

## Submission 3: Vulkan Space Program

VU Computer Graphics 2025W - Group 6 - 18.01.2026

Ott Marvin, Werdermann Wilhelm

### 1. Description

Our demo aims to showcase a Vulkan real-time rendering framework for a modular space environment populated by various procedural generated astronomical objects on fixed orbital paths. Simulated gravity allows for a deterministic path of a spacecraft through an example solar system using predefined maneuvers and smooth camera and time transitions. Around a star we showcase several traversing planets with distinct surfaces, biomes and atmosphere, fully customizable using an optional ui inspector. As well as gas giants, where simulated fluid is used to animate mixing oppositely accelerated color bands, perturbed by cyclones, where the resulting colors are sampled for a textured atmosphere.

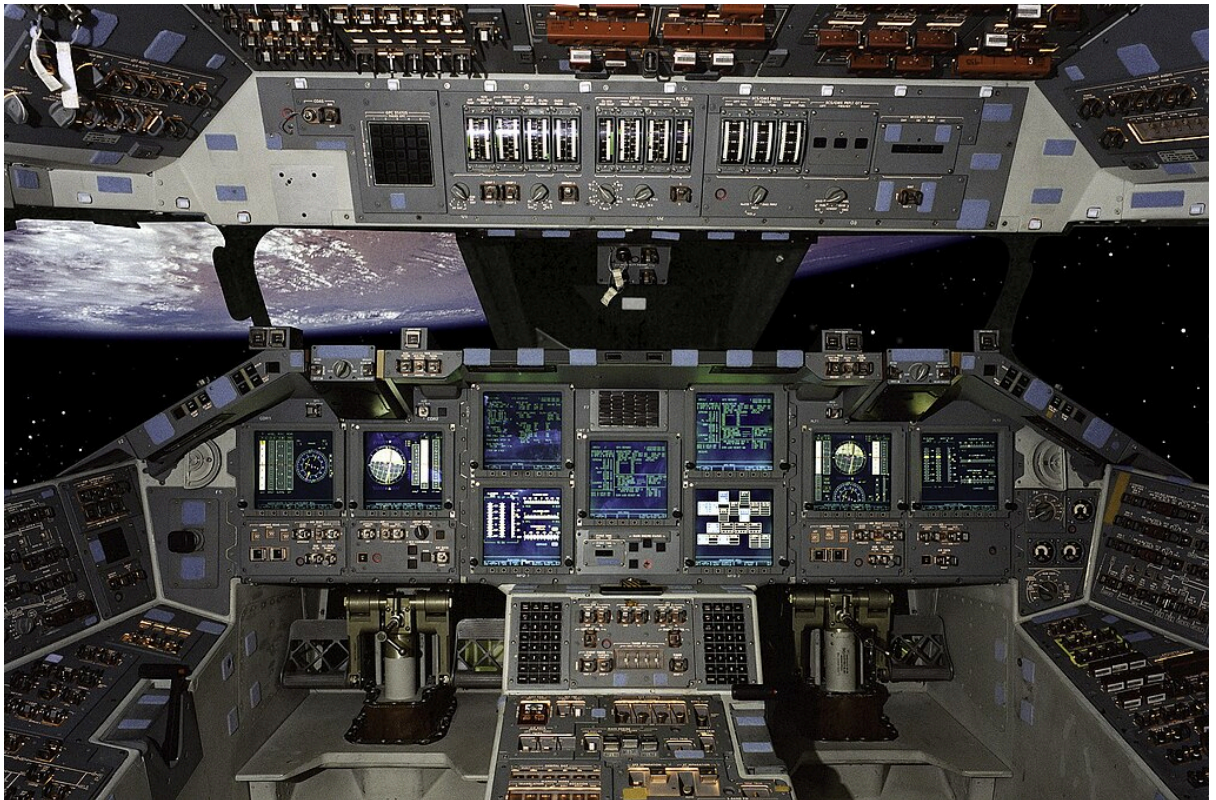
### 2. Technology

#### 2.1 Libraries used

- **Assimp** model loader: <https://github.com/assimp/assimp>
- **Lygia** Procedural Noise: <https://lygia.xyz/>
- **GLI** dds image loading: <https://github.com/g-truc/gli>

#### 2.2 GPUs tested

- 3060, 3090 Ti, 5090, 2060 Mobile



How using our demo feels like... ;)

### 3. Controls

Press **F5** to restart the demo.

You can switch the camera modes for interactability, disabling the demo mode.

- **F1 - TARGET\_FOLLOW**: Follows a selected target as a fps-style camera. Rotation using right click. Movement with WASD, CTRL and SPACE. SHIFT is a speedboost. Scrollwheel changes move speed.
- **F2 - TARGET\_FOCUS**: Works as an orbit camera around the target object. Rotation using right click. Scrollwheel the distance can be changed.
- **F3 - TARGET\_VELOCITY**: Cycles an orbit camera for the target object in the direction of the target's **prograde** / **radial** / **normal** velocity relative to its dominant object. Scrollwheel the distance can be changed.

“**U**” toggles the UI visibility, including scene inspector, allowing the modification of most objects and component values, given a target object of our scene. Also a camera inspector allows manually changing the camera's target reference and mode, as well as the speed parameters.

“**T**” toggles the wireframe mode for tessellation inspection. Note that per default probably nothing changes, as the surface is very detailed. Either zoom in a lot or reduce the tessellation factor under “Astro\_Object\_Settings -> Surface -> Tessellation -> Factor”. Note that you have to select the correct solid planet you want to change as the target object.

“**O**” toggles the visibility of all orbits and the spacecraft's gravity prediction.

When the demo completes, we enable a game-ish experience, where the user can burn with the thrusters using “**M**” and “**N**” into the spacecraft's forward / backwards direction.

The prediction rendering will automatically update, where “,” / “.” control the time factor.

In addition, the orientation can be modified using the keys “**H**”, “**J**”, “**K**” for the **prograde** / **normal** / **radial**. Fly safe!

#### 4. Scene

**Implementation:** Our scene handles different objects with their individual behaviors using a simple generalized Entity Component System (**ECS**) including a transform hierarchy. We provide a framework to instantiate instances of SceneObject in the scene, where each manages its SceneComponents and propagates calls from the scene (i.e Start, Init, Update, Render, etc.) to each individual behavior. This way we can configure all astronomical objects with optional behaviors, such as its surface or atmosphere implementation. We also utilize abstractions of AstroObject to enable an AstroFactory to easily populate our scene using create structs and predefined templates

**Goal:** We want to leverage our fully customizable components to produce a variety of visually interesting scene objects for our astronomical systems. Including different implementations and configurations of multiple surfaces, biomes, atmospheres.

We use a simplified orbital model to enable a ballet of cratered moons, diverse planets and menacing simulated Gas Giants following a fixed path around a Star. We will explore the system following a high detail spacecraft model through an artistic interpretation of Sol, including its major planets and moons. The spacecraft's movement is simulated using gravity and accelerates with its thrusters following a predefined but customizable maneuver node plan or the user input. The camera movement is interpolated and multiple camera modes exist for exploring the scene relative to all objects.

## 5. Objects

**AstroObject:** Interface to **SceneObject** for all astronomical objects, including Orbit, Atmosphere, Surface components and other shared information, buffers, shaders. Star populates a StarSurface, PlanetGas a PlanetSurfaceGas and the other different Planets and Moons (i.e PlanetTerrestrial, PlanetIcy, ...) populate a PlanetSurfaceSolid. We also feature an imported procedural skybox and use our star as a pointlight source.

We precompute each astro object position in the system up to some point, to speed up the gravity simulation and its prediction. This is done in an extra background thread which idles when done. To make it deterministic, we use fixed timesteps.

## 6. Components

**6.1 Surface:** manages Mesh and Object Rendering

**6.1.1 StarSurface:** shared point light source for an astronomical system

**6.1.2 PlanetSurface**

- **Gas:** Fluid Simulation of surface producing iconic bands and cyclones/storms. The fluid is a 2d eulerian simulation, which is wrapped around a cylinder which is then projected on the planet. The bands are simply slowly injecting color and blending a wanted velocity into the fluid. The storms create a rotation and also inject color into the simulation.
- **Solid:** Compute shader for generating a cubemap as heightmap for a sphere surface using weighted layered noise. We sample the cubemap to displace the surface of a tessellated icosahedron mesh and colorize by mapping height and slope to biomes using separate materials. The tessellation scales with the objects radius and distance to camera)

**6.2 Atmospheres:** Volumetric Blendable Atmospheric Scattering

The atmosphere works as a post processing effect, to allow proper blending between the different atmospheres. Each atmosphere is rendered from back to front as a sphere containing each atmosphere. During each pass, the current colors are read and used as an input for the next atmosphere calculations. This blending is done manually in the shader. Each astronomical object can cast a shadow onto all other astronomical objects, which can create for example a solar eclipse. There are two atmosphere types. A generic model for solid planets and one which uses a texture for the coefficients. The latter is used for the gas giants with the result of the fluid simulation.

**6.3 Orbit Animation:**

Our model allows for elliptical and hyperbole orbits relative to a target body, given non extreme parameters. With specified orbital parameters we provided functions to compute orbital state at any point in time. This allows the astronomical object to be positioned and animated along its orbit based on variable simulation time.



