





Visualization of the BM@N experiment in Unity: intermediate results

16th Collaboration Meeting of the BM@N Experiment at NICA

K.R. Tukhvatulin, Saint-Petersburg State University
S.A. Nemnugin, Saint-Petersburg State University
A.V. Lebedeva

Aim and objectives

The aim: to develop a new instrument for analysis and immersive visualization the BM@N detector and particle tracks.

Objectives:

- Choose a development environment
- Build an application for visualizing the BM@N detector geometry
- Import the ROOT geometry file of BM@N
- Implement basic functions for viewing (rotate, zoom, adjust transparency), interaction (select) with the detector
- Add particle track visualization
- Integrate VR support for immersive visualization
- Conduct performance and usability tests

Advantages of an immersive visualization

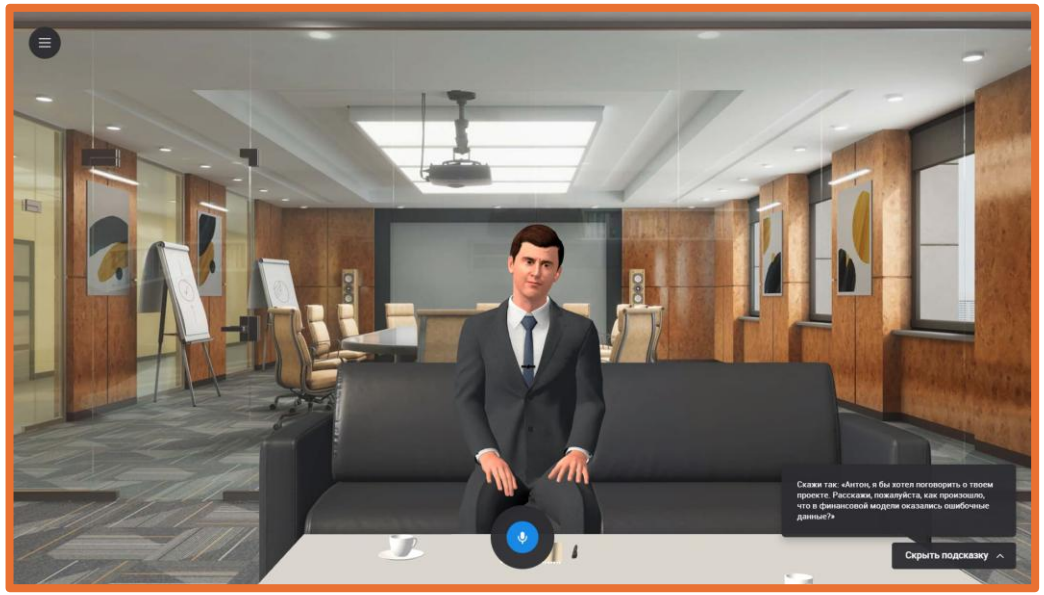
[https://tobias.isenberg.cc/personal/papers/Wang_2019_VBI.pdf]

- 3D VR environments and scientific visualization naturally match
- VR – less frustrating compared to traditional tools
- The best for users' preference and performance of understanding spatial relationship

