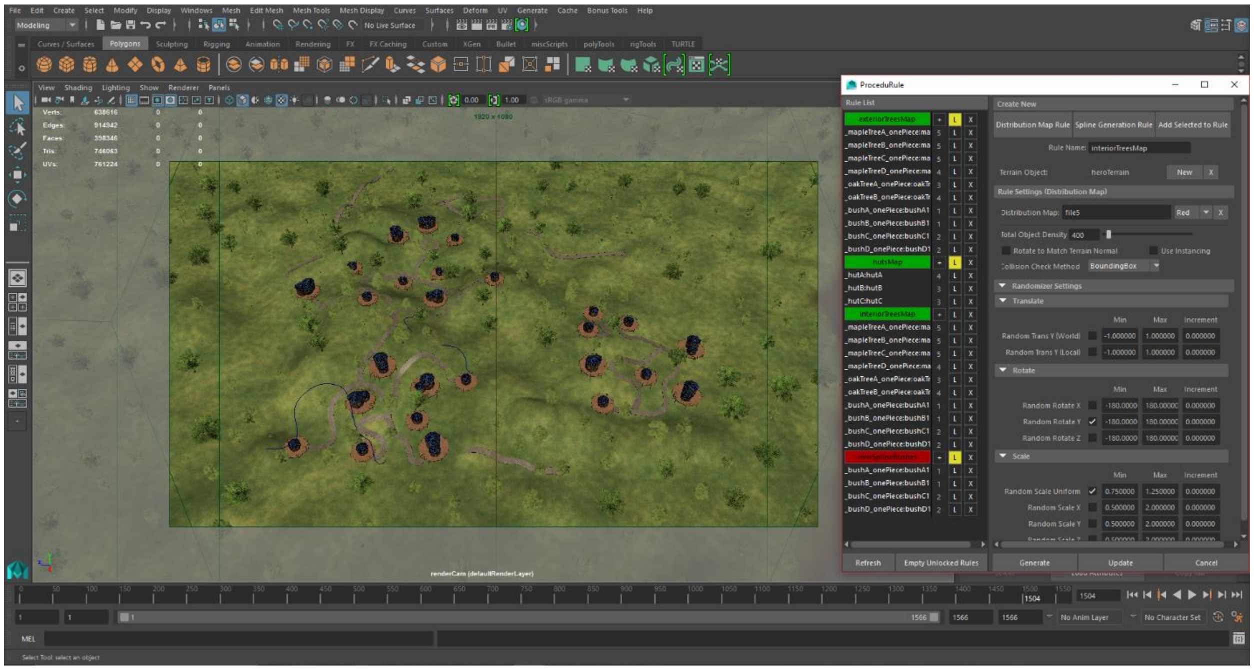
Overview of the ProceduRule Automation Tool



Rule List:
exteriorTreesMap
_mapleTreeA_onePiece:ma 5
_mapleTreeB_onePiece:ma 5
_mapleTreeC_onePiece:ma 5
_mapleTreeD_onePiece:ma 4
_oakTreeA_onePiece:oakTr 3
_oakTreeB_onePiece:oakTr 4
_bushA_onePiece:bushA1 1
_bushB_onePiece:bushB1 1
_bushC_onePiece:bushC1 2
_bushD_onePiece:bushD1 2
hutsMap
_hutA:hutA 4
_hutB:hutB 3
_hutC:hutC 3
interiorTreesMap
_mapleTreeA_onePiece:ma 5
_mapleTreeB_onePiece:ma 5
_mapleTreeC_onePiece:ma 5
_mapleTreeD_onePiece:ma 4
_oakTreeA_onePiece:oakTr 3
_oakTreeB_onePiece:oakTr 4
_bushA_onePiece:bushA1 1
_bushB_onePiece:bushB1 1
_bushC_onePiece:bushC1 2
_bushD_onePiece:bushD1 2
_bushA_onePiece:bushA1 1
_bushB_onePiece:bushB1 1
_bushC_onePiece:bushC1 2
_bushD_onePiece:bushD1 2

Rule Name: interiorTreesMap
Terrain Object: heroTerrain
Rule Settings (Distribution Map)
Distribution Map: file5 Red
Total Object Density 400
Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox

		Min	Max	Increment
Random Trans Y (World)		-1.000000	1.000000	0.000000
Random Trans Y (Local)		-1.000000	1.000000	0.000000
Random Rotate X		-180.0000	180.0000C	0.000000
Random Rotate Y	✓	-180.0000	180.0000C	0.000000
Random Rotate Z		-180.0000	180.0000C	0.000000
Random Scale Uniform	✓	0.750000	1.250000	0.000000
Random Scale X		0.500000	2.000000	0.000000
Random Scale Y		0.500000	2.000000	0.000000

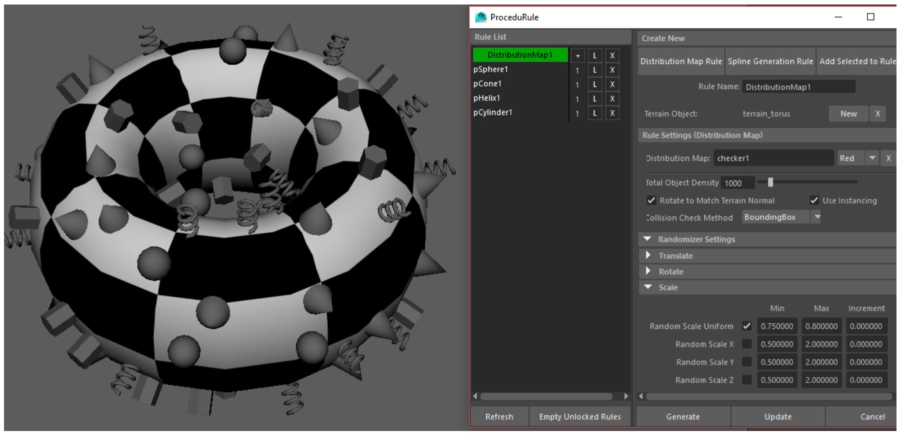
Fig. 1. A scene being created using ProceduRule in Autodesk Maya

Given the utility that a multi-purpose procedural modeling tool could provide for smaller studios and individual artists, I have created an environment automation tool for Autodesk Maya called ProceduRule. This tool was developed for Maya due to the software's leading reputation across multiple different industries and the existing limits of the software's current procedural modeling offerings. This new tool allows the user to propagate 3D models throughout a virtual scene by associating them with a series of "rules" that control how they are duplicated and placed. Each of these rules consist of a few required components: a user-defined selection of 3D assets, a surface onto which these models will be procedurally generated (aka the "terrain"), a guide that determines where the asset will be duplicated across the terrain, and a series of user settings that randomize instance transformation. The tool supports two types of "rules" that determine what kind of data will be used to guide asset placement along a given terrain. Once the user has defined these “rules”, assets can then be quickly generated and distributed across one or more terrain geometry with a single button press. Meshes created by ProceduRule are automatically organized in a hierarchy that groups instances of the same base mesh together and assembles all of these groups into a parent node associated with the user-defined ruleset that generated them. The tool also ensures that objects do not collide with one another when placed throughout the scene using efficient collision detection methods. After generation, the transformation settings for each rule can be altered and quickly reapplied to the procedurally-generated content until the desired look is achieved.

Methods of Asset Generation

ProceduRule generates assets along terrain using two distinct rule types; each places geometry throughout a scene using different kinds of data as input. The first of these is the Distribution Map rule which utilizes a black and white image as its guide when determining where assets should be placed. This image can be created within Maya using the built-in 3D Paint Tool or by using a variety of pre-existing texturing nodes to construct an appropriate data graphic. In addition, the image can be created using any additional image generation or manipulation software outside of Maya and can be imported as a 2D texture file. When a black and white texture or a channel from a color graphic is assigned to a Distribution Map rule, the image is mapped to the terrain object and is used to determine valid locations in which an asset can be instanced during the generation process.

When using a distribution map to propagate geometry along a terrain mesh, the terrain's UV coordinates determine the locations along its surface on which instanced assets may be placed. If the distribution map is white at that UV position, the asset can be duplicated there. A darker section of the map indicates a location that is more likely to remain empty after generating assets, and a black section of the map denotes areas of the terrain on which assets may not be placed at all. Theoretically, this functionality could be extended to allow variables within the 3D environment to be read as distribution maps. Some example of this behavior would be to propagate fallen leaves only on the top of a terrain mesh using a Y-axis texture operation or to place mushrooms in areas that receive less light by generating a graphic based on the terrain's surface luminance. This would further extend the high degree of control Distribution maps provide when populating a scene.



Rule List:

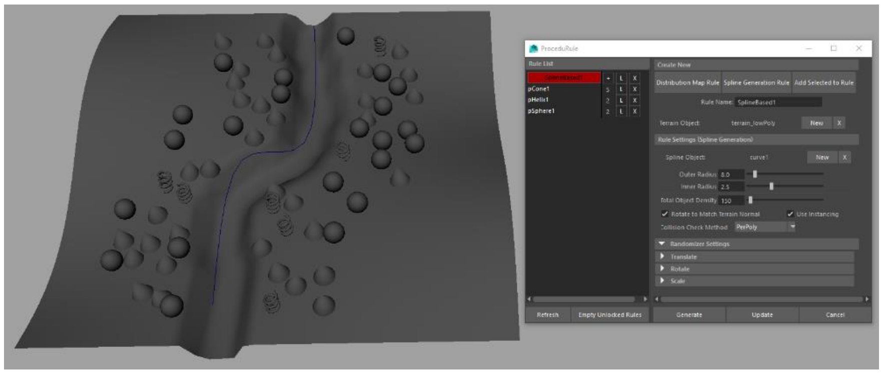
Rule			
DistributionMap1	+	L	X
pSphere1	1	L	X
pCone1	1	L	X
pHelix1	1	L	X
pCylinder1	1	L	X

Rule Name: DistributionMap1
Terrain Object: terrain_torus
Rule Settings (Distribution Map)
Distribution Map: checker1 Red
Total Object Density 1000
[✓] Rotate to Match Terrain Normal [✓] Use Instancing
Collision Check Method BoundingBox

		Min	Max	Increment
Random Scale Uniform	✓	0.750000	0.800000	0.000000
Random Scale X		0.500000	2.000000	0.000000
Random Scale Y		0.500000	2.000000	0.000000
Random Scale Z		0.500000	2.000000	0.000000

Fig. 2. Generated objects appear only in the white sections of this checker pattern using a Distribution Map rule.