## 6.4 Spacecraft

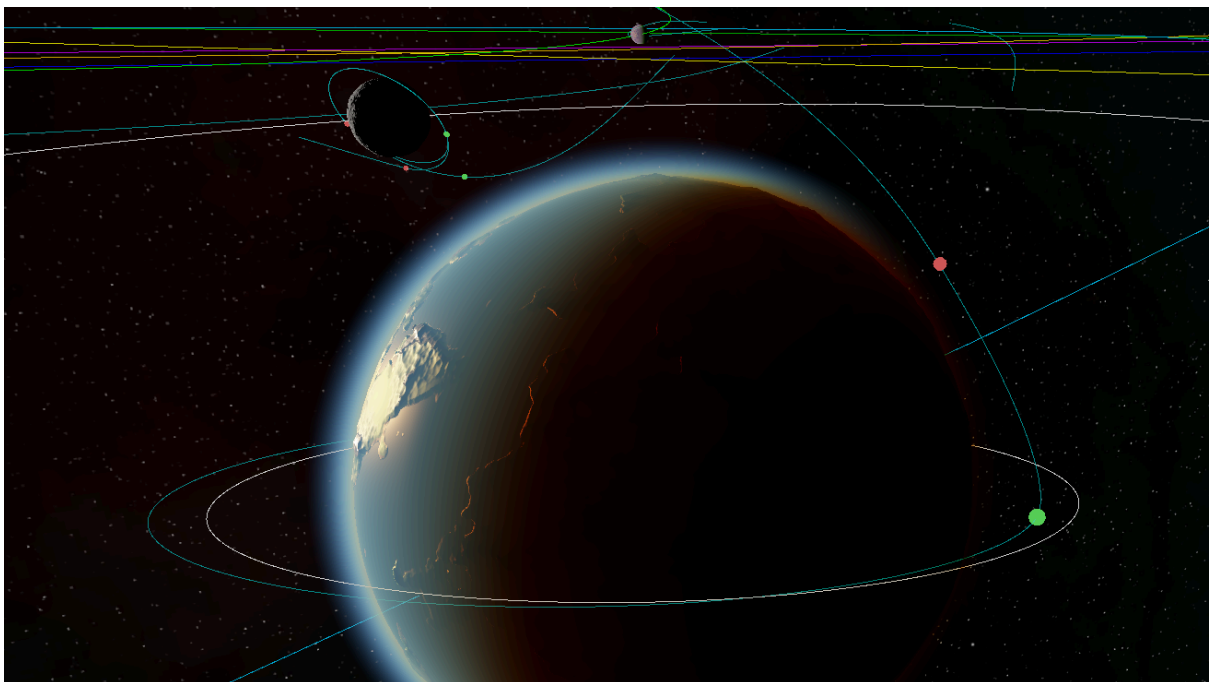
The spacecraft model uses normal maps, metallic, and roughness textures as inputs to the physically based shading. It can also have multiple light sources, which are currently used for thrusters.

### 6.4.1 Prediction:

The Spacecraft prediction component predicts the path the spacecraft takes throughout the system in the future. It is fully deterministic by using the same fixed timesteps as with the astro object position precomputation. This deterministic property allows us to create a flight plan using maneuvers. This prediction runs in its own thread and is reset when something changes like a manual thruster input.

### 6.4.2 Navigation:

The spacecraft navigation component is responsible for applying thrust for the specified maneuvers, orienting itself and simulating gravity for the current timesteps. The gravity simulation works the same as with the prediction to ensure it does not deviate from it.



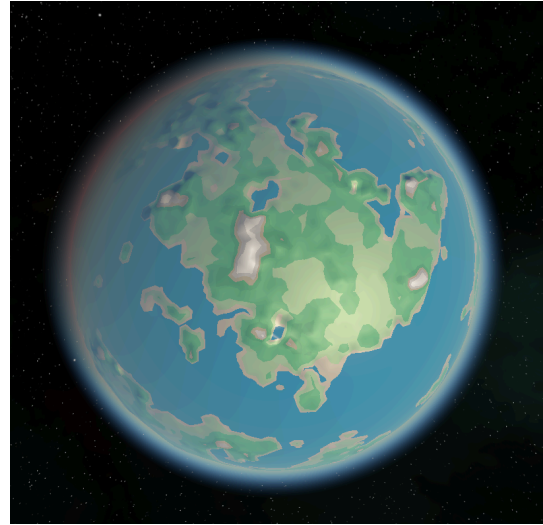
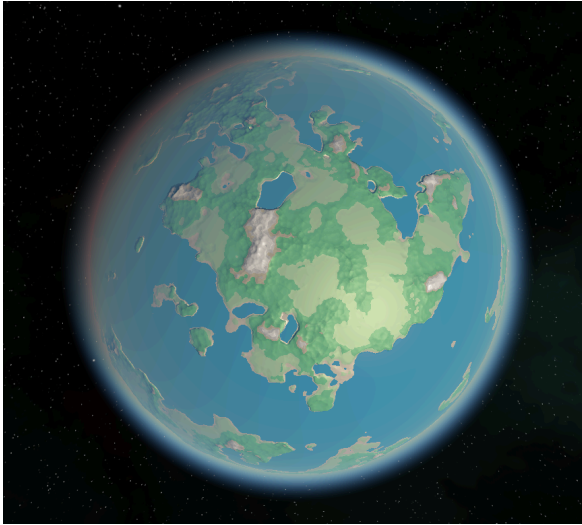
### 6.4.3 Particle System:

The particle system component can be used to simulate particles on the GPU. They are fully managed on the GPU, allowing each particle to have its own lifetime and other attributes. Particles exceeding their lifetime are removed and their memory is freed to allow for new ones. This is accomplished by using atomic operations. Particles are also simulated for gravity around a specified target. The particle system is used by the spacecraft thrusters (RCS and main thrusters). RCS thrusters fire when the spacecraft is reorienting itself and the main thrusters during a maneuver.

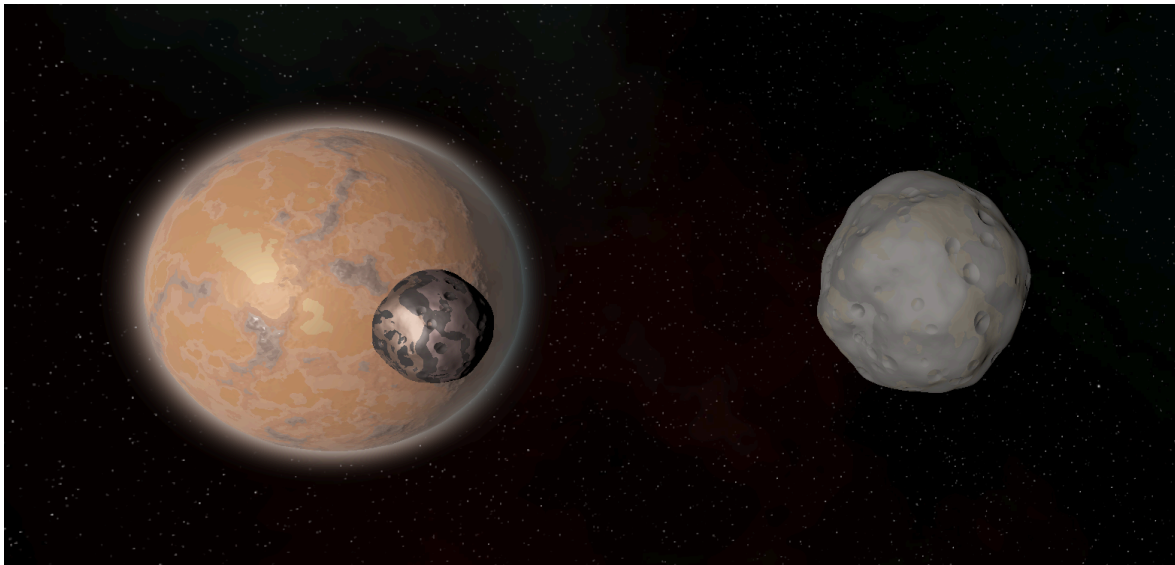
## 7. Effects

Wilhelm Werdermann:

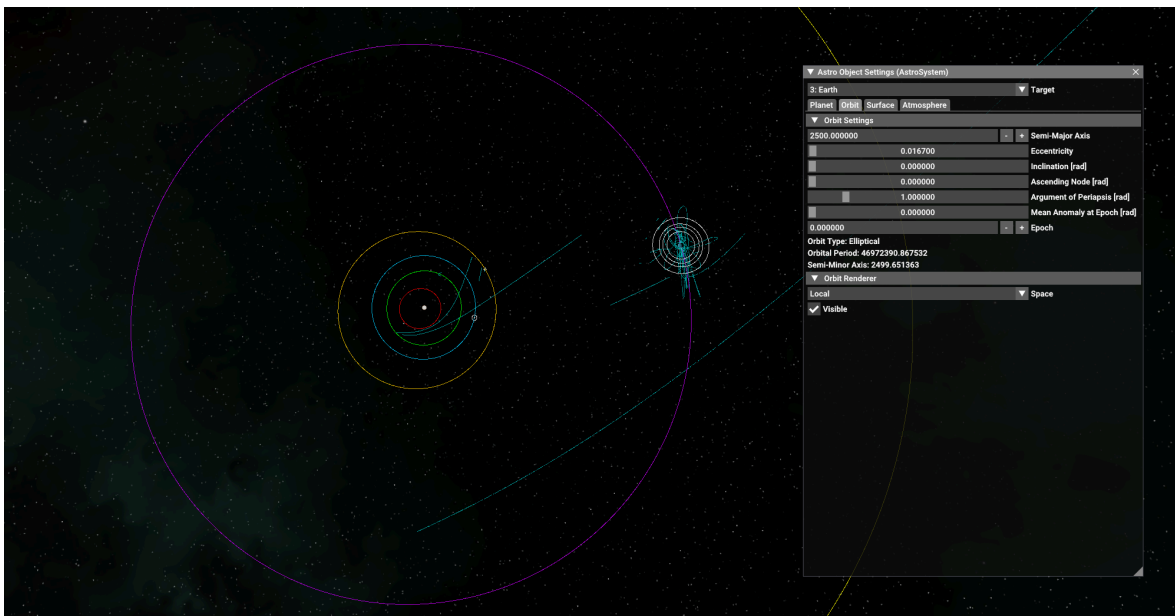
- Adaptive Tessellation for solid terrain (complex effect)



- Procedural Geometry (simple effect)

