

YAPP - YET ANOTHER PREFAB PAINTER

Version 1.4.4

About this Document	2
Product Overview	3
The Idea	4
Features	5
Getting Started	5
Modes	10
Brush Mode	11
Brush Mode Distribution	14
Distribution: Center	18
Distribution: Poisson	23
Distribution: Fluent (with Preview)	26
Spline Mode	28
Interaction Mode	31
Operations Mode	33
Prefab Templates	35
Prefab Settings	37
Specific Use Cases	41
Align along Terrain Normal with random rotation offset	41
Rotation 0..360 vs -180..180	42
Integrations	43
Vegetation Studio Pro Integration	43
Frequently Asked Questions	44
Credits	45

About this Document

Please find the most recent documentation online at:

<https://bit.ly/yapp-doc>

Product Overview

YAPP is Yet Another Prefab Painter. YAPP can be used to conveniently and quickly distribute all kinds of prefabs in your world. Please take a look at the asset videos on the Unity Asset Store in order to see some of its many uses.

YAPP is container based. All painted prefabs are integrated into a parent gameobject. One important philosophy is that the Unity project remains independent of the prefab painter. This means that after you painted the prefabs via brush or spline you can keep YAPP in your scene or remove it. The painted prefabs stay and are independent of YAPP.

The Idea

When I started Unity I thought this might be a good idea to learn coding. Creating a prefab painter is something rather trivial. All you need to do is analyze an area around the mouse cursor and instantiate a prefab there.

However as usual one thing leads to the other and it turned out that there's always something that one wants more. The existing prefab painters - free and commercial - didn't meet my requirements. So that's another reason why I created my own.

And over time more and more requirements and requests came up and so YAPP became to be a lot more than what was initially intended.

Also, a big shoutout to the awesome people of the Unity Community who provided tutorials which I used for the more advanced features. Please see the credits section below.

Features

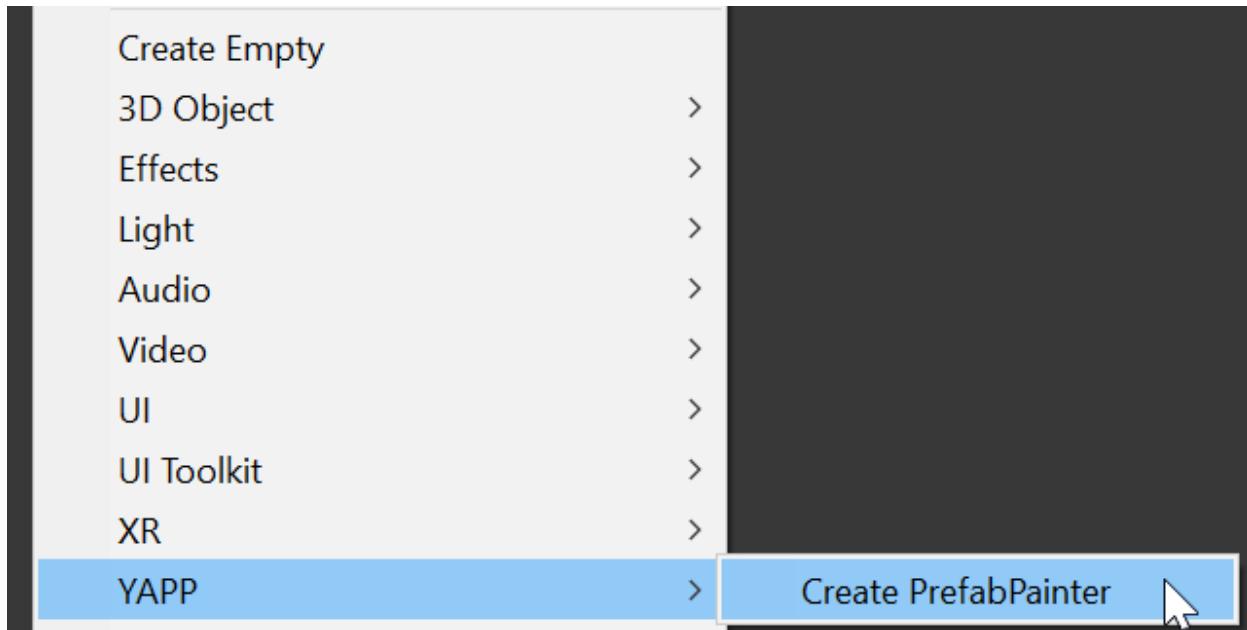
I just love to create worlds with Unity. All kinds of. However I do Unity only in my spare time which is rather limited. So in order to make distributing prefabs convenient, flexible and fast I created YAPP. Yet Another Prefab Painter.

Among many other features the core features are:

- Container based
- Paint modes Brush and Splines
- Prefab instantiation
- Batch operations
- Adjustment operations
- Physics
- Vegetation Studio Pro integration
- Terrain Trees integration

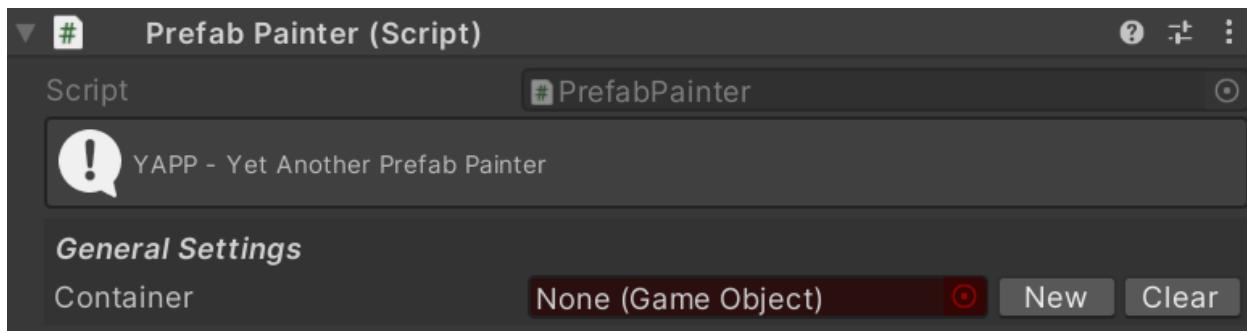
Getting Started

First create a new Prefab Painter via context menu in the hierarchy:

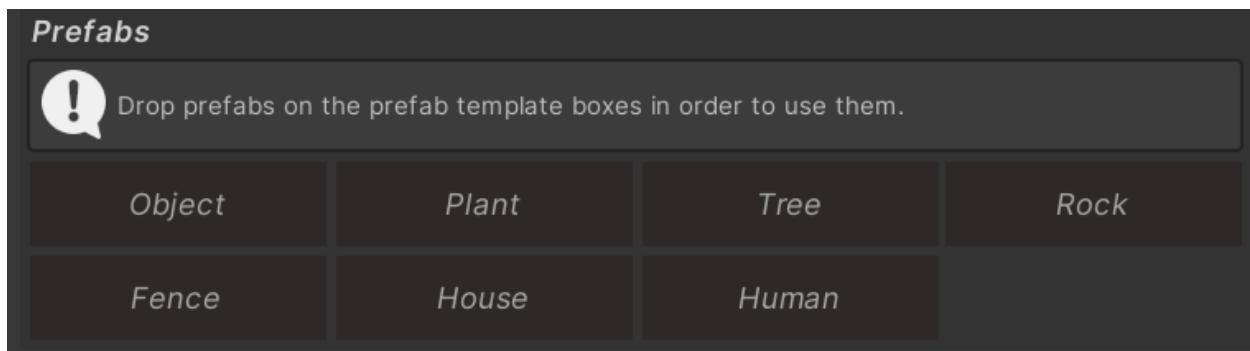


This creates a gameobject which you can name however it pleases you. Let's call it "Cliff Painter". After you named the gameobject the prefab painter will be added to it.

Now create a container (a gameobject) or add one to the Container component in the inspector. There's a "New" button for your convenience.



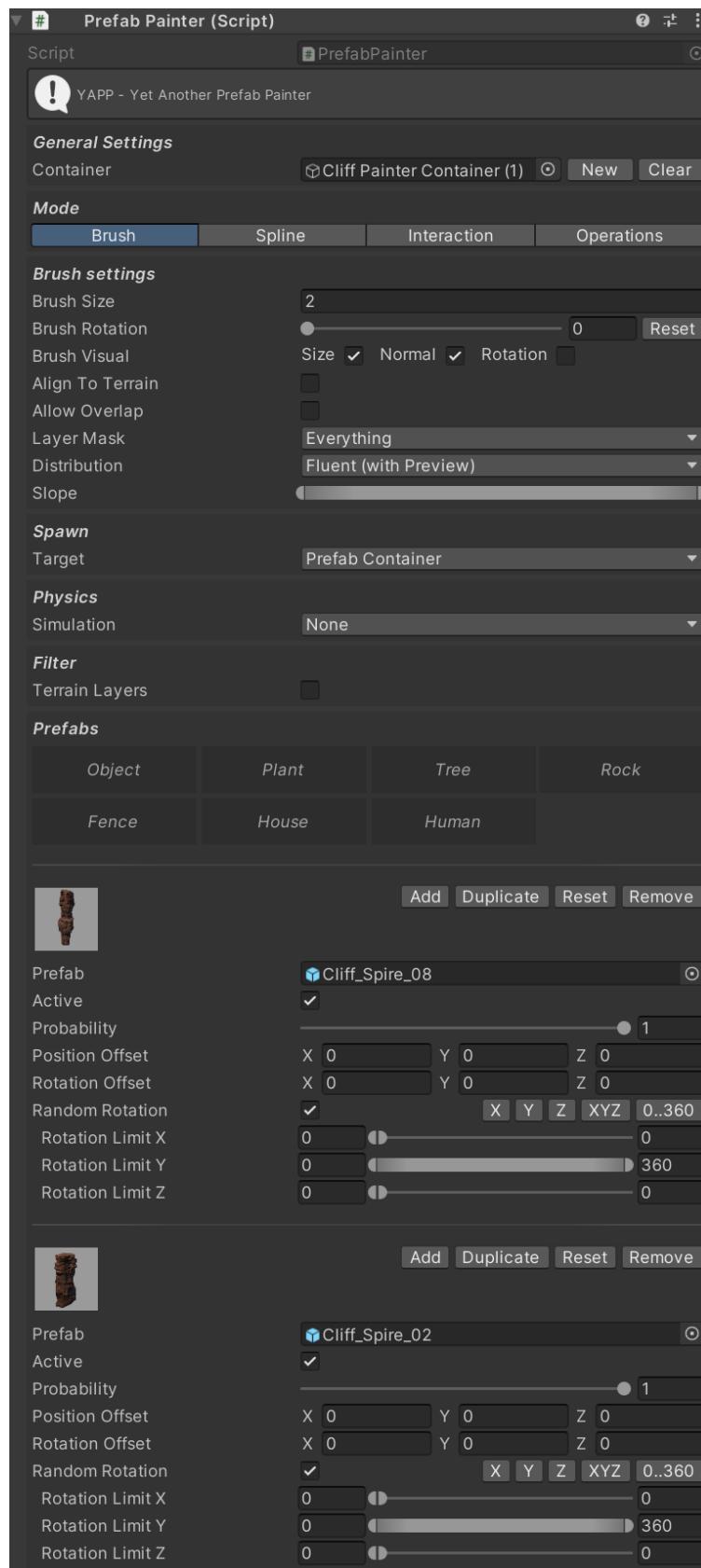
Then drag some prefabs on any of the slots



The slots are presets that you can use.

