UPATTERN TOOLS
USER MANUAL

Hello dear customer,
thank you very much for supporting uPattern Tools.

uPattern Tools is a simple but powerful generative design toolset to extend Unity3D's utilitiy tools. It started as an university project to fill a gap of missing generative design tools that are like the scatter and array tools from 3ds Max, Maya and Blender. uPattern Tools was created because i kinda missed those features in Unity3D :)

This tool is still under developement and will be updated in future.

I hope you enjoy this tool.

Cheers!
Lewnatic

1 QUICK INSTALLATION & START

Quick installation
Simply select uPattern Tools from the Unity Asset Store and download it to your project. If uPattern Tools is a package simply import it in Unity3D with Assets -> Import Package -> Custom Package...

Quick start
Use the uPattern Tools Menu to select your tools. uPattern Tools uses a lot of tooltips to help customers to learn how to use its features.

2 USING UNITY PATTERN

Using uPattern Tools is quite simple. Just select the generators from the uPattern Tools menu. You can also select them from the generators folder. It is also recommended to keep the generator prefabs in the generators folder. **Nearly all generators uses tooltips, to make them more userfriendly.**

Module Number

Most generators use a module treshold to restrict the maximum number of modules that can be generated. If a threshold value of 15000 modules is reached, a warning appears. The input can still be higher but it is recommended to keep it lower then that number to ensure a good performance in Unity3D. Also the used modules should not be to complex and should not have more then 5000 triangles. The Volume Shuffler even uses a timer to prevent Unity3D from crashing when you generate to many GameObjects. The timer is set to 5 seconds.

Context Menu

Pressing **Tab** in the scene also opens a context menu for various options. Those options include a function to create a uPattern Tools Camera that renders screenshots in a folder, a way to add a connector gameobject to a selected scene asset to use it as a Tube Agent Module or Volume Modules. Its also possible to generate mazes, paths and volumes. The used values for those complex objects are taken from the current generator values of those tools. In future, more options to the menus will be added. In near future, there will be also a uPattern Tools settings menu to change various options like key bindings for the context menu.

3 FEATURE OVERVIEW

This Asset includes the following generators:

- **Simple Maze:** Array tool to arrange assets in a 2D Array and randomly rotate them by 90 to generate mazes with diagonal asset modules.
- **Grid Shuffler:** Array tool ro arrange multiple assets in a 2D Array its possible to arrange the assets in different ways like deterministically.
- **Move Agent:** A array generator that uses 2D Textures and Pearlin Noise maps to transform asset arrays.
- **Tube Agent:** A path generator that uses assets for pathing.
- **Volume Shuffler(Pro):** A powerful tool to arrange multiple assets in a volume and use modifiers to translate, scale or rotate them. Its also possible to project and voxelize 3D assets.

4 SIMPLE MAZE

The Simple Maze is an array generator. Arrays can be created by choosing one of your assets as a **Simple Maze Module**.

Maze Name defines the name of the generated array in your current Unity3D scene. **Maze Position** defines the position of the generator array in the Scene. Since the uPattern Tools Arrays use a list to store the array objects, the **Index Number** shows the number of the current generated object array.

Maze Width and **Maze Height** define the size of the 2D Array. The **Seed** changes with every Generate Maze command except if the **Keep Seed** flag is set to true.

The **Pivot Point** can be set to fixed values or altered freely even with arrays of the size of1x1. This means that uPattern Tools is capable of changing the pivot point by creating a mantled gameobject around your asset.

Press **Generate Maze** to create your asset array and use **Delete last** or **Delete all** to clear them.

Fig.1: The inspector of the Simple Maze.