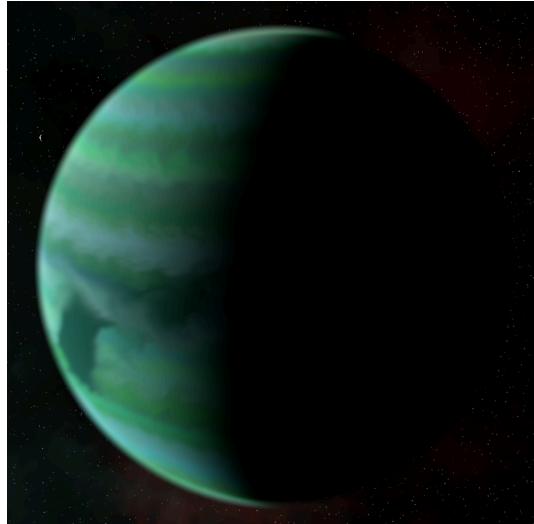
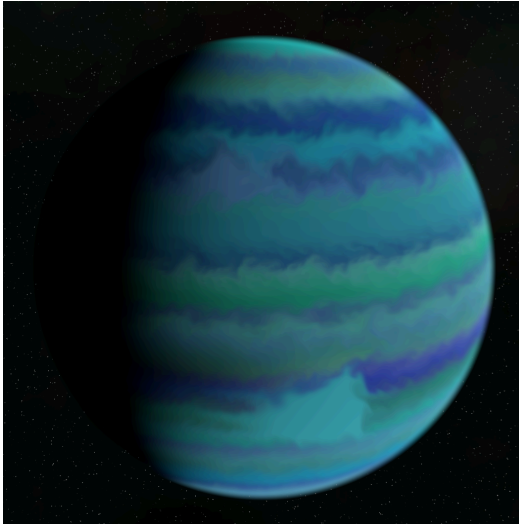


- Hierarchical Animation (simple effect) (orbits)

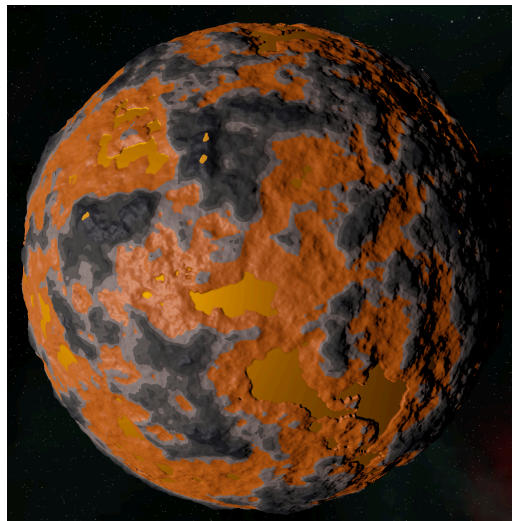
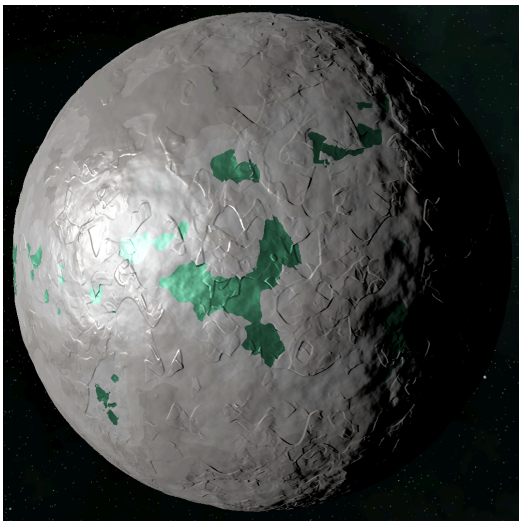


Marvin Ott:

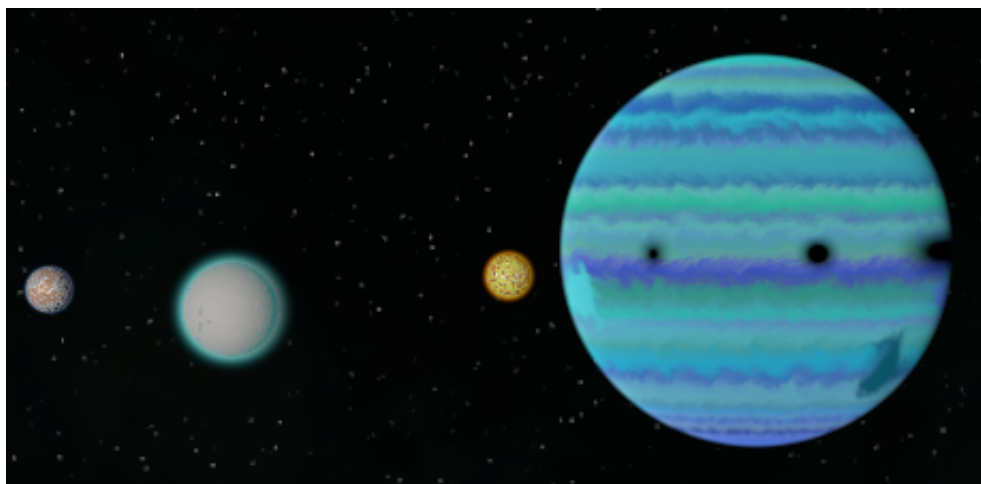
- Eulerian Fluid Simulation for gas giants on GPU (complex effect)



- Physically Based Shading (simple effect) (already done previously so no points)



- Planet shadows (simple effect)

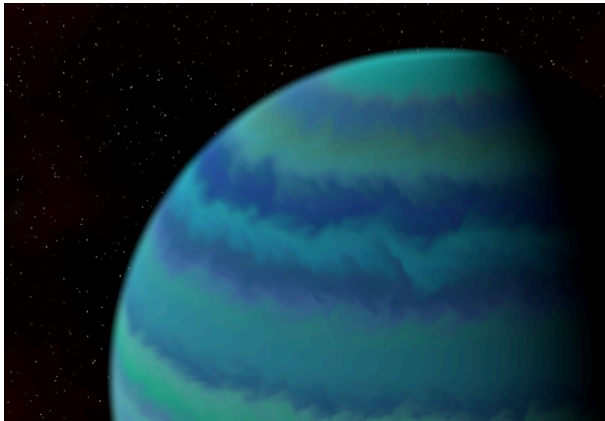


- GPU Particle System using Compute Shader (complex effect)

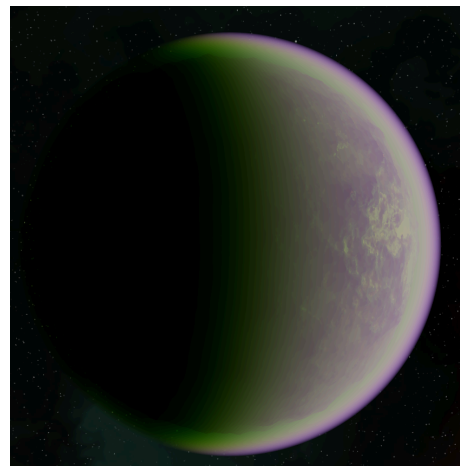
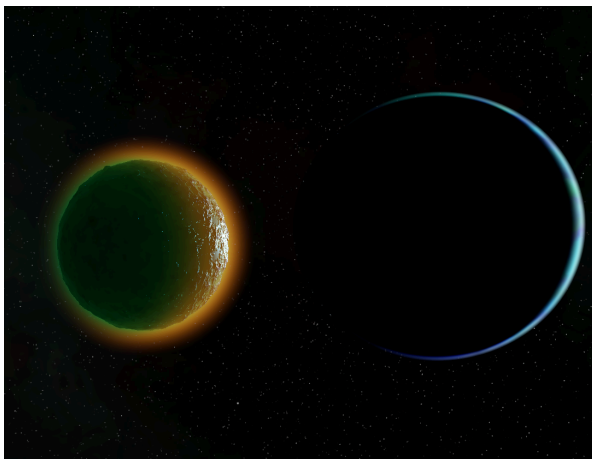


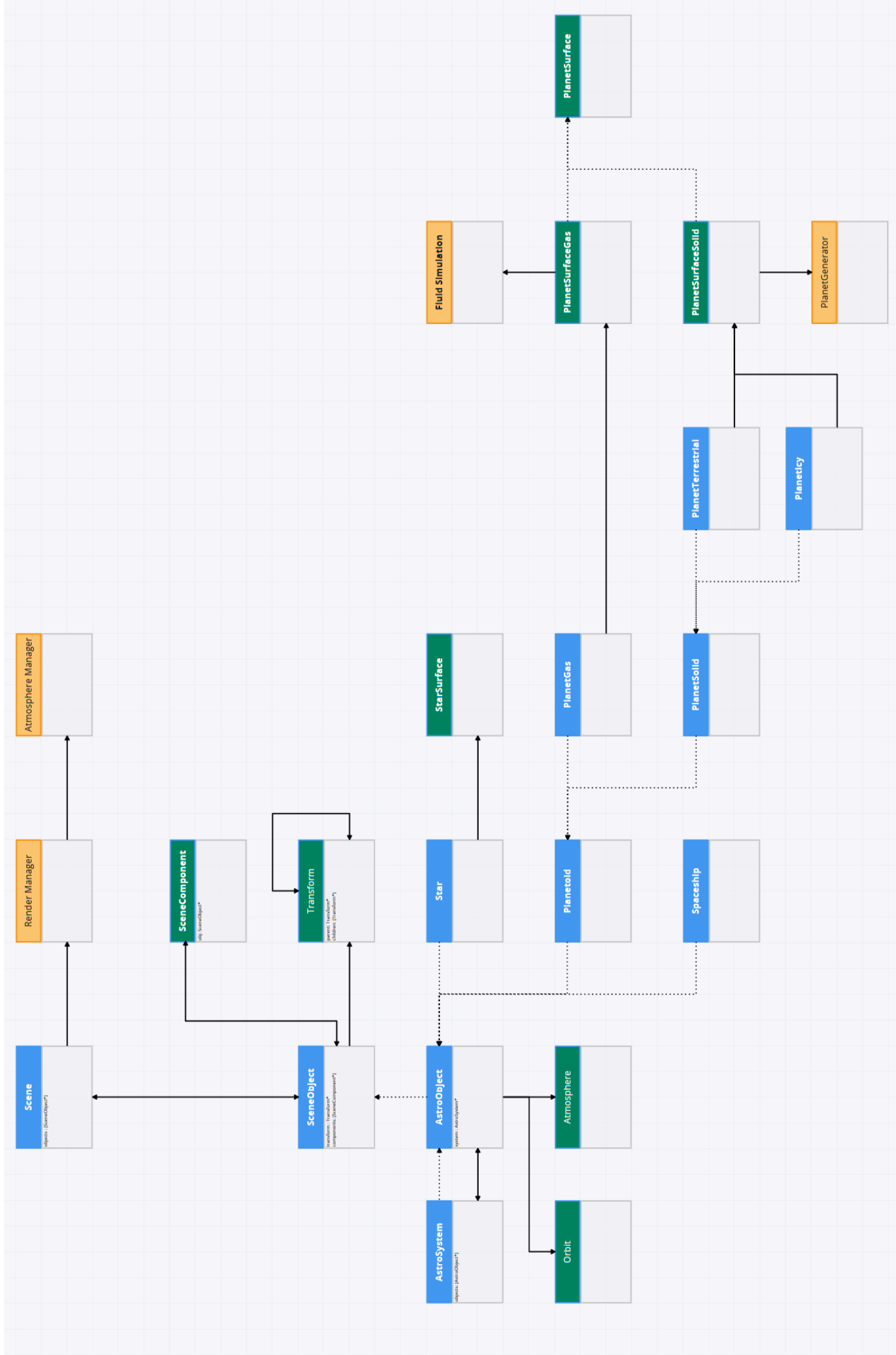
Both:

