JUMP POINT

ISSUE: 06 09

IN THIS ISSUE →→→

Editor: Ben Lesnick Copy Editor: Martin Driver
Layout: Michael Alder

FROM THE COCKPIT

GREETINGS, CITIZENS!

You've made it through the Jump Point! We've got a great issue for you this September as we take a deeper dive into the phrase that's on everyone's fingers right now, Object Container Streaming. OCS is a big deal for Star Citizen's future and is going to make some very big things possible.

This month's Behind the Scenes article covers the brand-new Crusader Mercury Star Runner, which is definitely one of my favorite recent ships. I think you'll enjoy seeing just how much work went into getting that asymmetrical look right. I was certainly impressed!

September is also the time of year we start looking forward to CitizenCon. By the next issue, the event will have come and gone… so we're at peak excitement level as I write this! CitizenCon is coming home to Austin, Texas this year and I couldn't be more excited. I remember the very first event back in 2013, when we had a small theater full of developers and a few-dozen incredibly dedicated backers (I have no doubt some of you are still reading this, chime in if that's the case!). We shared a little presentation and movie and hung out afterwards. It was a wonderful night that really let us get to know the people making the game possible… but we couldn't have predicted how big it would get!

You've seen CitizenCon get bigger and more robust each year and this is going to be another evolutionary step. CitizenCon 2948 will give 2,000 attendees (more than twenty times as many as attended the first one) a full day of Star Citizen-related excitement. We have panels, presentations, activities, chances to meet developers and to see what's coming next for the game… and of course plenty of opportunities to meet the wider Star Citizen community. There's absolutely nothing in the world like sitting in a room packed with excited backers cheering on a big presentation and it's so cool that more people than ever are going to have a chance to do it this year. Being right there with people who share your particular passion… it's thrilling. But whether you're making the trek to Austin in person or tuning in online, get ready to have some fun… it's shaping up to be a good one (I'd say more, but I'm sworn to secrecy)!

I hope you enjoy this month's Jump Point. We'll be following on from CitizenCon in October's issue, so keep sending your feedback and let me know what you'd like covered in the future. There's plenty more Star Citizen to explore!

Until next month, I'll see you in the 'verse.

Ben

JumpPoint@Cloudimperiumgames.com

OBJECT CONTAINER STREAMING

If you've been following *Star Citizen* for a while, it's likely you've heard of Object Container Streaming (OCS). While it's a major technical undertaking that touches on almost every aspect of the game, it's much more difficult to follow and understand compared to a new ship, planet, or gameplay mechanic that you can experience first hand.

So, to catch everyone up, we spoke to the team working on OCS to clarify what it is, what it does, and where it's taking *Star Citizen*.

Editor's Note: Because the OCS team is hard at work on the feature, we were not able to conduct our usual roundtable interview. Instead, we spoke to stakeholders individually to help us get to the bottom of it. Thank you to Producer Luke Hale for helping us answer all of our questions!

BEGIN TRANSMISSION →

JUMP POINT: *We've heard a lot about OCS over the past year. Let's clarify once and for all, what is it?*

OCS TEAM: OCS is a major system that's a key part of the foundation of Star Citizen as our 'verse expands to include more planets and solar systems.

JP: *In broad terms, how does it all work? What's an object and where is it being streamed to?*

Think of a computer's memory as an empty tub. The more available memory, the bigger the tub. As you play a game, objects you interact with are taken from storage and then dropped into it. Objects, in this case, mean things like level maps, 3D objects, textures, audio, pre-rendered video… all the individual parts that make up the game experience. It takes a little time to move objects into the tub, which is what you experience as loading. As long as the object you need has already been dropped in, though, you can make use of it immediately. OCS takes these objects and puts them into the memory to allow the game to use them as soon as needed. When they're not needed anymore, it takes them out to make room for more.

JP: *Why not start with all the objects in the tub at once? Load them all at once and leave it at that?*

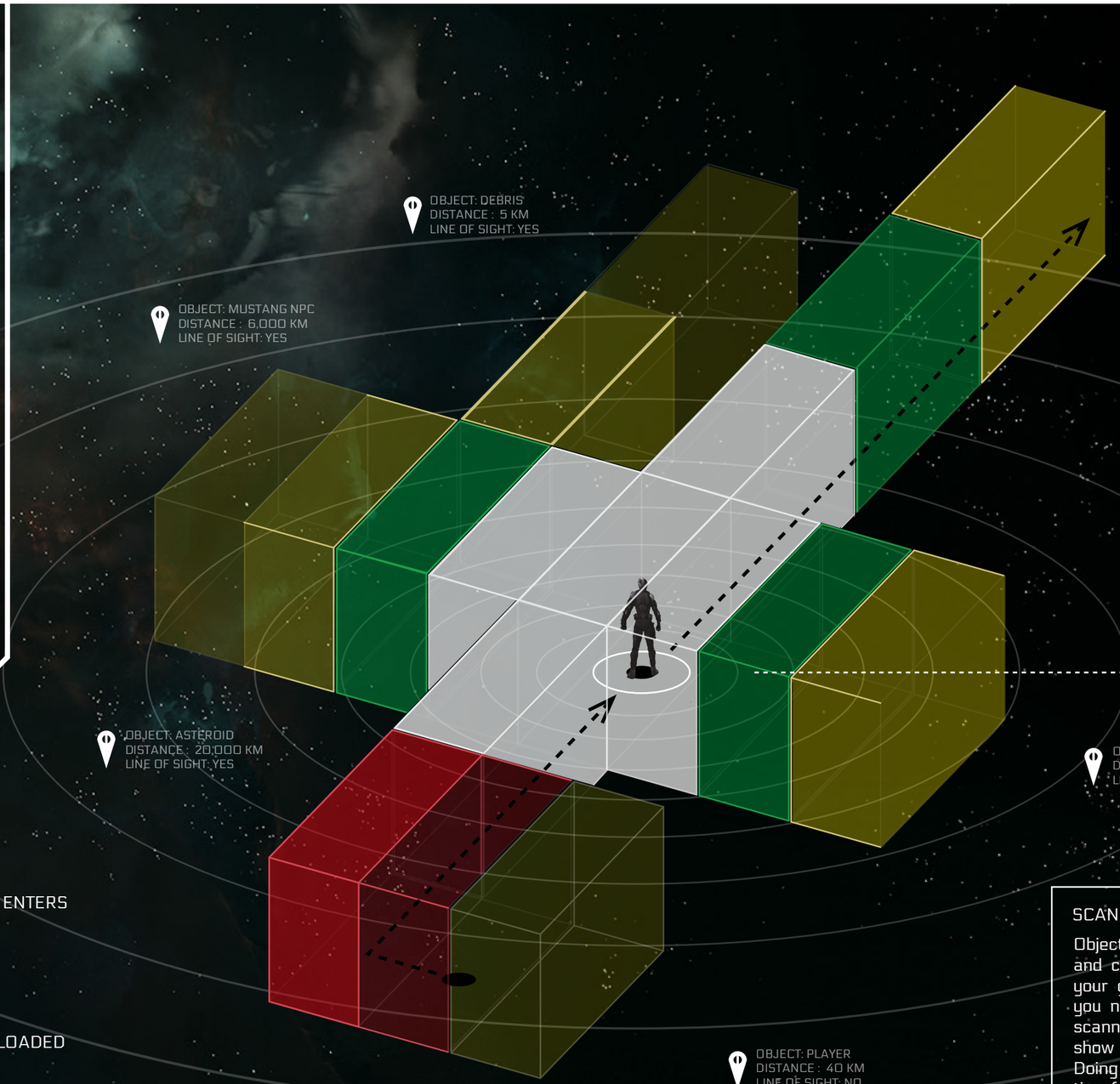
That's how the Persistent Universe works right now… and we're already hitting the limit of what's possible. Computer memory becomes more plentiful and processes become more efficient with every new

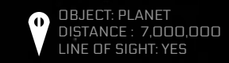
OCS

OBJECT CONTAINER STREAMING

As Star Citizen expands with an ever-increasing number of planets, locations, and ships, the demand on the player's computers grows. So, we created a feature to ease the strain and allow us to continually expand the universe as planned.

Currently, all of the game's assets load whenever a player logs in. While this works for smaller games or those willing to utilise loading screens, it doesn't suit our huge Persistent Universe where an individual's actions have lasting effects on everyone else around them. As of Alpha 3.2, the 'verse is at full capacity, so while we're working on new locations for players to explore, there's no room left to implement them. Object Container Streaming fixes this by removing the reliance on computer memory and changing the way the world loads.





AREAS PREPARED TO LOAD BEFORE PLAYER ENTERS

AREAS LOADED FOR PLAYER

EXITED AREAS WILL BE PREPARED TO BE UNLOADED

AREAS CURRENTLY ON STANDBY

HOW IT WORKS

Rather than load everything when you log in, OCS intelligently cues up the assets you need as you need them. When you leave an area, it unloads all the unused assets to make room for new ones.

It works with NPCs and other players too; when they're no longer within your sightline, they're unloaded. However, they still exist in the universe. They don't disappear, their assets just won't load in your game until you can see them again. If they destroy something, crash, or affect something else, you'll still the see the aftermath if you follow their path later on.

All of this is seamless and you won't notice it at all while playing.

SCANNING

Objects, ships, and characters out of sight still exist and can be scanned. Rather than be fully loaded into your game, the game will only load the information you need to see until you get closer. For example, scanning an unknown ship in the distance will still show its details way before it enters your sightline. Doing this means NPC and player ships only use a tiny fraction of your computer's memory until they need to be fully realised.

OCS detects where the player is heading and pulls in relevant entities. When they leave or have no use for them, they're unloaded to allow others to take their place.

generation of technology, but the objects that make up games increase at a greater rate. Sure, if your game is small enough, you can make everything available immediately, but Star Citizen isn't a small game and it's only getting bigger! We need to be able to store content on servers (bigger tubs that are connected together) and then move it to the player before they know they need it. That's why we need to build in dedicated systems that let us manage those objects. OCS is our system for determining which objects need to be streamed to which players at which times, all while avoiding loading screens and accounting for a whole lot of dependencies.

JP: *How does OCS tackle this problem?*

Object Container Streaming breaks the game into smaller chunks so that we always have what the player needs available, while also letting us continue to expand the size and complexity of the game world in other directions. It helps us kill loading screens and make the Star Citizen experience seamless. With OCS in place, the game detects where the player is heading and pulls in all relevant entities.

