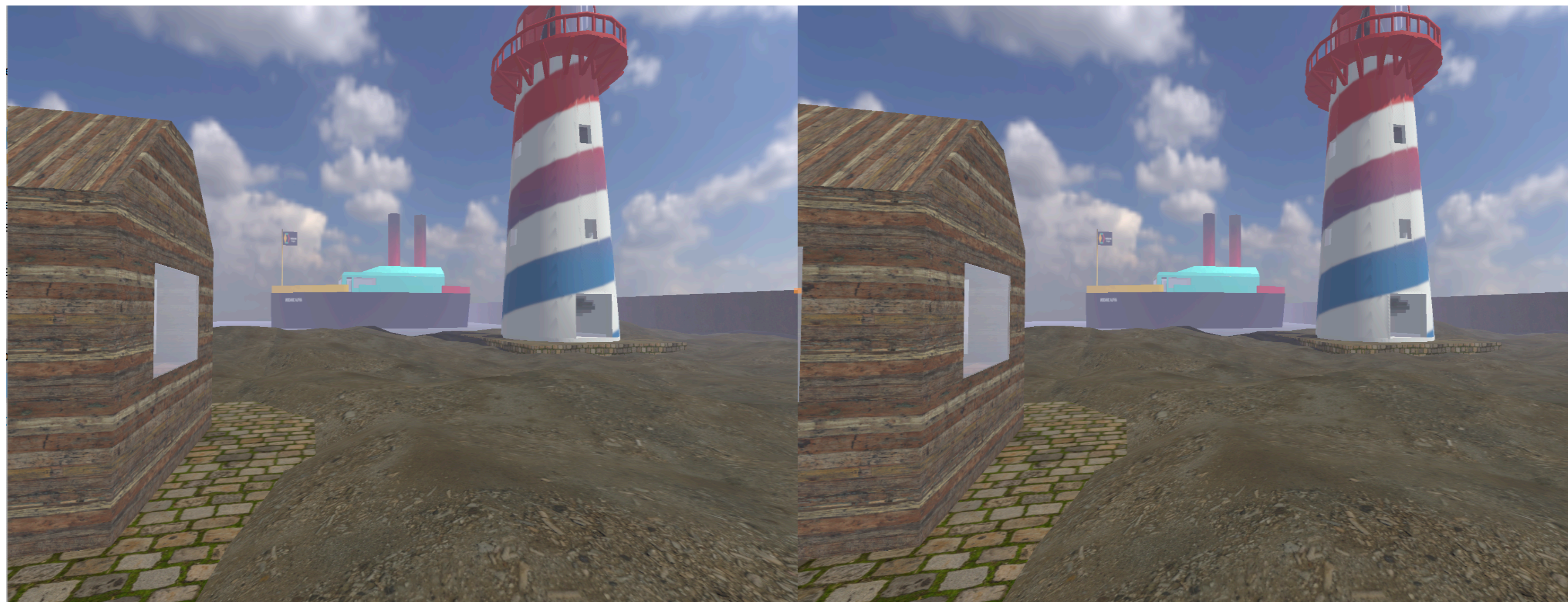


# Virtual Reality in Pharo: Challenges and Demo



Ronie Salgado  
Desromech E.I.R.L.





# Talk Outline

- Massive Refactoring
- Rigid Body Physics Engine in Pharo
- Game Framework inspired in Unreal
- Level Editor
- VR Support and Challenges
- Demos

