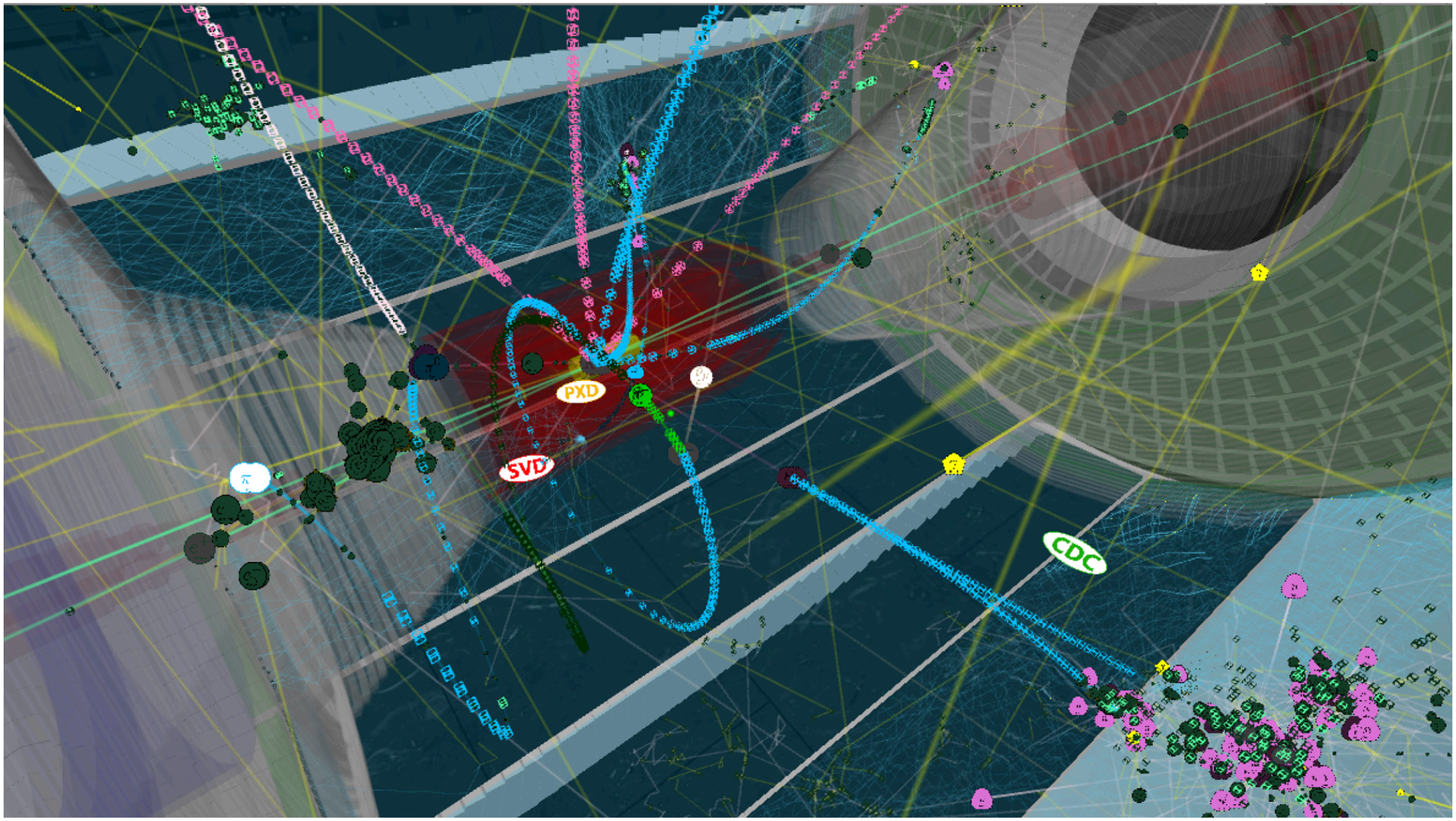
# Belle II in Virtual Reality

Leo Piilonen, Virginia Tech




*HSF Workshop March 2018*

# History

In early 2016, we submitted an internal grant proposal at Virginia Tech to develop a virtual reality model of Belle II.

ICAT SEAD grant proposal:

Select which grant: Major SEAD $25K

Project Title: An Educational Tool to Explore the Dynamics of Subatomic Physics Interactions

Team Members:
Leo Piilonen, Physics, Principal Investigator
George Glasson, School of Education
Nicholas Polys, Computer Science
Dane Webster, School of Visual Arts
Todd Ogle, TLOS
Zachary Duer, Institute for Creativity, Arts and Technology

Project Description:
The goal of this project is to develop a new immersive educational tool for experimental subatomic physics using a virtual reality (visual + sound) world in the ICAT CUBE. This tool will be used primarily for education of Physics majors but can be adapted for other audiences, including the general public. This project will be in congruence for recommendations for STEM teaching pedagogy in the Next Generation Science Standards (NGSS). The project also will serve as a valuable extension of the NSF-supported PHYSTEC project (Physics and SoE), which was designed to recruit more students into the MAED licensure program to prepare for a career in secondary school physics teaching. Two such students will participate with the team in designing and field testing the virtual learning environment.

**Funded for one year**
**Start July 1, 2016**

# Project Participants at VT (who did all the work)

Zach Duer
formerly ICAT Staff
now SOVA faculty
(lead programmer)

Tanner Upthegrove
ICAT Staff
Media Engineer

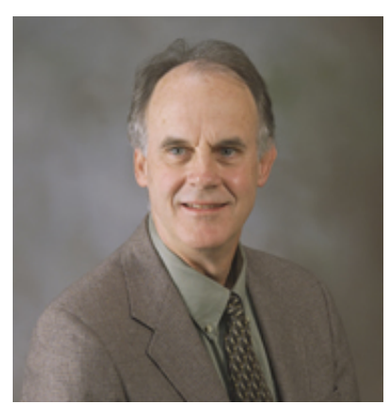
Jesse Barber
Physics Major

Samantha Spytek
Physics Major
(graduated)

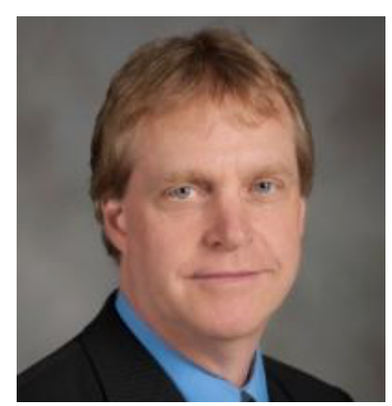
Christopher Dobson
Physics Major
(graduated)

# Project Participants (kibitzers)

Leo Piilonen
Dept of Physics



George Glasson
School of Education



Ben Knapp
ICAT Director

# Platform

**Choose Unity** (unity3d.com) as the software-development platform

- ✓ targets many 3D displays (Oculus, Vive, Cyclorama, …)
- ✓ free for non-commercial use
- ✓ Zach Duer, our programmer, is experienced in using Unity
- ✓ the associated scripts in C# look familiar to any C++ user
- ✓ Unity itself is written in C++ ➠ provides C# ↔ C++ interface

**Choose the Oculus Rift** (oculus.com) as the first display target

- ✓ robust high-performance 3D/viz support built into Unity
- ✓ VT-ICAT had two already
- ✓ can be integrated with other Rifts (“classroom”)

# Geometry (1)

**Incorporate the Belle II detector geometry** in Unity

- ✓ must be identical to our GEANT4 model (no simplifications)
- ✓ requires a method to export the geometry in a cross-platform format that can be imported into Unity
  - NOT TEve nor any other format that requires GEANT4 or ROOT to be incorporated into the Unity project
  - 3DS, OBJ, STL, PLY, etc: not supported by Unity
  - FBX (Filmbox) – modern, de facto standard for 3D-model exchange, supported directly by Unity. *But proprietary (defined by Autodesk Corp), undocumented.*
  - VRML (Virtual Reality Modeling Language) – archaic, requires an intermediary program to convert to FBX. GEANT4 already contains code to export to VRML.