- Procedural Textures (heightmap=>biomes, fluid texture) (simple effect)



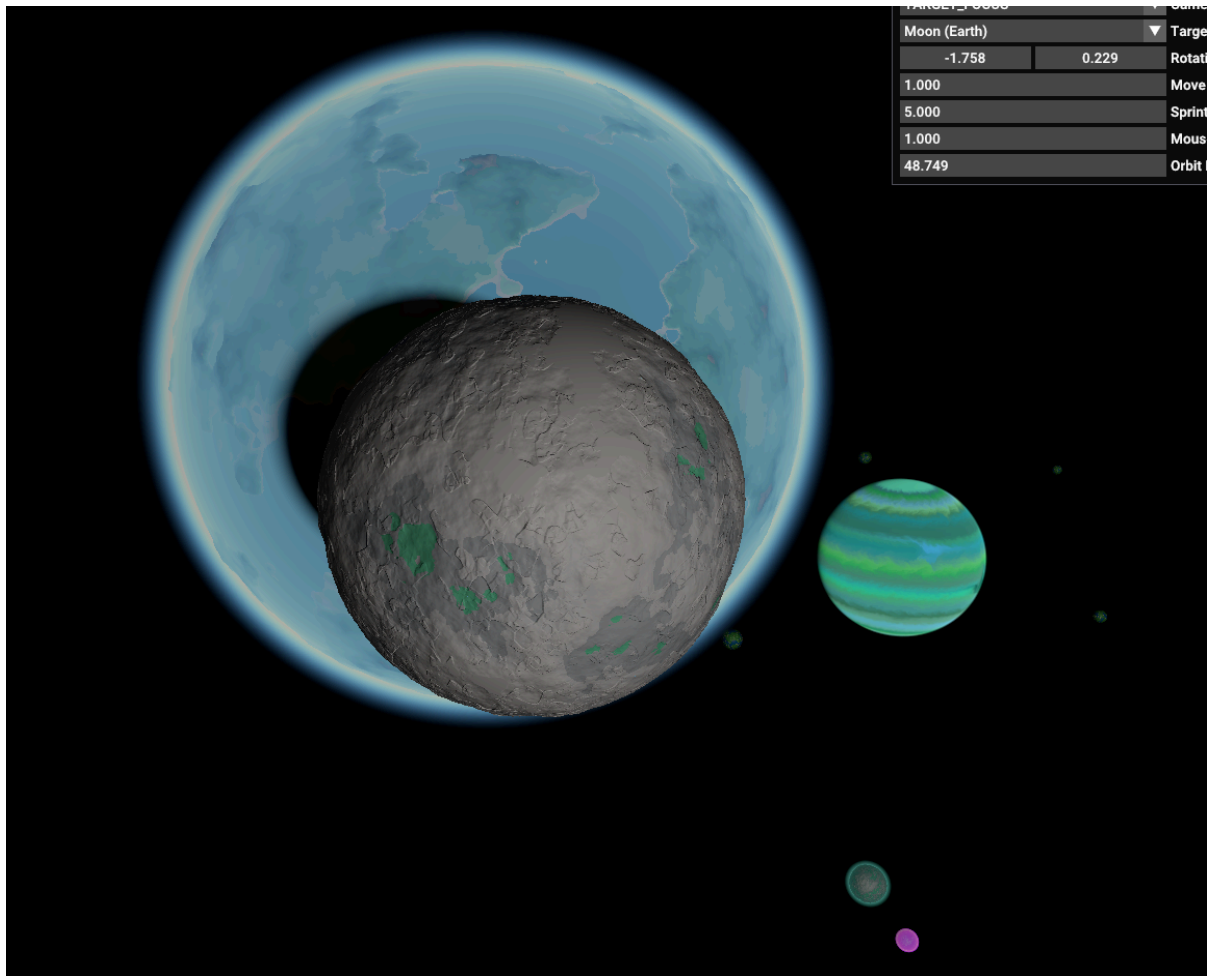
- Volumetric Lighting for atmosphere (complex effect)



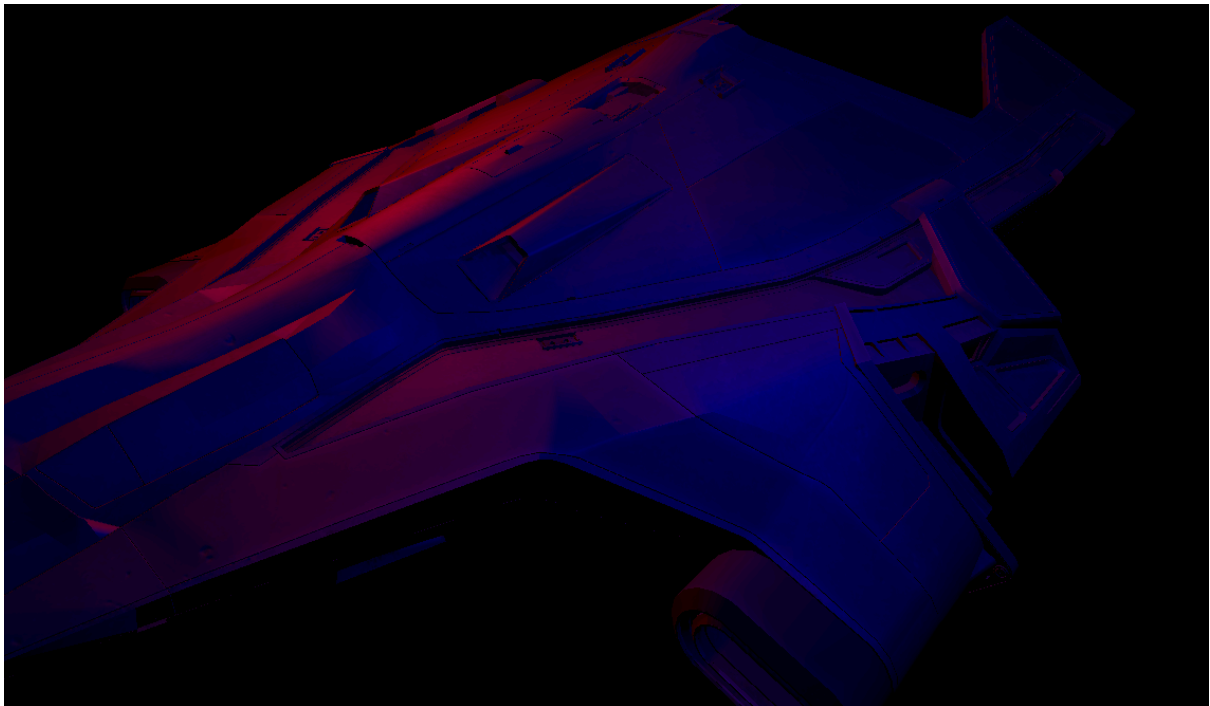




### Impressions (including older versions):

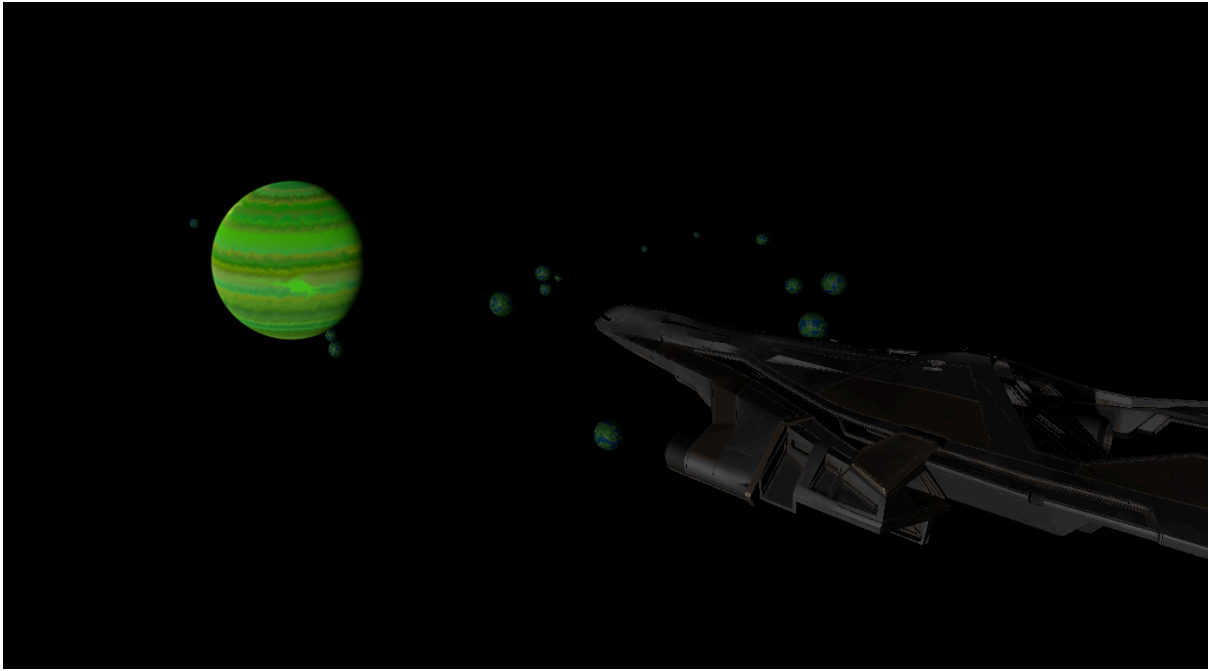


Example terrain with traversal shadow of the moon and planets as background.

