Paintz

Unity Paint System

Important Notes:
Universal Render Pipeline and Obi require Unity 2019.3 or Newer

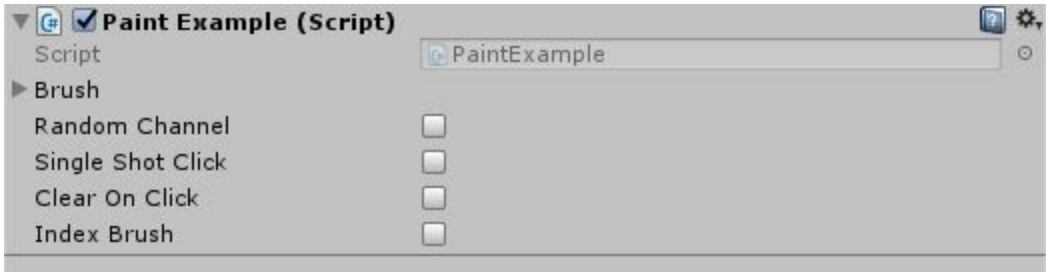
Quick Start

Any object that will be painted will require:

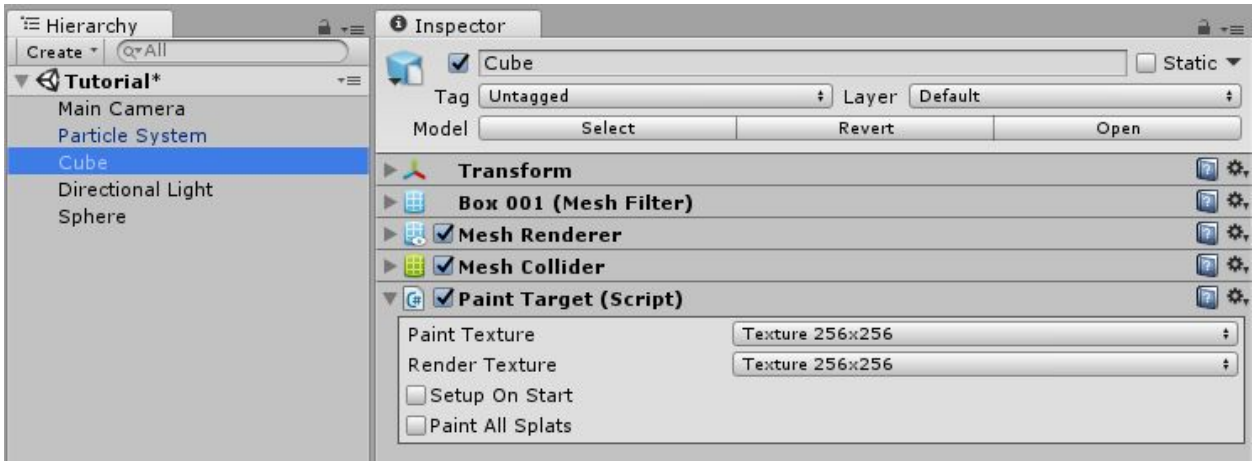
- Paint Target Script
- Material with a Paint Shader

Open Tutorial Scene

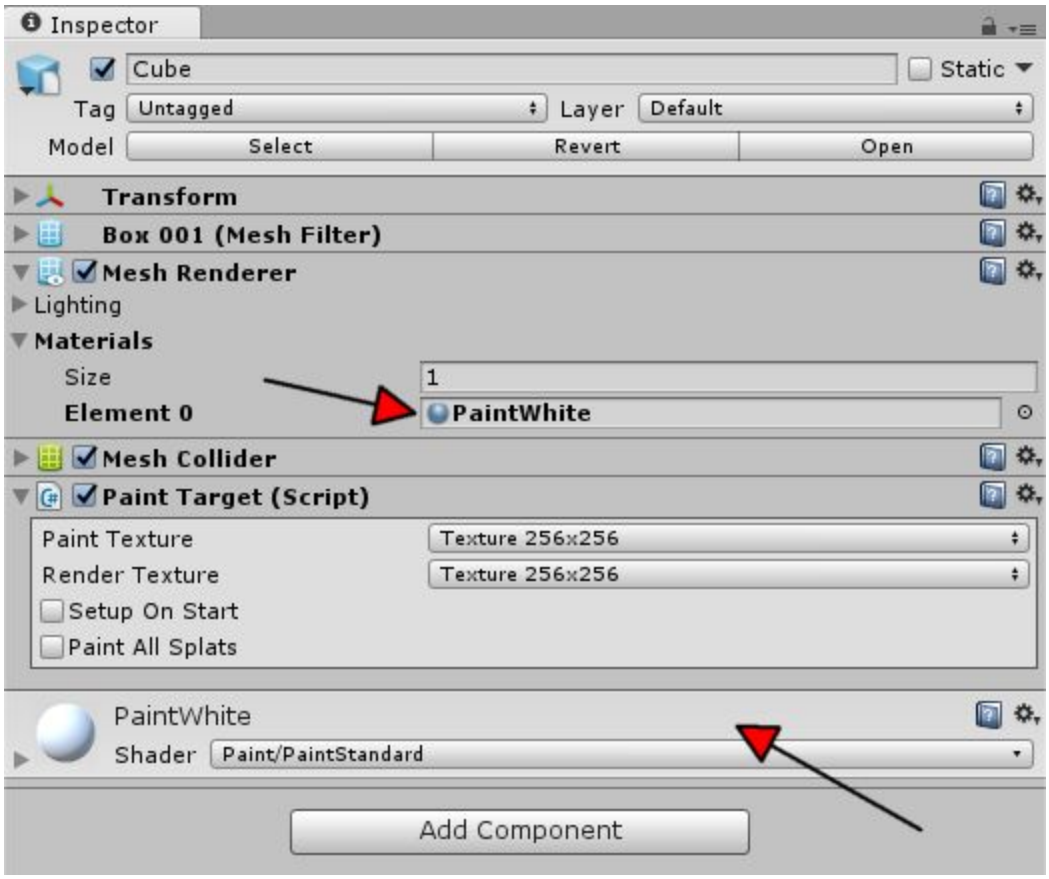
Add “Paint Example” to Main Camera
*This is a sample script that uses Paintz features and basic GUI with WASD controls