# Geometry (2)

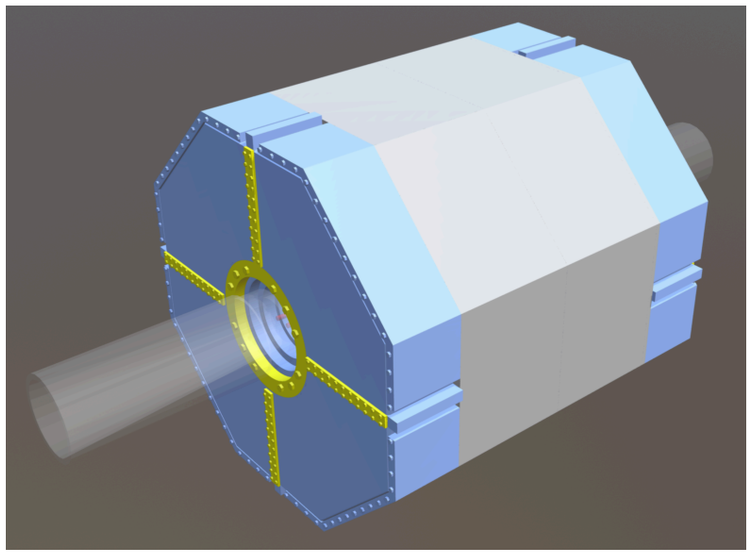
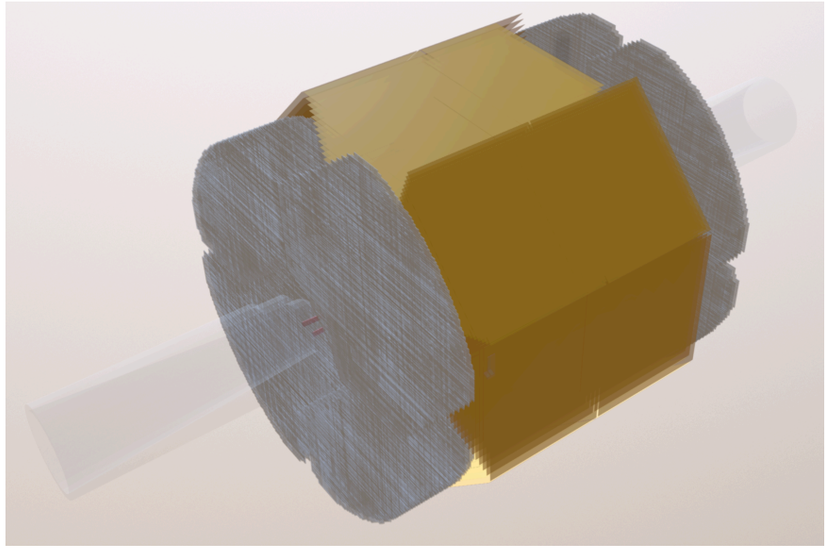
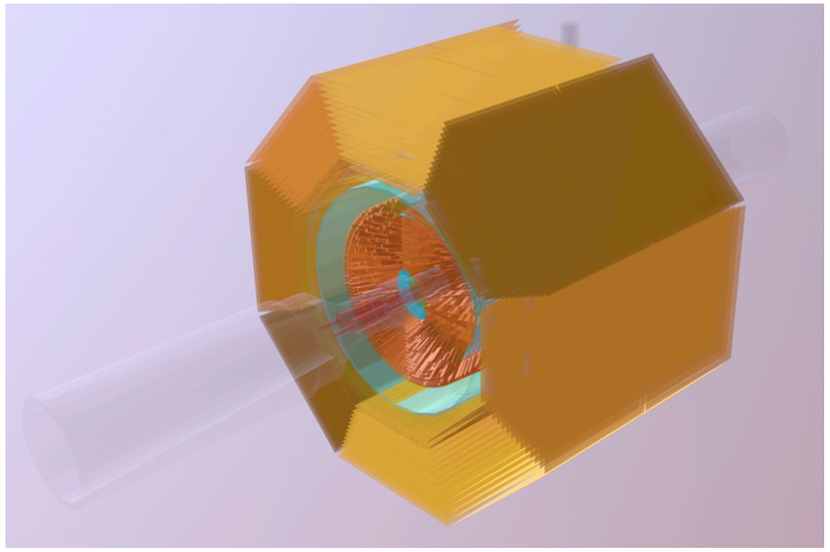
**Export the Belle II detector geometry** from basf2 framework

- ✓ all detector elements are rendered as polygons of the surface before exporting, using GEANT4's GetPolygon()
- ✓ GEANT4 accepts a UI command to write its polygonized geometry to various formats (HepRep, DAWN, VRML, VRML2) – only VRML2 would be viable here. *But this barfs on parts of our geometry, and the output file is unstructured.*
- ✓ **write two new basf2 modules to export to VRML2 or FBX**
  - geometry/modules/vrmlWriter
  - geometry/modules/fbxWriter

  } ⇢ *structured text files*
  - can export geom subset via python-steering parameter
  - examine the geometry using FBX Review, for example (www.autodesk.com/products/fbx/fbx-review)
  - Unity can import FBX files directly *(VRML2 via translator)*

# Geometry (3)

**Unity can then export the geometry to glTF™**
(https://www.khronos.org/gltf/)

https://sketchfab.com → search for **belleii**
*(can be viewed on smartphones)*

glTF™ (GL Transmission Format) is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by applications. glTF minimizes both the size of 3D assets, and the runtime processing needed to unpack and use those assets. glTF defines an extensible, common publishing format for 3D content tools and services that streamlines authoring workflows and enables interoperable use of content across the industry.