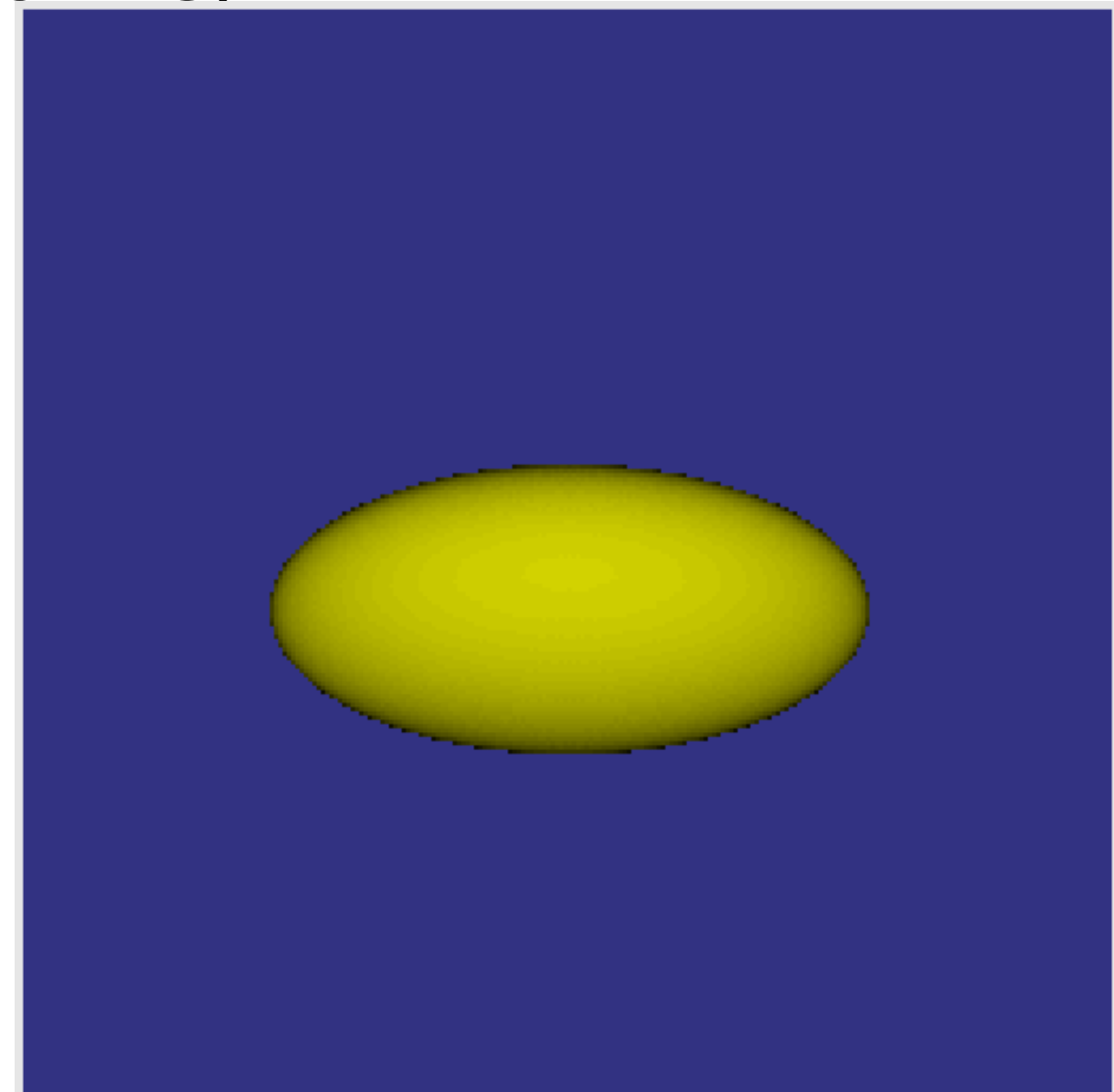
# Massive Refactoring

- Focus on Scientific, Engineering and Prototyping Support
- Favor Flexibility vs Raw Speed (e.g. Physics in Pharo)
- Highly Modular Reimplementation
- Stricter Separation between Model and Presentation
  - Duplicates static Data between CPU and GPU (e.g textures, meshes)
  - Improves Stability across image sessions
- Separate Git Repos
- Reduce loading time if fewer repos and needed

# Math Library

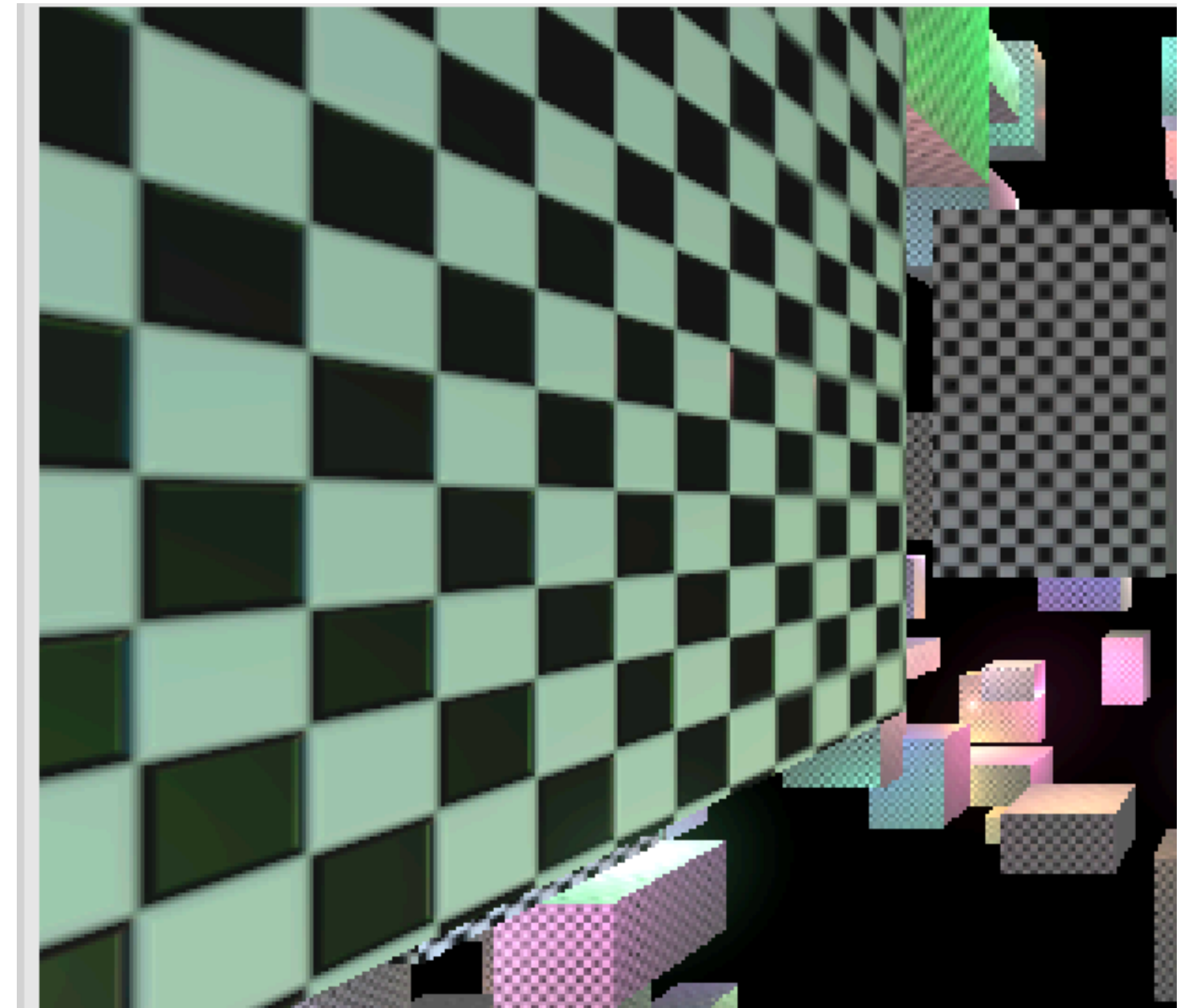
<https://github.com/desromech/woden-core-math>

- Linear Algebra math for 3D graphics (Vector, Matrix, Quaternion, Transforms)
- Collision Detection Algorithms
  - GJK Distance Function for Convex-Convex distance/intersection
  - GJK Sweep Test
  - Several analytical recasting methods
- Bounding volumes and spatial subdivision data structures
- Noise functions for procedural generation (value, gradient/perlin, voronoi)