JP: *Is OCS a new idea, something on the cutting edge that we're pioneering?*

Object Container Streaming is not unique to Star Citizen. Every game that approaches a certain level of complexity faces the same problem - there's more content to be delivered to the player than can be held in the computer's memory at any one time. In the 'olden' days, games were organized into separate levels or regularly required the player to sit through a loading screen while the data was brought to bear. Today, games address the problem depending on their nature.

The most common solution for multiplayer and massively multiplayer games is to try and address the issue from a level and art design standpoint: build leaner levels and smaller objects that all players can keep in memory permanently. However, this won't work for Star Citizen, as building realistically-sized space stations, moons, planets, and star systems means we need very large maps available to many players pretty much all of the time.

The client-side work we're doing today is more akin to open-world single-player games, with processes for knowing what the player is doing and accurately estimating what they will do next. This

information allows the game to stream content that will be needed while dropping anything that won't appear again and enables modern single player games to have huge, seamless maps. We're just taking the tech a bit further by using it to build an entire universe.

JP: *I've seen games where things visually 'pop in' once they're close to the player. Is that what we're talking about?*

No, our solution needs to be much broader and our system needs to be able to handle many cases beyond whether or not the player is close enough to see a particular object. If anything, the easy part is knowing whether a ship, character, or even a whole planet is close enough to see. In a single player game, that alone would be fine… but, with Star Citizen, we have a vast galaxy inhabited by multitudes of players whose actions will constantly impact one another. Even if you can't see a ship, character, or place at a specific moment, it may still be impacting your game in some way.

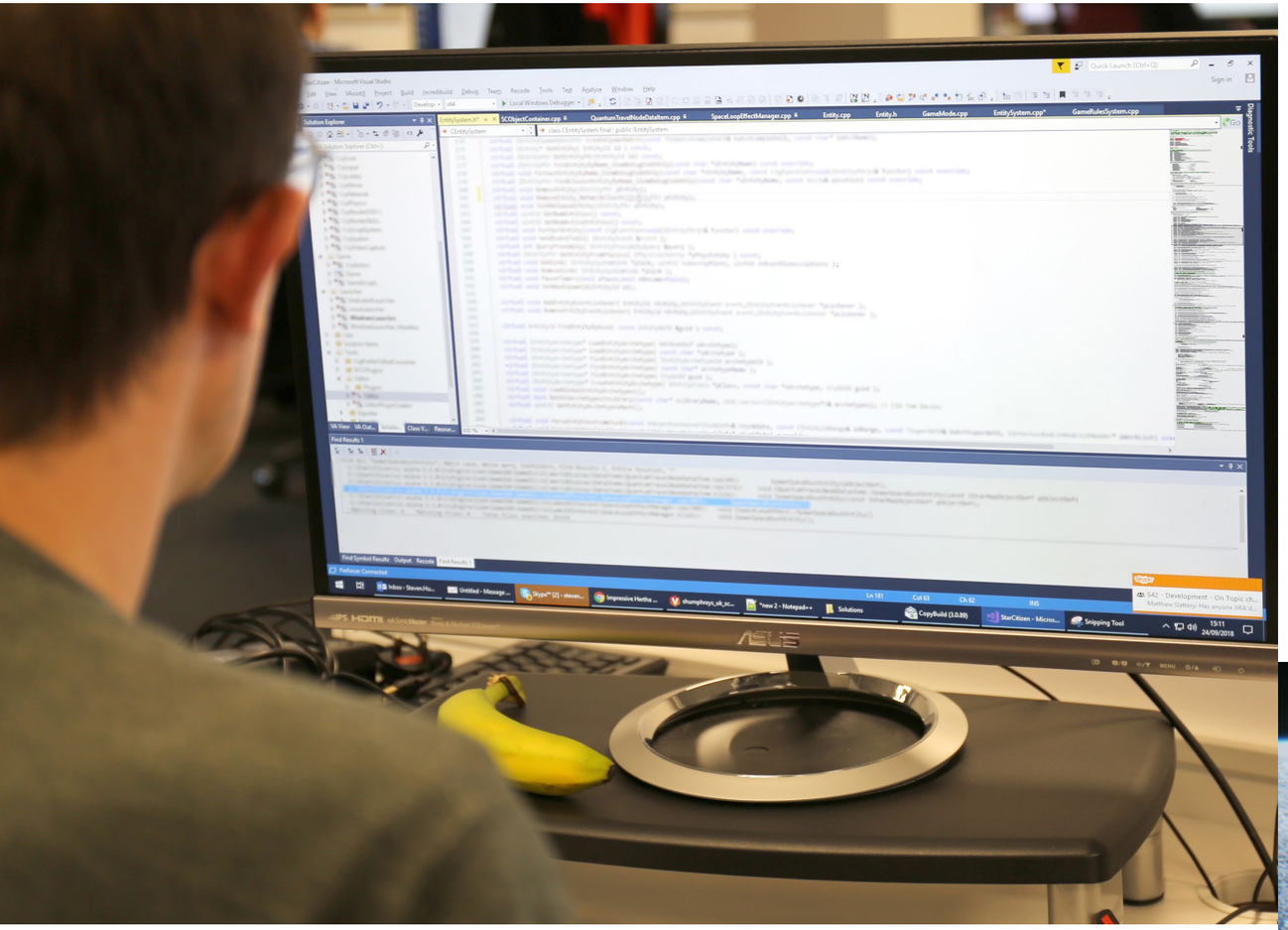
The challenge is to build a system that takes into account the million cases that aren't immediately apparent: How do we handle scanning ships outside of visual range? How do we handle a mission that assigns you to a marker three systems away? And most importantly, how do we handle several players in the same area impacting one another in ways we can't predict?

The answer to these questions is something else you might have heard us talk about: serialized variables, which the game uses to communicate internally. Every 'piece' of the game has values that change over time and form a sort-of remote control over the state of game objects. That helps inform us of more distant changes that might impact the player.

Another big part of this is making the process of loading entities asynchronous to reduce the time they take to load. We're changing the fundamental network messaging that handles spawns. Right now, the old system spawns something immediately using the main thread. So, if you need to load a whole planet, it could take a noticeable second or two because the game processes these requests. Once we're asynchronous, objects can be readied in advance to stream in more quickly without interrupting anything; think of it like meal prep for starships. Right now, you have to collect everything from the cupboard, but soon the game will have all the ingredients measured out in advance.

JP: *Can you give me a real-world example of how it all works?*

Right now, Star Citizen loads everything at once. With OCS, the player might start off with only their initial location in memory. So, if you start on Grim HEX, only Grim HEX is in memory. As you leave your quarters, the game knows to stream-in the area around you, including the level geometry, other players, docked ships, and more. When you call your ship, it starts streaming the area around the planet and as you move into space, it unloads your quarters and anything not needed on the planet's



surface. Once in space, anyone Quantum Traveling towards you has their entity loaded in and anyone traveling away is loaded out. All of this reduces the initial load time for the game AND the overall footprint.

JP: *How long has the OCS system been in development?*

Work began in earnest about three years ago. Our overall plan has three parts. The first of these premiered in alpha 3.1 with serialised variable culling, which is basically the foundation of OCS. It's the system by which the game decides what is and isn't close enough to drop from memory. Next, we're hoping to premiere the client-side OCS, which gives each individual computer the ability to decide what needs to be in memory as described above. Right now, the server loads everything that's needed in the Stanton System to every player in a particular instance all the time. So, every ship model being flown, every portion of planetary surface, every audio clip… the whole bag. Once we premiere OCS on the live branch in the future, that will no longer be the case.

JP: *Does that mean we can expect to see a huge performance increase?*

That's a very tough question to answer and I want to be cautious here. It's true that our testers have noticed some great improvements with regards to server-side frames-per-seconds (FPS). But, it's very important to understand that this is NOT a polish pass aimed at making the game run more smoothly. Instead, it's the key to letting us do MORE in the game world. While we'll see an improvement as client-side OCS comes online, we're also adding a lot more to the game and plan to continue to do so for the foreseeable future. Therefore, the new system is much more about letting us give you a game world with more ships, more planets, more star systems. It's really a microcosm of the history of game development I referenced earlier: as soon as resources are freed up the priority becomes to make a bigger, better, and more immersive game.

