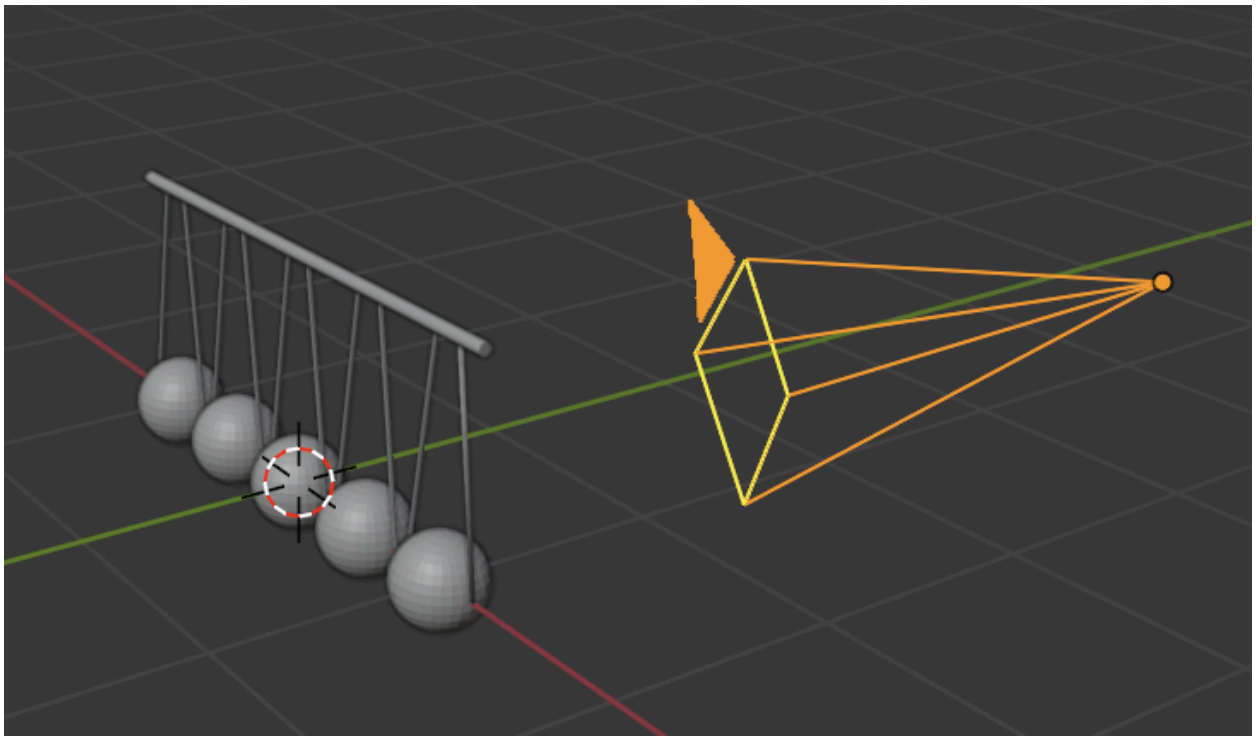
CGR 2025 Lab 2: Scene Write/Read

Duration: 1 hour
Tools: Blender, g++ (or other C++ compiler)

Objective

Export a Blender scene to an ASCII file and write a C++ program to read it. Ensure that you have installed Blender and g++ (or your compiler of choice).

You will be given a test scene in Blender. For example, it may look like this:



Write scene information

Use GenAI coding assistants to write Blender python code that will obtain all the objects in the scene and write them out to an ASCII file. You may choose your own format provided it contains the following information:

a. For cameras: location, direction of the gaze vector, focal length, sensor width and height and its film resolution (pixels in X and Y directions).
b. For point lights: location and radiant intensity.
c. For spheres: location and radius (scale in Blender).
d. For cubes: translation, rotation and 1D scale.
e. For planes: 3D coordinates of its four corners.

