OpenGL 3D Scene
Project Documentation

Graphic Processing 2024-2025

STUDENT: TOCAN ROBERT-ALEXANDRU

Group:30432

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

1. Contents

2. Subject Specification

The project utilizes OpenGL to develop a visually rich and interactive 3D winter-themed scene. It showcases advanced computer graphics techniques and provides an immersive experience through the following features:

1. **Dynamic Environment Rendering**:
   - A detailed 3D environment simulating a snowy winter setting with a realistic ground, lake, and surrounding low-poly trees.
   - A seamless **skybox** is employed to create the illusion of a winter setting, enhancing immersion.
2. **Advanced Lighting and Shadow Effects**:
   - Implementation of **directional and point lighting** to illuminate the scene dynamically.
   - Shadow computation using **shadow mapping** provides depth and realism by dynamically casting shadows from light sources onto the terrain and objects.
3. **Special Effects**:
   - **Fog Effect**: Creates a cold, misty atmosphere, activated or deactivated by the user.
   - **Rain Simulation**: A particle-based rain effect adds a dynamic layer to the environment, with raindrops interacting with the scene lighting.
4. **Object Animations**:
   - **Fish Animation**: The project incorporates an elliptical motion algorithm with sinusoidal vertical oscillation, simulating a fish swimming in a lake. This combines mathematical precision with visual fluidity.
5. **Camera Navigation**:
   - Offers both **manual and automated camera control**. The manual mode allows the user to explore the scene freely, while the automated tour takes the user along predefined or circular paths, showcasing the scene.

6. **Rendering Options**:
    - The user can toggle between **solid, wireframe, and point rendering modes**, enabling visualization of the scene in multiple styles.
7. **Interactive User Experience**:
    - The application allows real-time toggling of effects such as rain, fog, and light sources, as well as modifying camera paths, creating a highly customizable user experience.
8. **Integration with Blender for Scene Formatting**:
    - **Modeling**: Blender was used to design and format various 3D models used in the scene, such as trees, the lake, fish, and other environmental elements. These models were exported in the .obj format for seamless integration with OpenGL.
    - **Texture Mapping**: Blender's UV mapping tools were utilized to map textures to objects like the lake, ground, and trees, for visual consistency.
    - **Scene Composition**: The initial layout of the scene, including positioning and scaling of objects, was prototyped in Blender.

This project integrates multiple OpenGL features, including shaders for rendering and effects, object modeling, texture mapping, and matrix transformations for object positioning. Blender played a pivotal role in designing the assets and ensuring that the models and textures were optimized for OpenGL rendering. The whole serves as a complex example of using OpenGL and Blender for advanced graphical applications.

3. Scenario

3.1 Scene and Objects Description

The project features a winter-themed scene with the following objects:

- **Ground/Terrain**: A snowy ground is textured with fine details to simulate uneven terrain, creating a sense of realism.
- **Lake**: A dynamic lake with animated wave-like effects to simulate water movement.

- **Spring**: The starting point of the lake, composed of 3 different boulders and a dead plant for immersion.
- **House**: A wooden house that fits the scenery
- **Fish**: An animated fish moving in an elliptical path with sinusoidal vertical motion for realism.
- **Trees and Scenery**: Low-poly trees surround the lake, providing an immersive environment.
- **Dead trees**: are scattered throughout the landscape, adding to the atmosphere of a harsh, cold environment and creating variety in the scene.
- **Boulders**: Snow-covered boulders and normal boulders are placed sporadically across the landscape to create depth and natural diversity. Their rough textures contrast with the smoothness of the lake .
- **Skybox**: A cubemap skybox that gives the illusion of an expansive winter sky.

3.2 Functionalities

Key functionalities implemented include:

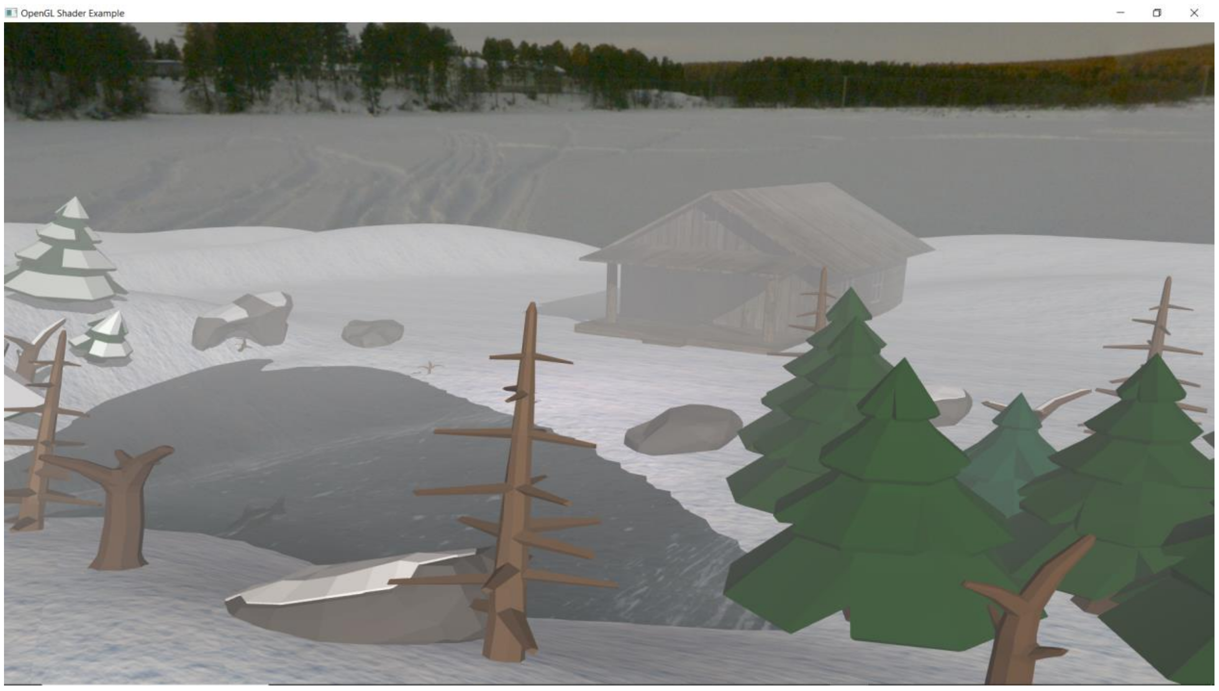
1. **Camera Movement**:
   - Keyboard controls (W, A, S, D, Z, X) for translation.
   - Mouse-based pitch and yaw adjustments for rotation.
2. **Rendering Modes**:
   - Solid, wireframe, and point views toggle with keys 1, 2, 3.
3. **Lighting**:
   - Directional light (default)
   - Point light(deafult) and toggleable point lights using the Keypad K(5 point lights colored yellow, blue, red, dark and light green).
   - Adjustable light rotation using keys U (clockwise) and J (counterclockwise).

4. **Fog and Rain Effects**:
    - Fog activation with F.
    - Rain simulation toggle with R.

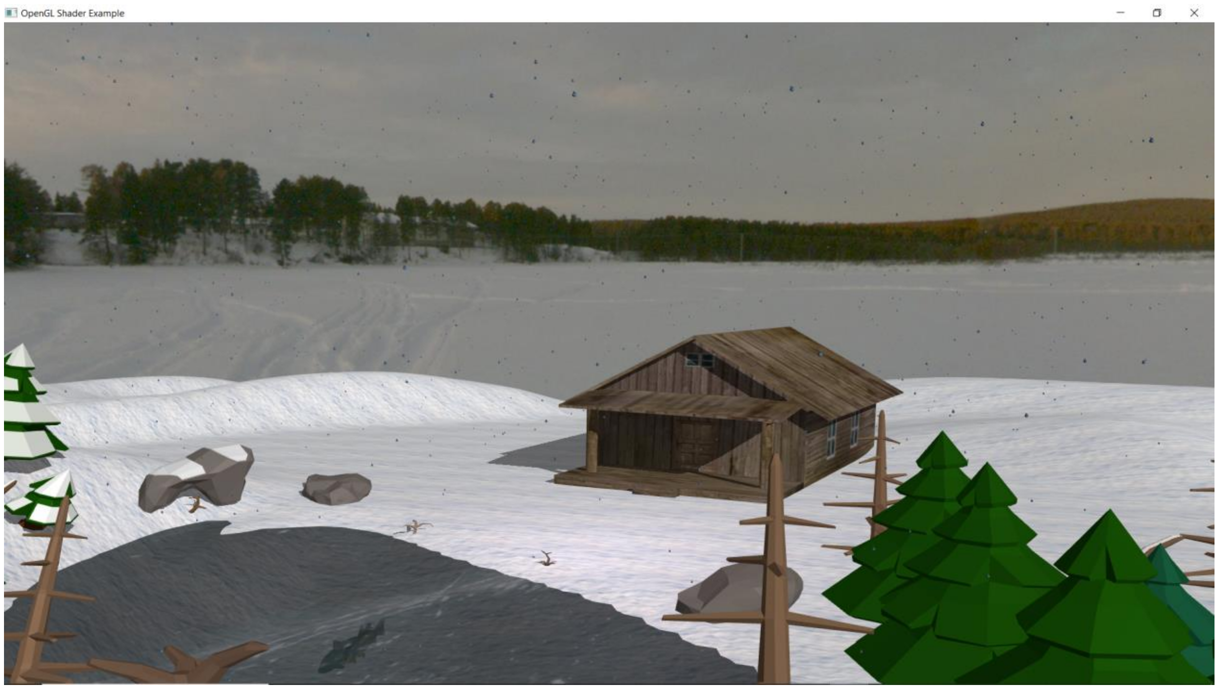
5. **Fish Animation**:
    - The fish moves along an elliptical path, incorporating sinusoidal motion for vertical movement.
6. **Automated Camera Tour**:
    - Predefined or circular camera tours of the scene (T to start/stop, C to toggle circular mode and toggle back).
7. **Dynamic Shadows**:
    - Shadows cast dynamically from light sources using shadow mapping.

