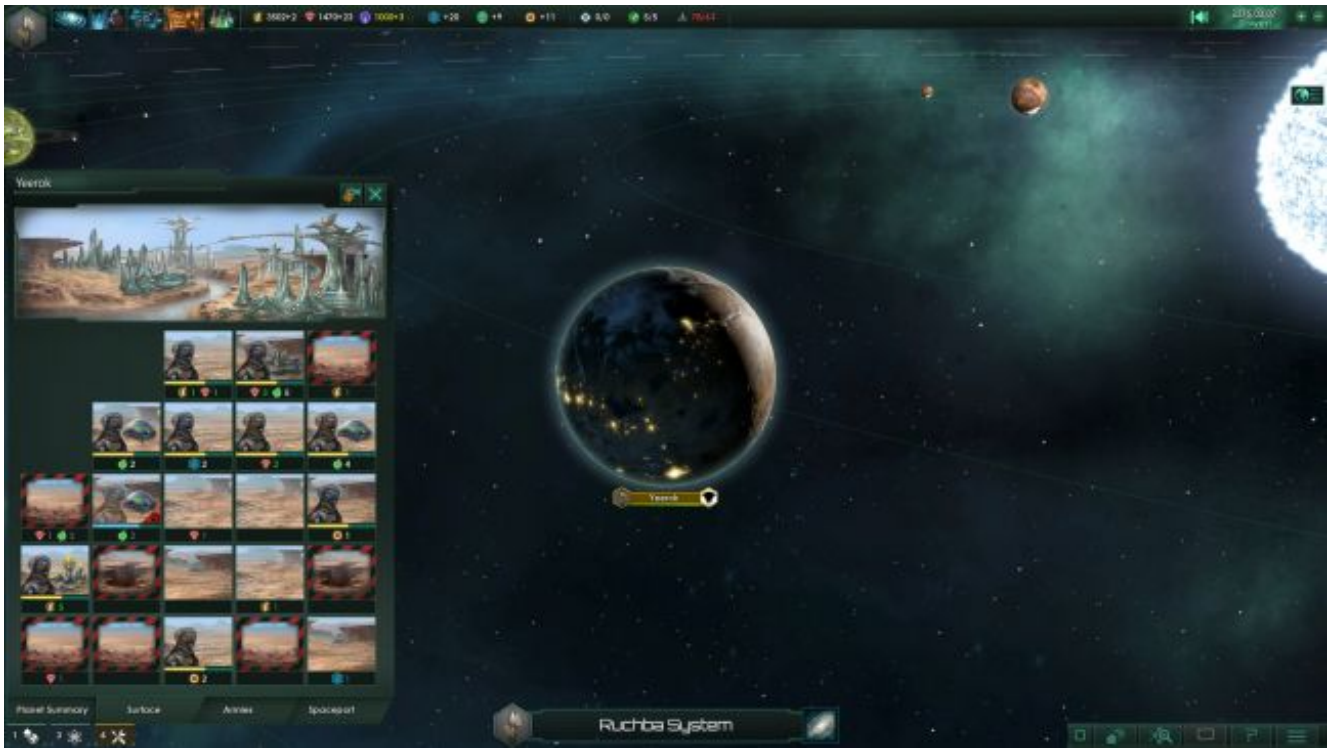
5. Multiplayer

Rikard: When I worked on *EU4* we put a lot of effort on making sure that MP (multiplayer) worked well with very little OOS (out-of-sync, the state of the game is different between the host and client) behavior. However, back then, we didn't have MP as a priority right from the start of development, and so, we didn't make sure the code architecture was suited for MP.

I was certain this was a mistake that I didn't want repeated with *Stellaris*. Right from the start of development we made it clear that OOS and issues with hotjoin (joining an already running game) were considered critical issues and should be resolved as soon as possible. A huge problem we had in our other games is that it was troublesome to reset the game to the same state as when it was started. Loading a save game or resigning always leaked states over to the next new game you started. This causes huge problems in multiplayer, because it is super important that the clients and host have the exact same state of the game, otherwise they will go OOS. Knowing this from previous experience we took the decision right from the start that the game should only "live" in one single object, the "game state". This allowed us to reset it or load another one without any states leaking through between sessions.