Examples: a rock can have random rotation in all directions and arbitrary size. A tree however should only rotate in Y direction. Same for plants. We'll get to that later.

In the current example I'm using the awesome Castle Valley Collection. After adding some spires to the prefab slots the prefab painter looks like this in the inspector:



Now we can already start painting. Per default the Fluent distribution brush is selected.

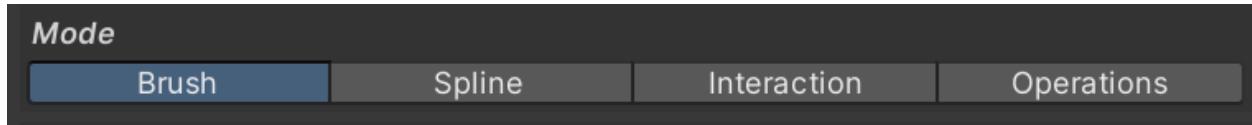
A blue circle means the brush mode is in use. Hold shift, the circle turns to green, you can start painting. Click left mouse button while keeping shift pressed to instantiate and position the prefab. Use ctrl to have the circle turn red, now you can remove prefabs via painting. You can see a text with the relevant options at the bottom of the scene view:



And that's basically it to get you started.

Modes

YAPP has various modes for distribution and interaction. You can switch the modes in the tab:

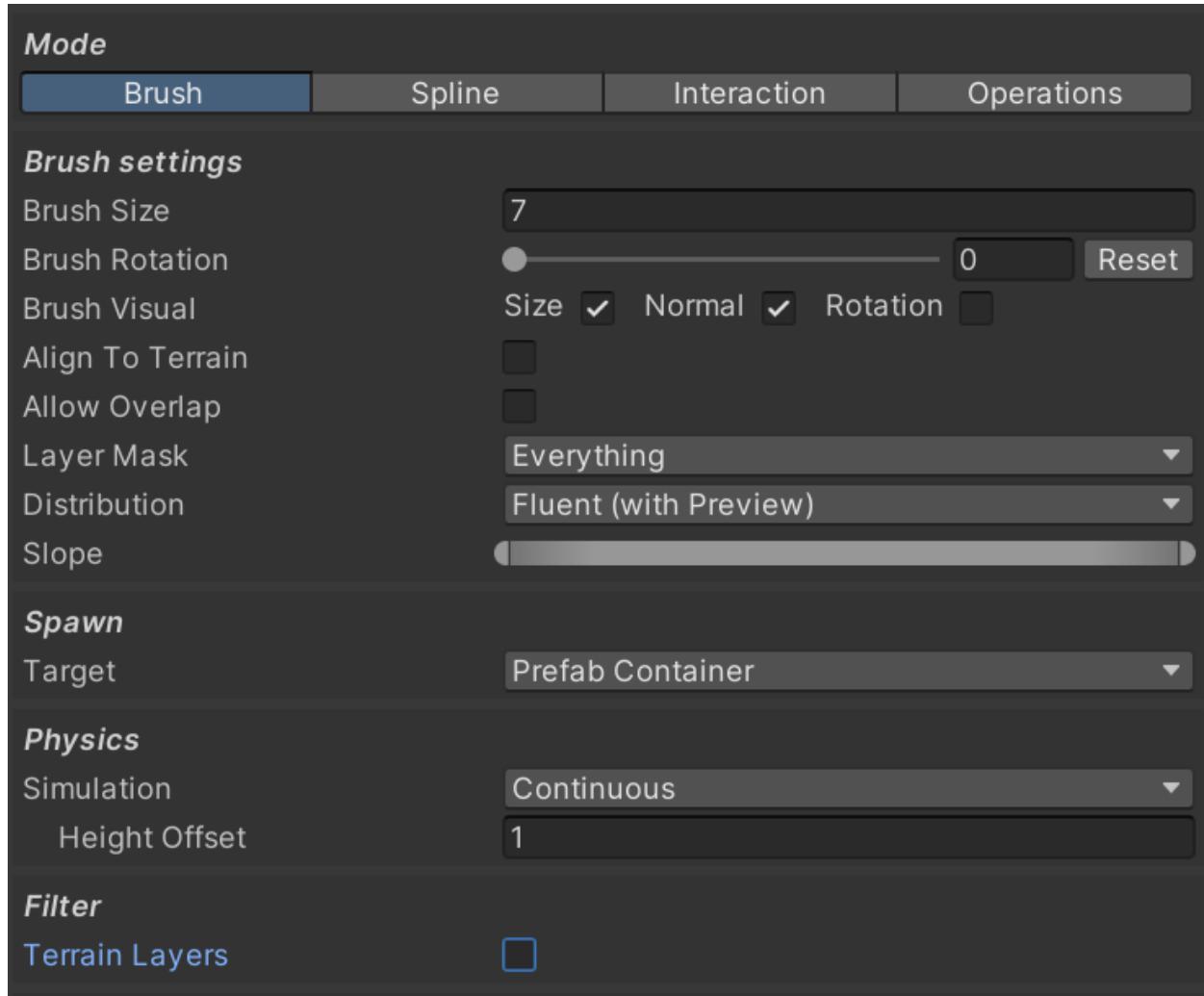


- Brush
Paint via dragging the mouse
- Spline
Spawn prefabs along a spline
- Interaction
Interact with children in the scene view
- Operations
Perform operations on the container children

Personally I prefer to have the GUI speak for itself instead of having details in the documentation. So here's just a brief overview. However, if you think something needs more information, please let me know.

Brush Mode

With the Brush mode you can paint prefabs like in eg a painting app. The important thing to note about the brush mode is the distribution setting and that the painting via brush is based on density. There's a detailed chapter about that.



- **Brush Size**
Show current brush size or perform precision changes
Normally you change the size via key bind and mouse wheel
- **Brush Rotation**
Show current brush rotation or perform precision changes

Normally you change the rotation via key bind and mouse wheel

- Brush visual
Visibility settings of the brush gizmo
- Align To Terrain
Automatically align the prefab to the terrain normal
- Allow Overlap
By default the painter checks against children of the current container and doesn't allow overlaps. With this option you can override this setting
- Layer Mask
Use this to raycast the brush only against selected layers
- Distribution
 - Fluent (with Preview)
Detail painting with preview. Scale, rotate and align in Y direction
 - Center
Instantiate the prefab at the brush center
 - Poisson (Any Collider)
Create a poisson distribution within the brush and instantiate in all islands. This works on all colliders
 - Poisson (Terrain Only)
Create a poisson distribution within the brush and instantiate in all islands. This works on terrain only
- Slope
Limit painting to the specified slope range
- Spawn
Where to store the instantiated prefab
 - Prefab Container
 - Terrain Trees
 - Vegetation Studio Pro

- Physics

Perform Physics simulation after a prefab got put into the scene

- None

Physics disabled

- Continuous

Apply Physics

- Height Offset

The offset in Y in which the prefab will be spawned.

Example: Spawn a rock 1 meter above its destination and then have physics applied so that it drops on the ground and aligns more naturally.

- Auto Collider

For physics simulation a rigidbody and a collider is required per spawned gameobject. These will be added and removed automatically.

- SpawnedOnly

Add collider only to objects which are spawned during the current physics simulation

- Container

Add a collider to all container children during the current physics simulation

- Filter

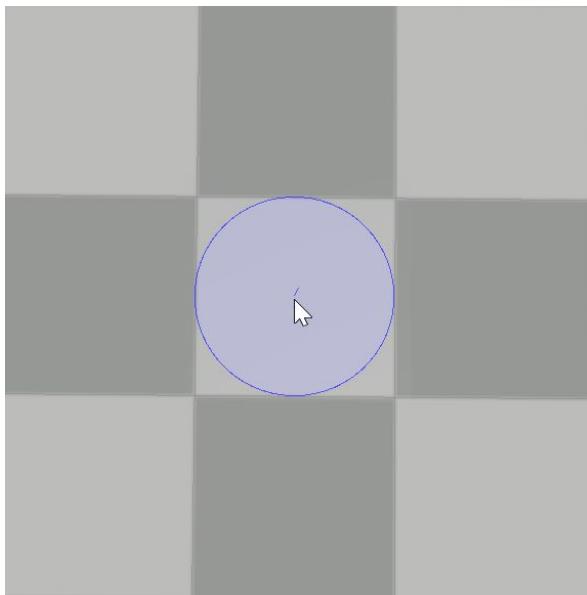
- Terrain Layers

Spawn only on the selected terrain layers

Brush Mode Distribution

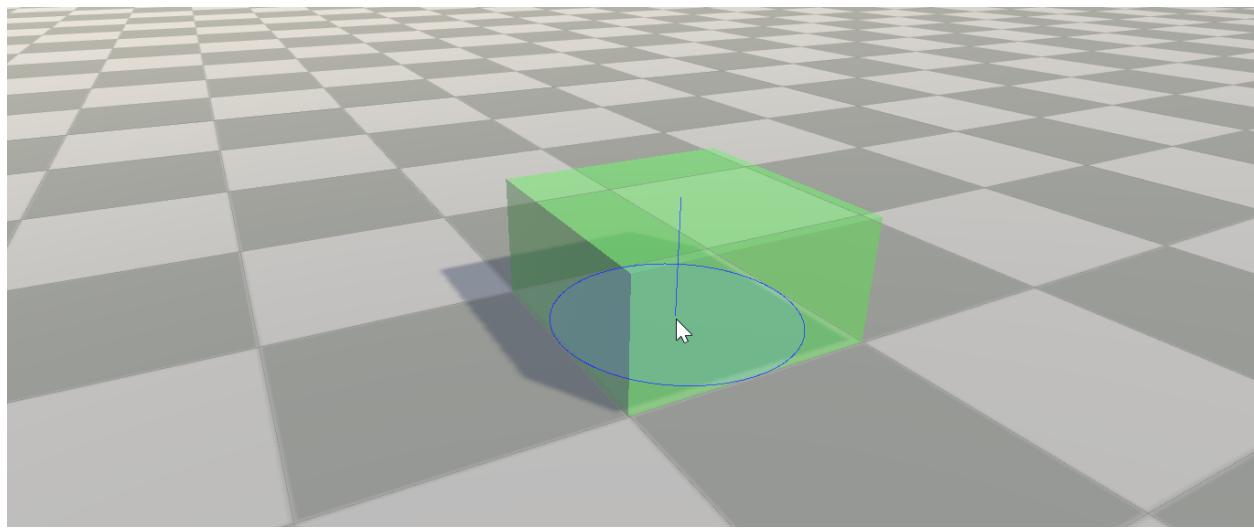
In Brush Mode the prefabs of your palette are being chosen depending on your prefab settings, and instantiated at the brush positions. The prefab properties are applied accordingly. Example: For a tree you may want rotation in Y axis, but keep X and Z as they are.

