📘 CS350/CSD3150 Project 3: Bounding Volume Hierarchy Representation

📅 Submission Details

> **Files (submit archive) due:**
>
> - 📆 **Week 9, Friday, 23:59 hrs**
> - 🕘 **Before your demonstration, Week 10, Thursday, 16:00 - 18:00 hrs**

📝 Reminder

> *"The major point of this project is for you to learn how to submit your homework correctly. Follow all instructions exactly as specified. If unsure, ask your instructor, a tutor, or another student."*

🎯 Purpose of This Project

> *"This project focuses on building the Bounding Volume Hierarchy (BVH) for given objects, using different construction strategies."*

You will:

- Load a scene with multiple models.
- Compute different types of bounding volumes (AABB, Bounding Sphere, OBB).
- Build and visualize Bounding Volume Hierarchies (BVH) using top-down and bottom-up methods.

🧩 Requirements

🧱 Scene Setup (25%)

- Load a scene using **any 8 models** from the provided dataset.
- For each object (OBJ file), compute:
  - Axis-Aligned Bounding Box (AABB)
  - Bounding Sphere (PCA-based method)
  - Oriented Bounding Box (OBB using Covariance Matrix and Eigen Vectors)
- Implement:
  - **First-Person Camera** (WASD movement)
  - **Top-Down Debug Camera** or **PIP Camera Overview**
- Allow toggling between bounding volume types.

🏗️ Bounding Volume Hierarchy (70%)

You must build BVHs using **AABB** and **Bounding Spheres**.

🏛️ Top-Down Method

- Experiment with different **split-point strategies**:
  - Median of BV centers
  - Median of BV extents
  - K-even splits on one axis
- Choose **split planes** along X, Y, or Z to minimize total volume of child nodes.
- Implement multiple **termination conditions**:
  - Stop when a leaf has a **single object**.
  - Stop when a leaf has a **maximum of two objects**.
  - Stop when **tree height = 2**.
- Allow toggling between the different split methods and termination conditions.

🏛️ Bottom-Up Method

- Create a bounding volume for every object.
- Use heuristics to merge child nodes into parent nodes:
  - Nearest neighbor
  - Minimum combined child volume
  - Minimum combined child surface area
- **Discuss the effect of choosing different heuristics in your README.**

🌈 Visualization

- Color each level of the BVH with a unique color:
  - Red for level 0
  - Orange for level 1
  - And so on...

⚙️ Submission

1. Create a copy of project directory project-3 named <login>-<project-3>. That is, if your Moodle student login is foo, then the directory should be named foo-project-3. Ensure that directory foo-project-3 has the following layout:

```
📘 foo-project-3            # 🖋 You're submitting Project 3
├─ 📂 include                   # 🖨 Header files - *.hpp and *.h files
└─ 📂 src                       # 🌟 Source files - *.cpp and .c files
└─ 📂 shaders                   # 🌟 Shader files - *.vert and .frag files
|   └─ 📝 my-project-3.vert   # 📝 Vertex shader file
|   └─ 📝 my-project-3.frag   # 📝 Fragment shader file
└─ 📝 README.txt                # 📝 [IMPORTANT] Don't forget to add this
```

2. Make sure you are not using absolute path for shader files, it will not work on instructor's machine.
3. Inside foo-project-3, add README.txt file. README must Include:
    - UI usage instructions (especially if not described in the project brief)
    - Assumptions and crash conditions
    - Completed parts
    - Incomplete/buggy parts with explanation
    - File paths, function names, and line numbers of key logic
    - Test platform details (e.g., `Windows 11, NVIDIA 3070, OpenGL 4.6`)
    - Weekly time breakdown
    - Any other useful notes
        - Add **key mappings**, assumptions, and known issues.
        - Track your weekly effort hours.
        - Be explicit about:
            - What is completed
            - What is not working, and why
            - Platform and GPU details (lab machine or your home setup)
4. Re-run the CMake command to build the new project named foo-project-3. If you're unsure how to run the CMake command, please refer to the <your-sample-framework-location>/README.md file.
5. Build and execute project foo-project-3 by opening the Visual Studio 2022 solution in directory build. Test it, make sure it works good.
6. Use File Explorer to open directory <your-sample-framework-location>/projects. Open the command-line shell by typing cmd [and pressing Enter in the Address Bar]. Execute the following PowerShell command to zip it with name [by typing the script's name in the shell and then pressing Enter].

    ```
    powershell if (Test-Path foo-project-3.zip) { Remove-Item foo-project-3.zip -Force };
    Compress-Archive -Path foo-project-3 -DestinationPath foo-project-3.zip
    ```

    [IMPORTANT] Please use only the command provided above for zipping, as it generates the archive in the specific format required by the automation tool for grading. Using any other method may result in incompatibility or failed evaluations.
7. Submit this zip file on Moodle.

🧪 Grading Breakdown

Component	Weight
Interactivity and Display	25%
└ Objects loaded and displayed	5%
└ 3 Bounding Volumes (AABB, Sphere, OBB)	10%
└ Cameras and Scene Rendering	10%
Bounding Volume Hierarchy	70%
└ Top-down (AABB)	10%
└ Top-down (Bounding Sphere)	10%
└ Split-point Strategies	10%
└ Termination Conditions	10%
└ Bottom-up (AABB)	10%
└ Bottom-up (Bounding Sphere)	10%
└ Merging Heuristics	10%
Miscellaneous Issues	5%
└ Missing README	-2%
└ Compilation/Execution/Scene Errors	-3%
Total	100%

⭐ Extra Credit (10%)

- **Using Oriented Bounding Boxes (OBB) to build the Bounding Volume Hierarchy**

🔗 References

- Review the course syllabus and lecture slides for further implementation details.

🛡️ Legal Notice

> *"This project teaches you about real-world scene acceleration structures like those used in rendering, collision detection, and physics engines."*

🛠️ Geometry Toolbox Tips

Build Bounding Volume Hierarchies (BVH)

(Project 3: BVH Construction)

- Implement **Top-Down BVH construction**:
  - Use different split strategies:
    - Median of centers
    - Median of extents
    - K-even splits
- Implement **Bottom-Up BVH construction**:
  - Start from leaves, merge using heuristics:
    - Nearest neighbor
    - Minimum combined volume
    - Minimum combined surface area
- Allow toggling:
  - Between BVH types (AABB, Sphere)
  - Between construction methods (Top-Down, Bottom-Up)
- Visualize BVH levels:
  - Use different colors for each depth level.

🛠️ *Tips*:

- Always **test** tree balance after construction.
- Optimize tree traversal for culling or collision tests.