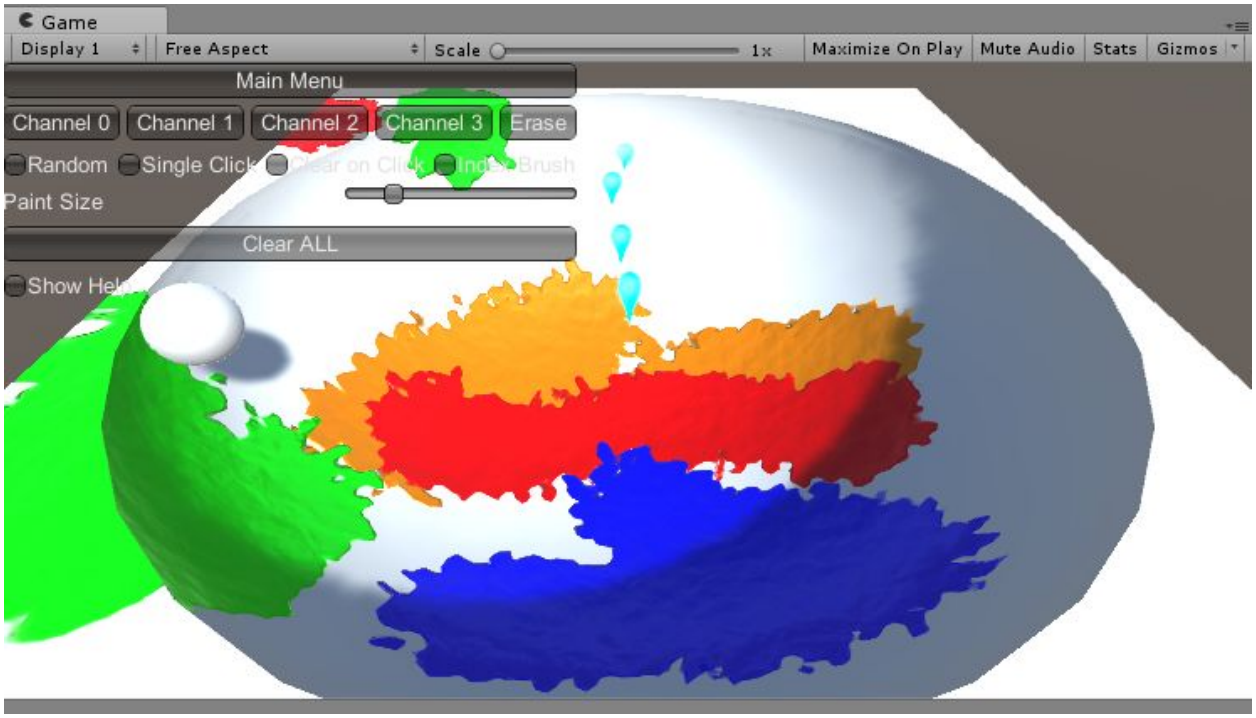
Select “Cube” and add “Paint Target”



Change Material to “PaintWhite”

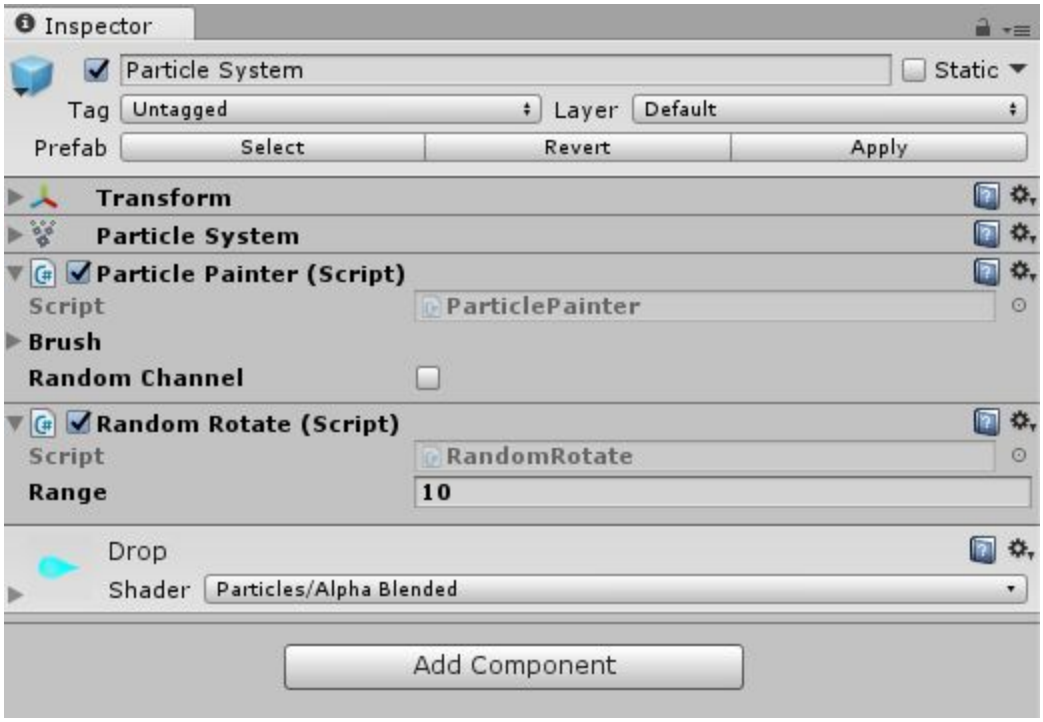


Now at this point you can paint on the “Cube” object!