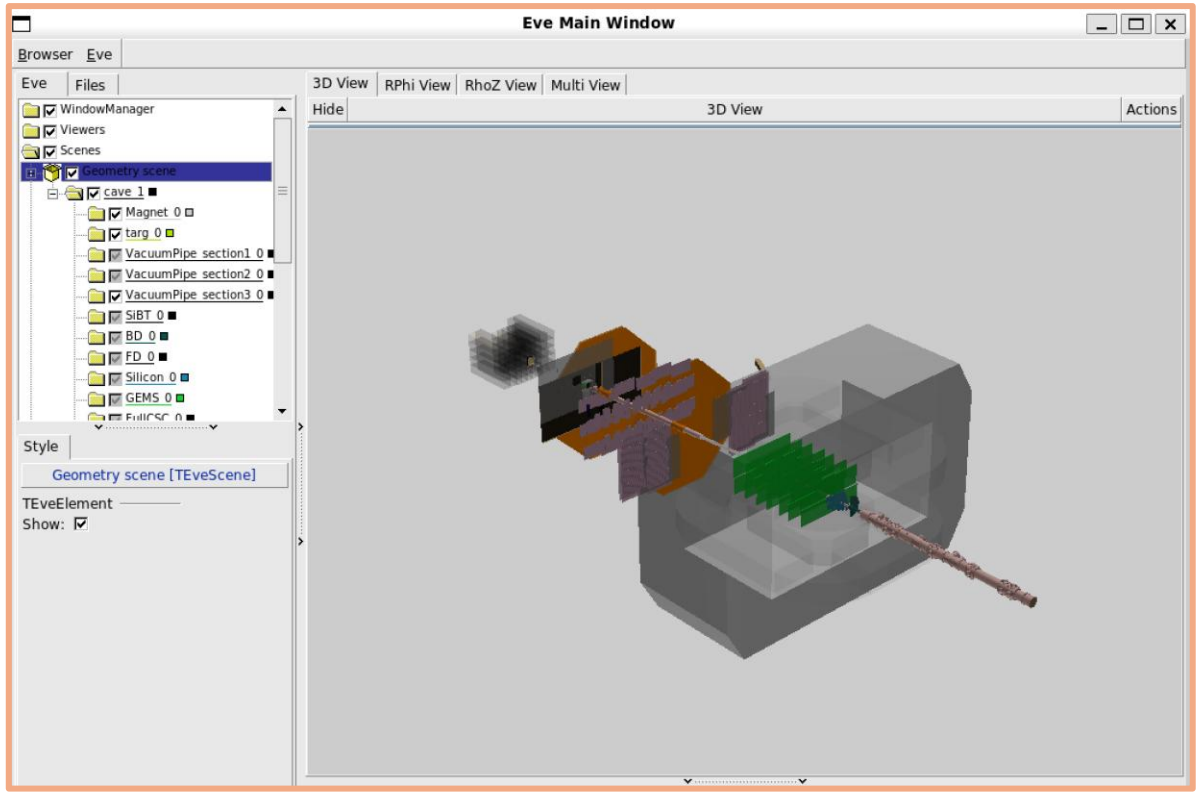


VR medicine simulator (SamSMU) [https://samsmu.ru/edu/vr/]

VR business simulator (SPBU) [https://cdet.spbu.ru/vr-feedback]



Visualization of the BM@N. Current status



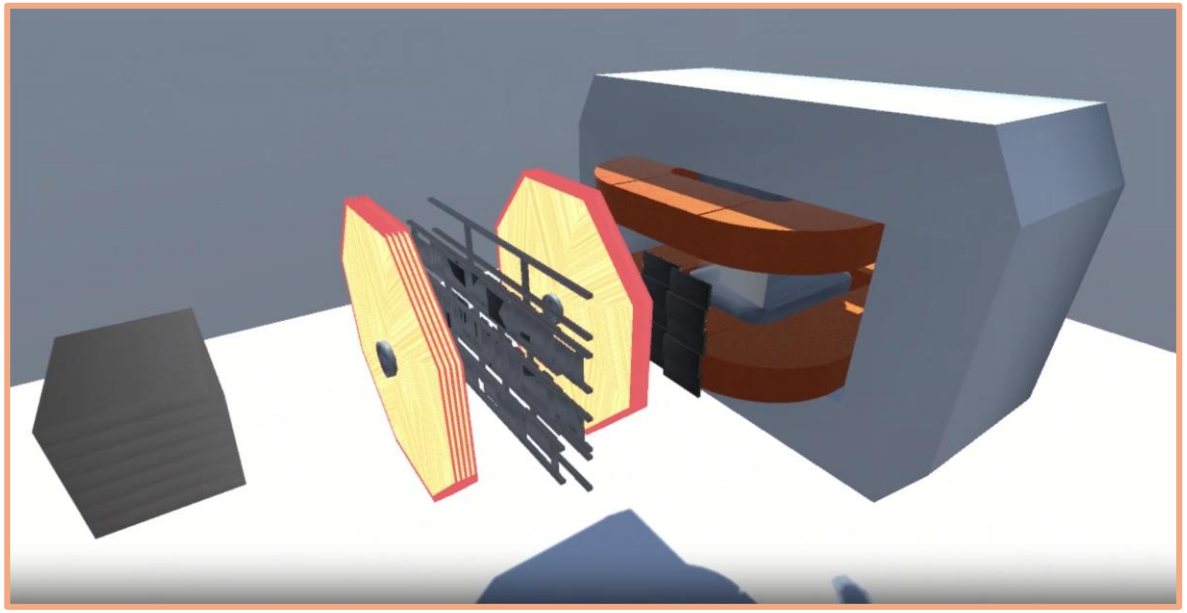
Event Display.

