

Documentation

Group/Game Name: Rinceworld

Brief description of implementation:

We made our own framework with a static lib (PTVC_2025) as the engine and an application (Rinceworld) where we designed the world. So Rinceworld would design the game in different layers and call to the engine for rendering, handling of the shaders, the layer stack, physics, etc.

Additional libraries:

- Assimp - <https://assimp.org/>
- Bullet - <https://pybullet.org/Bullet/BulletFull/index.html>
- FreeType - <https://freetype.org/>
- Glad - <https://glad.dav1d.de/>
- GLFW - <https://www.glfw.org/>
- GLM - <https://www.opengl.org/sdk/libs/GLM/>
- spdLog – for debugging - <https://github.com/gabime/spdlog>
- STB image - <https://github.com/nothings/stb>

Gameplay:

Mandatory:

- 3D Geometry: Pillars and the Candle Stick are loaded FBX objects
- Playable: yes
- Advanced Gameplay:
 - Portals that teleport the player between them
 - Floating platforms that are only visible if you have found the LightCube
 - The need to light your way so you can find the finish line
- Min 60 FPS and Framerate Independence: FPS through Vsync of OpenGL and deltaTime for framerate independence
- Win/Lose Condition: finding the door out of the puzzle room / the time & light runs out
- Intuitive controls:
 - WASD – movement
 - Q – throw / summon Cube (after pickup)
 - Space - jump
 - Mouse – look around
 - LMB – shoot spell
 - Left Shift - Sprint
 - Esc – Pause Game
- Intuitive Camera: Perspective Camera Controlled by mouse, Debug cam if F5 pressed
- Illumination model:
 - Light Cube as only light source (PointLight)

- Each object as a material with different Maps associated with this texture
- If the Material has no normal map it takes the normal of the shape it is put on
- Textures: every object has a texture attached either onto the UVs of a simple mesh or onto the one of the complex objects
- Moving Objects: The LightCube can be thrown and moves with the player character and the spell you can shoot is a moving particle
- Documentation:
- Adjustable Parameters: The parameters are first set fixed and can be adjusted in game either by clicking the Fullscreen button in the menu or by resizing the window

Optional:

- Collision Detection (Basic Physics): Bullet so the player character cannot move out of the level and doesn't fall through the floor
- Advanced Physics: the spell projectile changes the tele walls to an active portal that the player can go through
- Heads-up Display: the HUD can be toggled with F2 and consists of a crosshair, a timer and the selected Spell

Effects:

Lighting:

- Omnidirectional Shadow Map: LightCube as point Light that the player can move <https://learnopengl.com/Advanced-Lighting/Shadows/Point-Shadows>

Texturing:

- Specular Map: switch with F1 the shader to a phong specular shader to see the specular map <https://learnopengl.com/Lighting/Lighting-maps>

Shading:

- Physically Based Shading: is the startup shader, the metallic object is the candleholder, and a nonmetallic one is the stone floor texture <https://learnopengl.com/PBR/Lighting>

Post Processing:

- Bloom/Glow: <https://learnopengl.com/Advanced-Lighting/Bloom>

Other special features:

- Pause Menu by pressing Esc that pauses the Game
- Object loader

Walk-through:

1. Shoot the first portal, LMB, on the dirt behind the lightCube
2. The second one on the dirt in the middle
3. Go through the portal in the middle to get the lightCube
4. Pick up the lightCube, can be thrown with Q
5. Jump down and go behind the wall in the middle
6. If you bring your lightCube, you can see the platforms that lead onwards
7. Shoot a portal on the dirt behind on top of the pillar by the light sensitive platforms
8. The other one in the middle and go through
9. Do some platforming, sprinting (Shift) might help for the last jump
10. Finish the game by walking through the door