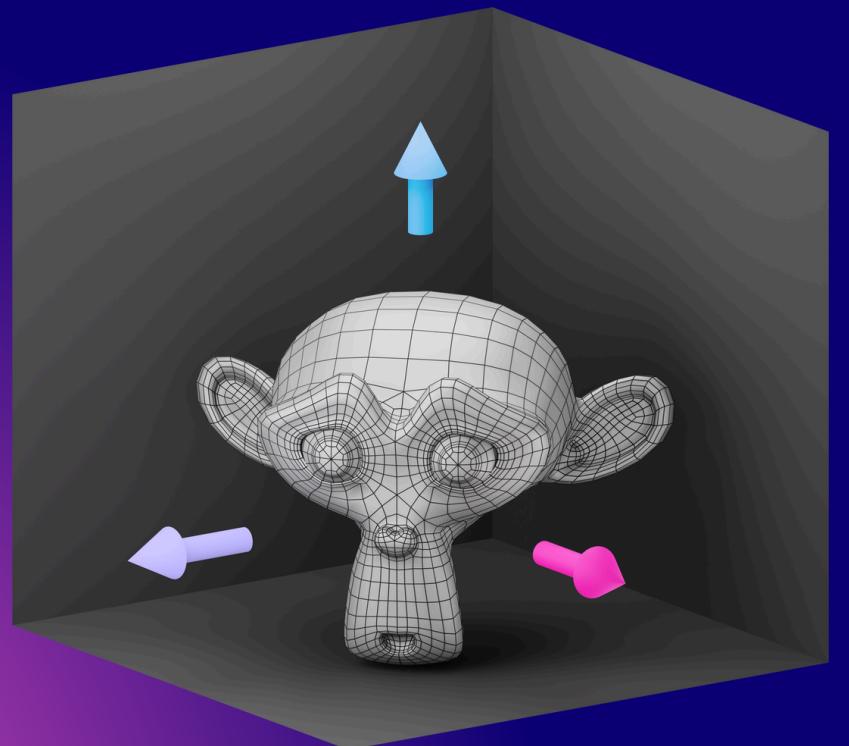


RASTERIZER

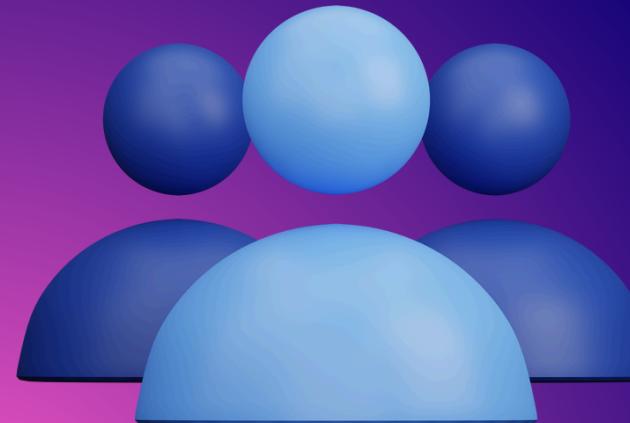
Monkeys learn to code



DE CE ACEST PROIECT? MOTIVAȚIA NOASTRĂ



Am ales să facem acest proiect deoarece ne-am dorit să învățăm mai multe despre concepțele fundamentale din grafică computerizată, în special despre rasterizare – una dintre componentele de bază ale multor aplicații moderne, de la jocuri video la interfețe grafice. Considerăm că înțelegerea detaliată a acestui proces ne oferă o perspectivă practică asupra modului în care scenele 3D sunt transformate în imagini 2D.



CE ESTE UN RASTERIZER?