Based on the ROOT EVE package

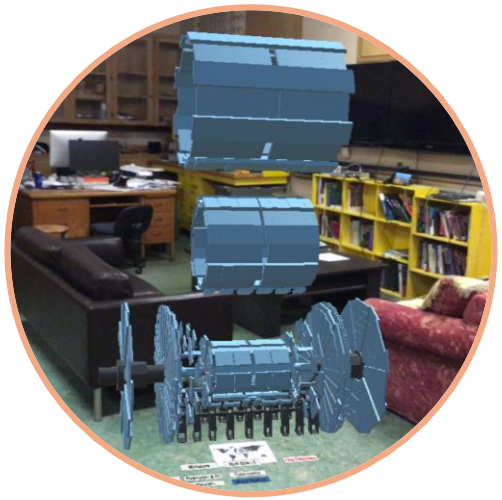
[https://root.cern/doc/master/group__TEve.html]

VR prototype for the BM@N experiment visualization by Anastasiia Iusupova

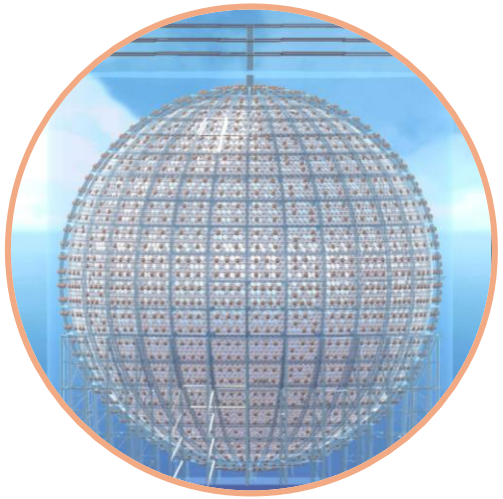
[https://doi.org/10.1016/j.nima.2024.169619]