The Brush works with density. To visualize this, here is the brush on a grid based on 1 meter distance, the brush size is set to 1 unit of Unity which is 1 meter. In my example I just use the Unity terrain without layers, it shows a grid by default.

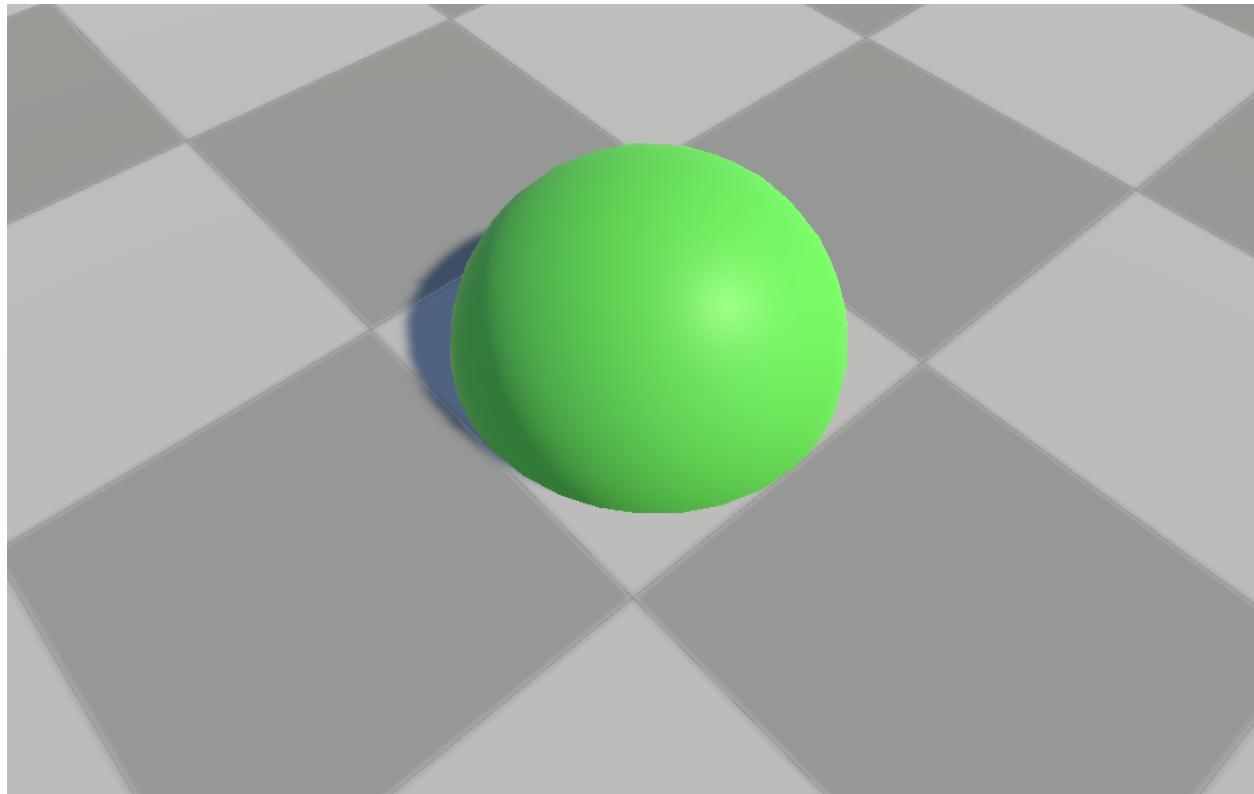


You see the 1 meter sized brush fills the 1 meter cell of the grid, best visualized in orthographic view.

Now if you add a default cube as prefab for painting, the default cube - which has also 1 meter dimension by default - will fill that brush and grid cell. The cube is transparent here and green, just to visualize the effect better:



Or visualized with the unit sphere painted via brush:



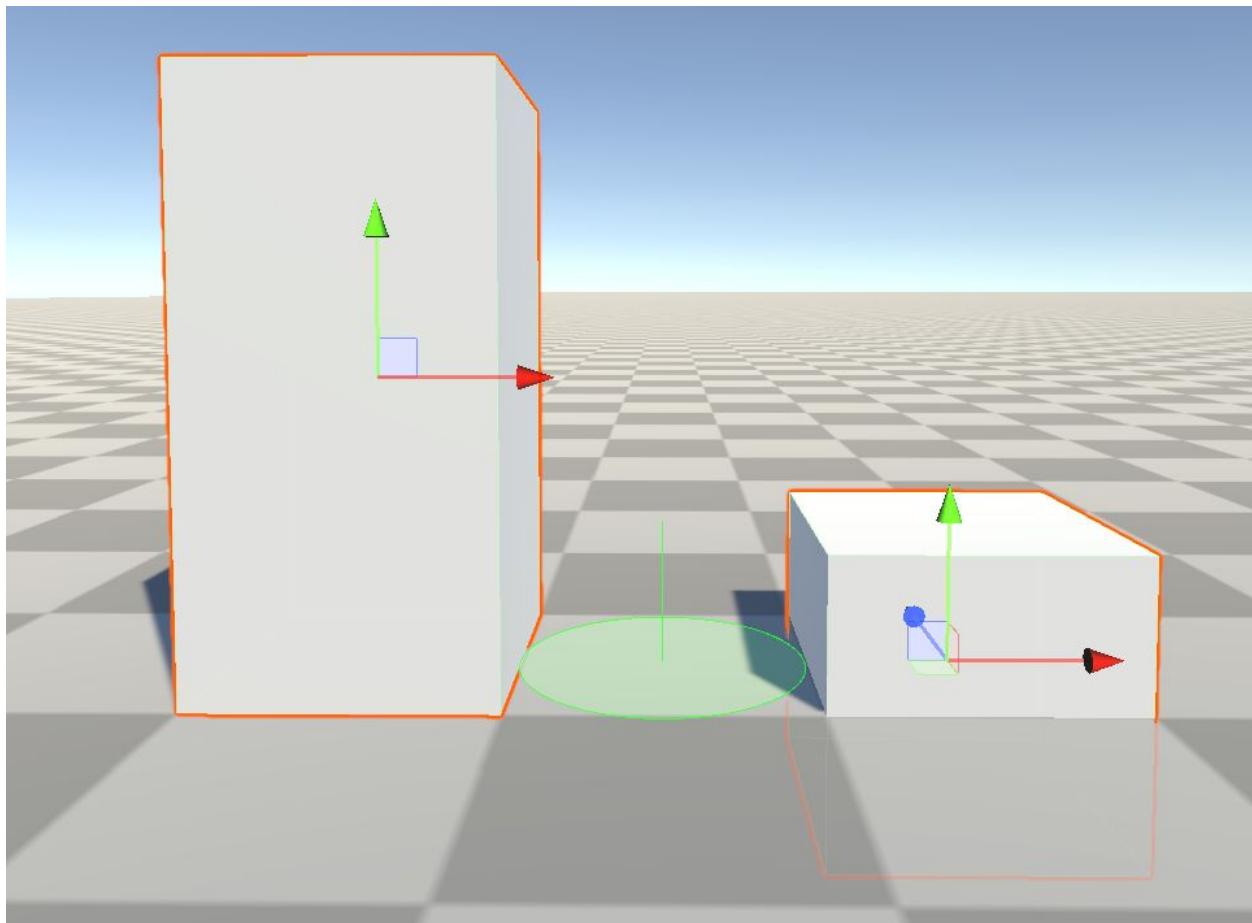
The brush works in combination of these 2 criteria:

- Brush size
- Container children

The prefab distribution is being performed by density. With that in mind a prefab can be instantiated if there is no other prefab of the same container within the brush.

For the distance calculation between prefabs and brush Unity's Distance method is being used.

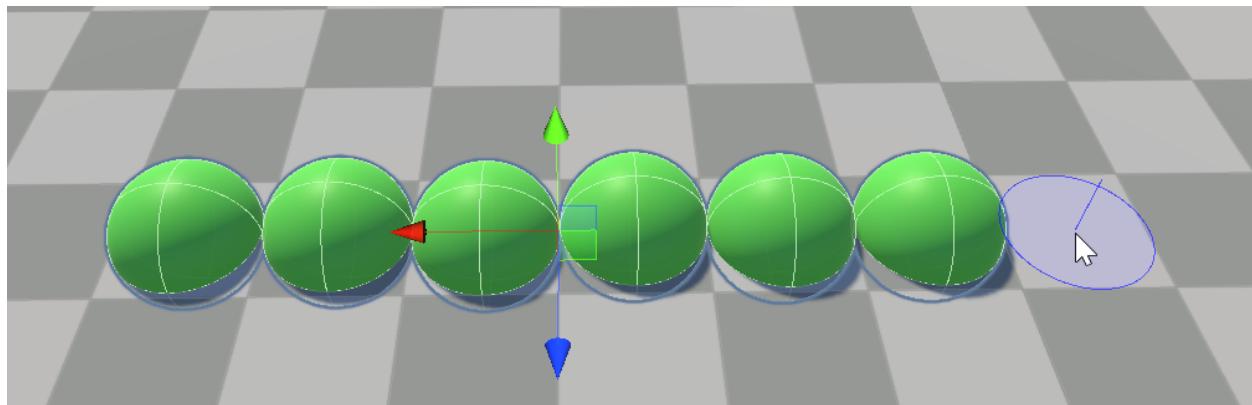
As an example how this matters, take a look at this picture:



Depending on the transform of the object the result of the distance calculation can become quite different. If the brush is at the bottom of the terrain and is calculated against the right cube, the distance is 1. Against the left cube it's in this special example case the square root of 2 because of rotation and position offset ($y=1$ in this case) of the placement settings of the left prefab.

