

PLANET EDITOR & GENERATOR

Guide

Table of contents

Demo scene.....	3
Planet prefab structure	4
Main functions and variables.....	4
Custom use and integration.....	7
Unity 5.....	8

1. Demo Scene(big_planet).

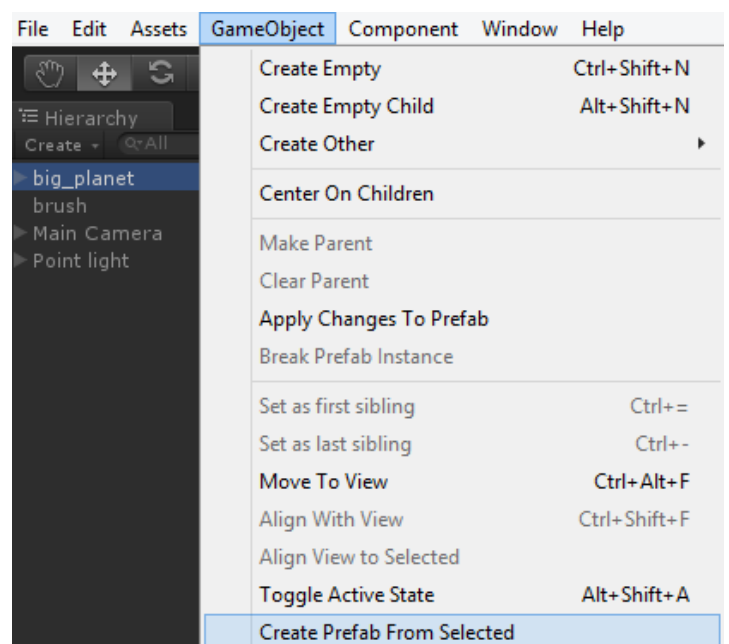
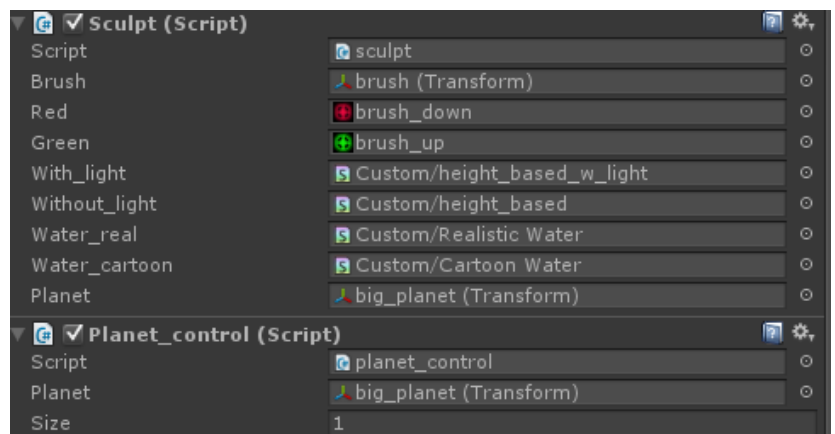
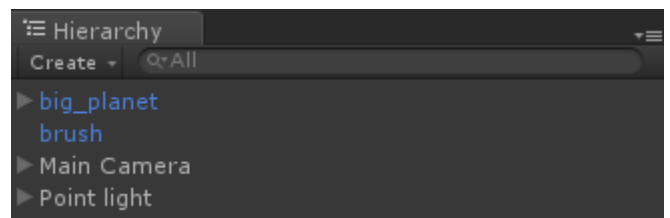
Demo scene is a simple planet editor. Let's check what we have here:

We have a planet prefab, a brush(projector), a Main Camera with 2 scripts

(Sculpt – main script, responsible for planet editing and generating, and planet_control – script, responsible for planet rotation and

providing main variables to the shader) and a light source. If you press play you will see the same scene you have seen in the webplayer demo in the assetstore.

If you want to save the planet, choose it in the hierarchy and press a "Create a prefab from selected" btn in gameobject menu. The planet will be saved into the savedMesh folder.



2. Planet_prefab structure.

Each planet consists of 4 elements:

- Planet mesh with planet shader and 7 blended textures
- Water sphere with water shader
- Glow sphere(special glow shader)
- Clouds sphere with clouds shader and script that makes clouds texture look dynamic.

3. Main functions and variables.

Variables:

- Sculpt - main sculpting script - has 6 main variables:

- 1) Brush transform - brush projector
- 2) Brush texture down (decreasing texture)
- 3) Brush texture up (increasing texture)
- 4) Realistic shader(the one that works with lights)
- 5) Cartoon shader
- 6) Planet transform – current planet

- Planet_control - script that defines planet size and is used to rotate camera around planet. It has 2 main variables:

- 1) Planet transform – current planet
- 2) Size - basically, planet scale how big you want your planet to be.

Unity scales high poly objects very slow, so it's much better to do it only once. Basic planet radius is 25. Use this scale in order to make the planet bigger. You may need to increase light, projector and camera influence zones(range and clipping planes), and light position.