JP: *You said we're on step two. What's after client-side OCS?*

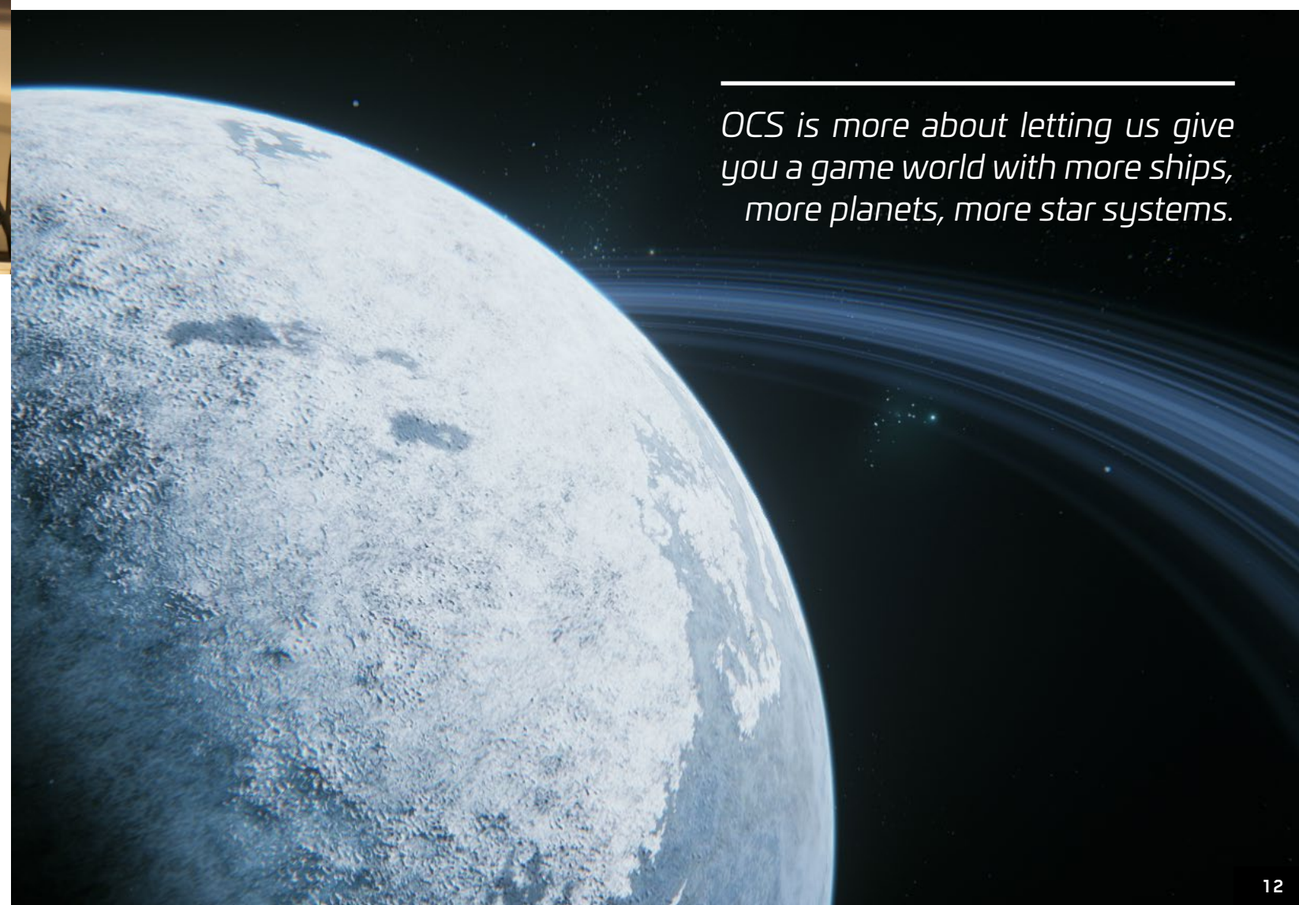
Step three is server-side OCS, which will come online much later when the overall shape of the game world is closer to being finished. Server-side OCS is an essential part of what will eventually let us use multiple servers to create one cohesive game universe for all players. The work we're doing today is foundational for that. In short, we know what we're doing, we just have some dependencies to work out before we can start that part of the work.

JP: *What's the biggest challenge to building this system?*

The nature of the thing. Star Citizen is already a very complex game, so a system like this ends up touching almost everything that has come before it. A major part of our work has been preparing existing game elements for OCS, dividing content into logical containers and teaching the game how and when to stream them. We're turning Star Citizen's existing world into a sort of nesting doll of object containers. On the largest side, you have the Stanton System which encompasses many smaller containers that hold moons, asteroid fields, and the like. Then, you can go down several levels to individual ships, locations, NPCs, etc. The other very difficult aspect is production. 'Producer' is an often-misunderstood role outside the industry and one that doesn't get enough credit for what's contributed into building even a regular feature. In the case of OCS, producers are absolutely essential for keeping things planned and organized. We have tasks that might have two, three, or even four dependencies, meaning tasks given to other teams that need to be completed for our tasks to begin. Our producers keep a very complicated machine humming along nicely.

JP: *How do you test it works?*

The challenge in doing QA work is that we really need to test unusual, extremely complex situations. It's easy to imagine all the use cases



OCS is more about letting us give you a game world with more ships, more planets, more star systems.

Everything in future releases will only exist because of OCS. So, think of it when you're landing in Area18 or blasting off in an 890 Jump.

when there are one or two players in an area, but Star Citizen allows many more complicated scenarios to happen at any time. What if four players are in the same container and one drops a cargo box? The other players need to see it immediately, of course, but there are also any number of ways they might leave and come back or otherwise impact what's going on. If one damages the box, then it needs to appear the same way to everyone no matter when they revisit it. To make testing things like this possible internally, we use 'headless' clients, which are computers running the game in certain ways to help us help simulate multiple players in the same container.

Of course, we're eager to move from headless clients to real, live players. OCS is another system that's going to benefit greatly from letting thousands and thousands of dedicated backers in. As with everything else, we're eager to see what happens when theoretical work meets the real world.

JP: *What will those players 'see' when OCS goes live?*

Players won't necessarily 'see' anything apart from the new locations and features made possible by OCS. When Alpha 3.3.5 goes live, every new location and feature is the result of OCS doing its thing, so the first *things* players see will probably be Hurston and its moons.

All content in future releases will only exist because of the 3.3.5 update. So, think of OCS when you're landing in Area18 or blasting off in an 890 Jump.

JP: *When do you aim to release OCS for the community?*

We're eager to have it out as soon as it's ready. We're releasing Alpha 3.3 first to give the community access to the new ships and features that don't rely on OCS.

OCS will come in Alpha 3.3.5, which we'll push out to the Evocati for testing as soon as we can. Keep an eye out, it's not far away!

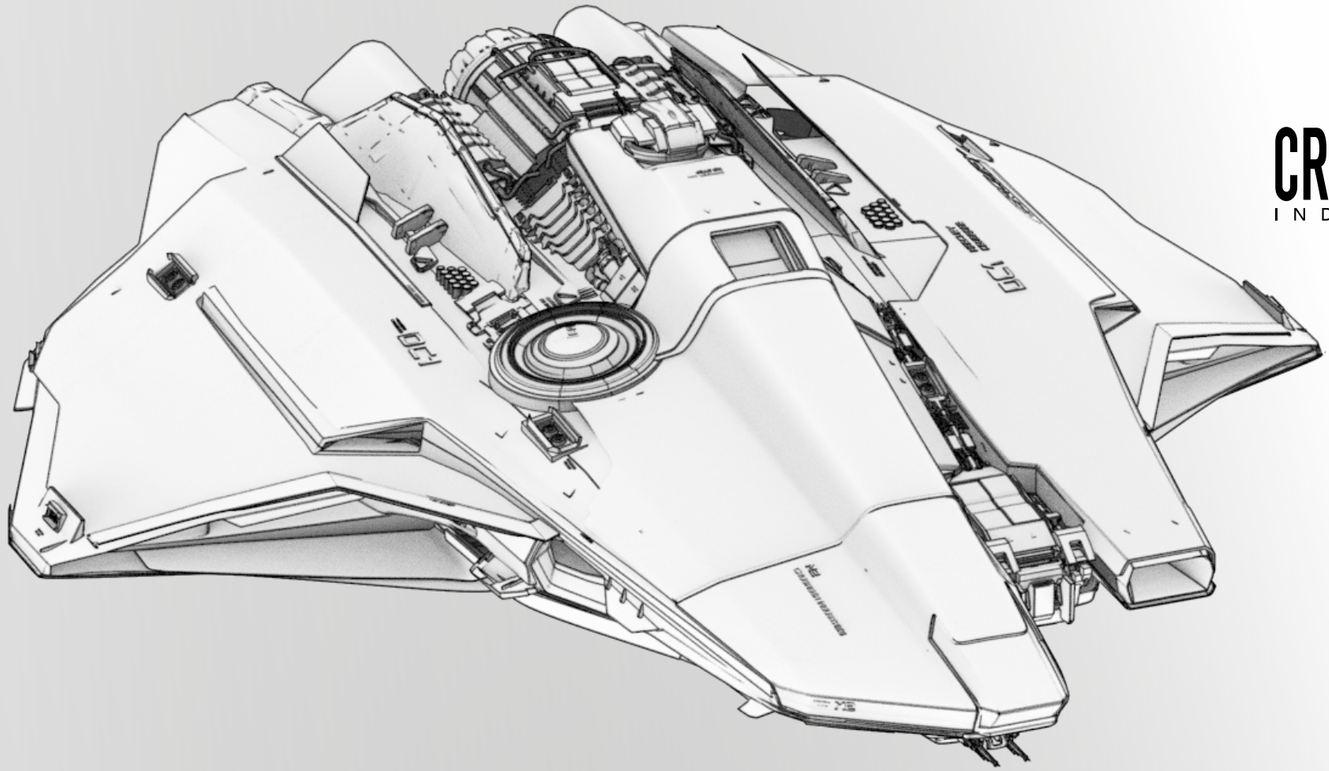
JP: *Do you have any final words for the community?*

We want to thank everyone for their support and their patience. It goes without saying that the community is why we're here. A system like OCS isn't something a normal game developer would ever talk about in the first place because it's so 'under the hood'… it says a lot about the Star Citizen community that we have backers out there trying to understand all this work and cheer us on. It means a lot and it truly makes this feel like a collaborative effort. We can't wait to let you see what we've been doing.

It's a cliché, but OCS is the thing that gets you to the thing. It enables Star Citizen to get bigger and to achieve what we set out to do early on. Getting it live is an exciting step forward and we're glad to have the community along as part of that process.

END TRANSMISSION ←

WORK IN PROGRESS... CRUSADER MERCURY STAR RUNNER



AIMS

- Star Runner - high-speed data/cargo courier with exceptional planetary takeoff and landing capability.
- Combination transport ship - less cargo than a Freelancer but more data storage than a Herald.

AESTHETIC

- Similar shape/styling to the Genesis Starliner with stabilizing wings for enhanced atmospheric operation.

Length	40m
Width	38m
Height	11.6m
Mass	114,591kg
Speed	215 m/s
Max Crew	3
Powerplants	2 x Medium
Shield	2 x Medium
Cargo Capacity	96 SCU
Armour	Light

Weapons	2x S3 Hardpoints •2x Gimballed S2 Laser Repeaters
Turrets	2x Manned Turret •2x S2 Laser Repeaters
Missiles	2x S2 Missile Pylons •2x S1 Missiles per pylon
Countermeasures	1x S1 Joker Flares •32 xS1 Behring Flares 1x S1 Joker Chaff •32 x S1 Talon Chaff

The vehicle depicted herein is undergoing concept and design as of the release of this publication. Specifications and appearance are subject to revision during development.

KEY CONTRIBUTORS :
DESIGNERS: JOHN CREWE & KIRK TOME
CONCEPT ART: SARAH MCCULLOCH
ART DIRECTOR: PAUL JONES

In early 2014, the *Star Citizen* Ship Team began work on a new type of spacecraft, something not quite like others yet revealed: a tiny yet speedy rocket called the Drake Herald. Its role? Data Runner, a new career built specifically for the growing *Star Citizen* world. Information, designers reasoned, would be just as valuable in *Star Citizen*'s future as ore or manufactured goods. Whether that information was enemy fleet movements, priceless mining data, or the location of a newly discovered jump point, they saw the opportunity to let players deal in something more closely tied to the ongoing game experience than just simulated commodities.