# Events (1)

**Export the events** from basf2

- ✓ must show *almost* entire event history from GEANT4
- ✓ must be in human-readable format → excel csv file
- ✓ one csv file per event
- ✓ add print line to inherited G4UserSteppingAction hook
  - write a line for each step (PreStepPoint, PostStepPoint, volumeName, trackID, parentID, PDGcode, etc)
  - *… but no heavy nuclei (they don't move)*
  - *… and cut off after 100 ns (neutron walk, late decays)*
- ✓ use perl script to add beam-line particles then sort each csv file (by ParticleName, then TrackID, then StepNumber)

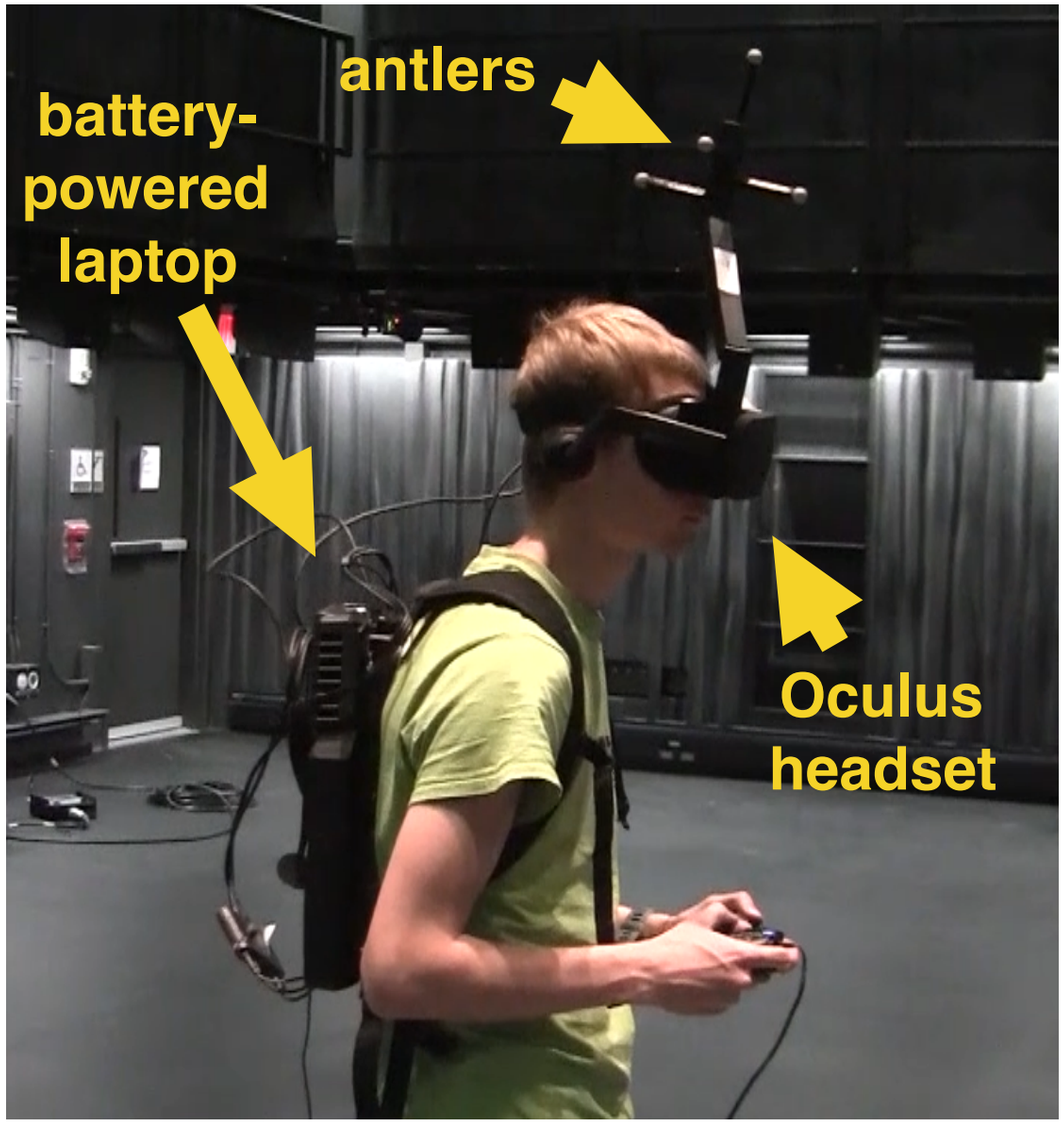
# Events (2)

**Import and display the events** into Unity

- ✓ C# scripts in Unity to read csv file, parse the data into internal Unity structures for efficient / responsive animation
- ✓ persistent faint lines to show the entire simulation history
- ✓ particle sprite to show each particle during the animation
  - colour-coded, shape-coded
  - de-emphasis (faded) when particle history ends
- ✓ dynamic trails to highlight particle motion during animation
- ✓ sensitive-detector hits, with detector-specific sound
- ✓ last few seconds of animation: show only the detector hits

# Belle2VR operation: in CUBE Facility at Virginia Tech

- ✓ In-game placement of the detector and beam line within the CUBE at Virginia Tech to accommodate $N$ students *(also avoids vertigo experienced with a context-free detector)*
- ✓ Untethered locomotion with backpack laptop + headset
- ✓ User-keyed antlers provide 3D position and orientation via CUBE's motion-capture system
- ✓ Students see each others' avatars in-game
- ✓ Can be projected onto a huge cylindrical screen ("Cyclorama") for large audiences




https://youtu.be/LxIW6Zv9uTM

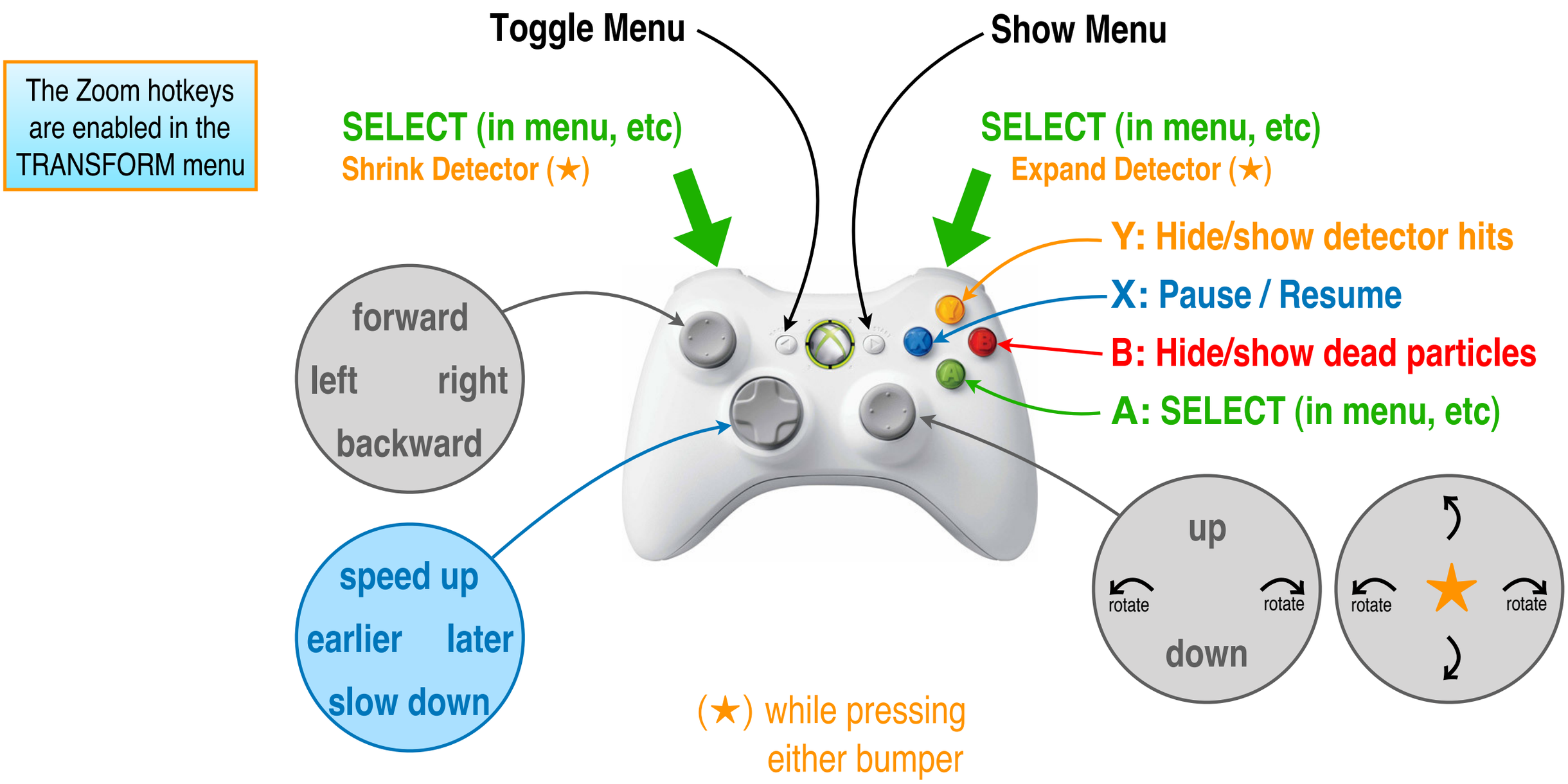
https://www.elumenati.com/projects/virginia-tech-cyclorama/

