Procedural Terrain Generation – Guide

# A screenshot of a computer Description automatically generated with medium confidenceHow to run the project

1. Set the **Game** project as startup project
2. Run the project

# How to run different solutions

In the file **Scene\_Game.cpp** are several functions that are in comments.

1. **CreateLandscape\_2DPlane()**

This creates a normal terrain using a 2D plane and changing the height by using a noise map that generated.

1. **CreateLandscape\_VoxelsCPU()**

This creates a voxel terrain that is completely made and stored on the CPU.

1. **CreateLandscape\_VoxelsGPU()**This creates a voxel terrain where the center points of the voxels are made on the CPU and the voxels itself are made on the GPU using the Geometry shader
2. **CreateLandscape\_2DPlaneOctree(*true*)**

This creates a normal terrain using a 2D plane and changing the height by using a noise map that generated.

The **paremeter** gives the option to visualize this data structure using the *line rendering* and *GPU instancing*.

1. **CreateLandscape\_2DPlaneKDtree()**

This creates a normal terrain using a 2D plane and changing the height by using a noise map that generated.  
For this data structure there is no option to visualize it. (Yet)

**Text

Description automatically generated**

# Controls

|  |  |
| --- | --- |
| W | Forward |
| A | Left |
| D | Right |
| S | Backward |
| RMB | Rotate |

# Text Description automatically generatedImportant files

Both data structures can be found underneath the **Data** folder.

The file that handles all terrain generator is in the **Generator** folder.

Line & Cube meshes can be found in the **Geometry** folder.