MATTERialist

Computergrafik UE 2020S

Laurenz Edmund Fiala (11807869)
Lena Dolinek (11808609)

Submission 2

Effects

Shadow Map with PCF

Status	✔ Working
Details	Supports a single directional light. For smoother edges, a Poisson disc for multisampling (16x) is used.
Sources	https://github.com/opengl-tutorials/ogl/blob/master/tutorial16_shadowmaps/ShadowMapping.fragmentshader (15.4.2020) https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping (7.4.2020) http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/ (15.4.2020)

GPU Particle System using Compute Shader

Status	✔ Working
Details	Supports particle emitters as well as particles. TTL, manual removal of emitters, velocity, spawning of n particles per second per particle, emitting velocity, emitting TTL.
Where to find	The beam of an activated projector.
Sources	https://tuwel.tuwien.ac.at/pluginfile.php/1721131/mod_page/content/37/GPU_Particles_SS18.pdf (12.6.2020)

Hierarchical Animation

Status	✘ Not used
Details	Class “Animator” handles animation of floats using their pointers. Bezier curves (easing) and repetition are supported. Callbacks after completion can also be used. Since we use a recursive tree-like structure to render the scene, animations from a parent geometry is also applied to its children. **Update:** Feedback clarified that we don’t in fact fulfill the requirements. We did not try to add this, since our physics system is not equipped to handle hierarchical animations (even though rendering is).
Where to find	The moving platforms. (Only the main geometry is actually animated, but apply to its children as well.)
Sources	https://stackoverflow.com/a/21642962/2740014 (5.6.2020) used for calculating Bezier curves.

Specular Map

Status	✔ Working
Details	Specular maps can be passed to the TextureMaterial and are used to modify the specular intensity at a certain point on the texture.

Where to find	On the tiled floor and the ice. We intentionally used a high Ks so it should be easy to see.
Sources	https://learnopengl.com/Lighting/Lighting-maps (14.6.2020)

Simple Normal Mapping

Status	✔ Working
Details	We encountered a problem with our normal maps, where the generated normal were rotated by 90° along the Y-axis. Therefore we decided to disable this feature so as to not influence the specular maps. **Update:** We found the issue: objects without a normal map were assigned a wrong placeholder-color (0,0,1 instead of 0.5,0.5,1).
Where to find	On the floor texture and the ice.
Sources	http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-13-normal-mapping/ (14.6.2020) https://learnopengl.com/Advanced-Lighting/Normal-Mapping (14.6.2020) https://learnopengl.com/Getting-started/Textures (14.6.2020) https://learnopengl.com/Getting-started/Camera (14.6.2020)

BSP Tree

Status	✕ Not implemented
Details	This feature was not implemented due to time-restraints.

Additional systems

LevelImporter: in combination with the custom python script for exporting .obj and .mtl files, this class enables creation of Levels in Blender. The custom script is a heavily modified version of Blender's default .obj-exporter and exports local model matrices, some custom properties and the "Specular" shader (which in turn has support for diffuse, specular and normal textures).

DeferredExecutor: Simple class which executes a std::function (usually a lambda) after a given amount of time.

Gameplay

3D Geometry

We added projectors like in our proposal, but due to time-constraints no energy cells were added to re-gain lives. The geometry is loaded from blender (using aforementioned LevelImporter).

Gameplay

The player can walk around the level (W/A/S/D). It is also possible to jump (SPACE), toggle crouch (LCTRL) and pick up/put down a projector (E). Use Q to (de-)activate a projector. R can be used to respawn manually if you get stuck crouching beneath an object.

When crouching, your movement is disabled. You can only move using your previous momentum. You can't place projectors.

If you fall from any platform, you will die and automatically respawn after 3 seconds. You can speed this up by pressing R.

A projector can be used to change the surface of an object into ice or change gravity. Point the projector towards any surface and put it down (E) to see the change. We have two types of projectors in-game: ice and low-gravity. These two may be used to finish the level.

Gravity-projected surfaces apply their low-gravity effect in all axis-parallel directions (global: up, down, left, right, top, bottom).

Min. 60 FPS and Framerate Independence

The game is completely framerate independent. This was verified on two PCs, both in Debug and Release Mode (which vary drastically in performance). The 60 FPS-mark was reached on our computers, though not tested on the Vislab-PCs.