# Belle2VR operation: standalone

- ✓ In-game placement of the detector and beam line is still within the CUBE at Virginia Tech *since we don't have yet a 3D model of the Tsukuba experimental hall at KEK* 😐
- ✓ Can use Oculus Rift or HTC Vive for immersive 3D experience
- ✓ Can use computer screen for 2D projection of the VR world *(no need for 3D hardware)*
- ✓ Can display in a web browser
- ✓ User controls the animation via
  - tethered Xbox controller
  - Oculus Touch hand controllers
  - Vive hand controllers
  - keyboard/mouse

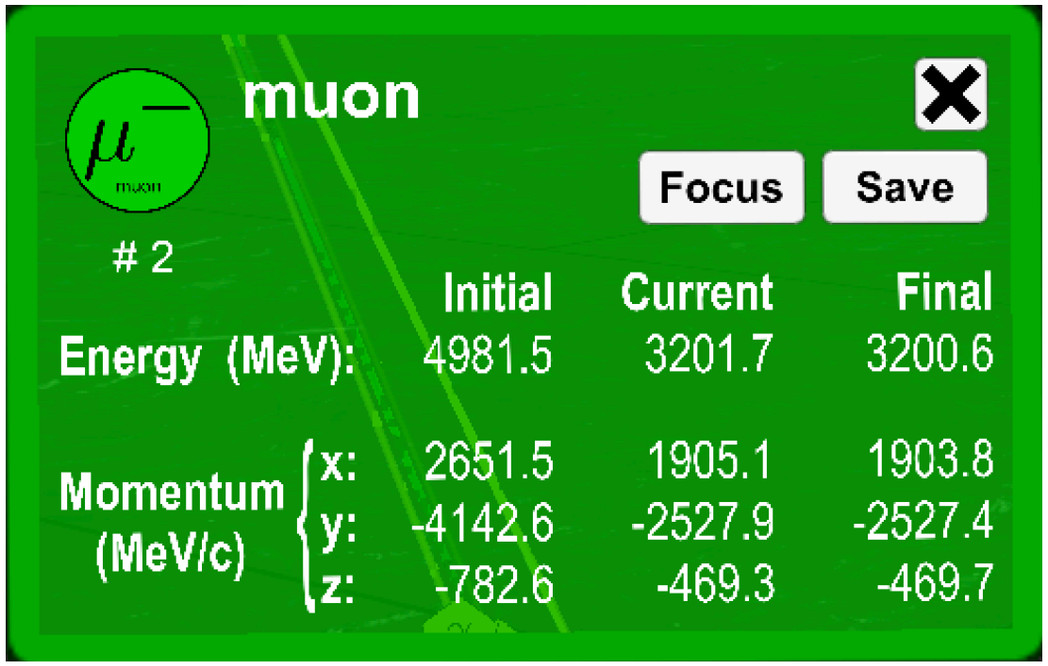
# VR world features (1)

- ✓ Your gaze is always indicated by a green dot in front of you.
- ✓ You interact with the in-world features with this gaze dot and your preferred hand controls. E.g., for Xbox One controller:



# VR world features (2)

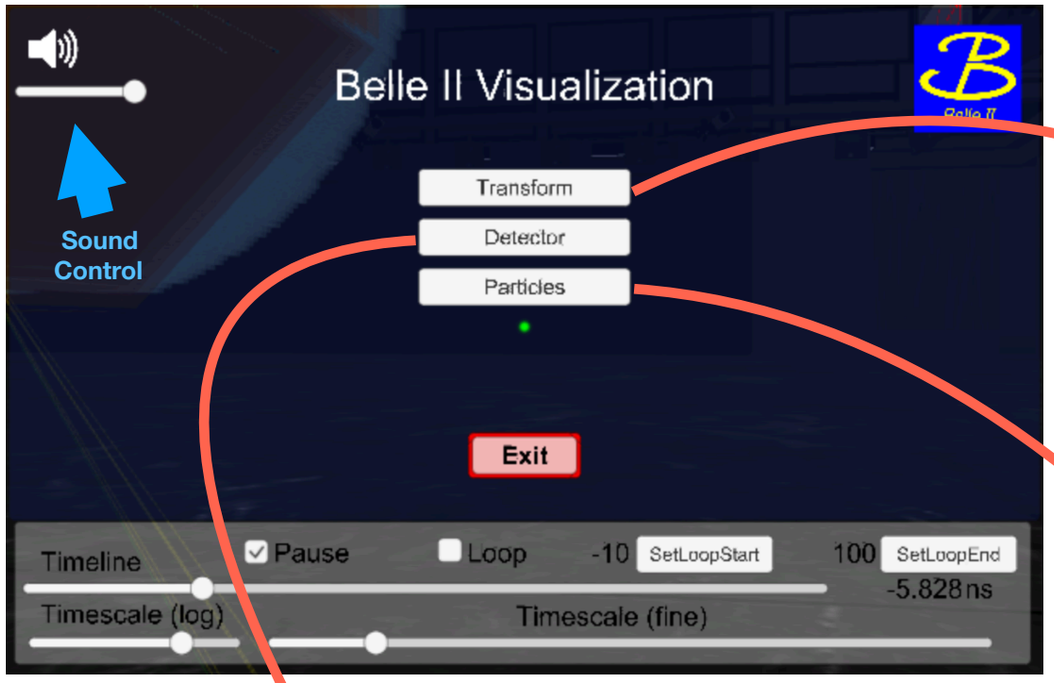
- ✓ If you gaze at a particle and **SELECT**, an information panel appears.