Distribution: Center

Regarding the density distribution mentioned earlier, if you have a unit sphere of 1 meter diameter and a 1 meter brush and drag it along an axis in Center distribution on a 1 meter grid, the spheres would be equally distributed like this:

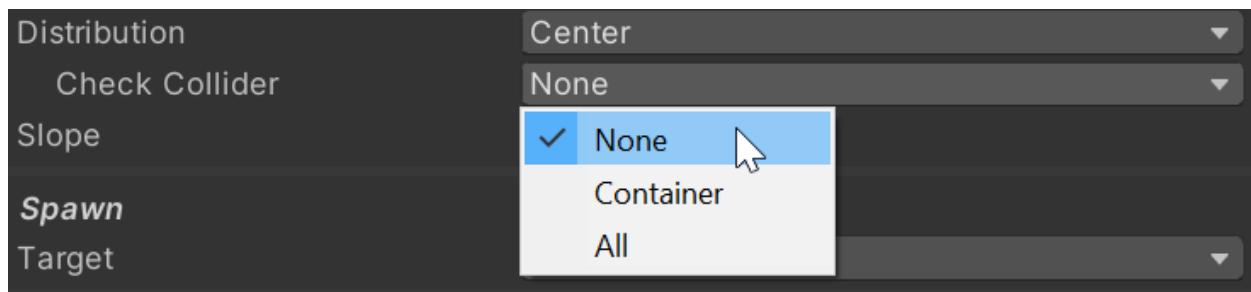


The Center distribution places a prefab with its pivot in the center of the brush.

The spacing depends on the mouse speed that you paint your prefabs with. Should the need arise, an interpolation between spacings may be implemented, but the Spline mode is better suited for this purpose.

The placement distance depends on the prefab position and the brush size. This may have the effect that a new prefab is placed on an existing one, eg when you have a sphere of size 2 and a brush of size 1 the distance limit is already met with the bounds of the sphere. This could lead to the unwanted effect of the new prefab being placed on top of the sphere instead of the terrain.

To circumvent this there's an option Check Collider available:



If this is active, then additional checks against colliding gameobjects are being performed before the prefab is placed.

The options are:

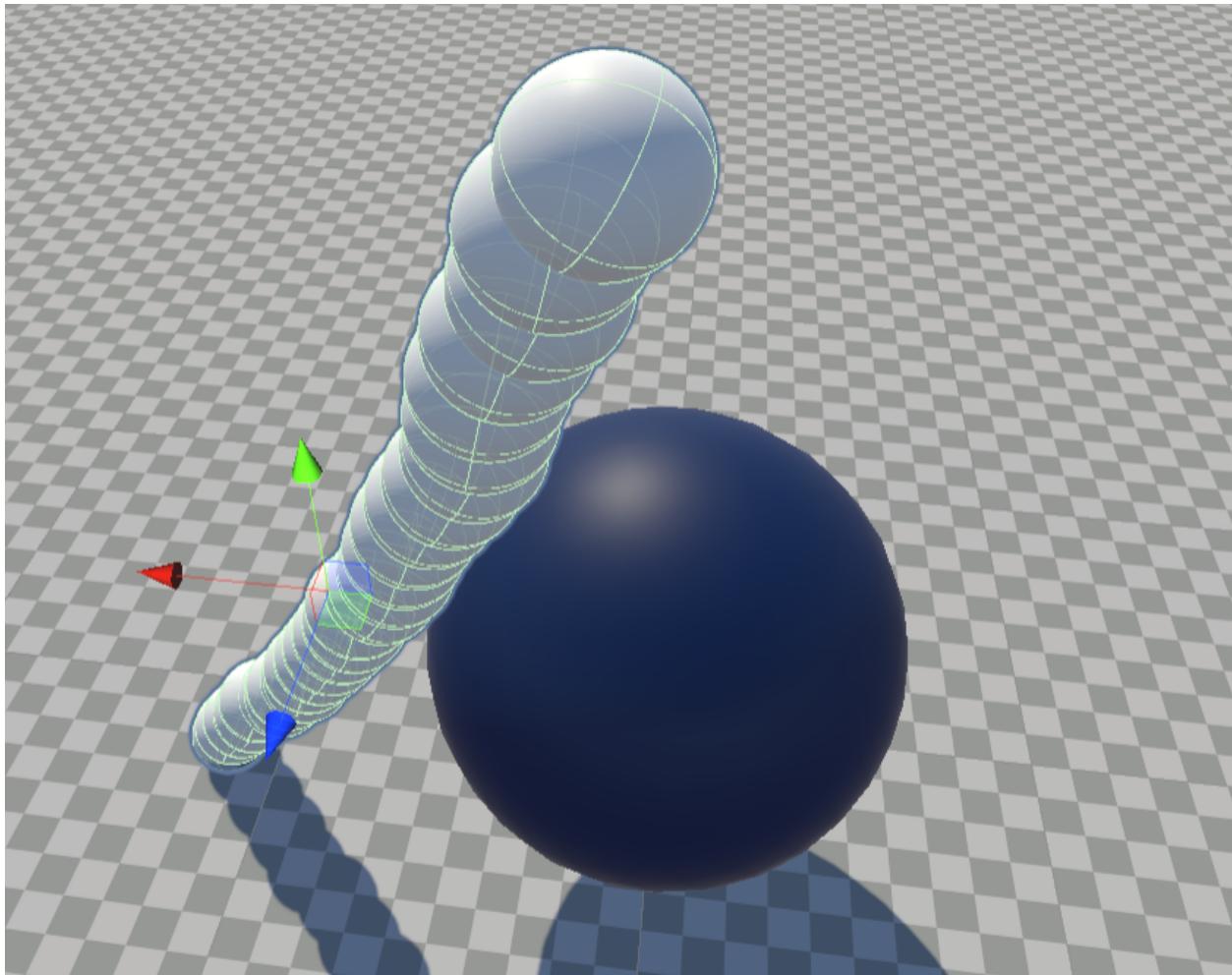
- None
Default behaviour
- Container
Check the about to be placed prefab's collider against the collider of the siblings, ie the other prefabs in the same container. If there's an overlap, then the prefab won't be placed
- All
Similar to the Container mode, but checks against all colliders, not just the container children

Please note that checking against colliders is using several Physics functions and might impact performance. As such there are some limits, like eg a maximum of surrounding colliding gameobjects of 32. This can be increased if the need arises or a property being created, depending on your feedback. So far it seems a good option the way it is.

To visualize the differences a big blue sphere is used as target and the prefab has a scale with size 2, the brush of size 1.

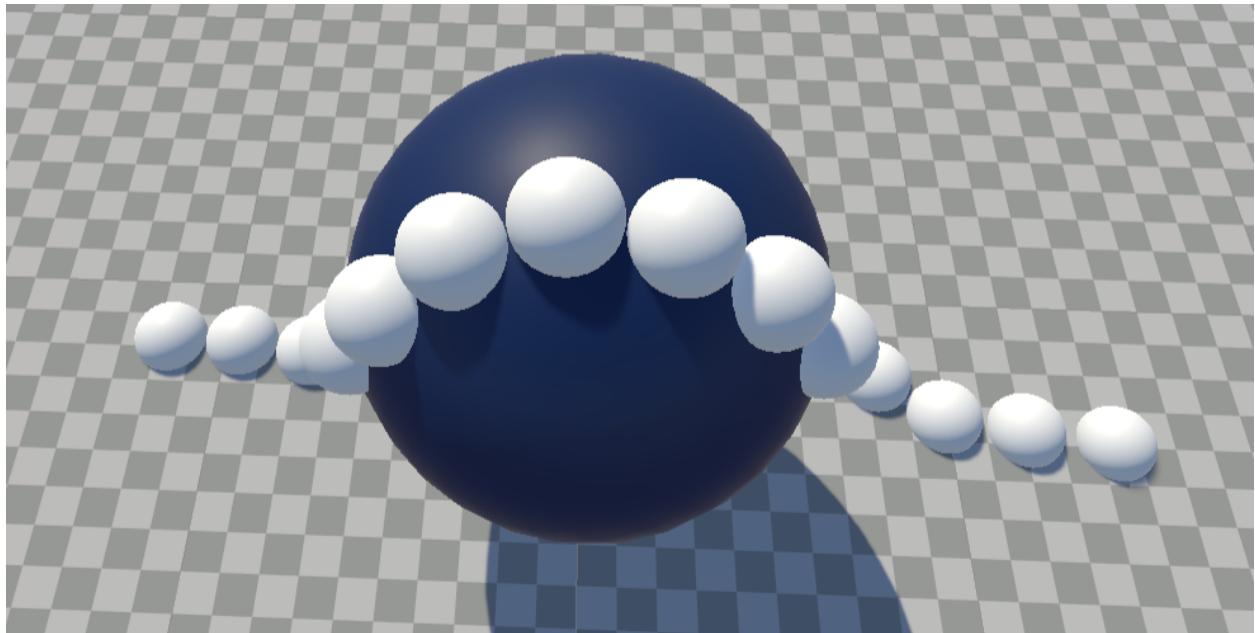
None

Brush bounds are met during painting, so the prefabs are stacked on another



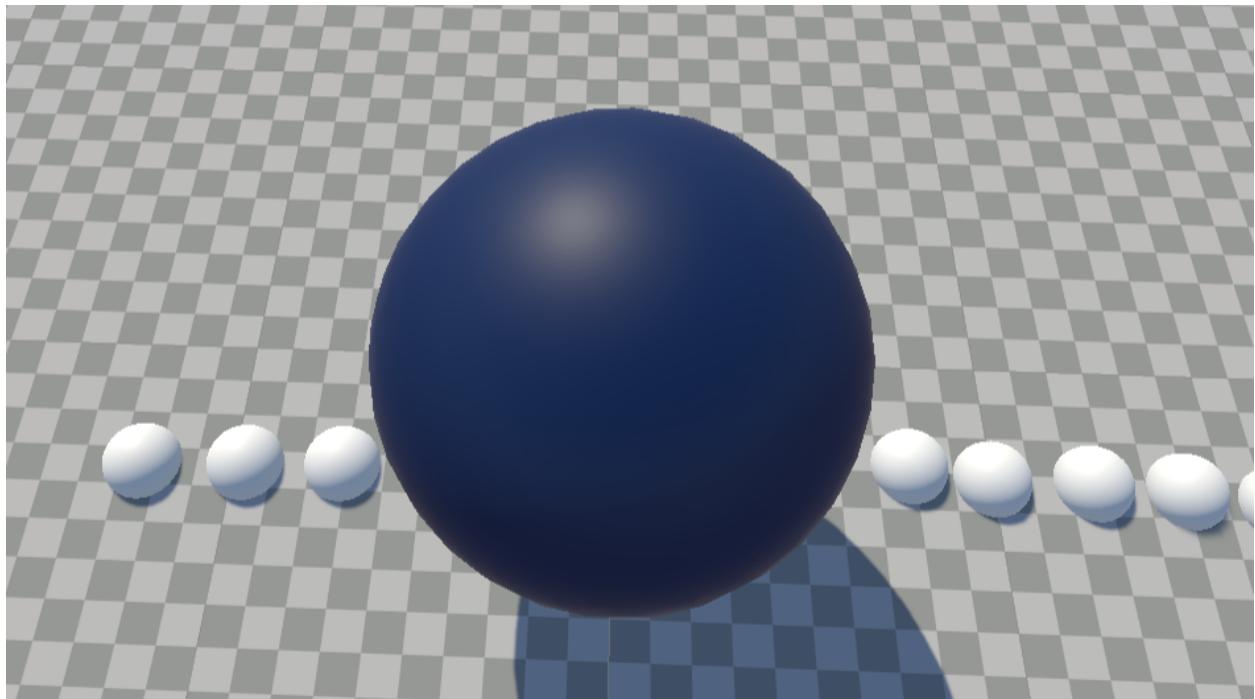
Container

The prefab collider is checked against the container children collider, but not against the target which is the blue sphere.



