



Choose a Module

**50% Individual Coursework**

**2023-24 Autumn**

**Student Name: Anish Kayastha**

**London Met ID: 21040641**

**College ID: np01mm4a210024**

**Assignment Due Date: Wednesday, January 10, 2024**

**Assignment Submission Date: Wednesday, January 10, 2024**

**Word Count: 6305**

**Project File Links:**

| **Sample Music:** | Keep Unlisted YouTube URL of your Project Here |
| --- | --- |
| **Google Drive Link:** | Keep Google Drive URL of your Project Here with Anyone in Organization can View Option Enabled |

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

Table of Contents

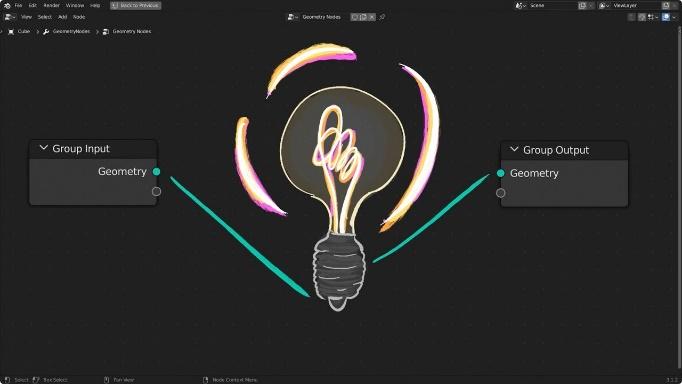
**No table of contents entries found.**

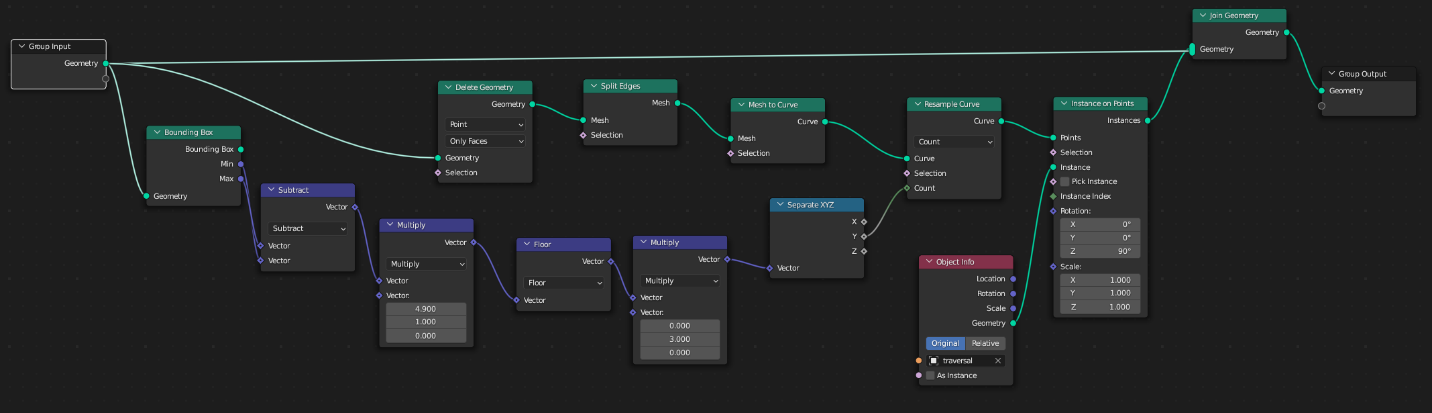
# 1. Introduction

## Overview of the project

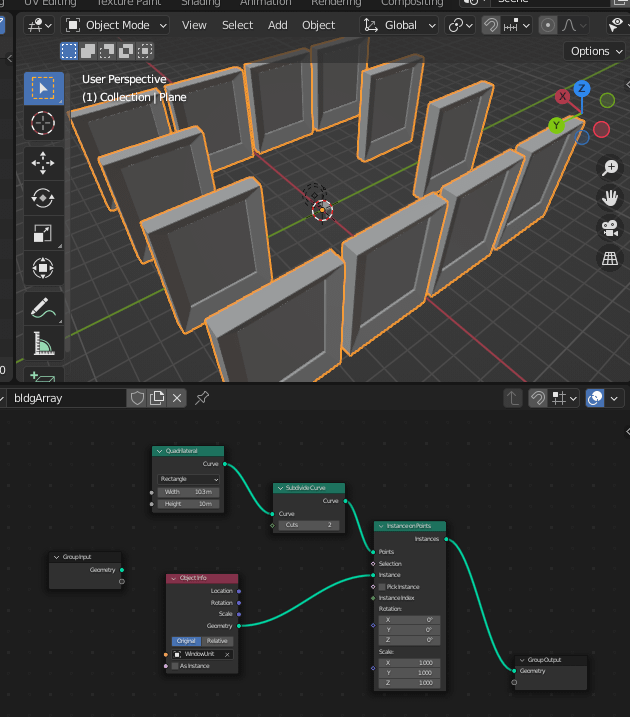
## Objectives and goals

# Geometry Nodes

The introduction of geometry nodes in Blender marks a significant step forward in 3D design practices. Geometry nodes helps to transition from traditional modifier constraints, which permanently alter mesh data, to a dynamic workflow where everything is changeable. This new approach allows for non-destructive modifications, where alterations made via interconnected nodes are instantly updated in real-time. The visual programming interface provided by geometry nodes offers a high degree of flexibility, enabling users to generate, duplicate, and transform elements with ease. Furthermore, these nodes can target specific mesh properties, allowing for intricate customization. Geometry nodes also facilitate the creation of custom node groups, facilitating automation of repetitive tasks and tailored workflows. The potential applications of geometry nodes are manifold, impacting areas such as procedural modeling, animation, hard surface modeling, and data visualization.



A key advantage of this system is its non-destructive workflow. Contrary to traditional modifiers, modifications made through nodes are dynamic. This feature allows for effortless adjustments and experimentation. For instance, a single node can be altered to modify an entire building structure, or randomized to generate unique variations of a plant model.

