

JUMP POINT

ISSUE: 10 03



IN THIS ISSUE →→→

03	BEHIND THE SCENES : Refueling
15	BEHIND THE SCENES : Rivers
27	GALACTAPEDIA : Triggerfish
29	PORTFOLIO : Clark Defense Systems

FROM THE COCKPIT

GREETINGS, CITIZENS!

Happy March and happy (now slightly belated) Stella Fortuna! As the old saying goes, March comes in like a lion and goes out with a brand-new issue of **Jump Point**. So, if you're reading this then you made it through! I have good news and bad news this month. The good news is for everyone who says we have too many articles about ships: this month we've got two interviews about new game mechanics! The bad news is for everyone who looks forward to articles about ships: you'll have to wait until April for the next one.

First up we've got an article consisting of two big interviews covering the new refueling mechanic that's launching in Alpha 3.17. Way, way back at the start of the project I pitched the MISC Starfarer as part of a second wave of ships that would introduce unique game mechanics we knew we had to work towards... and now some much smarter people have made that dream a reality! Refueling is both incredibly complex of a project to implement and it's something that stands on the top of so many other features, meaning it wasn't even possible to do until just recently. It's always extremely interesting to me to learn about how each feature comes together and what it relies on, so these interviews were a joy for me. Big, big thank you to Thorsten Leimann and Guillermo Bibao for sharing their expertise. I always tell developers they're free to go into extra detail in these because **Jump Point** readers know the project more intimately than anyone else... and I'm grateful they both took me up on that! And as an added bonus, I'll promise you this: you will never, ever find the word "nozzle" printed so many times in the pages of this publication.

Our other feature this month is another interview covering the rivers system, which is also coming online with Alpha 3.17. When we set out to do a **Jump Point** interview, I typically ask our producer for a list of folks we can talk to about a given subject. We normally get anywhere from five to ten names of people involved in different aspects and have to narrow it down from there to who is available, who best represents what's happening, and so on. Well, for rivers, the answer came back immediately that there was only one person we needed to talk to: Programmer Will Hain, the driving force behind the effort. I immediately thought of the scene in *Hitchhiker's Guide to the Galaxy* where Arthur visits Magrathea and meets the man responsible for designing Norway's fjords (he won an award!). You have seen rivers in action on the most recent *Star Citizen* and I hope this provides a nice supplement to that history!

But wait, there's lore! On that side of the **Jump Point** we've got another in-depth portfolio of one of *Star Citizen's* companies, armor manufacturer CDS. It already has a fair amount of equipment available in-game so it's likely not the first time you've heard the name! We've also got a brand-new Galactapedia update covering Triggerfish. That's not another fish for your hangar aquarium but the *Star Citizen* equivalent of April Fool's Day and... hey, that's next week!

I hope you enjoy the issue and I'll look forward to seeing you next time, through the **Jump Point!**

Ben

JumpPoint@cloudimperiumgames.com

Editor & Writer: Benjamin Lesnick Narrative Team Writers: Adam Wieser & Cherie Heiberg Design: Michael Alder

Copy Editor: Martin Driver In-Engine Imagery: Charline Wegria, Simon Ravenhill & Simon Jennings

Associate Marketing Producer: Dan Houbrick Marketing Art Producer: Matthew Simpson Narrative Team Producer: Blythe Gumminger



BEHIND THE SCENES

REFUELING



One of the biggest updates coming with *Star Citizen* Alpha 3.17 is the refueling gameplay loop, a long-awaited mechanic that will begin to make fuel matter. Refueling has been in the works since the very early days of the project when Chris Roberts and company first imagined the MISC Starfarer, a tanker that introduced the concept of in-flight refueling (plus fuel refining, storage, and sale). Now, as other game mechanics have been achieved, such as docking, the first major release of refueling is about to come online. To learn about the incredible amount of work that went into making refueling happen, we spoke to the designer behind the system and the engineer responsible for implementing it.

BEGIN TRANSMISSION :

DESIGN

JUMP POINT: Please start by giving us your name, title, and letting us know what you've worked on for *Star Citizen*.

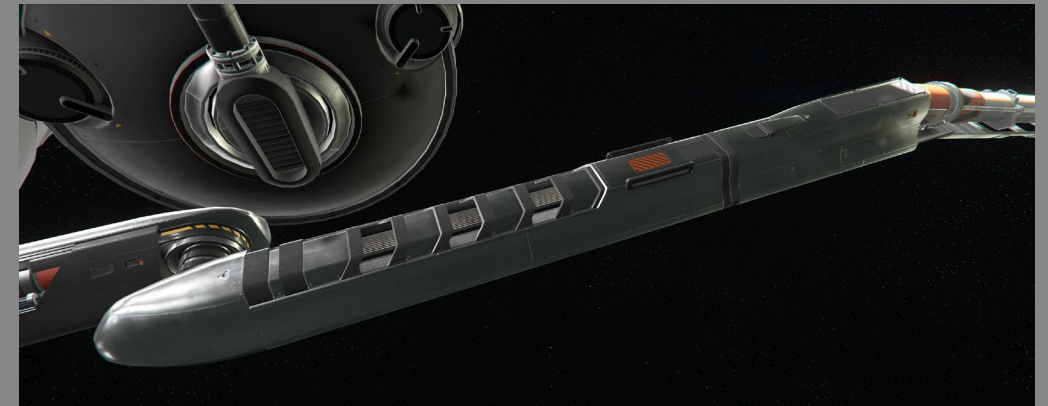
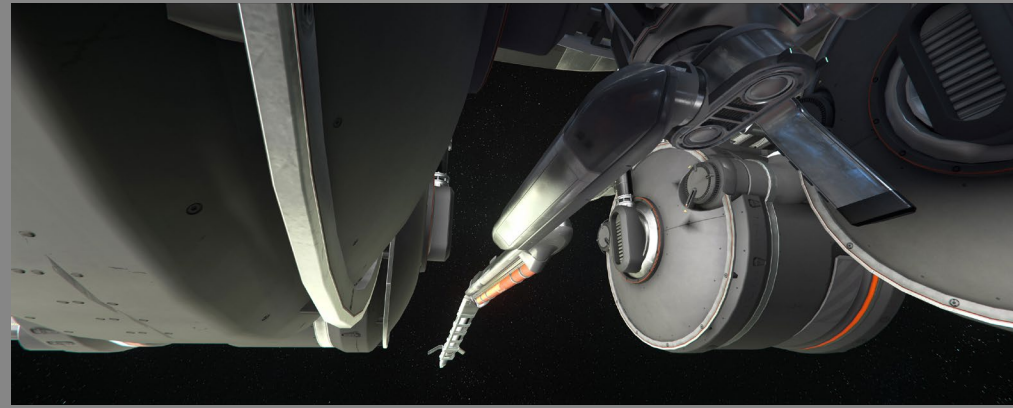
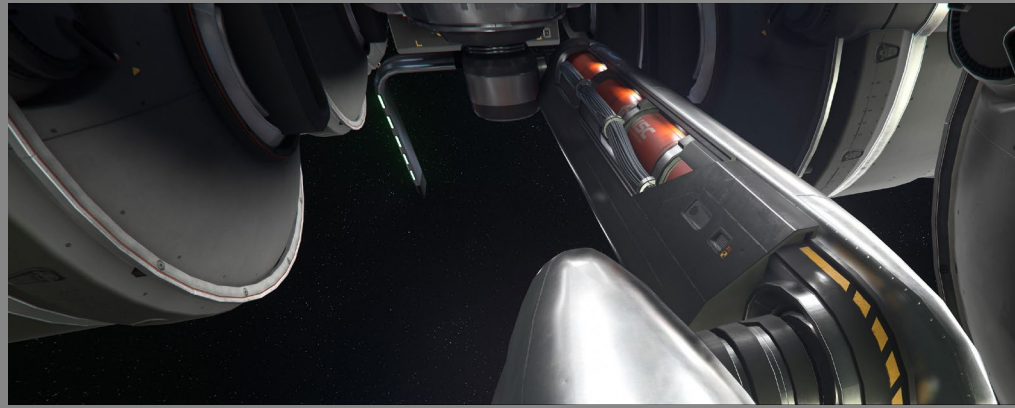
THORSTEN LEIMANN: Hi, I am Thorsten, a senior systems designer, and I worked on the laser trip mine, mining gadgets, atmospheric depth damage, loot generation, and now refueling.