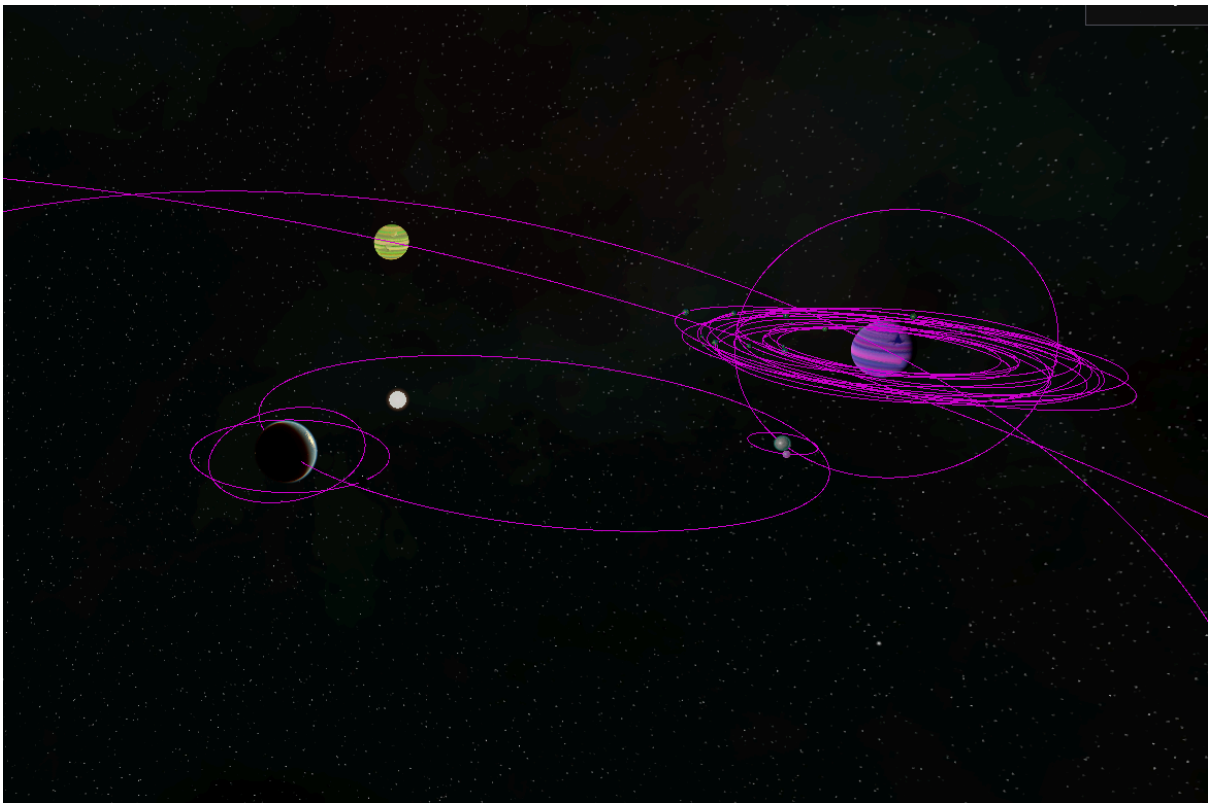


Dynamic Light sources on spaceship

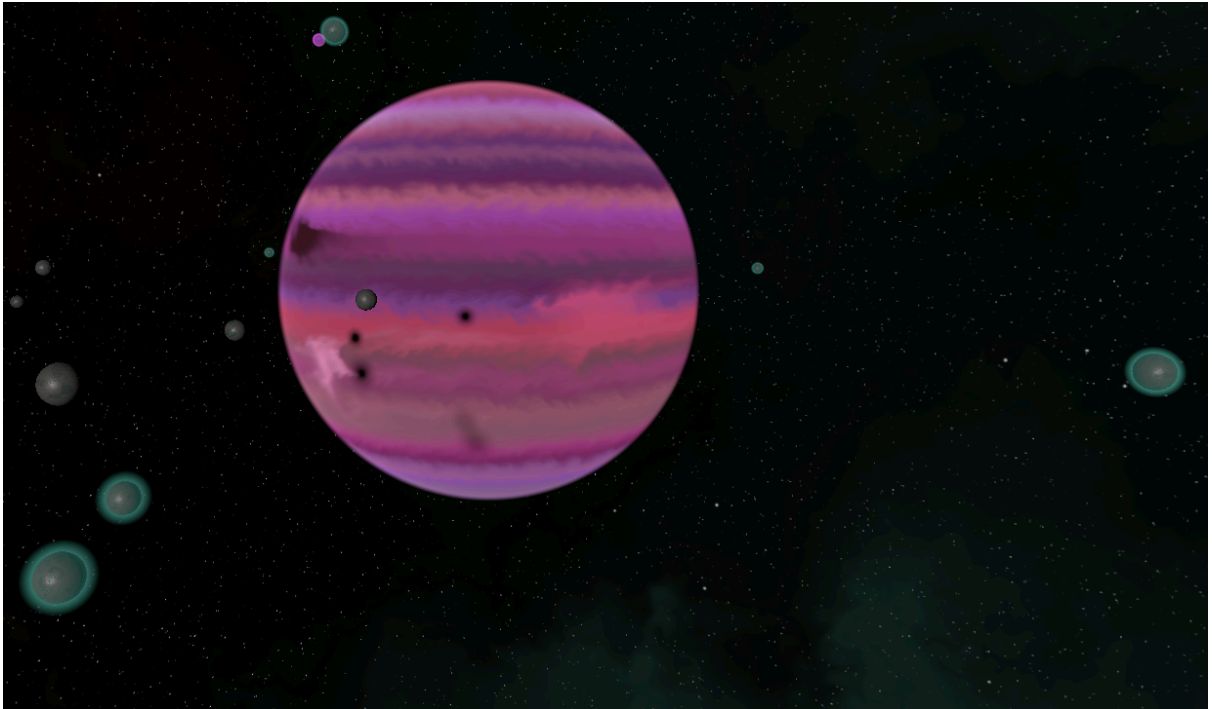




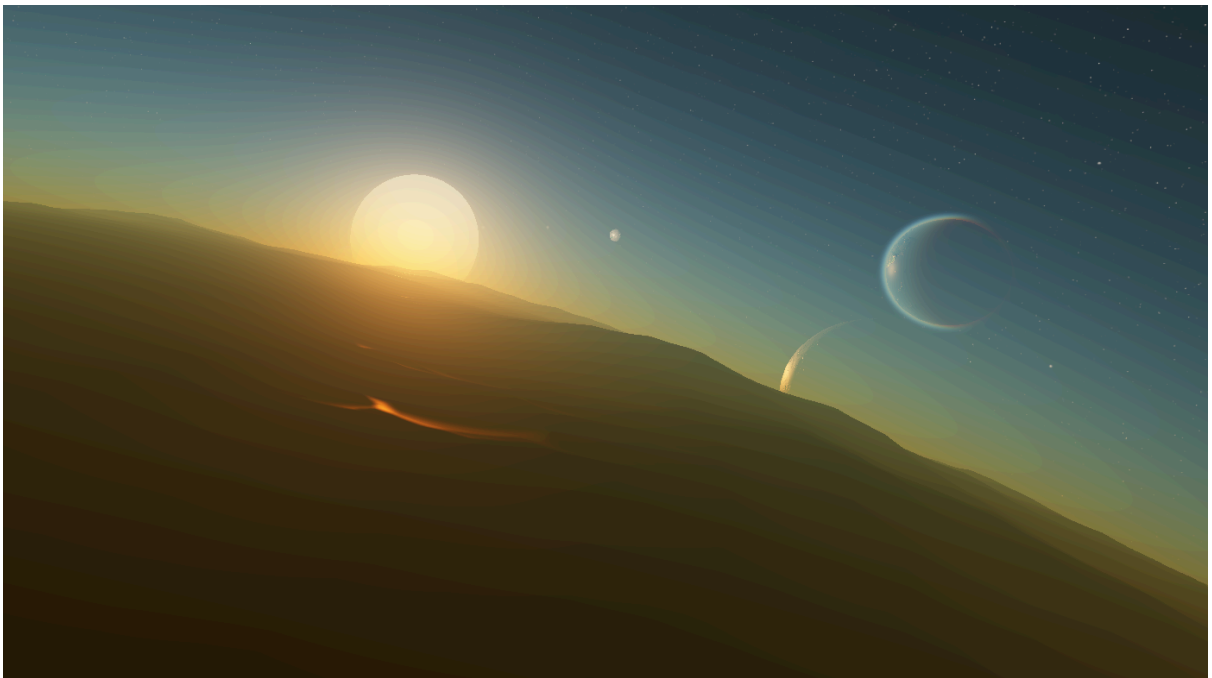
spaceship (old picture) transit arrival



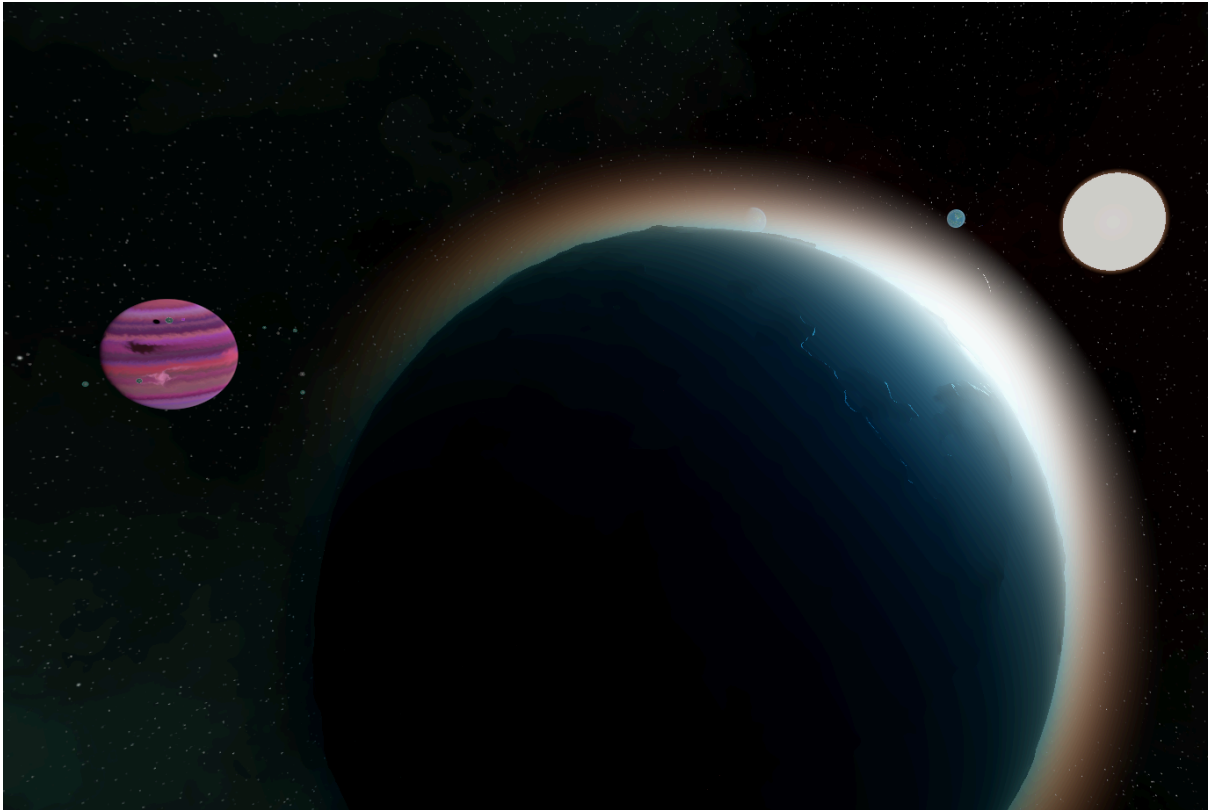
Orbit visualization