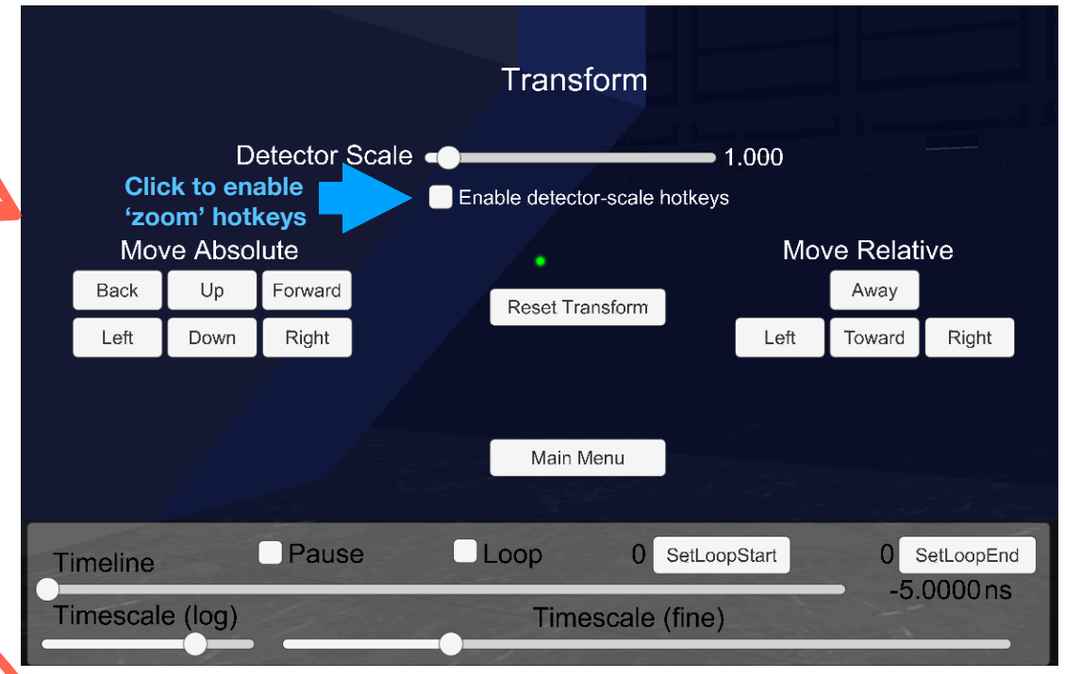

- ✓ The panel's border is black if the particle is dead.
- ✓ If you gaze at **Focus** and **SELECT**, only this particle and its relatives are shown. *(If you then open another such panel, you can "Unfocus" this chain.)*
- ✓ If you gaze at **Save** and **SELECT**, this particle's information is saved to the panel on one wall of the room.
- ✓ You can sum selected entries on the wall display panel to test conservation of energy and momentum.

# VR world features (3)

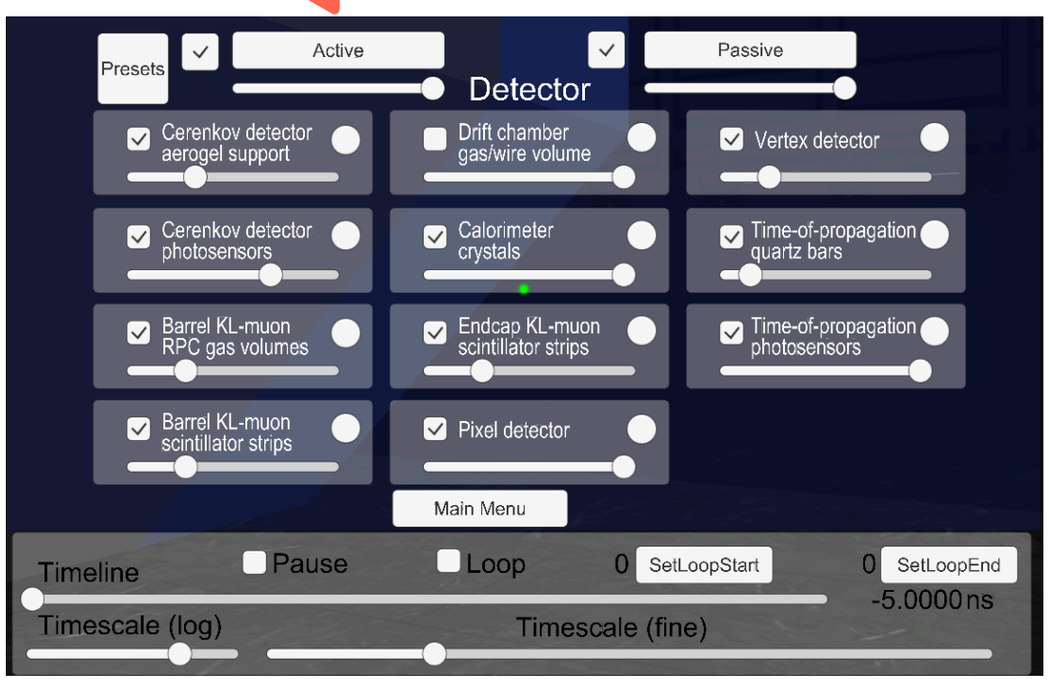
- ✓ Show the in-game menu by pressing the **Start button**.
- ✓ Move your gaze to place the green dot on an item then press **SELECT**.