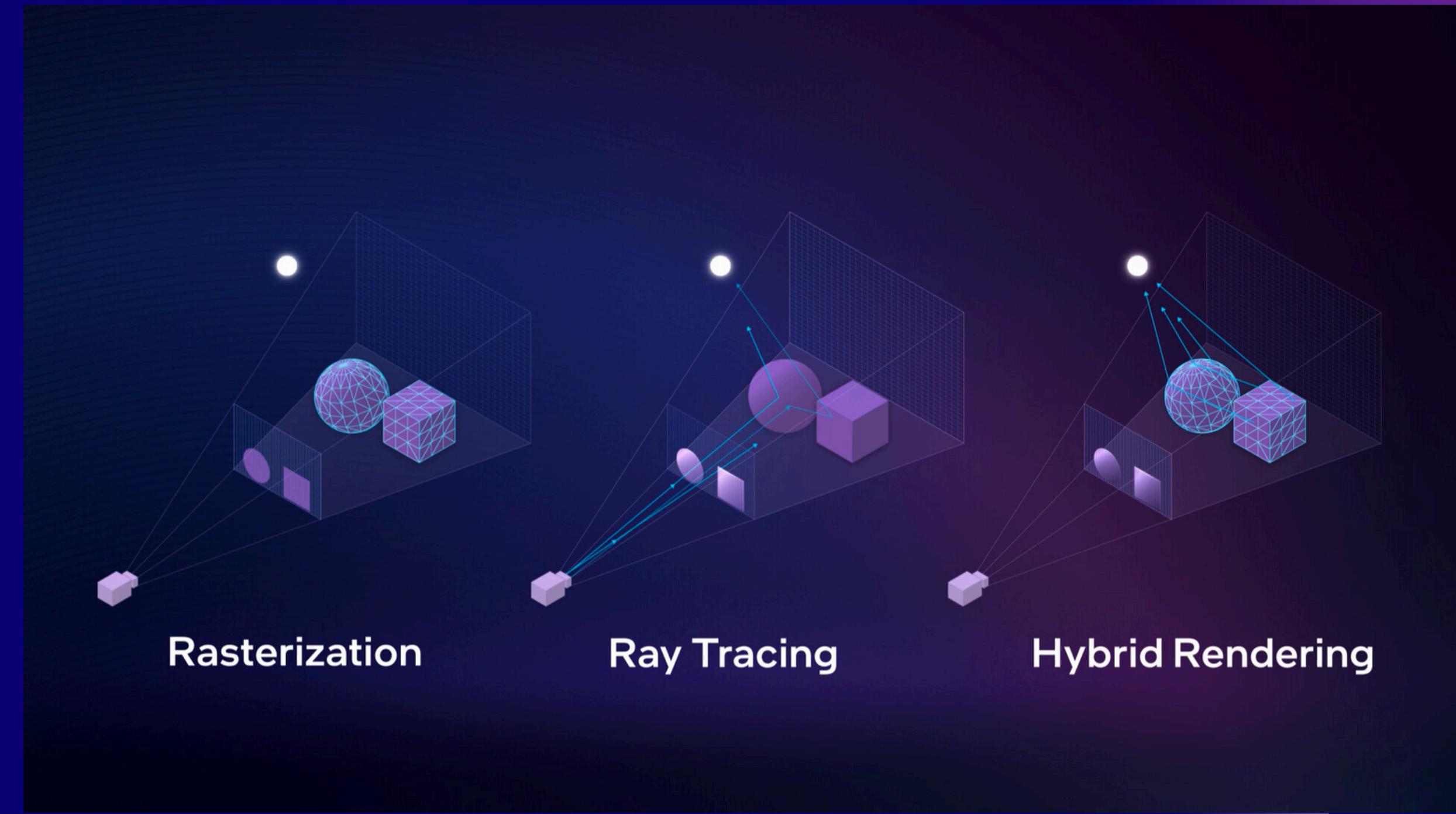
