



McNeel Europe

European division of Robert McNeel & Associates

Grasshopper Advanced Workshop on RhinoCommon

This is a scripting course based on the RhinoCommon and its fundamental link with Grasshopper. It is designed to allow the user to accomplish more with this algorithmic modeling program and has the goal to enter the programming world and tinker more complex, interactive solutions. We will also explore advanced programming paradigms. There is no class official programming language, as both C# and Vb.Net are possible on the participant's side, and all examples will be provided in both C# and Vb.Net. Additionally, we will see how to get started writing full .Net plug-ins. Finally, we will have time to explore user's own proposals on the third day.

Day 1 Morning: programming introduction in .Net

- The Grasshopper scripting components. Choosing a .Net language. Language developments
- Variables declaration, assignment and utilization. Operators. Methods [functions]. Calls
- Classes: declaration and instancing. Constructors. Importing a namespace. On3dPoints, OnLines
- Arrays declaration and usage. Lists. Adding to arrays and lists, advantages and opportunities.

Afternoon: patterns

- About OOP (object oriented programming) as opposed to procedural programming. Discussion
- Example of OOP good code reuse: sorting points by coordinates using the .Net SDK classes
- Lists as input parameters. Trees as input parameters. Usage and limitations
- Finding resources: on the net with website that can help getting started and troubleshoot. And books. Parametric mesher

Day 2 Morning: extending Grasshopper functionality with our definitions

- Store data between updates. The use of fields [globals, or static locals]
- Examples on how to use stored data between updates: a simple agents simulation
- Baking geometry with scripting directly into the Rhino document. Baking with names
- Passing custom types from a scripted component to another one. Our own code reusability
- Rendering an animation from Grasshopper. How to get started and final results

Afternoon: customizing our tools

- Our Rhino plug-in with Visual Studio C# [Vb.Net] Express Edition & wizard.
- Writing a custom Grasshopper component: an exporter for our data to Excel.
- The Perlin noise and simplex noise components.

Day 3 All day: personal project

- Rehearsal and deepening on any example from first two days. A project that you want to start on your own, being it a Rhinoceros plug-in, a Grasshopper assembly or a script. Example might be to send data through network with UDP to Processing

MINIMUM REQUIREMENTS

A good foundation of Grasshopper visual programming is mandatory. You will need a level which corresponds to the Grasshopper Primer course outline. Examples of things that will not be covered in this course are: sorting document spheres by diameter, paneling of a surface with grasshopper components. You are expected to already know these from the Grasshopper course.

The naming convention in this paper is C# [Vb.Net]