JP: What was the high-level goal for refueling?

TL: The high-level goal for refueling was to allow players to explore areas with no space stations or outposts without getting stranded. For the gameplay, we wanted to have an interaction that players have to pay attention to. We did not want to have the system feel too passive, but on the other hand, we did not want to make refueling too complex.

JP: How do you approach designing such a complex addition to an already complex game?

TL: The first step was to know the ship inside and out. Luckily, the Starfarer was already available, so I spent a lot of time flying the Starfarer in the PU and sometimes pretending to refuel other players.



The next thing I did was gather as much information as I could get on real-life comparisons. From this research I compiled a list and defined goals for our game, which I discussed with our game directors to define the intention for refueling in *Star Citizen*.

From there on it was experimenting with paper prototypes, Lego ships, and whatever I could find to get a feeling for the flow.

We had to compare those ideas with our set goals and intentions as well as check them on technical do-ability, which eliminated a bunch of the ideas. We took what was left and started working on them. From that moment on it was a very iterative process.

JP: *What other disciplines were involved in making refueling happen? What kind of work needed to be done on existing environments and spacecraft to prepare them for your design?*

TL: We spent most time with the UI team since this is a very UI-heavy feature and we had to make sure that it delivered an industrial, heavy-

machinery feel while being understandable without a tutorial. So, all the functions had to be self-explanatory and have the flow clearly visible on the UI itself.

In addition, we needed some work from the Vehicle team as the boom arm needed to be added to the ship. We had already added some things for the on-ship refinery but had to adjust the screens that players can use to interact with the pods and, of course, add the refueling terminal to the cockpit.

The Vehicle team also had to remove scooping from ships that were not supposed to automatically refill hydrogen fuel, which was placeholder until refueling came into the game.

Alongside these two teams, the classics were also involved, like Audio and VFX.

JP: *What other work needed to happen to make developing refueling possible in the first place? Were there any other requirements that made NOW the time to implement refueling?*

TL: Docking was the big thing that made this idea possible. Apart from that, nothing else was required.

JP: *How much prototyping happens to get to the new system working? Did you go down any paths that didn't work out as intended?*

TL: We spent a lot of time prototyping the UI to get the flow right before we talked to the UI team. This was beneficial since we already had a working whitebox UI that helped us explain the feature better to other teams. UI also knew exactly what was needed.

The gameplay itself was also something that we had to change a bit based on some feedback from our internal playtests. For example, we wanted to have a pressure mechanic the player has to constantly pay attention to. The idea itself was really nice on paper but, upon trying it out, the mechanic was very tedious and resulted in players wasting too much time setting up to refuel.

So, we went back to the drawing board and looked at our intention,

which was that players have to pay attention to the refueling mechanic and don't do it in an automated way. As a result, we developed the idea of risk around the nozzle. In the end, this was the better way to go since it has the potential to tie in with other mechanics, like fire through spillage. We still met our goal to have a little risk/reward scenario to the refueling process (higher flow rate = faster refueling but potentially damages the nozzle).

JP: *How do you test and iterate on a new game mechanic like this?*

TL: This was quite challenging since it is a pure co-op mechanic. That means we always need two players to be able to test the process. In addition to our internal QA, who was of great help here, we also used our internal playtest sessions to test the feature regularly.

So, we gathered all the feedback after this play session and assessed it, created tasks or improvements out of the feedback, and kept on iterating. Actually, we are still improving the feature now using the very same method.



JP: Does this pass establish how fuel is created in the first place or just how it's transferred from ship to ship?

TL: This is the first step, so it is all about delivering it from one player to another.

JP: Was there anything you needed to cut that you wished could've been included?

TL: Plenty of things were cut. The most prominent aspect that we did not put into the initial release was the nozzle camera – a camera directly attached to the nozzle that allowed the player operating the refueling UI to see what was happening at the nozzle.

Another feature that was postponed is stress on the nozzle. If you are refueling while in motion, the pilot of the attached ship would be required to counter the stress on the nozzle. Flow rate would also have an impact on the stress on the nozzle. So, a higher flow rate would require the attached pilot to counter the tension more actively.

One smaller thing we really tried to fit into our initial release was a nozzle variant that had a limited flow speed but would automatically close on successful fuel delivery. Maybe we can add this in a later patch. As always, it depends on testing and feedback. An idea can sound nice on paper but always has to be evaluated for its overall contribution to the player experience.

JP: What are you especially proud of in the finished system? Going in, was there anything you especially wanted to make real?

TL: We are really happy with how refueling actually plays out. We are aware it is a “specialist” type of gameplay and might not be for everyone, but we are really happy that it is fun in its own way. It will bring some value to support players in the ‘verse and is the first official player-to-player trade we have.

What is always important to me is that there is very little “magic” in simulation gameplay. Here, I am really happy that we managed to have an explanation for almost everything and it fits the real-life fantasy.

JP: Do you see this as the final form of the refueling system or are there plans to continue working on it?