In many ways this is probably the technical decision that had most significant positive effect throughout the entire project. Most of the credit for how well MP works in *Stellaris* goes out to one of our programmers, Alexander Ivannikov, who throughout development has done outstanding work tracking down OOS and making sure we find and fix issues as quickly as possible. One of the ways he did this was through automatic testing with the help of a version of the game that runs without any graphics, allowing him to run multiple clients on the same machine. Even if I know that we could have done a lot more for MP, I still feel satisfied with how much better it works in

comparison with our other titles.



WHAT WENT WRONG

1. UNDERESTIMATING THE 4X GENRE

Henrik: In my hubris, I assumed that making a 4X game would be easier than making one of our grand strategy games (although I did dread the visualized ship combat.) In reality, however, the more limited ruleset means that the core mechanics need to be a lot tighter. Notably, the basic economy needs to be extremely well balanced in a 4X, and such resource models were new to us.

> ***"My biggest lesson from this that it is super important to establish an 'identity' of the game early on that everyone in the team understands and follow."***

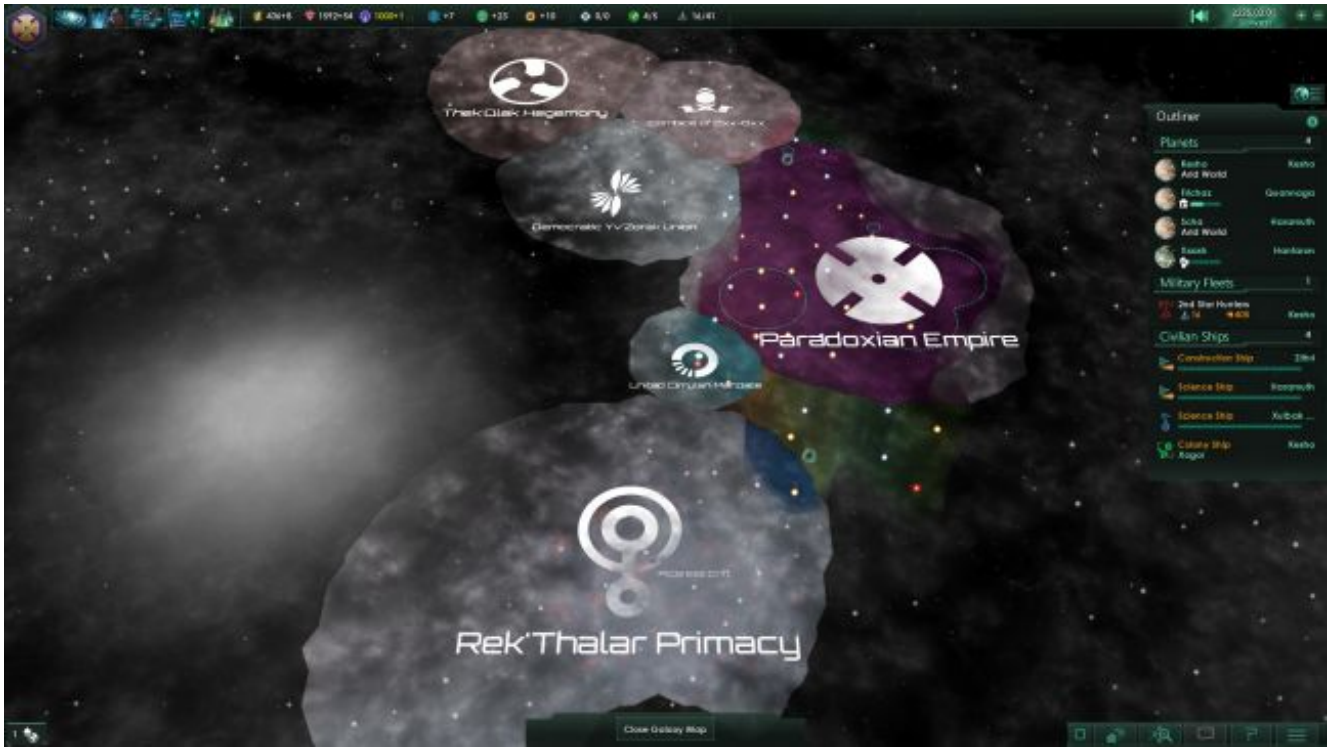
So, for example, early on we had to simplify the whole initial conception of resources, where you used to have a base resource (Minerals) that you mined and then had to process into a second resource you used for actual production. I also had to give up some of my favorite features, like adjacency bonuses on planet surfaces - all buildings of the same base type used to give each other an adjacency bonus until relatively late in development. (In the final version, you only get such a bonus from the planetary capital building.) We had to abandon the adjacency bonuses because of resource inflation; you used to bathe in Minerals and Energy Credits. I still feel bad about that because it makes planning your buildings less interesting...