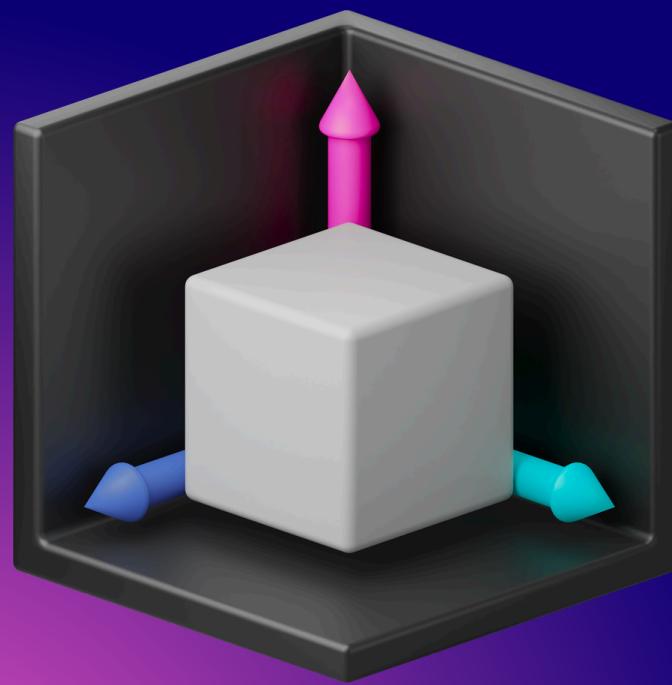
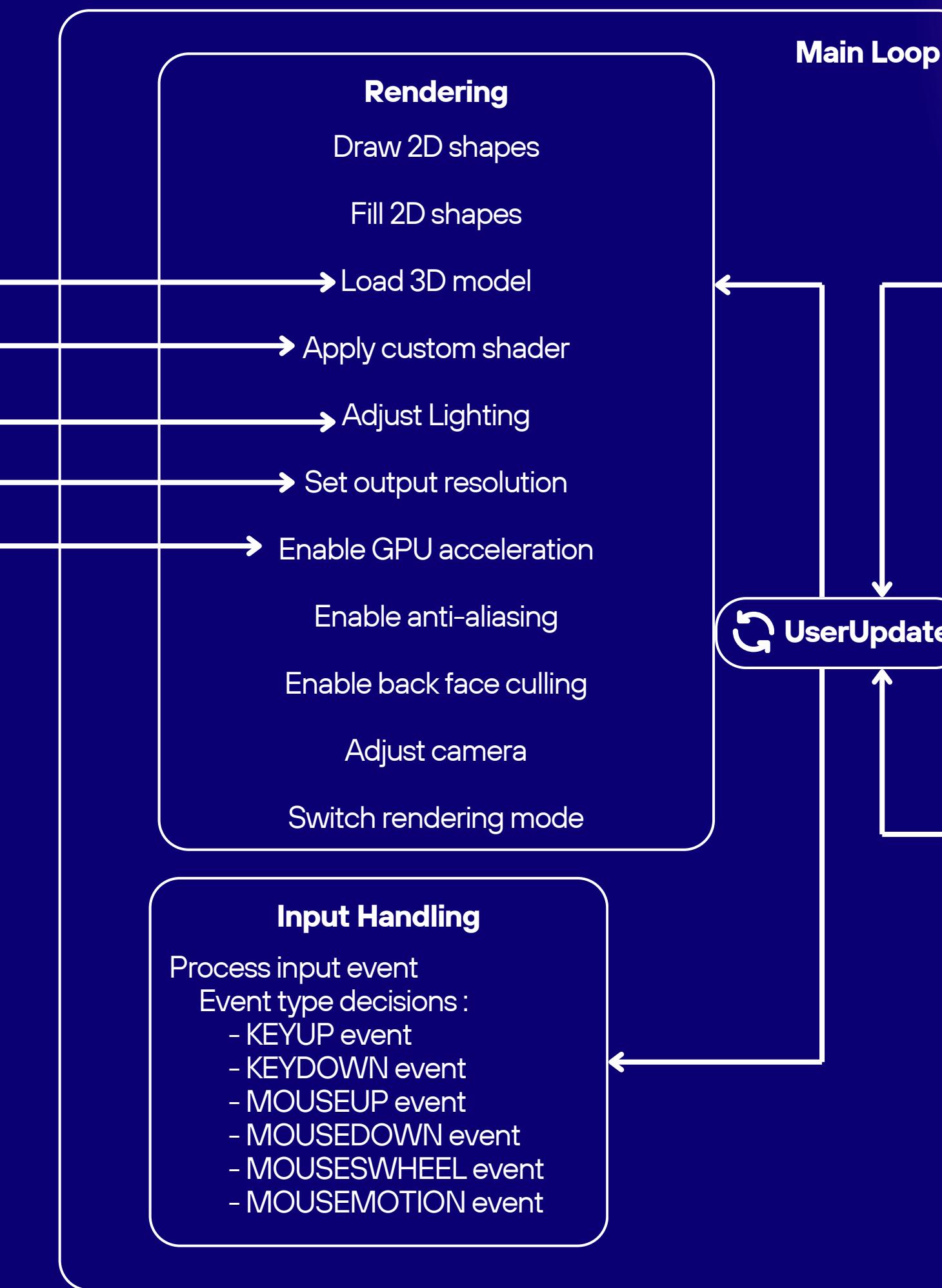
