v2.0

Deformable Surface Documentation

1. Overview

This asset allows you to create and manage the system with realtime volumetric surface deformation. You might already noticed similar feature in some AAA titles. Such feature will definitely make your game more dynamic and realistic looking. To achieve the best performance it uses true power of compute shaders along with DX11 hardware tessellation.

- Edit surface with brush or import heightmap. Just like Unity built-in Terrain
- Well-configurable. Mesh size, map resolution, blend height, deformation smoothing, speed, layer mask and tesselation settings
- Well-optimized. Let GPU deal with it!
- Deformation state save/load and physics implementation
- PBR shaders

2. Quick Start

a) After package has been imported, go to GameObject > 3D Object > Deformable Surface. This will create new object named New Deformable Surface X on your scene and scriptable object named New Deformable Surface X in your Assets folder which contains the actual data of the surface. Select New Deformable Surface X object on your scene and go to settings tab to configure it.

Settings Tab

Field	Description
Surface Data	Link to the scriptable object with surface data
Deform Enabled	Enabled or disables deformation during runtime
Width	Base mesh polygons count by X
Length	Base mesh polygons count by Z
Scale	Base mesh scale
Resolution	Base map resolution. Multiplied by mesh width and length to determine the final resolution
Max Height	Maximum height in units
Max Offset	Maximum offset in units (deformable part of the surface)
Deform Mask	Layer mask used to determine which objects on the scene may deform the

	surface. It is reasonable to keep there only layers with dynamic objects which may actual deform the surface to avoid unnecessary batches
Deform Speed	Framerate dependent deformation speed of the surface
Smoothing	Apply Gaussian blur to depth texture. Requires additional compute shader passes
Radius	Smoothing radius in pixels
Passes	Number of smoothing passes. NOTE: Higher value may affect performance
Tesselation	
Quality	Number of tessellation quad patches for each base mesh polygon
Distance	Tesselation camera distance in units

b) After you configure the settings go to **Material Tab.** Here you can set texture size and blend height. Material Textures lets you assign two types of textures (height and offset). Normal Intensity adjusts the intensity of base normal map.

c) In **Edit Tab** you may edit the surface as like Unity built-in terrain.

Height and Offset edit modes lets you separately adjust hard and deformable parts of the surface.

Randomizer applies Perlin noise to the surface. Seed may be changed manually.

Import Map lets you import heightmap texture.

Terrain Alignment allows you to put the surface on selected terrain by reading its heightmap data.

3. DeformableSurface component API

Property / Function	**Description**
DeformEnabled : bool	Enables or disables deformation. It is reasonable to disable deformation when it's not needed for performance
SrcMapRt : RenderTexture	ARGB32 RenderTexture with depth data comes from DepthCamera. Renders only opaque objects and renderers that writes to depth buffer
BaseMapRt : RenderTexture	ARGB32 RenderTexture contains height (R channel), offset (G channel) and normal (BA channels)
Restore() : void	Removes all changes done in deformable part and restores initial state of the surface

GetCollisionOffset(Vector3) : float	Returns offset value clamped between 0 and 1 at given point. Returns -1 if the point does not touch surface offset
GetCollisionsOffset(Vector3[]) : float[]	Same as GetCollisionOffset but accepts array of points. Use it for calculate large amount of points for performance. Up to 1024 points

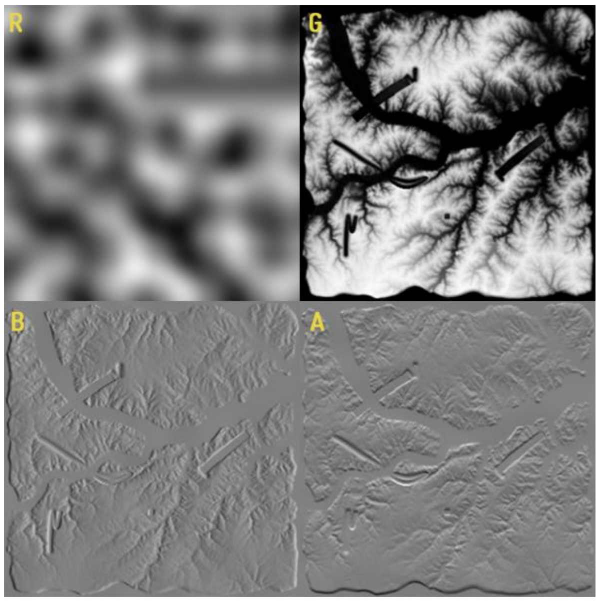
4. Tips

- View the code of RollTheBall component in example folder to learn how to **Save / Load** state of deformation
- View the code of Player component in example folder to learn how **collision physics** may be used along with SurfaceOffsetCollision component
- Use Deformable Surface Cutout shader in order to make the lower part transparent. Useful for blending with terrain surface for example
- For smoother deformation results use sphere-like opaque geometry
- Use separated layer for deformation and lightweight unlit renderers for best performance
- You may disable mesh collider component if it's not needed

5. Custom shaders

If you not satisfied with default Deformable Surface Standard shader you could implement your own. There are few things you need to know for that:

BaseMapRt render texture that processed by compute shader is assigned to material when the game starts. In shader it named as `_baseMap` 2D sampler. All four channels responsible for height (R) offset (G) and normal (BA)



G channel gets updated based on SrcMapRt render texture. Normal (BA) is also updated based on R and G channels by compute shader. R and G channels multiplied by `_height_max` and `_offset_max` float properties for vertex displacement. All other properties are optional for custom shader.