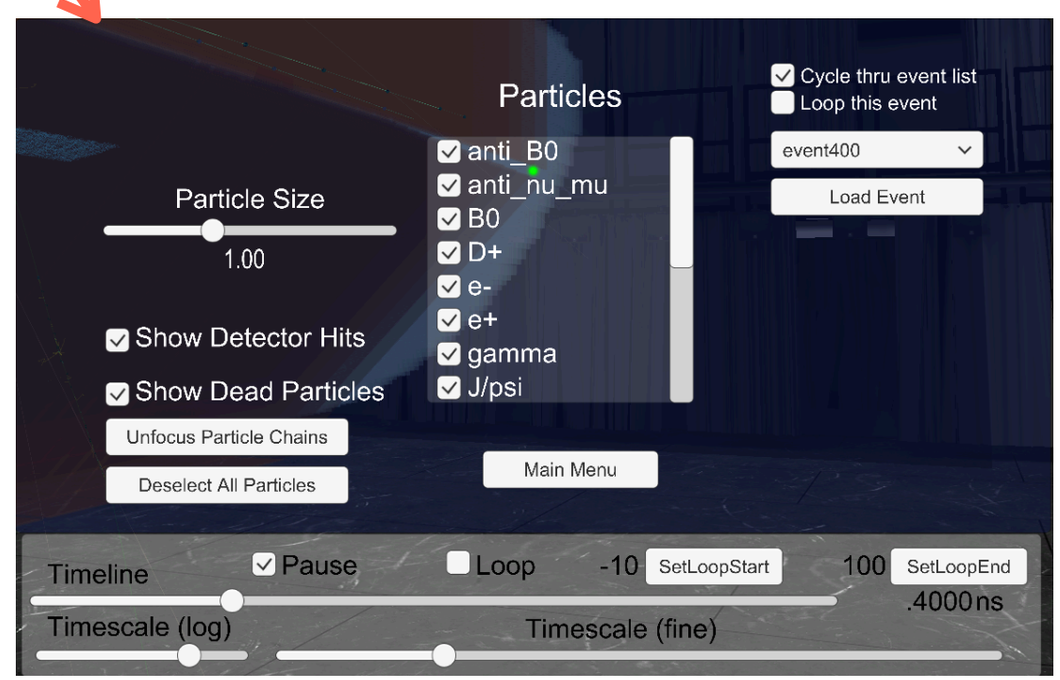

Main menu




Transformation menu



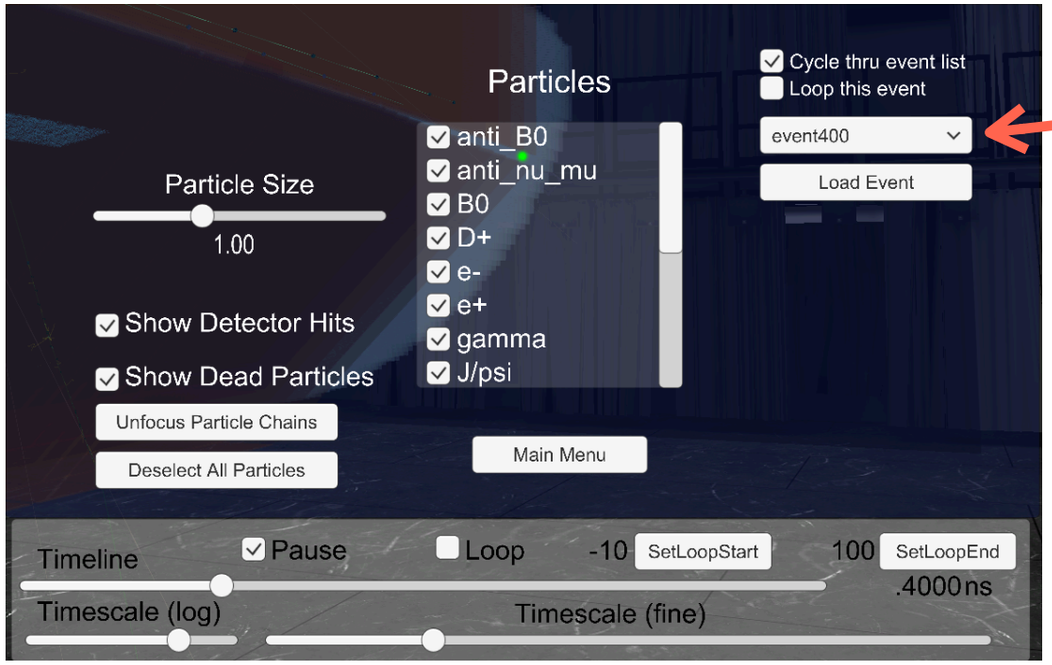

Detector hide/show menu




Particles and events menu

# VR world features (4)

- ✓ In the **Particles** menu, select one of the events to animate.
- ✓ With your gaze, scroll the list to highlight the desired event then **SELECT**.




- ✓ **Or wait:** the animation automatically skips to the next event

**Event legend:**

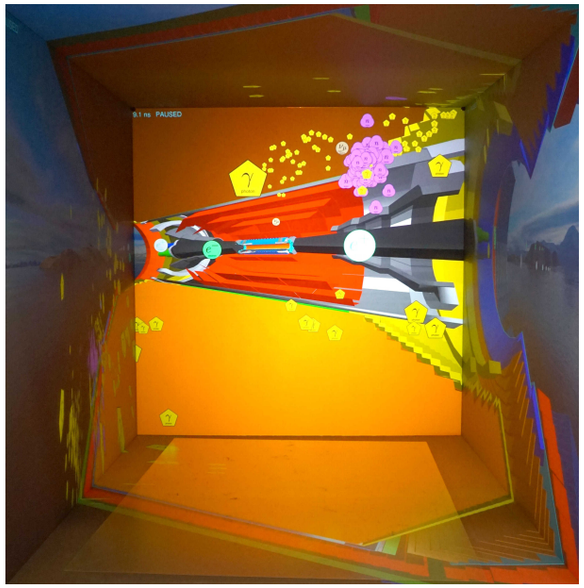
100-109: $e^+ e^- \to \mu^+ \mu^-$
110-119: $e^+ e^- \to \pi^+ \pi^-$
120-129: $e^+ e^- \to e^+ e^-$
130-139: $e^+ e^- \to \gamma \gamma$
140-149: $e^+ e^- \to K^+ K^-$
150-159: $e^+ e^- \to K_S K_L$
160-169: $e^+ e^- \to \Lambda \bar{\Lambda}$
170-179: $e^+ e^- \to p \bar{p}$
180-189: $e^+ e^- \to s \bar{s}$
190-199: $e^+ e^- \to c \bar{c}$
200-209: $e^+ e^- \to B^0 \bar{B}^0 \to (J/\psi\, K_S)(D^+ \mu^- \nu)$
210-219: $e^+ e^- \to B^+ B^- \to (\tau\, \nu)(D^0 \pi^-)$

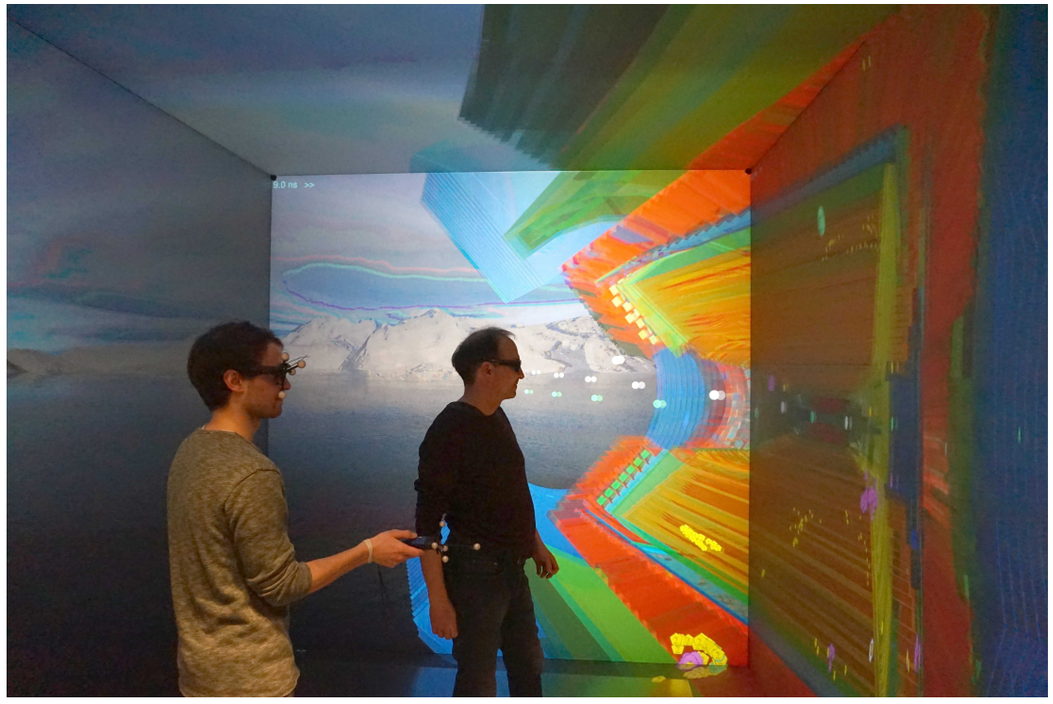
*second B is hidden* {
220-229: $e^+ e^- \to B^0 \bar{B}^0 \to (J/\psi\, K_S)(D^+ \mu^- \nu)$
230-239: $e^+ e^- \to B^+ B^- \to (\tau\, \nu)(D^0 \pi^-)$