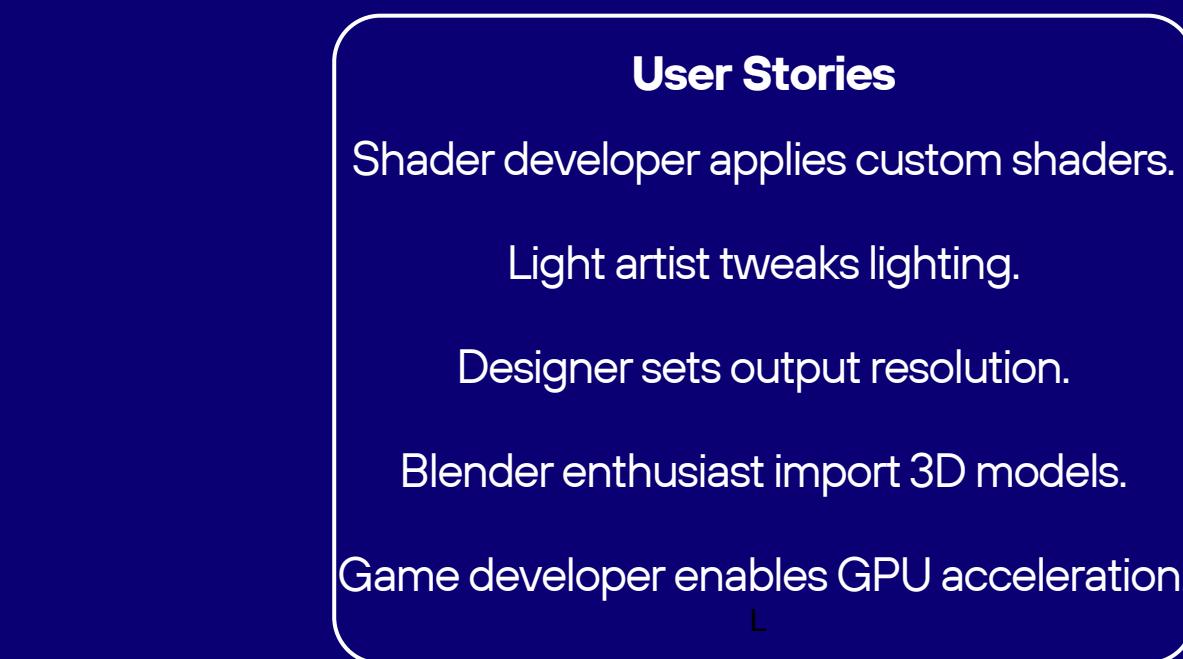