Similarly, we added the caps on resource stockpiles fairly late. I could go on talking about a dozen similar balance and pacing issues that we ran into and had to overcome; Planet Habitability, the Sector system, Food, Happiness, etc.

Now, iteration and revision is an expected part of the process, but these challenges were a lot tougher than I had thought, and it's only due to the dogged efforts of designers like Joakim Andreasson and Daniel Moregård that we pulled through. The main lesson I took away from this is to identify the most unfamiliar and risky features right from the start and thoroughly prototype them with a smaller team in a pre-production phase. Do not underestimate a feature just because it looks simple on the surface.

Rikard: When we started out the development the general feeling was that we were quite certain of what game we were trying to make. After a while it however became pressingly clear that we were not certain on how the game should be played out, or how it should feel. We have experience with making grand-strategy game, but that doesn't necessarily mean that we know how a 4X game should be played. Even if the combined experience within the team from other 4X games is massive, it became clear that we were constantly being dragged in either the 4X or GSG direction. We knew that we wanted especially the early game to feel like a 4X and it was a mistake that we kept forgetting that during development.

For example, in a 4X you typically need to let the player be alone for a while, but in a GSG you already know about a bunch of other empires right from the start. The tweaking and arguments regarding how soon you should encounter other empires went on for way too long. My biggest lesson from this that it is super important to establish an "identity" of the game early on that everyone in the team understands and follow. You should have a clear idea how the game should feel and be played during the different parts of the game, whether it's the first 10 minutes or 10 hours in. It is also super important to continuously update the team on this in a way to make sure that everyone is on the same page.



2. MARRYING GRAND STRATEGY MECHANICS TO 4X

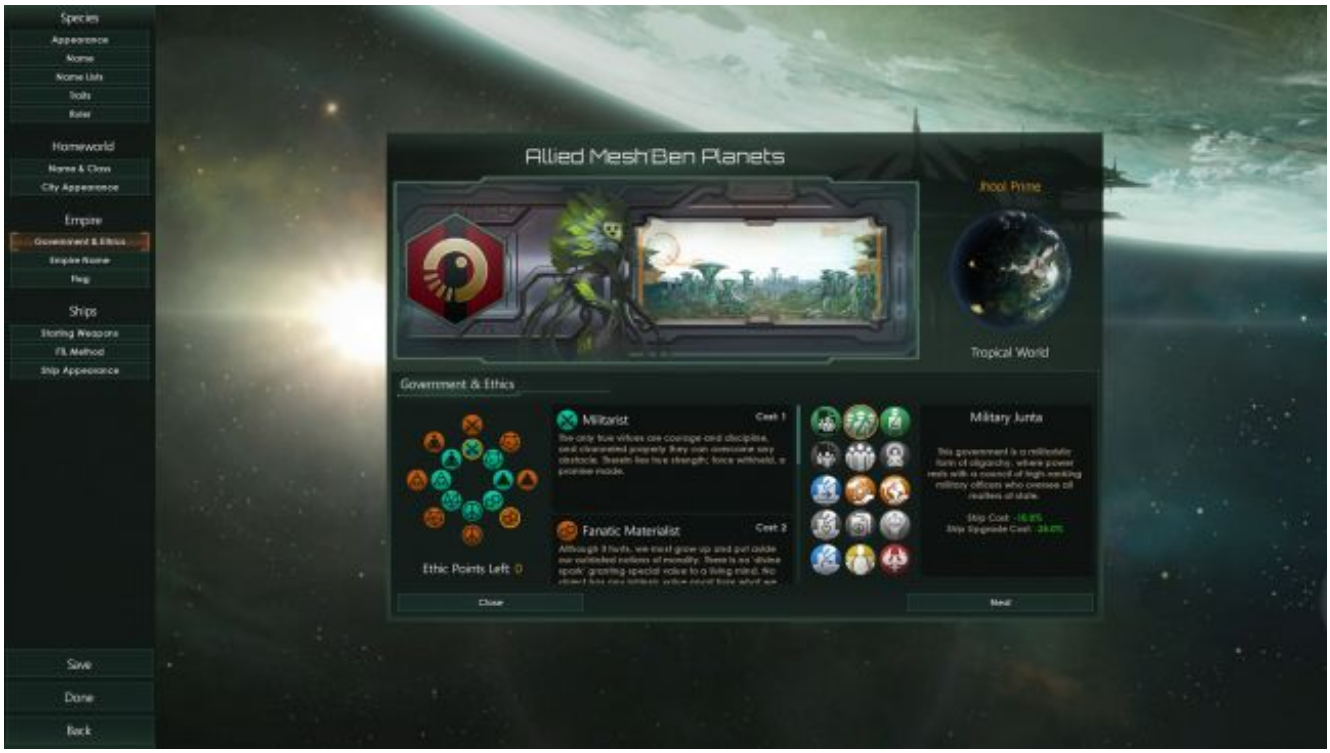
Henrik: I thought that we would relatively easily be able to adapt some of our best mechanics from games like *Europa Universalis*. In reality, it wasn't easy at all. Here's just one example: In our historical games (except for *Sengoku*), it's not enough to occupy the territory of your enemies; you need to negotiate for it in the peace treaty before you get to own it. This mechanic has worked very well for us in the past, but it wasn't simple to translate to a situation where empire borders grow dynamically. For example, we had a lot of discussions about how to handle "empty" solar systems (that is, systems within your borders but with no colonized planets.) Should you be able to negotiate for them too, and how would that affect the borders? That, and several related questions, eventually led to the conception of the "Frontier Outpost" station, which can also extend your borders. It works now, but it was a tough process.