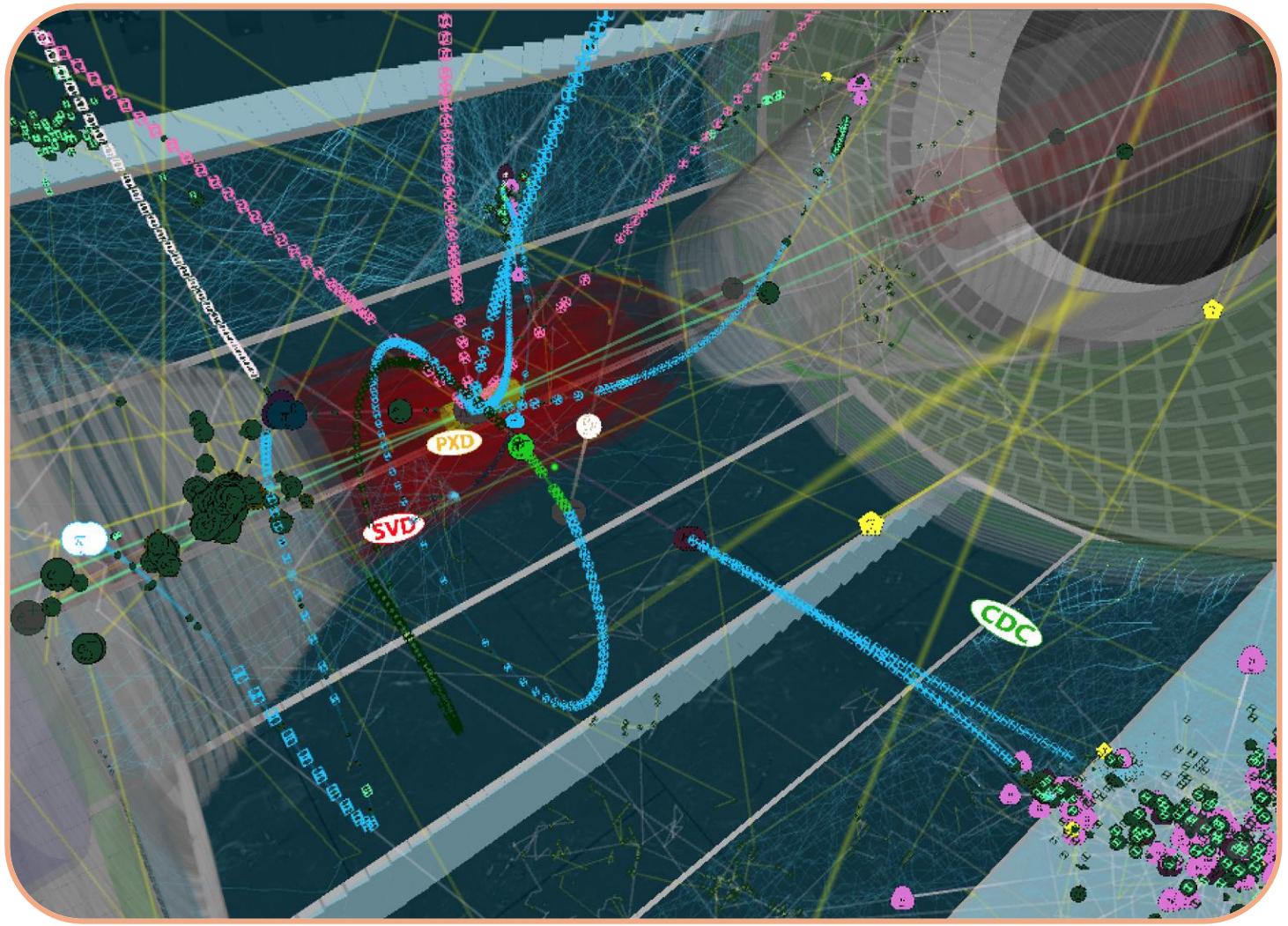
HEP projects in XR



Atlas in your pocket (CERN)
[https://atlasinyourpocket-archived.web.cern.ch/]

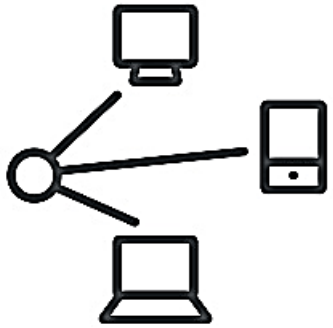


JUNO detector in the VR application (China)
[https://doi.org/10.48550/arXiv.2509.16285]

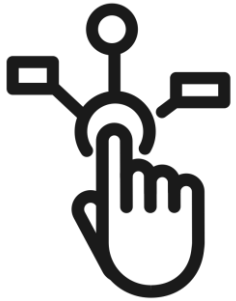


Belle II VR (KEK, Japan)
[https://doi.org/10.1109/MCG.2018.032421652]

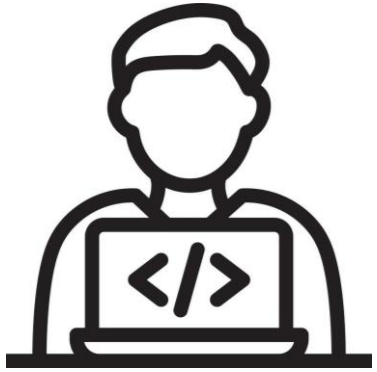
Why a game engine?



