God of Arrows - Submission 2

//For the game event submission we changed the camera to follow the arrow and fixed the contours of the objects

For this task we added more objects and set a clear win/lose condition. We also added a God Power of making the arrow invulnerable. If you find yourself stuck at the door just try hitting spacebar before it ;). You can use this power twice.

For the second task we focused more on the effects part of the game. We encountered some problems with the Visual Studio but we think we managed to surpass/solve them. Apart from effects we also make it feel more like a game with a bit of a challenge.

The Game upon execution has the camera following an Arrow until the end of the tunnel with constant speed and the player can only move the arrow up, down, left and right. The controls being the arrow keys, those moving both the camera and the arrow. We made this in the classes camera.cpp (ProcessKeyboard) and gameobject.cpp (KeyboardProcess) and listening to the keypresses in main.cpp.

The light source is a point light in the origin defined in the Gamecontainer and we made a cel shading effect in the sm.frag file. We use two classes for that (Model and Mesh). The Model class reads the nodes from the model file and saves them as Mesh Objects. We created some models in blender while some others we got from model sites. We loaded textures with SOIL and pass them to the shaders ourselves.

playable - Arrow moving towards the target

3d geometry - Through assimp loaded; .obj files

intuitive controls - Arrow keys, space bar

intuitive camera - Movement fixed with the arrow

moving objects - Arrow moves and rotates while doing so

Adjustable parameters - through the settings.ini file in the resources folder (you can also adjust the speed of the arrow there, default is 14.0)

Documentation

Win/Lose Condition - The winning condition is met when you get at the end of the tunnel and hitting the target. You lose if you hit an obstacle on the way or miss the target.
s
Cel shading - ~~Made in sm.frag with simple rounding~~. Implemented the cel shading on a color level to better suit our textures (RGB -> HSV -> RGB)

Contours via back faces - we outline certain objects by drawing the back faces bigger and in a dark color

Hierarchical animation - The feathers of the arrow simulate wind movement by shaking.

CPU particle system - When the God power is used particles come out of the arrow, particles are drawn using instancing

Shadow maps with PCF - implemented depth map, shadow acne removal through bias, PCF and peter panning mitigation

For the effects we got all of the information from either learnopengl, stackoverflow or stackexchange from various authors and articles.

Resources

- **Learnopengl**
- **Stackoverflow**
- **Stackexchange**
- **Tuwel forums**
- **Turbosquid**
- **Cgtrade**