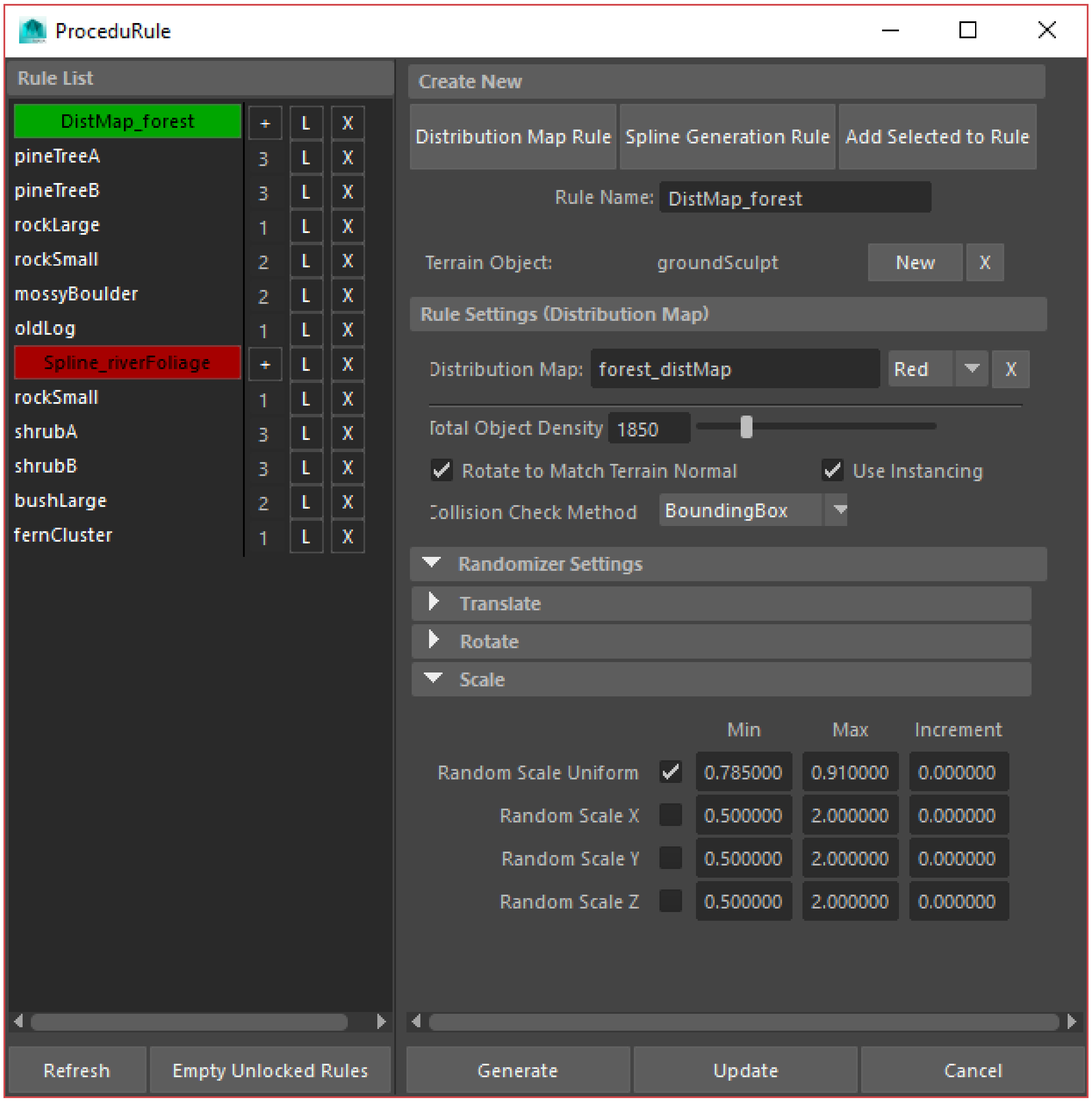
The second rule type in ProceduRule is the Spline Generation rule which restricts asset placement to a specified region along or near a given spline curve. Using either a NURBS or Bezier curve as the guide, users can define a path around which a supplied list of assets can be propagated. In its simplest form, this rule allows these assets to be placed randomly along a line on the terrain mesh and could be useful when placing a row of leaf clusters along a tree branch or when placing rocks in a winding river. The user can optionally define an outer radius around the spline within which assets may be duplicated, making the objects' placement appear more natural and randomized. An inner radius around the spline can additionally be defined to limit asset placement to the region between the inner and outer constraints. For example, the inner radius can be used to quickly place shrubs around a river but not in the river or to place debris along the edges of a room or hallway while leaving the center cleared away. If the radii values are increased enough, a guide spline's influence becomes less obvious and the placement of assets appears more natural and random.



Rule Name: SplineBased1
Terrain Object: terrain_lowPoly
Spline Object: curve1
Outer Radius 8.0
Inner Radius 2.5
Total Object Density 150
✓ Rotate to Match Terrain Normal ✓ Use Instancing
Collision Check Method PerPoly

Fig. 3. Propagated objects appear around a NURBS curve between a specified inner and outer radius using a Spline Generation rule.

Although ProceduRule currently offers two rules for controlling asset generation, a theoretical third functionality could allow for a combination of the two methods. The implementation of a Hierarchal Recursion system would enable rules to be associated with each other in a parent-child relationship, combining the benefits of each rule to form more versatile and specific rules. This hierarchy structure could additionally generate assets that, in turn, could become the terrain for secondary rules that recursively generate their own objects. For example, a user could create individual tree trunk, branch, and leaf assets. By assigning one rule to place the tree trunks on a terrain, a second rule to place branches on the trunks, and a third rule to place leaves on the branches, one could theoretically create a randomized forest from only a few simple assets. Challenges with implementing this functionality would include maintaining reasonable asset generation times when utilizing multiple recursion levels as well as ensuring that understanding and navigating this hierarchal interface was self-explanatory.



DistMap_forest	+	L	X
pineTreeA	3	L	X
pineTreeB	3	L	X
rockLarge	1	L	X
rockSmall	2	L	X
mossyBoulder	2	L	X
oldLog	1	L	X
Spline_riverFoliage	+	L	X
rockSmall	1	L	X
shrubA	3	L	X
shrubB	3	L	X
bushLarge	2	L	X
fernCluster	1	L	X

Rule Name: DistMap_forest

Terrain Object: groundSculpt

Rule Settings (Distribution Map)

Distribution Map: forest_distMap Red

Total Object Density 1850

✓ Rotate to Match Terrain Normal ✓ Use Instancing

Collision Check Method BoundingBox

		Min	Max	Increment
Random Scale Uniform	✓	0.785000	0.910000	0.000000
Random Scale X		0.500000	2.000000	0.000000
Random Scale Y		0.500000	2.000000	0.000000
Random Scale Z		0.500000	2.000000	0.000000

Fig. 4. The graphic user interface for ProceduRule.

Results of using ProceduRule Automation Tool

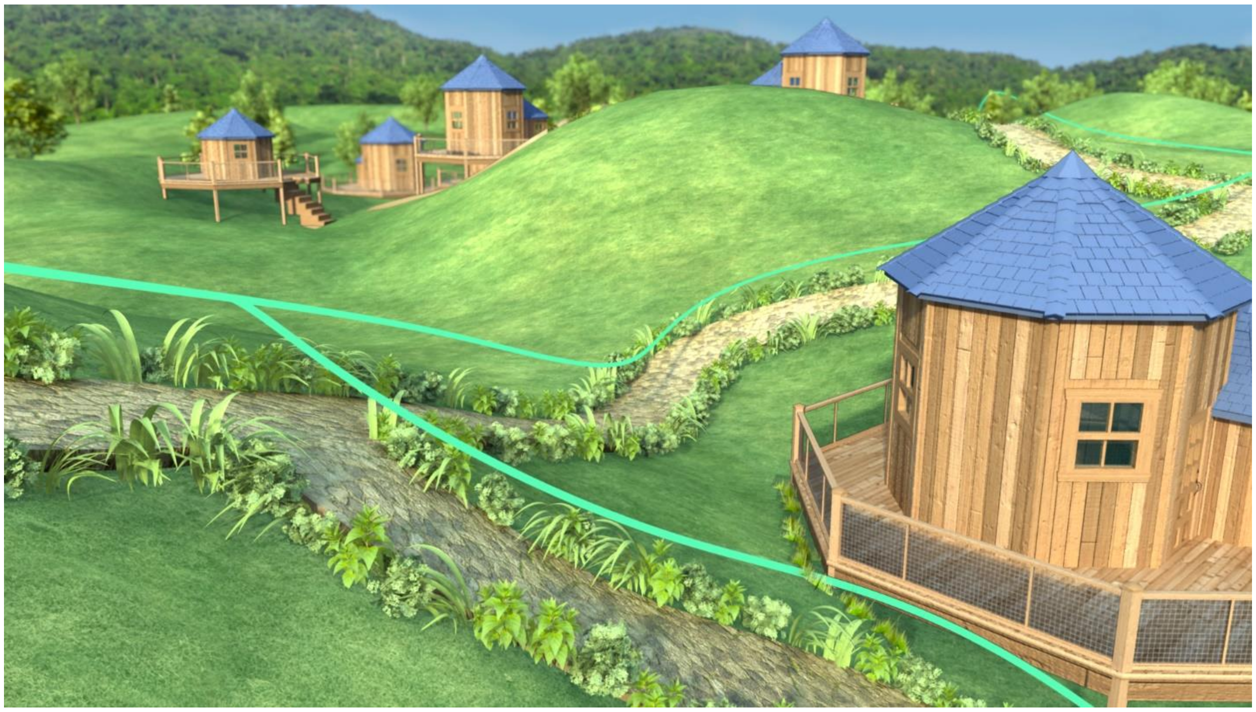


Fig. 5. Final beauty frame from visual component

In order to test the efficacy of ProceduRule, the tool was used to create a test environment. The goal was to make a small village nestled among rolling hills that were covered with trees and shrubs. In this environment, the buildings would be linked together by stone pathways, and there would be a river lined with bushes nearby. The environment would normally be time-consuming to produce due to the tedium of achieving the desired foliage density, variety, and placement by hand. A useful procedural modeling tool should significantly expedite such a process and should handle placing a wide variety of assets while ensuring that they do not collide unrealistically. In order to construct the environment, six tree assets, six varieties of shrub, and three versions of the hut were created ahead of time. The hills, stone pathways, and river would be added without ProceduRule, but the placement of the other assets was executed using the tool.