Cross-platform



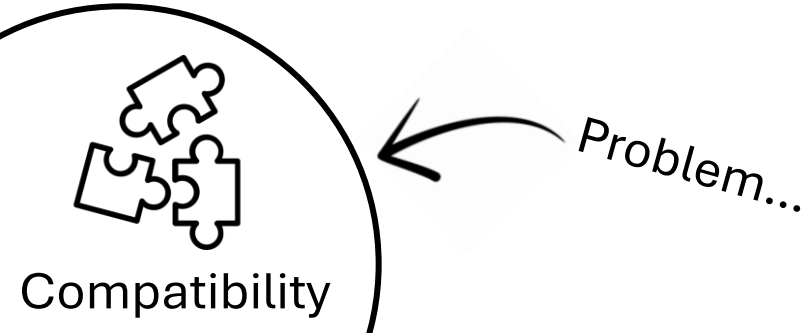
Usability



Ease of development



VR support



Unity vs Unreal Engine

[https://doi.org/10.48550/arXiv.2508.13116]

	Unity [https://docs.unity3d.com/2023.1/Documentation/Manual/system-requirements.html]	**Unreal Engine** [https://dev.epicgames.com/documentation/unreal-engine/hardware-and-software-specifications-for-unreal-engine]
Cross-platform	✓	✓
Usability	C#, beginner-friendly	Blueprints, C++
System requirements	Low (OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs)	High (Graphics RAM 8+ GB, 32 GB RAM)
Level of details	Low/Medium	High

3 ways of converting the geometry

❌ **Parser**

[https://doi.org/10.1016/j.nima.2024.169619]

- The unoptimized code
- An incorrect visualization
- Lack of documentation

❓ **ROOT → GDML → FBX**

[https://github.com/gttcz/FBXWriter]

- Problems with automation
- An unstructured hierarchy of elements

✅ **ROOT → glTF**

[https://github.com/eic/root2cad?tab=readme-ov-file]