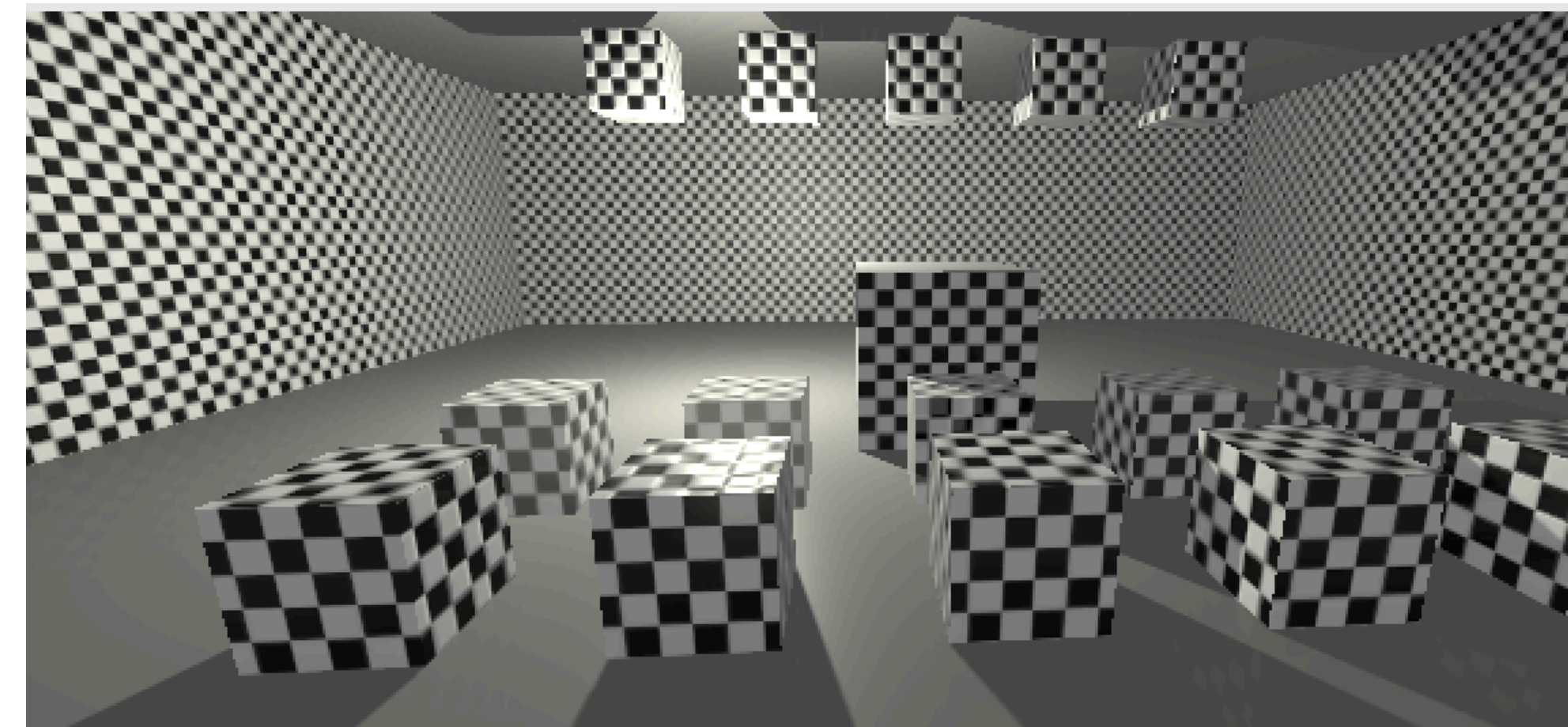
# Scene Graph

- <https://github.com/desromech/woden-core-scene-graph>
- The Graphics Engine Core
- Focus on only displaying 3D scene
- Highly limited support for interactions
- Scenes are integrated with the inspector



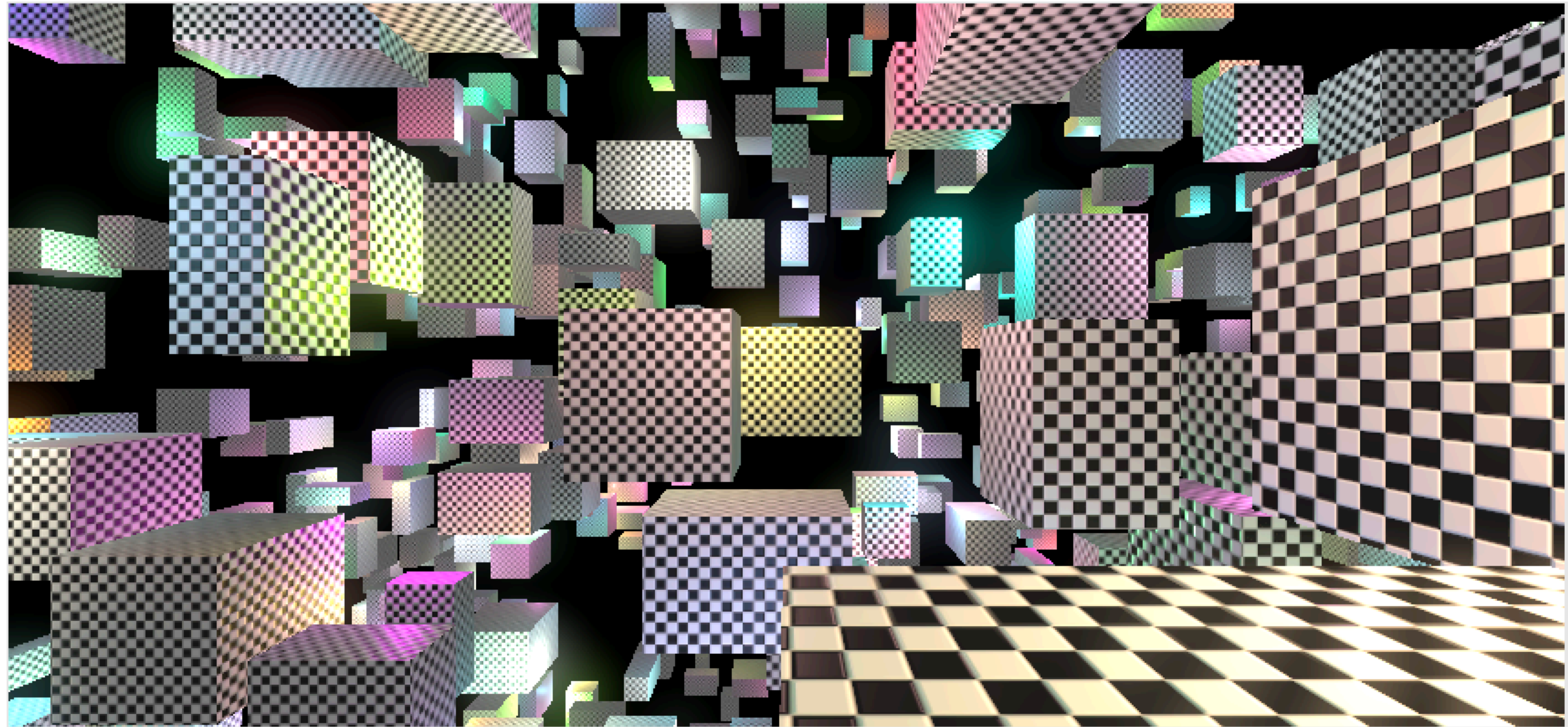
# Scene Rendering Algorithm

- Base on Doom 2016 Algorithm (See Adrian Courrèges Article)
- Clustered Forward Rendering Algorithm
- Support for many non-shadow casting lights
- Good support for transparency
- AbstractGPU thin layer
  - Vulkan, Direct3D 12 and Metal