One large, subdivided plane was sculpted within Maya to create the rolling hills which would act as the foundation for the environment. That plane was split into two meshes: the inner terrain that would hold much of the important content (the huts, the river, etc.) and the outer terrain that surrounded this main area and contained mostly trees and shrubs. Dividing this plane was an optional decision that allowed for slightly more organization regarding which assets spawned where. As an additional bonus, this approach optimized the surface area that ProceduRule needed to process when placing each set of assets. The distribution map for the huts was painted using Maya's Artisan toolset; this texture defined the regions of the inner terrain where the buildings could be placed. Generating the huts only took a few seconds and could be repeated quickly with brief adjustments to the distribution map until the desired results were achieved. ProceduRule chose from the three hut varieties with which it was supplied based on the relative density controls set by the user. Each hut's rotation and scale were automatically randomized by the tool and required only minor manual adjustment afterwards.

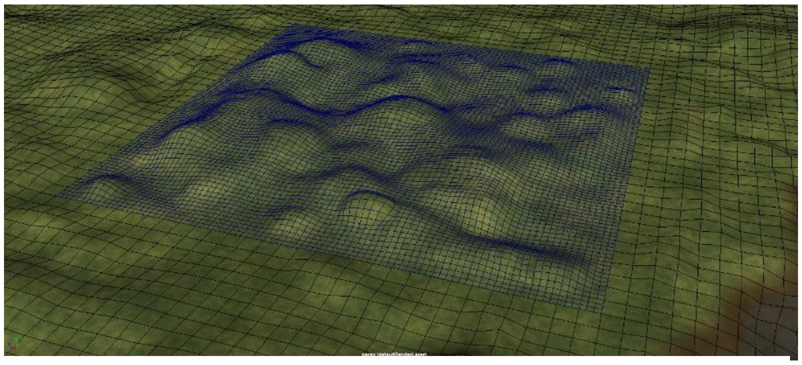


Fig. 6. The terrain mesh used for the village and forest generation, divided into two separate meshes.

The generation of trees and shrubs calculated more slowly, but using ProceduRule for this process still saved time and allowed for enhanced visual flexibility and scene organization. Three individual asset groups were generated to create this foliage: the trees and bushes on the outer terrain, the trees and bushes on the inner terrain, and the shrubs that would generate along the river. The first two of these groups were driven by distribution map rules while the latter was placed using a spline-based rule. The fluid simulation for the river itself was calculated and rendered in Houdini and would serve as a visual cue for the appearance of the bushes. In Maya, a NURBS curve was drawn over the river bed and, using the inner and outer radius controls, the shrubs were easily placed just beyond the area that the fluid touched. Controlling both the number of assets that ProceduRule generated and the relative proportion of each kind of asset was handled using the tool's density settings. Additionally, the collision accuracy was reduced for some assets in order to generate them more quickly when a large volume of meshes was needed. Automating the placement of the two largest tree groups took roughly thirty minutes each due to the high complexity of the meshes, the number spawned, and the need for collision checking. Over 700 assets and more than eight million polygons were placed in those rules. After the assets were generated, ProceduRule's hierarchical object grouping proved beneficial later when selecting objects to assign to render layers and when creating matte passes for each type of foliage.

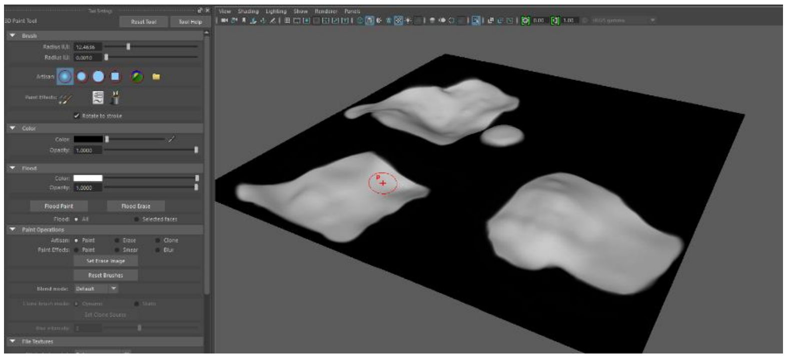


Fig. 7. The distribution map used to generate huts on the inner terrain, painted in Autodesk Maya using the built-in Artisan toolset.

After connecting many of the huts to one another by creating stone pathways between them, an unforeseen issue arose: these paths intersected with the ground in a way that created a hard, undesirable visual line. To solve this problem, the NURBS curves that were used to model the original paths were repurposed for new spline-based ProceduRule rules. These rules were responsible for placing small bushes and shrubs along the border of each path to cover up this jarring edge. ProceduRule generated 2,830 assets (a total of more than 4.56 million polygons) along nine paths with collision checking disabled and spent approximately 2-5 minutes per path. Minor positional and rotational tweaks were made manually for hero assets in order to get the desired look for certain shots, but approximately 95% of those assets required no additional adjustments. One of the shrubs needed to be slightly altered after ProceduRule had made hundreds of duplicates throughout the scene. The tool's optional instancing functionality allowed tweaks to the original model's geometry to automatically propagate to the thousands of duplicated shrubs.

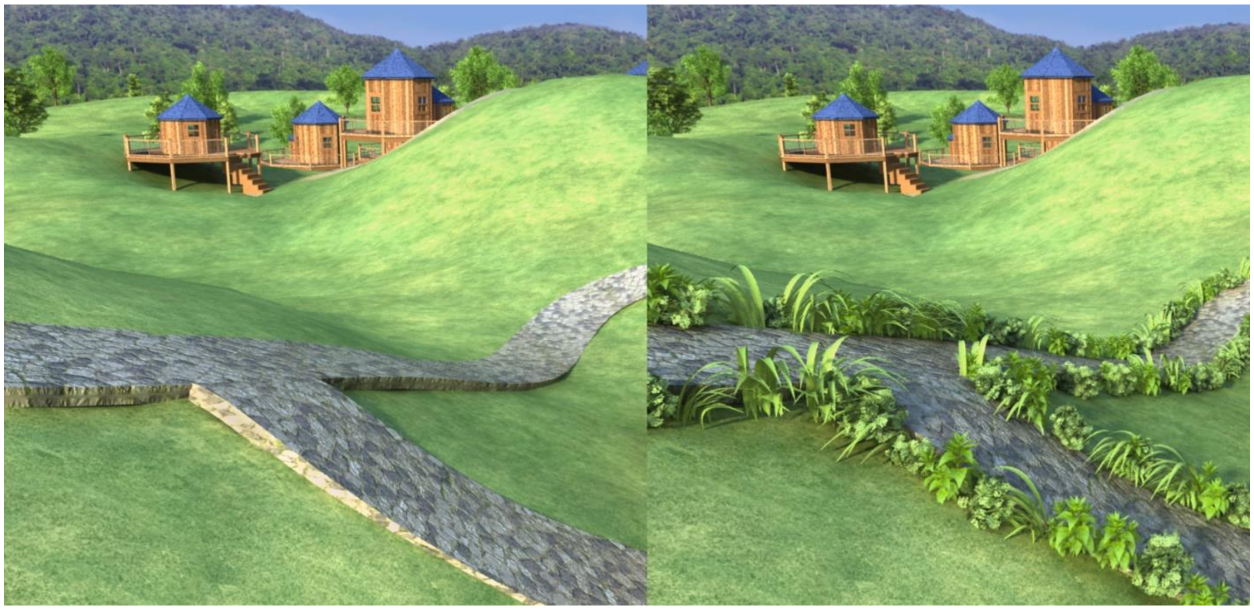


Fig. 8. Creating foliage along the stone walkways greatly improved their visual appearance.

While the use of ProceduRule made creating this environment a much easier and faster task than was otherwise possible, the experience was not without its limitations. Firstly, when high-polygon models were propagated in large numbers with collision detection enabled, the overall calculation time grew significantly. ProceduRule especially slowed down when duplicating object groups under these conditions, but the tool created warnings that suggested combining the group into a single model to prevent the additional slowdown. The time spent generating assets was still far less than the time it would take to complete the same process without an automation tool, and the artist could still work on other tasks outside of Maya while ProceduRule was processing, but the delay highlighted the cost of ProceduRule's collision detection algorithm. Secondly, at no fault of the automation script, Maya slowed down considerably when selecting or operating on large numbers of objects. This problem became worse when render layers were introduced as the scene had to be reprocessed by Maya each time a render layer was selected, causing prohibitively long wait times in certain cases. By combining objects that were generated by ProceduRule into a single model, this problem could be mitigated at the cost of losing post-generation ProceduRule functionality on the given assets.