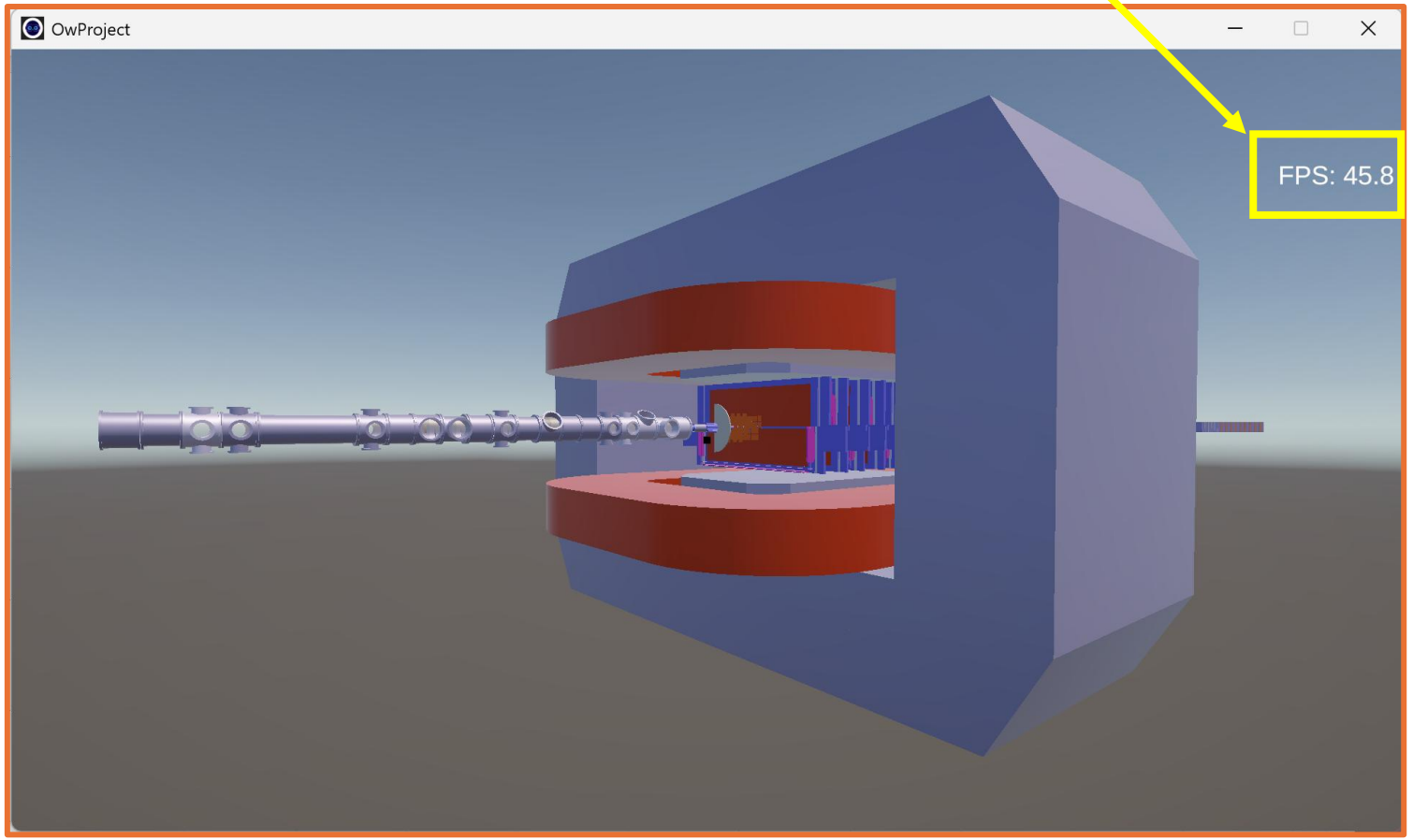
- Correct visualization
- A structured hierarchy of elements