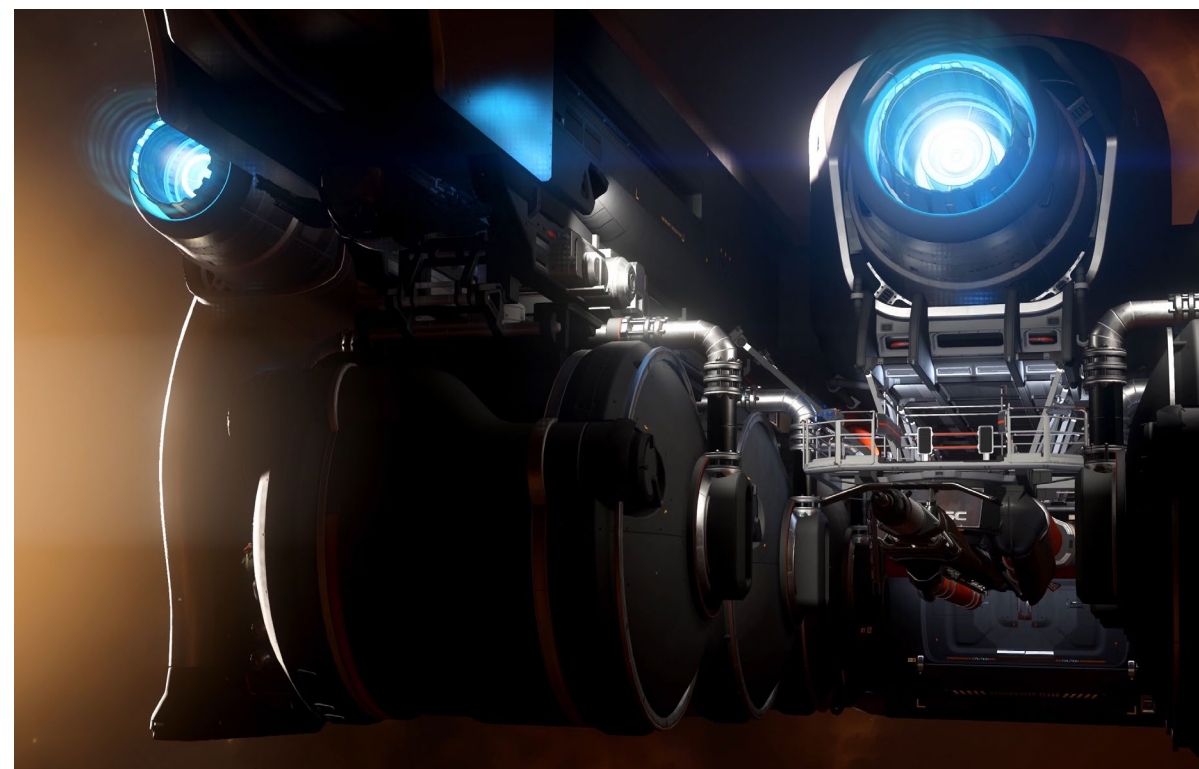
TL: Well, as stated above, there are some things we still want to do and there is also the Aegis Vulcan, which will require a different take on refueling.

JP: Are you allowed to say what you're working on next?

TL: Yes, right now we are working on several features at the same time. We are helping with the vehicle side of salvage with the Reclaimer and Vulture. We also started work on engineering gameplay and life support for ships, which I am really looking forward to.

JP: Do you have any messages for the players who are about to explore your work?

TL: Enjoy! I'm really looking forward to the Starfarer centipede!



ENGINEERING

JUMP POINT: Please start by giving us your name, title, and letting us know what you've worked on for Star Citizen.

GUILLERMO BILBAO: I am Guillermo Bilbao, a gameplay programmer here at CIG. I have worked on mining (particularly gadgets, consumables, and volatile cargo), the station refining system, and, of course, refueling.

JP: What is the high-level goal for refueling?

GB: I would say the main goal is to expand resource management. Up until now, running out of fuel is something that rarely ever happened, but mostly because everything happens inside a star system with a very high density of spaceports and services, and we had stopgap features like scoops to keep things convenient while we developed refueling. With the future arrival of Pyro and other similar systems, players will have fewer spots to refuel at and will need to plan accordingly. This also opens an obvious niche for ships such as the Starfarer that can carry large amounts of fuel and provide it to buyers or to their organization's fleet.

There's also the fact that this puts fuel trade in the hands of the players, which has been limited to something only stations did up until now. This will open up more opportunities for contracts, missions, and the dynamic economy.

JP: What does an engineer do to implement a new game mechanic? How do you go from design concepts to something that's actually working in the game engine?

GB: No two mechanics are the same but I like to start by looking at the design directions and making notes of what existing systems can be reused or repurposed. In this case, the fuel management part of refueling is just making use of the ship pipe system by adding a series of state machines to control the flow from tanks and nozzles. Similarly, refueling uses the existing docking system to establish the connection between both ships.

Afterwards, I will sometimes build a prototype if there are any unknowns I still need to figure out. This way we won't commit to a technical design that doesn't do what we need it to.

At that point, a proper technical design document will be written by me and inspected by other engineers to make sure we're not duplicating functionality or introducing performance issues and to ensure the plan itself makes sense. The document may be rewritten multiple times and even after it's approved it's a good idea to come back and adjust it to better reflect the final implementation.

If the feature is going to rely on another team's code (i.e., it will be making use of services), this is a good point to set up meetings and discuss necessary changes.

Once there is a clear picture of work needed to build the feature, we subdivide the tasks, assign them, and then start work. Though it is important to note that even at this point the designs are still only there to guide us; early feedback can result in changes to the goals and design. Then the feature is iterated on until all the stakeholders are happy with the result and approve its release.

JP: How do you approach engineering such a complex addition to an already complex game?



GB: I believe that any system is best built from simple blocks. I think that if the goal is systemic gameplay, we want to avoid bespoke solutions that only cover our current design. Obviously, we cannot plan for every possible future use case but we can try to keep things simple and build our features from that. At its core, refueling is just using the existing mechanics of fuel and the associated ship components and expanding the control the player has over them.

JP: *The concept of refueling in Star Citizen goes all the way back to 2012 and the Starfarer tanker itself is nearly as old. How much of that did you need to take into account today?*

GB: We were careful to go over the players' expectations and our overall design goals regarding the Starfarer's role in the game. That's why it's important to note that this is the first iteration of refueling. Some features such as collecting fuel and refining it won't be available in this initial release.

JP: *What other disciplines were involved in making refueling happen? What kind of work needed to be done on existing environments and spacecraft?*

GB: Vehicle content was very involved as this feature means nearly every ship in the game has to be able to dock to refuel. Of course, there were also changes and updates to the Starfarer itself. This meant new art, animations, VFX, SFX, etc. The fuel control UI and the mobiGlas changes meant that the UI designers and artists were involved. Additionally, since this feature made use of the existing docking code,

engineers from the Vehicle Feature and Physics teams were involved in making the necessary changes for this type of docking to work.

JP: What other work needed to happen to make developing refueling possible in the first place?

GB: Aside from docking, I think the Building Blocks UI contributed a lot to the feasibility of this project, as UI development resources are finite and are often spread thin (we only have so many UI programmers after all). However, I think the key point was the commitment from several teams to work on the feature.

JP: How much prototyping happens to get to a new system working? Did you go down any paths that didn't work out as intended?

GB: Prototyping time was quite short as most systems worked largely as needed with minimal changes, so we were able to move on quickly. There were some features that we had to put on our backlog due to either tech or time constraints, but we had a realistic idea of what we wanted to do for our initial release and I think we did a good job sticking to it.

JP: How do you test and iterate on a new game mechanic like this?

GB: I have taken to building a set of tools for quick debugging and modification of most aspects of the feature. Even so, considering the scope of changes, a significant amount of QA time was required to test all the possible scenarios and use-cases as well as ensuring sure every ship worked as intended.

JP: Is there anything that are you're especially proud of in the finished system?

GB: It's hard to point at just one thing. I think the new UIs and the fuel management aspect turned out nicely. I hope we can apply some of what we learned to other engineering and profession screens.

