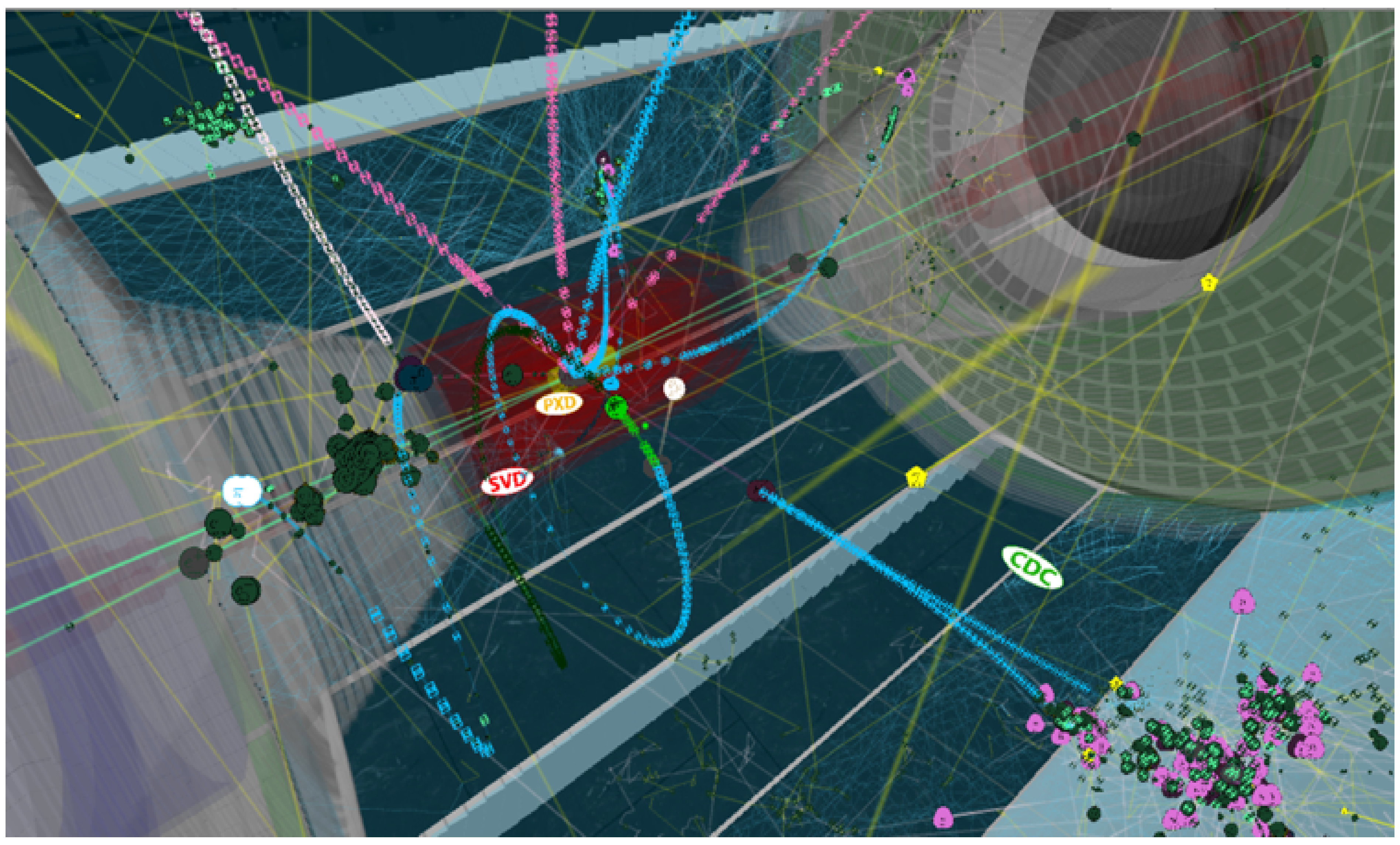
Belle II VR: A Virtual Reality Visualization of Subatomic Particle Physics

Leo Piilonen, Virginia Tech (for the Belle II Collaboration's Outreach Group)