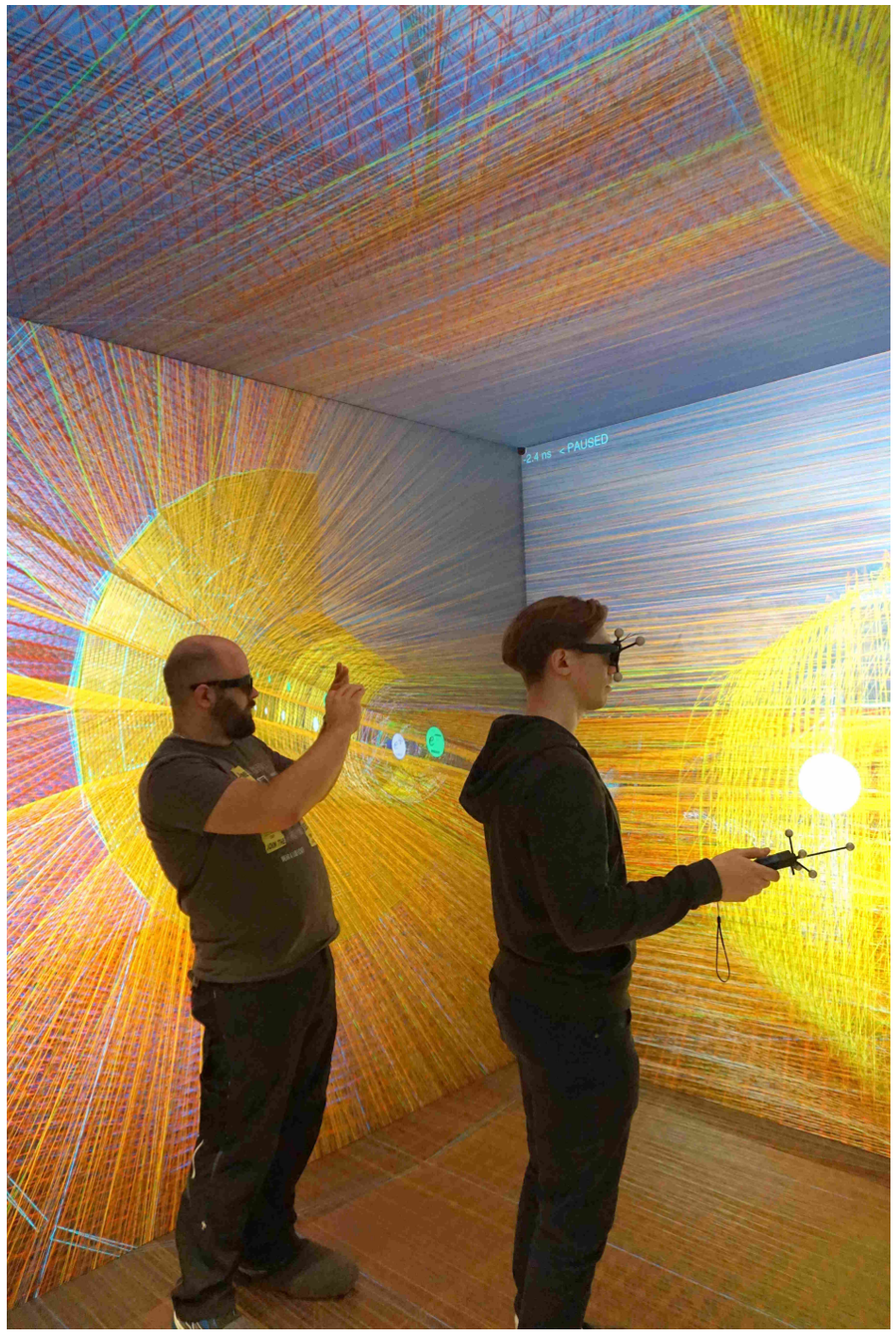
# **Belle II VR has been adapted for a CAVE environment**
by a team at Ludwig Maximilians University

Belle II GRETCHEN (II)

GRETCHEN







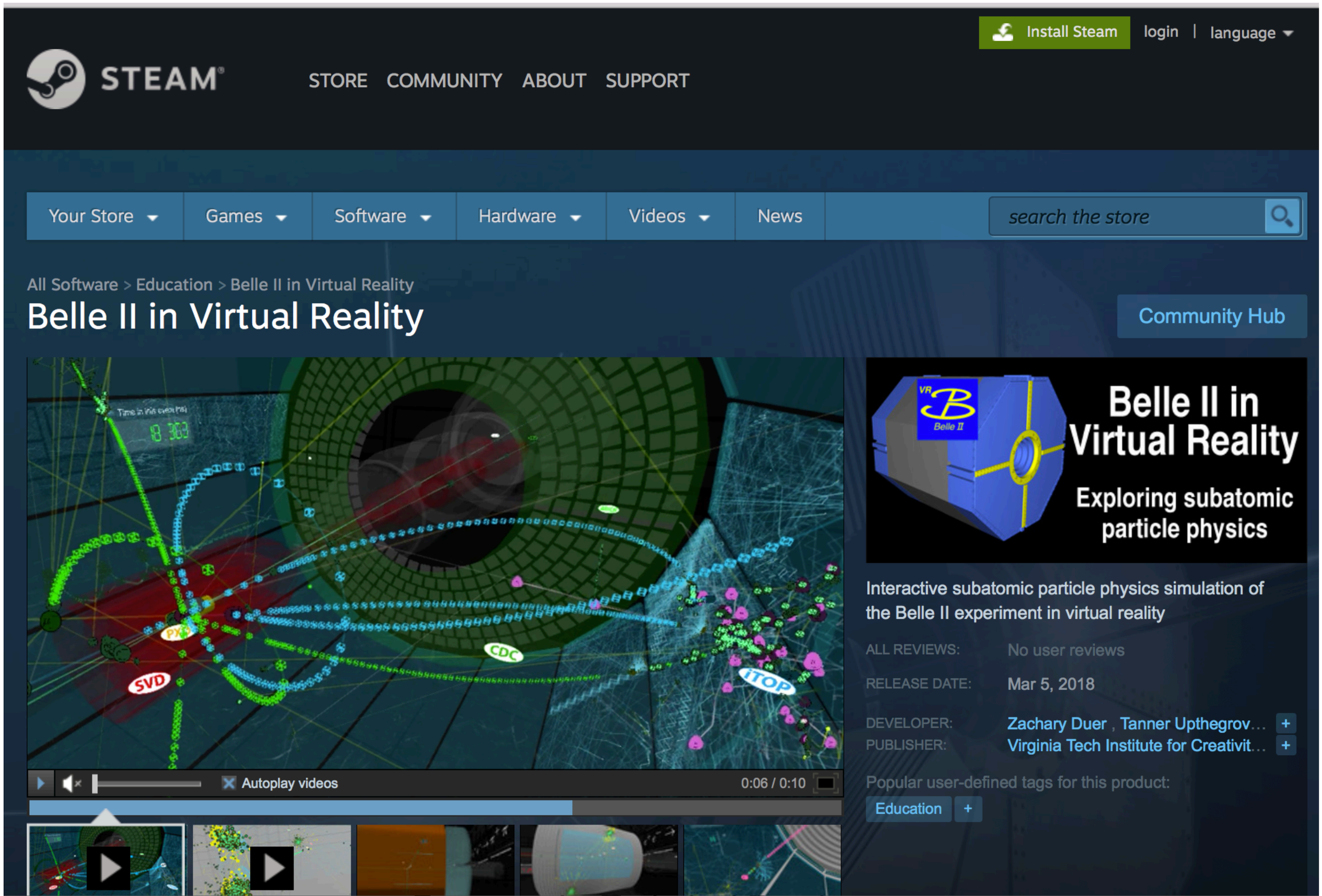
# ~35 institutions (plan to) use Belle II VR for public outreach

