Object Inspection Project (Computer Graphics)

- SUDIKSHA SHARMA

Project Plan

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

TimeLine : 21.11.2022. - 31.3.2023. ~ 19 weeks

1. 21.11.2022 - 02.12.2022
   - Get acquainted with basics of UE
   - Setup C++ environment for UE

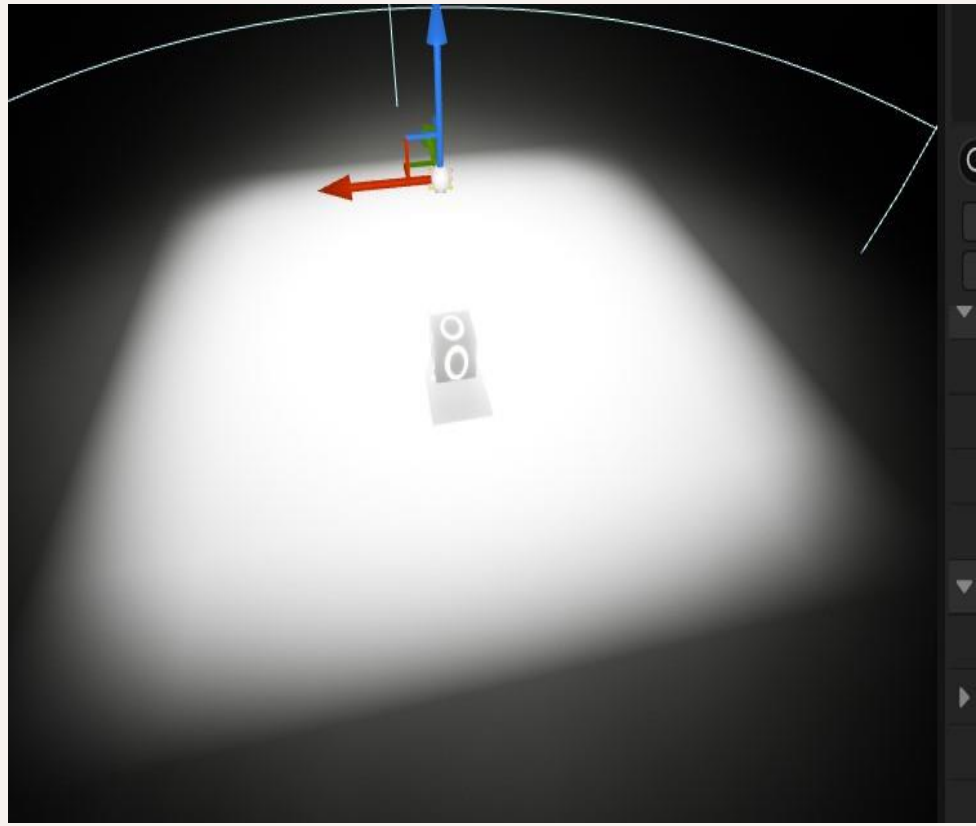
2. 02.12.2022 - 23.12.2022
   - Setup the Light source in the scene
   - Load the mesh and place it into the scene
   - Write a shader from OSL to UE material graph
   - Setup git repository of the Project

2.1 Setup Lightsource in the scene

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Tried 2 light setups to replica the lights of the inspection device :

1. Area Light
   - Light is emitted from surface of a shape. In UE, there's generally rect-light available for area light (light emitting from rectangular source).
   - Current Issues :
     - Need to extract out the torus shape from rectangle light to mimic the light source of inspection model which will take extra efforts. (As discussed with Juraj, masking the central pixels in form of torus using circle equations). Also, there's no prewritten configuration available in UE to do the same.
     - Emissive materials are quicker and give better visual results.