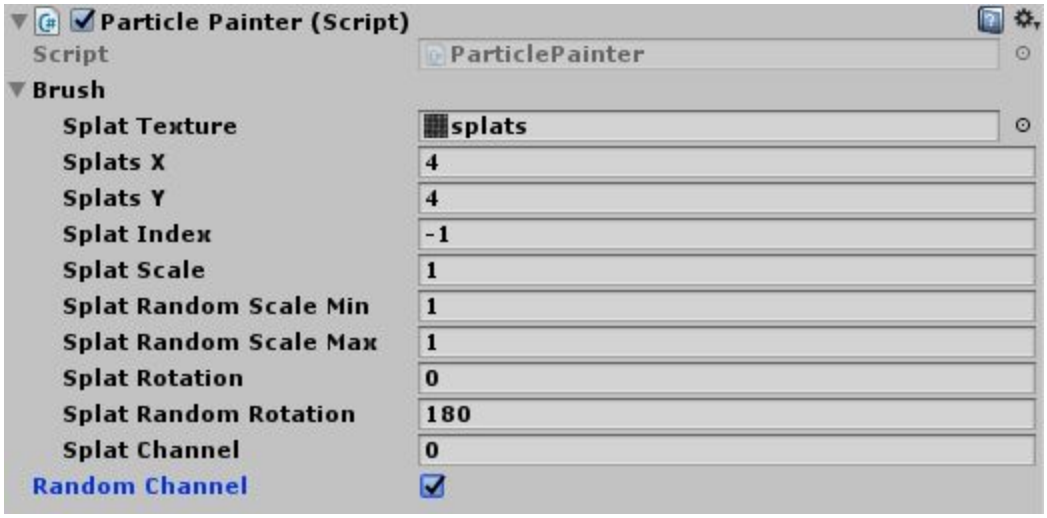


Particule setup

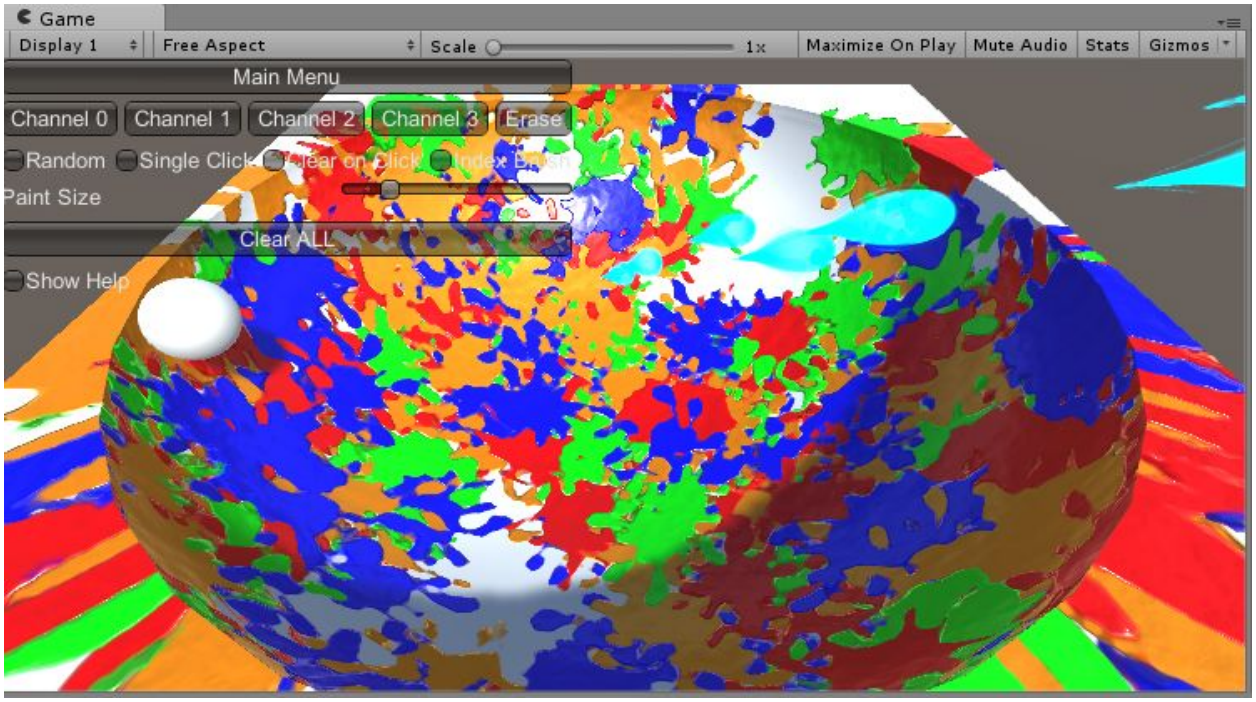
Just add the “Particle Painter” and “Random Rotate” scripts



Add the splats texture and SplatsX and SplatsY (4 x 4)

