

Digital Terrain generation in 3D, utilising real-time, performance enhancements.

Software user manual.

Modifying launch settings

Step 1:

Resources	25/04/2023 15:49	File folder	
runtimes	25/04/2023 15:49	File folder	
Shaders	25/04/2023 15:49	File folder	
Textures	25/04/2023 15:49	File folder	
HonoursStageProject.deps.json	25/04/2023 15:48	JSON Source File	3 KB
HonoursStageProject.dll	25/04/2023 15:48	Application exten...	42 KB
HonoursStageProject.exe	25/04/2023 15:48	Application	145 KB
HonoursStageProject.pdb	25/04/2023 15:48	VisualStudio.pdb.f...	26 KB
HonoursStageProject.runtimeconfig.json	25/04/2023 15:48	JSON Source File	1 KB
Log.txt	25/04/2023 15:48	Text Document	1 KB
Microsoft.Win32.SystemEvents.dll	22/09/2022 21:22	Application exten...	28 KB
OpenTK.dll	10/09/2022 15:34	Application exten...	5,623 KB
System.Drawing.Common.dll	22/09/2022 21:23	Application exten...	171 KB

Navigate to the resources folder within the program directory.

Step 2:

Calibration.asc	07/04/2023 14:35	ASC File	3 KB
Calibration2.asc	07/04/2023 10:52	ASC File	1 KB
settings.txt	25/04/2023 11:25	Text Document	1 KB
settings_chunk_test.txt	13/04/2023 10:43	Text Document	1 KB
settings_test.txt	13/04/2023 10:43	Text Document	1 KB
TA12NE.asc	23/04/2023 16:28	ASC File	4,953 KB
TestInput.asc	07/04/2023 10:52	ASC File	28 KB

Open "settings.txt"

Step 3:

```
// Chunk sizes can only be 2, 3, 4, 9, 17, 33, 65, 129
// Settings with an asterisk are compulsory
// When loading a file, map size and chunk size will be overridden based on the dimensions of the height map being loaded.

map_size*:100
chunk_size*:17
map_scale*:0.5
render_distance:5.0
roughness:0.5
seed:
filename:resources/TA12NE.asc
terrain_algorithm:DiamondSquare
culling_algorithms:DistanceCulling FrustumCulling
```

Modify the values as desired and save the file.

Settings descriptions

- “map_size” is the length and width of the generated map in chunks, this value is compulsory.
- “chunk_size” is the size of the chunk in squares, one size unit is one square. This value is compulsory.
- “map_scale” Scales the size of the chunks. This value is compulsory.
- “render_distance” If distance culling is enabled, this value specifies the number of chunks you can see from the camera, a value of five means you can see five chunks away.
- “roughness” corresponds to the roughness of the terrain, a higher value, the rougher the terrain.
- “seed” dictates the program's random seed to generate the map, allowing for deterministic results if the same seed is used. If left blank a random value is generated.
- “filename” is the path relative to the executable of an external, .asc height map. If left empty, a random map will be generated.
- “terrain_algorithm” is the class name of the terrain algorithm to be used for map generation, if left blank, a flat map will be generated.
- “culling_algorithms” are the class names, separated by spaces, of the culling algorithms to be used, if left empty, none will be used.









In-app controls.

Once a map has been generated the following key binds can be used:

- “W, A, S, D” allow for directional movement, forward, left, backward, right, respectively.
- “E” turns on wireframe mode.
- “Q” turns off wireframe mode.
- “F or ESC” closes the program.
- “Z” saves the generated map to the directory with the program.
- “.” scales the map positively.
- “,” scales the map negatively.
- “G” regenerates the map (Does not work when external maps are loaded).

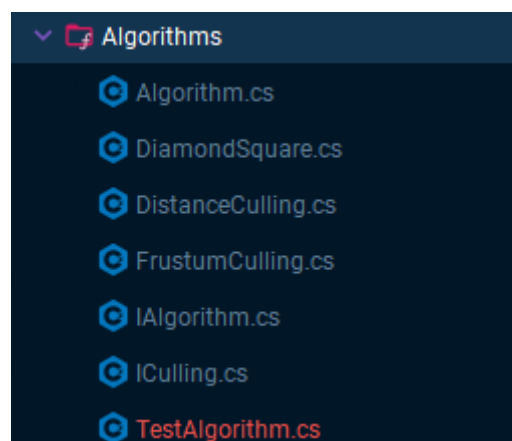
Adding your own algorithms.

Step 1:

	.git	25/04/2023 09:57	File folder	
	.idea	09/11/2022 21:12	File folder	
	.vs	07/04/2023 10:52	File folder	
	HonoursStageProject	23/04/2023 16:01	File folder	
	HSPUnitTests	25/04/2023 11:57	File folder	
	.gitignore	09/11/2022 21:12	Git Ignore Source ...	1 KB
	HonoursStageProject.sln	07/04/2023 10:52	Visual Studio Solu...	2 KB
	HonoursStageProject.sln.DotSettings.user	24/04/2023 11:14	Per-User Project O...	2 KB

Launch the “HonoursStageProject.sln” in your IDE of choice.

Step 2:



Add your own algorithm .cs file to the “Algorithms” folder.

Step 3:

```
using HonoursStageProject.Objects;

namespace HonoursStageProject.Algorithms;

public class TestAlgorithm : IAlgorithm
{
    // 0+1 usages
    public float[,] GenerateData(int pSeed, float pScale, float pRoughness)
    {
        throw new NotImplementedException();
    }

    // 0+1 usages
    public float[,] GenerateData(int pSeed, float pScale, float pRoughness, float[,] pPreSeed, bool pSeedCorners)
    {
        throw new NotImplementedException();
    }
}

public class TestCulling : ICulling
{
    // 0+1 usages
    public bool Cull(Chunk pChunk, Camera pCamera, Settings pSettings)
    {
        throw new NotImplementedException();
    }
}
```

Inherit from the interface you want to implement i.e., “ICulling” for culling algorithms and “IAlgorithm” for terrain algorithms and implement your functionality.

Step 4:



Build the solution (Method depends on IDE used). The built program will be located at “bin” -> “(Release or Debug)” -> “net6.0”

Follow “Modifying launch settings” to run your algorithm within the built program.