All

The prefab collider is checked against any collider, including the blue sphere.

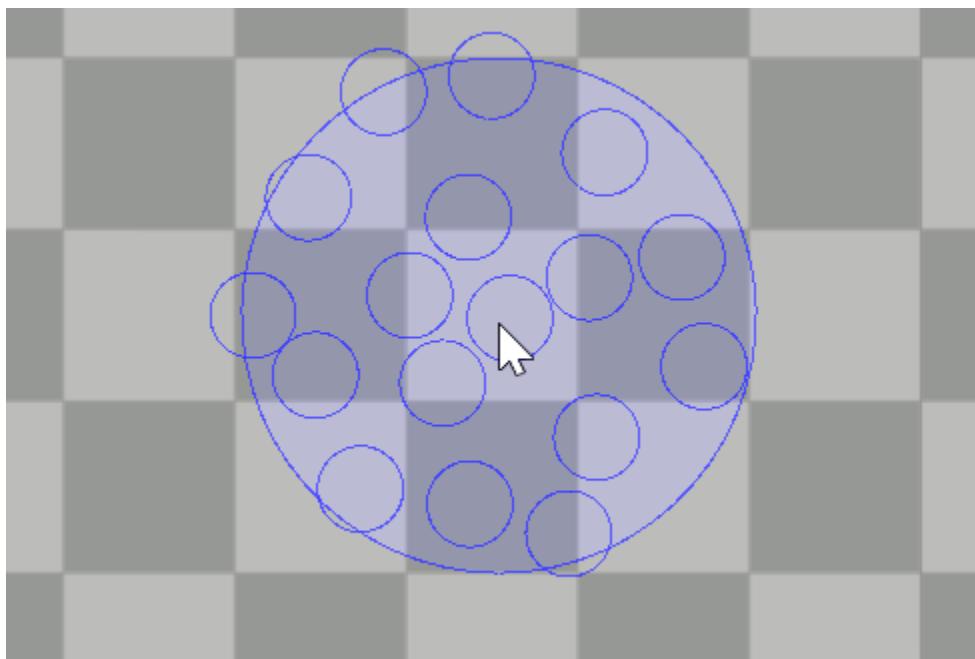


Distribution: Poisson

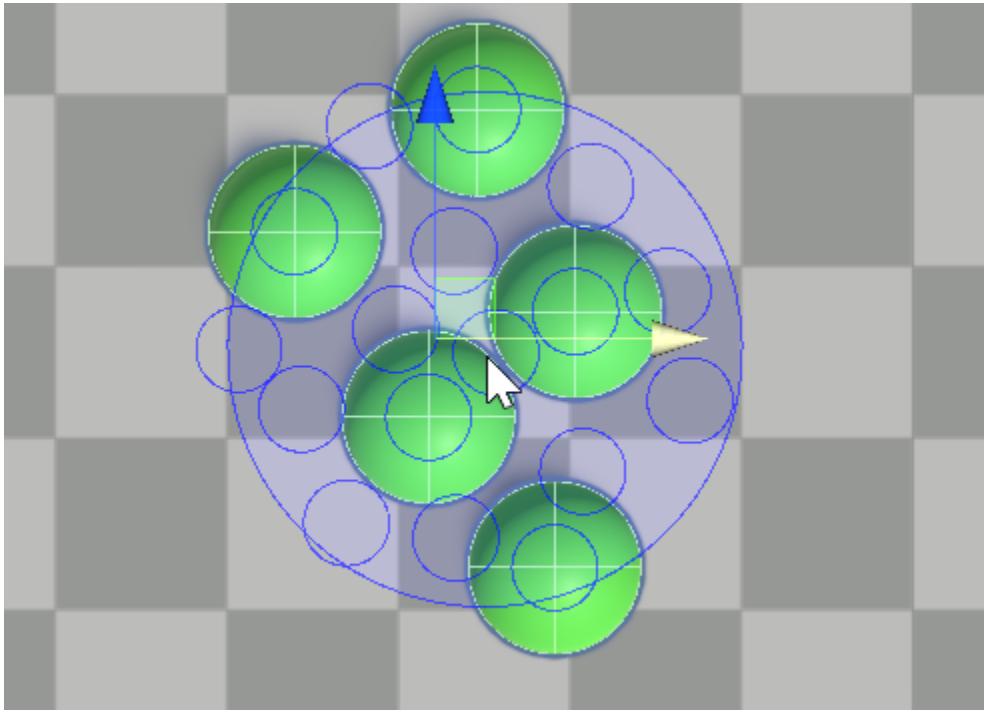
The poisson distribution is a convenient way to paint multiple prefabs with a single brush placement. Its use cases are eg painting trees of a forest, vegetation, humanoids in a market place, etc.

Essentially it's similar to Center distribution except that the brush is split into equally distributed smaller disks, each of them is being used as a center brush.

To visualize this, here's a brush with poisson discs in it:



Each disc is a candidate for painting a prefab. However the limits of density distribution still apply: If there's space available, a prefab will be placed. If there's already a prefab of the same container which would be overlapping, then painting will be skipped. Looks like this then for the mentioned unit sphere:

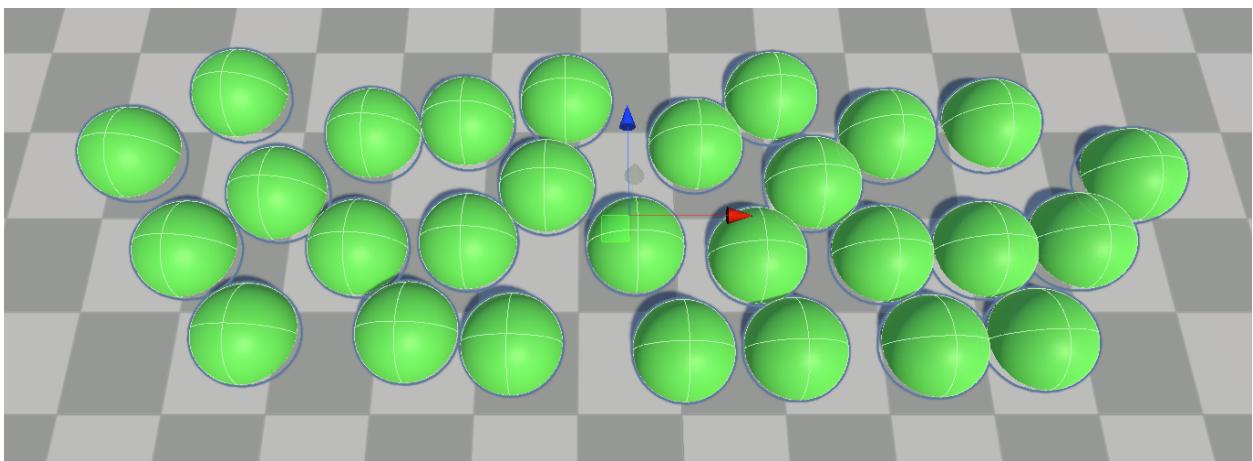


You can see the green sphere prefabs at the centers of the poisson discs.

Please note that the pivots of the prefabs are used for the distance calculation.

You may allow overlapping as an option which then would result in spheres at all poisson disc centers. It really depends on the use case.

Dragging, ie painting a stroke would result in an example like this:



Nice and quick non-equal distribution.

There are 2 options for the poisson distribution:

- Any

This works on any gameobject as raycast target. Plane, sphere, including terrain

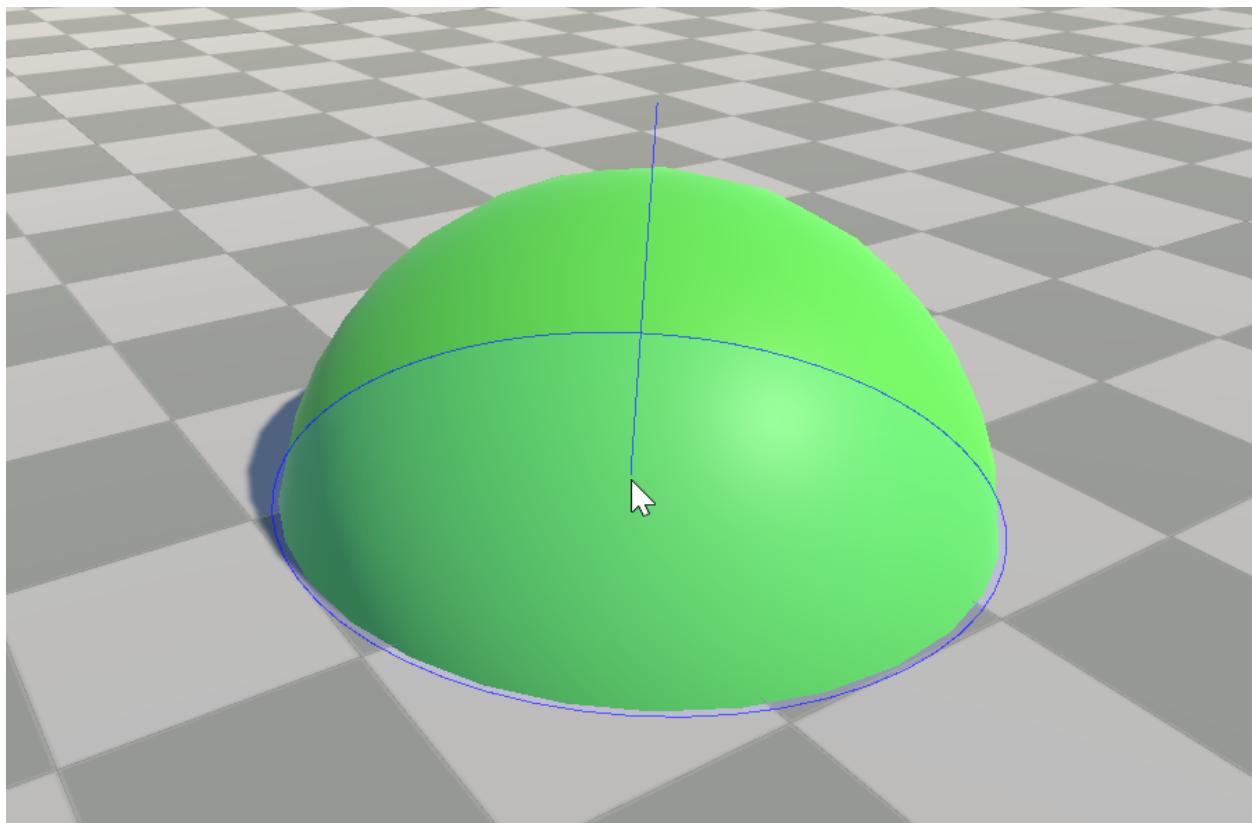
- Terrain

This works only on terrain as raycast target. This is a quick option so that you don't have to use dedicated layers.