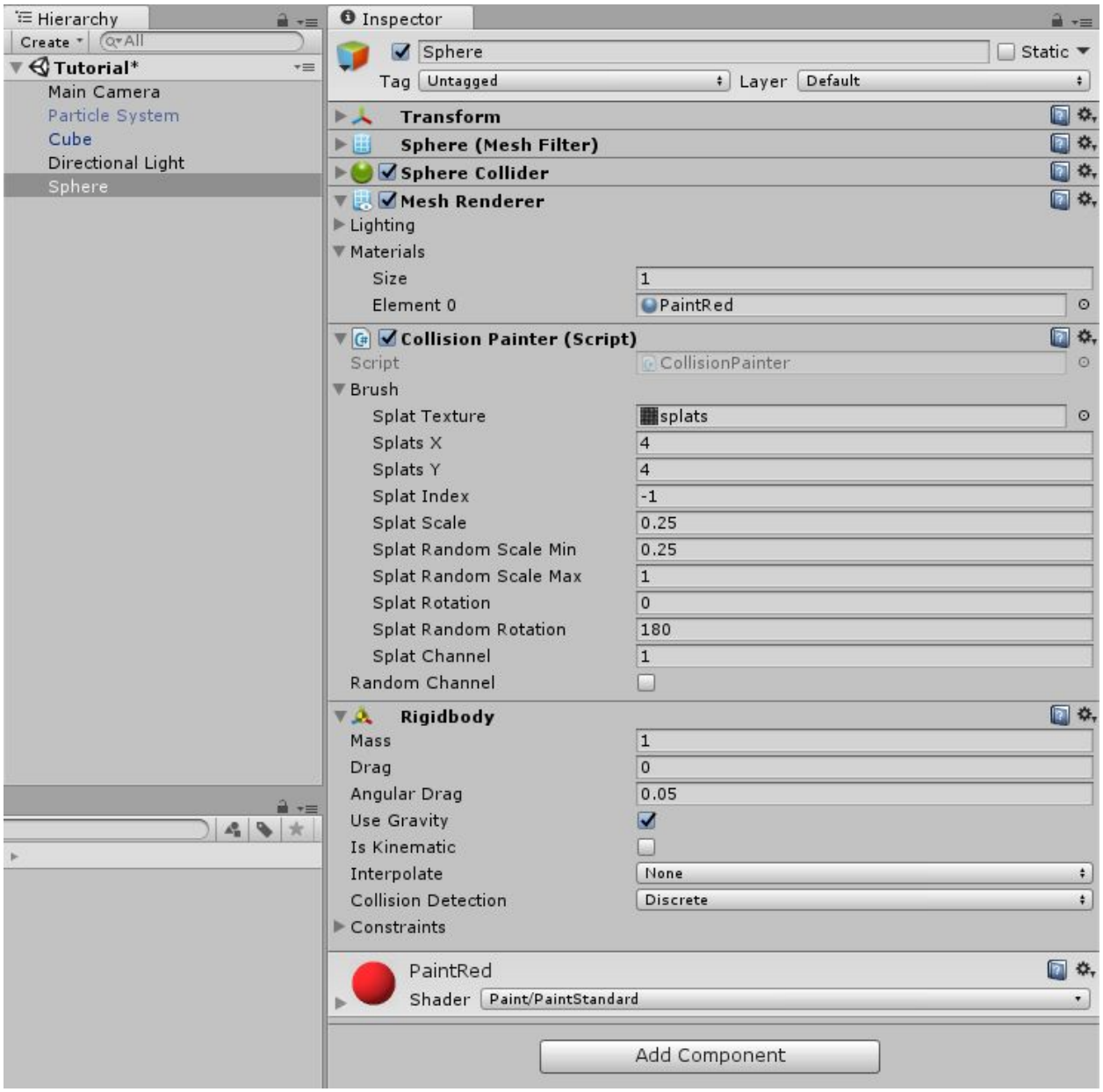


Now the play and you should see this:

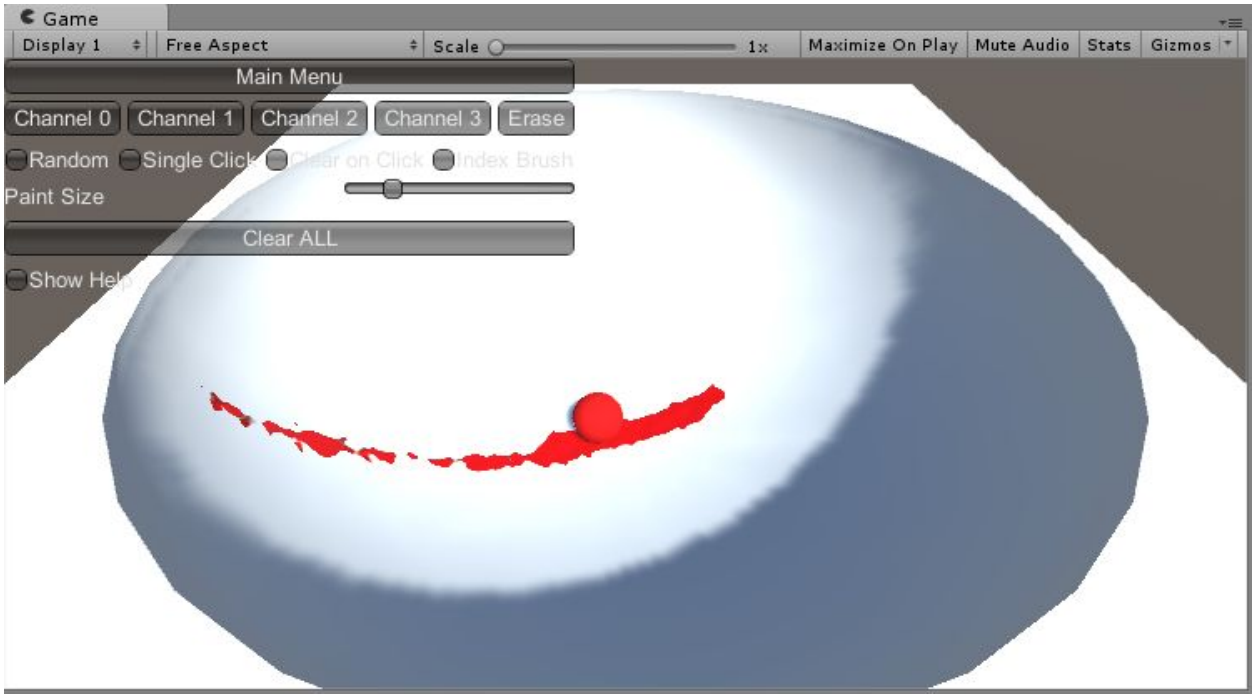


Collision Sphere

Add the “Collision Painter” and “PaintRed” material and Rigid Body
*Add the splats texture and SplatsX and SplatsY
*Change Splat Channel 1