Overall, the use of ProceduRule resulted in the desired environment being created much more quickly than one could without using the tool. The tool's organization, flexibility, and fast iteration times allowed the software to be useful both for creating the majority of the environment as well as troubleshooting unexpected issues.

Conclusion

Based on its design goals, robust functionality, and ability to speed up the 3D environment creation workflow, ProceduRule demonstrates that thoughtfully-crafted automation tools can successfully fulfill the needs of smaller studios and freelance artists who lack the resources of larger studios. These utilities enable the creation of massive, densely-populated worlds in reasonable timeframes which can mean the difference between whether a project is viable or impossible. By expanding the possibilities of what can be accomplished by smaller studios or within tighter schedules, the capabilities of the computer graphics industry expand as well, especially in commercial or broadcast environments where time constraints can dramatically limit a project's scope or quality. ProceduRule helps demonstrate the efficiency of working with procedurally-generated content and how automation tools enable 3D studios and artists across all vocations to expand beyond their current creative boundaries.

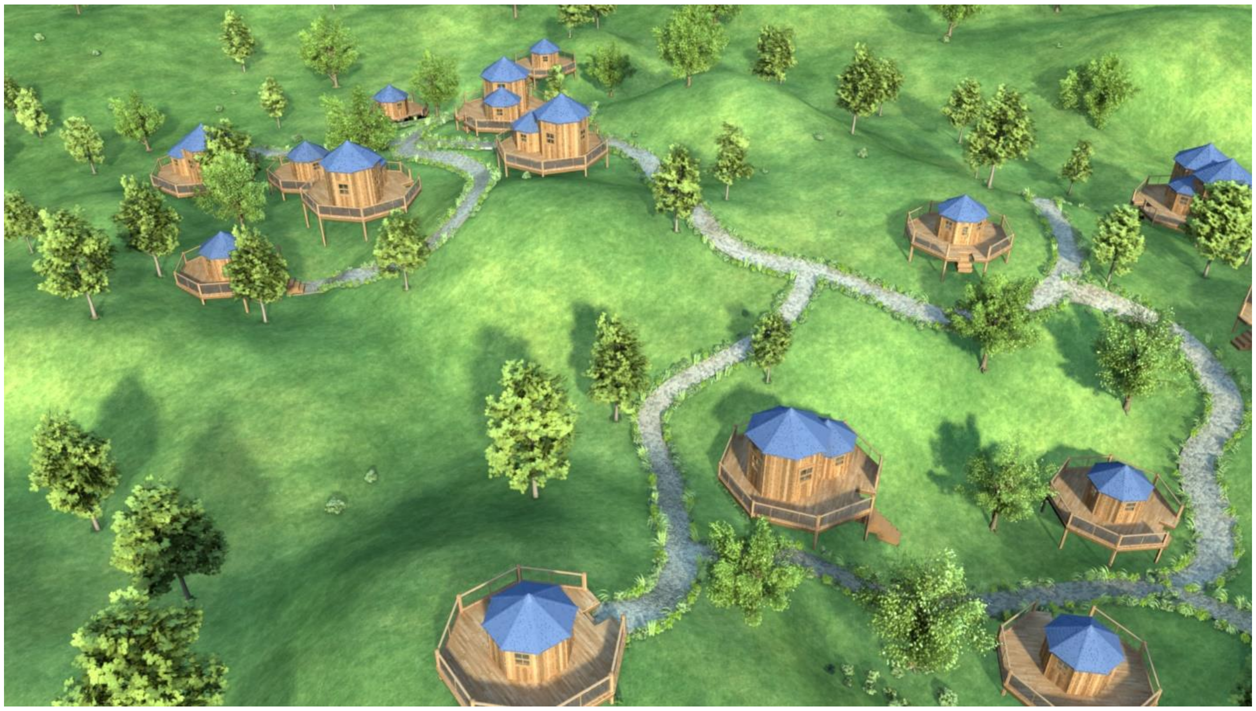


Fig. 9. A finalized shot of the village created using ProceduRule.

Appendix A: Written Tutorial

I. Introduction

ProceduRule is a procedural modeling tool that specializes in propagating polygonal meshes across large environments. This is accomplished by creating rules, selecting a terrain object, assigning geometry to be generated via those rules, and then adjusting transformation parameters as needed. In this tutorial, you will learn how to navigate the user interface, how to create and use the Distribution Map rule type, and how to create and use the Spline Generation rule type. Before covering this material, a list of important terminology must be defined:

- Rule: A set of guidelines which define how assets are created by ProceduRule. Rules have a type (Distribution Map or Spline Generation) and contain various user settings (collision detection, random transformation, etc.) that control how the rule places assets.
- Asset: Any element that is part of a digital media composition. In this tutorial, “asset” will refer to any polygonal mesh in a 3D scene. In ProceduRule, these are assigned to rules, and duplicates or instances of these assets are generated along terrain objects.
- Terrain Object: A single polygonal mesh on which the duplicates or instances will be generated.
- Distribution Map: A texture in Maya that determines where duplicates or instances are allowed to generate. This refers to either a black-and-white texture or a single channel from a color texture. On distribution maps, a black (or zero) value corresponds to areas

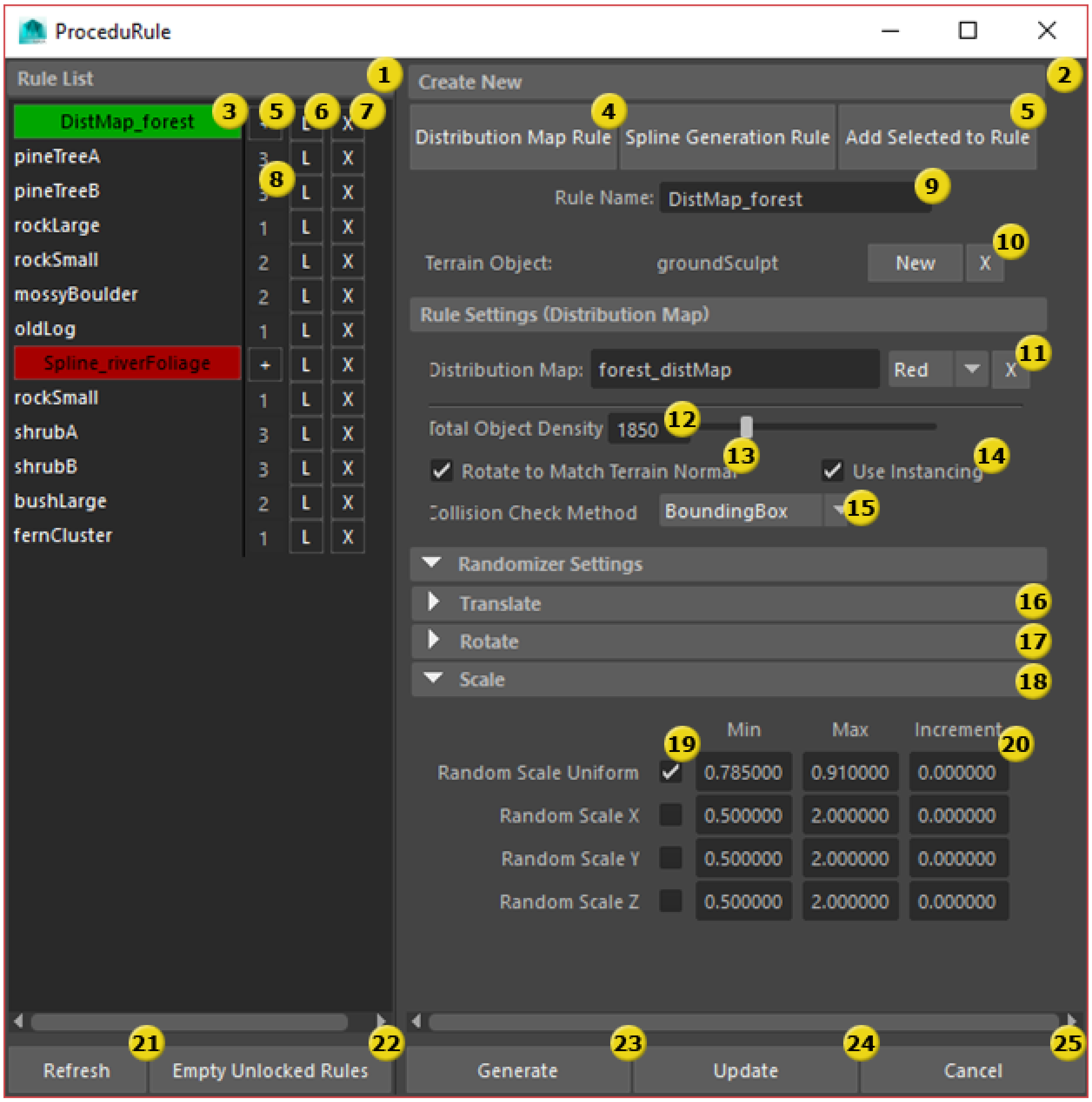
where assets cannot generate, and a white (or one) value corresponds to areas where assets can generate.

- Spline: A NURBS curve or Bezier curve within Maya. These are used in “Spline Generation” rules to define where assets may be placed.
- Duplicate: An object within Maya that was created by copying another object’s shape and transformation data. Once a duplicate object has been created, it is completely independent from its source object. In ProceduRule, these objects will always be copies of polygonal meshes that the tool has generated when “Use Instancing” is disabled.
- Instance: An object within Maya with unique transformation data that shares another object’s shape data. An instance is dependent on the geometry of its source (or parent) object, and all changes made to the geometric structure of the parent will also affect the children objects. In ProceduRule, these objects will always be copies of polygonal meshes that the tool has generated when “Use Instancing” is enabled.
- Generate: The act of creating copies of the geometry associated with one or more rules. The generation process assumes that duplicates or instances previously associated with each rule will automatically be deleted and replaced with the new assets that are being generated. “Generating a rule” refers to generating all assets associated with that rule.
- Update: To reapply new, randomly-generated transformation settings to the objects within one or more rules without actually generating new objects.
- Lock: To disable generation or updates for specific rules or certain assets in these rules.
- Total Object Density: The maximum number of duplicates or instances that a rule can create.