#36
Track 2
Offline computing

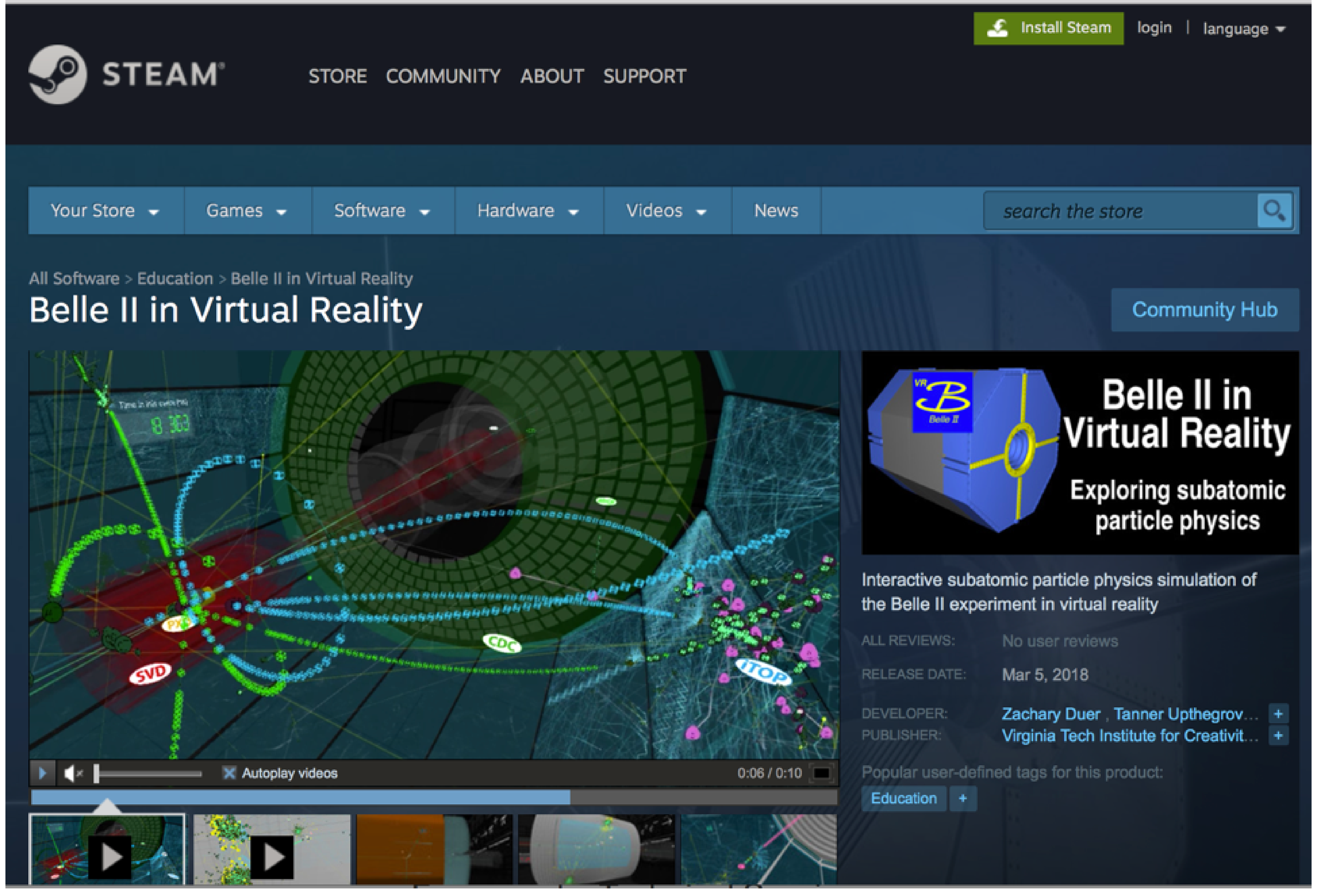
Interactive virtual reality 3D visualization of event simulations in the Belle II detector at the SuperKEKB colliding-beam $e^+ e^-$ accelerator in Tsukuba, Japan.

www.phys.vt.edu/~piilonen/VR/



Now available for free on the Steam app store.

