Weave - 1st Submission

Pajer Herbert, Schaefer Peter

June 28, 2016

1 requirements

1.1 Free movable camera

This was realised by using the GLM. Transform functions to create model,view and projection matrices wich get used by the shader.

1.2 Moving objects

The player is freely movable through the world by using WASD and Space or the left mouse button for jumping.
Turning levers will open doors.

1.3 Texture Mapping

Textures are loaded with the FreeImage library.
Models are made with Blender and their .dae files which include their UVs (as well as their normals) are loaded with the assimp library.
Materials are saved as Vectors with 4 entries: The intensities for the 3 different light source types and one for the glossiness.

1.4 Simple lighting and materials

There is one point-lightsource, one ambient light source and one directional light source.
The ambient and directional lights are programmed dynamically and can be adjusted. There is one point-light hardcoded into the shader files. It is located near the lever in the first level.
Only the directional light casts shadows.

1.5 Controls

The camera can be controlled with the mouse.
You can control the player with the WASD Keys and jump with Space or the left mouse button.
Levers can be pressed with the E key.
If you hold the shift-key you can reverse time, meaning that if you jump down somewhere you can get back up by rewinding time until you're back to before you jumped.

Pressing F2 shows the frames per second in the window name.
Pressing F3 toggles Wire Frames.
Pressing F4 toggles the texture sampling quality.
Pressing F5 cycles through the Mip Mapping qualities.
Pressing F6 makes the Bounding Meshes and Event Boxes visible.
Pressing F8 toggles frustum culling on and off.
Pressing F9 toggles transparency on and off (the doors are partly transparent at the top).
Pressing F10 toggles full Screen.
You can exit by pressing ESC twice and you can restart the level by pressing ESC and then R.

1.6 Basic Gameplay

The player can move around the world and jump around. Collisions work properly and you can stand on ground and all that good stuff. Colliding with the boar will lose the game. This can be reverted by turning back time. Levers can be turned by pressing E. Green glowing objects can't be reverted back in time. Colliding with the win-box loads the next level. There are currently two levels.

1.7 Effects

Shadow Mapping with PCF (1.5 points). This is only done on the directional light but not on the pointlights.
Motion Blur (1.5 points) when turning back time.
Bloom (1 point): There is always a bit of bloom active, but there are also special green glowing objects in the second level that can't be turned back in time.

1.8 Complex Objects

We have several complex objects like the player, the level and the boulders.

1.9 Animated Objects

The player has a move animation. All other moving objects are simple transformations (like doors opening and levers turning).

1.10 Frustum Culling

This is realized with Bounding Spheres. The Number of drawn objects is written while displaying the FPS (F2).

1.11 Transparency

The top of the doors are transparent. And at some points you can look through the doors.

1.12 Experimenting with OpenGL

VBOs, VAOs, FBOs, Mip Mapping and Texture Sampling has all been implemented.

2 Features

Colliding with any of the boars will lose you the game and turn the screen red-brown.
You can also jump on the to kill them.
There are two levels. Winning the first one will transport you to the second one. The second one doesn't have a win condition implemented, but the goal is to open and free all boars. In order to do that you have to solve puzzles.

3 Illuminated Objects

There are multiple complex objects, all of which have textures and are illuminated. Also, there is a skybox which is not illuminated. Bloom is implemented, which is visible on the lever where the point light is located.

4 Libraries

freetype 2.3.5.1
FreeImage 3160
zlib 128
glm 0.9.7.2
glew 1.13.0
assimp 3.2
sdl2 2.0.4
bullet 2.83.7