

Documentation

Group Name/Game Name	Ghost
GitHub Link	https://github.com/Gismo30/cgue20-Ghost.git
Students	Lisa Fürst, 11775842 Lukas Kloibhofer, 11776882
Genre	Room Escape/Puzzle Game
Goal	Escape the cave before the ghost catches you or your lamp oil runs out.
Story	You are the ruler of a kingdom in the Middle Ages and your enemies have breached your castle's walls. Only equipped with an oil lamp, you have to flee through a secret underground tunnel to escape the bloodthirsty soldiers. To make matters worse, the ghost of a former prisoner, who is not happy to see you, starts to hunt you through the tunnels.
Gameplay	The main character is able to move freely. As soon as the game starts, a "ghost" starts to follow you around and tries to catch you. If the ghost gets too close to the player the game is lost. As time goes on, the player will lose lamp oil and the game is lost if there is no lamp oil left. The game is won if the player finds the exit before getting caught by the ghost and before the time running out.
User interaction	<ul style="list-style-type: none"> • WASD: Movement of player • Mouse: Camera view direction • ESC: Quit • F1: Toggle wire-frame mode (shows only quads due to post-processing) • F2: Toggle back-face culling • F3: Toggle free camera • F8: Toggle View-Frustum Culling • M: Toggle HUD • N: Toggle normal mapping • E: Interact • Space: Jump
3D Objects	Static: <ul style="list-style-type: none"> • Single-level cave Dynamic: <ul style="list-style-type: none"> • Hand-held lamp • Ghost • Guillotine, door, lever/switch (animated)
Configurable (settings.ini)	<ul style="list-style-type: none"> • Window width, height • Refresh rate • Fullscreen true/false • Window title • Camera: FOV, near plane, far plane • Illumination: ambient light coefficient

Features

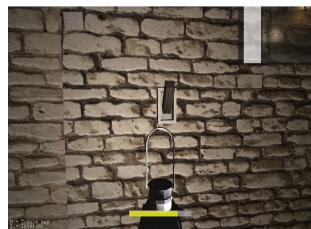
Category	Feature	Description	Tutorial(s)/resource(s) used
Gameplay	3D Geometry	Assimp used to load external .obj Models, used stb for Texture loading: Level, lamp, torches, ghost,... All objects except the level itself, the door handle and the switch were downloaded for free from different websites.	https://learnopengl.com/Model-Loading/Assimp , https://github.com/nothings/stb , https://www.assimp.org/ , https://free3d.com/ , https://sketchfab.com
	Playable, Advanced gameplay	To end the game, the player must move through the level to find the exit. To win, the player must first disable a guillotine, find a key and open a door.	
	Min. 60 FPS and frame rate Independence	Delta time used for calculations, PhysX frame rate capped. Frame rate on NVIDIA GTX 1660 Ti: 300+ FPS	
	Win/lose condition	If the player finds the exit, a success message is displayed. If the ghost comes too close, the game is lost. If the lamp oil runs out (remaining time/lamp oil shown on HUD), the game is lost. If the player steps under the guillotine before disabling it, the game is lost.	
	Intuitive controls	WASD for movement, E for interaction. M for toggling mini-map. Polling and callbacks used.	
	Intuitive camera	Choice between free camera (flying possible, debug purposes) and normal camera (toggleable via F3).	https://learnopengl.com/Getting-started/Camera
	Illumination Model	Multiple light sources: 3 torches, hand-held lamp. Models have configurable material coefficients. Normals are loaded via Assimp, or in case normal mapping is turned on, provided via normal maps. The illumination multiplier (ambient light) can be increased/decreased via editing the settings.ini.	Lighting provided via ECG-Framework, https://learnopengl.com/Model-Loading/Assimp , https://www.smart-page.net/smartnormal/
	Textures	Textures and corresponding UVs are imported via stb.	https://github.com/nothings/stb