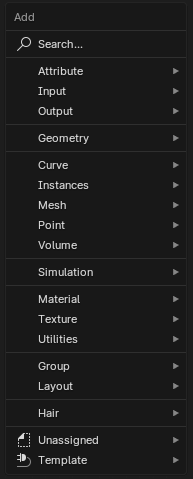
Moreover, geometry nodes provide fine-grained control, enabling users to target specific parts of the mesh based on various properties, such as vertex groups, material assignments, or custom data. This capability allows for highly detailed effects, such as creating patterns that only appear on specific edges of a mesh or scaling objects based on their color information.

The ability to build custom node groups is another significant advantage. These groups can function as mini-programs within Blender, facilitating automation of repetitive tasks or creation of unique workflows tailored to specific project needs. For example, a custom node group can be built to scatter rocks across a scene with variations in size and rotation, enhancing efficiency and consistency.

The applications of geometry nodes in 3D workflows can be very transformative for any 3D artist. In the realm of procedural modeling, these nodes enable the creation of complex and adaptable shapes, such as buildings with parametric rules or realistic-looking foliage. In the field of animation, they enhance the dynamic aspect by animating the transformation or duplication of objects based on various factors. In hard surface modeling, geometry nodes streamline the process by providing precise control over extrusions, bevels, and other details, making the creation of sharp edges and technical designs more achievable.

In conclusion, geometry nodes are not just another tool in Blender. They with their visual programming approach, commitment to non-destructive workflows, and exceptional level of control empower users to push the boundaries of creativity. From procedural modeling and animation to hard surface modeling and data visualization, geometry nodes unlock a world of possibilities for 3D artists and designers.

# Geometry Node Categories

**1. Attribute:** These node types handle data for different parts of our mesh, like vertices, edges, and faces. These can include data related to position, color, surface orientation, or custom data. We can perform calculations on attributes, merge them, or use texture data to influence them. This affects the visuals, behavior, and characteristics of our mesh.