The Herald was the first data runner, allowing players to physically move valuable data from place to place using a specially-shielded system of computer cores. In August 2018, the ship team revealed the Crusader Industries Mercury Star Runner, the next take on the data running career and proof that Drake's speedster was a herald of bigger ships to come.

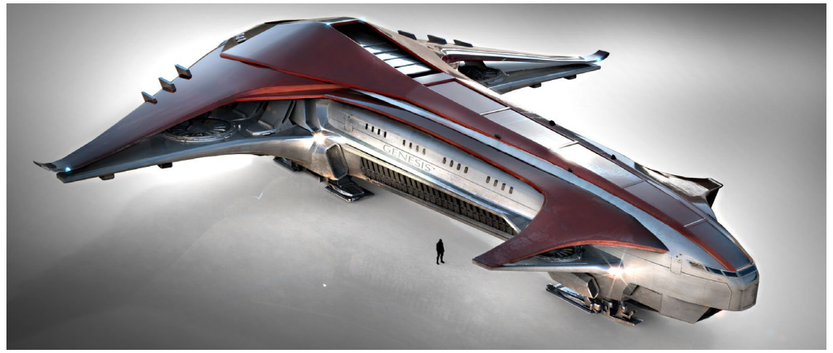
The Mercury, however, wasn't just imagined as a new data runner. The team had a second major goal that would make the development process much more difficult but much more rewarding - build a signature smuggling ship for the *Star Citizen* universe.

Ask any space opera buff what their fantasy role would consist of and they'll likely shoot you back a classic captain and their beloved personal spacecraft: Han Solo and the Millennium Falcon, Malcolm Reynolds and the Serenity, Grayson Burrows and his Tarsus. Indeed, that fantasy speaks to the very core of the *Star Citizen* dream: giving players the option to roll their own lovable rogue and chase their own fortune with their own ship. So, the Mercury needed to follow in the footsteps of those and other beloved tough-in-a-pinch private spacecraft.

CRUSADER GROWS UP

Crusader Industries was born as *Star Citizen*'s 'commercial' ship company. They build the ships that make the galaxy go around: freighters, cargo ships, liners, and landing craft. For several years, Crusader's only output was *Star Citizen*'s equivalent of the Boeing 737, the Genesis Starliner. Starting in 2017, the team began a concentrated effort to build out more of Crusader's history and catalog, resulting in both the Hercules Starlifter and the Mercury Star Runner. By following the latter close on the heels of the former, planners reasoned broader development work could be easily reused; lines, interiors, styling, and animations needed for the 'lifter' could likely be iterated on for the 'runner'.

Handling the look of Crusader's newest ship were lead designers John Crewe and Kirk Tome. The pair were tasked with the difficult challenge of initially pitching a ship with two broad goals: be the next 'level up' in terms of data running and be a signature design useable by smugglers.



To that end, they worked up the traditional set of specifications by marking up their thought processes behind each decision:

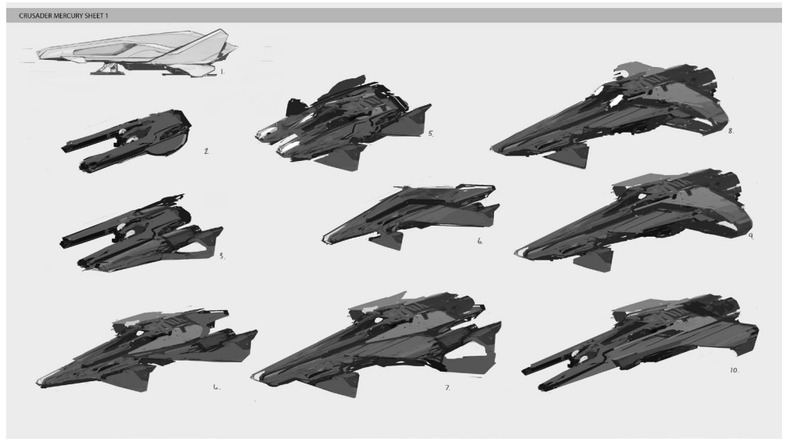
- Length: Long enough body for two main compartments, cargo and operations, with a shape/profile built for exceptional planetary takeoff/landing.
- Mass: Comparable weight to ships like the Cutlass and Freelancer.
- Crew: Pilot, Copilot, and Engineer for primary ship operations.
- Engines: Main engines in horizontal array, comparable to the Genesis Starliner. Outer engines slightly larger than center. Built for point-to-point speed.
- Thrusters: Fixed maneuvering thrusters along central body for on-axis thrust. Joint thrusters at the outer corners for maneuvering control. Four front-mounted retro thrusters.
- Power Plants: High-output industrial power plants to provide constant thrust and shield generation at the expense of creating a larger signature to track.
- Weapon Hardpoints: Limited overall weaponry but leveraging manned turret with combined weapon and pylons from a single operator seat.
- Shield: Twin medium shield generators, comparable defenses within the same size class but running industrial-leaning parts for higher raw shield health.
- Armor: Should be able to take a modest amount of incoming fire before really starting to worry about hull integrity.

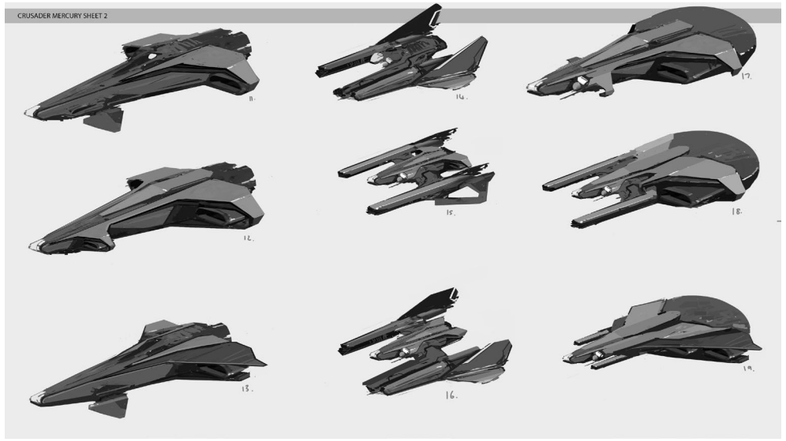
With these thoughts applied to the brief, the design team moved to pitch plans for the ship's interior and overall animation requirements. They even produced a rough wireframe of the design's general orientation. The initial design pass was sent on to the Ship Team at Foundry 42 where the challenge of building such an interesting ship would become clear.

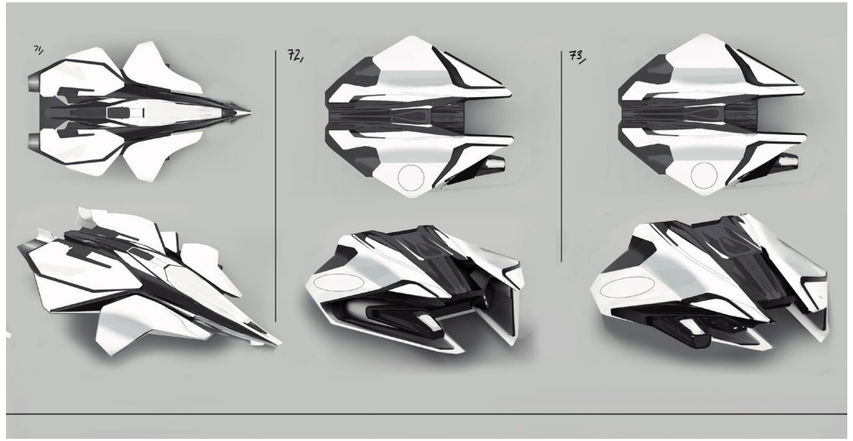
MERCURY TAKES WINGS

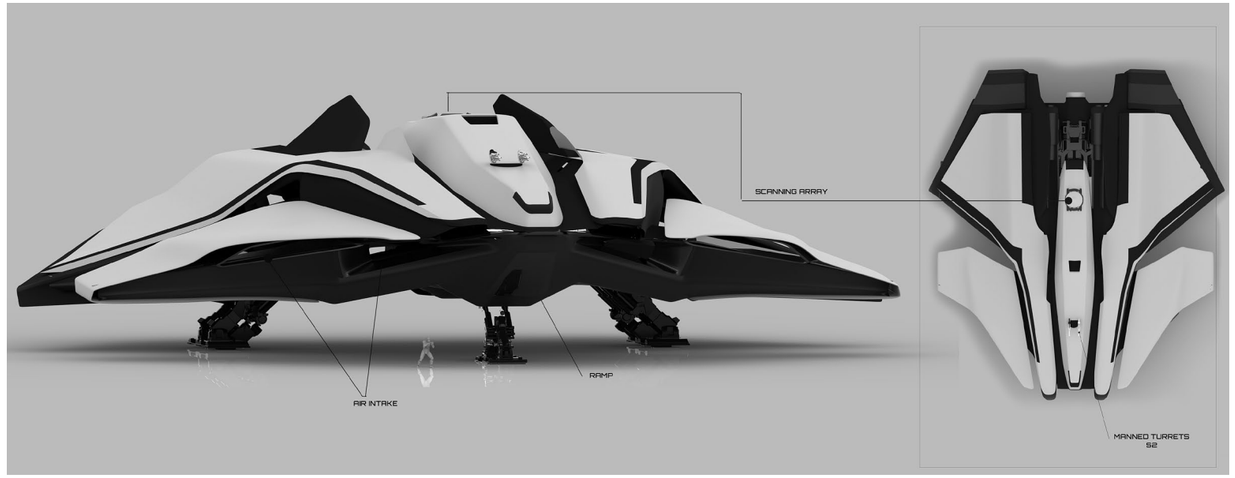
Tasked with another Crusader Industries design, Lead Artist Paul Jones knew just the person for the job: veteran Foundry 42 artist Sarah McCulloch. A fan-favorite artist, McCulloch's passion for vehicle design was well known to the community, though she had never designed her own *Star Citizen* ship. Jones prefers to start new ship artists on smaller single-seat vehicles before working them up to larger ships as their second or third designs. As McCulloch had already done extensive work on marketing pieces for the game and was very familiar with the game's vehicle requirements, he had no problem waiving this process in her case. What's more, she sits next to Michael Oberschneider who, at the time, was finishing Crusader's second design, the Hercules Starlifter. McCullough could call on him for assistance as she started out work developing another part of the Crusader canon.