Distribution: Fluent (with Preview)

The fluent distribution can be used for detail painting. It's initial purpose was to have eg cliffs and rocks placed on the terrain, rotated according to artistic choice and even have them raised or lowered into the ground without having to use Unity's own gizmos. However, the Unity transform changes may as well be used after placement for additional changes.

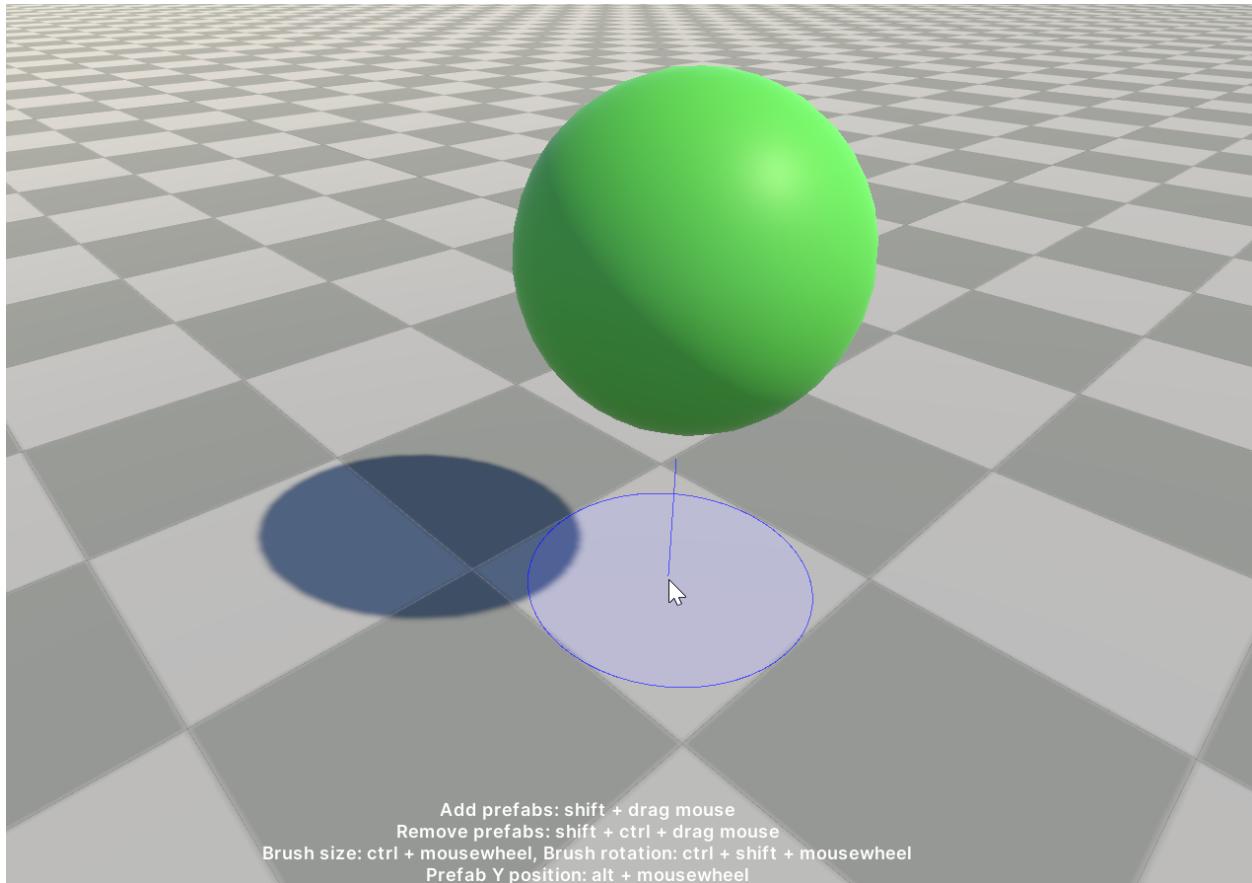
Fluent distribution works differently to the Center or Poisson distribution. In Fluent mode the brush size is being used for the size of the prefab. To visualize this with the unit sphere:



You can see the unit sphere isn't the 1 meter unit anymore, it's got the size of the brush.

This has several advantages, among them of course that you can see instantly how your prefab looks when it is placed. For fluent placement you can change the position naturally, the size and

the y position. There are various shortcuts for that visualized at the bottom of the scene window like this:



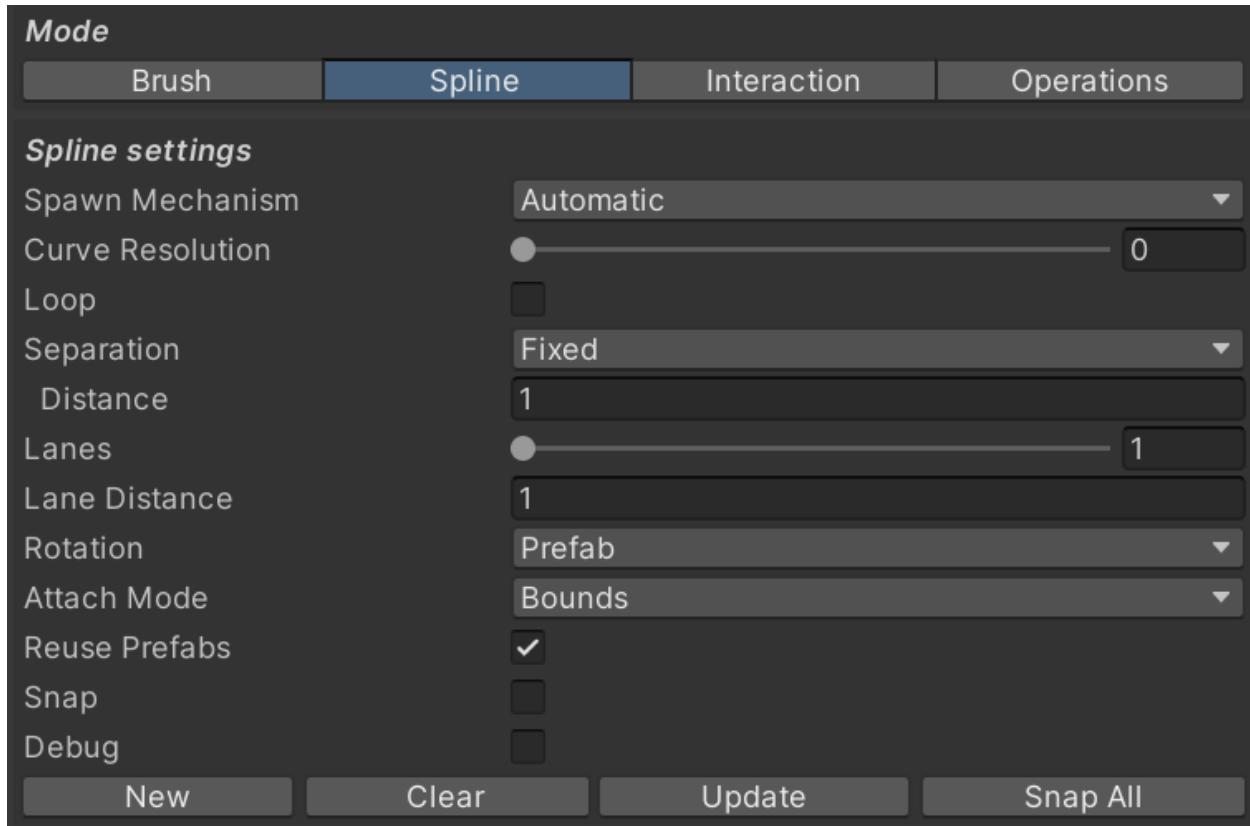
Here the height was changed before painting the previewed prefab.

To visualize this in an actual game scene creation use case, please take a look at this example video:

<https://www.youtube.com/watch?v=lwegMdqNMRA>

Spline Mode

The Spline mode allows you to distribute prefabs along a spline dynamically.