JP: Do you see this as the final form of the refueling system or are there plans to continue working on it?

GB: There's still work to do for sure. As other features and professions are developed, we will want to go back and revisit refueling and its relation to them. We also want to add more control and functionality to

the ship, and fuel management is certainly going to be part of it. Fuel collecting and refining are the most obvious additions to the system and I think they will really make the Starfarer gameplay stand out from most other ships.

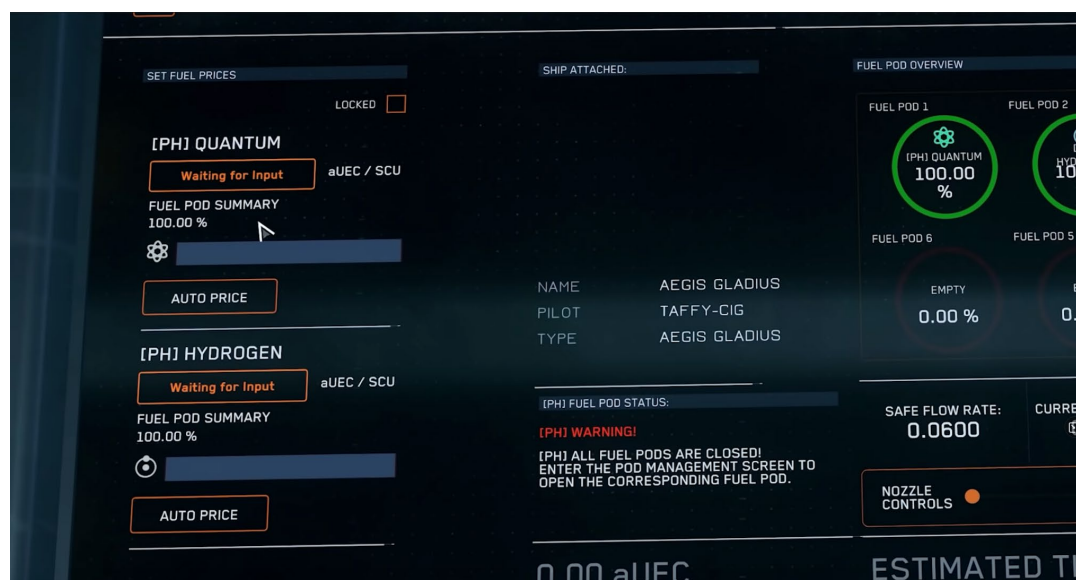
JP: Are you allowed to say what you're working on next?

GB: Sure, we will be taking what we've learned from refueling and applying it to ship engineering. We're still in the early stages but multi-crew players are the ones most likely to enjoy the added complexity.

JP: Do you have any messages for the players who are about to explore your work?

GB: I hope they enjoy it and be sure to give us your feedback on the feature so that we may improve it for future releases

END TRANSMISSION



REFUELING QUICK GUIDE

BUYING: HOW TO REFUEL FROM A STARFARER

Both ship-to-ship and station-to-ship refueling use a very similar interface, so if you figured out refueling at Olisar, chances are you shouldn't have any trouble buying fuel from someone else.

1. Send a docking request
2. Dock with them
3. Open the Ship Services app on your mobiGlas and make your request.

Tip: Autopilot will handle most of the docking process, but remember to fly slowly near the Starfarer.

SELLING: HOW TO SELL FUEL FROM A STARFARER

1. Buy fuel, making sure you fill the external pods as opposed to the internal fuel tanks. This can be expensive, as the Starfarer can hold a lot of fuel
2. Determine your selling price via the fuel console on the bridge. Automated pricing will provide a 10% profit margin, but you're free to set it at whatever you like
3. Find a customer. Once found, extend the fuel arm and approve the docking request
4. Your customer will submit a request detailing how much fuel of each type they want. Payment is held in escrow by the servers and paid out as the fuel is provided
5. To start refueling, open the correct fuel tank and nozzle and set the nozzle's throttle. You cannot transfer quantum and hydrogen at the same time

Tip: Fuel delivered that the customer did not request will be lost, so be careful to reduce spillage. However, moving excessive volumes of fuel will damage the nozzle.



BEHIND THE SCENES

RIVERS

A river runs through it! Starting with *Star Citizen* Alpha 3.17, players will have their first chance to explore the game's brand-new river system. When *Star Citizen* started generating entire worlds to explore, the technology was astounding. But, from the very start, the team knew that that first iteration of planets would need to be enhanced with key subsystems to create more and more of the details needed to make them truly real. From caves to vegetation to rivers, each has been developed with an eye towards continuing to build on *Star Citizen's* high standard of detail. To learn more about the new river system, currently in Evocati testing at the time of writing, we spoke to the person who made it happen, Programmer Will Hain.

BEGIN TRANSMISSION :

JUMP POINT: Please start by giving us your name, title and letting us know what you've worked on on *Star Citizen*.

WILL HAIN: I am Will Hain, I'm an engine programmer III and I've worked on a few things for *Star Citizen*:

- I created the terrain modification system
- I work on the planet object scattering system (foliage, geology, etc.)
- I've worked on lots of little things, like our floating seaweed and the wet edge effects
- I created our procedural river tech :)

JP: Why add rivers now?

WH: I've been working on rivers on and off since September 2020. We're finally getting one out now because after a few periods of work on it (interrupted by work on other things) it's finally reached maturity. That means, in addition to the core tech itself, more of the surrounding changes have been made to integrate it into our game engine properly. This includes audio, harvestables, distant LODs,



player interaction, vehicle interaction, and more appropriate scattering as well as many performance and optimization improvements.

JP: How do you implement a new addition to *Star Citizen's* planets like this? What's the high-level design goal for rivers and how did you make them happen?

WH: Implementing rivers is actually something I mentioned in my original interview in February 2020. I was super excited to work on this because in my own dissertation project I'd focused on rivers on procedural planets but not to the extent that would be up to the standard for *Star Citizen*. From a high-level perspective, the aim was to be able to place a spring point at any position and trace the water's path from there. I did an experiment with placing an estuary and finding where the water should have come from in order to form, but this ultimately didn't work out and I got better results from doing the opposite. Design was iterative throughout but a high-level design of four classes was worked out from the start: the River Placement Tool, the River Calculator, the River Data Manager, and the River Populator, as well as various terrain modifiers, storage classes, and helper classes. Each of these has a very specific purpose, from processing inputs from artists to serializing the river data. Of the four main classes, only the River Data Manager and River Populator are used at runtime, with both the placement tool and calculator remaining editor-only.

JP: How do you approach engineering such a complex addition to an already complex game?

WH: When I joined, our planet tech didn't allow for any local modification.