An example output (not corresponding to the scene shown above) might look like this (below).


```
[
  {
    "name": "Cube",
    "type": "MESH",
    "location": [
      0.0,
      0.0,
      0.0
    ],
    "rotation_euler_radians": [
      0.151293,
      -0.206914,
      0.099181
    ],
    "scale": [
      1.0,
      1.0,
      1.0
    ],
    "material_base_color_rgb": [
      0.8,
      0.8,
      0.8
    ]
  },
  {
    "name": "Light",
    "type": "LIGHT",
    "location": [
      4.076245,
      1.005454,
      5.903862
    ],
    "rotation_euler_radians": [
      0.650328,
      0.055217,
      1.866391
    ],
    "scale": [
      1.0,
      1.0,
      1.0
    ],
    "material_base_color_rgb": null
  },
  {
    "name": "Camera",
    "type": "CAMERA",
    "location": [
      7.358891,
```

```
            -6.925791,
            4.958309
        ],
        "rotation_euler_radians": [
            1.109319,
            0.0,
            0.814928
        ],
        "scale": [
            1.0,
            1.0,
            1.0
        ],
        "material_base_color_rgb": null,
        "camera_properties": {
            "lens_type": "PERSP",
            "focal_length_mm": 50.0,
            "sensor_fit": "AUTO",
            "sensor_width_mm": 36.0,
            "sensor_height_mm": 24.0,
            "shift_x": 0.0,
            "shift_y": 0.0,
            "clip_start": 0.1,
            "clip_end": 100.0
        }
    },
    {
        "name": "Sphere",
        "type": "MESH",
        "location": [
            0.0,
            0.0,
            0.0
        ],
        "rotation_euler_radians": [
            0.0,
            -0.0,
            0.0
        ],
        "scale": [
            1.0,
            1.0,
            1.0
        ],
        "material_base_color_rgb": null
    }
]
```

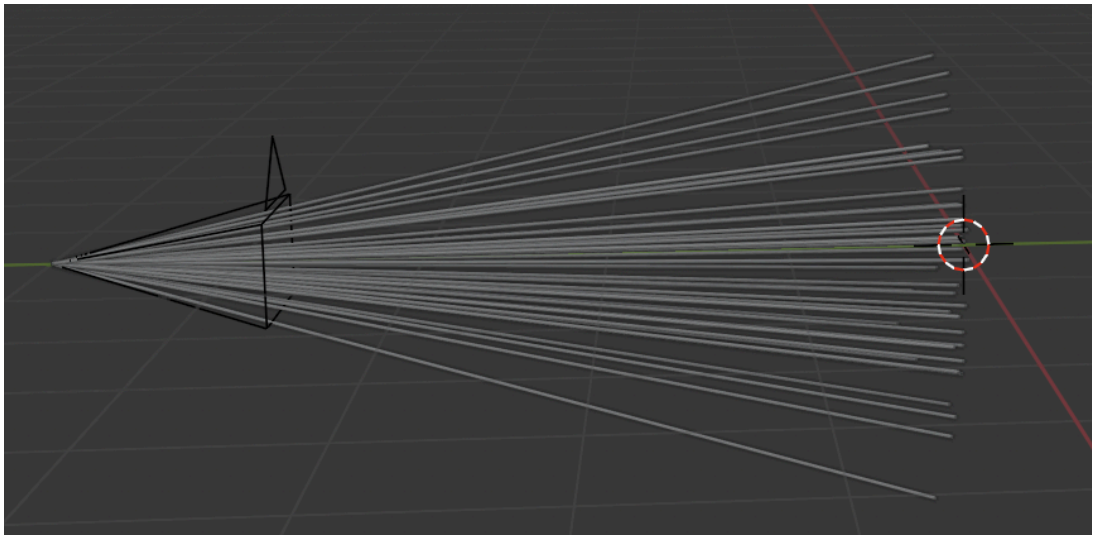

Find and read online resources about Euler angles for rotation as used by blender.

Read scene information

Use GenAI coding assistants to generate C++ source code that can read information from the above file and create objects of the appropriate classes. Use a hierarchy of Shape classes to represent the shapes. Each shape should store its relevant parameters (e.g. radius and length for a cylinder) and its transformation (translation, rotation and scale).
Once you have read the file and created the objects. Write a separate loop, over all objects in the scene, outputting their properties to standard output stream (console text).

Ray visualiser

Write a blender script that will read a text file containing rays (origin, direction) and visualise them as thin cylinders in 3D. Use a sphere or cube as an arrowhead to show directionality. You will be given a .blend file with a camera, show 50 primary rays for that camera within the Blender scene (as cylinders above) by:
1) exporting to your ASCII format.
2) reading the file via the C++ reader
3) generating 50 primary rays in the C++ file
4) writing the rays to a text file.
5) reading rays from the text file from within Blender and
6) visualising the rays. For example, see the simple Blender scene below.



Presentation

Once you have completed the above tasks, prepare a brief presentation of max 2 mins [1 mark] containing:

1. A slide explaining how to convert Euler angles to rotation matrices [1 mark]
2. Screenshots of the test scene and one of your own modelled scenes (they do not need to be rendered) [1 mark]
3. Any challenge(s) you faced during writing/reading files [1 mark]
4. Demonstrate your visualiser on both scenes (does not need to be rendered) [1 mark]