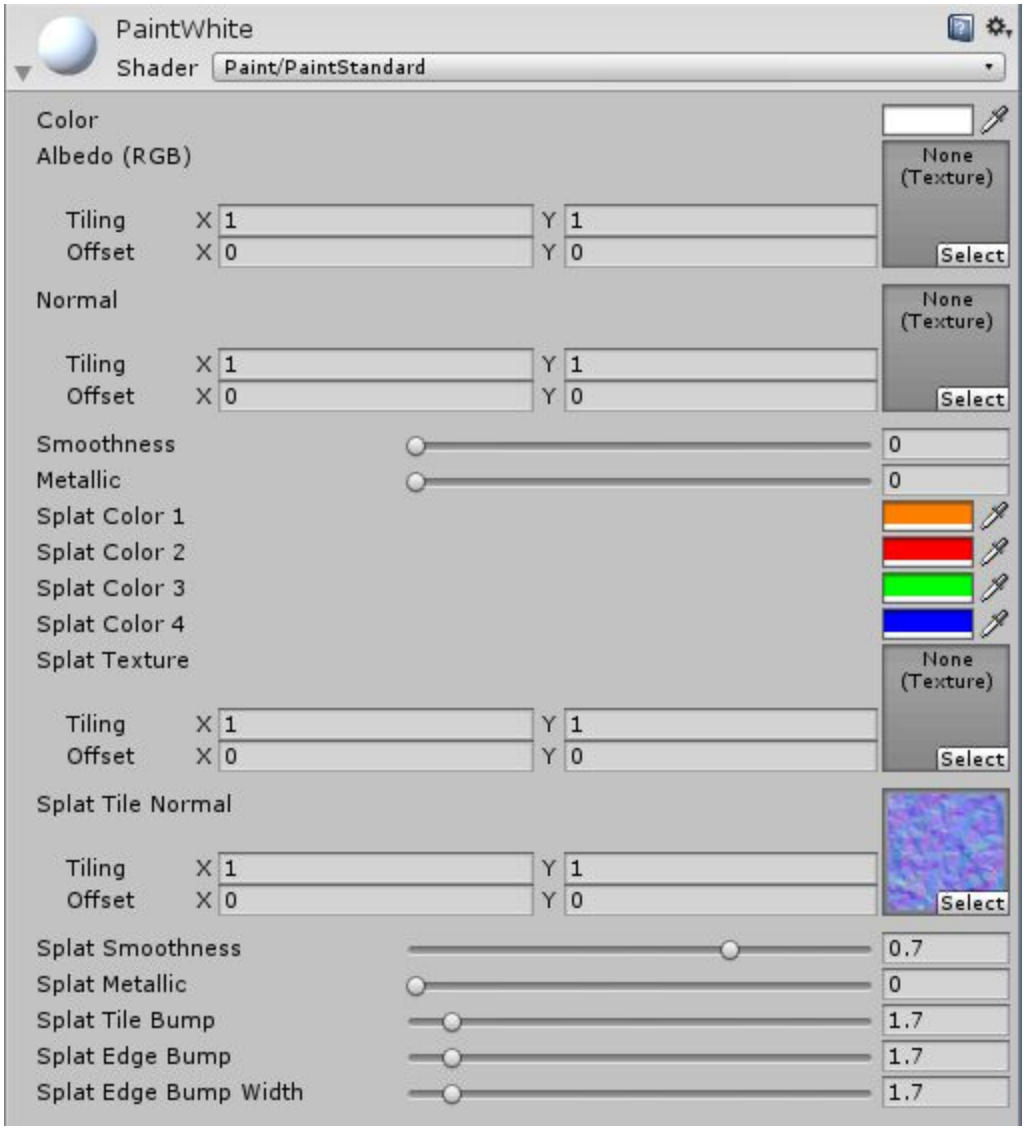
Play and you should see this:



Any object that will be painted

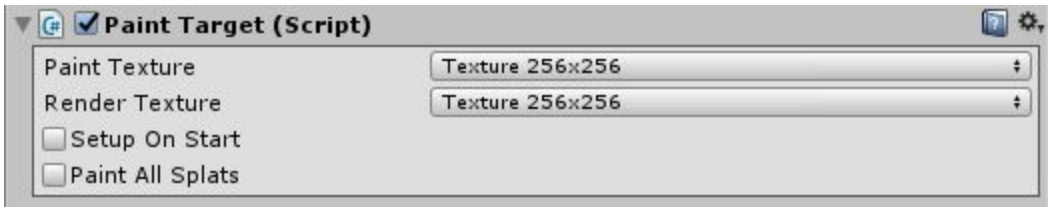
*Needs Paint Target component
Material with Paint Shader

Paint Material

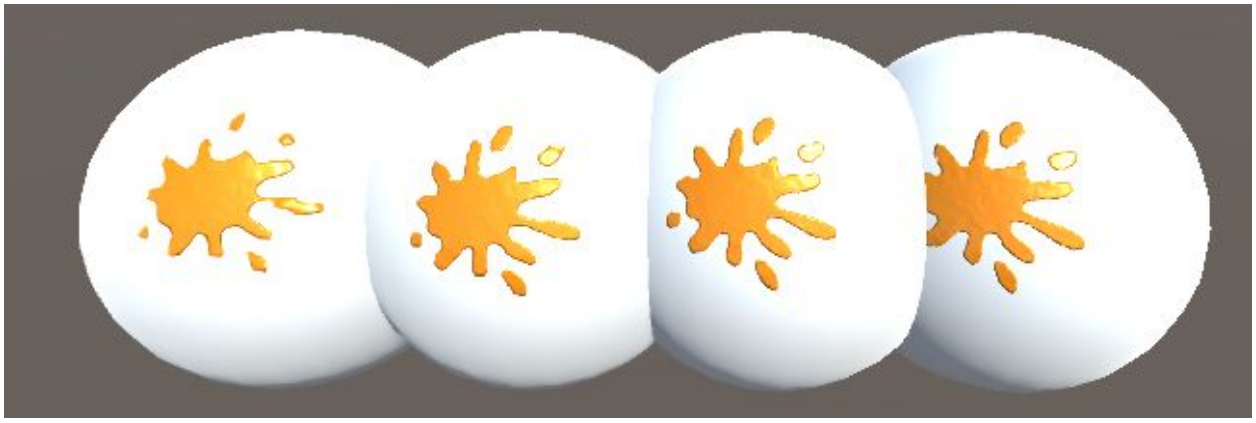


Splat Color 1/2/3/4 - Paint Colors
Splat Texture - LEAVE EMPTY - created by paint target!