store.steampowered.com/app/810020/



All Software > Education > Belle II in Virtual Reality

Belle II in Virtual Reality

Belle II in Virtual Reality
Exploring subatomic particle physics

Interactive subatomic particle physics simulation of the Belle II experiment in virtual reality

ALL REVIEWS: No user reviews
RELEASE DATE: Mar 5, 2018
DEVELOPER: Zachary Duer , Tanner Upthegrov…
PUBLISHER: Virginia Tech Institute for Creativit…

Popular user-defined tags for this product:
Education +

Developed at Virginia Tech's Institute for Creativity, Arts & Technology

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)

Leo Piilonen
Dept of Physics

George Glasson
School of Education

Ben Knapp
ICAT Director

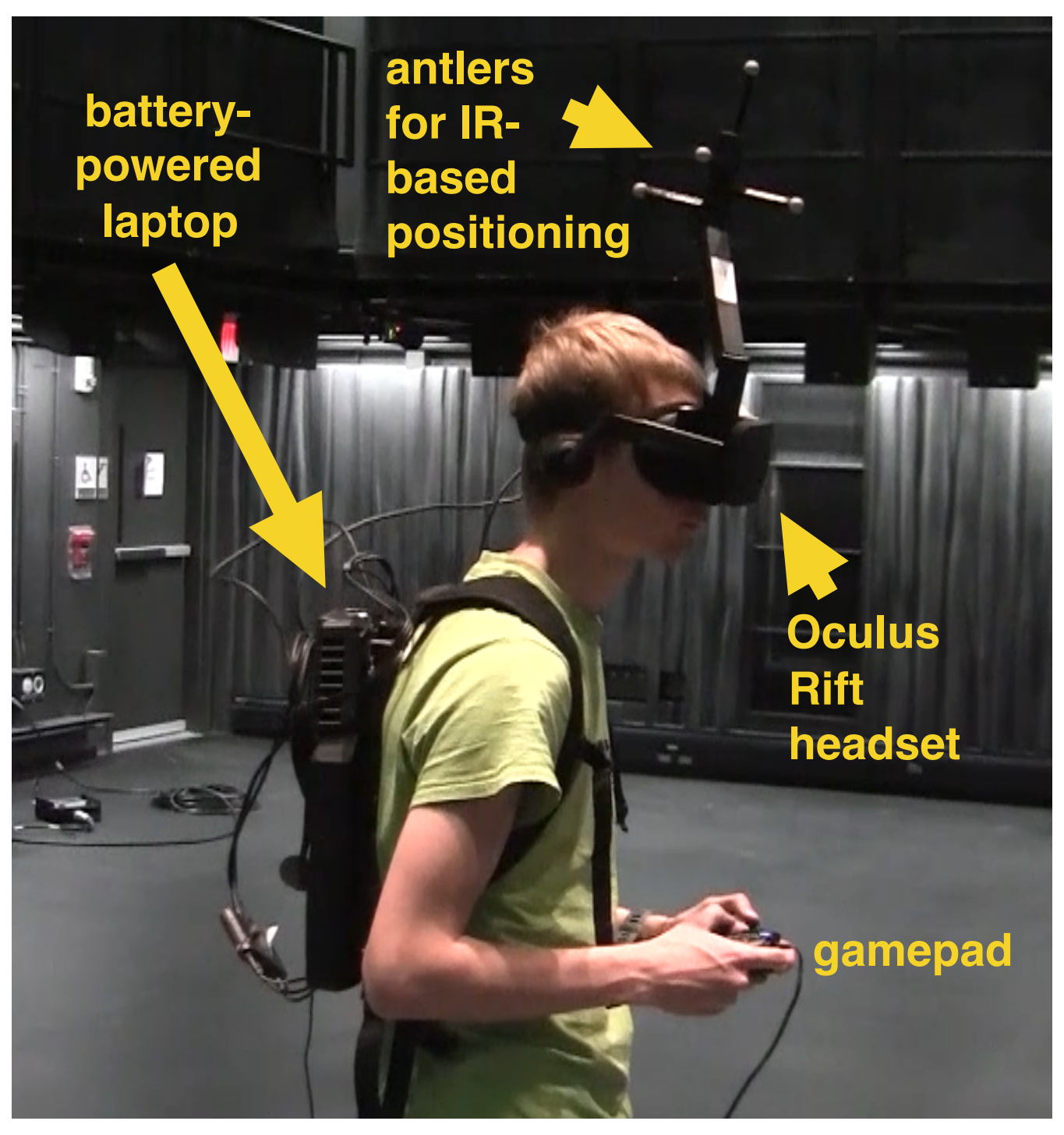
Unity (unity3d.com) is the software-development platform