That is because high-resolution, planet-sized meshes can't be stored practically, so we have to generate our planet's height and mesh data on the fly from a collection of 16 heightmaps, a distribution map, and some random offsets. What this means is that nowhere do we store position-centric data – everything is global and combined to create the same planet on everyone's machines. We wanted to introduce a solution to this in the form of the terrain modification system. After finishing work on this, I was able to start work on rivers, which fundamentally depend on being able to deform the terrain. In terms of approach, an incremental, iterative approach is the best way: First, I worked on erosion paths and a basic placement tool without actually modifying the terrain at all. Then I created a basic terrain modifier to form a trough and placed these along the river path. From there, many, many iterations have produced what we see today. When working on an already complex project, it's important to keep in mind the engine as a whole and how best to utilize already existing tech. For example, the BiomeBuilder is responsible for distributing assets and already handles the complex spawning and de-spawning of objects as streaming bubbles/players get closer or move on. Therefore, using BiomeBuilder to do the same for the river water meshes made sense rather than reinventing the wheel to detect when river meshes need to be created/destroyed. There are many examples of this within river tech.

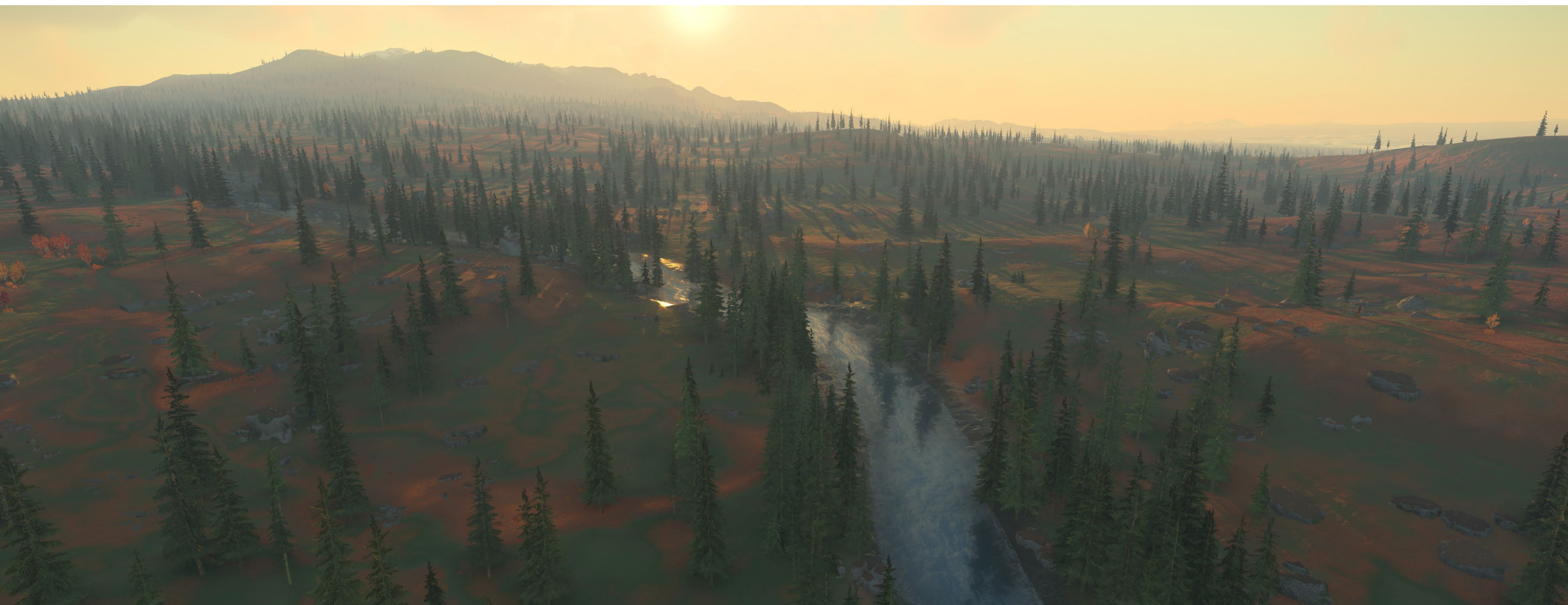
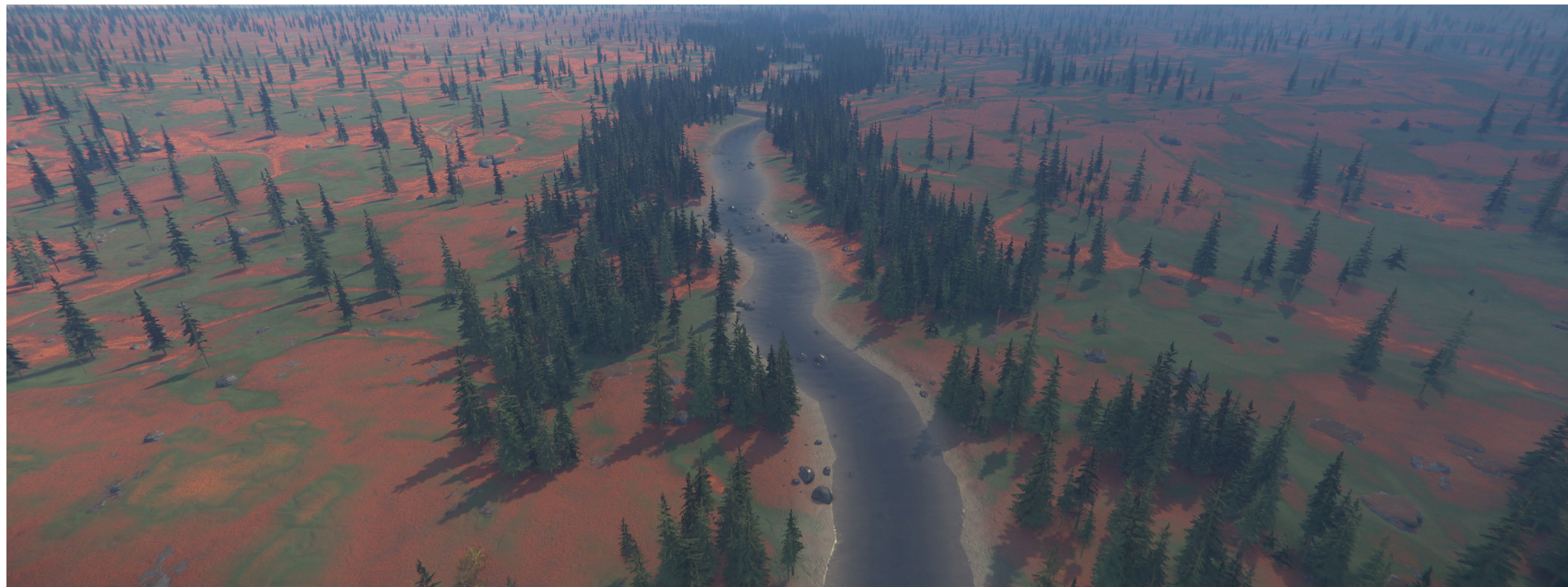
JP: *Were other disciplines involved in making rivers happen? Did additional art, audio, etc. need to be generated for this update?*



WH: One of the biggest mistakes I made producing river tech is not involving other tech earlier – people knew about it but there was no coordinated effort to get other tech compatible with it until more recently. Other disciplines have come in now (to do the audio implementation, for example) but as a result of the work being created so late in the development schedule, I've personally been responsible for making the grav-lev work over water, making sure ships correctly explode when going under a river or basin, making sure the actor tech knows it's underwater, and breathing/o2 tank refilling can happen appropriately. Even still, I hope the actor and VFX teams are able to find time to implement more specific work for the rivers, as it will make a huge difference once we have splashing, mist, and wading/swimming animations.