- Relative Density: The priority of one asset compared to all others in a given rule. Assets with higher priorities have a higher chance of appearing when a rule is generated.

II. Navigating the User Interface of ProceduRule

Below is a breakdown of the various elements that comprise the UI of ProceduRule:



DistMap_forest	+	L	X
pineTreeA	3	L	X
pineTreeB	3	L	X
rockLarge	1	L	X
rockSmall	2	L	X
mossyBoulder	2	L	X
oldLog	1	L	X
Spline_riverFoliage	+	L	X
rockSmall	1	L	X
shrubA	3	L	X
shrubB	3	L	X
bushLarge	2	L	X
fernCluster	1	L	X

Rule Name: DistMap_forest

Terrain Object: groundSculpt

Distribution Map: forest_distMap Red

Total Object Density 1850

✓ Rotate to Match Terrain Normal ✓ Use Instancing

Collision Check Method BoundingBox

		Min	Max	Increment
Random Scale Uniform	✓	0.785000	0.910000	0.000000
Random Scale X		0.500000	2.000000	0.000000
Random Scale Y		0.500000	2.000000	0.000000
Random Scale Z		0.500000	2.000000	0.000000

1. <u>Rule List:</u> The left half of the ProceduRule interface. This displays the rules that have been created in this Maya scene file as well as each asset that has been assigned to those rules (which are displayed in a list below each rule). This pane is used to assign or remove assets to rules, to select rules to be edited in the Settings Panel, to adjust the relative density of a rule's assets, and to lock or unlock rules or assets.
2. <u>Settings Panel:</u> The right half of the ProceduRule interface. This displays the various parameters that define how a rule behaves. This pane is used to assign terrain objects to rules, assign distribution maps or splines to rules, set a rule's total object density, adjust collision checking and instancing settings, and alter randomized transformation settings.
3. <u>Rule:</u> A user-created rule. If clicked, this makes that rule active by bringing its settings up in the Settings Panel.
4. <u>New Rule Buttons:</u> These, when clicked, create a new rule of the given type. If an asset is selected when a rule is created, this is automatically assigned as the rule's Terrain Object.
5. <u>Add Selected To Rule Button:</u> If an asset is selected when this button is clicked, it will be assigned to the currently active rule.
6. <u>Lock Rule / Lock Asset Button:</u> These, when clicked, lock or unlock the rule or asset which sits along the same horizontal row as the button. As long as a rule is locked, it is assumed that all of its assets are locked as well.
7. <u>Remove Rule Button / Remove Asset Button:</u> If this is next to a rule, this deletes the rule. If this is next to an asset, it removes it from the associated rule.

8. Relative Density Box: Displays the relative density of the associated asset within its parent rule. If clicked, the user can edit these values directly. When the rule is generated, the value for each relative density compares to the others as a ratio, roughly determining the proportion of each kind of asset that will be generated for that rule.
9. Rule Name Box: Allows the user to change the name of the active rule in the Rule List for organizational purposes.
10. Terrain Object Buttons: These allow you to define a rule's terrain object. The "New" button sets the terrain object to be whichever polygonal mesh is currently selected. The "X" button removes the current terrain object from this rule.
11. Distribution Map or Spline Settings: Allows the user to assign a distribution map or spline to the active rule. This setting looks different depending on the currently-selected rule; the distribution map version is shown in the above image.
    a. If this is a Distribution Map rule, the user must type the name of the texture node within Maya and press ENTER to set the distribution map. The user can then either choose which channel of the texture to use or can click the "X" button to empty this value.
    b. If this is a Spline Generation rule, the user assigns a new Spline to the rule by selecting a NURBS or Bezier curve and clicking the associated "New" button in the Settings Panel.
12. Total Object Density Box and Slider: Allows the user to set the rule's total object density (the maximum number of duplicates or instances that the rule can generate).

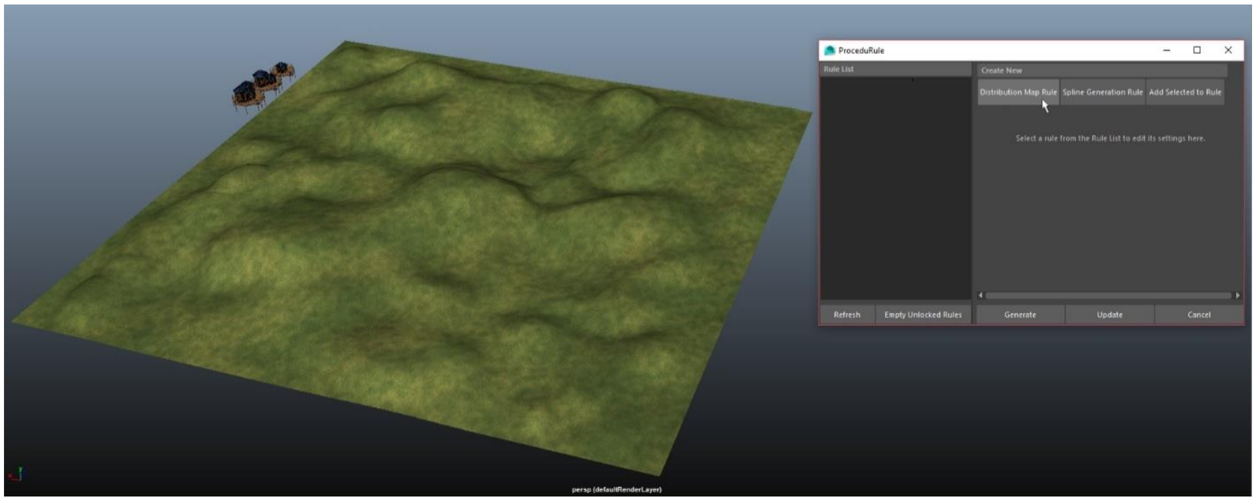
13. Rotate to Match Terrain Normal: If this box is checked, propagated geometry will be rotated to conform to the curvature of the terrain object, pointing in the direction of its surface normal. If this box is unchecked, propagated geometry will be oriented with a default rotation of 0 in all three axes (not including the effects of randomizer settings).
14. Use Instancing: Enabling this option creates instances instead of duplicates when generating rules.
15. Collision Check Method Dropdown Menu: This allows the user to select from one of three collision checking methods. Collision checking in ProceduRule removes inter-penetrating assets within the same rule to ensure that objects do not collide with each other. The three methods are:
    a. Disabled: No collision checking occurs.
    b. Bounding Box: If objects' bounding boxes overlap, then the objects are colliding with one another. This method is relatively fast but can be less accurate.
    c. Per-Poly Collision: If one object has faces that overlap with another object, then the objects are colliding. This method is relatively slow but is very accurate.
16. Translate Randomizer Settings: These settings alter the vertical placement of objects generated by this rule. The user can randomize a world-space offset (an additional Y translation that ignores the object's rotation) and/or a local-space offset (an additional Y translation that moves the object "upward" based on the object's rotation).
17. Rotate Randomizer Settings: These settings alter the rotation of objects generated by this rule along the X, Y, or Z axes.

18. Scale Randomizer Settings: These settings alter the scale of objects generated by this rule. The user can adjust the scale both uniformly and along the X, Y, or Z axes.
19. Enable Randomizer Checkboxes: Each row of randomizer settings does nothing unless this checkbox is enabled for the associated row.
20. Min, Max, and Increment Values: These boxes determine how an enabled randomizer setting operates. When enabled, a random value between the Min and Max numbers will be selected for each duplicate or instance associated with the active rule. If the Increment value is greater than 0, then all selected values will be divisible by the provided value (ex.: an asset may be set to only rotate in 90-degree increments).
21. Refresh Button: If the Rule List no longer accurately reflects what is currently within the opened Maya scene, clicking this button will update the Rule List to match the scene. This can happen if a terrain object is deleted from the scene, if assets or rules are deleted manually by the user, or if a new scene is opened while ProceduRule is open.
22. Empty Unlocked Rules Button: If clicked, this deletes all duplicates or instances in all currently unlocked rules. This does not affect duplicates or instances whose parent assets are locked.
23. Generate Button: If clicked, deletes all duplicates or instances in all currently unlocked rules and generates new ones according to the latest user settings for each rule.
24. Update Button: If clicked, applies new randomized transformation settings to the duplicates and instances within one or more rules without actually generating new objects.
25. Cancel Button: Closes ProceduRule.

III. Populating an Environment Using Distribution Map Rules

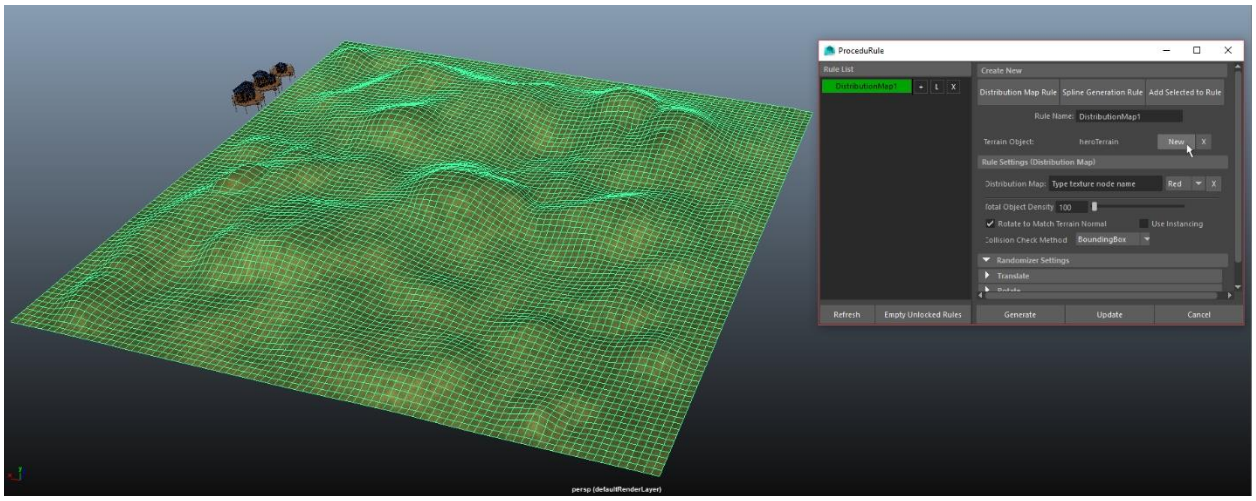
The Distribution Map rule type places objects based on a supplied Maya texture. This can be a 2D texture node, a texture painted within Maya using the 3D Paint tool, or an external image file that has been imported into Maya using a File node. Below is a step-by-step tutorial on how to create, set up, and successfully generate objects using a Distribution Map rule.