- ✓ targets many 3D displays (Oculus, HTC Vive, Cyclorama, …)
- ✓ the associated C# scripts look familiar to any C++ programmer
- ✓ Unity itself is written in C++ → provides C# ↔ C++ interface

Oculus Rift (oculus.com) was the first display target

- ✓ robust high-performance 3D/viz support built into Unity
- ✓ multiple Rifts can be integrated (for "classroom" use)

… now supports HTC Vive, Oculus GO, WebGL, 2D displays, +++



(1) Operation in the CUBE Facility at Virginia Tech

- ✓ In-game placement of detector within the CUBE at Virginia Tech accommodates N students
  - ◉ *avoids vertigo that was experienced with a context-free detector*
- ✓ Untethered locomotion with battery-powrered backpack laptop + headset
- ✓ User-specific antlers provide 3D position + orientation via CUBE's IR motion-capture system
- ✓ Students see each others' avatars in-game *(collision avoidance!)*
- ✓ Can be projected onto a huge cylindrical screen ("Cyclorama") in CUBE for large audiences
  - ◉ youtu.be/LxIW6Zv9uTM and www.elumenati.com/projects/virginia-tech-cyclorama/

(2) Standalone operation

- ✓ Can use Oculus Rift or HTC Vive or Oculus GO 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 (uses WebGL Javascript API)
- ✓ User controls the animation via
  - ◉ tethered or Bluetooth gamepad
  - ◉ Oculus Touch hand controllers
  - ◉ Vive hand controllers
  - ◉ keyboard/mouse

Rift

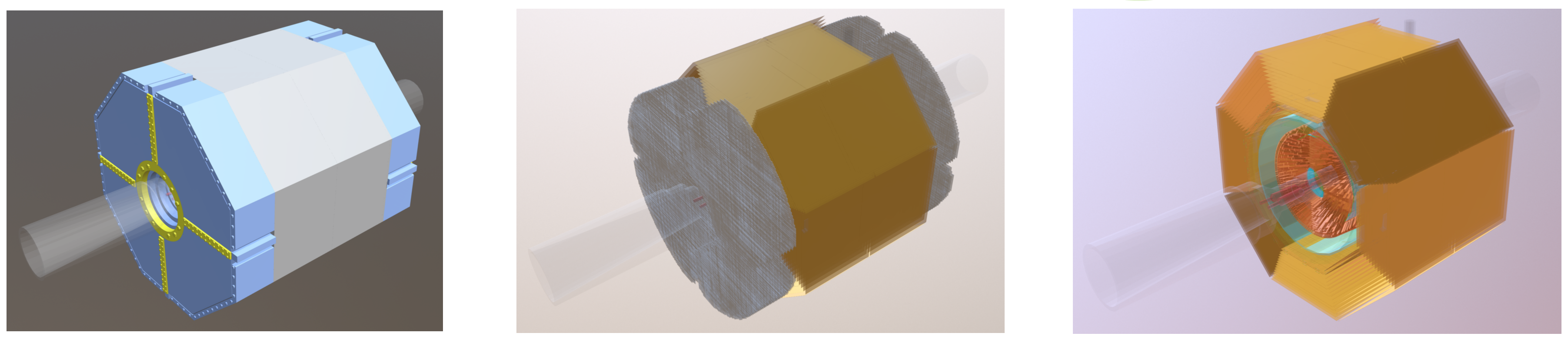
Vive

GO

(1) Incorporate the Belle II detector geometry in Unity

- ✓ must be identical to the Belle II detector's GEANT4 model used in simulation and reconstruction
- ✓ requires a method to export the geometry in a cross-platform format that can be imported into Unity
  - ◉ FBX (Filmbox) – modern, de facto standard for 3D-model exchange, supported directly by Unity. *Proprietary format (defined by Autodesk Corp), undocumented.*
  - ◉ VRML (Virtual Reality Modeling Language) – requires translator to convert to FBX for Unity import.
- ✓ write two new basf2 modules to export the Belle II detector geometry to VRML2 or FBX
  - ◉ geometry/modules/vrmlWriter
  - ◉ geometry/modules/fbxWriter

