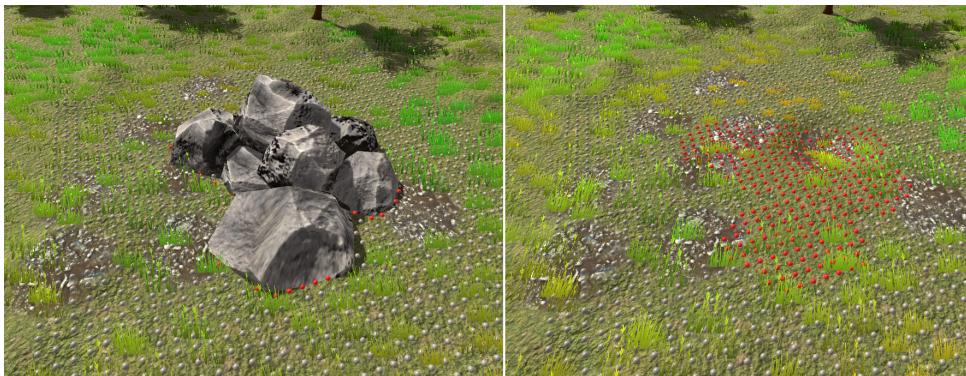


## Deformer Class

The Deformer class is the primary interface for manipulating terrain at runtime. When an instance of the Deformer Script is attached to a GameObject, several things happen:

1. An instance of the [Area](#) class is created. An Area represents an adjustable space around the Deformer object that tracks Terrain space coordinates. Anytime you move the Deformer object (and the `keepTerrainAreaUpdated` setting is turned on) the Area instance is updated with new Terrain space values. In the screenshot below you are seeing a debug view of a Deformer's Area instance. The grey spheres each map to a Terrain heightmap [Vertex](#). The red spheres indicate vertices where the Deformer object intersects with the Terrain. These are the vertices that currently would be directly modified by any height adjusting functions (Add, Subtract)



2. A [LocalHeightMap](#) is instantiated. This is a cached representation of the Deformer object's geometry represented in terrain heightmap values. If your Deformer object has a Box or Sphere collider, then simple math is used to calculate this mapping of the collider. If a MeshCollider is used then raytracing is used to find the min and max height for each terrain vertex. This is obviously more expensive than a Box or Sphere, but it is only done when the Deformer is first initialized or if it is rotated/scaled. Simple transform movement will use the previously cached mapping for all colliders.
3. A [LocalAlphaMap](#) is instantiated. This is a cached representation of which positions within the terrain Area should be affected by texture updates. It behaves very similar to LocalHeightMap, but tracks only a 2d representation of where

the Deformer currently intersects with the terrain's alphamap grid. When you call Texture() on the Deformer, this is what determines where those texture updates will occur. This has to be tracked separately from the HeightMap because heights and alphas can, and usually are, set to different resolutions on the terrain.

4. Current position, scale, and rotation are tracked on the Deformer so that if they change we can update the Area instance and LocalMaps accordingly. Note this only occurs if the variable "keepTerrainAreaUpdated" is set to true (default).

## Variables

---

### **terrainAreaPadding**

How many vertices should we pad out from the objects bounding box? Default is the objects bounding box + 1

### **terrainTextureIndex**

Texture to apply when updating the alphamap. Must be added to the Terrain object via the inspector and then referenced by its index 0, 1, 2, etc

### **terrainTextureOpacity**

What opacity should any texture updates be set to - 0.0 to 1.0

### **alphaPadding**

padding of texture area in terrain units

### **AlphaFitType**

Should the alphamap be changed for any column that is touching the deformer (column fit) or should we only change for cells completely inside the collider (vertex fit)

### **HeightFitType**

Should the heightmap be changed for any column that is touching the deformer (column fit) or should we only change for cells completely inside the collider (vertex fit)

### **keepTerrainAreaUpdated**

If checked, any change to the position or rotation of the game object with this script will update the location and size of the Terrain Area so that subsequent terrain modifications will occur in the correct location and in the correct shape. Turning this off is more performant if you are able to call "UpdateAreaPosition()" manually, but if an object is making continuous updates to a terrain while moving you probably want to have this set to true (on).

### **terrain;**

The terrain instance you want this object to modify. Defaults to Terrain.activeTerrain if not set. Cannot currently handle multiple terrains, but you could always switch them on the fly via scripting.

## **Public Functions**

---

### **AddNow()**

Adds height to the terrain immediately. Any portion of the object that is higher than the terrain Area will be raised to match the object's geometry. If the object is lower than the terrain, no changes will be made.

### **Add()**

Exactly the same as AddNow(), but does not automatically commit the height changes to the terrain. This allows you to batch together multiple height edits to the same Area with minimal performance impact. Call SetHeights() to commit height changes set to the Area by Add().

### **SubtractNow()**

Subtracts height from the terrain immediately. Any portion of the object that is lower than the terrain Area will be removed to match the object's geometry. If the object is higher than the terrain, no changes will be made.

### **Subtract()**

Exactly the same as SubtractNow(), but does not automatically commit the height changes to the terrain. This allows you to batch together multiple height edits to the same Area with minimal performance impact. Call SetHeights() to commit height changes set to the Area by Subtract().

### **SmoothNow()**

Immediately smooths terrain relative to the Deformer object's position. Only smooths terrain within the boundary of its terrain Area (the object + padding)

### **Smooth()**

Exactly the same as SmoothNow(), but does not automatically commit the changes. Call SetHeights() to commit height changes set to the Area by Smooth().

### **TextureNow()**

Immediately update the texture relative to the Deformer's position, rotation, scale, and any alphaPadding specified for this object.

### **Texture()**

Exactly the same as TextureNow(), but does not automatically commit the changes. Call SetAlphas() to commit alpha changes set to the Area by Texture().

### **Displace()**

Simulates the creation of a crater if the object is positioned partially below the terrain. Creates a pedestal type formation when the object is positioned above the terrain. This function is a combination of Subtract(), Smooth(), and Texture() which demonstrates a simple way the core functions can be composed together to create more complex deformations.

### **ShowAreaVertices()**

[DEBUG] Create a grid of spheres that help visualize the terrain area and which vertices are inside of the deformer object's collider

### **ShowAreaAlphas()**

[DEBUG] Create a grid of boxes that help visualize the terrain area alpha columns and which of those columns are inside of the deformer object's collider

### **DestroyDebugObjects()**

[DEBUG] Remove any debug objects in the scene created by this class

### **UpdateAreaPosition()**

Manually update the Deformer instance position, scale, and rotation for the Area and LocalMaps. This is necessary to call before saving heights or alphas only if "keepTerrainAreaUpdated" is set to false so that the updates occur in the correct place on the terrain.

## **SetHeights()**

Update the terrain with any height changes currently queued. Used when making changes to the terrain in batches (ie: an Add() + Smooth()). In other words: if you are not using the "now" version of functions ("AddNow() vs. Add()") then you need to call this function in order to see your changes show up on the terrain.

## **SetAlphas()**

Update the terrain with any alpha changes currently queued. If you want to make several texture changes to a stationary terrain Area, then its best to batch them by using Texture() instead of TextureNow() and then manually calling this function when done.