Step 1: Create a new Distribution Map rule using the button in the Settings Panel.



Select a rule from the Rule List to edit its settings here.

Step 2: Assign your terrain to the rule (select it and clicking “New” in the Settings Panel).



DistributionMap1

Rule Name: DistributionMap1

Terrain Object: heroTerrain

Distribution Map: Type texture node name | Red

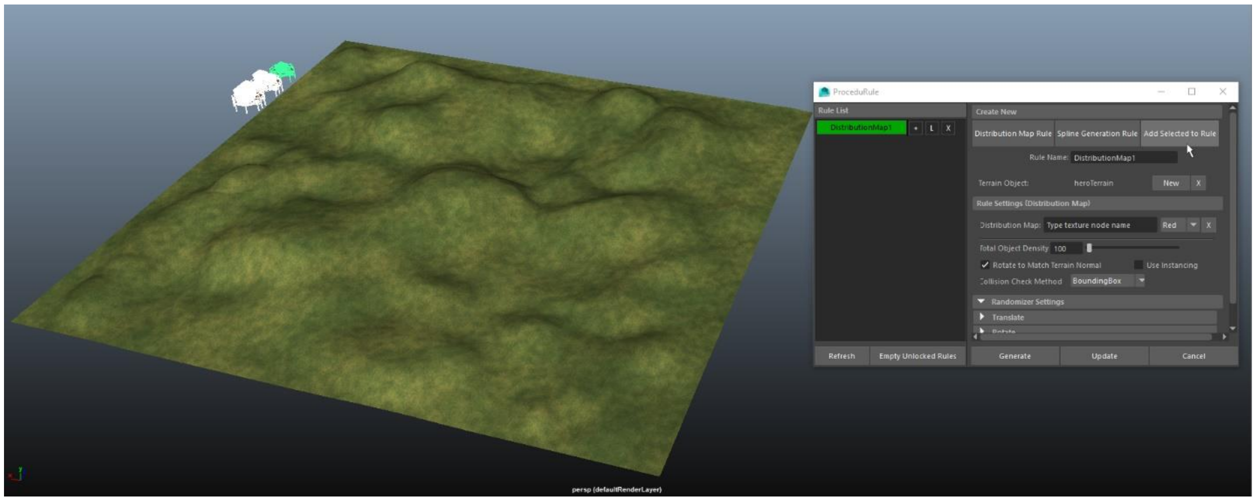
Total Object Density 100

✓ Rotate to Match Terrain Normal

Use Instancing

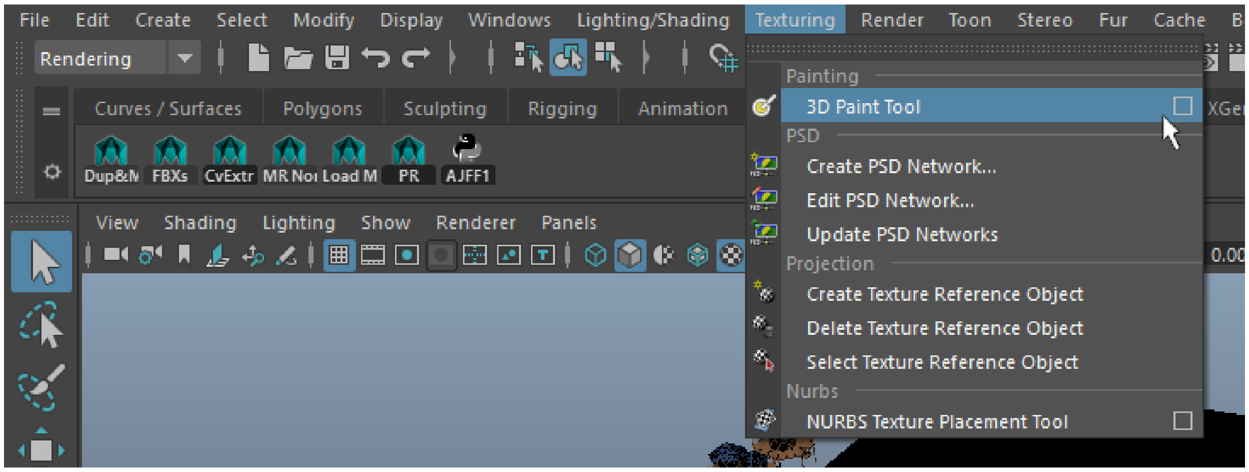
Collision Check Method BoundingBox

Step 3: Add the assets you want to procedurally generate to the rule by selecting them and pressing the "+" sign by the rule or by pressing "Add Selected to Rule" in the Settings Panel.

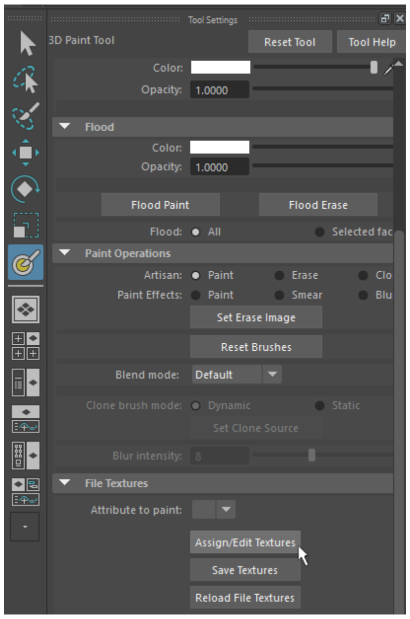


Rule Name: DistributionMap1
Terrain Object: heroTerrain
Distribution Map: Type texture node name Red
Total Object Density 100
✓ Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox

Step 4: Create your distribution map and assign it to the rule. An easy way to create a distribution map is by painting one using the 3D Paint Tool built into Maya.



This tool can be found in the "Rendering" menu set under "Texturing".

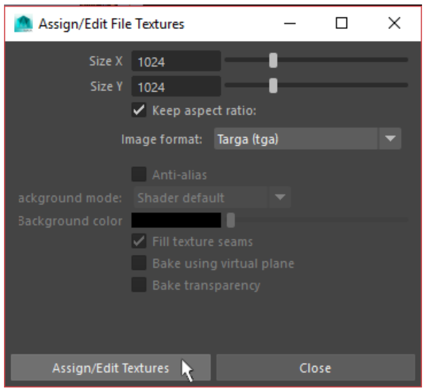

3D Paint Tool

Color:
Opacity: 1.0000

Flood
Color:
Opacity: 1.0000
Flood: ● All ○ Selected fac

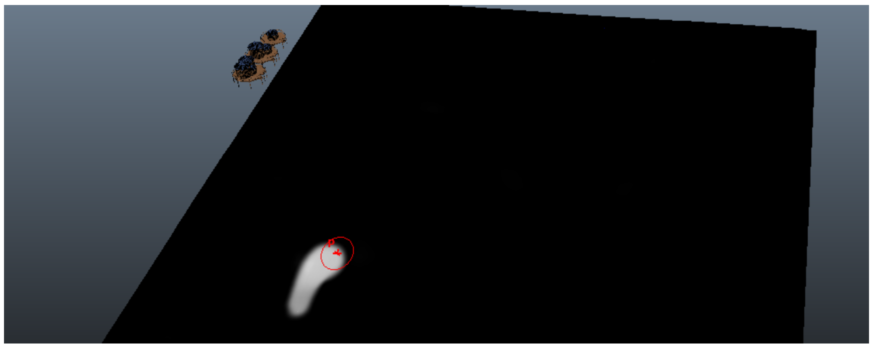
Paint Operations
Artisan: ● Paint ○ Erase ○ Clo
Paint Effects: ○ Paint ○ Smear ○ Blu
Blend mode: Default
Clone brush mode: ○ Dynamic ○ Static
Blur intensity: 8

File Textures
Attribute to paint:

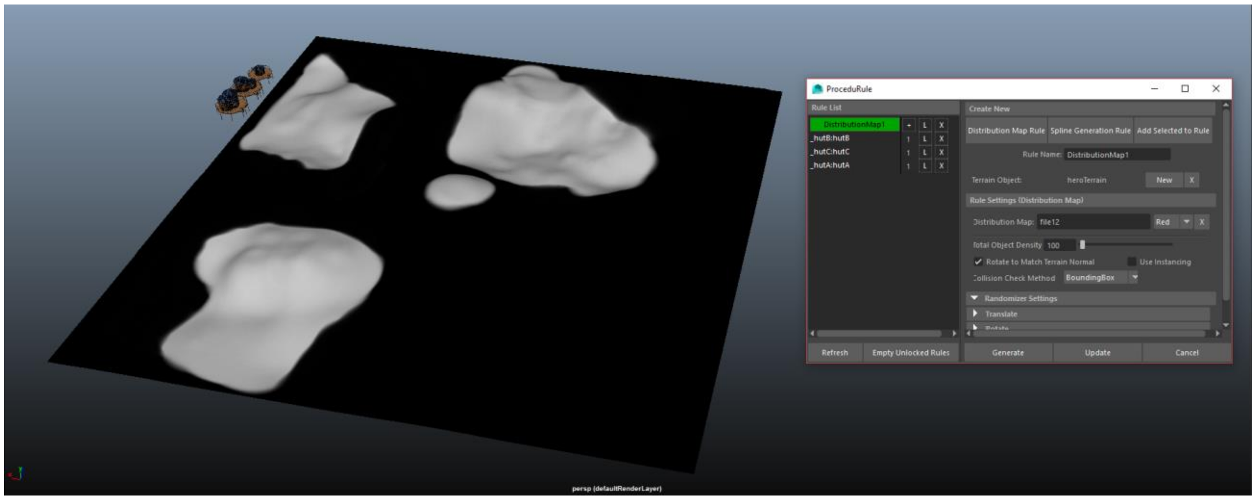

Assign/Edit File Textures

Size X 1024
Size Y 1024
✓ Keep aspect ratio:
Image format: Targa (tga)
Anti-alias
Background mode: Shader default
Background color
✓ Fill texture seams
Bake using virtual plane
Bake transparency

Click on “Assign/Edit Textures” in the 3D Paint Tool settings to create a new 3D Paint Tool texture, then specify its size and format. To save your changes, click “Save Textures”.