Design reflections like the above also led to more abstract questions of what actually defines the sub-genres of strategy games and what we and our faithful fans tend to expect from a good strategy game. For example, 4X games by nature tend to snowball; you grow bigger, which makes you proportionally more powerful, allowing you to grow even bigger, and then nothing can stop you. This is fun early on, but you tend to lose interest in the game once you realize there's nothing left to oppose you. Our grand strategy games also suffer from this syndrome, but not as much as your standard 4X game. In *Crusader Kings II*, you may grow bigger, but not proportionally

more powerful since you have to rely on vassals to control more territory. In *Europa Universalis IV* you are limited by several relative constants like your Monarch Points; you can't get more of them by expanding your territory. In *Stellaris*, we have Influence, which fills a similar function, and also the Sectors which take the place of CK vassals. However, I am still not entirely happy with the snowballing tendencies and the fairly heavy-handed approach we took to prevent it for technological research, especially.

Rikard: Is *Stellaris* a typical 4X or a GSG? Well, to be perfectly honest I'm not sure, it is a hybrid. This meant that we couldn't take everything that we know works in a GSG and apply it to *Stellaris*, and we couldn't apply all typical 4X elements either. An example of a system that we had severe problems with fitting into the the game was the technology research system. In a 4X game you are expected to have an advanced technology tree, in our GSGs we usually use something much more simple and more linear. Technology trees are hard to visualize and you never feel surprised because you always know what you are going to get next (I know some people think certain games do this well, but I don't agree). The focus of *Stellaris* was always the sense of exploration, we didn't feel that a classic 4X technology tree gave that. We iterated on the technology research design much further into development than both me and Henrik would have liked, but we knew that this was going to be one of the core mechanics and that we had to get it to feel right. It was a total time-sink to come up with new designs, implement them and then iterate on them to get them to work.

Iterating can sometimes be like digging a hole, you need to be wary so that you don't dig so deep that the ladder no longer reaches to the top so that you can't get out. What we did right (but maybe were a bit slow on) was that we allowed ourselves to stop and decide that it was better to try and dig somewhere else instead. Quite late in the development we had a design meeting with the purpose to finalize the last redesign of how technology research should work; if this didn't work out we still would have to ship the game with what we had (no one in the team wanted this). During that meeting Johan Andersson basically came with this completely new idea to think of the technology tree as a deck of cards that is stacked and that we can reshuffle behind the scenes (knowing how much Hearthstone he has been playing I think I know where his inspiration came from). This allowed us to present a limited set of choices to the player and only provide relevant choices to choose from. It took a lot of work from the design team to balance but I love the fact that everytime I play *Stellaris* and choose a new technology I'm not certain what choice to make, that is exactly the feeling we tried to achieve. The lesson from this is that you should never be afraid to throw away what you have if it really isn't working and if you continue to fail it might be a good idea to look for inspiration somewhere else and try something completely new.



