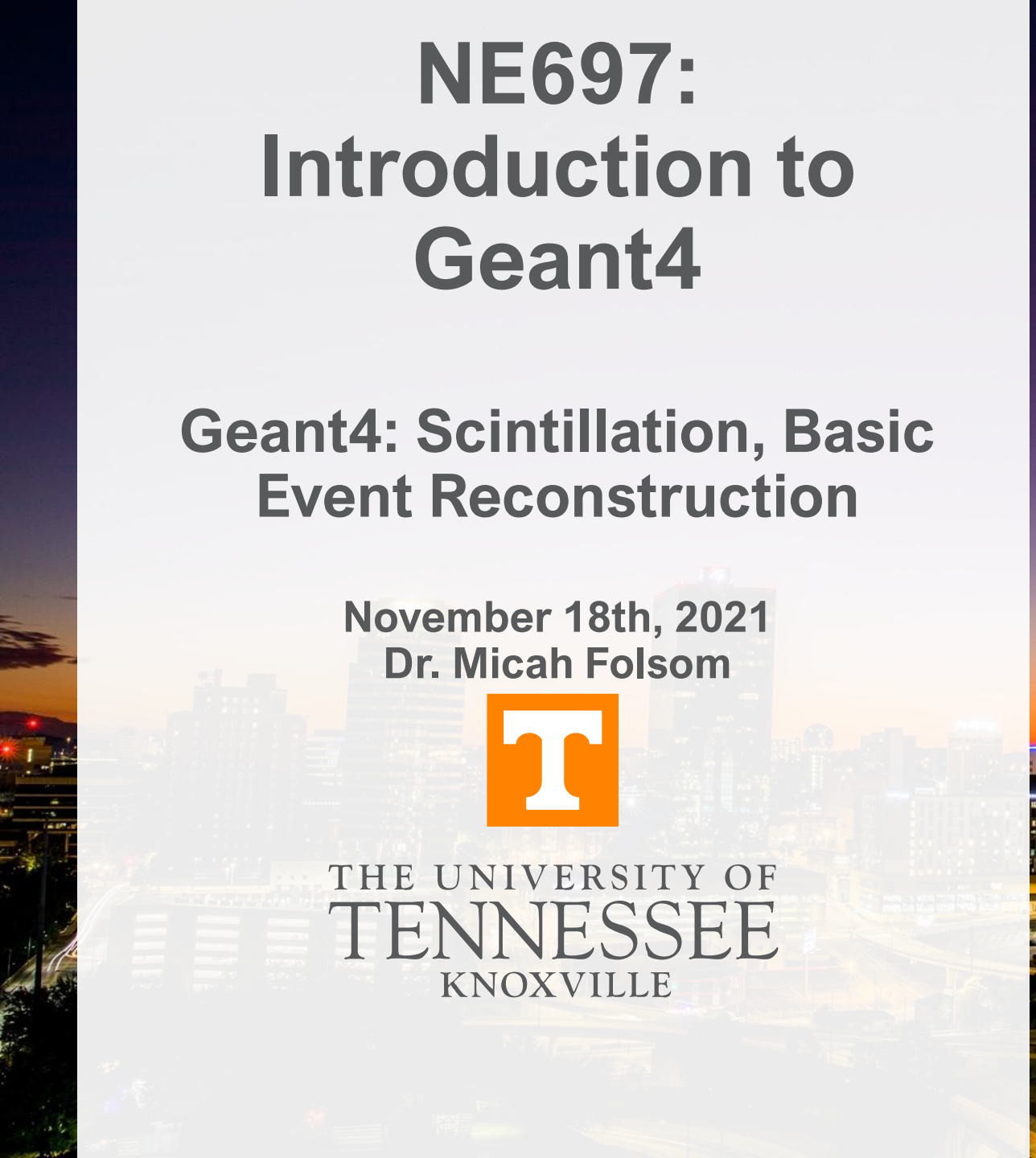
