Game and simulation development is very much a data and math-intensive activity. This course is designed to fill in some of the gaps in our CS curriculum to better prepare OSU graduates for job opportunities in those fields.

This is not a game-programming course. We will not be creating any games. Instead, it is a CS course that will fill in many of the missing pieces for those wanting to enter the simulation and game development worlds. Similarly, this is only a little bit a graphics course. We will not be writing graphics programs, but you will sometimes be given graphics program skeletons to test your coding. Even if you aren’t looking for a job in one of these fields, you will likely find these fast-paced and varied topics useful and enjoyable.

Course topics will likely include:

**C++**
- Virtual constructors and destructors
- Virtual functions
- Templates
- Replicating some of the STL functionality (e.g., vector) in a more efficient way
- More efficient dynamic memory allocation and freeing

**Numerical Methods**
- Solving systems of linear equations: exact and progressive refinement
- Newton’s method for solving for roots of nonlinear equations

**Graphics Math**
- 3D coordinate systems
- Vectors: dot products, cross products, coordinate-free geometric constructions (distance from a point to a line, distance from a point to a plane, distance between 3D skew lines, etc.)
- Gram-Schmidt orthogonalization
- More detail on 3D transforms
- Quaternions
- Euler Angles ↔ Quaternions ↔ Matrices
- Camera manipulation
- Forward kinematics (i.e., hierarchical transformations)
- Inverse kinematics using Jacobian matrices

**Graphics Awareness**
- Particle systems
- Tristimulus color theory
- Color spaces (RGB, CIE)

**Computational Geometry**
- Intersecting triangles
- Collision detection
- Geometric level of detail

**Physics**
- Rigid-body dynamics (motion of a single-body system)
- Different kinds of forces (weight, drag, friction, buoyancy, etc.)
- Integrating equations of motion
- Motion of a multi-body system
- Collisions, impulse-momentum, rebounding
- Modeling the world as a mesh of springs (e.g., simplified fluid flow, cloth)

**Architectures**
- Cache coherency
- Multicore

**Multithreading**
- Critical code sections
- Synchronizing

**Guest Lectures**
- Game development
- Game business management
Enforced Prerequisites:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 261</td>
<td>Data structures</td>
</tr>
<tr>
<td>MTH 232</td>
<td>Discrete math</td>
</tr>
<tr>
<td>MTH 251</td>
<td>Differential calculus</td>
</tr>
<tr>
<td>MTH 252</td>
<td>Integral calculus</td>
</tr>
</tbody>
</table>

Other Recommended Classes to Take:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 450</td>
<td>Introduction to Computer Graphics</td>
</tr>
<tr>
<td>CS 475</td>
<td>Parallel Computing</td>
</tr>
<tr>
<td>CS 519</td>
<td>Shaders</td>
</tr>
<tr>
<td>MTH 451</td>
<td>Numerical Linear Algebra</td>
</tr>
<tr>
<td>MTH 452</td>
<td>Numerical ordinary differential equations</td>
</tr>
<tr>
<td>MTH 453</td>
<td>Numerical partial differential equations</td>
</tr>
<tr>
<td>PH 221</td>
<td>Physics with calculus</td>
</tr>
</tbody>
</table>