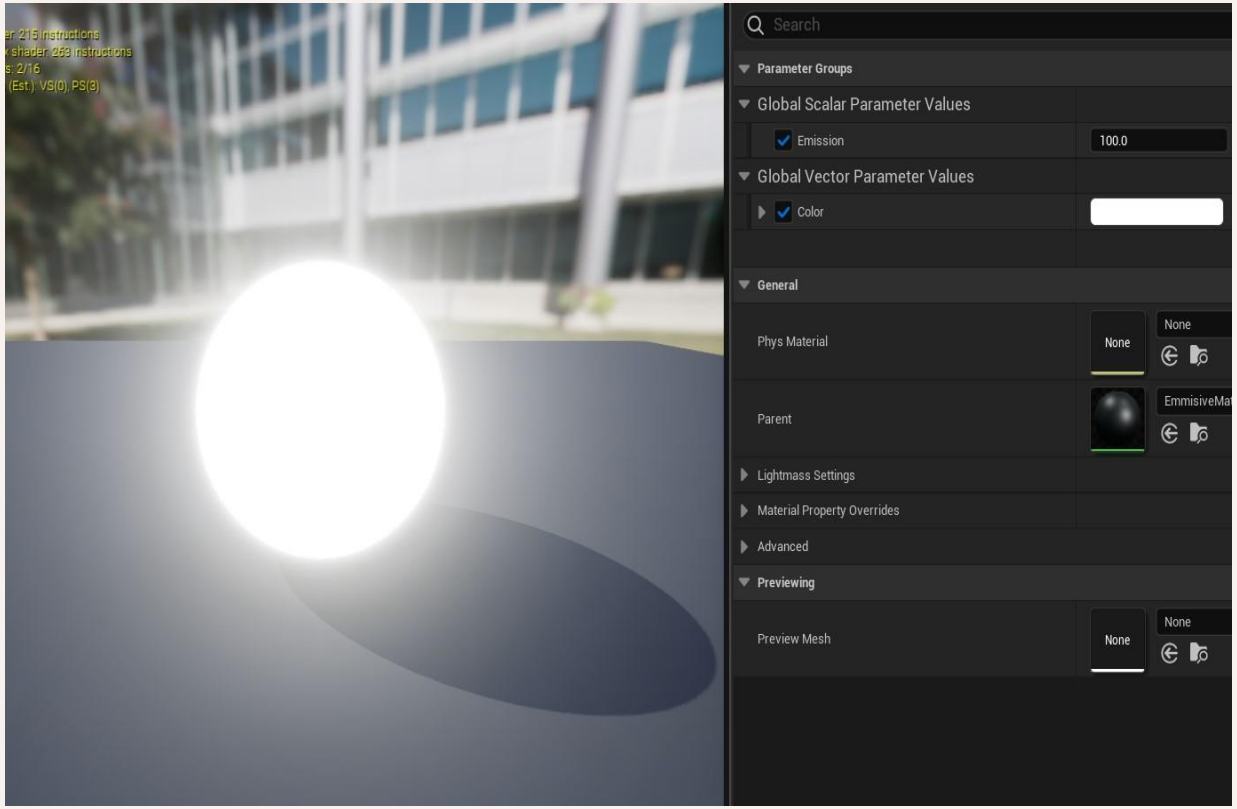


2.1 Setup Lightsource in the scene

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

2 Types of LightSources to mimic the scene :

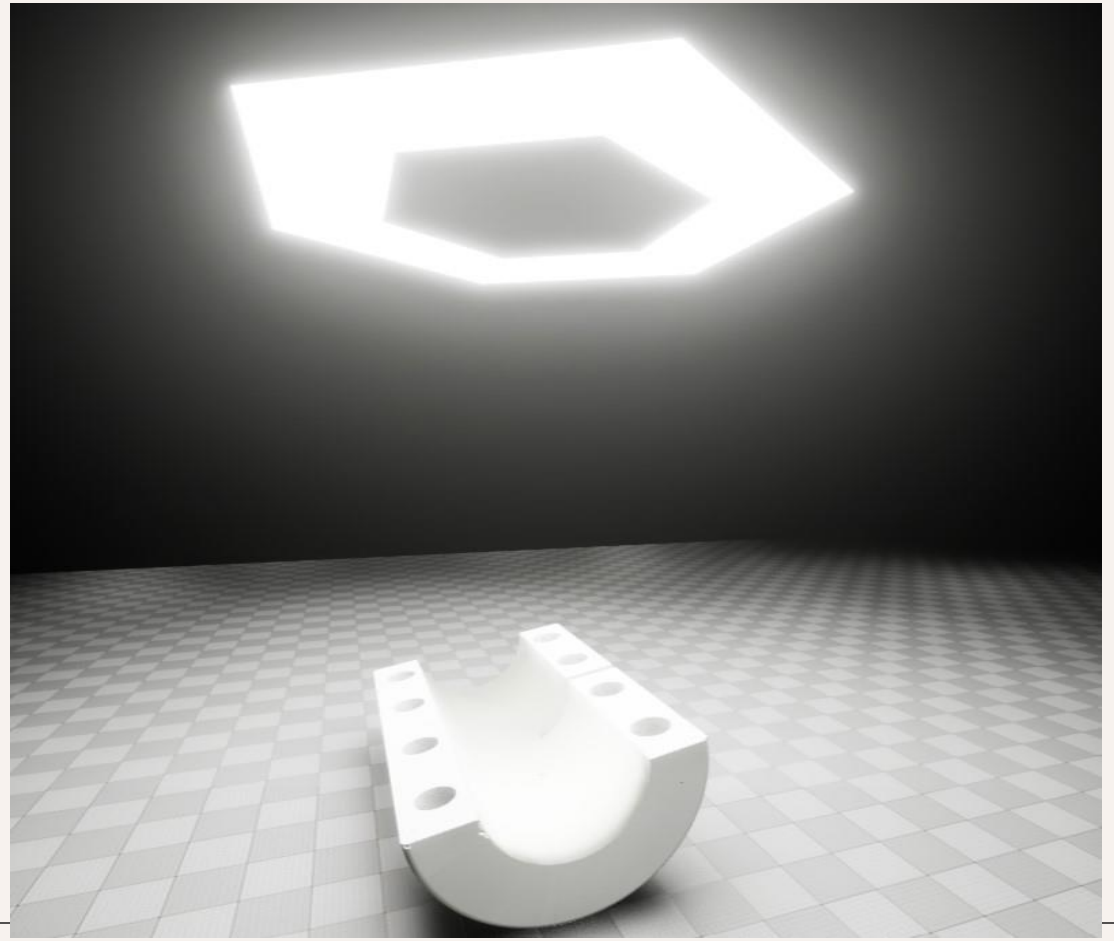
2. Emissive material
   - Self illuminating materials.
   - Created two 2d torus mesh in Blender with the measurements provided. (Circular Torus and Hexagonal Torus)
   - Created custom emissive material in UE, with emission value (emissive intensity) and rgb range, both defined by user.
   - Turned off eye adaptation (doesn't support by real camera) and other unnecessary properties which take up computational power.



