SnakeBox

Maximilian Langer, Daniel Pucher

June 18, 2014

SnakeBox is a 3D snake-like game, where the player has to control a virtual snake that has to eat energy orbs to grow stronger and unlock new levels. There are six levels in total, which are accessible through the six sides of the so called game cube. The energy orbs always appear in another level of the game cube, signalized by colored light beams.

1 How to play and Controls

The player has to collect energy orbs (red pulsative balls) to grow his ever hungry snake. While collecting he has to avoid eating himself or the orange grid.

W	Turn Head Up
S	Turn Head Down
A	Turn Head Left
D	Turn Head Right
Q	Roll Head Left
E	Roll Head Right
X	Look back
C	Switch between First and Third Person
P	Toggle Pause
M	Toggle Music

Additional Controls:

F2	Frame Time on/off
F3	Wire Frame on/off
F4	Textur-Sampling-Quality: Nearest Neighbor/Bilinear
F5	Mip Mapping-Quality: Off/Nearest Neighbor/Linear
F6	Anisotropy on/off
F7	Print performance on/off
F8	Frustum Culling on/off
F9	Transparency on/off

2 Implementation

2.1 Complex Objects

The Snake head is a complex object designed in Blender and loaded via assimp into the game. (All models are loaded via assimp...)

2.2 Animated Objects

The Snake (Head and Body) moves using calculated positions. The Ring (and teeth) rotate around the haed in a hierachical way (using already computed matrix of Snake-head).

2.3 View Frustum Culling

View Frustum Culling is used on Snake Body parts, portals and food. It's implemented using the clip space approach (lighthouse3d.com

2.4 Transparency

Transparency is used in the texture of the world grid (to show stars behind) and the food is also slightly Transparent.

2.5 Experimenting with OpenGL

VBOs, VAOs, FBOs, Mip Mapping and Texture-Sampling-Quality are used in the programm. The two later can be changed using F4 and F5. All Objects use VBOs and VAOs. FBOs are used for rendering the portals.

2.6 Gameplay

The game is playable. There is a Score and Game Over.

3 Features

- Collect energy balls to grow
- Walls and snake collision
- 3rd and 1st person
- Cubseption (Transparent Grid Cube in Universe/Stars-Cube)
- Snake body parts are rotated the right way
- Portals connecting World to Levels in Cube

- Particles when eating energy balls
- Self made sounds
- Shadows in Levels
- Effect when passing portals

4 Illumination and Textures

All snake parts and levels are illuminated using Lambert-Phong shading. The light source for these is the food. The universe cube and the grid cube are textured using freeimage to load images. The food uses a cloud texture (loaded with freeimage) to displace it's vertices in the vertex shader.

5 Effects

5.1 Protal Effect

Portal Effect connecting several "Levels". The first implementation used the stencil buffer based on this tutorial: Mini-Portals. The final implementation now uses FBOs and Texture Mapping in Screen Coordinates. The levture slides where used.

5.2 Particle Effect (CPU)

The particle effect (CPU-Based) was rendered with one object renderd with different Model Matrices. No source was used.

5.3 Screen Effect on Portal Switch

The whole scene is rendered into a texture for Screen Effects. The effect when passing through a portal is done in the shader. Source: opengl-tutorial.org

5.4 Shadow Maps - Omni-Directional

The shadow maps are drawn on a depth-only cube map and used in the level and snake-shader afterwards. Source: TU Wien Omni-directional Shadows

6 Additional Libraries

assimp used for loading obj-files exported with blender

freeimage used for loading png-textures

glm used for vector and matrices handling and other math

glfw used for GL window and event handling

glew used for OpenGL

sfml used as sound library

cmake used as build system