Diagrama UML



TECHNOLOGII FOLOSITE

- **C++17** - limbaj principal, performanță
- **SDL3** - abstractizare cross-platform a sistemului de fereastră
- **GLM** - matematică vectorială/matricială
- **Python 3.13** – scripting, UI
- **pybind11** - expune API-ul C++ către Python pentru scripting



ASPECTE TEHNICE

```
class CustomRenderer(IzerRaster.Renderer2D):
    def __init__(self, appName, width, height):
        super().__init__(appName, width, height)
        obj_path = "suzanne.obj"
        self.object_loaded = self.loadObj(obj_path)
        print(self.object_loaded)

        self.theta = 0.0
        self.renderMode = IzerRaster.RenderMode.SHADED_WIREFRAME

    def UserUpdate(self):
        events = self.poolInputEvents()

        self.theta += self.deltaTime()
        transform = IzerRaster.translate(
            IzerRaster.vec3(0.0, 0.0, 8.0))

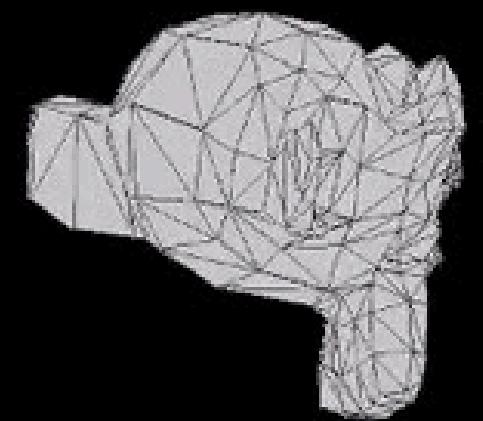
        rotateY = IzerRaster.rotate(self.theta, IzerRaster.vec3(0.0, 1.0, 0.0))

        new_mesh = self.applyRenderMatrix(
            transform * rotateY, self.object_loaded)

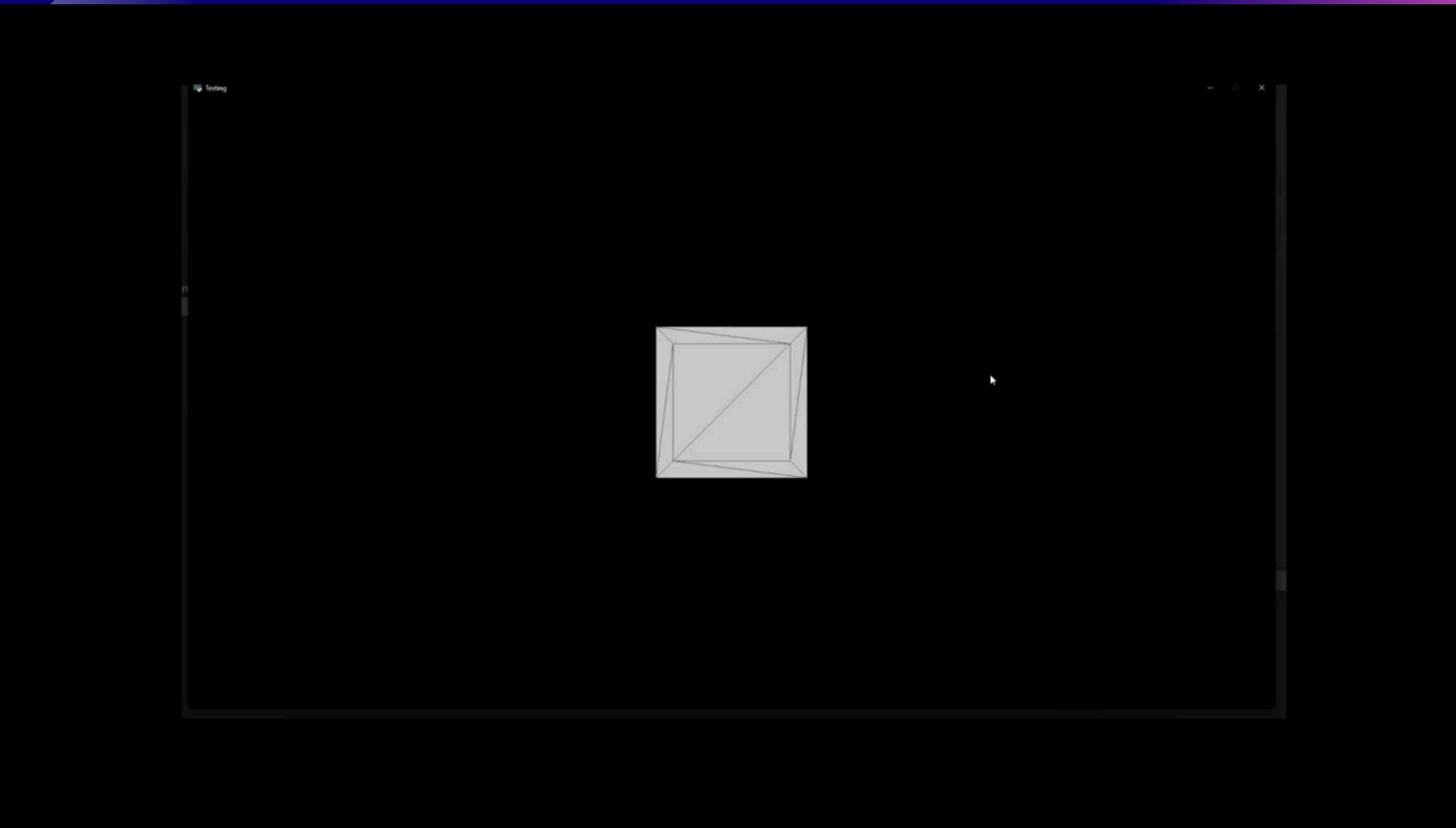
        self.drawObj(new_mesh)

renderer2D = CustomRenderer("Testing", 1920, 1080)
renderer2D.Init()
renderer2D.Run()
renderer2D.Quit()
```

