Parametric Design For Cloud Applications

A workshop for advanced Grasshopper users

Introduction

In the past few years, cloud features have taken a front seat within the CAD industry. The main actors (Dassault, Autodesk) are exploring cloud versions of their products, while other actors have appeared with native cloud technologies (OnShape, Vectary…). McNeel is developing a promising strategy with Rhino Compute and Rhino Inside, which makes Rhino a strong contender for CAD-based cloud applications.

There are more and more examples of parametric cloud applications, both in the AEC industry and big design firms who have the need to collaborate on an international scale. On the other hand, the online retail industry has seen the rise of product configurators, which are evolving to include more flexible and powerful parametric choices for the consumers.

As an introduction to this workshop, we give an overview of the cloud tools available within the Rhino+Grasshopper ecosystem (Rhino Compute, Speckle, Swarm, Trinckle, Emarf, Ghshot...). We identify common design challenges and strategies for all these types of applications:

- Computational efficiency
- Lightweight solutions
- Data management
- Maximizing automation

Mastering these design principles in Grasshopper is the main goal of this workshop, which is targeting already experienced Grasshopper designers.

The participants first go through the process of creating and uploading a simple definition to ShapeDiver as a concrete example of how a cloud application based on Grasshopper gets started.

Afterwards, in parts 1-3 of the workshop, we provide advanced training for preparing and optimizing Grasshopper definitions for cloud applications. We make use of some popular tools of the Grasshopper ecosystem (Metahopper, Human, OpenNest, Squid, FabTools...) as well as tools included in the ShapeDiver plugin.

In the last part, we go deeper into some fundamental use cases for cloud applications based on parametric definitions. We build examples using the ShapeDiver online platform but keep the focus on high-level implementation principles which apply to any technology leveraging CAD files in the cloud.

Part 1: Optimizing Definitions for Size and Speed (90 mn)

Optimizing Grasshopper definitions is often critical in the context of front-end applications and high-volume applications such as batch processing. At ShapeDiver, we recommend our users to create definitions that typically load in 5 seconds or less, since the average B2C end user won't wait for longer than this time before heading elsewhere. In this first part of the workshop, we give tools to the designers to better understand the factors of performance of a parametric model, and provide strategies to achieve fast and lightweight definitions.

Design strategies

We first give some context regarding computational and differential geometry and explain which Grasshopper operations are efficient, and which ones are more computationally intensive. Since there is always more than one way to achieve a specific result in Grasshopper, the goal of this part is to give the participants a good intuition regarding which design strategies to choose for efficient solutions. Without being completely exhaustive, we give some examples of operations that result in intensive computations and propose alternative ways to solve problems efficiently. Such operations include boolean operations on solids, Brep deconstruction, offsetting curves, extracting mesh edges, etc...

Meshing principles

Grasshopper represents geometry in two ways: using b-reps or using meshes. This is something the majority of Grasshopper developers already know, but a deeper understanding of these concepts can help clarify how they can affect the performance of Grasshopper definitions when they run online.

This performance is especially crucial for any model that will be accessed or interacted with via a web browser, where the available internet connection or even the device's own hardware can play a significant role on the UX.

In this part, we discuss the difference between b-reps and meshes and present strategies to optimize the loading of solutions online. We also present some scripts for fast operations on meshes to substitute to their native Grasshopper equivalents (mesh extrusion, mesh piping, mesh lofting…).

Visualization versus production

Depending on the application, the level of detail needed for the solutions of a definition can vary. If the goal is only visualization, output geometry will typically only need to look good, and not necessarily ready for production purposes. Even if CAD exports are needed by the application, it might be optimal to use a simplified version for online visualization. We discuss in this part how to optimize models for those different purposes.

- Use mesh normals and textures
- Skip intensive operations (booleans…)
- Leverage transformations

Part 2: Designing Data (90 mn)

We need strategies to structure the large amounts of data flowing within cloud applications, as well as methods to read them, process them and export them efficiently. This is necessary to build custom user interfaces, as well as connecting the definition with ERP systems, online databases and production facilities. One can think of how to input a point cloud from a web interface or how to export a complete bill of materials from a definition.

JSON

The JSON data structure is the standard way to communicate large amounts of data online, and it can do wonders in Grasshopper definitions as well. It organizes inputs and outputs in clean, structured objects that prevent the need for countless sliders, and facilitate communication with a web application. In this part, we touch the following points:

- How to design the architecture of JSON objects as the input of a definition
- How to output data in JSON form for online parsing
- Using the ShapeDiver tools to process JSON efficiently

Advanced tree management

Manipulating trees in Grasshopper might seem complicated at first, but a good understanding of some core principles leads to much improved data management and definitions that often perform faster while becoming more readable by other designers.

We present those principles and go through several concrete examples where advanced tree manipulation leads to drastic improvements. We also see how those principles facilitate usage and maintenance of JSON objects.

Part 3: Production Automation (90 mn)

By giving access to parametric design online, we open the door to innovatives processes in the context of product life cycle, design iterations and remote collaboration. We present in this part the fundamental tools allowing automation of complex workflows controlled by a single Grasshopper definition.

Throughout this part, we build a fully functional use case as a support to presenting the different concepts.

File I/O

When a parametric definition is part of a cloud application, it needs to interact with many moving parts, from client-side interfaces to backend servers and databases. In particular, all Grasshopper functionalities making use of the local file system need to be replaced with inputs and outputs that are ready to be connected to a website or online storage systems.

In particular, we present the export tools available in Grasshopper using the ShapeDiver plugin and how to make the best use of them.

Nesting

True automation sometimes involves complex operations such as the nesting of CNC-cut parts. In this part, we present the general challenges with automated nesting and the more specific points to take into account in the context of Grasshopper and cloud applications.

We build a concrete example using the open-source OpenNest plugin and discuss data workflows within online applications.

Quotes and drawings

By combining native Grasshopper components such as Make2D, as well as additional bitmap drawing tools like Squid and the ones included in the ShapeDiver plugin, we present ways to build dimension drawings, listings of parts and even full quotes directly from Grasshopper, that can be exported and integrated into e-commerce, ERP and CPQ systems.

Annex

Target audience

Participants in the workshop should be intermediate to advanced Grasshopper users interested in advanced design techniques to optimize their parametric definitions.

No prior ShapeDiver experience is required.

C# or Python scripting are a plus but not required.

Required plugins

We like to make use of the amazing tools developed by the Grasshopper community. During the workshop, we will present and use tools from the following plugins:

- Metahopper
- Human
- FabTools
- OpenNest
- Squid (ShapeDiver edition)
- … and of course ShapeDiver