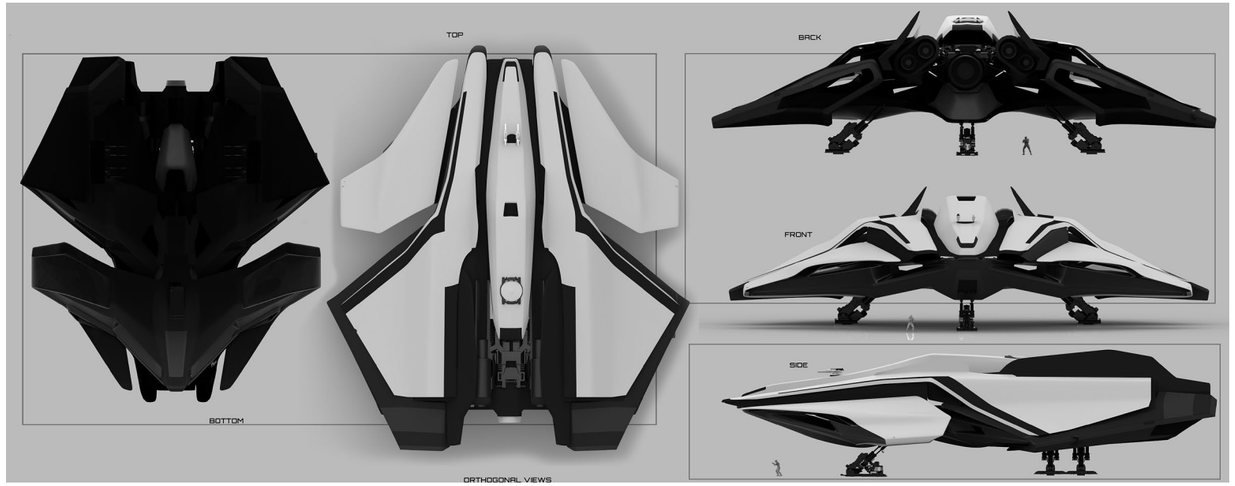
Jones began the process by considering the brief and the discussion with the Design Team. If this was to be *Star Citizen*'s equivalent of a movie's iconic smuggler ship, what exactly does that mean? Did the Mercury simply need to be a cargo ship or did it need to set itself aside in some way that makes it special? Putting together reference work, he collected stills from familiar films alongside a wide variety of civil and military aircraft (ranging from the SR-71 to the Dreamliner) and of course key artwork of the Starliner and the Hercules. How, he considered, could we turn the look of *Star Citizen*'s quiet, corporate spacecraft into something that might stand alongside the all-time great private spacecraft designs?







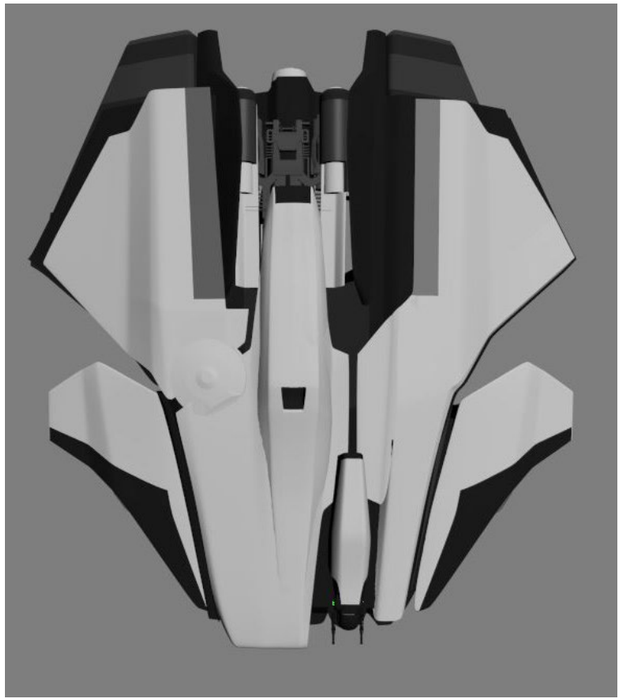


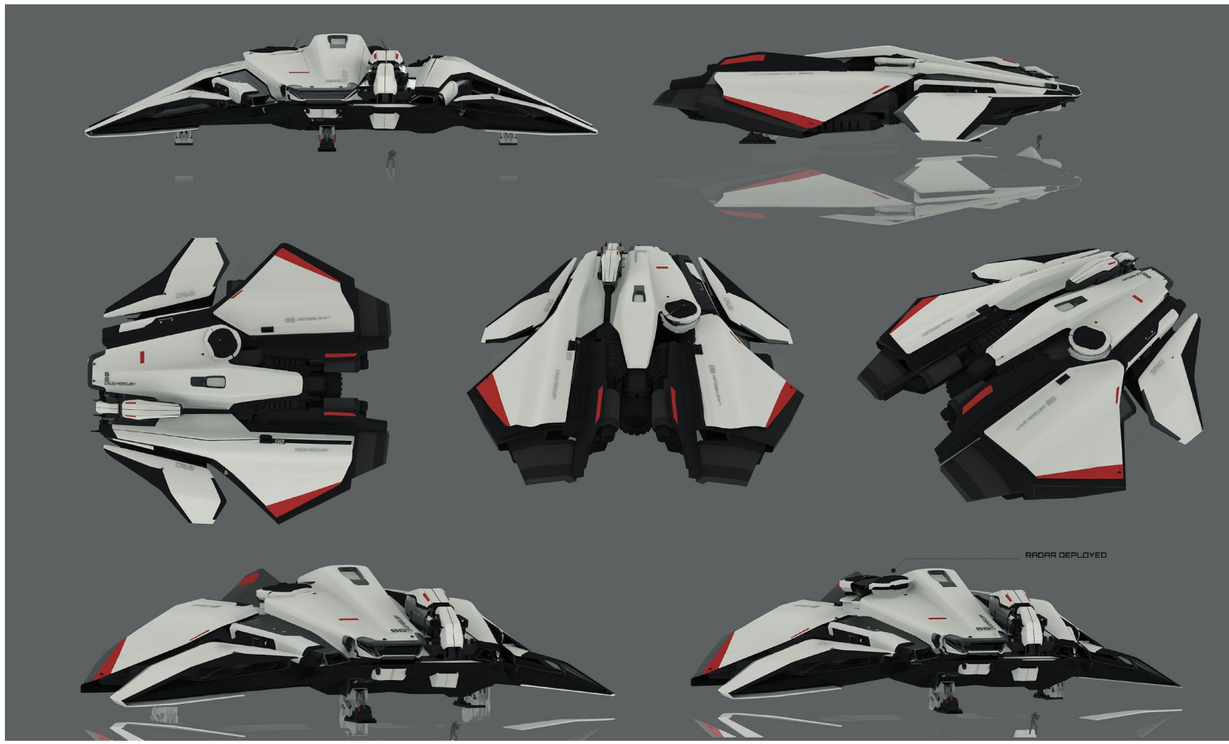


McCulloch began her task with an extensive range of explorations. With great latitude in the overall shape and size of the ship, she experimented with several dozen possible looks that varied in many aspects. Extra care was taken to break up the traditional Crusader look with options like a twin forked fuselage, asymmetry, negative space, large vents and large, rounded lower fuselages. Looking at all of the explorations, it's easy to see the transition from the initial Genesis 'space plane' look to the finished product.

Initial reviewers liked the asymmetrical design and the large rounded wings. For the first 3D round, McCulloch and Jones focused on developing a functional ship with a cockpit that enforced that idea. With the primary 3D model in hand, the team went through the process of distilling it into several options they were happy with. One problem right off was the discovery that asymmetrical ships were difficult to do justice to in static images. Getting the angle of the artwork just right became a special focus as it would be needed to make it through the first review.

Jones presented Chris Roberts with three options at this point: a four-winged version, an elongated version with an asymmetrical bow, and a version with only two large wings. Roberts liked the direction and wanted to further push the asymmetry. He asked to drop the fins and to make the ship sleeker overall. McCulloch experimented with several cockpits and positions to achieve this, trialing them all at the front and on the top of the ship. She also dropped the forward wings and experimented with more of an overall 'flying wing' shape. Jones was careful not to reference any familiar ships too closely, wanting to make sure everyone knew that *Star Citizen* could develop its own iconic ships. As the model approached the second pass review, the team was feeling that they were on the right track.

