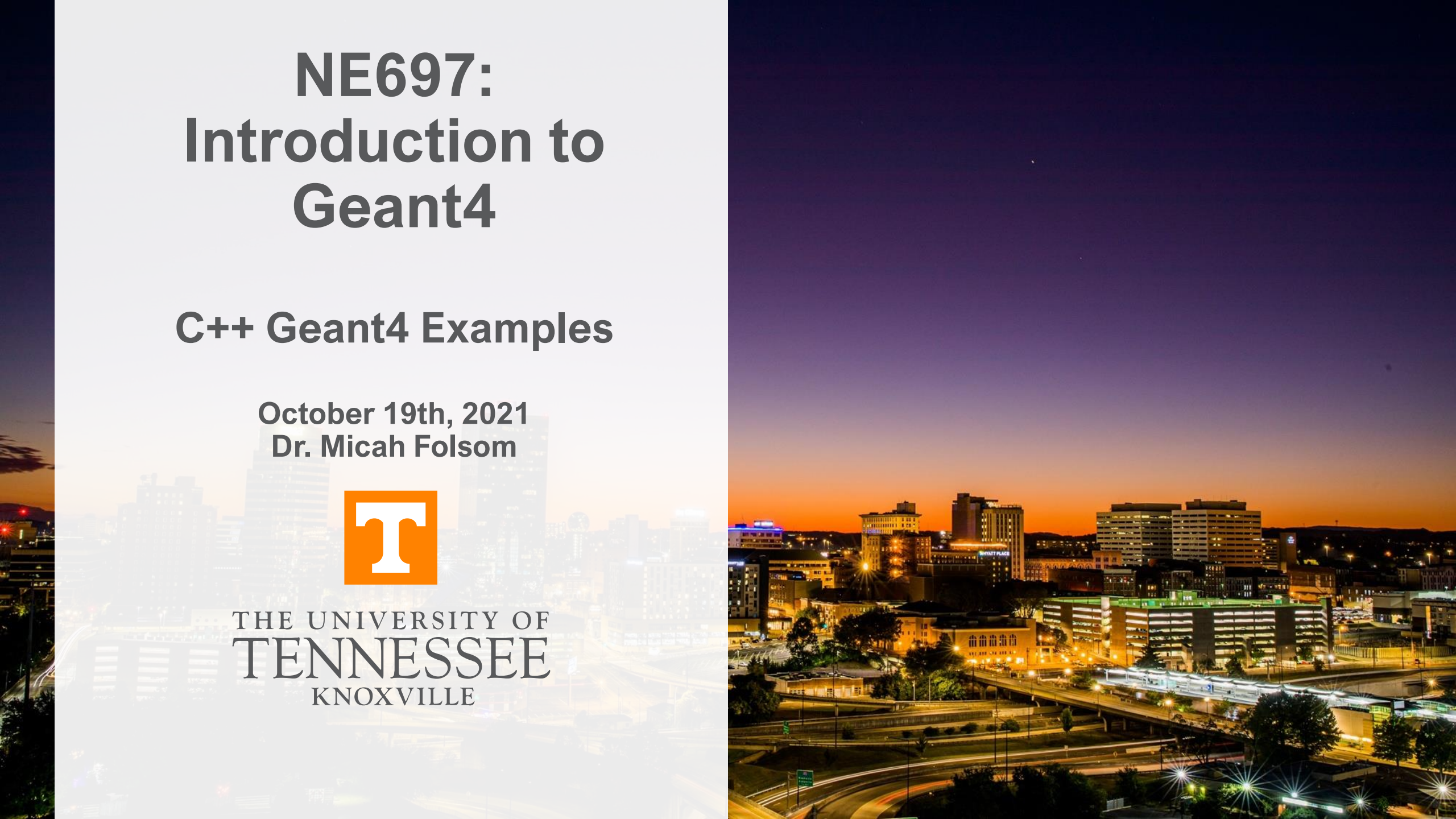
NE697: Introduction to Geant4

C++ Geant4 Examples

October 19th, 2021

Dr. Micah Folsom



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Today's Agenda

- Finish Virtualbox demo
- Quick Geant4 application anatomy review
- Geant4 examples