# One Thousand Lights





# Scene Graph Inspector

The image shows a Playground IDE window titled "Playground" with a subtitle "a WDSScene". The interface includes a toolbar with icons for "Do it", "Publish", "Bindings", "Versions", and "Pages". The code editor on the left contains the following code:

```
1 WDSScene new
2   add: (WDMeshBuilder new
3     addCubeWithWidth: 1.0 height: 1.0 depth: 1.0;
4     mesh);
5   add: ((WDPointLightSource new
6     color: (Vector3 x: 0.8 y: 0.8 z: 0.2);
7     intensity: 5.0;
8     influenceRadius: 4.0;
9     asSceneNode)
10     position: (Vector3 x: -1.5 y: 1.5 z: 1.6);
11     yourself);
12   yourself
```

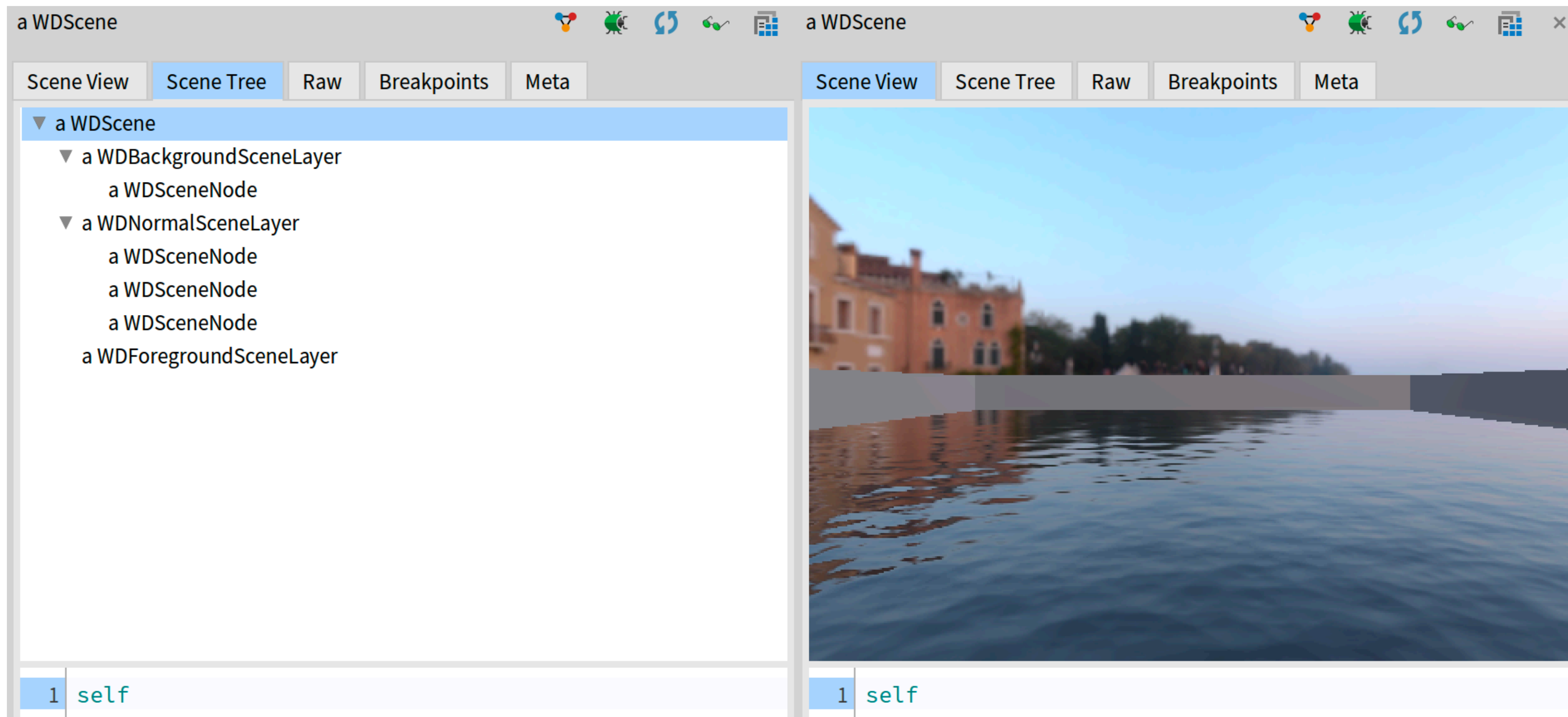
The scene view on the right shows a 3D scene with a yellow cube in the center. Below the scene view is a console area with the following output:

1	self
---	------

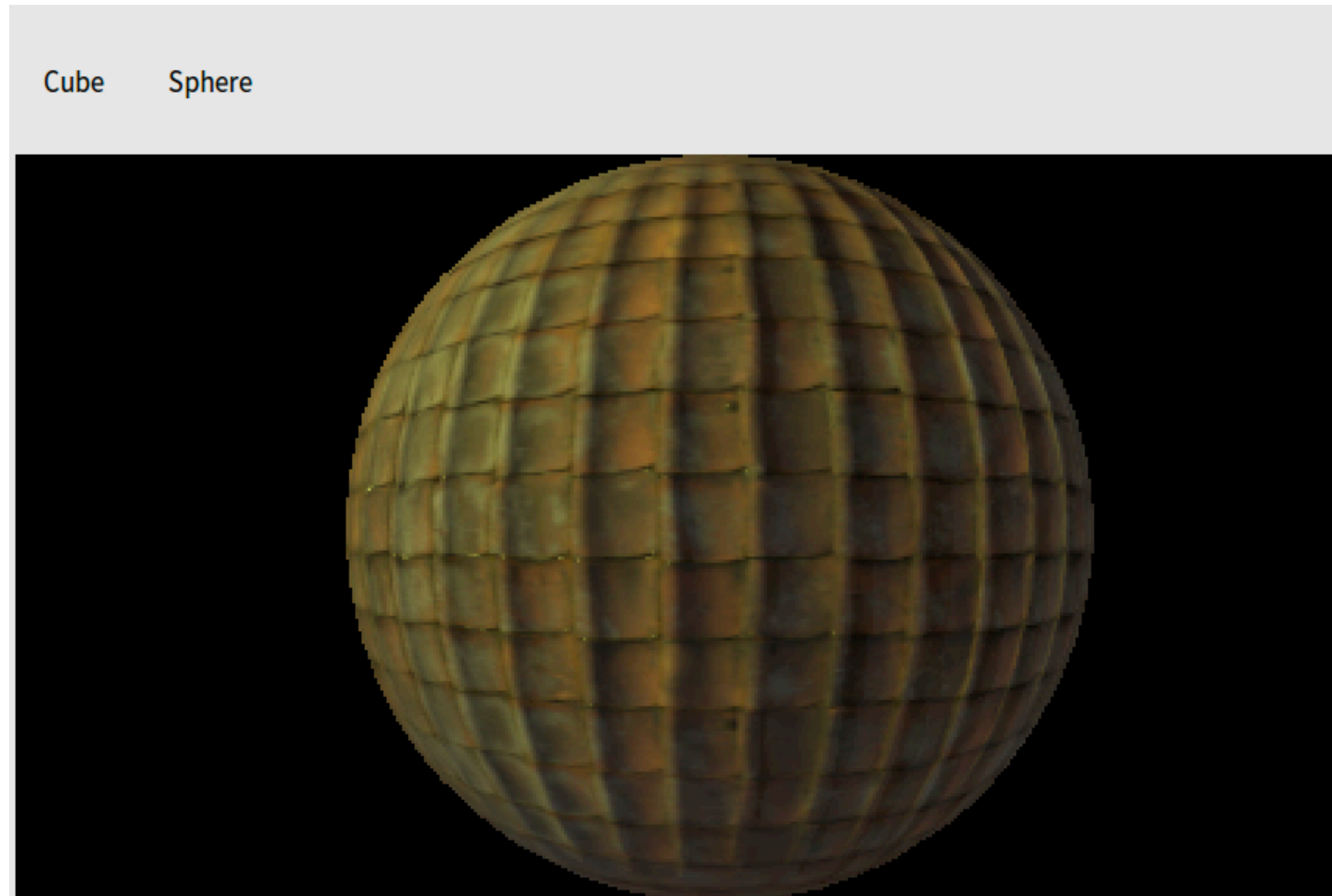
The status bar at the bottom left shows "Line: 12:11" and a "+L" button.



# Scene Graph Inspection



# Material Preview Inspection

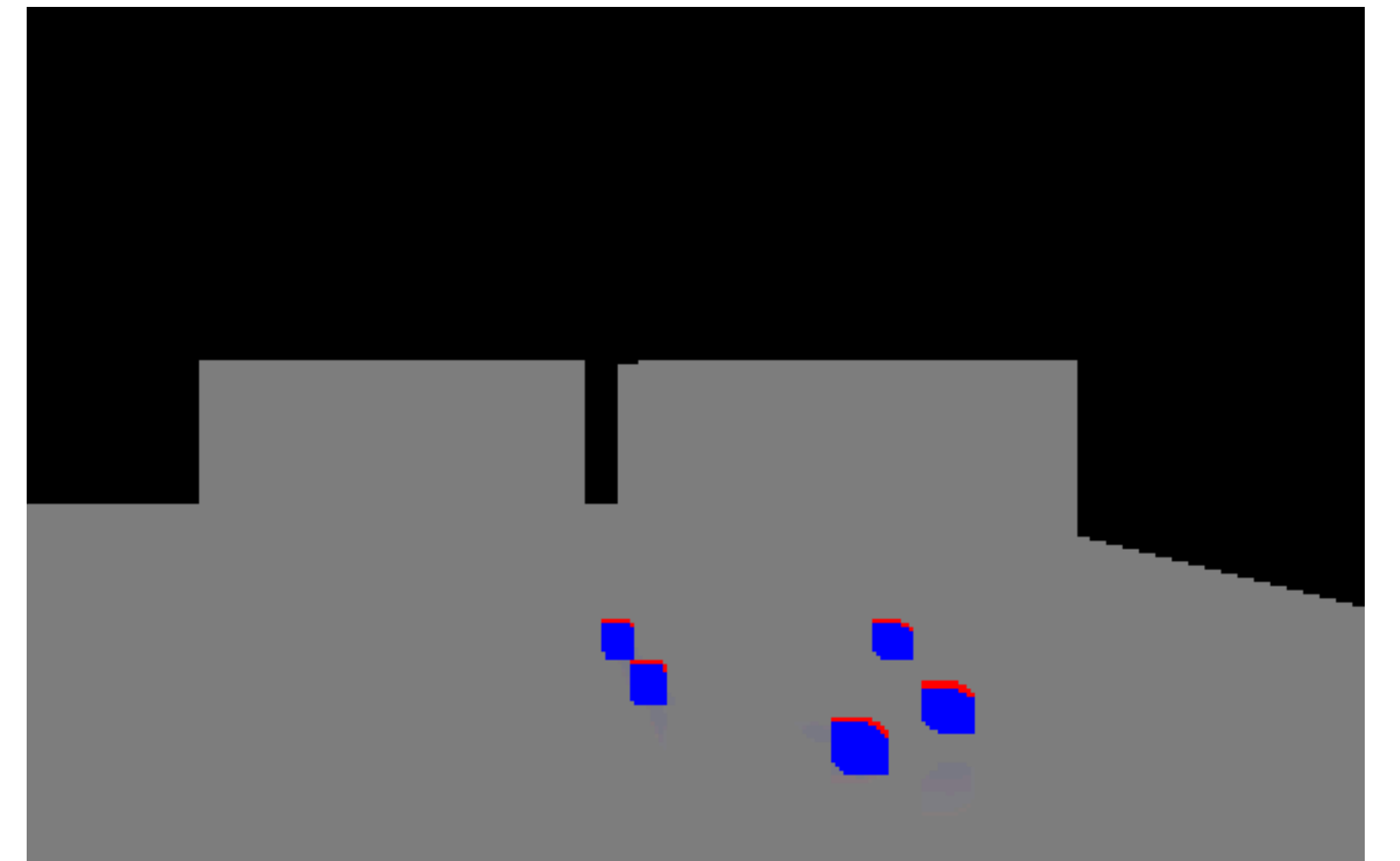




# Rigid Body Physics

<https://github.com/desromech/woden-core-physics>

- Strong Inspiration on Bullet
- Completely implemented in Pharo
- Discrete Collision Detection
- Based on Millington book “Game Physics Engine Development”
- Bugs remain to be solved



