



CGUE Submission 2 Description – Supernova

Post Gameevent Note:

After the first feedback, we updated the following components:

- Win/Loose Condition
- Correct colors on the asteroids
- Increased Motion Blur visibility

Implementation:

Gameplay:

The Helix galaxy currently in a peaceful state. After exploring the mysterious cubes Isaac's current mission is to gather probes by destroying interestingly colored asteroids. Yes, even the life of a sky-traversing scientist/adventurer can be dull at times. But there is one thing, Isaac does not know yet. Precisely on Wednesday, June 28th, (Earth Time) something is about to happen.

Complex objects:

There are different complex objects in the game. First of all, there is the spaceship. With almost 26.000 vertices it is the most complex object used. The asteroid is another complex object and the sun can also be counted as one.

Animated Objects:

Isaac Callahan's spaceship, SS Asteroid Destroyer, uses the newest radar technology to navigate its way through space. The radar sits on top of his spaceship and rotates using hierarchical animation.

View-Frustum-Culling:

Bounding Spheres are used for View-Frustum-Culling and the view frustum is calculated using a geometric approach (<http://www.lighthouse3d.com/tutorials/view-frustum-culling/>). Due to numerous space-time anomalies in the Adenine system, objects sometimes seem to disappear without rational cause.

Experimenting with OpenGL:

For all post-processing effects Frame Buffer Objects are used. The FBOs use different amounts of color-buffers and sometimes even depth-buffers. A special type of FBOs are the Ping-Pong-FBOs used for blurring during the bloom effect.

Blending is used for particles and text overlay. Mip Mapping and Texture Sampling is used when loading textures using FreeImage.

Features:

- Fly through the endless nothingness of the Helix Galaxy.
- Destroy asteroids to gather probes.
- Finish the level by destroying all the asteroids.

Illumination & Textures:

The scene is illuminated using Point-lights, Directional lights and Spot-lights. The sun is simulated using six directional lights that make the scene look like being illuminated from one central point. At the front of the spaceship spot-lights are placed, that illuminate everything in front of them.

The spaceship is textured using FreeImage. The skybox is textured using imagery from <http://amethyst7.gotdoofed.com>.

Additional Libraries:

FreeImage for image loading (<http://freeimage.sourceforge.net/>).

Assimp for model loading (<http://assimp.sourceforge.net/>).

FreeType for text overlay (<https://www.freetype.org/>).

GLM

GLEW

Effects:

Particle System:

The particle system is based on the tutorial from “Megabyte Softworks” (<http://www.mbsoftworks.sk/index.php?page=tutorials&series=1&tutorial=26>). The implementation uses geometry shaders and transform feedback buffers to update the particles on the GPU. In addition, we implemented numerous methods to change the particle system in each update step.

The particles are used to simulate the exhaust of the spaceship, as well as the laser.

Bloom:

Bloom is based on the tutorial from “Learn OpenGL” (<https://learnopengl.com/#!Advanced-Lighting/Bloom>).

It is especially visible when the spaceship faces towards the sun.

Lens Flares:

Lens Flares are based on the tutorial from John Chapman (<http://john-chapman-graphics.blogspot.co.at/2013/02/pseudo-lens-flare.html>) . The texture for the colors of the lens flares is self-made using Gimp and the parameters are set to our liking.

Lens Flares are visible when looking at the sun or other very bright spots.

Motion Blur:

Motion Blur is based on the tutorial from Nvidia (https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch27.html). Motion Blur is implemented as a post processing effect. The view-projection-matrix of the previous frame is saved and the fragment shader compares both the current inverse view-projection matrix and the previously saved one for each world-space point to get a blur vector.

Motion Blur can be observed well while rotating or looking at the stars in the background.

Spot lights:

Spotlights are based on the tutorial from “Learn OpenGL” (<https://learnopengl.com/#!Lighting/Light-casters>). Notable are the soft edges of the spotlight, achieved by calculating a smooth transition from pixels that are illuminated by the spotlight and those that are not.

Spotlights are well visible when moving towards an asteroid.

Models:

The spaceship and sun models were generated using Blender and the spaceship texture was painted with GIMP using the UV-Map generated by Blender. The asteroids are downloaded models.

Controls:

W – accelerate

S – throttle

A – rotate left

D – rotate right

Q – rotate radar right

E – rotate radar left

Move Mouse – pitch the spaceship

Left Mouse Button – shoot laser

F2 – Show Frame time

F3 – Toggle Wireframe

F8 – Toggle View-Frustum-Culling

F9 – Toggle Blending