Functions - Sculpt:

OnGUI() – if you need to get rid of gui elements, just comment everything inside this function.

DoMyWindow0-3() – these functions are called inside the OnGUI(), they display the gui elements on screen.

PlanetParameters() – in this function we set standard planet parameters, based on the chosen shader. In the demo scene most of them can be changed using sliders.

DeformMesh (Mesh mesh , Vector3 position , float power , float inRadius) – the most important function. As inputs we have current planet mesh, brush position(deformation position), power of deformation and brush radius. As an output we have the deformed mesh.

Planet_data(RaycastHit hit) – here we just get some data from planet we just hit. May be used for creating a scene with multiple planets awaiting editing.

height_texture_set() - this function is used for setting shader parameters. (Basically applying texture junction, mesh flatness and water height values).

`brush_set()` – here we set the distance from brush projector to planet based on current planet radius.

`Update ()` – function that is called every frame. A few things are done here: brush positioning, checking if the mesh collider needs to be updated, changing clouds sphere size based on the planet size, calling `DeformMesh()` function or `Generate()` function if necessary.

`ApplyMeshCollider ()` – coroutine, that is used to create mesh colliders from mesh.

`recalculate(Mesh mesh)` – this function is used to smooth normal and to get rid of seams.

`Generate()` – function for random surface generation. It uses perlin noise to create random planet surfaces.

Functions – `planet_control`:

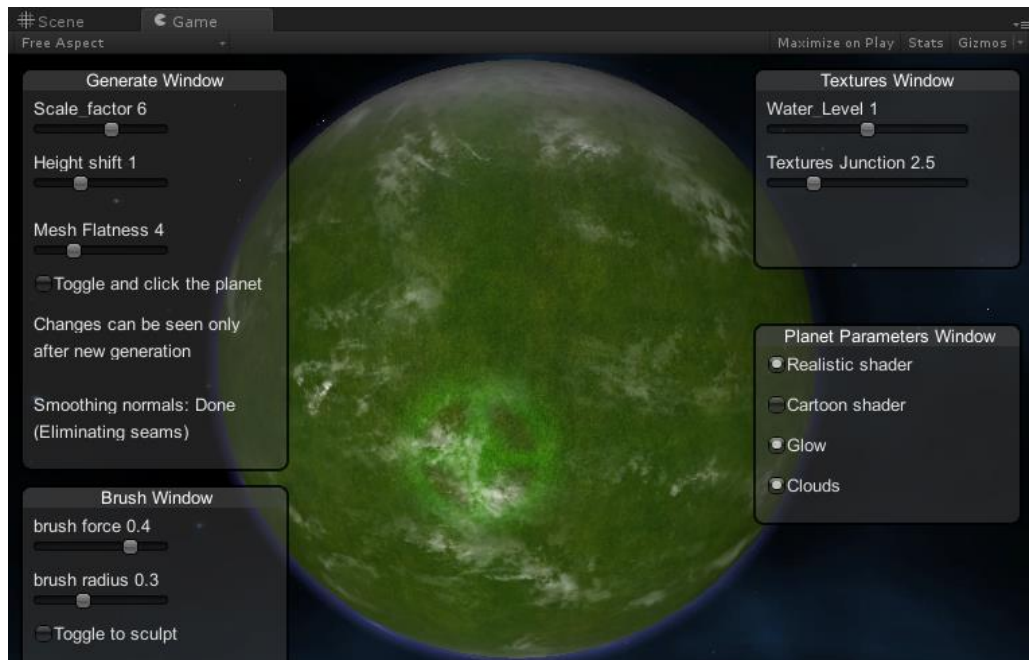
`Awake()` – this function is called at the very beginning, the planet size is being set during it and a few parameters are being send to the current shader.

`LateUpdate()` – this script rotates the planet when the user presses arrows on the keyboard.

4. Custom use.

In order to integrate this asset into your project you need to put the brush prefab(Projectors/Blob light projector) and the planet into the scene. Then you need to assign planet_control and sculpt scripts to the camera and fill the public variables. You might instantiate planet instead of putting the to the scene manually, but in this case, please, don't forget to assign it to the planet variables in sculpt and planet_control scripts by code.

If you don't need GUI elements, you can just comment or delete the code inside OnGUI() function. In this case you might need to replace guibtns controls with your own controls(for example, generate toggle btn is used to allow or not to allow generation of the planet on click).



5. Unity 5.

Demo scene for Unity5 is called Unity5_big_planet. In order to start using it you need to add a brush prefab from

“Planet_Editor\Standard Assets\U5_Projectors\Prefabs”.

Then choose Main Camera and assign brush prefab to Brush variable in a script on Main Camera.

That's it.

All scripts\shaders\prefabs\materials that should be used with Unity 5 only have a prefix like “u5_” or “Unity5_”

Thank you for purchasing this asset.