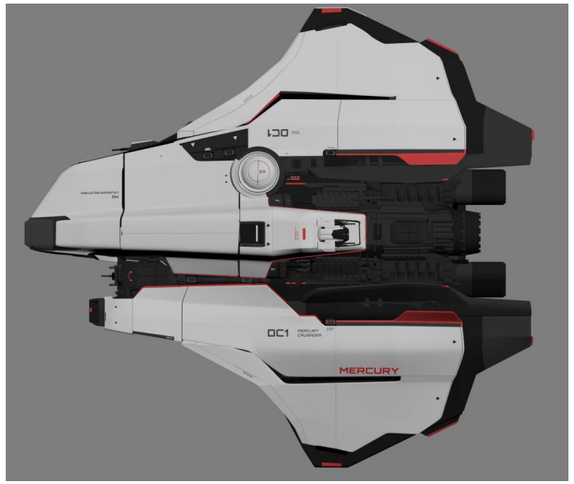


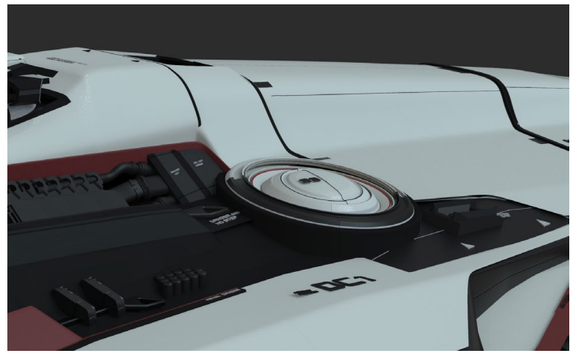


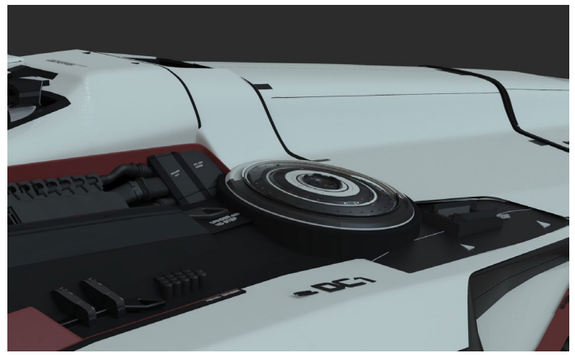
SECOND PASS

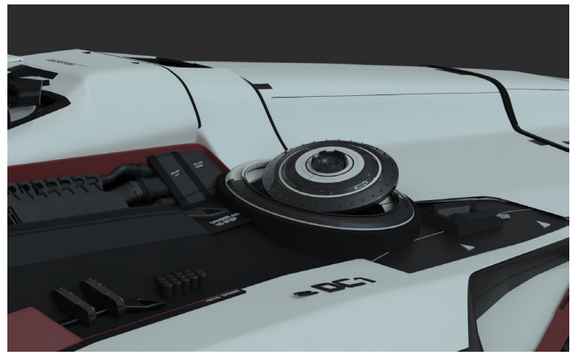
The second design pass was, on one level, a success: the overall direction was approved and the reaction to the work already done was very positive. Unfortunately, the second pass also found more work to be done, with Design providing new specifics about the data-running computer systems and a request to work in smuggler's compartments which could hide precious physical goods from opponent's scanners. Overall feedback from the first pass review was very specific, but the additional requirements would lead to more work on the ship's overall shape:

- Link the fins on the front to the back wings; it could almost be one big wing.
- Move the turret back onto the wing in the same position as the radar.
- Make the turret move 360 (look into 2 turrets).
- Look into doubling the cargo, work with Design on cutting out avionics racks. On par with a Herald but more cargo than a Cutlass.
- Make the walk away 1.25 width in the cargo area.
- Try and implement some sort of smuggler section for cargo, drugs, etc.
- Move the support station into the data room instead of the sleeping quarters.
- Work in a kitchenette for toilet, shower, eating, drinking and add lockers.
- Work up a skin for the default Crusader livery.
- Implement VTOL.









Jones and McCulloch were up against the clock at this point as the work creating a distinct look for the ship had taken additional time. Jones asked Michael Oberschneider to assist with model paint-overs to help deal with the challenge of making sure the already good asymmetrical design read correctly. The result showed exactly what he wanted. McCulloch rebuilt the model, tweaked the exterior, made some changes to the scanners, and reworked the interior. The second pass review went quickly and cleanly. Chris Roberts was extremely happy with the ship and, with the exception of requests for additional color schemes and a minor fix to the thruster setup, the Mercury was ready to run.

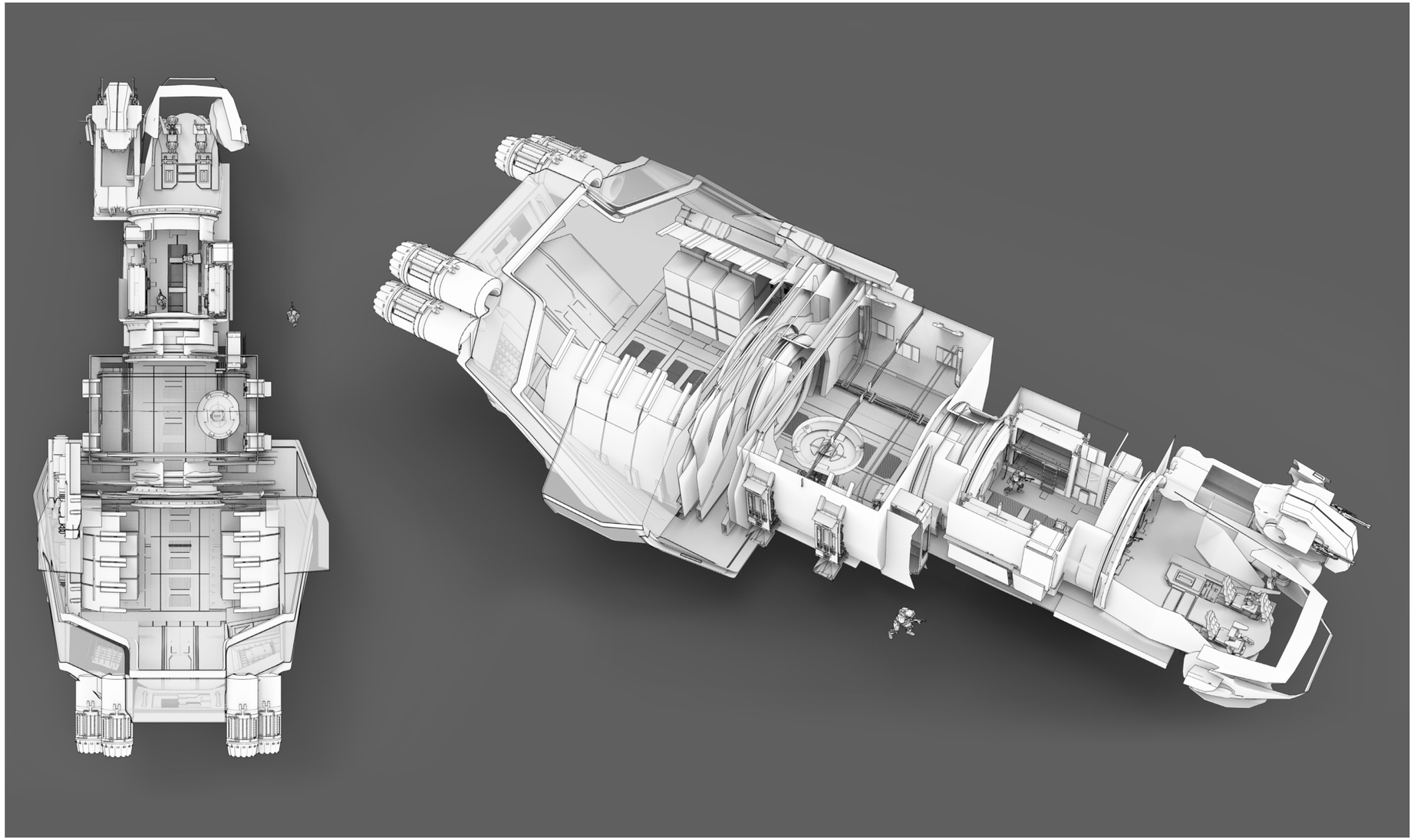
THERE'S INTEL INSIDE

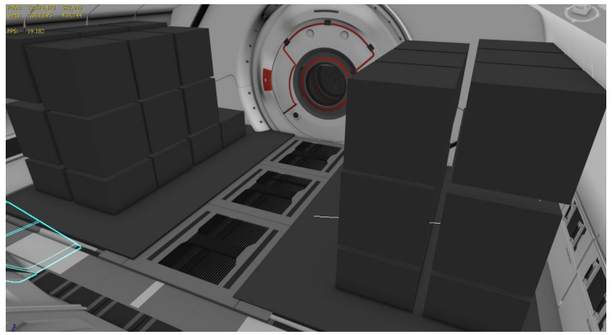
The initial design briefing for the Mercury included a list of interior spaces the designers wanted to be sure were worked into the ship's floor plan:

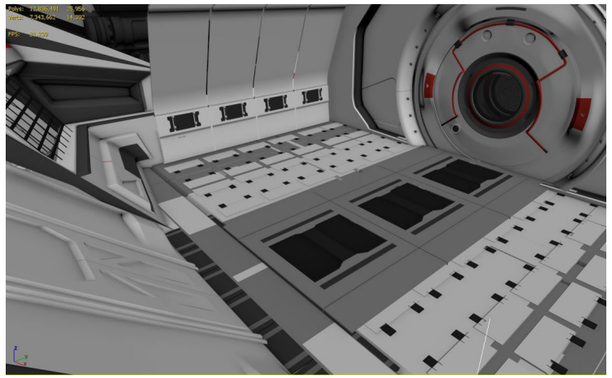
- Cockpit/operations room with basic living quarters for crew of three.
- One support station – datapod prep and management.
- Three beds – one bunk bed, one single-bed above support station.
- Toilet/shower combination.
- Weapon locker – capacity for four rifle-sized weapons and ammunition.

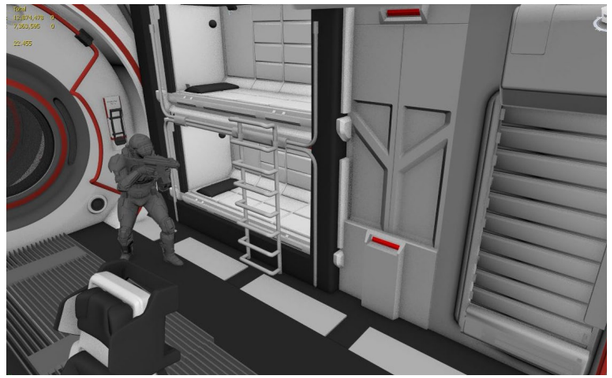
The designers also called out a series of interior features that would always need to be accessible to the player: one medium powerplant, two medium shield generators, five medium computers (one for operations and four for datapod storage), two medium coolers, one medium gravity generator, one medium jump drive, one quantum drive, one medium life support panel, and one medium radar. The interior would also need to accommodate two fuel tanks, though they would not need to be accessible in flight.