- Spawn Mechanism
 - Automatic

Spawn while the spline is being created
 - Manual

Spawn when the “Spawn Prefabs along the Spline” button is pressed, it appears when Manual is selected

- Curve resolution

The smoothness of the spline curve

- Loop

Whether start and end are connected or not

- Separation

- Fixed

Fixed distance spawning

- Range

Random distance spawning

- Prefab Radius Bounds

Distance along the radius of the prefab

- Prefab Forward Size

Distance along the size in forward direction

- Lanes

Create multiple lanes along the spline

- Rotation

Rotation of the spawned prefabs along the spline

- Spline

Rotate along the spline direction

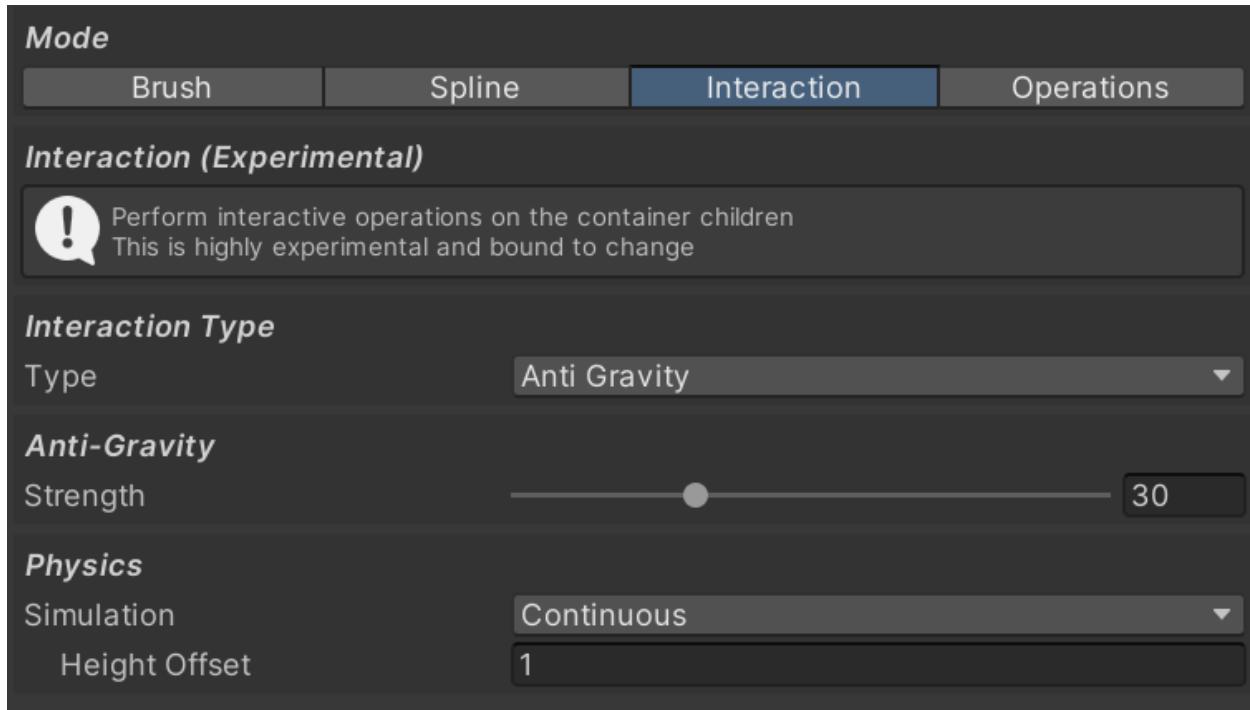
- Spline Random

Position along the spline, but consider the rotation settings of the prefab

- Prefab
Use only the settings of the prefab
- Attach Mode
Where new spline nodes are created
 - Bounds
Either at the start or the end
 - Between
Anywhere along the spline
- Snap
Snap spline nodes to the top or bottom raycast hit

Interaction Mode

The Interaction mode allows you to operate on the container children in scene view.



- Interaction Type
 - Gravity
Lift the prefabs inside brush and apply gravity on them
 - Magnet
Attract or repel prefabs along the brush
 - Change Scale
Change the scale of the prefabs which are inside the brush
Can be used eg to change the trees of a forest

- Set Scale

Set the scale of the prefabs which are inside the brush to the specified value
- Physics

Some of the interactions require physics. Here you can specify the settings for the physics simulation

Operations Mode

The operations mode allows you to perform operations on the container children.

Mode

- Brush
- Spline
- Interaction
- Operations**

Container

! Perform operations on the container children

Physics Settings

- Force Apply Type: Initial
- Force Min/Max: X 0 Y 0
- Force Angle (Degrees): 0
- Randomize Force Angle:

Simulation

- Time: 3
- Steps: 1

Undo

Start Stop

Copy/Paste

- Copy Transforms
- Paste Transforms

! Use in combination with Physics to revert to another state than the previous one.

Selection

- Few
- Medium
- Many
- All

Tools

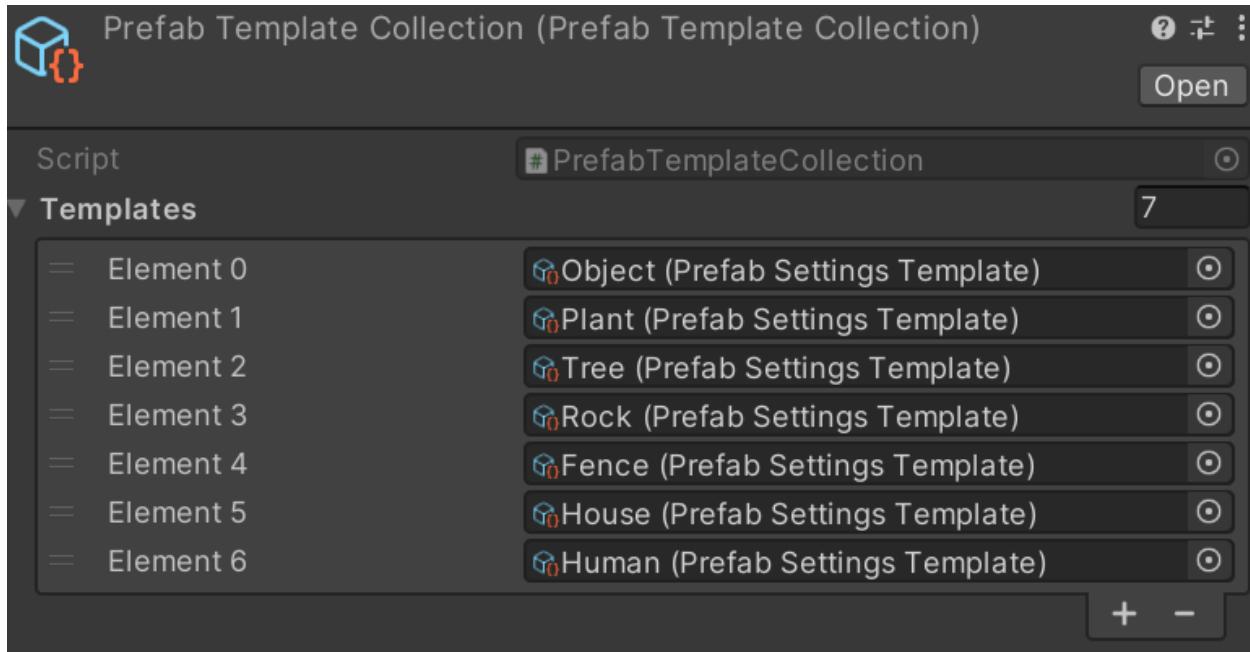
Remove Container Children

- Physics Settings
Apply various physics settings on the container children
- Undo
Undo the last physics simulation which was performed on the container children
- Copy/Paste
Copy/Paste the transforms of the container children
- Selection
Select random container children
- Tools
Remove all container children

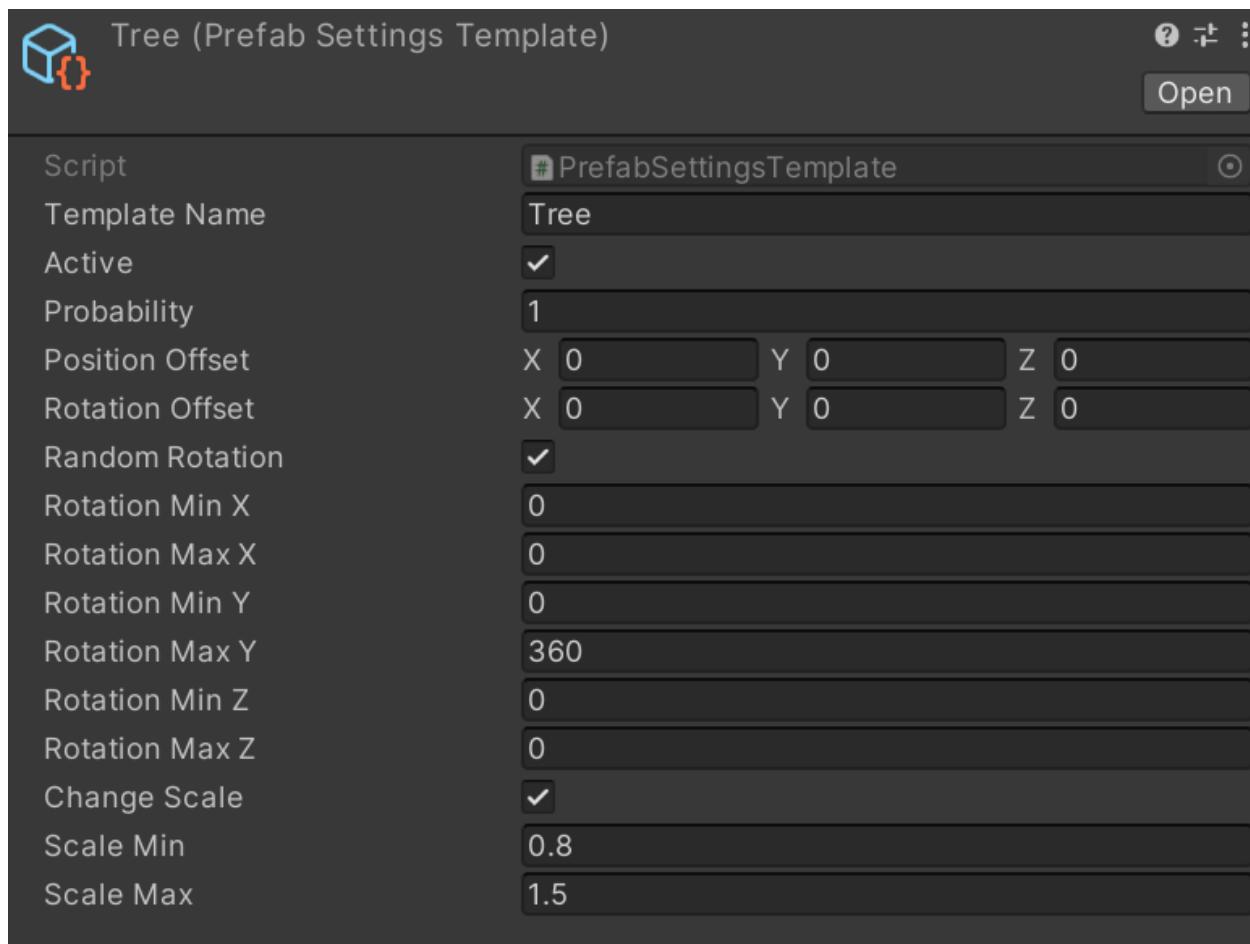
Prefab Templates

The prefab slots are templates which you can extend and modify as you require. For this please look at the Templates folder and the PrefabTemplateCollection object.

