# Dev Env

Bullet Physics

* Code complicated
* Bad doc

PhysX

* Couldnt get started
* Lots of code for basic setup (scene, lighting etc.)

Unity

* offers lighting, rendering, user input etc.
* DLL offers speed needed for particle computation

# Problems and solutions during development

Loading TetWild files

* .msh/.mesh & surface .obj files have to be parsed

Unity Mesh for representation

* Mesh object needs verts, tris and normal as minimal setup for correct rendering
* Normal are not given by TetWild
* Normal have to be calculated
* Unity calculates normal as follows:
  + Split vertices according to a “smoothing angle” (angle between face normals)
  + Each vertex then gets assigned the average of all face normal of the faces using the vertex
  + On smoothing angle == 0 each vertex gets split and assigned the faces normal
  + Behavior may have to be reproduced to sufficiently render the tet mesh surface

DLL Setup

* DLL needs information about:
  + Vertices
  + Constraints
  + Colliders
  + Parameters
* Colliders
  + Unity offers colliders but they imply overhead -> making own colliders
  + Colliders only contain data. Calculations are done in DLL
  + BoxCollider:Collider

Issues with the setup

* Requirement is inclusion into game code -> car controlled by the user -> in this case unity’s Rigidbody