	Moving objects	Lamp moves together with camera. The ghost follows the camera.	
Optional Gameplay	Collision Detection (Basic Physics)	Player may collide with walls in the tunnels and cannot fall through the floor. Walls and obstacles are static actors and the player is a kinematic actor. These collisions are detected.	https://gameworksdocs.nvidia.com/PhysX/4.1/documentation/physxguide/Manual/Index.html , https://github.com/NVIDIAGameWorks/PhysX
	Advanced Physics	Physics of the player: can fall down in the tunnel and jump. The guillotine is a dynamic object and can be triggered with a pressure plate by standing below it. Player hit by guillotine and pressure plate trigger is detected with SimulationEventCallback. The cannonballs are dynamic objects and can be moved.	https://gameworksdocs.nvidia.com/PhysX/4.1/documentation/physxguide/Manual/Index.html , https://github.com/NVIDIAGameWorks/PhysX
	Heads-Up Display	Minimap which uses alpha blending and which shows part of the discovered tunnels, plus the status of the remaining lamp oil. Helpful commands shown. Minimap is rendered to an additional framebuffer, using a second camera looking down onto the scene.	https://learnopengl.com/Advanced-OpenGL/Framebuffers , https://learnopengl.com/In-Practice/Text-Rendering (FreeType lib)
	View-Frustum Culling	Implemented with spheres as bounding boxes. Count of the currently rendered models is printed once per second in the console. Can be toggled with F8.	http://www.lighthouse3d.com/tutorials/view-frustum-culling/
Effects			
Lighting	Shadow Map with PCF	Lamp and torches throw shadows. Hand-held lamp's shadows are updated and recalculated each frame. For each point light, 6 cubemap faces in different directions are rendered to, to provide the possibility to have shadows in multiple directions (point shadows). PCF used to reduce jagged edges, bias applied to reduce unwanted artifacts.	https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping , https://learnopengl.com/Advanced-Lighting/Shadows/Point-Shadows
Animation	Hierarchical Animation	The door model matrix is used by the door handle to calculate its position in the animation (The guillotine and lever are animated as	https://tuwel.tuwien.ac.at/pluginfile.php/1721131/module/content/37/Animation_SS18.pdf

		well, but not hierarchical animations.)	
Shading	Simple Normal Mapping	Stone walls and floor have additional normal maps which are used for normal mapping. Normal mapping can be toggled via N key. A TBN matrix is calculated using tangents and bitangents passed to a shader via Assimp, which is then used for calculations in case normal mapping is turned on.	https://learnopengl.com/Advanced-Lighting/Normal-Mapping , https://www.smart-page.net/smartnormal/
Post processing	Bloom/Glow	The ghost is glowing white in the dark. Ghost is blurred in a loop (6 times). The blurred image of the ghost is combined with the “normal” scene and rendered on the screen via a separate shader onto a single quad.	https://learnopengl.com/Advanced-Lighting/Bloom
Advanced Modelling	CPU Particle System	Torches emit black particles with random size (capped so they don't get too large) and direction. The particles' origins are positioned at the corresponding point light positions. For rendering particles, separate shaders are used.	https://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-in-stancing/ , https://github.com/andersonfreitas/opengl-tutorial-org/blob/master/tutorial18_billboards_and_particles/tutorial18_particles.cpp
Other	Text	Freetype library used to display text (HUD, win/lose messages).	https://learnopengl.com/In-Practice/Text-Rendering , https://www.freetype.org/

Notable Gameplay

The player spawns in a dungeon that can be explored freely. In the dungeon is a locked door that leads to the exit. A key can be found in another room, but the entrance is blocked by a guillotine that kills the player if he stands underneath it. The player can jump over a obstacle by pressing space. In the same room as the door, a switch/lever can be found to the left that deactivates the guillotine. Now, the player can grab the key by pressing E, open the door and escape the dungeon.



Lever/switch which needs to be activated to be able to win game