ASPECTE TEHNICE



DEMO



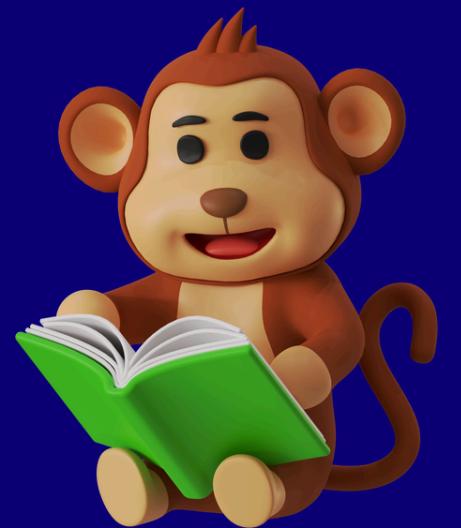
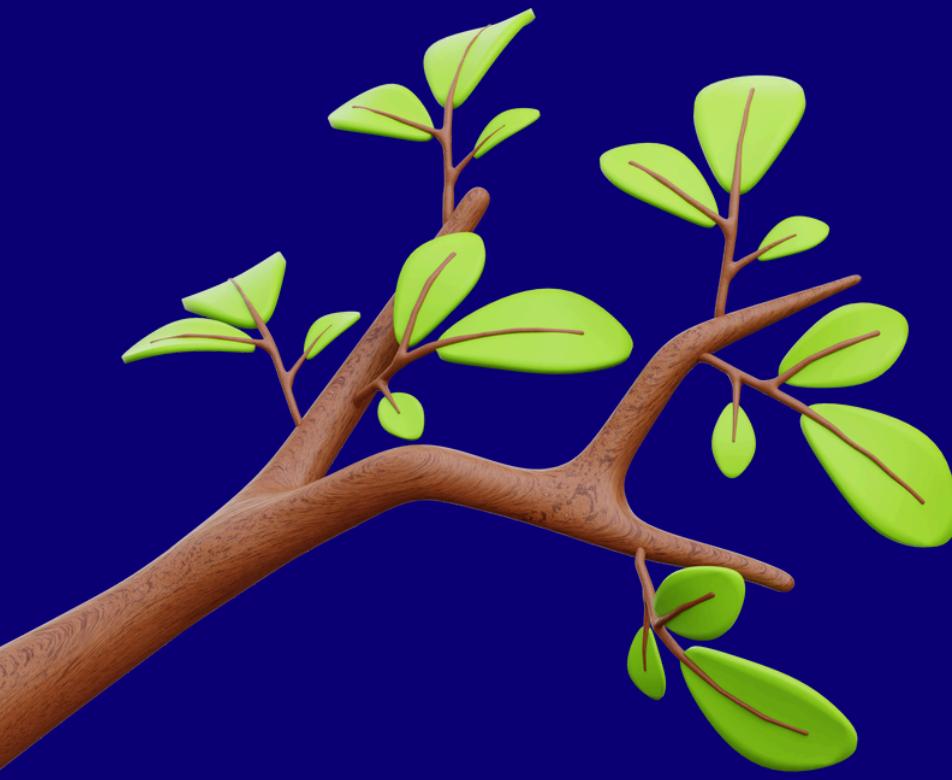


PLANURI DE VIITOR

- adăugare texture mapping
- update GUI
- optimizare performanta
- imbunatatiri vizuale



VA
MULTUMIM!



,