* **Boolean:** True or false value.
* **Integer:** 32-bit integer.
* **8-Bit Integer:** Smaller integer with a range from -128 to 127.
* **Float:** Floating-point value.
* **Vector:** 3D vector with floating-point values.
* **2D Vector:** 2D vector with floating-point values.
* **Color:** RGBA color with 32-bit floating-point values.
* **Byte Color:** RGBA color with 8-bit positive integer values.
* **Quaternion:** Floating point quaternion rotation.

**2. Color:** These nodes are dedicated to modifying the color data of our object. Color data is often used for materials, vertex colors, or textures. We can combine two colors, adjust brightness or contrast, or split a color into its Hue, Saturation, and Value components. This helps us to create complex color variations, adjust existing colors, and define color schemes.

**3. Geometry:** Nodes that can operate on different types of geometry, like volumes and meshes. We can combine meshes, separate a single mesh into its parts, or perform boolean operations on two meshes. This helps build complex shapes, create intricate geometry, and work with both mesh and volume data.

**4. Input:** These nodes provide the data your setup will use. They act as gateways, importing information from various sources like meshes from your Blender scene, numerical values, or texture data. This defines the foundation of your workflow.

**5. Mesh:** Nodes that operate specifically on mesh data, providing a more focused set of tools for manipulating meshes. You can increase mesh resolution, recalculate mesh topology, or apply transformations to your mesh. This helps perform detailed mesh manipulations, remesh for simulations, subdivide for smoother surfaces, and transform for positioning and shaping your objects.

**6. Mesh Primitives:** Offers a quick way to generate basic geometric shapes like cubes, spheres, cylinders, or Icospheres. This provides a foundation for our mesh creation.

**7. Point:** Nodes that focus on modifying individual points within a mesh. You can modify the location of specific vertices, extract point data, or sample points based on various factors.

# Fields

Traditional workflows often involve editing individual elements (vertices, edges, faces) of a mesh, a laborious and inefficient approach. Fields offer a revolutionary solution. Akin to mathematical functions, fields operate on each element of a mesh and produce a corresponding output value. This establishes a procedural approach, where rules and calculations automatically govern the data for each element, unlocking a world of dynamic effects.

## 3.1 Field Visualization and Socket Shapes:

Geometry nodes leverage socket shapes to visually distinguish between fields and single values. This distinction is crucial for constructing efficient node workflows. Here's a breakdown of the prevalent socket types:

**3.2 Circle Socket:** This signifies a single value input or output. The value remains constant for all elements of the mesh. Imagine a node that takes a single "scale" value as input with a circle socket. This node applies that same scale uniformly to the entire mesh.

**3.3 Diamond Socket:** This represents a field input or output. The value can vary for different elements within a mesh. For example, a node might have a diamond-shaped socket for "position" output. This implies it calculates a unique position (X, Y, Z) for each vertex in the mesh, creating variations in the mesh's overall form.

**3.4 Diamond with Dot Socket:** This signifies a field with constant domain input or output. It's a special type of field where the data points have a fixed or predetermined order. Imagine a node that generates a series of evenly spaced points along a curve. The "distance from curve" information would be represented by a diamond with a dot socket, as the distance is calculated for each point (field), but the order of these points is always fixed (constant domain).

* Attribute Nodes
* Input Nodes
* Output Nodes
* Geometry Nodes
* Curve Nodes
* Instances Nodes
* Mesh Nodes
* Point Nodes
* Volume Nodes
* Simulation Zone
* Material Nodes
* Texture Nodes
* Utilities Nodes
* Group
* Hair Nodes

### Attribute Nodes:

Attribute nodes are the meticulous organizers of the node world. They handle the specific data points associated with the different parts of a mesh, such as vertices, edges, and faces. By manipulating attributes like position, color, and normals, these nodes can dramatically alter the visual appearance and behavior of a mesh. They are the fine-tuners, adjusting the minutiae that contribute to the overall characteristics of the model.

