

Final Frontier

Important

Due to restrictions on some ATI graphics cards (they do not allow more than one sampler type per texture unit), shadows are automaticly(!!!) disabled when this feature is not available. Please use an NVidia card for testing!

Gameplay

Goals:

The goal of the game is to reach a score as high as possible, succeeding in semi-randomly generated levels by destroying the eye of the Eterni nest in the middle of the map. The eye is invulnerable as long as it is being protected by the giant red turret batteries. The higher level you play, the more red batteries there are. However in the beginning levels the eye will be unprotected.

In order to destroy the eye or any turret battery on the field, you need to charge up your laser. Your weak standard weapon is only good for killing the blue and red flying saucers.

To charge your laser, kill a flying saucer and pick up the powerup it drops. This will give you ONE shot with your laser. Aim it well! After that you will need to pick up another powerup.

Note, that the green turret batteries do not need to be destroyed in order for the eye to become vulnerable. Only the giant red ones need to explode.

As soon as you blow up the eye with your laser you will advance a level.

Features:

- semi-random levels which get more difficult the further the player gets
- Arcade style Space flight
- 3D Sound

Controls:

Game Controls:

The game is designed to be played with an XBox 360 controller, using the right/left trigger to accelerate/decelerate, button A to shoot the standard gun and button X to fire the siege laser if available. The left analog stick steers the ship up, down, left and right.

However the game works just as well with any gamepad and even a joystick. Please note that the game chooses one(!) gamepad for input. So if you have multiple gamepads plugged in, only one of them will work.

Esc: open game menu

Debug Buttons on the Keyboard

Mandatory shortcuts:

F1: Show Help

F2: Framerate on/off (also turns debugmode on/off)

F3: Wireframe on/off
F4: Textur-Sampling-Quality: Nearest Neighbor/Bilinear
F5: Mip Mapping-Quality: Off/Nearest Neighbor/Linear
F8: Viewfrustum-Culling on/off
F9: Transparency on/off

Special Commands can be activated by holding the **RIGHT(!!!) ctrl**-key and press **TAB** while playing. A debug-textoverlay will appear and the following commands will be available:

C: Switch between a freely movable Camera(**WSAD**/Mouse (button1 pressed), Roll with **Q/E**) and the Ship cam
T: toggle static Level-of-detail. As most of our objects are big the distances when lod comes into play are great. However the asteroids are a good example to show.
B: Show collision boxes, bounding boxes and the view frustum sphere.
V: This is to show off our view frustum culling. While you are using the free-moveable camera (**C**), this command will set the ship camera for view frustum culling. So if you turn the ship around and watch it from the distance, you can see objects popping in and out. The View-frustum sphere can be seen when pressing (**B**).
G: toggles god mode, so you can freely play around without taking damage
U: Shoot a shot right where you are looking at
H: Toggle between wireframe, no-draw and normal drawing mode
I: reset the ship position
R: Reload shaders
J: Self-destruct ship

In addition you can reload all settings files when in debugmode (**r-ctrl + TAB**) when you press "**S**" while in the main menu (the one saying "New Game", not "Continue")

Settings file:

We do not have a settings menu, but you can set settings like fullscreen, screen size and the level you want to start at (we do not recommend starting at level 20 right at the beginning :))

Animated/Complex/Transparent Objects

The main Structure and the Ship are both hierarchical objects, where in one case the laser cannon of the ship can be opened/closed by picking up powerups and the Eye of the main Structure Always follows the Ship. Furthermore the siege turrets aim their cannons at the ship. You can see this well when flying close to them. The flying saucers move around on fixed path and look at the player if he is close enough.

View-Frustum-Culling

In the debug output (**RCTRL+TAB**) there is a line showing how many objects are currently culled.

Static Level-of-detail

We use 2 different models for our objects which are switched out if the camera is far away. The distance when the low-detail meshes are being used depends on the object's size. you will rarely see it (except on asteroids) as it made little sense with all the huge object we have.

Advanced OpenGL features

- VAOs, VBOs
- Mip-Mapping
- FBOs
- Bilinear Filtering
- Geometry shaders

Lighting:

The Scene is lit by a Single Directional lightsource, the objects are shaded using phong shading. Additionally shadows are cast by Shadow maps, using three cascading mapsscaling the scene by factor 4 in each map. Input came from the shadow mapping tutorial in the repetitorium, [this](#) tutorial from <http://www.opengl-tutorial.org/> and [this](#) article on msdn.microsoft.com. Shadow acne is reduced by frontface culling and an light/normal-angle dependent bias, the shadow edges are smoothed using a combination of pcf and an additional 3x3 gauss-blur. Transparent objects are treated like solids and cast a full shadow.

Special Effects:

- Bloom
- Glow
- Particle effects:
 - Explosions consist of multiple animated point sprites
 - Sparks when the laser hits something are the second particle effect
 - A “particle-snowfall” supports the impression of movement in the game
- Normal mapping
- Specularity maps: Specularity properties of the materials are defined by texture maps
- Lightmaps: used to define areas which are always lit and glow.
- Shadow maps

Additional libraries:

- **devIL**: used in the TextureLoader to load the Textures into memory
- **openAL**: used for everything soundwise: Static music and FX sources, and during gameplay 3D Sound