

Pigpee - piss 'em all off

Gameplay

Our game is a third person shooter. The player controls a pig and has to shoot (pee on) all people on the island. If you get them all, you win!

Complex Objects

We load the models into our program via the ASSIMP-Modelloader. We created most of the objects ourselves in Blender and tried to keep them low-poly. Some objects were retrieved from the internet, e.g. the people and boat.

Animated Objects

The wings of our pig are moving up and down. Obviously it's a flying pig.

View-Frustum-Culling

The View-Frustum-Culling is still in work. It used to work, now it doesn't - improvements will be made asap.

Features

Pigpee provides sound-effects when people are hit, background music and a bird-eye view map for orientation which will be improved. To be implemented in the very near future: goodies and "badies". Future enhancements: lighting, shading, water reflection.

Illumination

Our scene is illuminated by one static light source. The water is reflecting. All objects have a texture. The textures are being loaded with the FreeImage library.

Libraries

We used several libraries for effects and features as 2D Text rendering (freetype), modelloading (ASSIMP), sound (irrKlang) and FreeImage (Textures). Since we are using a 64-bit executable, our libraries are compiled only 64-bit with Visual Studio 2015 as well.

Effects

Our effect points consist of:

Water (+Fresnel-Shading,Normal Mapping)	0,5
---	-----

Water Reflection	1
Water Refraction	0,5
CPU-Particle System (+Instancing)	0,5
Shadow Maps (with PCF)	1,5

Water Effect:

Concerning water effects we used framebuffer objects for rendering the scene from different angles. The water quad uses those framebuffer objects as textures together with a water surface texture. For water normal mapping a normal map of the texture is used, and specular light is based on the normal map. For distortion the texture is sampled based on the strength of the wave and a scalar. Fresnel scalar is calculated by angle between water normal and viewing vector.

The best view to see normal mapping and water effect is to look to the sun horizon and Fresnel by flying to the water and looking down at the water-surface.

Shadow Maps:

You can see the shadows especially good when looking at the palm-trees.

We calculate the depth of the objects in a framebuffer and bind these values to a texture. Once we draw the scene, we retrieve these depth-texture-values and take them into account when calculating the lighting. To improve mapping we used methods to prevent shadow acne, peter panning and the PCF (Percentage Closer Filtering) to prevent strong shadow edges.

Particle Effect (Pee):

You can see the particle effect, when peeing by pressing space.

We have an array of particles. We only update the center of these meshes, which is much more efficient than updating the whole mesh. The particles are only "alive" for a certain amount of time and are reused after they've "died".

Particle Effect (Pee):

<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/>

Shadow Maps:

<http://learnopengl.com/#!Advanced-Lighting/Shadows/Shadow-Mapping>

Water Effect:

https://www.youtube.com/watch?v=HusvGeEDU_U (6 Part Tutorial)

For general implemenations and help we used <http://learnopengl.com> because we really liked the tutorials and explanations.

Tools for modeling

We used Blender to create our objects.

How to play the game!

You move around by using the keys (W, S, D, A), you pee by pressing space.

W = forward

S = backward

D = right

A = left

Space = pee

You can rotate the camera with your mouse.

When you rotate the camera the pig doesn't stay attached to the camera. This is totally a feature. Just kidding, this problem will be solved asap.

Pee on all the people. Once you've achieved to piss off all persons, you won the game! YAY! Congratulations!

F-Keys

F1 - Help

No help is available.

F2 - Frame Time on/off

Go ahead and open your debug-tool ;) This actually used to work.

F3 - Wire Frame on/off

Working.

F4 - Textur-Sampling-Quality: Nearest Neighbor/Bilinear

Not implemented yet.

F5 - Mip Mapping-Quality: Off/Nearest Neighbor/Linear

Not implemented yet.

F6 - Depth map for shadows

See depth map of depth framebuffer

F7 - Water Reflection/Refraction for water effects

See water reflection and refraction view

F8 - View-Frustum-Culling on/off

Used to work - will be fixed.

F9 - Transparency on/off

Not implemented