Paint Target



Paint Texture - How detailed the paint splats will be

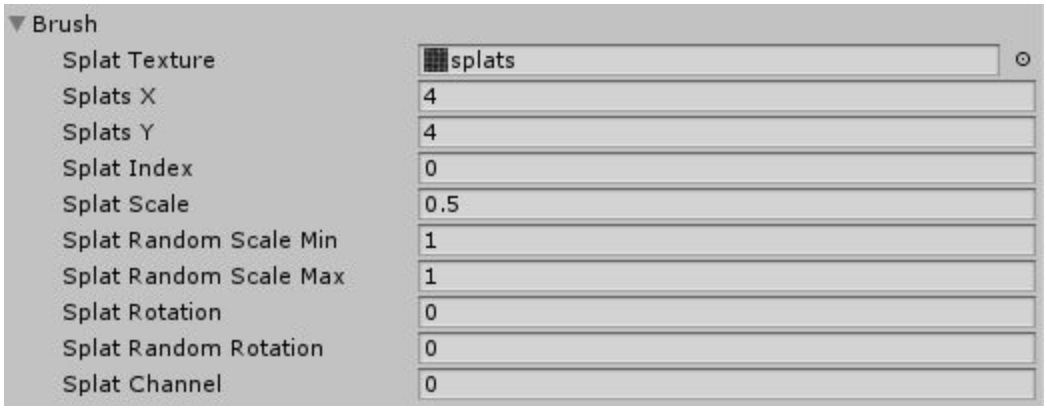


Render Texture - Used for normal/tangent maps (increase if you see artifacts)

Setup On Start - Create textures on startup or wait til object is painted

Paint All Splats - Update ALL splats added, otherwise it will paint 10 per frame

Brush



Splat Texture - Texture used to paint splats
Splats X/Y - Number of “Frames” per texture
Splat Index - Frame offset to paint, -1 = random
Splat Scale - Overall scale to paint a “Splat”
Splat Random Scale Min/Max - Scale splat frame
Splat Rotation - rotate splat
Splat Random Rotation - random rotation added to rotation
*Rotation is added both positive and negative so 180 is 360 degrees

API

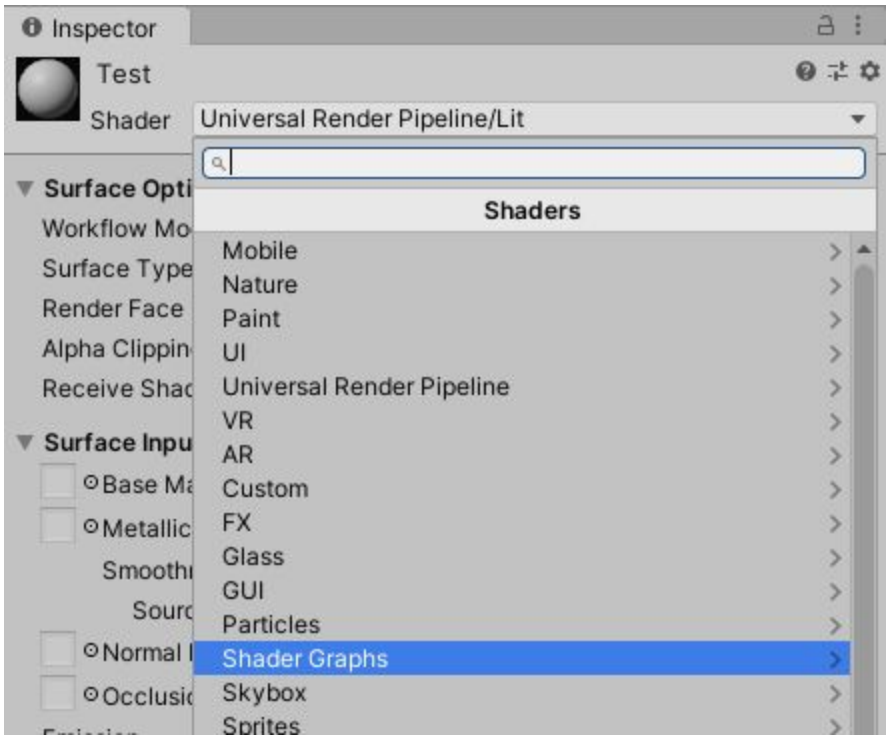
```
public static void PaintLine(Vector3 start, Vector3 end, Brush brush)...
public static void PaintRay(Ray ray, Brush brush)...
public static void PaintCursor(Brush brush)...
private static void PaintRaycast(Ray ray, Brush brush)...
public static void PaintObject(PaintTarget target, Vector3 point, Vector3 normal, Brush brush)...
public static void ClearAllPaint()...
public static void TallyScore()...
```

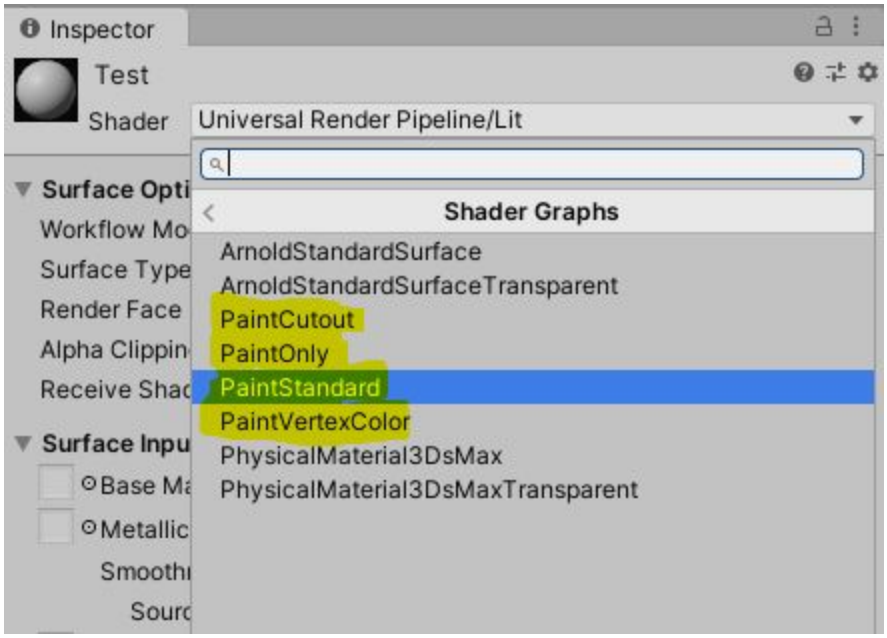
*TallyScore will be SLOW! - suggest only using it for final screen

Universal Render Pipeline

Everything works the same as the Built-in Pipeline, You only need to use URP shaders.