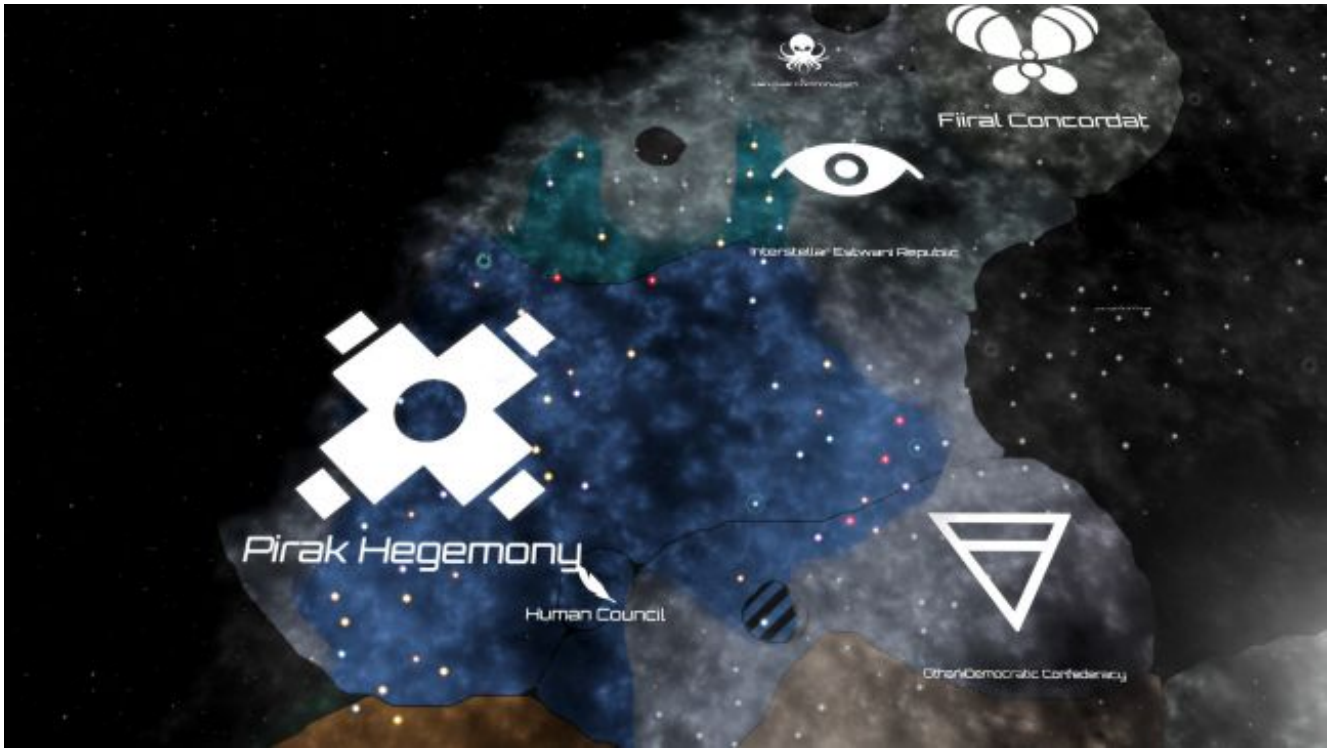
3) IMMERSION IN A PROCEDURAL WORLD

Henrik: The biggest danger with moving away from our historical home turf was the loss of immersion. One of the main reasons our historical games work despite the intimidating presentation is the satisfaction people get from "making things right" (nationalist power fantasies), or challenging themselves to alter history in other ways just for fun. Now, everyone knows what France and Russia is, for example, and their minds are already filled with various conceptions about them, so immersion comes easy. We don't even have to imbue these countries with much personality in the game; players will still perceive it.

Now, to deal with this lack of relatedness, we could have gone down the traditional route and made *Stellaris* about a few, pre-designed alien empires with very distinct personalities and backstories. However, we needed to keep the game Grand in scope, with many more playable entities than that. We also needed to stay on target; that first big X - exploration. The solution, I felt, was derived from the myriad possible combinations of Ethics, which should be reflected in how the empires behave and communicate. In the end, we didn't quite hit the mark, mainly because there are not enough ways for the empires to "act out": Regardless of ethics, they all just seek more planets and territory. I'd also like to give them some sort of quirks based on cultural traits and, perhaps, generated backstories. This is something I hope to explore in an update or expansion to the game.

