

Temples of Doom

The goal of this game is to find the secret crystal and fight against the monsters.

You start in a dungeon without any weapon. Step forward to collect the sword. Use SPACE to jump. Use WASD to move and the left mouse button to hit an enemy. The right mouse button blocks incoming attacks.

The secret crystal is hidden in one of the rooms, but nobody knows which one ;)

Gameplay

We fulfilled the 3D aspect by including jumping passages over platforms. This room can be found when you go straight through the first big rooms and then left.

Effects

We implemented:

Shadow-Maps + Omni Directional Shadow-Maps (2)

Normal Maps (1)

Environmental Mapping (1)

Complex Objects

Most of our objects are loaded with a model loader. Enemies, Shield and Sword are complex objects for example.

Animated Objects

We included hierarchical animations, for example the arms of the monster swing when it tries to hit you.

View-Frustum-Culling

We created a simple culling for complex objects such as the enemy. The enemy is not drawn if it is out of the viewing distance.

Features of the game

Our game features great battles against the dungeon monsters.

We have implemented a simple enemy KI, where monsters attack you if you are close to them.

Also you have survive the Canyon of Doom, where you have to jump over wood planks to avoid falling to death.

Illumination and textures

Most of our objects have a texture and a normal map. The scene is illuminated by 8 independent point light sources that change their illumination over time randomly. The light sources are the blue crystals that can be found within the level.

Effects

Shadow-Maps + Omni Directional Shadow-Maps (2)

Enemies the Level and the Crystals cast shadows. Shadows can be seen in almost any room with a light source.

Normal Maps (1)

We have used normalmaps for the walls, the enemies and crystals.

Environmental Mapping (1)

The final crystal, which is located after the jumping sequence, is drawn using a cube-map shader.

The player can activate the fly-through modus for the current game by pressing "M" on the Keyboard.

Environmental Mapping was implemented with help of slides and documents of "Einführung in die Computergraphic".

Tools

We created our models by exporting them with the .obj export of Blender. Most of our objects have been created by ourselves (Shield, Sword, Level...).

To create normal maps, we used the NVidia Photoshop tools library.

Libraries

We used the bullet collision engine for our physics system.

Sources

<http://www.opengl-tutorial.org/>

<http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>

<http://www.opengl.org/sdk/docs/man/>

<http://stackoverflow.com/questions>

<http://ogldev.atspace.co.uk/>