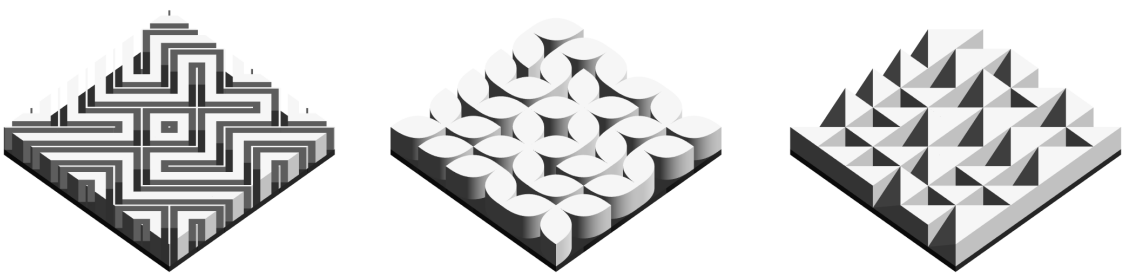

Fig.2: Asset arrays generated with the simple maze.

5 GRID SHUFFLER

The **Grid Shuffler** uses most of the features of the **Simple Maze** in a improved way.

It is possible to pick multiple assets and shuffle them together. The **Module Options** also provide different settings for object creation. Its possible to create an array objects with the following options:

- **SIMPLE_MAZE:** Random 90° rotated assets. Like the Simple Maze.
- **DETERMINISTIC:** Generates the modules deterministically by turns.
- **RANDOM:** Randomly picks the modules from the module list.

Another feature of the **Grid Shuffler** is the Regenerate button which generates the asset array with its new settings.

Keep in mind that no changes will appear if the **Keep Seed** flag is set.

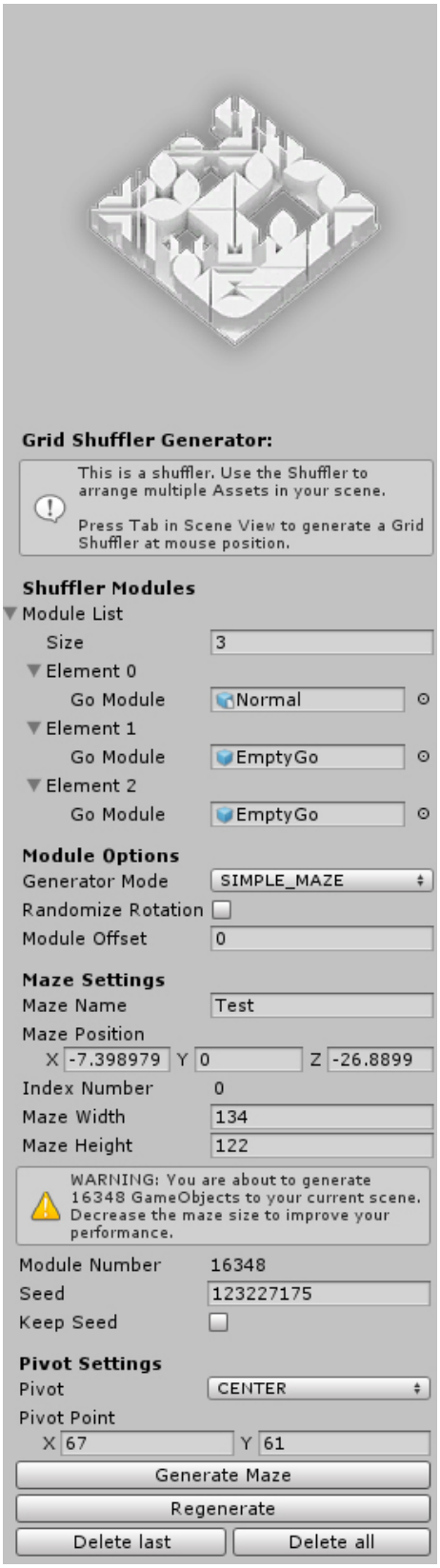

Fig.3: The inspector of the Grid Shuffler.