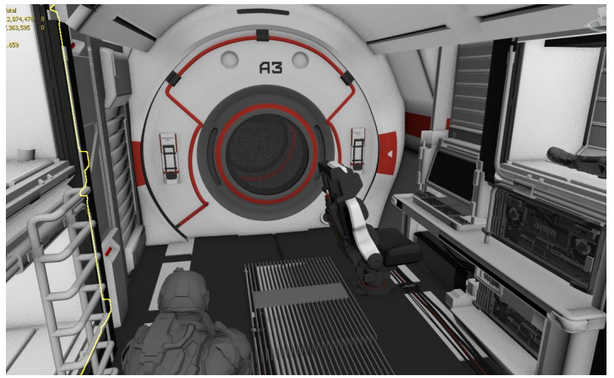
With this in mind, McCulloch began an initial workup of the interior alongside the first 3D pass of the ship. The interior was applied to the early concept, which placed the cockpit parallel to the rotating turret. This take reused the rounded interior styling that premiered on the Genesis and Hercules and had a greater focus on standard cargo rather than data. In this version, the solution to the data/freighter debate was simple: let it do both in some form. This plan lined up nicely with the original design proposal but unfortunately didn't match the added requirements following the second pass.

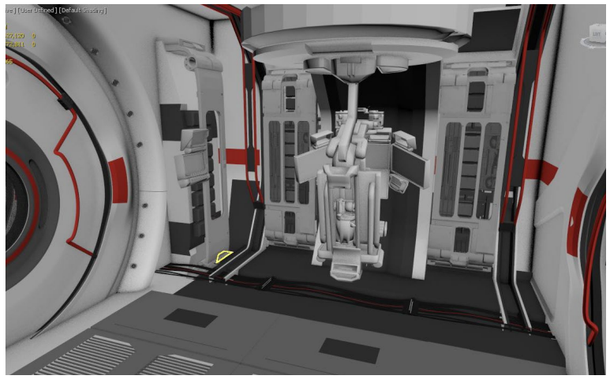


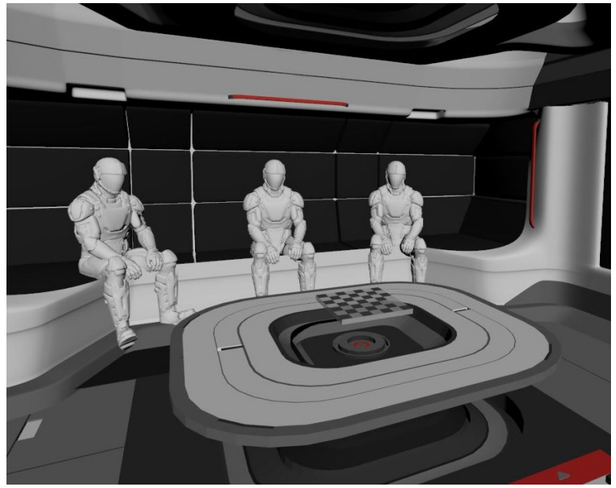


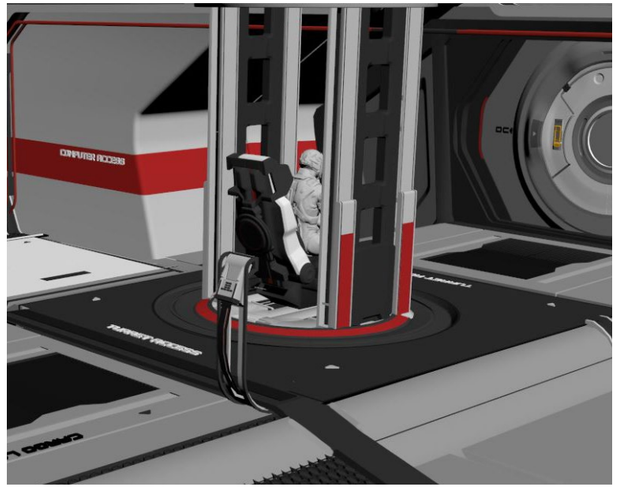


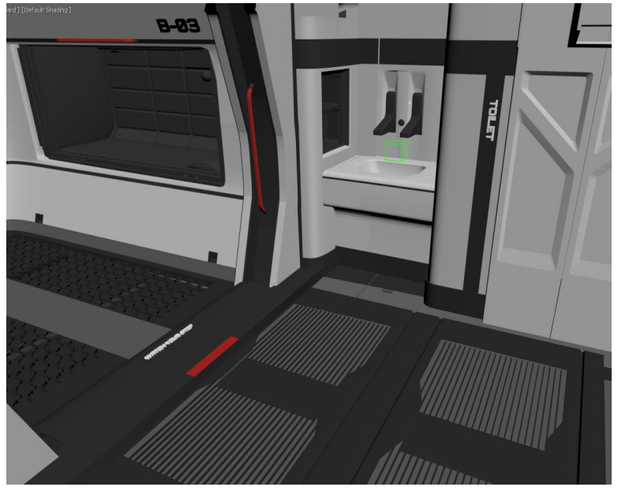




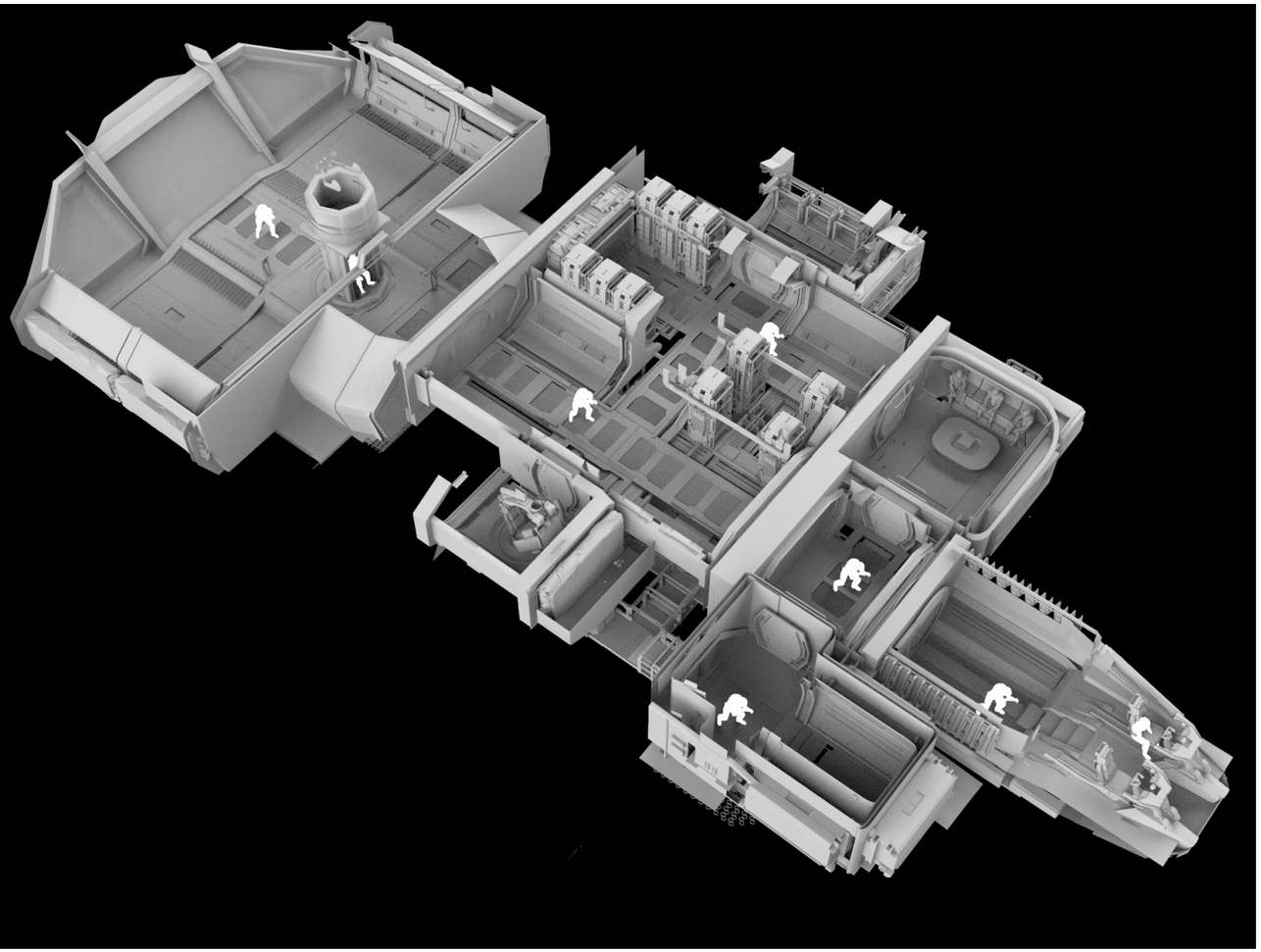








After the first pass review was completed, the difficult decision was made to scrap the existing interior work in favor of a fresh start to match the updated exterior. The artists would start again with a special focus on the newly requested features: more focus on the ship's computer cores and the addition of a series of secret compartments that could be used for smuggling. Paul Jones spent time referencing similar spacecraft in media and decided that the most popular designs had a sort of 'cat and mouse' feel to the interior. He worked out a rough plan with this philosophy in mind, organically placing smuggler's compartments and vents that would stretch from room to room. He debated moving the scanner and crew quarters, ultimately producing a layout with more of a focus on the data room. He also further developed the crew compartment, adding the holo-chess set and a table that sinks down to feel more secret. McCulloch worked from Jones' block-out to build out the reworked interior. The result was a success, with the review noting that it was now much more apparent that data running was the ship's primary mission.



MERCURY RISING

The final challenge was to present the Mercury to *Star Citizen*'s community. Art Director Paul Jones created a list of five shots for the marketing phase, plus the standard set of schematics that the team crafts for every finished ship. Jones tasked the Animation Team with preparing several characters for the marketing art phase, including armored soldiers, a jump-suited crewman, and a distinct hero character to stand in for the player. McCulloch's experience with marketing imagery again made her the perfect person for the job of creating the composite final shot. Jones sketched out the five proposed images:

LANDED (REAR VIEW): Ship being refueled or re-upped or something similar. There is a village nearby; water pumps, pipes leading to ship, steam venting. Cargo doors open and clearly showing someone coming down, another hero character hanging about. Locals in the background.

FLIGHT (REAR): Ship engaging in combat, turrets blasting away, maybe the turret is slightly angled. Ship is taking a few hits but nothing major to show how tough it is and up for a scrap. Nice amount of engine flare and trails. Structure on the side of the cliff or something to add more sci-fi-ness.

FLIGHT (DATA): Ship flying through deep space, scanner/transmitter up and catching the light and statuette of ship. Asteroids around, dusty asteroid field behind, sun flare off camera, long engine trails adding the dynamic-ness.

