Space Cleaner

3000

A

Bernd Artmüller / Bálint Kovács

Game

Our game Space Cleaner 3000 is a 3D dexterity game, where the player has to take control over a spaceship tasked with ridding the space-neighborhood of Earth from the space junk orbiting it. The player has to collect all the pieces of garbage, which are the in turn immediately burnt up by the spacecraft, making it go gradually faster and have a longer fire tail behind it. This fire tail and the floating meteorites pose a danger to the spaceship, so that the player has to navigate it around these objects avoiding a collision.

The development was unfortunately plagued by circumstances, which considerably hindered the progress and caused some problems prompting a partial completion. However, if this state can be considered acceptable, the remaining time until the Game Event will be spend on further development towards completion.

1 Brief description of the implementation, in particular a short description of how the different aspects of the requirements (see above) were implemented.

Gameplay

The spacecraft can in this stage be flown in the space sector marked by the green boundaries. If the boundaries are hit, teleportation occurs to the opposite side of the sector. We concentrated our efforts on the graphical effects, hence the gameplay aspect of the game fell relatively short. Collision detections works however, hitting an object resets the ships position.

Effects

Motion Blur has been implemented using an universal frame buffer (refactored for further use with Shadow Mapping). Effect can be observed in the free camera view.

Particle effects: Particle generation is working.

Shadow mapping and bloom/glow effects are not yet ready to be observed.

Controls

The implementation of a predictable and well usable control scheme posed one of the biggest obstacles during development. After some considerable effort a camera control scheme could be developed which makes a unified panning between the sides of the cube possible, essentially "turning" space in the desired direction. Unfortunately, the ship controls couldn't yet be unified in such a way, which make navigating the ship more of a challenge.

The current controls are as follows:

W, A, S, D : Turning "space" in the desired direction (panning the camera around the cube)

ARROW KEYS : Turning the spaceship around. Results can vary depending on the side the camera is positioned on

1, 2, 3, 4 : Setting the flying speed of the spacecraft

7 : Selecting free camera

8 : Selecting fixed position panning camera

9: simple cockpit view (was added to observe shadow mapping)

2. How and which objects were illuminated (description of light sources) or textured.

Every object of the scene is illuminated by a single light source, meant to represent the sun as the sole illuminating object. The rotating meteorites showcase this, as well as the panning around the cube, whereas the non-illuminated sides of the spacecraft can be observed.

3. What additional libraries (e.g. for collision, object-loader, sound, …) were used, including references (URL) (see restrictions)?

> The following libraries were used for development:
> GLFW - http://www.glfw.org/
> GLEW - http://glew.sourceforge.net/
> glm - http://glm.g-truc.net/
> FreeImage - http://freeimage.sourceforge.net/
> Assimp model loader - http://assimp.sourceforge.net/
> DevIL - http://openil.sourceforge.net/

4. What Tools have you used to create the Models (Maya, 3DS MAX, …).

Game assets have been created in 3D Studio Max from multiple primitives with separate materials, which then were combined into one texture and mesh and exported as .obj model data.