  } *create structured text files (human-readable)*
  - ◉ user can export a subset of the geometry via python-steering parameter
  - ◉ examine the geometry using FBX Review (www.autodesk.com/products/fbx/fbx-review)
  - ◉ you can get copies of these exporters (C++ source code) from github.com/HSF/Visualization
- ✓ Unity can then export the geometry to glTF™ (www.khronos.org/gltf/)



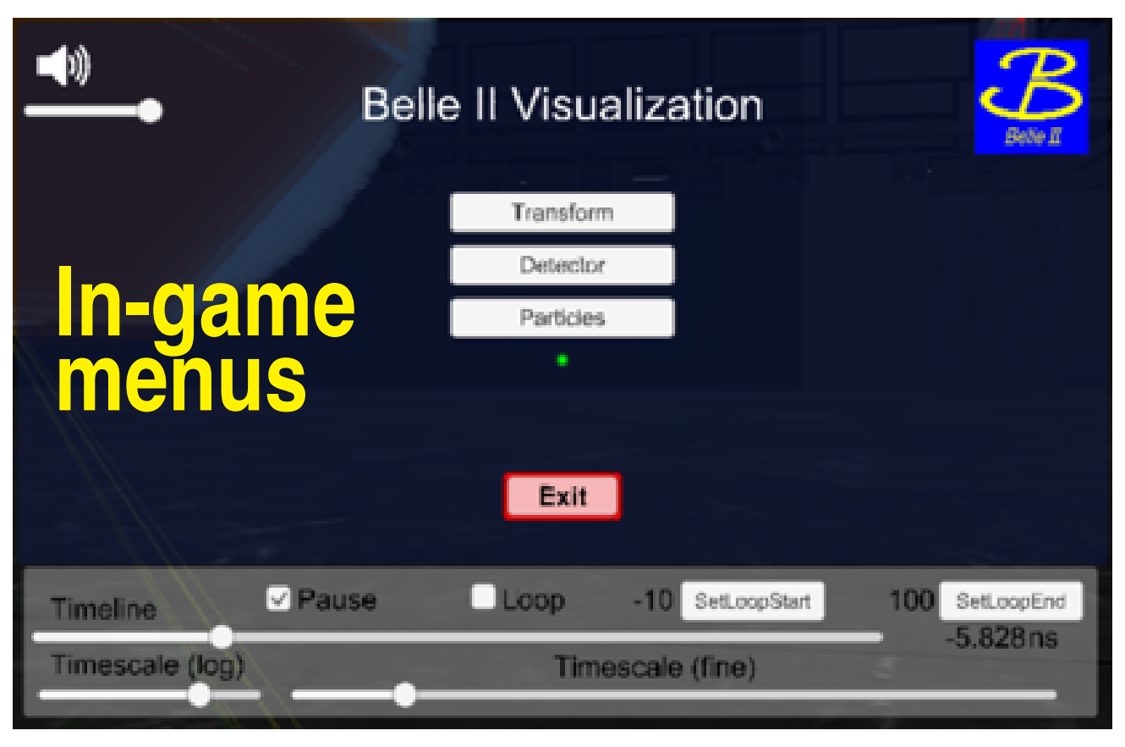
sketchfab.com → search for **belleii** *(can be viewed on smartphones)*

(2) Export the events from basf2 for import into Unity

- ✓ must show *almost* entire event history from GEANT4 *(excludes heavy nuclei; clips at ~100 ns)*
- ✓ must be in human-readable format → *Excel csv file, one csv file per event*
- ✓ add print line to inherited G4UserSteppingAction hook
  - ◉ one line per step (PreStepPoint, PostStepPoint, volumeName, trackID, parentID, PDGcode, etc)
- ✓ add beam-line particles and sort each csv file (by ParticleName, then TrackID, then StepNumber)