# Game Framework

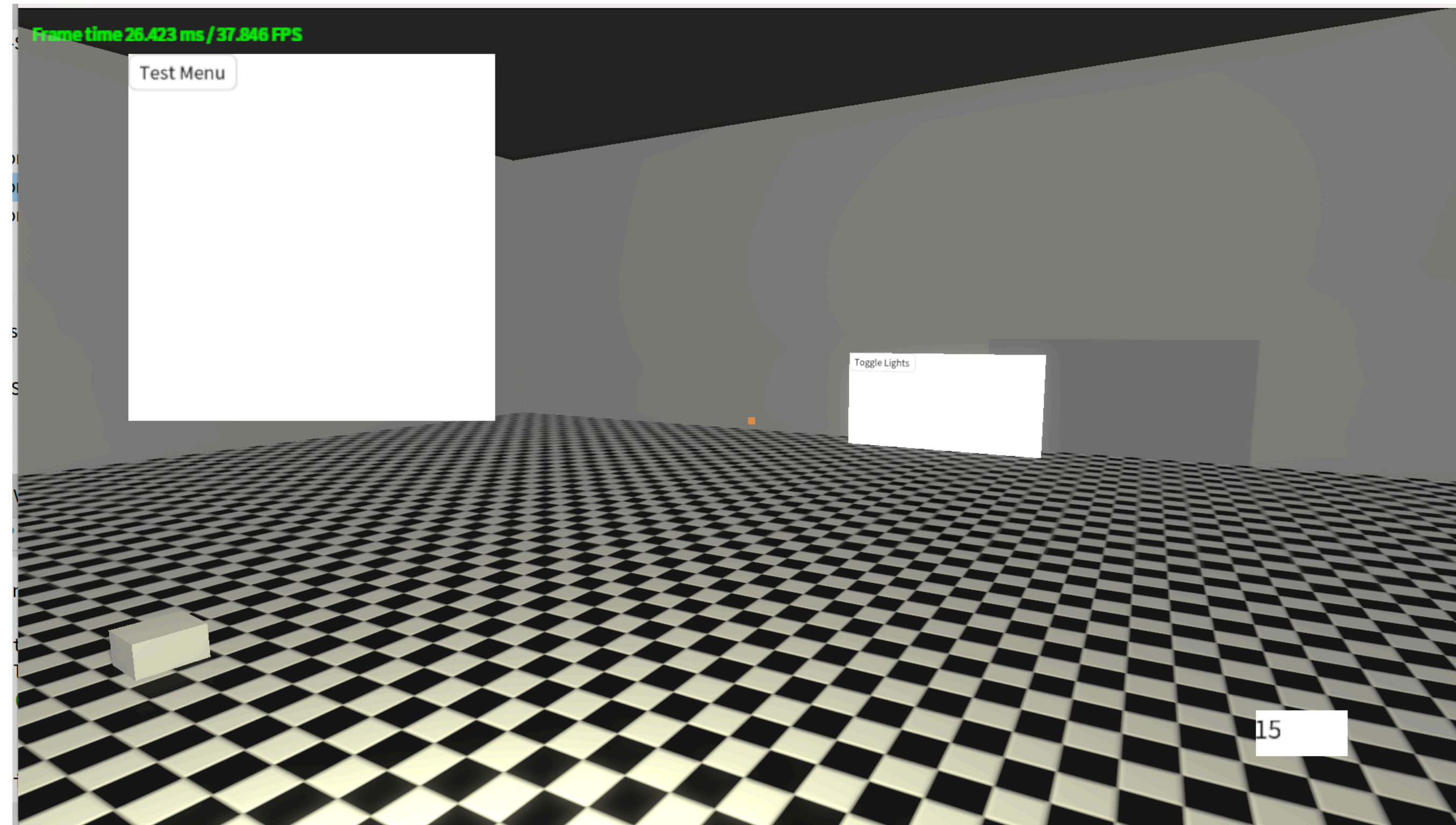
<https://github.com/desromech/woden-core-game-framework>

- Actor Model inspired on Unreal Game Framework
- Integrates different components
- For prototyping games, and highly interactive applications
- VR Interactions are implemented on this level



