

Final Game Distant Refuge

Controls

- Left: A or Left Arrow
- Right: D or Right Arrow
- Up: S or Down Arrow
- Down: W or Up Arrow
- Shoot missiles: Left Mouse Button or Space
- Progress UI: Enter
- Toggle HUD: Tab
- Toggle View Frustum Culling: F8
- Close game: Escape

Development Status

All planned effects and features have been implemented, and the game is fully playable. The most notable additions are new obstacles with effects (vertex shader animation, physically based shading, procedural texture), multiple levels with finished structure and goal as well as UI. Some special additions from the initial document (e.g. shield and laser) are unfortunately not implemented yet due to time reasons.

Implemented Features

- 3D Geometry: The game uses an object loader and already has different complex objects (e.g. satellites and spaceships)
 - Playable: Movement and missile shooting is implemented
 - Min. 60 FPS and Framerate Independence: Game runs with 60 FPS and uses a clock to measure time between the frames
 - Win/Lose Condition: Survive all the levels indicated by the UI and escape to a new home planet (win screen will then appear)
 - Intuitive Controls: Standard movement keys (WASD) and Shooting with Left Mouse Button or Space as well, Tab to toggle HUD and Enter to progress UI are also common
 - Intuitive Camera: Camera follows the player
 - Illumination Model: Directional Light, Spot Light and Point Light are implemented, Cook-Torrance Model, Materials
 - Textures: Every object has a texture, trilinear filtering and mipmapping are implemented
 - Moving Objects: Player and enemy spaceships are moving
 - Documentation: This document
 - Adjustable Parameters: Fullscreen-Mode and Screen Resolution can be set in the settings/window.ini file (just like with the GCG framework)
-
- Advanced Gameplay: Player has three lifes (resetted every level) and can be hit by obstacles or enemies, he can dodge obstacles and destroy obstacles by colliding with them or shooting, Player must survive multiple levels
 - Collision detection: Walls, Collisions with obstacles and enemies
 - Advanced Physics: Collision Events for hitting objects, objects can collide (for instance player and asteroids) the movement will be adapted according to the physics

- View-Frustum Culling: Can be toggled with F8, the number of the currently culled objects is in the window title
- Heads-Up Display: Level Progress, Level Overview and Health Bar are visible in the HUD (bottom left corner)

Implemented Effects

- GPU Particle System using Compute Shader: Emitted particles from the exhaust of spaceships
- Vertex Shader Animation: Multiple objects like stars use it, looks like spikes that move
- Procedural Texture: Textures for planets are calculated in the beginning when the game loads and are used in every level
- Physically Based Shading: Cook-Torrance Model has been implemented for all objects
- Bloom/Glow: e.g. planets and particles have bloom activated

Used Libraries

- Object Loader: tinyobjloader (<https://github.com/tinyobjloader/tinyobjloader>)
- Physics: PhysX (<https://github.com/NVIDIA-Omniverse/PhysX>)