### Input Nodes:

Input nodes serve as the primary gateways for importing external information into the node setup. Whether it's mesh data, numerical values, or texture information, these nodes lay down the foundational data that will be processed and transformed by subsequent nodes. They are the starting point of any workflow, defining the initial conditions and resources available for creative exploration.

### Output Nodes:

Output nodes are the culmination point of the node setup. They take all the processed data from the preceding nodes and finalize it for use within the Blender scene. These nodes are crucial, as they determine the ultimate output of the node system, translating the complex interplay of data into a coherent and usable form.

### Geometry Nodes:

Geometry nodes are the architects of the node world, allowing for the creation and manipulation of complex objects through a node-based interface. They are capable of handling both volumes and meshes, which means they can craft intricate geometries from scratch or modify existing structures with precision and creativity.

### Curve Nodes:

Curve nodes specialize in the creation and manipulation of curves, which are essential for crafting complex shapes and organic forms. These nodes provide the tools to sculpt smooth, flowing lines and paths, which can be used to define the contours of models or guide animation trajectories.

### Instance Nodes:

Instance nodes are the masters of duplication, enabling the manipulation and replication of objects within a scene. They are the key to creating patterns, arrays, or clusters of objects, making them invaluable for scenes that require multiple, similar elements, such as a forest of trees or a fleet of spaceships.

### Mesh Nodes:

Mesh nodes are dedicated to the manipulation of mesh data. They offer a focused toolkit for tasks such as increasing mesh resolution, recalculating topology, or applying transformations. These nodes are essential for refining the structure and detail of models, ensuring they meet the desired specifications.

### Point Nodes:

Point nodes concentrate on the individual points within a mesh. They enable artists to manipulate vertex positions, extract specific point data, or sample points based on a variety of factors. These nodes are the micro-surgeons of the node setup, capable of precise adjustments at the most granular level.

### Volume Nodes:

Volume nodes deal with the creation and manipulation of volumetric elements like smoke, clouds, or any other gaseous form. They are the magicians of the node world, conjuring complex shapes and effects that add depth and realism to a scene.

### Simulation Zone:

The Simulation Zone node defines the parameters and boundaries of simulations within the scene. It is the director of dynamic systems, controlling how and where elements like fluids, particles, or physics interactions occur.

In conclusion, nodes are the versatile and powerful tools that make 3D modeling and animation not just possible, but also efficient and deeply creative. They offer a modular approach to digital creation, where each node performs a specific function, yet collaborates within a larger, interconnected system. As technology advances, the capabilities of nodes continue to expand, opening new horizons for artists and developers to explore and innovate within the digital landscape.

1. Introduction

• Overview of the project

• Objectives and goals

2. Literature Review

• Evolution of 3D environment modelling

• Previous techniques and tools used in 3D environment modelling

• Advancements in procedural modelling

• Applications of geometry nodes in 3D modelling

3. Methodology

• Overview of Blender and geometry nodes

• Procedural modeling with geometry nodes

• Techniques used in bridge and river creation

• Asset creation and scattering

4. Procedural Modeling with Geometry Nodes

• Understanding geometry nodes

• Implementation of geometry nodes in Blender

• Procedural bridge creation

• Procedural river creation

5. Asset Creation

• Techniques for asset creation

• Overview of assets created

• Importance of assets in environment creation

6. Environment Design and Development

• Design principles for environment modeling

• Development process of the environment

• Challenges faced during the environment creation

7. Scattering Techniques

• Introduction to scattering

• Techniques for scattering assets

• Importance of asset scattering in environment creation

8. Visual Realism and Optimization

• Achieving visual realism in a 3D environment

• Optimization techniques for real-time rendering

• Balancing realism and performance

9. Case Studies

• Analysing case studies of procedural environment modeling

• Comparison with traditional environment modeling techniques

• Real-world applications of procedural modeling with geometry nodes

10. Results and Discussion

• Evaluation of the created environment

• Strengths and weaknesses of the approach

• Future improvements and extensions

11. Conclusion

• Summary of the project

• Achievements and contributions

• Future prospects