<u>Visualisation:</u>

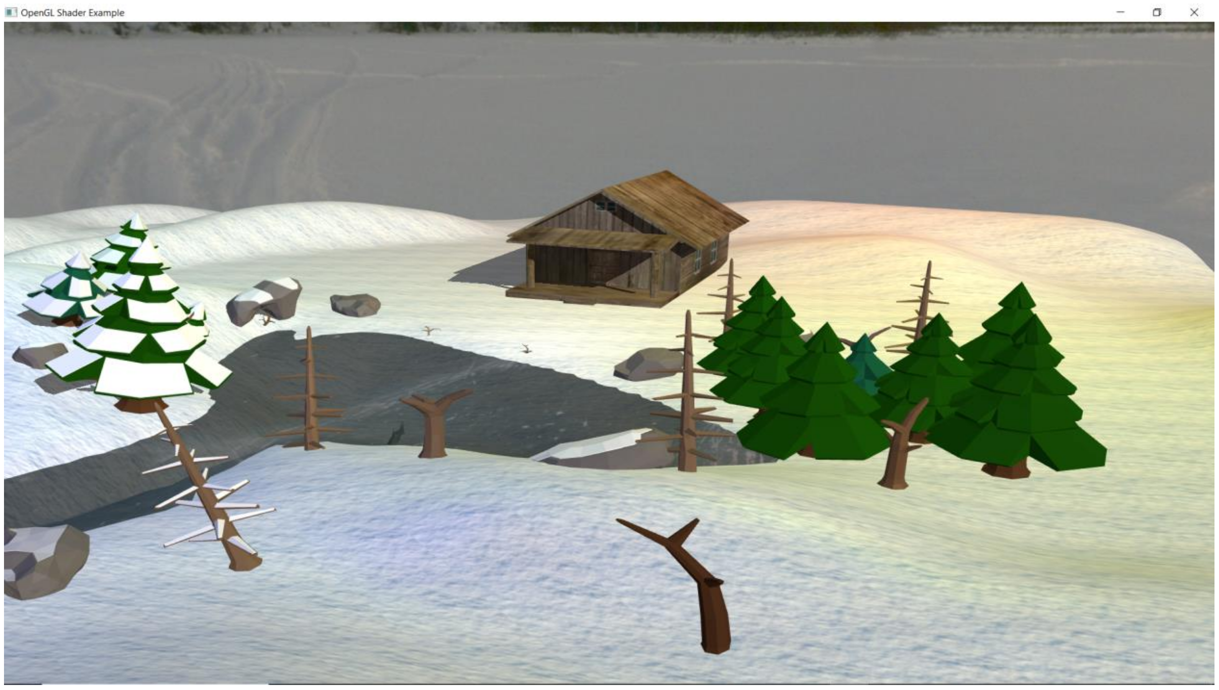
Fog on:



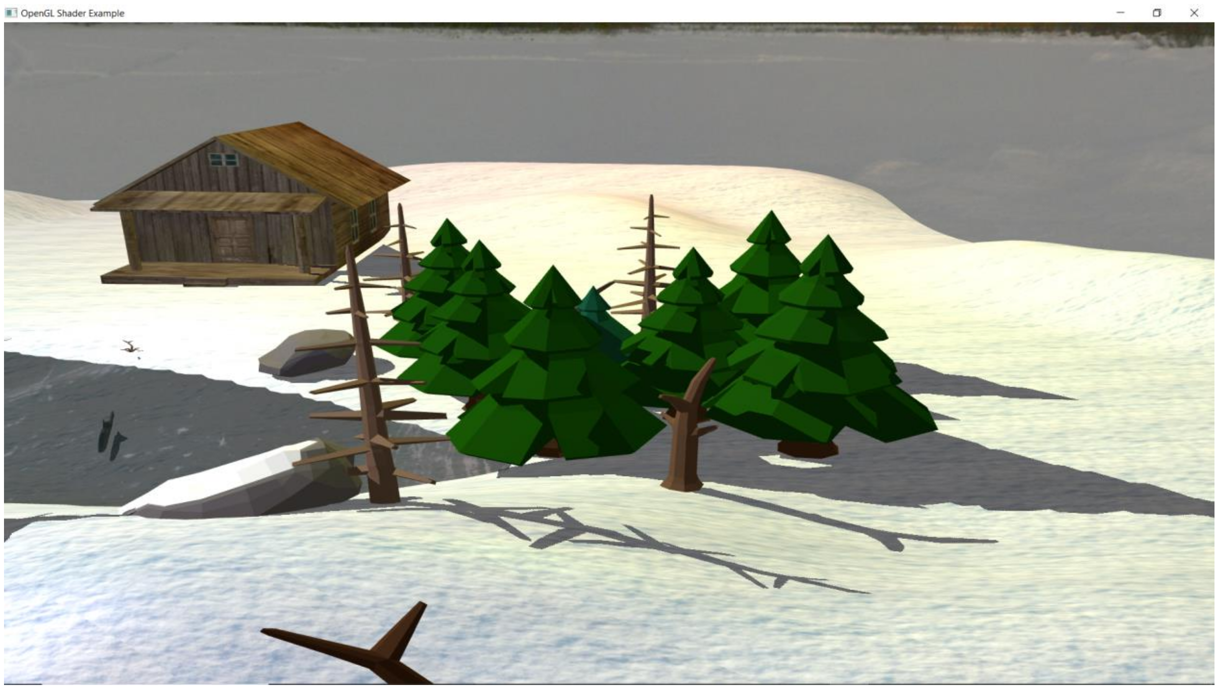
Rain on:



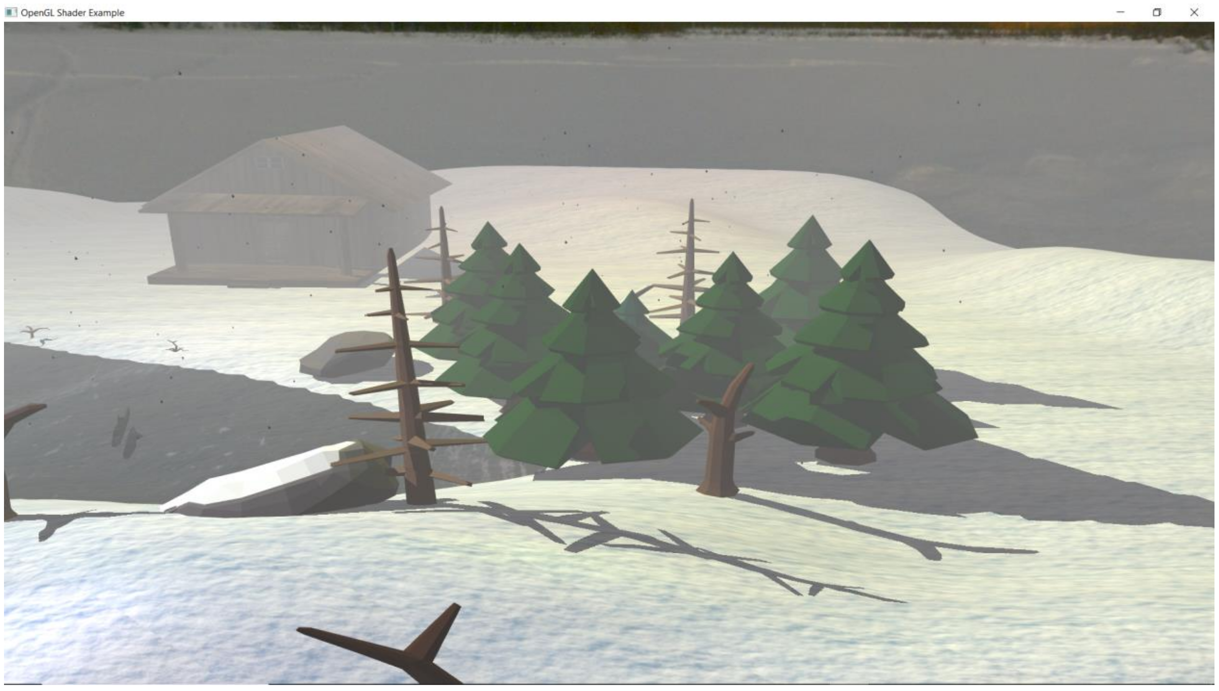
Point Lights On:



Shadows:



All combined:



4. Implementation Details

4.1 Functions and Special Algorithms

4.1.1 Possible Solutions

- **Fish Animation**: Alternatives like hardcoded motion paths or instancing were considered but were less flexible compared to the current approach of elliptical path generation.
- **Rain Effects**: Instancing multiple raindrops was the simplest and most GPU-efficient choice.
- **Shadow Mapping**: Shadow mapping was chosen for its balance of realism and performance.

4.1.2 Motivation of the Chosen Approach

- A particle based rain system and elliptical path animation were implemented to provide dynamic effects with minimal computational overhead.

4.2 Graphics Model

- **Shaders**: GLSL shaders manage rendering, lighting, fog, and shadow effects.
- **Models**: The scene incorporates .obj models (e.g., lake, fish, trees).
- **Textures**: UV-mapped textures applied for realism, optimized with mipmapping for quality.
- **Framebuffers**: a FBO object is used for shadow mapping
- **Lightning Model**: the lightning is computed using Phong model.

4.3 Data Structures

- **Vectors**: Used for storing raindrop positions, camera paths, and tour targets.
- **GLM Matrices**: Manage transformations for the camera, light, and objects.
- **Skybox**: Renders the dynamic skybox using a cubemap.

4.4 Lighting

The project uses the Phong reflection model, combining ambient, diffuse, and specular lighting components.

1. **Directional Light**: Simulates sunlight, providing uniform illumination and realistic shading. It adapts dynamically for day and night modes.
2. **Point Lights**: Add localized illumination, representing light sources like lanterns or glowing objects. Colored point lights create effects like the aurora borealis.
3. **Shadows**: Shadow mapping ensures objects cast realistic shadows that move with the light, enhancing depth perception.
4. **Fog and Lighting Interaction**: Lighting seamlessly interacts with fog to create atmospheric depth and a sense of distance.
5. **Shader Implementation**: Custom GLSL shaders handle real-time lighting calculations and integrate shadow mapping for accurate rendering.

4.5 **Effects**

1. **Fog**: A realistic fog effect enhances the depth and atmosphere of the scene.
2. **Rain Simulation**: A customizable rain effect adds realism to the environment. Raindrops are modeled as falling particles, with their movement and reset logic implemented to mimic natural rainfall across the scene.
3. **Fish Animation**: A fish animation adds life to the scene, with the fish following a dynamic elliptical path.
4. **Camera Tours**: The project features interactive camera tours that guide the user through the scene, showcasing its highlights. Circular and interpolated tours offer beautiful perspectives, with smooth transitions between key points.
5. **Shadow Computation**: Shadow mapping ensures objects cast accurate shadows based on the position of light sources, adding depth and realism.
6. **Seasonal Aesthetics**: Visual elements like snow covered boulders, frozen lakes, and dead trees contribute to the winter theme.
7. **Skybox**: A carefully designed high-resolution skybox provides the backdrop, immersing the user in a realistic winter setting.

5. Graphical User Interface Presentation/User Manual

User Controls

- **Navigation**: Move the camera with W, A, S, D (X/Z axes), Z, X (Y axis).
- **Mouse Look**: Rotate the camera view.
- **Rendering Modes**: Toggle between solid, wireframe, and point views with keys 1, 2, 3.
- **Fog and Rain**:
  - Activate/deactivate fog with F.
  - Toggle rain with R.
- **Lighting**:
  - Rotate the light source with U and J.
  - Toggle additonal point lights with K
- **Automated Tour**: Press T to start/stop and C to switch tour modes.

6. Conclusions and Further Developments

Conclusions

This project successfully integrates OpenGL features to simulate a dynamic 3D winter scene, focusing on interactivity, animations, and visual effects. The combination of lighting, shaders, and camera controls offers an engaging and immersive user experience.

Further Developments

1. **Improved Animations**: Adding additional animated objects like animals or snow falling dynamically.
2. **Interactive Environment**: Implementing interactions, such as allowing users to manipulate objects in the scene, and also solid rendering.

3. **Enhanced Visual Effects**: Incorporating more advanced techniques like reflections and refractions for water.
4. **Overall expansion of the scene**.

7. References

- [1] OpenGL Tutorials: Learn OpenGL
- [2] OpenGl-tutorial
- [2] Toturial Blender Moodle
- [3] Documentation from GP laboratories