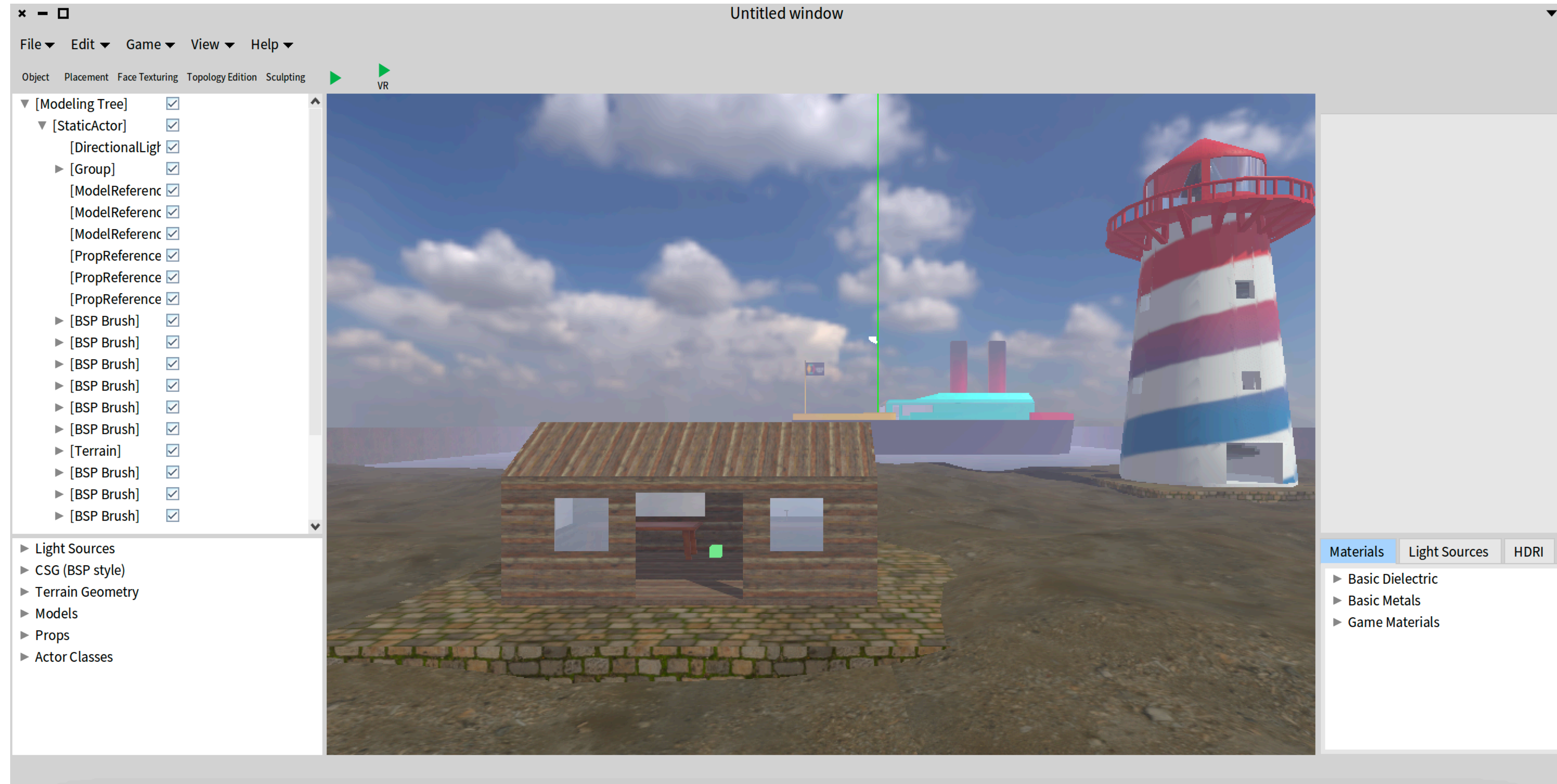
# Game Framework

## Bloc/Toplo in the Environment



# Level Editor

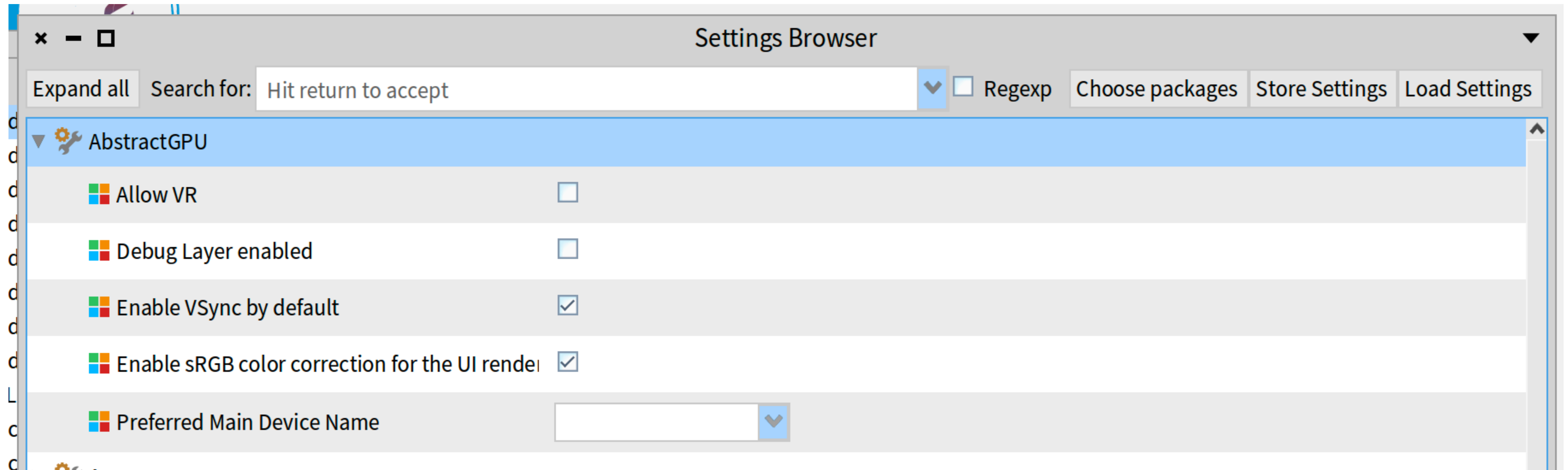
Level editor inspired on old Quake style BSP editor.