JP: *We understand the qualities of a spacecraft at this point, such as what systems need to work together to make a given one work... but what are the defining factors of a river? Do you have to take into account currents, sediment, volume of water?*

WH: When calculating the path of a river, the most important parts are volume, velocity, and direction. The system aims to find the path of water over the existing landscape rather than full hydraulic erosion, which would create the landscape. Our heightmap tiles have already got offline baked hydraulic erosion, so we want to avoid undoing that work with rivers – ultimately terrain modification will never look as good as the handcrafted heightmaps, so we want to preserve those as much as possible. Most of the time the water flow system



ends up finding the water paths that emerged from simulation in the height maps and then deals with how the water traverses from one heightmap to another on the actual surface of the planet. While there isn't any data baked into the heightmaps about sediment types etc., this is something I'd love for the rivers to be affected by going forwards. Different sediment and rock types can affect the ratio between depth, width, and velocity in a river, which would create more variation along with waterfalls and gorges. For the most part, the depth:width:velocity ratio is fixed in the current implementation, but there is no reason this couldn't change in the future and it's definitely something I'd like to investigate.

JP: *How much prototyping happens to get to the new system working? Did you go down any paths that didn't work out as intended?*

WH: In terms of prototyping, in agile development we're always trying things to see if they work. It's one of my favorite ways to work – rapid prototyping and testing. In terms of tested ideas that didn't make it into this release, we've looked at writing directly into our climate data for the river, changing ground type on the planet shader, a much heavier reliance on decals for coloring the edges of the river, and backwards erosion up from a desired ending instead of downwards flow. Some of these may be revisited in the future and some may never be revisited. Many of the finished features started as an exploration of 'does this work' and 'does this produce good results,' etc. For example, the On-Demand-Spawn-Points started as an afternoon trying methods to increase density around the river. These were then rewritten to become a permanent addition to the river tech.

JP: Was there anything you needed to cut that you wished could've been included?

WH: I really wanted to get the phys areas and flow in for Alpha 3.17. This would mean things could float and flow down the river. I'm hoping this will make Alpha 3.18, as it'll really spice up the gameplay opportunities around the river. This isn't my court, but I also wish we had swimming/wading. However, the Actor team's current work is much more important, so I look forward to when they do get to this.

JP: Is there anything that are you're especially proud of in the finished system?

WH: I'm proud of all of my work on the current system but if I had to pick something I'm most pleased with, it'd be the water mesh building. It achieves something ultimately simple, but I'm very happy with how it's implemented and the results it gives. I'm also very happy with my work on BiomeBuilder, which ultimately feeds into the overall look and performance of the rivers. Finally, I'd say the most complex math involved is the river spline terrain modifier – it uses the perpendicular distance to a Bezier curve to feed a parametric quadratic equation to form the river trough. It's probably been the part I've iterated on most.

JP: Are there plans to continue working on it?



WH: This is by no means the final form of rivers. As well as improving the placement tool to allow for automatic distribution of rivers across planets, I want more features such as chained basins/lakes, sediment considerations, waterfalls, forks/joins, VFX, more complex audio, a better water shader, support for different fluids, etc. The list can go on and on!

JP: *Are you allowed to say what you're working on next? Can we expect other similarly detailed planetary features?*

WH: My next tasks are mostly centered around expanding our river tech, but in the pipeline are all sorts of exciting things. I talked about the new foliage shader in my CitCon presentation – this is still ongoing and will help convert our planets to be fully seasonal. We also want

to improve the physical accuracy of the scattering on our planets to create more varied biomes and locations.

JP: *Do you have any messages for the players who are about to explore your work? As of this writing we hear the Evocati are already passing around coordinates to explore a river!*

WH: To the people about to see the rivers for the first time, I hope you enjoy the search, as well as the destination! Remember to keep your helmet on if you're going diving, and the fastest way to get from one end of the river to another is by grav-lev. I would personally love to see the fastest time to race all 12 km of it!

END TRANSMISSION

GALACTAPEDIA

TRIGGERFISH

Triggerfish is a Human holiday celebrated annually in the United Empire of Earth (UEE) on April 1. Those who observe it play pranks or participate in hoaxes, shouting "Triggerfish!" once the joke is revealed. Though the holiday can trace its historical roots as far back as the 16th century, it began to be called Triggerfish in the 2600s, when it became tradition for residents of Borea (Magnus II) to try and convince visitors that a fourth planet named "Triggerfish" existed in the Magnus system. Some outlaws would give victims coordinates to the planet, only to ambush them once they arrived. To put an end to this dangerous practice, the UEE launched an information campaign refuting the claim that Triggerfish existed, which spread the name outside the Magnus system. Triggerfish became a nickname for a gullible person, and it wasn't long before the name became the most common way to refer to the holiday.

ORIGIN

In 2653, siblings Bashir and Hija Orchard were celebrating the birthday of their cousin Yusef Orchard at a bar in Odessa. After hours of drinking and dancing, Hija and Bashir pulled their cousin aside and started regaling him with the story of a planet they'd discovered. They claimed that it existed just beyond the orbit of Magnus III, and the reason it hadn't been discovered yet was that the UEE Navy had been covering up its existence. After the Navy pulled its center of shipbuilding out of the Magnus system, Bashir and Hija explained, the coverup ended and they were lucky enough to have stumbled upon it. Hija added that they were only sharing this info with him because he was family, and that they'd named it Triggerfish after a stuffed animal she and her brother had shared as children.