6 MOVE AGENT

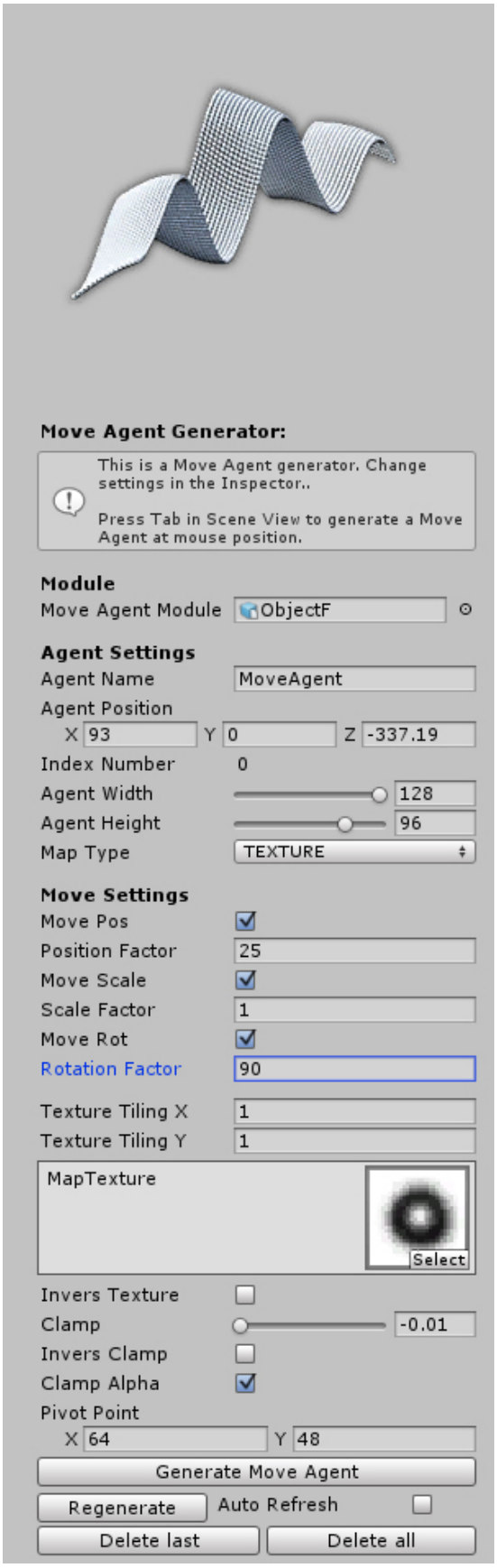

Fig.4: The inspector of the Move Agent in texture mode.

With the **Move Agent** it is also possible to generate a **Pearlin Noise** map or use a **Texture** to modify the asset arrays. It is also possible to use the **Texture** feature to rasterize the used textures.

Unique features of **Pearlin_Noise:**

- **POSITION FACTOR:** The amount the noise moves the assets in y-axis.
- **NOISE RESOLUTION:** The resolution of the **Pearlin Noise**.
- **NOISE POS X:** The X-Position on the Noise Map
- **NOISE POS Y:** The X-Position on the Noise Map

Unique features of **Texture:**

Move Pos, Move Scale and **Move Rot** activates the transform modifiers.

- **POSITION FACTOR:** The amount the grey scale of the texture modifies the asset position.
- **SCALE FACTOR:** The amount the grey scale of the texture modifies the asset scale.
- **ROTATION FACTOR:** The amount the grey scale of the texture modifies the asset position.
- **TEXTURE TILING:** The tiling of the texture.
- **INVERSE TEXTURE:** To invert texture.
- **CLAMPING:** Additional clamping features to set clamping amount, invers clamp and clamp alpha.

7 TUBE AGENT

The **Tube Agent** is a asset based path generator. The path is defined by modules. Each module is defined by a GameObject and a child. The position and rotation of the modules child determines the transform of the next spawned modul in the path. The pivot point of the parenting GameObject is also very importend. It should be always near the objects bounds depending of your preferences. The childs name must always be the same that is set for **Connector Name.**

Unique features of the **Tube Agent:**

- **CONNECTOR NAME:** Very important string to set the name of the child that is used to connect the modules.
- **RANDOM ROTATION:** If flagged true a random value for the local y-axis rotation of the connector is added.