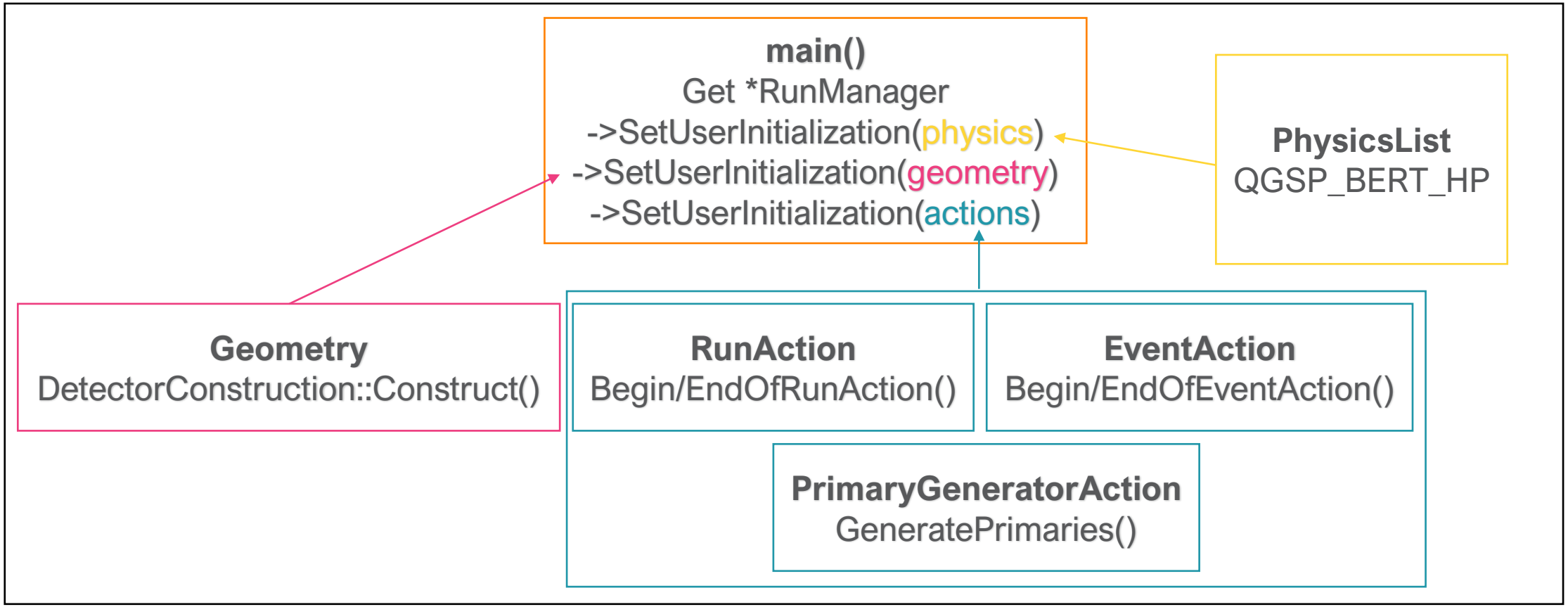
Compiling Geant4: Resources

- Official: https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/
- My coworker: https://github.com/dhellfeld/install_scripts
  - geant4_install.sh
  - Need to update version, turn multithreading and Qt viz to ON
- There is a lot of info on Google, but some is outdated
  - Pay attention to Geant4 version (10.6+), compiler version (gcc8+)
  - Official docs (and source code) are the most authoritative source of info
  - Significant G4RunManager change from 10.6→10.7

Geant4: Reminders

- All we're doing is "shooting" particles around a geometry we define according to a set of physics
  - Source characteristics are up to us!
- We have full access to all of the physics information
- What we collect is up to us!
  - Typically looking for energy depositions (this is where the fun is)
  - Typically collecting position, time, how much energy deposited, what physics process occurred, # of secondaries generated
- It's C++, so there are many pathways to many solutions

Geant4 Core Program Anatomy



Geant4 Geometry

- Capable of building very complicated geometries
  - Parameterized shapes, things with many identical elements (detector pixels)
- We're going to stick with the built-in primitives (Constructed Solid Geometry)
  - https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/Detector/Geometry/geomSolids.html
- Volumes are nested like MCNP
  - Solid stainless-steel cylinder, then a "solid" air cylinder **inside** of it → air canister
  - Better than defining a hollow cylinder (think about the boundaries)

Geant4 Geometry

- Split into 3 components; build in this order
  - Shape (solid): G4Box, G4Sphere, G4Cons, etc
  - Logical (material): **G4LogicalVolume**, takes pointer to Shape
  - Physical placement (position, rotation, nesting, copying): **G4PVPlacement**, takes pointer to G4LogicalVolume
- Every geometric item must be somewhere inside the outermost volume (world, experimental hall)
  - Returned by DetectorConstruction::Construct()
- Also must define materials (pre-builts in G4NISTManager)
  - G4NISTManager::FindOrBuildMaterial("G4_GALACTIC")

Lab Time

- Geant4 examples
  - B1: dose calculations
    - Modify B1 to output periodic updates of the event number being processed
      - BeginOfEventAction() or EndOfEventAction()
        1. Get the G4RunManager, 2. Get the current Run object, 3. Get the total # of events
    - Create .mac files that run with 100,000 particles for these source configurations
      - Gamma, 511 keV
      - Neutron, 2 MeV
      - e-, 1 MeV
    - Change the size of the phantom objects, recompile and confirm they changed
    - Modify the geometry to use a G4Sphere instead of a G4Cons (use a similar size)

Lab Time

- Geant4 examples
  - B1: dose calculations
    - Add another component to the analysis: eDep^3

Lab Time

- Geant4 examples
  - B3b: PET scanner system
    - Uses a hook we haven't discussed: Stacking Action
      - Check out B3StackingAction
      - ClassifyNewTrack() triggered when a G4Track is created
      - Gives the opportunity to kill particles we don't care about (secondaries)
    - Run with 100,000 particles, and count the "Nb of good e+ annihilations"
    - Change the material to NaI instead of Lu2SiO5
    - Run again with 100,000 particles – how does it compare?
    - Check out B3bRun::RecordEvent
      - What is the energy threshold?
    - Change the source to N-13 with the /gun/particle command (see run2.mac for help) and run again with 100,000 particles