4. SCRIPTED CONTENT

Henrik: While *Stellaris* has been almost universally lauded for the quality of its scripted content (the "Anomalies" and stories that can play out), I originally envisioned a lot more of it. I am still not entirely satisfied with the semi-random branching options of many Anomalies, but more than that, we have received some well-deserved flak for the fact that the mid-game has too little of such content in general. We spent a lot of time and effort coming up with Anomalies and things for your Science Ship to do, but, while other types of scripted content was conceptualized, with our near-sighted focus on the early and late game we neglected to properly flesh out the mid-game content. Our content designers have a really difficult job. There are so many challenges to overcome; the siren song of writing long texts (players will rarely read them and instead just find them intimidating), putting the writing ahead of interesting outcomes, not daring to ask programmers for more functionality, etc.



5. PROTOTYPING

Henrik: As a studio in the old days we used to live on a financial knife's edge. We had a tiny team making games in 24 month (or often shorter) cycles. There was no room for compromise and no time to dick around with prototypes at the start of a new project. Our dogged persistence in getting every planned feature in and still delivering games on time (regardless of the bug count) is what built the company. However, this kind of thinking

just doesn't serve us anymore. We have grown too big and well established and need to start doing proper prototyping, with everyone realizing that it's normal for some features to fail during this early, comparatively very cheap, stage of development. The failure to do this is, in large part, why *Runemaster* was ever announced (it should have been scrapped or reenvisioned much earlier, during a proper prototype phase.) And, of course, it would have made the development of *Stellaris* a much smoother ride.

> ***"I am going to be blunt here; not doing proper prototyping is stupidyplain and simple. There are no excuses."***

Rikard: Internally we have many times said that it is impossible to judge a system before all of the related systems are in place as well. The argument behind this is that our games are so complex and most systems interlock with each other. This idea of how we develop games usually has the consequence that we discover fundamental problems really late in development.

I'm going to be blunt here; not doing proper prototyping is stupid, plain and simple. There are no excuses. Even if you can't prototype everything, you can at least isolate some parts of a system that you can try out. It is always better to discover problems sooner rather than later. One of the systems that suffered for lack of prototyping in *Stellaris* was the fleet combat system. With this game, it's the first time that we have a proper graphical representation of the combat that takes place, with ships flying around and shooting at each other (corresponding in real-time to the combat logic). One of the issues we were having was that we were keeping to the notion that the fleets needed to obey the normal movement rules even in combat, but it didn't look good in combat since they needed to take much quicker turns to look good (trust me, it looked awful). It wasn't until Daniel Eriksson (former programmer on *Runemaster*) joined the project that we managed to get it to a satisfactory level. Daniel right from the start set up a special quick start of the game that was only for testing combat and fleet movement. He added a system to set up scenarios with different types of ships and weapons. You could then let the game to play out the combat in exactly the same way as it would have in the normal game. Almost immediately he came to the conclusion that the ships needed to obey different movement rules when in combat. With this system we could iterate quickly and prototype different scenarios and variables. Because of these hard lessons that we have learnt from the development (and from other projects) we now have a mandatory proper prototyping step of our development cycle. You should too!

CONCLUSION

Henrik: We (or at least I) underestimated many of the mechanics that we thought would be fairly simple to nail. This mired us in design iterations too late in the development cycle. Although the game turned out great in the end, some of the more advanced design concepts suffered for it. Things like Federations, Alliances and Uplift - although functional on release - should have received more love. We also neglected the mid-game due to obsessing over getting the early experience right and making sure the late game threats were awesome. Fortunately, thanks to everyone who has bought the game, we'll get a chance to improve on these things in the free updates we've got planned!

Rikard: Looking at the reception that *Stellaris* has received, it is impossible to call it anything other than a success. As a passionate developer however, you never feel completely satisfied and you always have things that you would have done differently. With the reception we have received it is obvious that we are going to be able to continue working on this game for a really long time. I have said many times during development that we will probably not know exactly what *Stellaris* is until a year after release, and we are really looking forward to be along for that ride together with our players!