2.1 Setup Lightsource in the scene

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

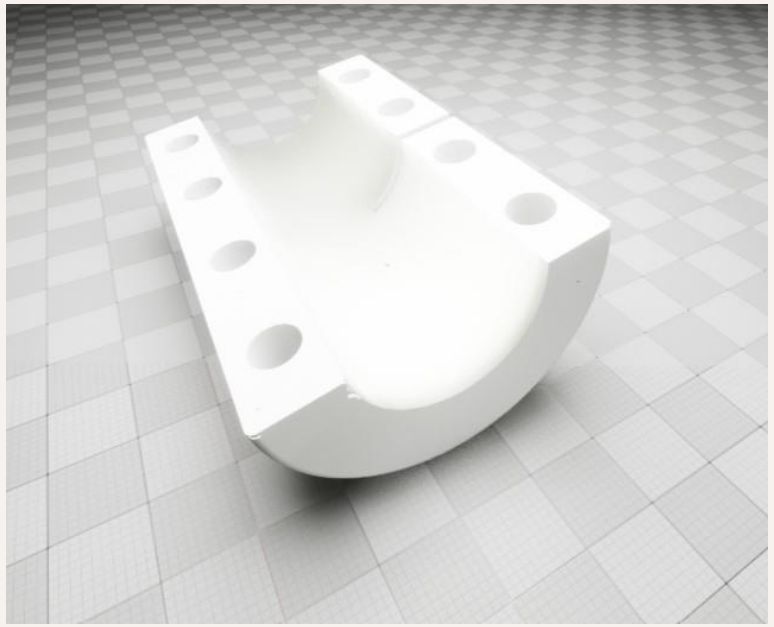
Scene With Emissive Material and object



2.2 Load the mesh and place it into the scene

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

- Since mesh was available in .obj file, this file format is NOT fully supported by UE.
- So, loaded the .obj file in Blender and exported it in .fbx format in parts and manually added materials to them in UE.



2.3 Write a shader from OSL to UE material graph

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

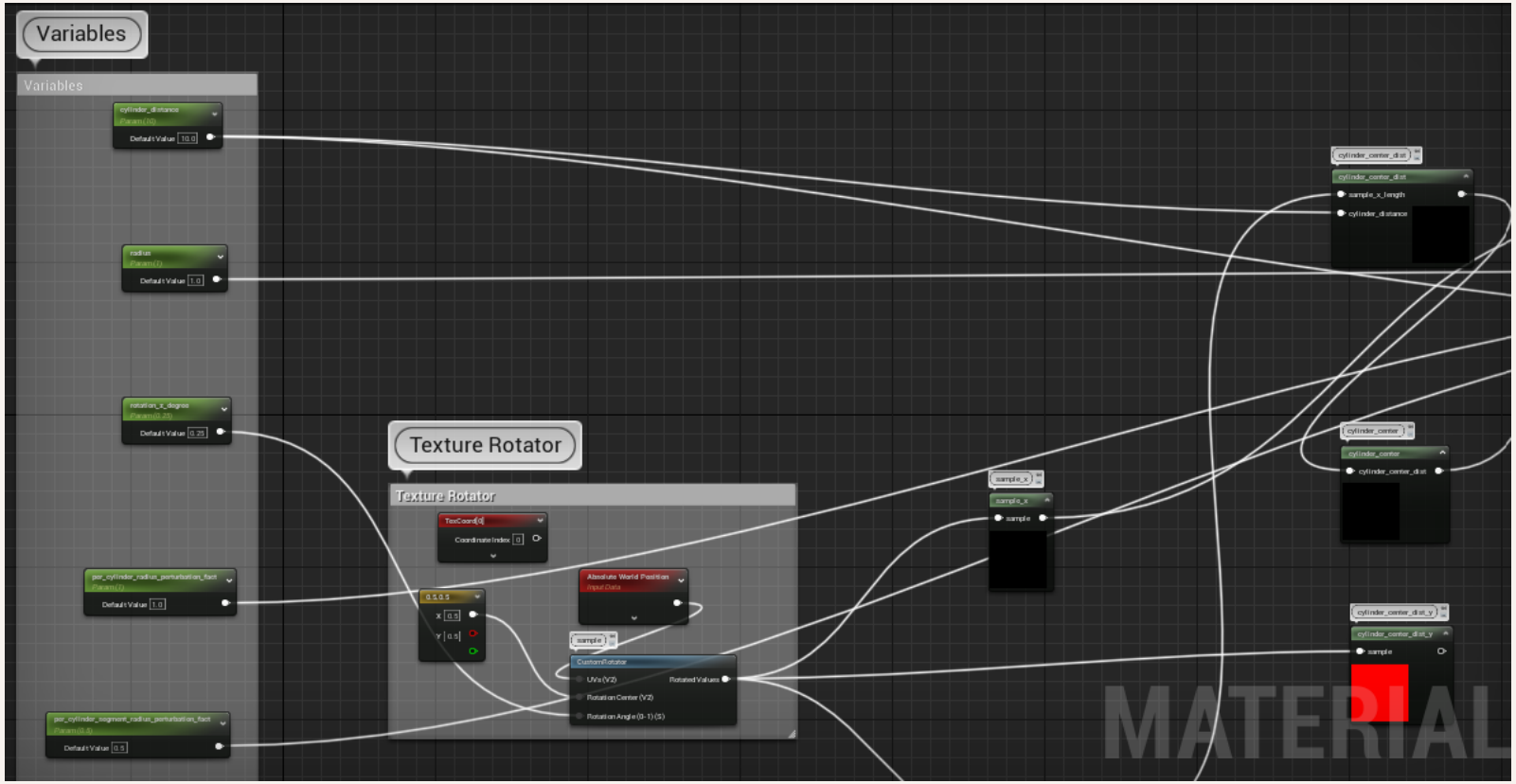
- **OSL (Open Shading Language)** is a programmable shading language used to describe lighting, materials and other shading properties.
- **UE Material Graphs** are the visual programming editor used to write shaders in UE in form of visual connected components forming a graph. It doesnot support OSL.

```
shader parallel_lines_cylinder_axis(
    point sample_in = P,
    float cylinder_distance = 1.0,
    float radius = 0.1,
    float rotation_z_degree = 0.0,
    float per_cylinder_radius_perturbation_fact = 0.0,
    float per_cylinder_segment_radius_perturbation_fact = 0.0,
    float surface_roughness = 0.2,
    float cylinder_roughness = 0.25,
    float normal_slope = 0.5,
    int axis = 0, // x - 0, y - 1, z - 1
    output float roughness = surface_roughness,
    output vector perturbed_normal = N)
{
    // Normal in shading point.
    vector in_normal = Ng;
    // Find optimal triplanar values based on normal.
    vector axis_normal;
    int cylinder_axis; // u, x // axis = radius
    int cylinder_axis2; // v, y
    if(axis == 0) //  x axis
    {
        // Take normal part oriented in x axis. Sign is important!
        axis_normal = normalize(vector(1, 0.0, 0.0));
        cylinder_axis = 1;
        cylinder_axis2 = 2;
    }
    else if(axis == 1) // y axis
    {
        // Take normal part oriented in y axis. Sign is important!
        axis_normal = normalize(vector(0.0, 1, 0.0));
        cylinder_axis = 0;
        cylinder_axis2 = 2;
    }
```