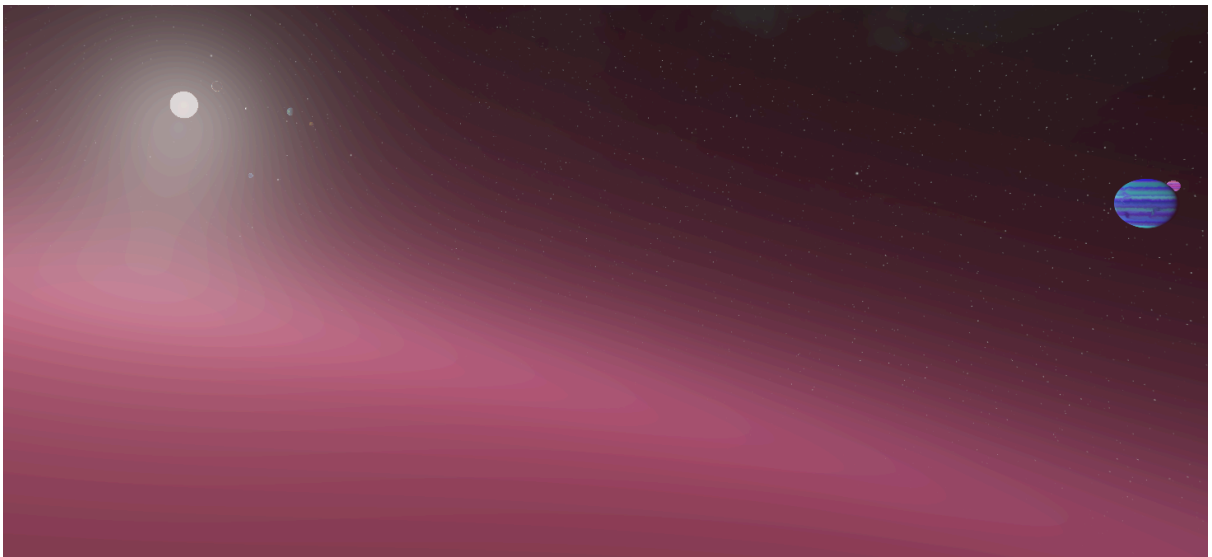
Planets/moons create shadows on other planets/moons



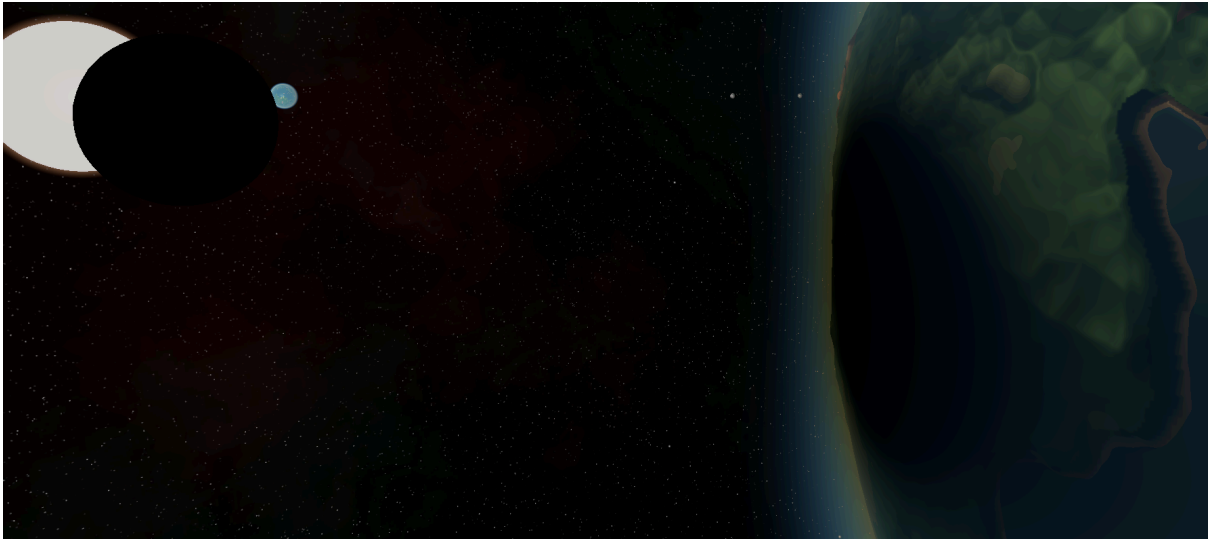
Example View from earth



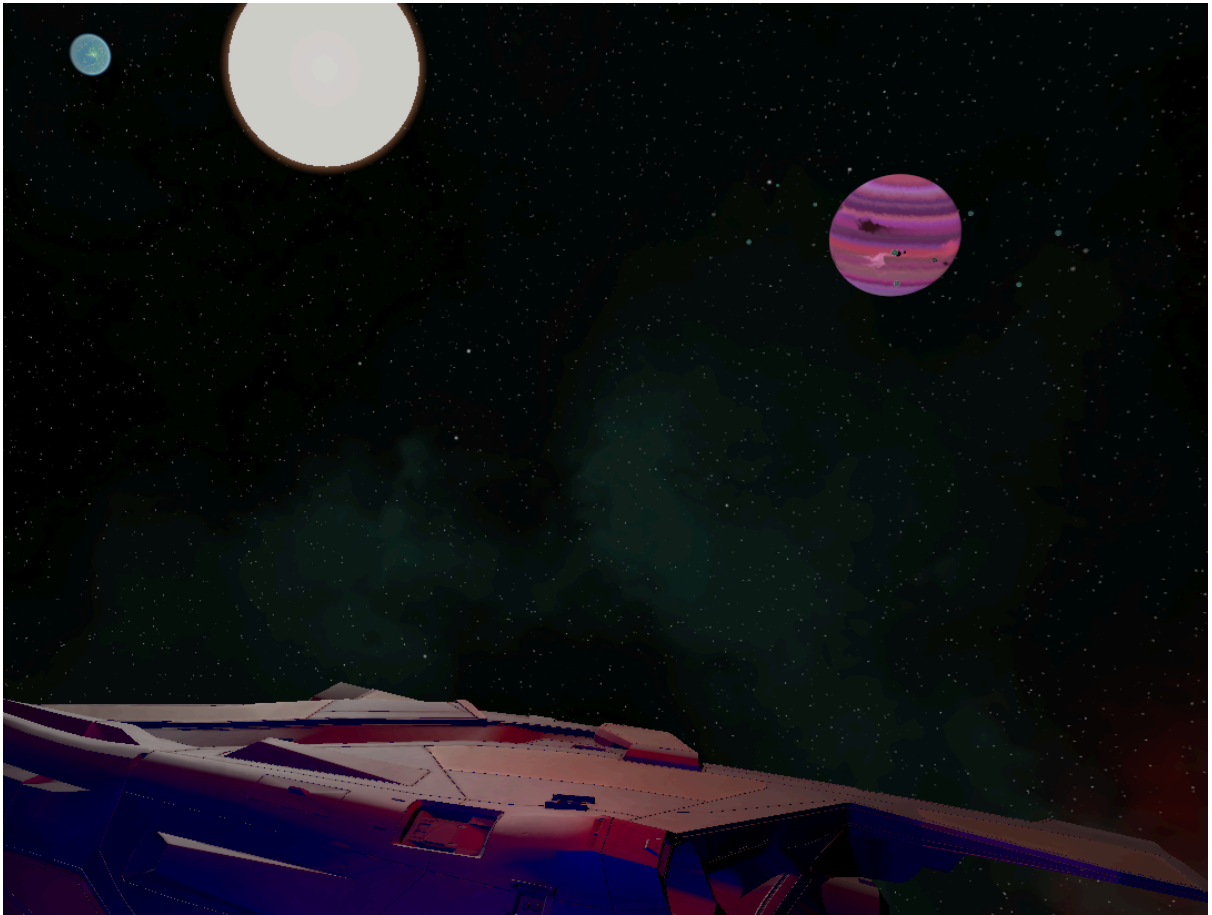
Color band (orange-blue) from atmospheric scattering



