

Documentation

Group/Game Name: LaserJump

Brief description of implementation:

The game somewhat looks as on the initial poster and is playable. However, unfortunately, due to severe time constraints, creation of a complex game with attention to detail was impossible and we had to cancel some features. We instead focused on the aspects essential and especially relevant for this course.

We hope you see that given more time, we are able to apply the skills learned and needed in this course!

Additional libraries:

- Freetype library: <https://freetype.org/download.html> (Last accessed: 11.06.2024)
 - Freetype License: <https://freetype.org/license.html> (Last accessed: 11.06.2024)
 - The FTL:
<https://gitlab.freedesktop.org/freetype/freetype/-/blob/master/docs/FTL.TXT> (Last accessed: 11.06.2024) (this license/copyright-notice can also be found in the root directory of the project (where the solution is) under the name "LICENSE_ftl.txt")
- PhysX 4.1 from NVIDIA Corporation is being used for the Physics-Simulation of the game. (Character is being pulled toward the homeBuilding to "walk" on it, ...):
<https://github.com/NVIDIAGameWorks/PhysX> (Last accessed: 11.06.2024)
 - License:
<https://gameworksdocs.nvidia.com/PhysX/4.1/documentation/physxguide/Manual/License.html> (Last accessed: 11.06.2024) (this license/copyright-notice can also be found in the root directory of the project (where the solution is) under the name "LICENSE_physx.md")
- ASSIMP has been downloaded from here: <https://github.com/assimp/assimp/releases> (Last accessed: 11.06.2024)
 - License: <https://github.com/assimp/assimp/blob/master/LICENSE> (Last accessed: 11.06.2024) (this license/copyright-notice can also be found in the root directory of the project (where the solution is) under the name "LICENSE_assimp.txt")
- Stb_image is a public domain image loader downloaded from here:
https://raw.githubusercontent.com/nothings/stb/master/stb_image.h (Last accessed: 18.06.2024)
 - License: The license can be found at the end of the document at the mentioned link above, but this license/copyright-notice can also be found in the root directory of the project (where the solution is) under the name "LICENSE_stb_image.txt"

Gameplay:

Mandatory:

- **3D Geometry:** Check. Disclaimer: The stickman is a custom model loaded with Assimp and *not* generated via cubes etc.! Its “cubiness” was an aesthetic design choice (see poster) – loading another model (e.g. dancing_vampire.dae from the learnopengl.com tutorial) works just fine as well. (6 pts)
- **Playable:** Check. Basic gameplay mechanics are present and the game can be started via the .exe file. (3 pts)
- **Min 60 FPS and Framerate Independence:** Check. Feedback from Sub1 regarding framerate independence was incorporated. (3 pts)
- **Win/Lose Condition:** Check. Win condition: Survive long enough so that the building is lit/white. Lose condition: Touch a laser beam or fall down the building. (3 pts)
- **Intuitive controls:** Check. WASD+Space. Polling/callback implemented correctly. (2 pts)
- **Intuitive Camera:** Check. Alternative to arcball camera (more elegant). Player character facing and moving direction influenced by camera view (3rd person view). (2 pts)
- **Illumination model:** Check. One light source above the building top (only one because of the intended aesthetic, see poster) is present and each object has a material and normals. (2 pts)
- **Textures:** Check. The skybox has a texture and also the reflection on the building surface. (2 pts)
- **Moving Objects:** Check. The player character and the lasers all move. (2 pts)
- **Documentation:** Check. This document. (1 pt)
- **Adjustable Parameters:** Check. Screen res, full-screen mode and further parameters like FOV can be changed in .ini file (feedback from Sub1 was incorporated). (1 pt)

Optional:

- **Advanced Gameplay:** Laser logic divided into levels. (? pts)
- **Collision Detection (Basic Physics):** PhysX is used for collision between the player character and other non-laser objects. (6 pts)
- **Advanced Physics:** -
- **Scripting Language Integration:** -
- **View-Frustum Culling:** -
- **Heads-up Display:** ~~Some text is displayed using FreeType. (4 pts)~~

Effects:

Lighting:

- **Lightmap using Separate Textures:** -
- **Lightmap using In-Game Calculation:** -
- **Shadow Map with PCF:** -
- **Shadow Volumes:** -

Advanced Modelling:

- **CPU Particle System:** -

- **GPU Particle System using Transform Feedback:** -
- **GPU Particle System using Compute Shader:** -
- **Blobby Object Using Marching Cubes:** -
- **Subdivision Surface:** -

Terrain:

- **Tessellation from Height Map:** -
- **Voxel Terrain using an Octree:** -

Animation:

- **Hierarchical Animation:** -
- **Vertex Shader Animation:** -
- **GPU Vertex Skinning:** A custom stick figure model as per the poster from Sub0 (created with Blender) with bones is loaded, skinned and animated as a running figure. Implementation closely followed the tutorial on learnopengl.com, which however mostly had to be heavily adapted to our very different code structure and data structures.

<https://learnopengl.com/Guest-Articles/2020/Skeletal-Animation> (20 pts)

Texturing:

- **Procedural Texture:** -
- **Video Texture:** -
- **Specular Map:** -
- **Environment Map:** Environment Mapping was implemented on the cube surface. The sky and moon are reflected on the floor. Implementation loosely followed the learnopengl.com tutorial for cubemaps. None of us did this effect in the earlier course. <https://learnopengl.com/Advanced-OpenGL/Cubemaps> (8 pts)

Shading:

- **Simple Normal Mapping:** -
- **Cel Shading:** -
- **Style Transfer:** -
- **Brush Strokes:** -
- **Physically Based Shading:** -

Advanced Data Structures:

- **BSP Tree:** -
- **kd-Tree:** -
- **LOD using an octree:** -

Post Processing:

- **Bloom/Glow:** -
- **Lens Flares:** -
- **Contours via Backfaces:** -
- **Contours via Edge Detection:** -

Other special features: -

Problems:

In Release-mode, the usage of processor power is much more apparent, thus maybe resulting in a crash.

Walk-through: -

References / Credits:

- “assets/models/dancing_vampire.dae” and its “Vampire[...]” textures are part of the learnopengl.com Vertex Skinning tutorial.
- [UNUSED] “assets/fonts/bitfont.ttf” has been uploaded by the user named “Mushroom Font” here <https://www.dafont.com/bitfont.font> (Last accessed: 11.06.2024) and is, according to the author, free for personal and commercial uses. However this font is NOT being used in our project anymore.
- “assets/fonts/arial.ttf”. Arial is a font from Microsoft Corporation. Arial is present in multiple Windows-versions. Copyright (c) 2024, Microsoft Corporation. A copy of this notice can be found in the root directory of the project under the name “LICENSE_arial.txt”.