Keep in mind that you can convert selected GameObjects to **Tube Agent** modules by pressing tab.

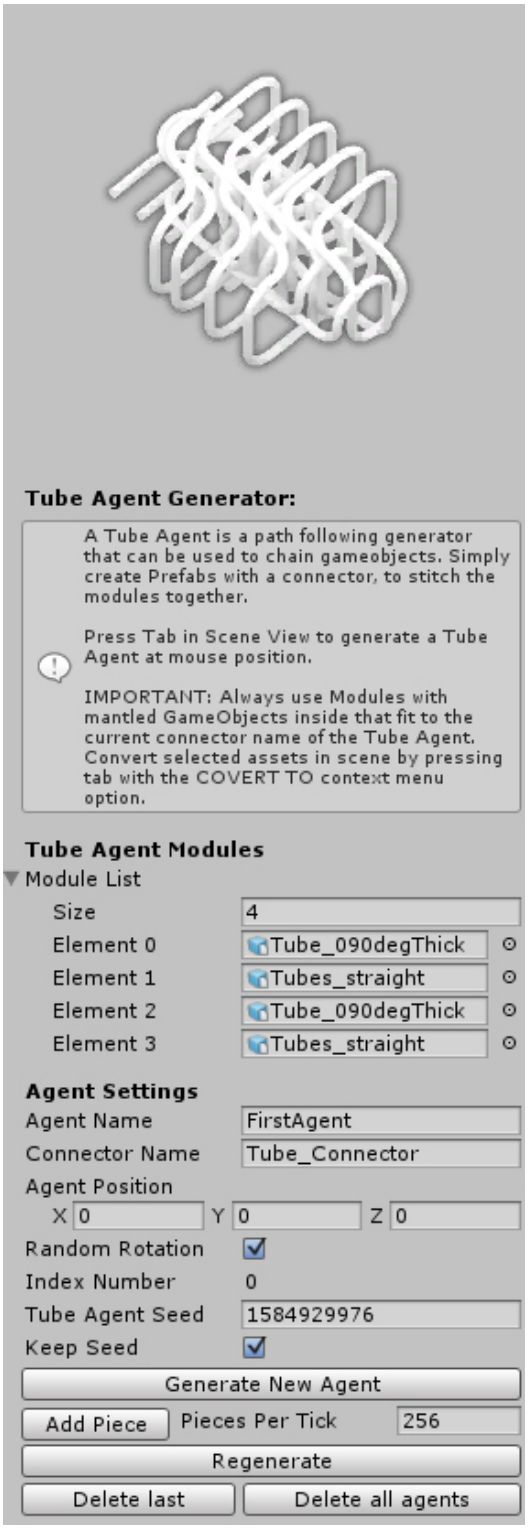

Fig.5: The inspector of the Tube Agent in texture mode.