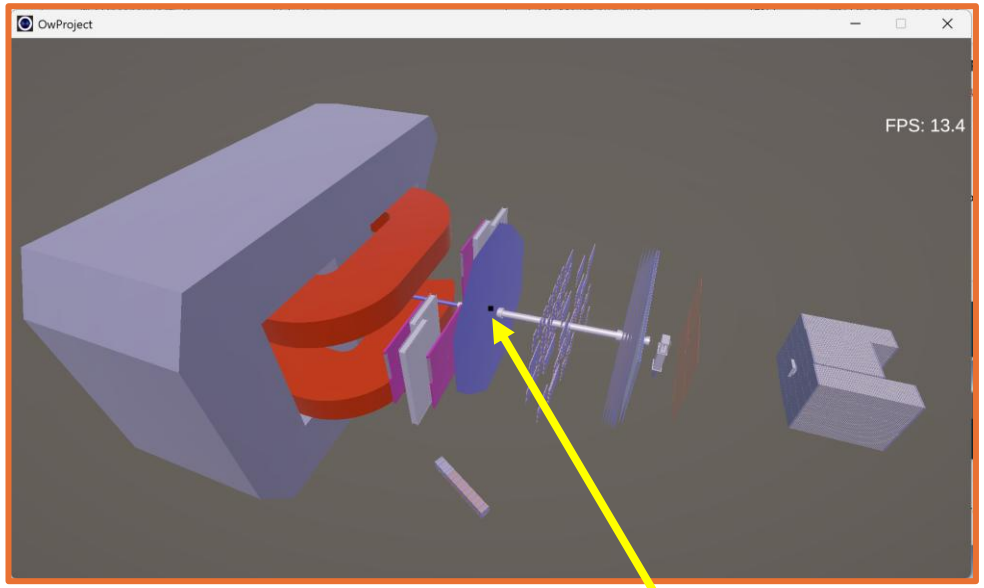
Prototype of the visualizer



The start menu

The FPS counter





The reticle

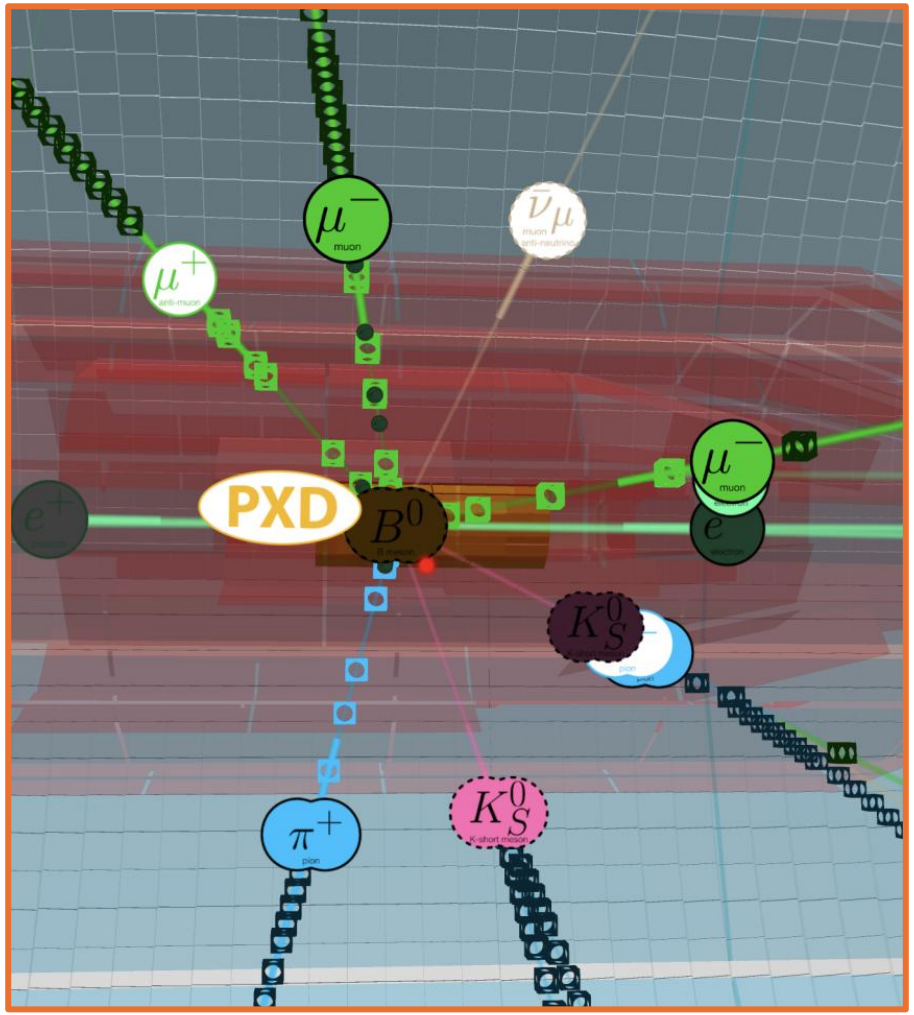
Summary

1. The conversion the detector geometry file from ROOT to glTF has been performed.