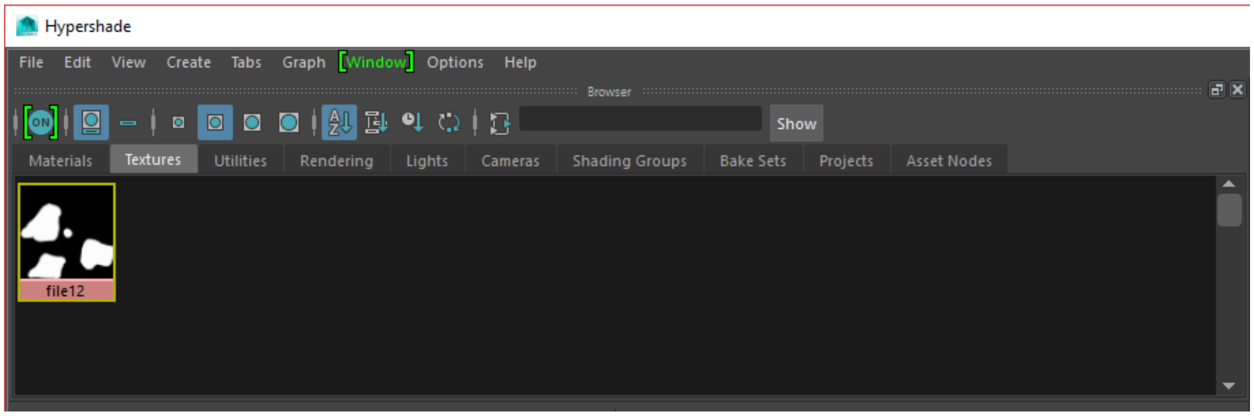
To assign this texture to your new rule, type the name of the associated Maya texture node in the Settings Panel and press ENTER.


ProceduRule

Rule List
DistributionMap1	+	L	X
_hutB:hutB	1	L	X
_hutC:hutC	1	L	X
_hutA:hutA	1	L	X

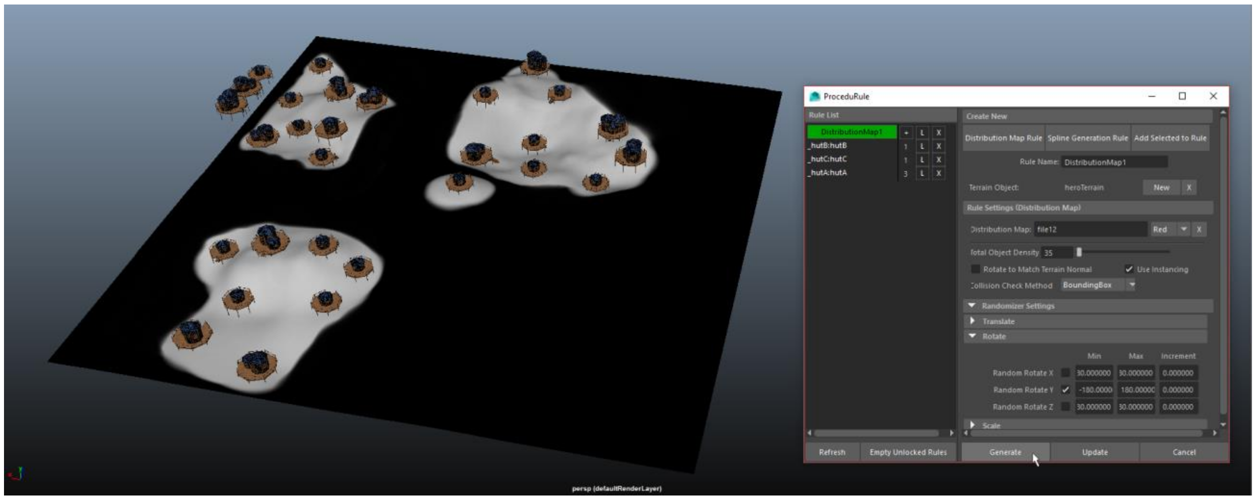
Create New
Rule Name: DistributionMap1
Terrain Object: heroTerrain
Rule Settings (Distribution Map)
Distribution Map: file12 Red
Total Object Density 100
✓ Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox

Sometimes it is helpful to apply this texture to a material and assign it to the Terrain Object as additional visual confirmation. You can find the names of all of the current scene's texture nodes in Maya's Hypershade editor.


Hypershade
File Edit View Create Tabs Graph Window Options Help
Browser
Show
Materials Textures Utilities Rendering Lights Cameras Shading Groups Bake Sets Projects Asset Nodes
file12

Step 5: Set the remaining parameters in the Settings panel to suit your needs. Be sure to define the Total Object Density and your desired Collision Check Method. Altering the other variables are optional but may improve the visual quality of the tool's output.

Step 6: Click "Generate", and wait for your geometry to appear. Congratulations, you have successfully created and used a Distribution Map rule!



Rule List			
DistributionMap1	+	L	X
_hutB:hutB	1	L	X
_hutC:hutC	1	L	X
_hutA:hutA	3	L	X

Rule Name: DistributionMap1

Terrain Object: heroTerrain

Rule Settings (Distribution Map)

Distribution Map: file12 Red

Total Object Density 35

Rotate to Match Terrain Normal ✓ Use Instancing

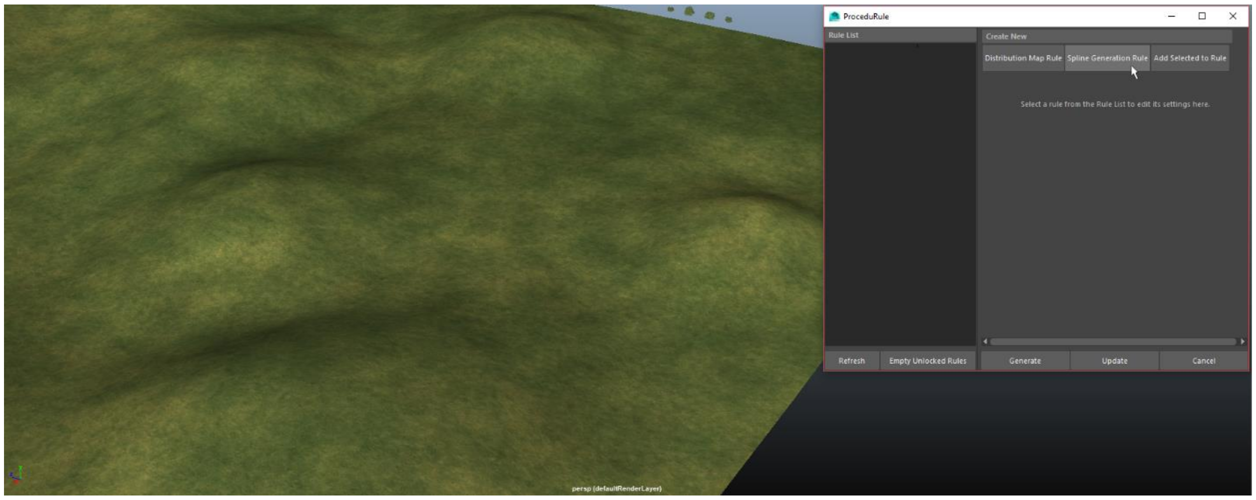
Collision Check Method BoundingBox

		Min	Max	Increment
Random Rotate X		30.000000	30.000000	0.000000
Random Rotate Y	✓	-180.0000	180.00000	0.000000
Random Rotate Z		30.000000	30.000000	0.000000