- KEK
- Jefferson Laboratory
- Niigata University
- Nagoya University
- National Taiwan University
- Fu Jen Catholic University
- Josef Stefan Institute
- University of Ljubljana
- Karlsruhe Institute of Technology
- Ludwig Maximilians University
- University of Strasbourg / IPHC
- Universidad Autonoma de Sinaloa
- University of Pisa / INFN
- University of Padua / INFN
- University of Roma 3 / INFN
- University of Frascati / INFN
- University of Trieste / INFN
- University of Perugia / INFN
- University of Hawaii
- University of Cincinnati
- Luther College
- …

at KEK (June 2017)

# App is available for free on Steam

store.steampowered.com/app/810020/



# Going forward

- ✓ Seeking new funding to continue this development, particularly as a pedagogical tool in undergraduate physics (university) and high school science education

For more information: www.phys.vt.edu/~piilonen/VR/

Thank you for your attention!

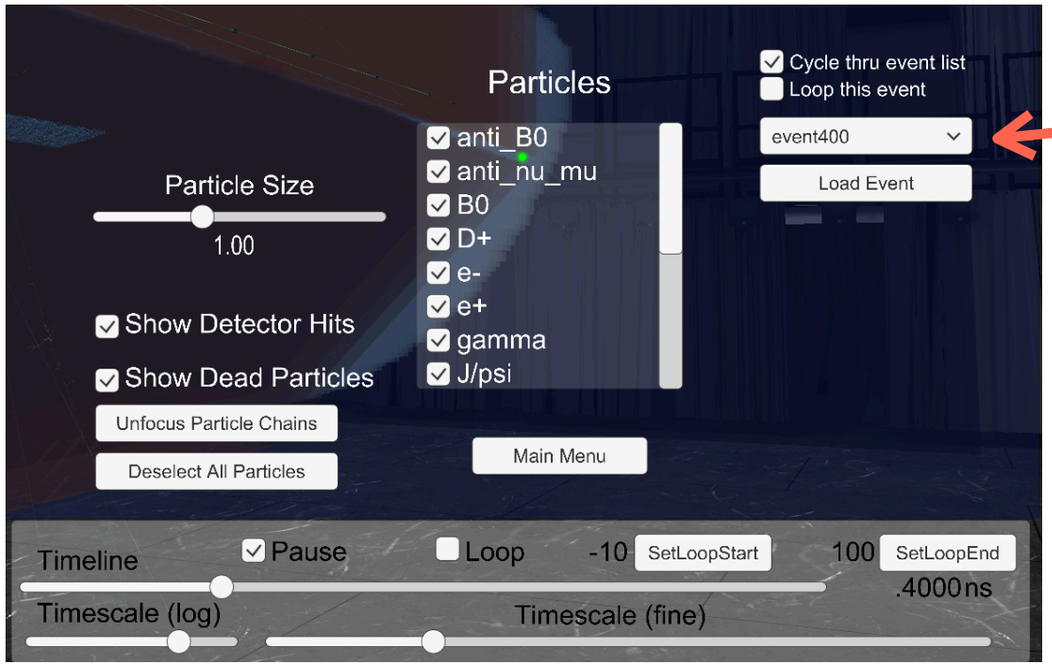
# Backup

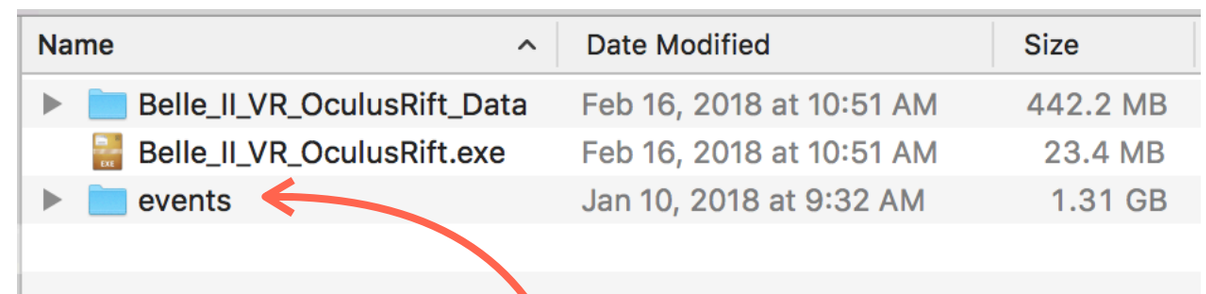
# Development Process

- ✓ Development has been documented in two movies:
  - vimeo.com/220004044 *(narrated)*
  - vimeo.com/214899668 *(captioned, no sound)*
- ✓ Public displays of work-in-progress:
  - Virginia Science Festival (10/2016)
  - ICAT Day at Virginia Tech (5/2017)
  - Belle II General Meeting (6/2017)
- ✓ Presentation/paper at the IEEE Visualization in Practice conference (10/2017, Phoenix, Arizona)
  - www.visinpractice.rwth-aachen.de
- ✓ Presentation at the Virginia Association of Science Teachers conference (11/2017, Roanoke, Virginia)
  - www.vast.org

# VR world features (5)

✓ You can customize which events appear in the **Particles** menu.



| Name | Date Modified | Size |
|---|---|---|
| Belle_II_VR_OculusRift_Data | Feb 16, 2018 at 10:51 AM | 442.2 MB |
| Belle_II_VR_OculusRift.exe | Feb 16, 2018 at 10:51 AM | 23.4 MB |
| events | Jan 10, 2018 at 9:32 AM | 1.31 GB |

The simulated electron-positron collision events are stored in a folder named **events** at the same level as the app itself. You may modify the contents of this folder to suit your needs, according to the following rules.

The **events.lis** text file in the **events** folder specifies the sequence of event files that are fetched by the app. In your favorite text editor, you may comment out and/or reorder lines here to suit your preferences. Each event file is a plain text file that is exported from the basf2 GEANT4-based physics simulation of the Belle II detector.

If the **events** folder contains the text file **events.url** and this file specifies a valid web address (URL) then **events.lis** and the event csv files will be fetched from this web address instead of the **events** folder.

If the above event-fetching mechanisms fail (due to syntax or file-corruption errors), the app will revert to displaying a baked-in event: $e^+ e^- \rightarrow B^0 \bar{B}^0 \rightarrow (J/\psi K_S)(D^+ \mu^- \nu)$.