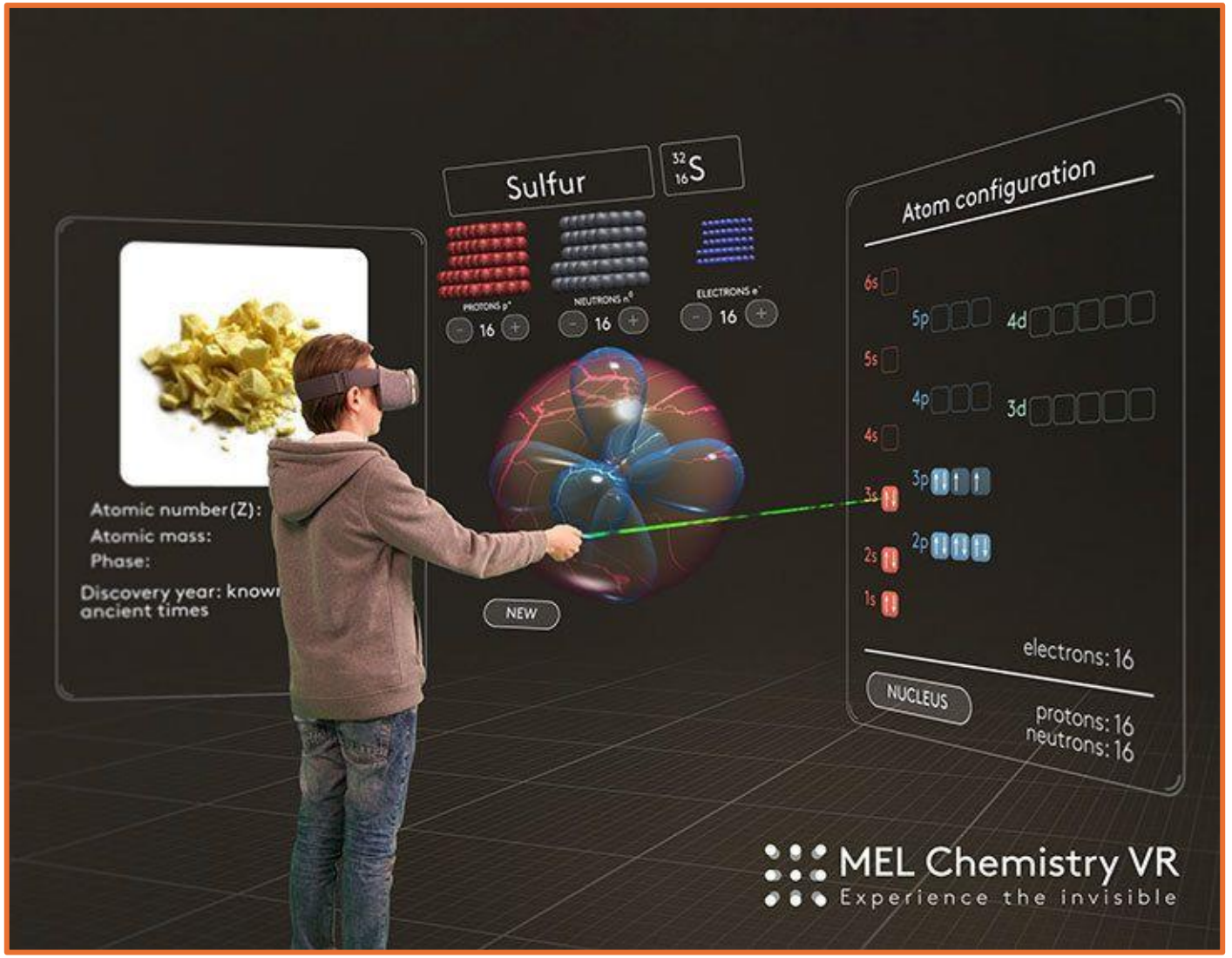
2. Unity engine has been chosen as the development environment.

3. The prototype of BM@N visualization has been developed.

Future plans



Add the track visualization



Add the VR support

Thank you for your attention!