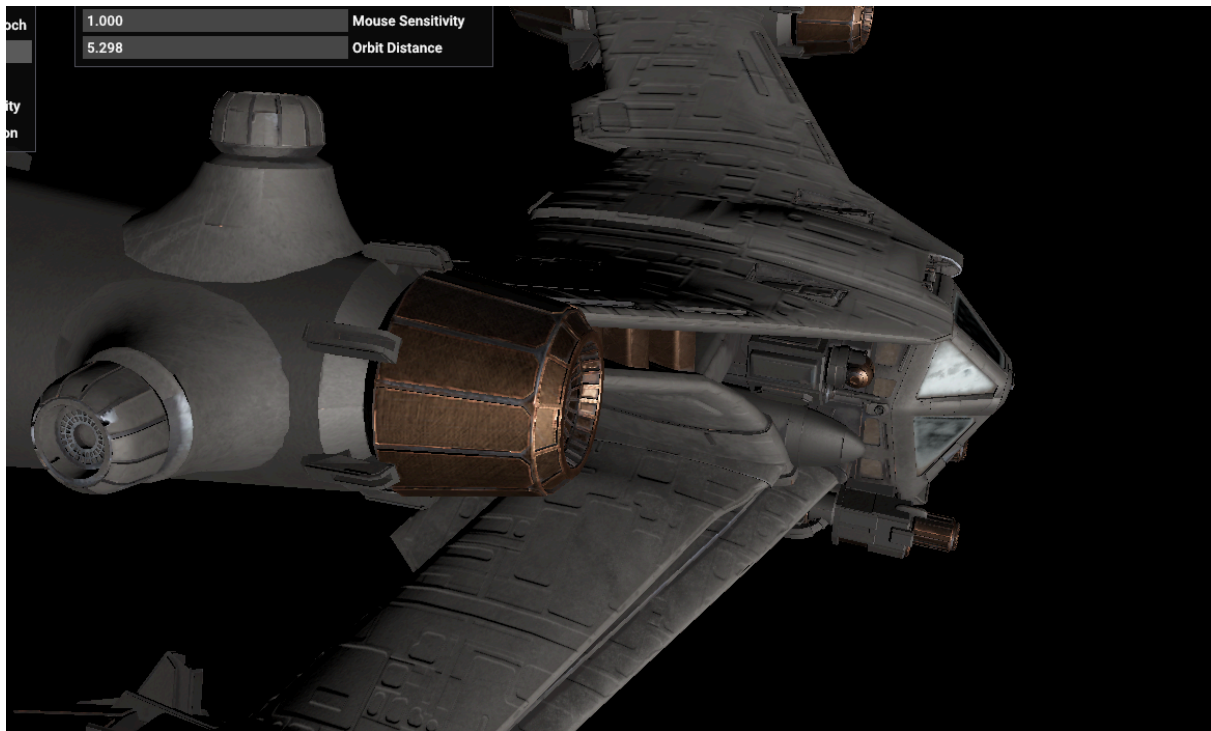
*View from inside jupiters atmosphere*



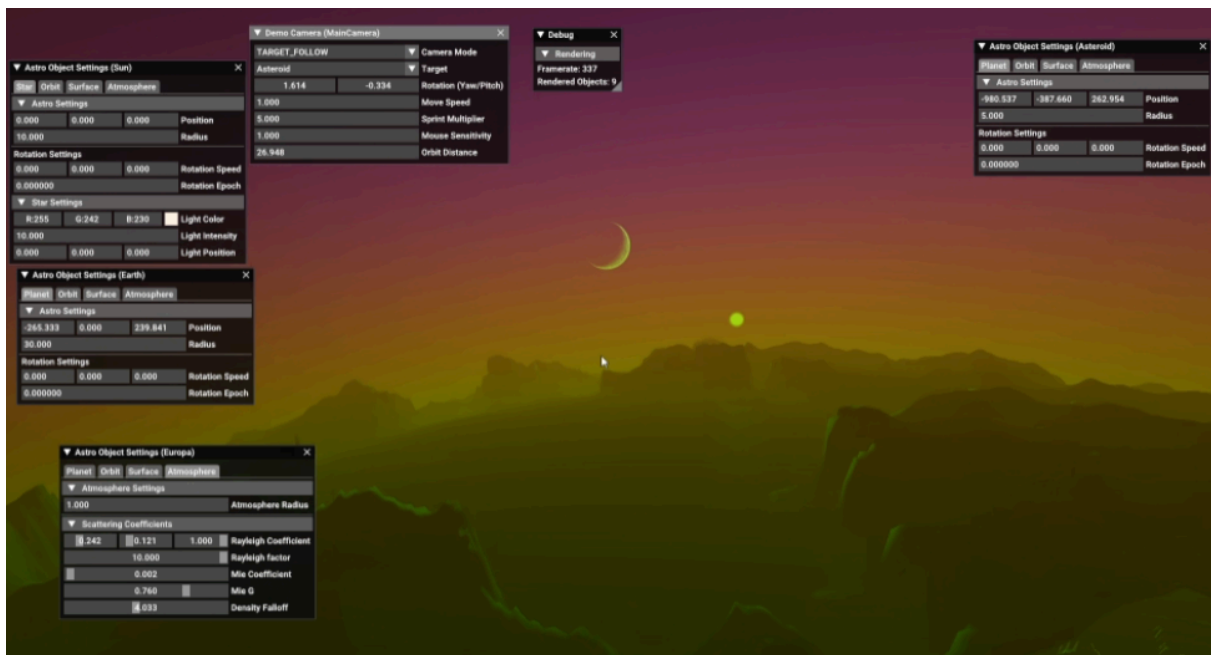
*solar eclipse on planet*



*Spaceship sun reflection*

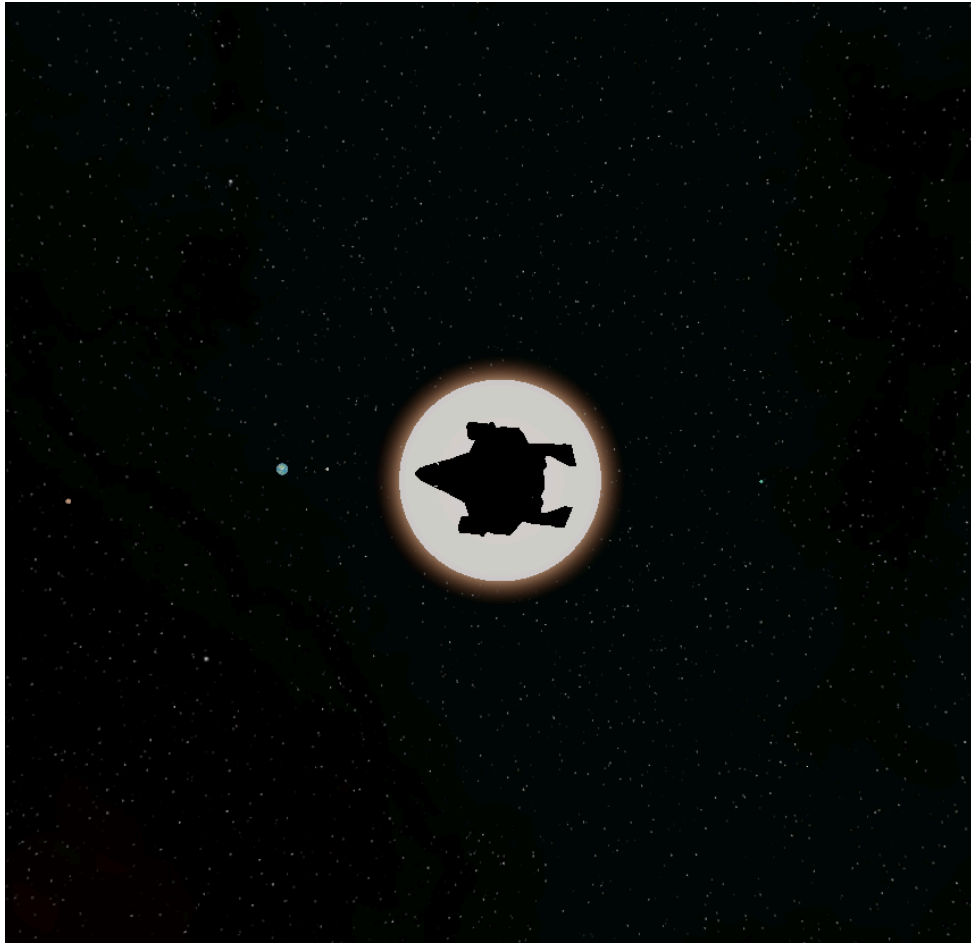


*Other spaceship we tried*

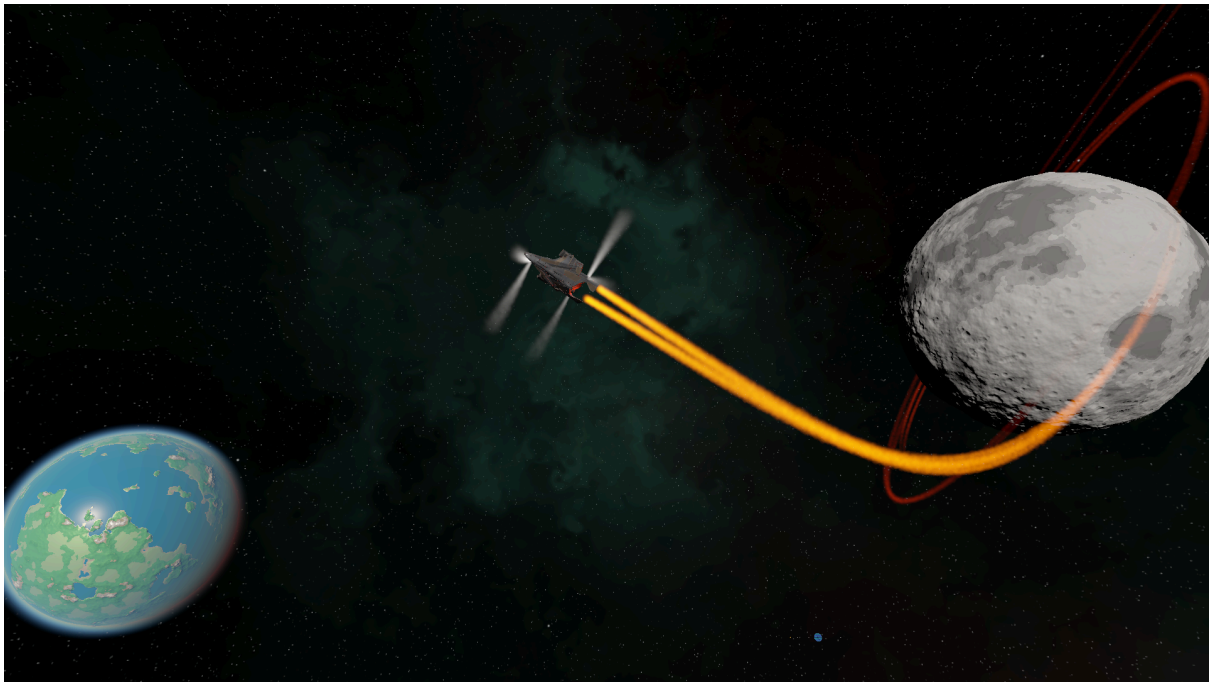


*ImGui Debug Inspector to modify all values of our objects and components*



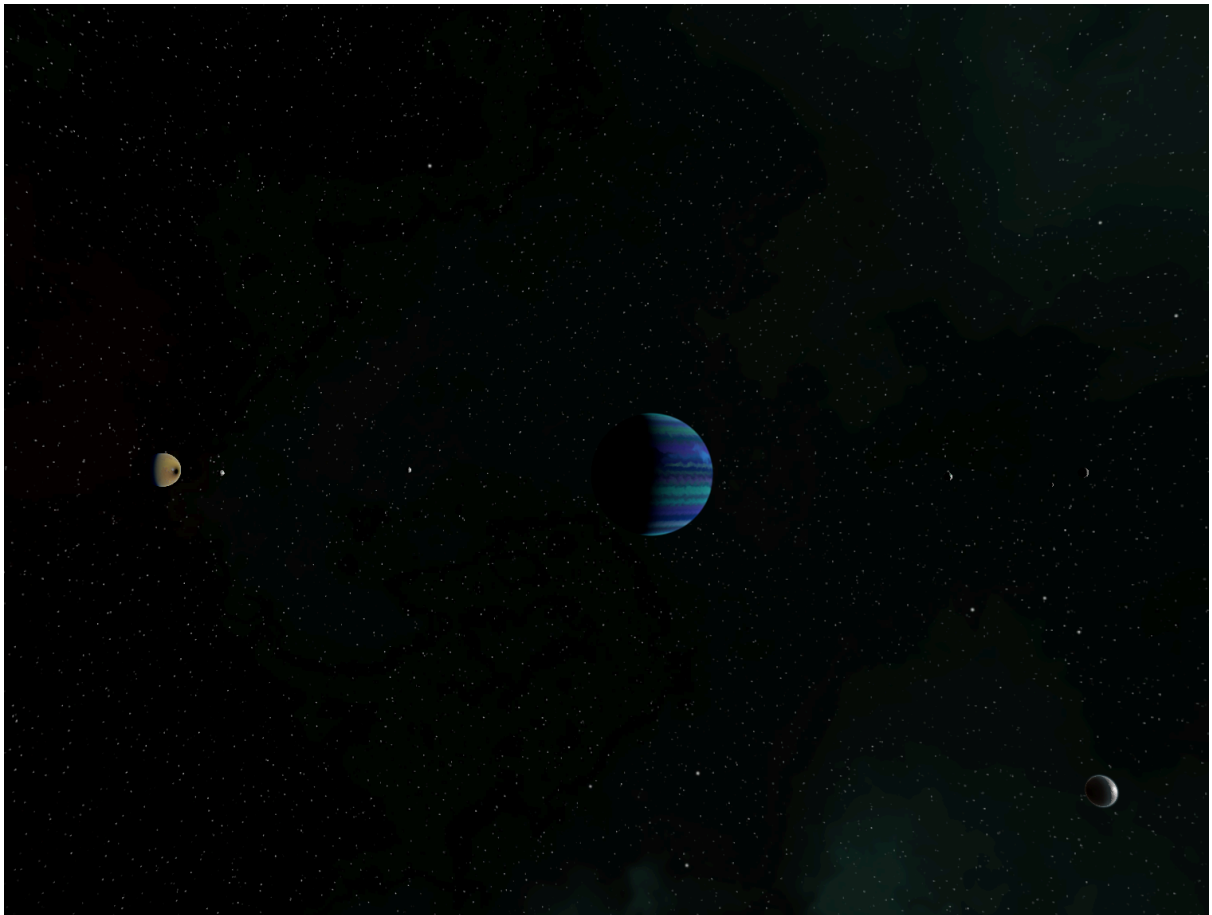


*NaN NaN NaN NaN ... Batman!*

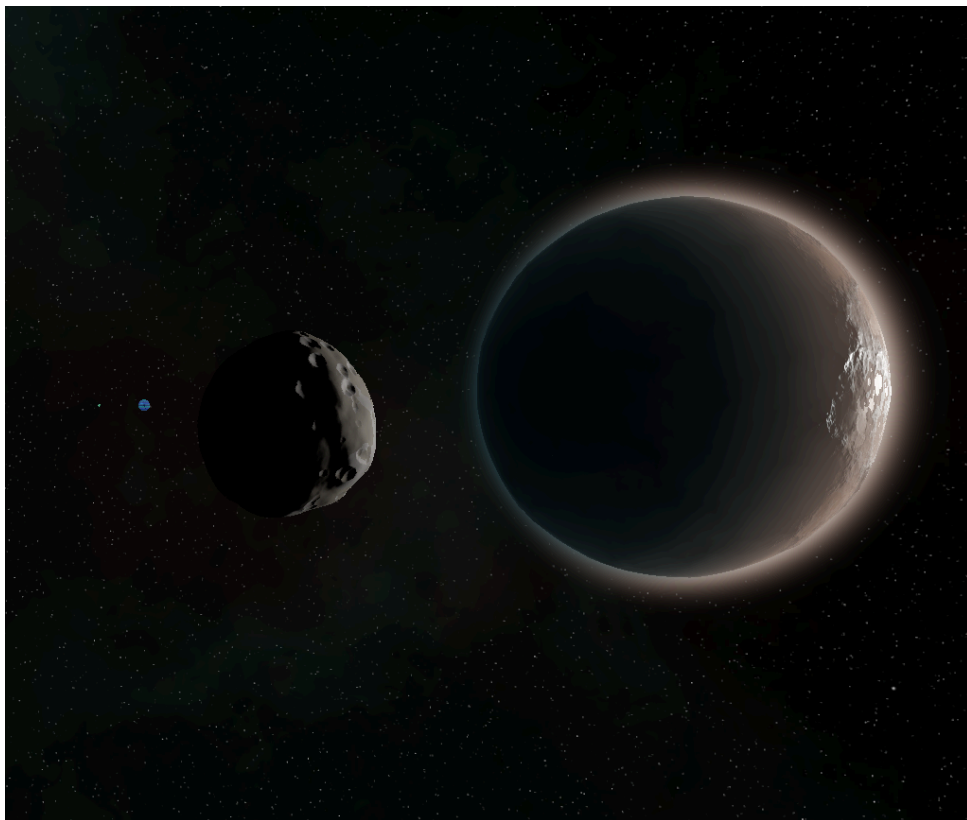


*Luna+Earth Departure*




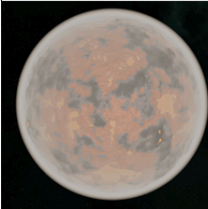

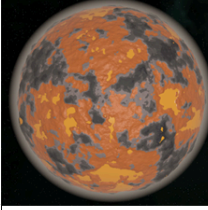





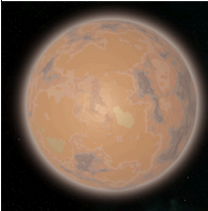




*Moon on moon shadow*



*The dark side of mars*

Mercury						
			<a href="https://en.wikipedia.org/wiki/Mercury_(planet)">https://en.wikipedia.org/wiki/Mercury_(planet)</a>			
Venus						
						
			<a href="https://en.wikipedia.org/wiki/Venus">https://en.wikipedia.org/wiki/Venus</a>			
Earth						
			<a href="https://en.wikipedia.org/wiki/Earth">https://en.wikipedia.org/wiki/Earth</a>			
	Luna					
			<a href="https://de.wikipedia.org/wiki/Phobos_(Mond)">https://de.wikipedia.org/wiki/Phobos_(Mond)</a>			
Mars						
			<a href="https://de.wikipedia.org/wiki/Mars_(Planet)">https://de.wikipedia.org/wiki/Mars_(Planet)</a>			

*“artistic impression”*