The PrefabTemplateCollection is an array:



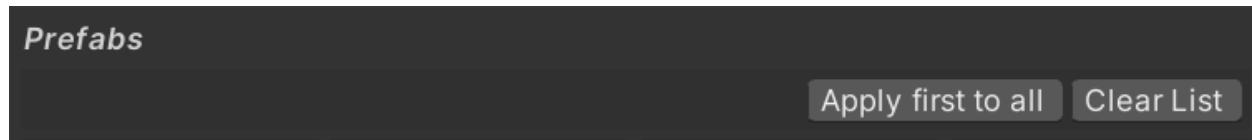
One of the details is eg “Tree” and looks like this:



When you drag in a new prefab to the Tree slot, then the prefab settings will be preset with these settings. You can change the settings afterwards as you prefer, the template is just an initial preset used to save repetitive clicks and get you quicker into prefab painting.

Prefab Settings

The prefab settings are a list of prefabs, each with their own properties. There are functions for convenience:



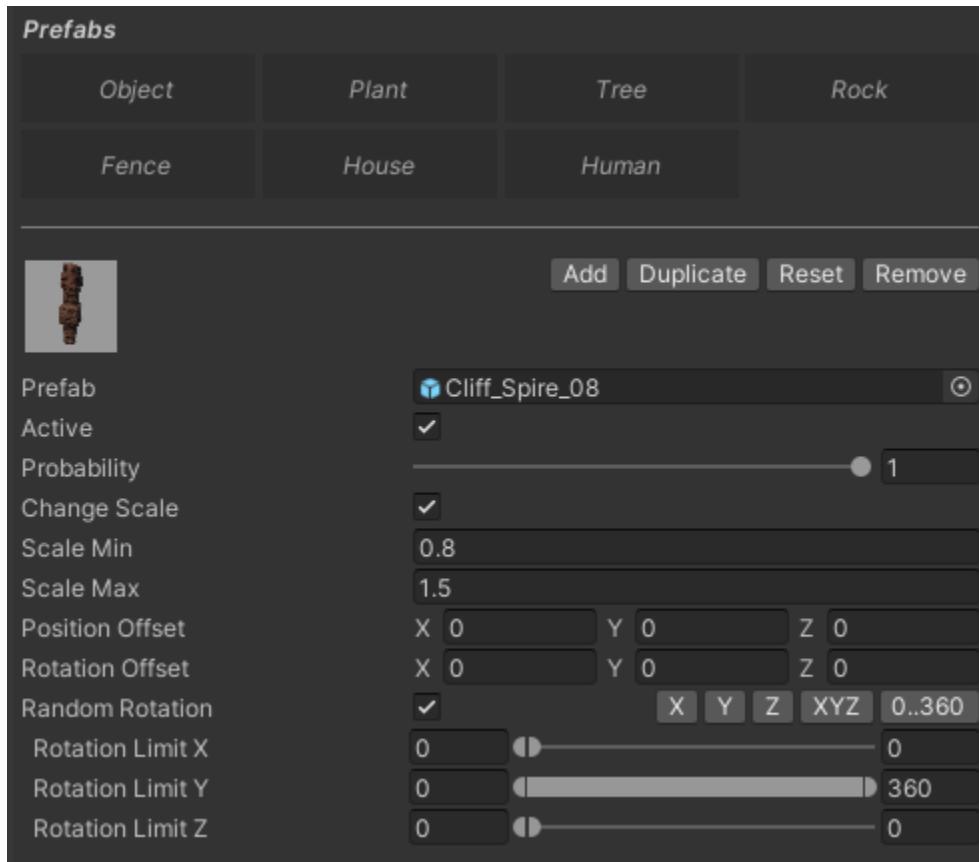
- Apply first to all

If you want to have the same properties on all the prefabs, you can modify the first prefab to meet your requirements and apply the same setting to all the others

- Clear List

Removes all prefabs in the prefab settings list

When prefabs are instantiated several options are applied to them. The options can be defined in the prefab settings. The initial value is used from the template that the prefab was dropped on.



Various tool buttons are available:

- **Add**
Adds a new empty prefab setting into which you can drag a prefab. It is recommended though to use one of the templates

- **Duplicate**
Duplicate the settings of the prefab. If you have set up eg a tree with your custom preferences, then you can duplicate the settings and assign another tree to the Prefab slot

- Reset
Reset the prefab settings to default values
- Remove
Remove the prefab. It won't be available anymore for painting

The settings are as follows:

- Active
If this is checked, then the prefab will be in the pool of prefab choices and it can be used for painting. If this is unchecked, then the prefab will be ignored
- Probability
The probability that the prefab will be chosen for painting. 0 probability means it won't be chosen. The higher the probability, the more likely this prefab will be used for painting.
- Change Scale
The scale change that will be applied when the prefab is instantiated. The scale range is between the specified Scale Min and Scale Max range.
- Position Offset
The offset that will be applied when the prefab is being painted. If you want eg a tree further down in the terrain, you'd specify a negative offset here. Another use case might be eg rocks that you want to have further down in the terrain.
- Rotation Offset
The offset that will be applied when the prefab is being painted. If you have a prefab that's per default not rotated in the direction you want, then you can correct the rotation here.

- Random Rotation

If Random Rotation is checked, then you can specify limits of which the prefab's rotation will be rotated additionally.

There are quick buttons for X, Y, Z and XYZ.

Example use cases:

Vegetation has most likely only random rotation in Y direction.

Rocks on the other hand can have arbitrary rotation in XYZ direction.

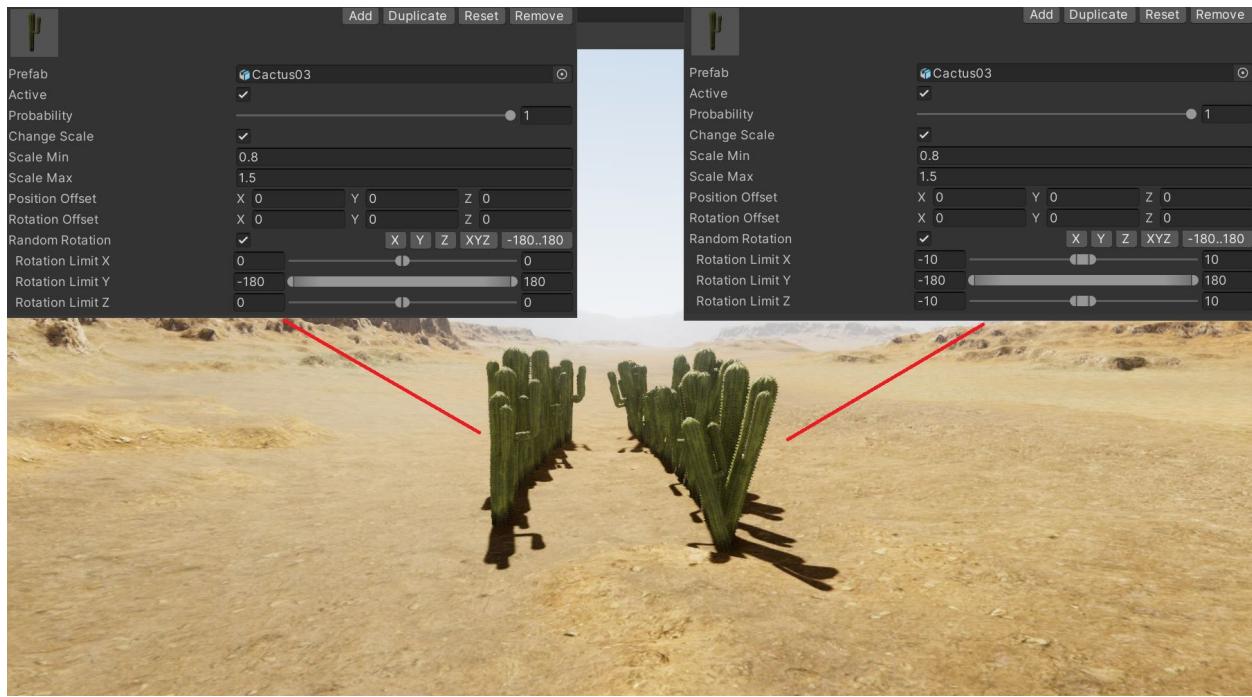
The random rotation can be specified in 0..360 or in -180..180 intervals. The reason is simple: If you have eg Humanoids that face forward direction, but should only rotate in Y direction by a -10..10 degree interval, then you'd use -180..180 because 0 is their forward direction.

Specific Use Cases

Align along Terrain Normal with random rotation offset

YAPP has an option to align prefabs along the terrain normal, ie Align To Terrain in the Brush settings. In order to not be so strictly looking you can apply an additional random offset.

To visualize the effect better here the alignment on the terrain normal is being used and an additional random rotation is being applied.



The screenshot shouldn't visualize the normal placement, but for better understanding the effect of applying an additional rotation in X and Z. The cactus are rotated randomly in Y direction in the left part of the image. The right part has an additional X and Z rotation so that the plants appear not so uniform and instead randomly slightly tilted.

Rotation 0..360 vs -180..180

Here's an example where -180..180 is being used. The humanoids were placed along a spline with their random rotation offset being slightly off from 0 in both positive and negative direction:



To visualize this you may also want to check out this video:

https://www.youtube.com/watch?v=_t0GRsXn3dY

Integrations

YAPP optionally integrates these assets:

- Vegetation Studio Pro
Paint into the persistent storage of Vegetation Studio Pro.
- Unity Terrain Trees
Paint into the Terrain Tree details of Unity's Terrain

Vegetation Studio Pro Integration

You can optionally paint into the persistent storage of Vegetation Studio Pro. Please note that this functionality is limited to

- adding and removing vegetation items
- Paint target is the Default biome
- prefabs are added as vegetation type Object

Frequently Asked Questions

Scale isn't visible in the prefab settings

The scale in Fluent distribution of brush mode is controlled by the user, hence the scale is hidden.

Painting becomes slower with increasing number of prefabs

Unity paints parents and children of selected gameobjects including colliders. If that becomes an issue, just drag the container out of the YAPP gameobject so that it doesn't become selected implicitly. You can put it anywhere in the hierarchy.

Credits

- Sebastian Lague

Physics Simulation

<https://www.youtube.com/watch?v=SnhfcdtGM2E>

Patreon

<https://www.patreon.com/SebastianLague>

- Kyle Halladay

Spline

<http://kylehalladay.com/blog/tutorial/2014/03/30/Placing-Objects-On-A-Spline.html>

- Fernando Zapata

Interpolate.cs

<http://wiki.unity3d.com/index.php?title=Interpolate#Interpolate.cs>

- Gregory Schlossmoff

Poisson Disc Sampling in Unity

<http://gregschlom.com/devlog/2014/06/29/Poisson-disc-sampling-Unity.html>