

Documentation

Group Name / Game Name: Quang's Quest

GitHub Link: <https://github.com/juliahah/-cgue20-quangsquest.git>

Students: Chiara Kanya (11709458), Julia Hahn (11776189)

Genre: Brain Game, Fantasy

The main goal of this game is to find the colored lanterns, which are placed all over the playing field, in the correct order. The correct order is displayed at the beginning of the game.

Description of the implementation

In our game, we use non-trivial **3D objects** (obj & fbx) like a temple, a bridge, trees and flowers. To load these objects, we are using the assimp library, which is linked below. Most of them are downloaded from the internet except our character, which we created in Blender. Additionally, we have a **Win/Lose condition**. You win, if you find the lanterns in the correct order in the given time, otherwise you loose.

For the **illumination model**, we are using the lanterns and a moon as a light source: all objects are **textured** and have normals. Furthermore we also implemented **moving objects** like our character and the lanterns. The **parameters** like the screen resolution, fullscreen-mode and the refresh-rate can be adjusted in the "settings.ini" file.

Our game runs with **60 frames per second** and is **framerate independent**.

Ad player movement control:

- Press keys W or UP to move Player forwards. *
- Press keys S or DOWN to move Player backwards.*
- Press keys A or LEFT to rotate Player right around its own axis. *
- Press keys D or RIGHT to rotate Player left around its own axis. *
- Press key SPACE to let Player jump.

*The longer the keys are pressed the farther the movement.

Ad 3D person camera control:

The camera follows the players movements and is fixed on the player.

- Scroll with the mouse-wheel to zoom in and out. The zoom area is restricted to max 23 away from the player and min. 17.
- Press and hold the left mouse-button and move mouse left or right to rotate camera around the player.
- Press and hold the right mouse-button and move mouse up or down to change the pitch angle of the camera to the player. The pitch angle is restricted to max 45 and min. 5 radians.
- Press key R to set the camera to default pitch, zoom and angle around the player.

Features of the game:

- **HUD:** 2D overlay at the beginning of the game, as well as a status of the collected lanterns and the time. We also display if you won/lost with an HUD. The following Links were used:
https://www.youtube.com/watch?v=n4k7ANAFsIQ&list=PLlrATfBNZ98foTJPJ_Ev03o2oq3-GGOS2&index=17
<https://www.youtube.com/watch?v=vOmJ1lyiJ4A&t=371s>
<https://www.youtube.com/watch?v=SMYi87VJRJM>
<https://www.youtube.com/watch?v=i1mp4zflkYo>
- A **sound** was added with the irrklang library (see below)
- **Skybox:** To give the game a nice design, we added a skybox using a Cubemap
<https://learnopengl.com/Advanced-OpenGL/Cubemaps>
- **Timer:** The Game is time-limited. We display the time as a text-overlay using the FreeType library (see below).
<https://learnopengl.com/In-Practice/Text-Rendering>
<https://learnopengl.com/In-Practice/2D-Game/Render-text>
- **Collision Detection:** self implemented

Effects:

- **CPU Particle System** surrounding and following the player
https://www.youtube.com/watch?v=6PkjU9LaDTQ&list=PLRIWtICgwaX0u7Rf9zkZHL_oLuZVfUksDP&index=34
<https://www.youtube.com/watch?v=GK0jHlv3e3w&t=33s>
<https://learnopengl.com/In-Practice/2D-Game/Particles>
- **Bloom** for lanterns and moon
https://tuwel.tuwien.ac.at/pluginfile.php/1721131/mod_page/content/37/Bloom_SS19.pdf
<https://learnopengl.com/Advanced-Lighting/Bloom>
<https://www.youtube.com/watch?v=LyoSSoYyfVU>
- **GPU Vertex Skinning** for our character
<http://oglddev.atSPACE.co.uk/www/tutorial38/tutorial38.html>
<http://www.xphere.me/2019/05/bones-animation-with-openglassimpglm/>
<https://www.gamedev.net/forums/topic/688121-skeletal-animation-assimp-glm-and-m-e-in-between33/>
https://www.youtube.com/watch?v=fDmMH8_WRok
- **Environment Mapping** on our river:
https://www.youtube.com/watch?v=QYvi1akO_Po
- **Physically Based Shading** on spheres/rocks in the river
<https://learnopengl.com/PBR/Theory>
<https://learnopengl.com/PBR/Lighting>

Used Libraries:

- Assimp for object-loader: <https://www.assimp.org/>
- STB_IMAGE https://github.com/nothings/stb/blob/master/stb_image.h to load images
- Sound <https://www.ambiera.com/irrklang/>
- FreeType <https://www.freetype.org/>
- SDL <https://www.libsdl.org/>

Additional Links/Tutorials:

Camera:

- <https://www.youtube.com/watch?v=PoxDDZmctnU&list=PLRIWtICgwaX0u7Rf9zkZhLoLuZVfUksDP&index=19>
- <https://learnopengl.com/Getting-started/Camera>

Player Movement:

- <https://www.youtube.com/watch?v=qsIBNLeSPUc&list=PLRIWtICgwaX0u7Rf9zkZhLoLuZVfUksDP&index=16>
- <https://www.youtube.com/watch?v=L2aiuDDFNik>

Including Assimp & Model-Loading:

- <https://learnopengl.com/Model-Loading/Assimp>
- https://www.youtube.com/watch?v=yQx_pMsYqzU&t=2976s
- https://www.youtube.com/watch?v=W_Ey_YPUjMk&t=332s

Blender Modelling - Texturing - Animation:

(all episodes of the following blender-series on how to create a character)

- https://www.youtube.com/watch?v=DiloWrOIIRw&list=PLFt_AvWsXI0fEx02iXR8uhDsVGhmM9Pse&index=1