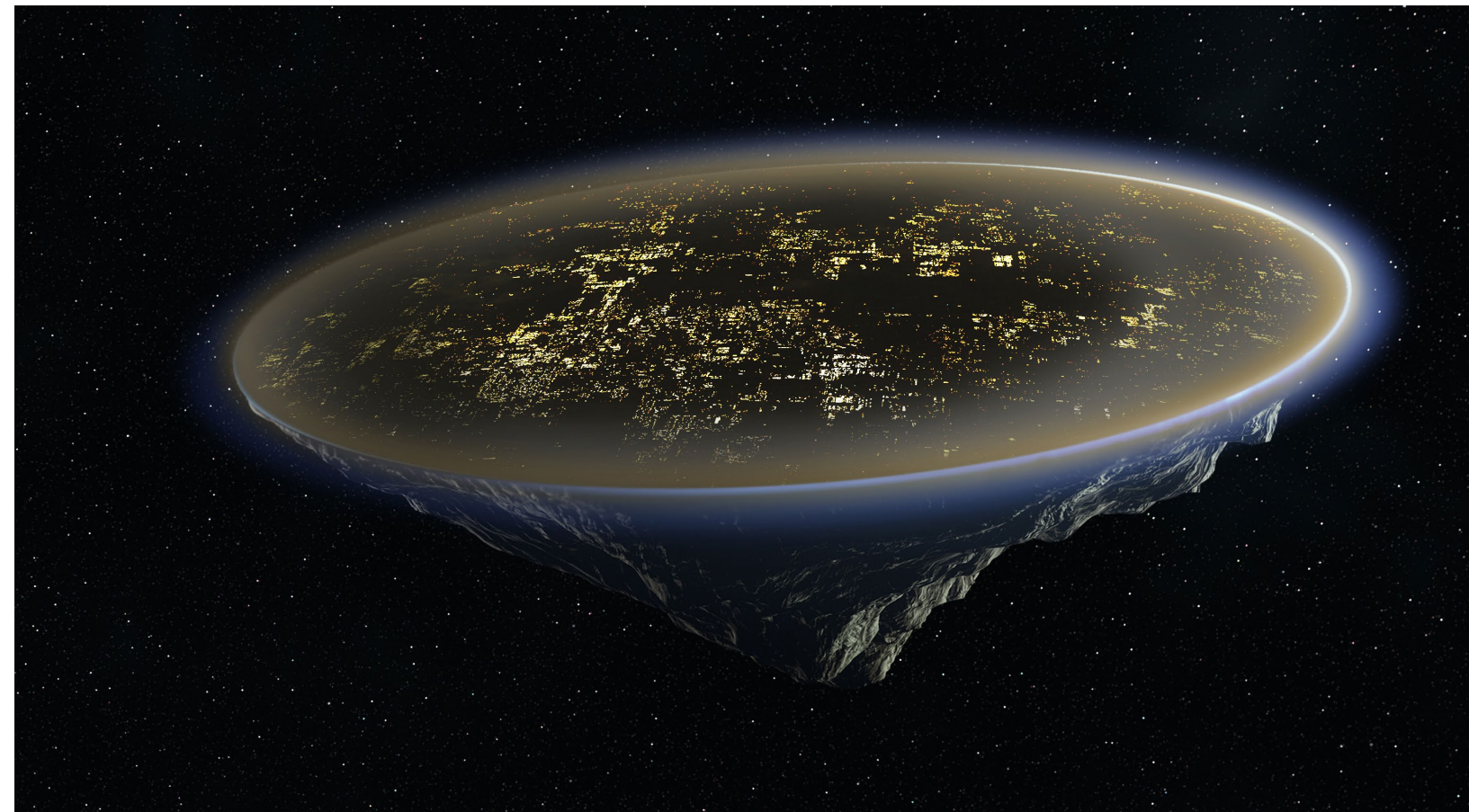
Yusef was tremendously excited by the prospect of this new planet and wanted to see it immediately. Hija and Bashir took their cousin from his

own party and made their way to the nearby hangar where Yusef stored his personal spacecraft. Bashir entered the coordinates into Yusef's nav system, and the trio took flight. Hours later, Yusef still hadn't located the planet. Hija discreetly recorded his increasing frustration as Bashir encouraged him to fly just a little farther. Eventually, Yusef caught on that he had been tricked and launched into a long, profane rant while his cousins laughed hysterically. The resulting video made its way across the Spectrum and sparked copycat pranks. By 2667, the Triggerfish prank had become common enough knowledge on Borea that it could only be played successfully on tourists.

CELEBRATION AND TRADITIONS

Today, the holiday is generally a good-natured affair between friends and colleagues where pranks are punctuated with shouts of "Triggerfish!" There's a healthy industry of products sold specifically for the day, including various trick crates marketed to haulers. Meanwhile, practical jokes that affect the space readiness of a ship are considered inappropriate and dangerous.

There have even been some incidents where Triggerfish pranks have turned deadly. In 2801, independent miner Harry Jasym's ship was discovered floating in the supposed location of Magnus IV, his frozen corpse trapped within. When authorities examined his mobiGlas, it was discovered that a group of local miners had met the newly arrived Jasym at a pub and decided to welcome him by sharing stories of the fabled riches to be found on "Triggerfish." Jasym's ship logs contained a series of entries showing his eagerness to discover the uncharted world and his growing panic as his older ship suffered from a host of mechanical issues. The remoteness of the location meant his attempts at contacting help went unanswered.





CLARK DEFENSE

S Y S T E M S

Despite an illustrious career, Gratia Katsaros never forgot her first in-field assignment. A recent Rhetor University business school graduate, Katsaros rejected job offers from prominent investment banks to join a crew of “turf-techs,” 24th-century slang for workers hired by terraforming companies to visit planets being actively terraformed. A mixture of scientist and repair technician, some turf-techs took important readings to track the process while others fixed essential ground-based equipment. The work wavered between exhilarating and extremely dangerous, which Katsaros, as an amateur climatologist and thrillseeker, admitted attracted her to the gig. While suiting up for her first mission, Team Leader Clark Rissolo handed her some extra homemade protective gear and explained that the company didn’t require that she wear it, but he did. Clark was a lifelong turf-tech who knew how quickly conditions could change and how ineffectual the standard environmental suit could be against unexpected elements like egg-sized hail suddenly falling from the sky.

In her memoir, *Protection for the People*, Katsaros revisits this first gig on Terra IV and some of the dangers her team faced.

“There was a sudden drop in air pressure followed by a cold, stiff breeze. It didn’t seem that strange to me, but Clark immediately called for an emergency evacuation. I took a moment to tighten another bolt when, suddenly, I was knocked down by a rock whipped up by the wind. Thankfully, Clark pulled me up and guided me into the ship. Once safe, I looked down and saw a rock embedded into Clark’s extra layer of armor. For that, I owe him not only my life but my eternal gratitude for setting us on a new path.”

Katsaros jokingly called these extra protective layers “Clark’s defense system,” and her experiences on the job pushed her to advocate for increased safety standards for turf-techs. In 2539, she convinced Clark and a few others to start a company manufacturing specialized

environmental armor. The team established their operation in Najita on Keene, Killian system, and picked the name Clark Defense Systems (CDS) in honor of their inspiration. Yet this slight name change wouldn’t be the only early adjustment the company made. Shortly after launching their inaugural line, the First Tevarin War erupted and the UPE desperately needed combat armor. Sluggish sales of their environmental suits drove CDS to apply for and land a military contract. “We jumped at the chance to expand our product line thanks to some government funding,” Katsaros wrote, “but believed we’d be back to environmental suits the moment the war was over. We couldn’t have been more wrong.”

GOOD ENOUGH FOR GOVERNMENT WORK

Clark Defense Systems refined their environmental suits to comply with military standards and quickly converted their production line to create a

variety of field armor and combat flightsuits. With the UPE on the verge of winning the First Tevarin War, a cadre of government officials visited CDS headquarters for the first time in 2546 and disclosed that their armor was the most well-regarded equipment among combat forces as well as pilots. They were so effective and well-liked that much of it was reserved for special forces or soldiers in the most dangerous areas of operation. The visit ended with UPE officials offering CDS a massive new government contract to expand upon their output to the military. Though the offer would keep them from being able to return to the environment suit market, it was too good to refuse.