8 VOLUME SHUFFLER (PRO ONLY)

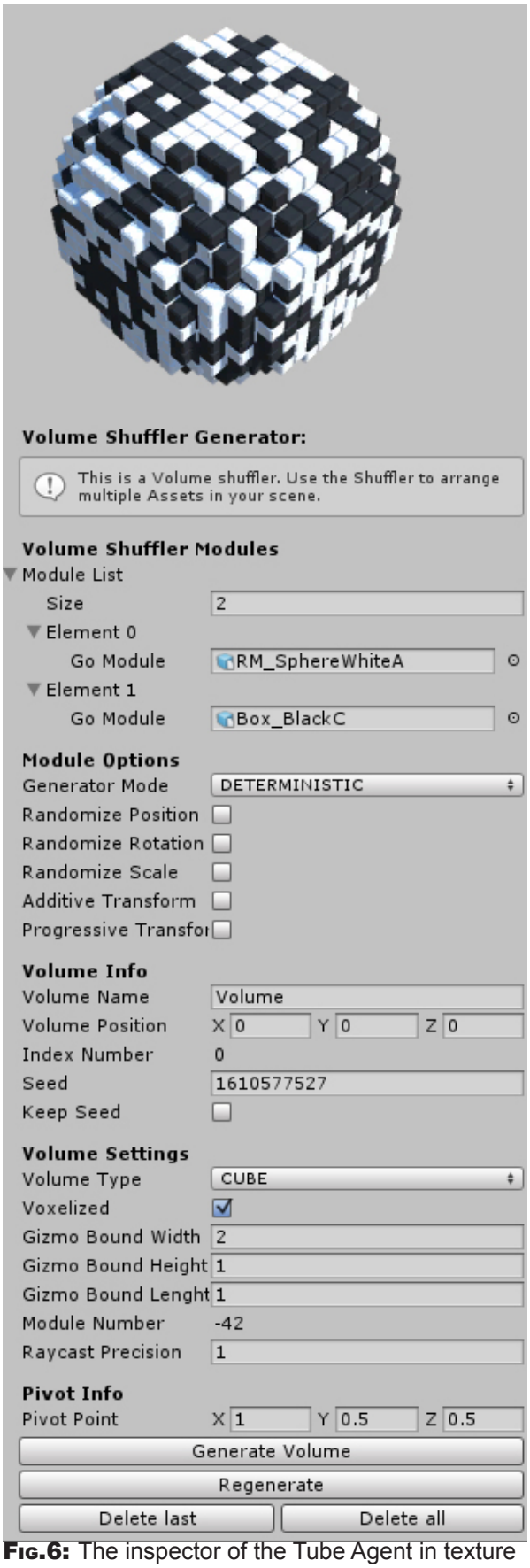

Fig.6: The inspector of the Tube Agent in texture mode.

The **Volume Shuffler** is a volume generator. It can be used to voxelize 3d meshs and project assets on surfaces. Besides those features it is also possible to use animation curves to modify asset volumes. It is also possible to use different generator modes to determine the asset creation process.

The **Generator Modes:**

- **SIMPLE_MAZE:** A random 90° rotation to the local y-axis will be added to the generated module. Resembling the generation mode of the simple maze.
- **DETERMINISTIC:** Sequential module generation. Very useful to generate complex module patterns.
- **RANDOM:** Random module generation.

Transform modifiers of the **Volume Shuffler:**

Modifiers define transformation values of position, rotation or scale. Transform modifiers have the following options:

- **RANDOMIZE POSITION:** If flagged true, enables options to set precise values to randomize the x, y and z position of the generated modules.
- **RANDOMIZE ROTATION:** If flagged true, enables options to set precise values to randomize the x, y and z rotation of the generated modules.
- **RANDOMIZE SCALE:** If flagged true, enables options to set precise values to randomize the x, y and z scale of the generated modules.
- **ADDITIVE TRANSFORM:** Adds a set value per iteration to a transform.
- **PROGRESSIVE TRANSFORM:** Powerful modifier which uses animation curves to modify asset transforms. The curves can be saved and loaded. The factor values must be higher or lower then zero to have an effect.

Volume settings of the **Volume Shuffler:**

Volumes define the space in which the assets are generated. Available volume settings are:

- **CUBE:** The volume is a Cube
- **SPHERE:** The volume is an perfect sphere.
- **MESH:** The volume is defined by the surface of an imported mesh. (Currently only works with surfaces)

Unique settings of **Voxelized/Non-Voxelized:**

Voxelized volumes have a predicted module number that can vary depending on the topology of the selected volume. For example, cubes will generate more GameObjects then spheres and meshs. The size of the volume defines the number of generated Gameobjects in the voxelization process. When the flag is set to Non-Voxelized, the GameObjects are randomly generated inside the volume of a cube or a sphere, or are generated on the surface of a mesh. In non-voxelized mode the number of GameObjects can be choosen freely.

Hint: Since the spawn point of the GameObjects in non-voxelized mode is random, the deterministic generator mode will not have the same result as before. But it can still be used to evenly scatter GameObjects in the volume.