Sample OSL

2.3 Write a shader from OSL to UE material graph

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN



UE Material Graph sample of an OSL

2.3 Write a shader from OSL to UE material graph

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

- Understood some important OSL concepts.
- Understood the material editor in Unreal. Unlike OSL and other shader languages, it is visually scripted in the form of connected graphical components. Though, there is scope of writing code as well but graphs are mostly preferred and are quick to use.
- Tried to translate the 2 sample OSL shaders written by Lovro in UE material graphs.

2.3 Write a shader from OSL to UE material graph

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

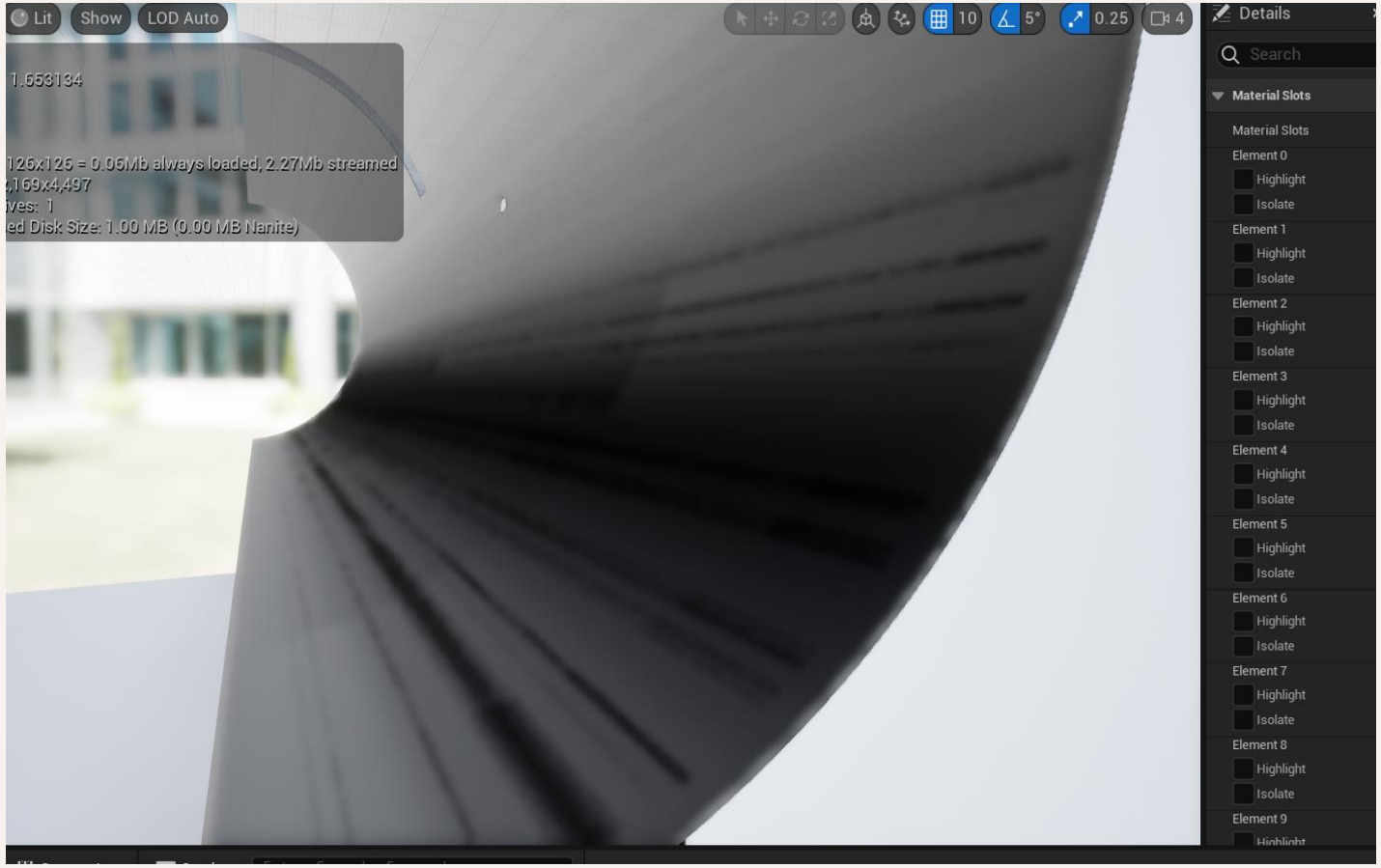
Problems Faced :

- In OSL, there are some in-built methods like rotate, cellnoise(to add noise). Unreal provides somewhat similar methods but not exactly the same, e.g. some parameters are different which may result in different output. So, the resulted rasterized image is different.
- Artifacts : Shaders needs hit and trial with the lighting, scene properties etc. because it is giving out many artifacts at the moment.
- Debugging :Strongly requires a way to debug the shader compilation, as there are large number of nodes, so its difficult to figure out which node is giving out wrong output and hence tearing the shader. Once we get it, then it will be relatively easy and quick to fix and implement all other shaders as well. Some small tricks can be used, as suggested by Juraj.

2.3 Write a shader from OSL to UE material graph

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

- Currently, as it is apparent in the image, the basic cylindrical texture that we want to achieve is there but its nowhere near to the renders from OSL. (No noise)
- There might be a need to tweak some parameters or change some code approach on importing to material graph.
- Also, there's might be a good scope of optimization in currently implemented material graph.



TODO : Read Lovro's paper on Texture Synthesis for Surface Inspection to get a better insight on the logic behind the texture generation.

2.4 Setup git repository of the Project

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

- Setup Git repository to track the and store the project files and ppts.

Github Link : https://github.com/sudikshasharma/object-inspection-project (Checkout branch 'dev')

Object Inspection Project (Computer Graphics)

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Thank You so much

QUESTIONS PLEASE ?