IV. Populating an Environment Using Spline Generation Rules

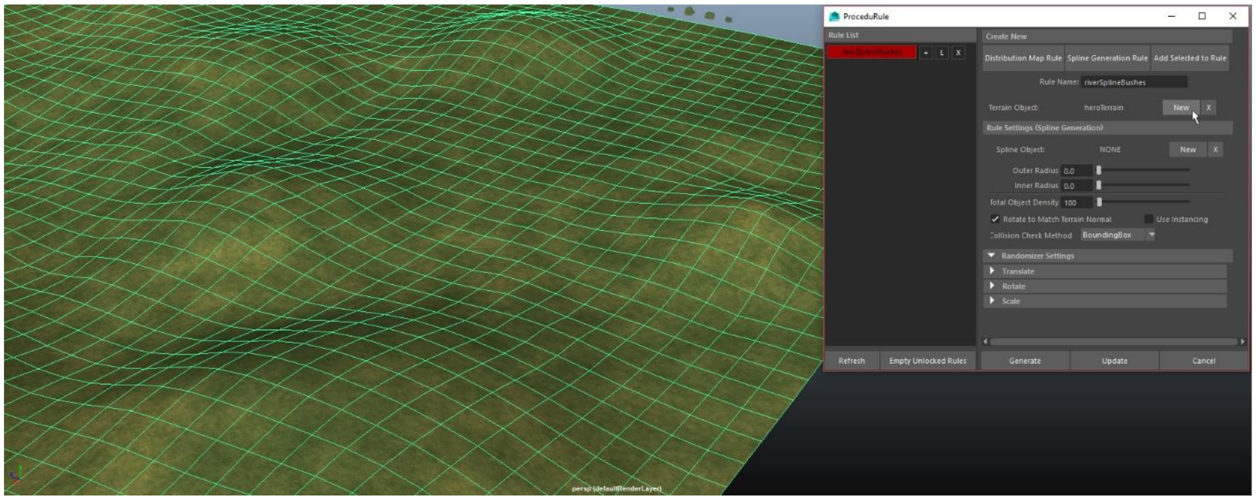
The Spline Generation rule type places objects based on a supplied NURBS or Bezier curve. This spline is placed above the terrain, and assets are procedurally generated at points on the terrain which are beneath and in close proximity to it. The surface area in which assets are allowed to appear can be expanded and constricted using two additional settings: the outer radius and inner radius. The outer radius expands the area around the spline in which objects can be created. The inner radius specifies an area beneath the spline in which objects must not appear. In order to use the inner radius, an outer radius must be specified and must be larger than the inner radius. Below is a step-by-step tutorial on how to create, set up, and successfully generate objects using a Spline Generation rule.

Step 1: Create a new Spline Generation rule using the button in the Settings Panel.



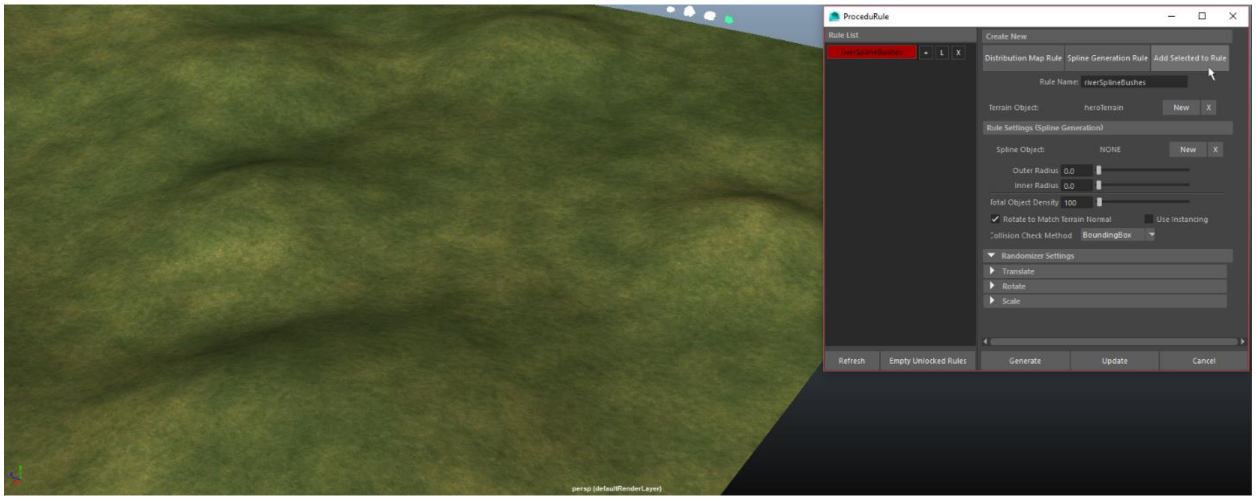
Select a rule from the Rule List to edit its settings here.

Step 2: Assign your terrain to the rule (select it and clicking “New” in the Settings Panel).



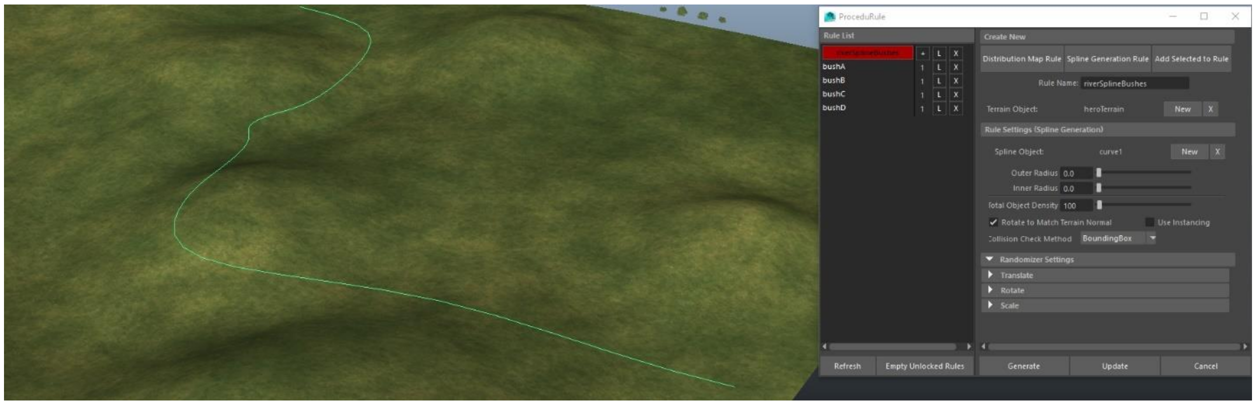
Rule Name: riverSplineBushes
Terrain Object: heroTerrain
Spline Object: NONE
Outer Radius 0.0
Inner Radius 0.0
Total Object Density 100
✓ Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox

Step 3: Add the assets you want to procedurally generate to the rule by selecting them and pressing the “+” sign by the rule or by pressing “Add Selected to Rule” in the Settings Panel.



Rule Name: riverSplineBushes
Terrain Object: heroTerrain
Spline Object: NONE
Outer Radius 0.0
Inner Radius 0.0
Total Object Density 100
✓ Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox

Step 4: Create your spline and assign it to the rule. An easy way to create a spline is by drawing a NURBS curve using Maya's CV, EP, or Bezier curve tools (Create > Curve tools).

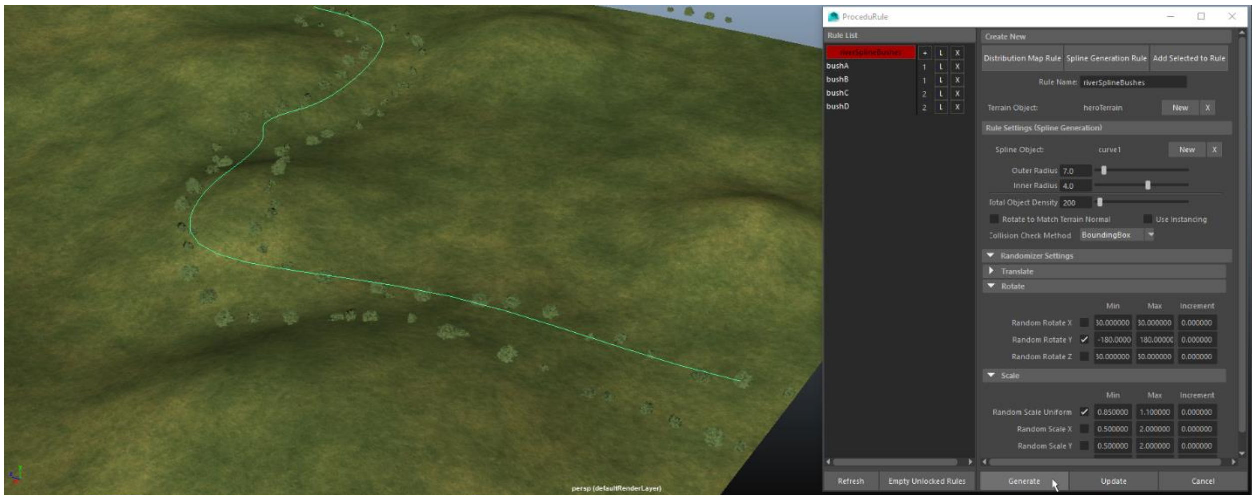


Rule List			
riverSplineBushes	+	L	X
bushA	1	L	X
bushB	1	L	X
bushC	1	L	X
bushD	1	L	X

Rule Name: riverSplineBushes
Terrain Object: heroTerrain
Rule Settings (Spline Generation)
Spline Object: curve1
Outer Radius 0.0
Inner Radius 0.0
Total Object Density 100
✓ Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox
Randomizer Settings
Translate
Rotate
Scale

Step 5: Set the remaining parameters in the Settings panel to suit your needs. Be sure to define the Outer Radius and Inner Radius as needed, and don't forget to set both the Total Object Density and your desired Collision Check Method. Altering the other variables are optional but may improve the visual quality of the tool's output.

Step 6: Click "Generate", and wait for your geometry to appear. Congratulations, you have successfully created and used a Spline Generation rule!



Rule List			
riverSplineBushes	+	L	X
bushA	1	L	X
bushB	1	L	X
bushC	2	L	X
bushD	2	L	X

Rule Name: riverSplineBushes
Terrain Object: heroTerrain
Rule Settings (Spline Generation)
Spline Object: curve1
Outer Radius 7.0
Inner Radius 4.0
Total Object Density 200
Rotate to Match Terrain Normal
Use Instancing
Collision Check Method BoundingBox
Randomizer Settings
Translate
Rotate

		Min	Max	Increment
Random Rotate X		30.000000	30.000000	0.000000
Random Rotate Y	✓	-180.0000	180.00000	0.000000
Random Rotate Z		30.000000	30.000000	0.000000

Scale

		Min	Max	Increment
Random Scale Uniform	✓	0.850000	1.100000	0.000000
Random Scale X		0.500000	2.000000	0.000000
Random Scale Y		0.500000	2.000000	0.000000