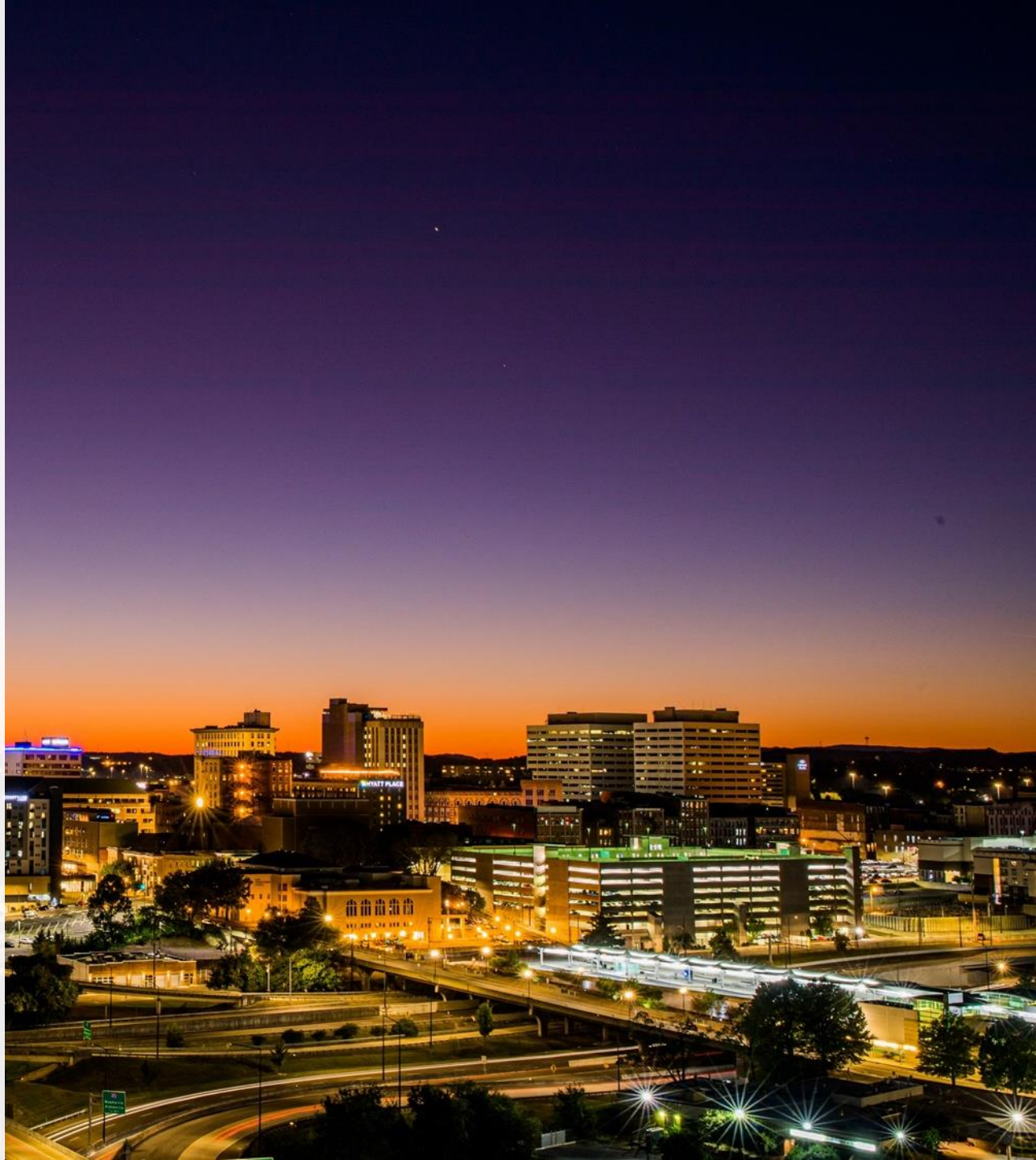
NE697: Introduction to Geant4