# VR Support

- VR Support Disabled by Default. Enabled via the Allow VR option



# VR Challenges

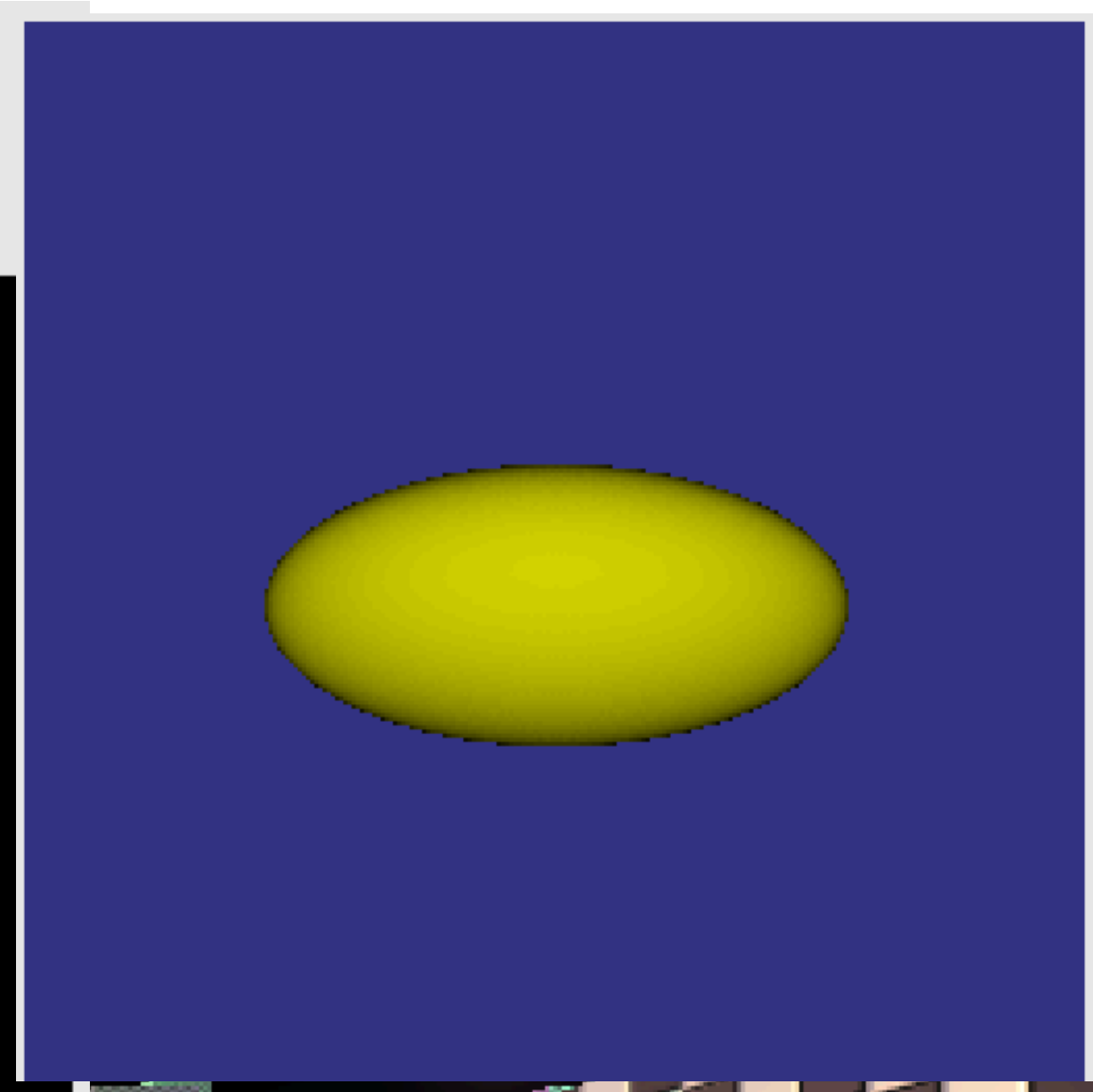
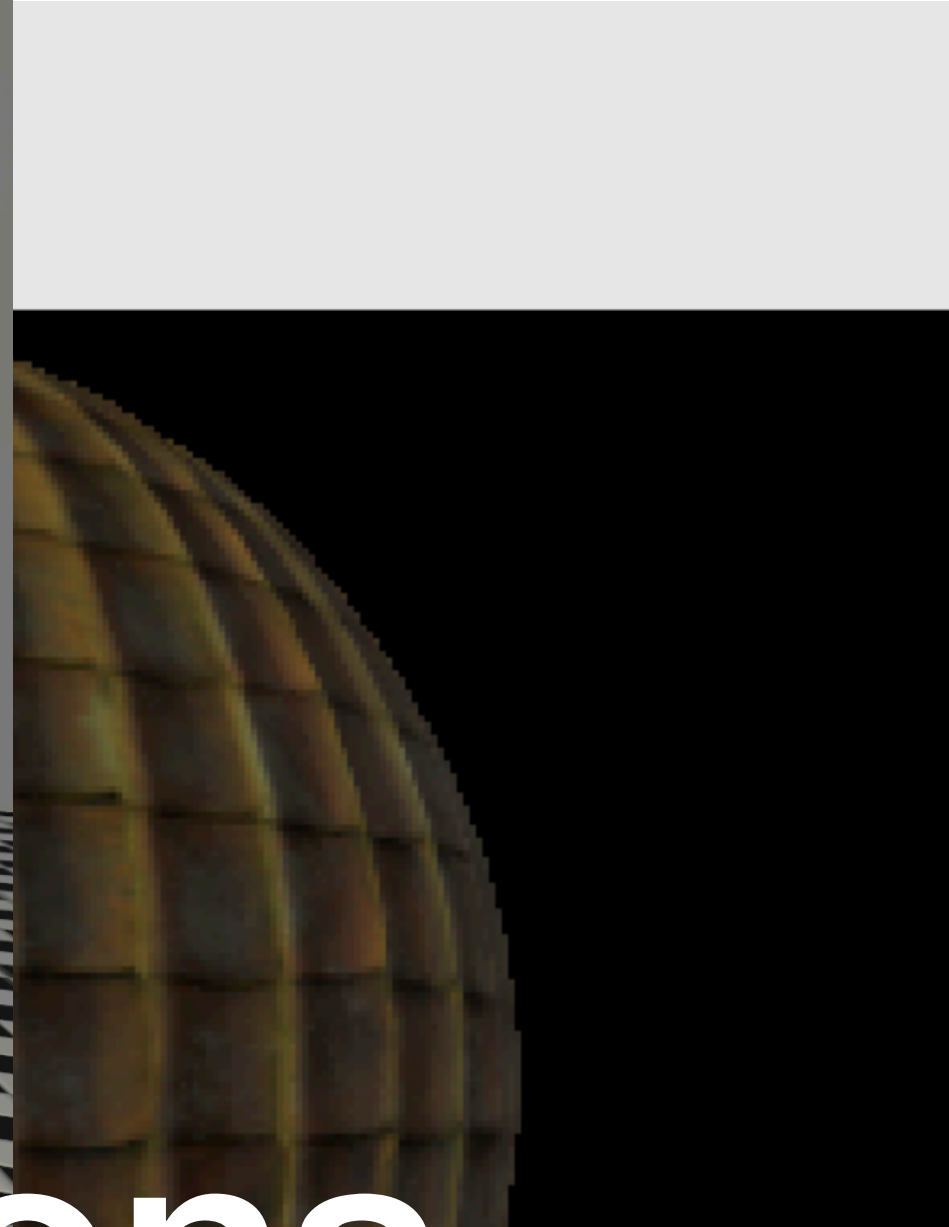
- Performance Constraints
  - Stereoscopic rendering
  - Higher FPS to avoid motion sickness
- Modeling 3D Interactions
  - Actual 3D cursors
  - Picking and handling objects physically





**Demo Time!!**





# Questions