Win/Loose Condition

You start out with 3 energy cells. After falling to your death for the third time, you die and don't respawn. If you have lives left and want to skip the 3 seconds until respawn, press R.

Winning is displayed with a green overlay at the end of the level (you have to reach the elevated platform). When you have won, you can also respawn (R-key) and will have your initial 3 lives again.

Intuitive Controls

The player can walk around the level (W/A/S/D). It is also possible to jump (SPACE), toggle crouch (LCTRL) and pick up/put down a projector (E).

Note: Crouching and projector-interactions are callbacks. Jumping was intentionally implemented using polling, so the player can jump continuously.

Function	Key	Info
Walk forward	W	
Strafe left	A	
Walk backwards	S	
Strafe right	D	
Jump	SPACE	Movement in the air is only applied partially. Hold to jump continuously.
Toggle crouch	LEFT CTRL	Movement is disabled.
Respawn	R	Only works when alive or won. Subtracts one life. When you win, your lives are reset.
Pick up/put down projector	E	Walk closely to the projector, point at it and press E.
(De-)Activate projector	Q	Walk closely to the projector, point at it and press Q.
Function	**Key**	**Info**
Quit the game	ESC	
Wireframe	F1	Toggle wireframe mode.
Backface culling	F2	Toggle backface culling.
HUD	F3	Toggle the HUD.
Disable cursor	O	Hide the cursor.
Enable cursor	P	Show the cursor.

Intuitive Camera

The camera is fixed to the player controller and cannot be moved freely.

Illumination Model

We have at least one directional light source and normal vectors are calculated by Blender. Smooth shading can be seen on the projectors.

Textures
We use the ECG Texture class and do not know how it handles mipmapping and filtering, but we think it is done like it is supposed to be done here. We have objects with textures.

Moving Objects
There are moving platforms in the level.

Documentation
You can find additional documentation in the code. The Header-files contain lots of additional technical documentation.

Adjustable Parameters
Most of this was already provided by the ECG framework. We added a modifier for brightness (1.0 is the default, 2.0 is twice as bright, etc.) to the .ini-file. It is only applied to the 3D geometry in the shadowmap.frag shader (not the HUD & particles).

Additionally, since `GLFW_REFRESH_RATE` did not work for us (neither windowed, nor fullscreen-mode), we added a framerate-limiter.

Collision Detection (Basic Physics)
The player and the environment can collide. If the player is located beneath a geometry, it can't stand up from a crouched position.

For the projector-targets we cast a ray to check which object intersects and then change that object's properties (material, physics properties). For this, a PhysX raycast is used. This is also used to check if the player is currently on the ground (which is necessary to know whether they may jump or not). Additionally, we use shape hit callbacks to enable trigger volume, which is used for the game-won-trigger.

Sources: https://documentation.help/NVIDIA-PhysX-SDK-Guide/index.html (14.6.2020)
https://gameworksdocs.nvidia.com/PhysX/4.1/documentation/physxguide/Index.html (14.6.2020)
https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/apireference/files/main.html (14.6.2020)
https://github.com/NVIDIAGameWorks/PhysX/blob/4.1/physx/samples/samplenorthpole/SampleNorthPoleCCT.cpp (15.6.2020)
https://docs.nvidia.com/gameworks/content/gameworkslibrary/physx/guide/Manual/CharacterControllers.html (15.6.2020)

Advanced Physics
Update: We added (physics-enabled) balls to kick around. You can only win if you nudge the ball and cube into their baskets.

Heads-Up Display
We also added a HUD for the player. It shows death, win, current amount of lives and a crosshair for easier placement of projectors.
The HUD can be toggled using F3.

(Very) Brief Tutorial
Walk towards first blue projector → look at it while standing very close, press E → Projector disappears → walk upwards and crouch before the obstacle to slide underneath → jump across → point projector at surface below obstacle → projector is put down → slide underneath again when platform is opposite of you → pick up purple projector using E → point at platform in the sky → walk across the pit → use 2nd blue projector to retrieve purple projector: slide under the obstacle again (2x) → point purple projector at floor in front of wall → jump while above purple texture → walk forwards onto the platform until the screen turns green.
Update: on the middle platform you now have to nudge the ball into the basket (use ice projector to make this easier). On the last part, you must use the ice projector to slide the cube into the basket. Then proceed towards the upper platform.