Geant4: Scintillation, Basic Event Reconstruction

November 18th, 2021

Dr. Micah Folsom

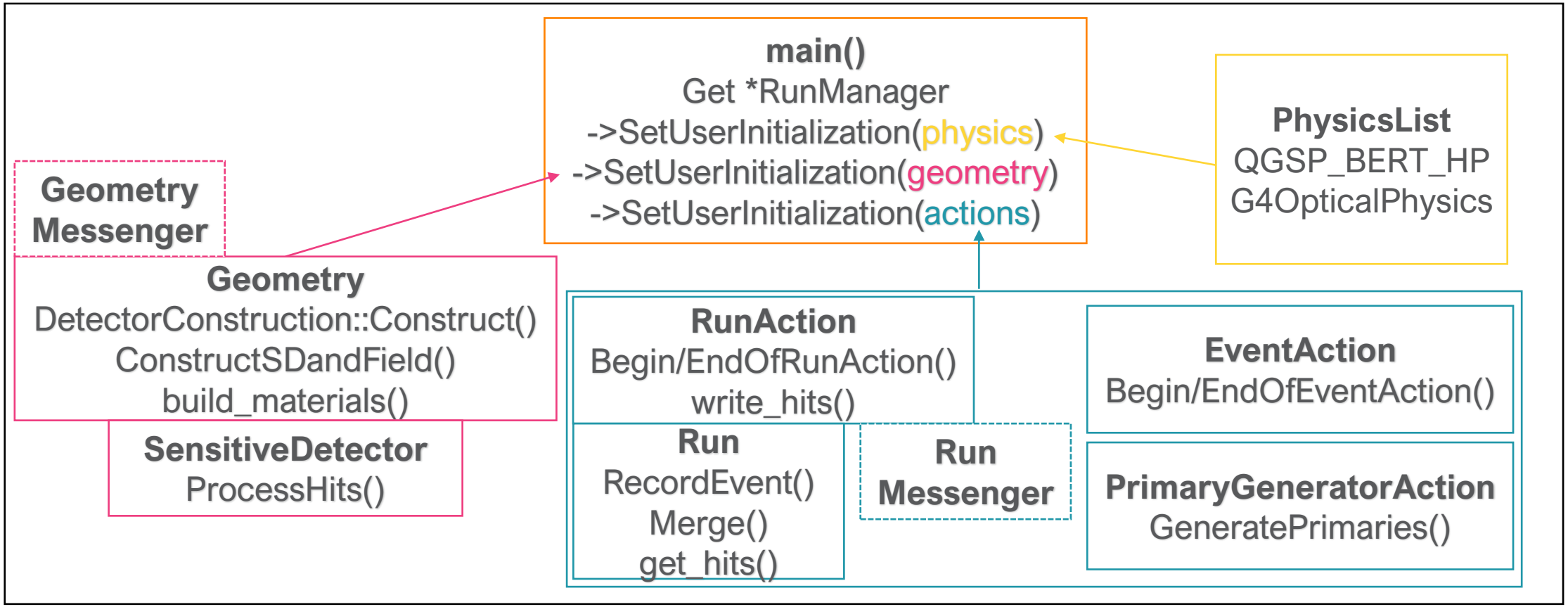




Today's Agenda

- Small change to CMakeLists.txt
- Scintillators, optical physics
- Reducing the number of Hits we generate
- Office hours TODAY, 8-9 PM ET
- Final project formal proposals – due TONIGHT, midnight PT

Geant4 Program Anatomy



Scintillators Review

- Characterized by:
  - Light yield per unit energy deposited [photons/MeV]
    - Different for recoiling e-, protons, and ions
  - Rise time(s)
  - Decay time(s)
  - Emission spectrum
- Optics in general
  - Absorption length
  - Refractive index
  - Boundary conditions (e.g. reflectors like Teflon, ESR)

Geant4: Optical/Scintillation Physics

- If you want optical transport, you must:
  - Enable optical physics (e.g. G4OpticalPhysics)
  - Define ABSLENGTH and RINDEX for the material(s) being used
- If no optical properties are defined, optical photons will die when they hit that volume
- Scintillation **does not conserve energy**
  - Energy is lost in each step - fine
  - Energy deposited * light yield → mean of Poisson/Gaussian → dice roll
    - Distinct from taking the energy and sampling from the emission spectrum (better)
    - …or determining the # as part of the energy loss calc (best, insanely complicated)

Geant4: Scintillators

- Define a G4Material like normal (or ask NIST)
- Add a **G4MaterialPropertiesTable**
- Two parts: AddProperty(), AddConstProperty()
  - **AddConstProperty()**: single values
    - SCINTILLATIONYIELD, FASTTIMECONSTANT, FASTSCINTILLATIONRISETIME
  - **AddProperty()**: energy-dependent values
    - RINDEX, ABSLENGTH, FASTCOMPONENT, PROTONSCINTILLATIONYIELD
- Then just G4Material::SetMaterialPropertiesTable() and voila
- Don’t forget: modifying a G4Material applies globally! (see: code)

Geant4: Scintillators

- For energy-dependent values, you need at least 2
  - Geant4 will linearly interpolate (set equal for a constant value)
- Changed significantly in 10.7! Backwards compatible, though
- As always, the docs: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/physicsProcess.html?highlight=scintillation#scintillation
- Seriously, read the documentation
- If you misspell a property (“RNIDEX”) *it may be hard to see the error*

Geant4: Optical Transport

- At a minimum, **must** define a refractive index and absorption length
- At the boundaries, will follow classic optics (Snell's, etc)
- Optionally, may define specific rules at specific boundaries
  - Can even provide look-up-tables with custom distributions (e.g. from empirical data)
- May also define custom properties from your custom physics
- Beware the rabbit hole – use it as a guide. Good agreement with experiment (in absolute numbers) is very hard!
- https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/physicsProcess.html?highlight=scintillation#the-unified-model

Geant4: Scintillators

- [DEMO]
  - What if we try to make optical photons?
  - Register G4OpticalPhysics in main()
  - Optical photons work now (well, kinda…)
  - Set up G4MaterialPropertiesTable
  - RINDEX, ABSLENGTH
  - Ok now they really work
  - SCINTILLATIONYIELD, FASTTIMECONSTANT, FASTSCINTILLATIONRISETIME, YIELDRATIO, RESOLUTIONSCALE
  - Making the visualization very unhappy

Geant4: Reducing Hit Info

- SensitiveDetectors refresher
  - What's the function we override to create Hits?
- Where should we make the decision whether to record a Hit?
- Can we make this dynamic?
- Scenario: we just want to make an energy spectrum from a gamma-ray source in a scintillator
  - What particles do we actually care about?
- [DEMO]
  - Filter only on particles we care about