CDS dedicated all the company’s resources to manufacturing combat armor and consulted extensively with veterans of the Tevarin War on what features they responded to best in the field as well as what they would need and want for future models. This led to the development of some

of their most famous lines of armor, including the Omni-Role Combat (ORC) and Advanced Defensive Protection (ADP) series. The outbreak of the Second Tevarin War in 2603 brought about another substantial increase to CDS' government contracts and further real-world testing. The company's willingness to adapt and improve their armor kept it as a favorite of soldiers who, when the war finally ended in 2610, would return home raving about a brand unavailable on the civilian market. This word of mouth created a bustling secondhand market for CDS armor, with entire companies being created to procure and refurbish the prized pieces.

With demand so high, CDS considered pivoting to the private sector. The company's long-term commitment to making armor exclusively for the UPE/UEE military was a boon, but company executives understood the dangers of being associated with the increasingly authoritarian Messer regime. CDS lawyers looked into ways to get out of the exclusivity clause in their government contract, but the Messer regime got wind of their plan and informed the company that their armor was of "strategic importance to the Empire." The UEE argued in documents released centuries later under the Historical Truth Act that none of the armor developed for the military, or even modified versions of it, could ever be publicly released and that any attempt to backout of their manufacturing commitment would lead to the UEE taking full control of the company, its facilities, assets, and patents. Sufficiently scared, CDS officials dropped their plans and sat back, waiting and hoping that the company could outlast the regime.

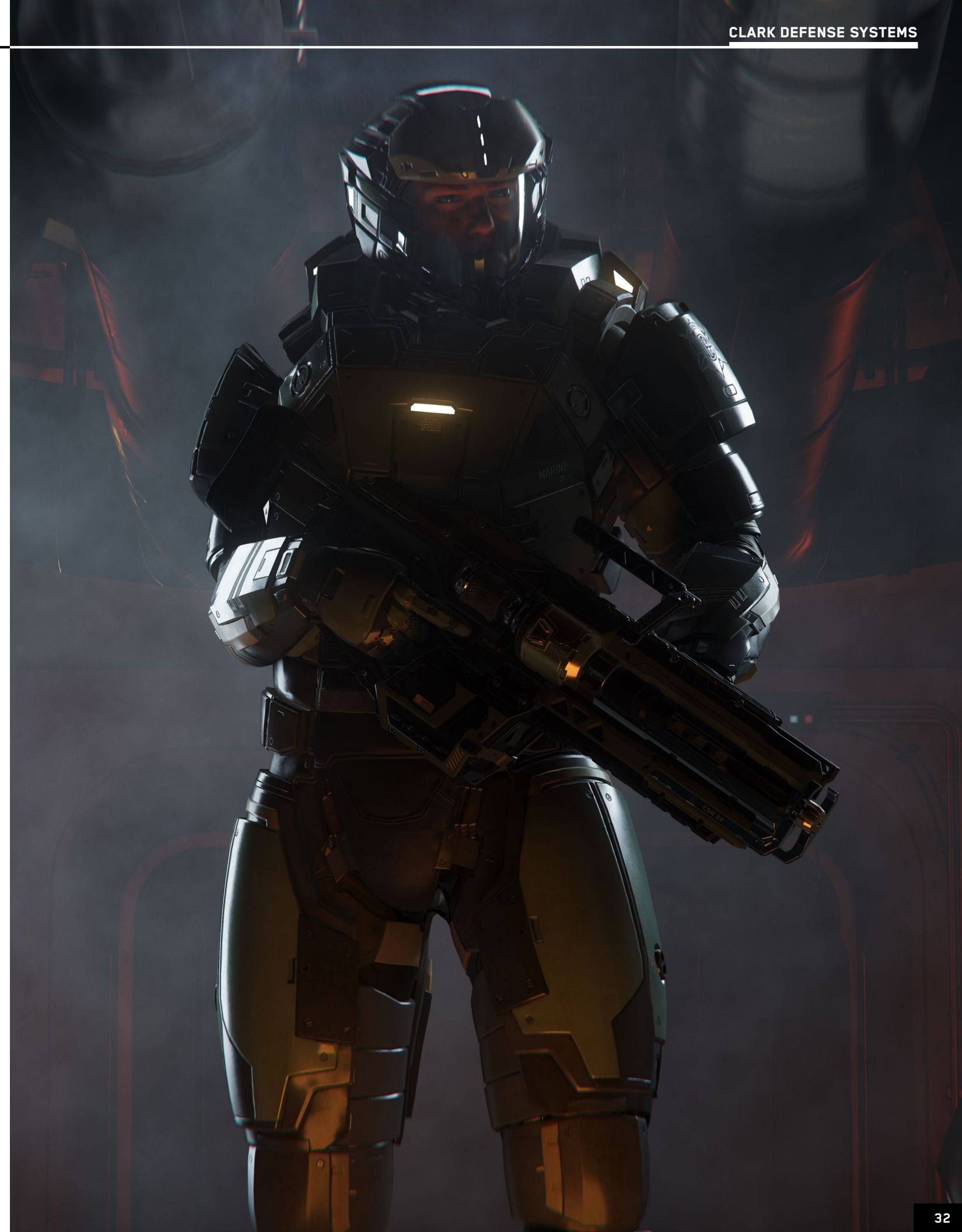
PROTECTION FOR THE PEOPLE

By the late 28th century, Messer regime power was weak enough for CDS to resurrect their plans to expand into the public sector. Their first

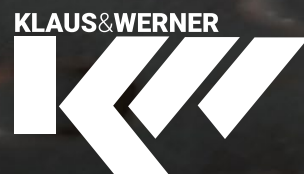
move came in 2789 and coincided with the former military system of Castra being opened to the public. The company bought office and manufacturing space on Cascom (Castra II) with the hopes of creating a separate company focused on developing armor for the public. CDS hoped that siloing this entity, its product, and development cycles from anything previously done for the military would shield them from government interference or retaliation. But these plans quickly became moot when, in 2792, the Messers finally fell.

Free from the iron-grip of the former regime, CDS held lengthy discussions with senators and members of Imperator Erin Toi's administration. The new government wanted to cut ties to companies associated with the Messers, a fate most famously dealt to Aegis, but the evidence of CDS's attempt to sever their contracts and the regime's subsequent takeover threats saved them from losing everything. Instead, the two sides agreed to dramatically reduce the size of CDS' contracts but allowed them to enter the civilian market to sell legacy versions of their combat armor.

Before entering the civilian market, CDS enacted a master plan to set up the brand for future success. In 2802, the company officially moved their headquarters to Cascom, Castra system, and significantly increased their manufacturing capacity there. Today, most of their civilian operations are based out of Cascom while their facilities on Keene focus on supplying and developing the next lines of groundbreaking armor for the UEE military; an area where CDS continues to excel as evidenced by them being awarded a government contract in 2945 to research and develop next-generation light armor. Today, Clark Defense Systems maintains a significant foothold in both the civilian and government sectors and is considered by many to be a cornerstone company of the empire.



Scrollwork Showdown



Last year, the master gunsmiths at Hedeby Gunworks and Klaus & Werner entered into a friendly bet over who could design the best scrollwork pistol. Now you can reap the benefits of this good-natured rivalry with three new finely crafted firearms featuring distinct and intricate engravings. While the best design might be in the eye of the beholder, the real winner is anyone who owns these stunning showpieces almost too beautiful to holster.