INTERIOR 01: Quantum travel but the shot is further back showing more of the cockpit with the hero character and jumpsuit dude in seats.

INTERIOR 02: Hero character with his back against wall looking over his shoulder, drink and game still in place on the table. Dude coming in and maybe another behind him checking the cargo, armored.

Meanwhile, the marketing and narrative teams worked together to create an interactive experience for players. Through a series of Spectrum and social media updates, backers were introduced to the *Belligerent Duck*, a Mercury with an interesting story behind it. Players followed the *Duck* 'live' with regular updates on its status that highlighted the capabilities and possibilities behind the Mercury design. Players were invited to 'Join the Manhunt' by tracking the errant ship using a special update to the ARK Starmap. Updates were shared across Spectrum as the hunt affected the ongoing story, giving a very early preview of the impact these new ships will someday have on the 'verse.

Banu Look Development

REFERENCES:

SHIP PAGE

https://robertsspaceindustries.com/pledge/ships/crusader-mercury-star-runner/Mercury-Star-Runner

SHIP PRESENTATION

https://robertsspaceindustries.com/comm-link/transmission/16725-Crusader-Mercury-Join-The-Manhunt

REVERSE THE VERSE

https://robertsspaceindustries.com/comm-link/transmission/16729-Reverse-The-Verse-LIVE

Q&A

https://robertsspaceindustries.com/comm-link/engineering/16753-Q-A-Crusader-Mercury

GALACTAPEDIA

TEARS OF FIRE

Tears of Fire is a painting by the early Messer Era landscape and military artist Aaron Fring. It was completed in 2610, after Fring witnessed Tevarin Warlord Corath'Thal and his fleet falling through the atmosphere of Jalan (Elysium IV, formerly Kaleeth) in an act of self-immolation.

Tears of Fire is Fring's only work to have been inducted into the UPE Historical Vault for Antiquity. It hangs in the Bentley National Gallery on Jalan.

BACKGROUND

The Tevarin lost their homeworld, then called Kaleeth, to the United Planets of Earth (UPE) at the end of the First Tevarin War (2541-2546). On 09 March 2603, a faction of Tevarin led by Warlord Corath'Thal sparked the Second Tevarin War; a protracted and brutal guerilla war for the Elysium system against the United Empire of Earth (UEE).

On 24 June 2610, after a catastrophic defeat in the Centauri System at the hands of *Squadron 42*, it became clear to Corath'Thal and his forces that they could not defeat the UEE military and an armed populace. Faced with the choice between surrender and death at the hands of his enemy, Corath'Thal chose a third option: he ordered his surviving loyal forces to charge through the UEE military blockade for Elysium.

SUBJECT

On 25 June 2610, the remains of the Tevarin fleet reached Jalan's atmosphere. The residents of the planet braced for an attack. Instead, Corath'Thal and his fleet dropped their thermal shields and dove for the planet. As they burned in the atmosphere, Corath'Thal's final words were picked up by human communication devices: "We die at home, and we die free".

ARTIST

Aaron Fring was born on Ferron on 13 October 2578 to teacher Max Fring and pilot Mary Fring. He learned to draw and paint as a child, during a period in which he was bedridden with Kilos' Malady. As an adult, his illness prevented him from following his mother into service with the UPE Navy, so he joined the UEE Expeditionary Force (UEEEF) as a Field Medic. The UEEEF eventually stationed him on Jalan. He fell in love with the landscape there and was inspired to resume painting.

The night of Corath'Thal's descent, Fring was taking an evening walk, having left all his communication devices at his home. As he approached the crest of a hill, he noticed strange lights in the sky, and saw the first of the Tevarin fleet push through the clouds. Fring watched until the last of the ships had either broken apart or crashed. He began work on Tears of Fire the next day.

The debut of Tears of Fire at the Gemma Gallery of Art launched Fring's career as a full-time painter. He went on to produce more pieces centered around the Tevarin Wars, to continual acclaim. His portrait of Corath'Thal, referenced from holos and oral descriptions, is noted for its use of smoke and lighting. He died on 11 November 2631 from Kilos-related complications, aged 53 years. Tears of Fire was inducted into the UEE Historical Vault for Antiquity in 2633.

Fring cited human Romantic-era Earth (Sol III) art as one of his primary influences, particularly the use of light and color in Ivan Aivazovsky's 1848 painting Battle of Chesma. A copy of the painting hung in Fring's studio. Tears of Fire inspired a revival of Romantic-style art in the UEE, notably pieces with glowing, effusive sources of light.

WHERE IN THE 'VERSE?

**Every month, we post a close-up image of something in the universe.
All you need to do is tell us where you think it was taken.**

JumpPoint@Cloudimperiumgames.com

**We'll reveal the answer next month, and share some of the best responses we received.
This month's image is courtesy of Ray Warner, our Assistant QA Manager in the UK.
Where in the 'Verse did he find it?**

Ray also gave us last month's image. But Where in the 'Verse did he find it?

Our winner this month is PELCKI, who correctly identified the image after a quick search:

BEGIN TRANSMISSION →

"I've been looking all over the Stanton system and did not seem to find the lamp which I was looking for. So I decided to take a trip to Levski. And since I landed next to the outside doors I noticed a familliar lamp on the biggest tower."

END TRANSMISSION ←

Congrats, PELCKI!
You get this month's coveted Jump Point no-prize.

Please remember to send us a screenshot of what you find, so that I can give partial credit if what you've found is close to the actual image.

ONE QUESTION

We asked the CIG staff to answer one question for us this month. Here's what they had to say.

DO YOU HAVE ANY HOBBIES THAT NO ONE WOULD SUSPECT?

ANDREW WATSON, TOOLS PROGRAMMER

I like to sew plushies.

SHANE JOHNSON, CONCIERGE VIP SPECIALIST

Well, I'm a competitive shooter. Other than that, I paint miniatures with my little brother.

IDREECE HADI, JUNIOR DESIGNER

I enjoy to dance, I regularly go to jazz/ballet, street dance, jazz funk classes and urban choreography.

CHRIS GALASSO, OFFICE ASSISTANT

I practice Brazilian Jiu Jitsu, which is super fun and I don't think people would suspect of me. Francesca got me involved about six months ago, but I've been really enjoying it and it's taken over a lot of my life already. I even earned a black eye while training about a month ago, so that was a good time, haha.

STEVEN KAM, JUNIOR COUNSEL

I cook, but I'm bad at it. That's why I don't tell anyone. Um. Oops...

MATT GANT, QUALITY ASSURANCE FPS SPECIALIST

I Cross Stitch! Quite an unusually hobby for someone under 60 years old I guess.

MARK HONG, PRODUCER

Geocaching, including Wherigos.

JEFFREY PEASE, DEVOPS ENGINEER

I own a sailboat and go sailing when the wind doesn't suck.

CHRISTIAN SCHMITT, PLAYER RELATIONS DEPARTMENT

I've been into live action roleplaying games (LARP) for 19 years now.

CHERIE HEIBERG, ARCHIVIST

I binge-read advice columns. My favorites are Captain Awkward, Dear Prudie, and Ask a Manager.

TOLU WINJOBI, JUNIOR FINANCE ASSISTANT

I enjoy sewing, I'm not great at it but I find it quite therapeutic.

JOSH COONS, 3D MODELER

I play a bunch of airsoft.

STEVE BENDER, ANIMATION DIRECTOR

I play bass and brew beer.

KYLE CUNNINGHAM, IT HELP DESK ASSISTANT

I'll say Hikaru Dorodango.

BEN CURTIS, PROPS ART DIRECTOR

I watch a lot of BMX and skateboarding on YouTube... I mean I used to ride to work on my BMX, but then got an office parking space and got lazy. I'm not sure it's a hobby exactly, but I spend most of my free time doing DIY. I've got three bathrooms to replace next year! Oh and I don't have one at the mo, but I'm planning on restoring a classic car soon. I did an evening course in car restoration, but that was before I made director and got busy!

PETER ROYLE, LEAD SHIP ARTIST

DIY I guess, I like to build stuff and have got a good selection of power tools, but between work and my kids I have very little time left for hobbies!

CORY BAMFORD, SENIOR PROPS ARTIST

I like gardening and growing vegetables. I guess people wouldn't suspect that looking at my pasty nerd-complexion.

MARTIN DRIVER, COPY EDITOR

I like fixing up old French cars. If anyone has a steering rack for a Peugeot 605 SV 24, get in touch!

GRAHAM ROBINSON, JUNIOR GRAPHIC DESIGNER

Making random brush pen strokes (with a real pen, not Photoshop) and then trying to turn the abstract mess into something... like a horse... or something.

TOBIAS JOHANSSON, SENIOR LEVEL DESIGNER

I love baking bread. I usually do it at least once a week!

ULF KURSCHNER, SENIOR COMMUNITY MANAGER

Geocaching. I go hunting for little containers other people have hidden in all kinds of places using a GPS tracker. My best find so far was on top of a 2700m mountain (that's about 8858 feet). Other than that? Beer brewing. Even my wife likes the days when the kitchen is wiped clinically and sterilized to not ruin the batch.

PAUL REINDELL, DIRECTOR OF ENGINEERING

I like to tinker with little electronics and SBCs. I also play the piano and like to make music for myself.

JUSTIN BINFORD, QA DIRECTOR

I like the occasional poker game and can play the violin.

PAUL JONES, ART DIRECTOR

I like gardening and riding my bike.

HUMBERTO ASPERO EYRE, ASSOCIATE PRODUCER

I have a teddy bear Instagram account, which I use to tell stories of the bear... which are actually dumb things that happened to me in real life.

ANDREW DAVIES AKA JOANNA CUDDLE, SENIOR GAMEPLAY PROGRAMMER

I'm a drag queen in my spare time, does that count?

GLENN KNEALE, LEAD TECHNICAL QA TESTER

Brazilian Jiu Jitsu.

TRACY GEESE, ECOMMERCE MANAGER

Hmm, well I really love downhill skiing. I don't have much of an opportunity to do it since transplanting to Texas but whenever I go back to Michigan in the winter I try to go as much as I can!

Do you have one question you want to ask the staff?

Send it to JumpPoint@Cloudimperiumgames.com and we might choose your question for next issue.