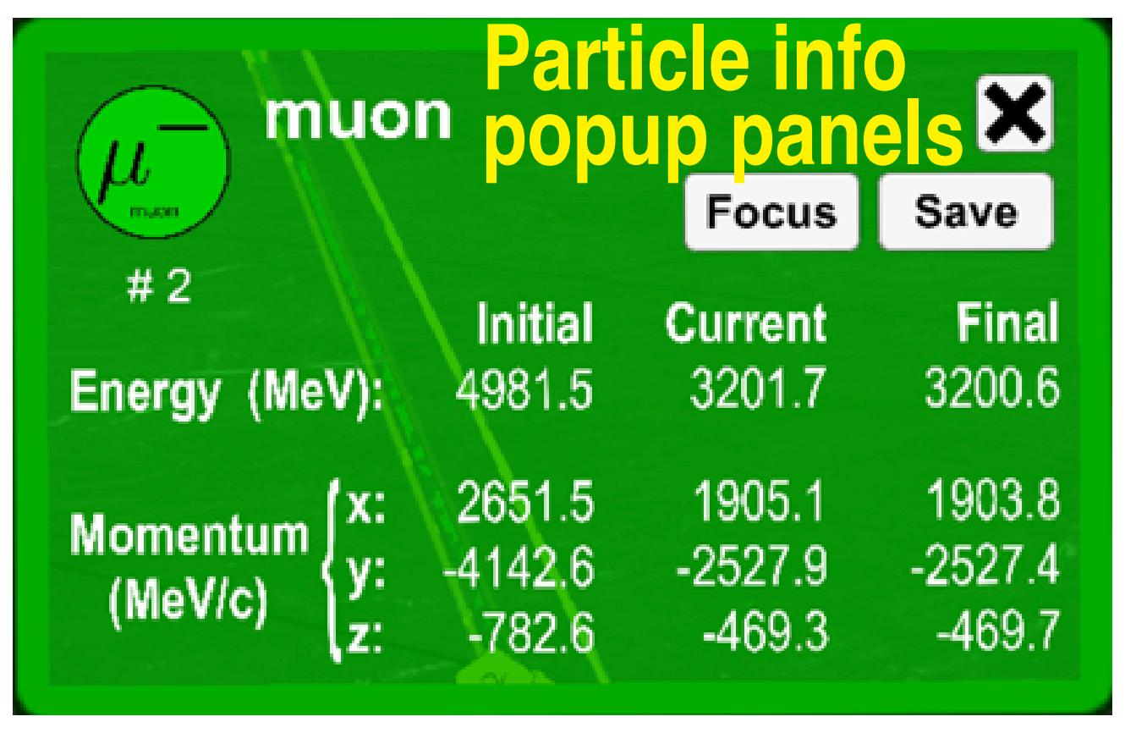
(3) Import and display the events into Unity

- ✓ C# scripts read csv file and populate internal Unity structures for efficient animation
- ✓ persistent faint lines show the entire simulation history
- ✓ moving sprite shows each particle during the event animation
  - ◉ colour-coded, shape-coded; faded when the particle's history ends
- ✓ dynamic trails behind sprite highlight the particle motion during animation
- ✓ sensitive-detector hits, with detector-specific sound as each hit occurs in animation
- ✓ for the last few seconds of event animation, show only the detector hits



Belle II Visualization

Timeline ☑ Pause □ Loop -10 SetLoopStart 100 SetLoopEnd
-5.828ns
Timescale (log) Timescale (fine)



muon
μ^-
2

		Initial	Current	Final
Energy (MeV):		4981.5	3201.7	3200.6
Momentum (MeV/c)	x:	2651.5	1905.1	1903.8
	y:	-4142.6	-2527.9	-2527.4
	z:	-782.6	-469.3	-469.7

Belle II Virtual Reality has been adapted for the CAVE environment
See Michael Bender's talk on Thursday afternoon (Track T2 - Offline Computing)