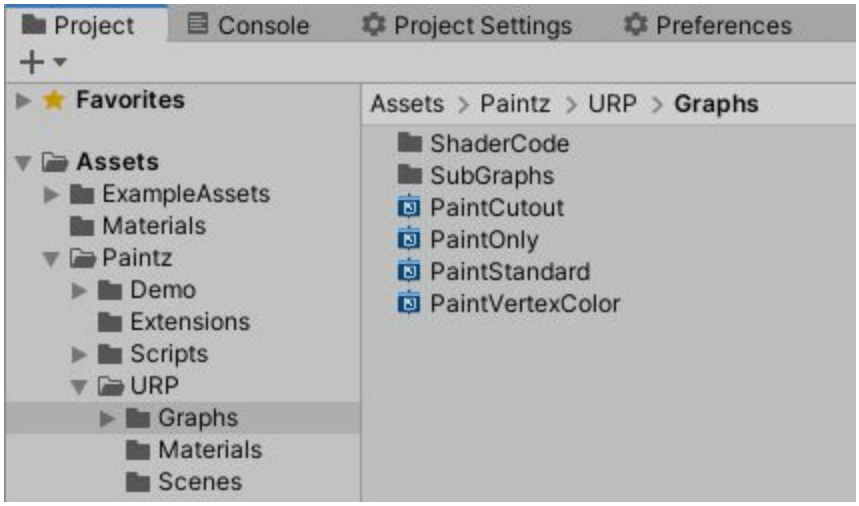
These are standard Shader Graph shaders that can be found here:

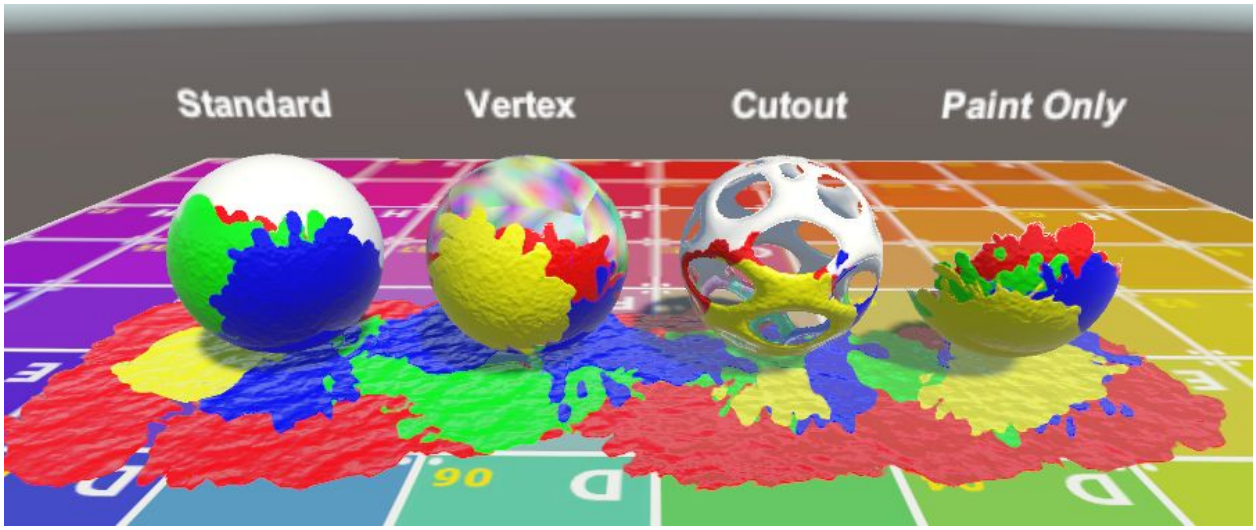




Included are 4 Common Variations, Standard/Vertex Color/Cutout/Paint Only

Suggest you Duplicate the “PaintStandard” Shader Graph, rename it and edit.
***IMPORTANT - You must start the shaders name with “Paint”**

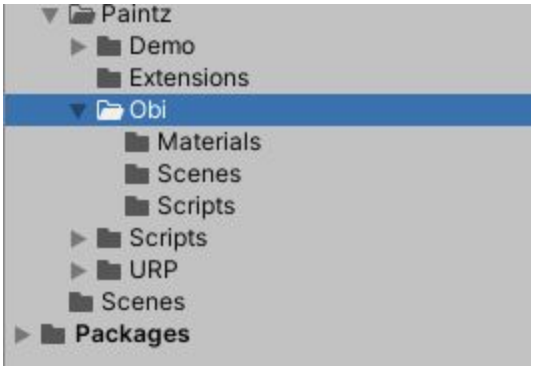




OBI Fluid Extension

IMPORTANT - Import Obi Fluid into your project FIRST
Import the Assets\Paintz\Extensions\Obi Fluid.unitypackage into your project

You should now have the following in your project:



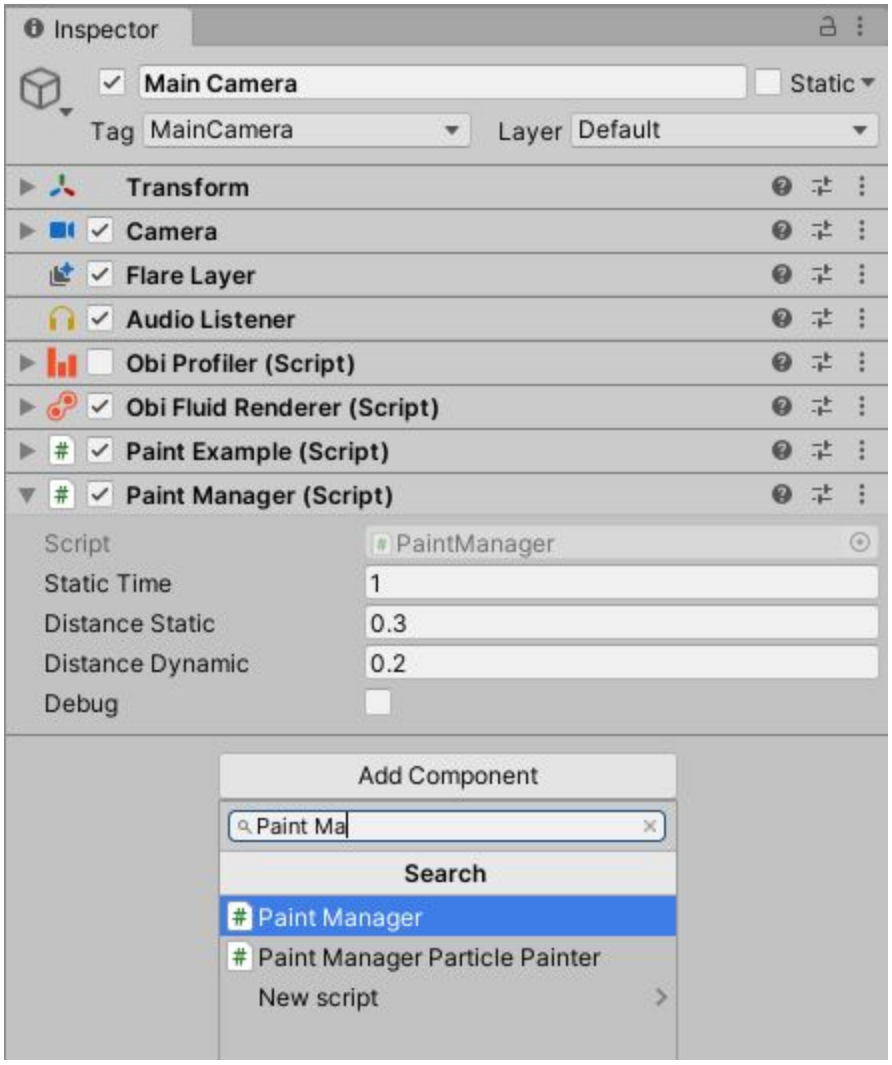
Step 1 - Add “PaintManager” to your Main Camera

Paintz can not handle 1000’s of splats from the tiny Obi particules.
This class will limit “re-painting” an area for a certain amount of time.

Static Time - How long until an area can be repainted
Distance Static - Size of areas to limit painting

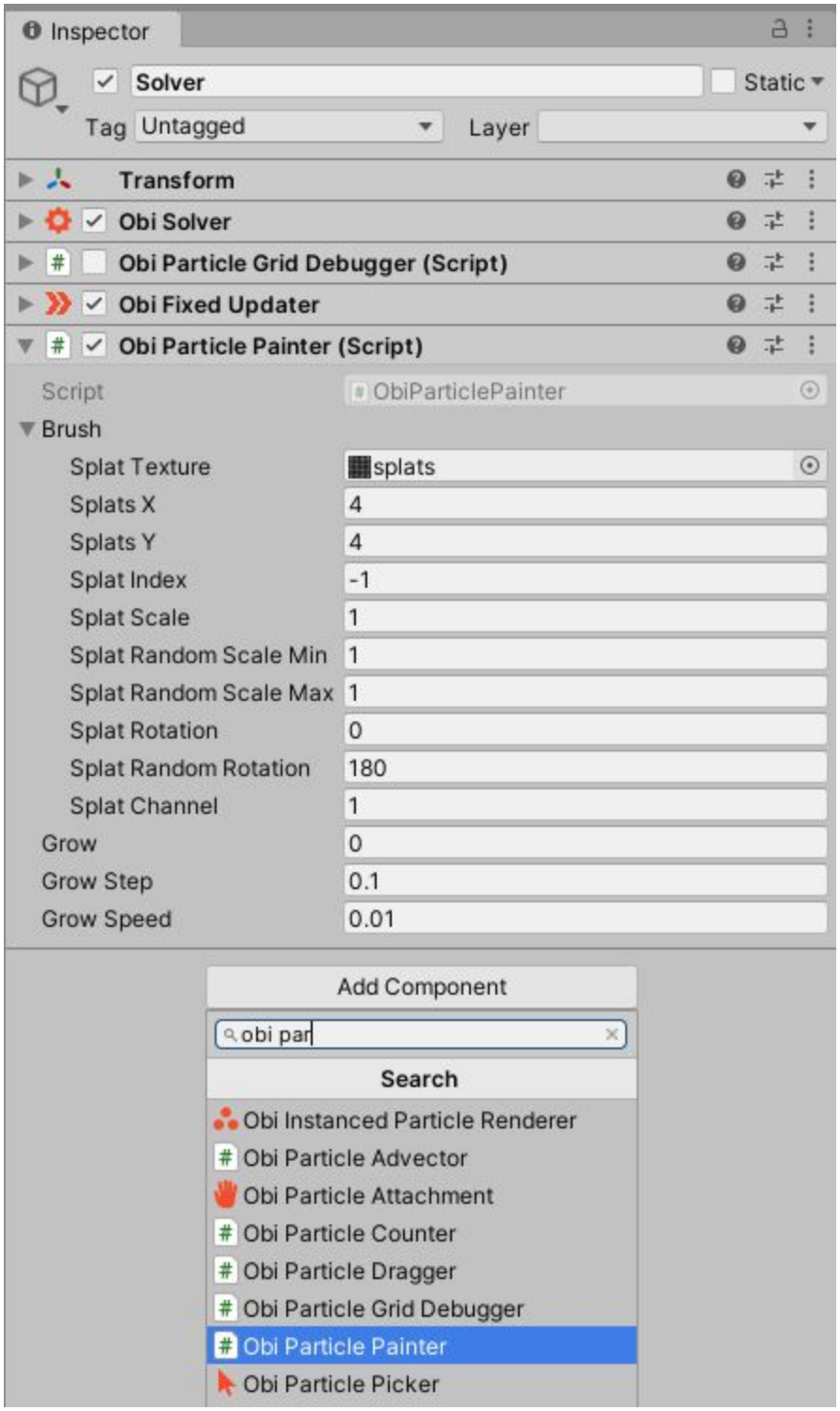
Distance Dynamic - Not used

Debug - Visually see area in editor viewport

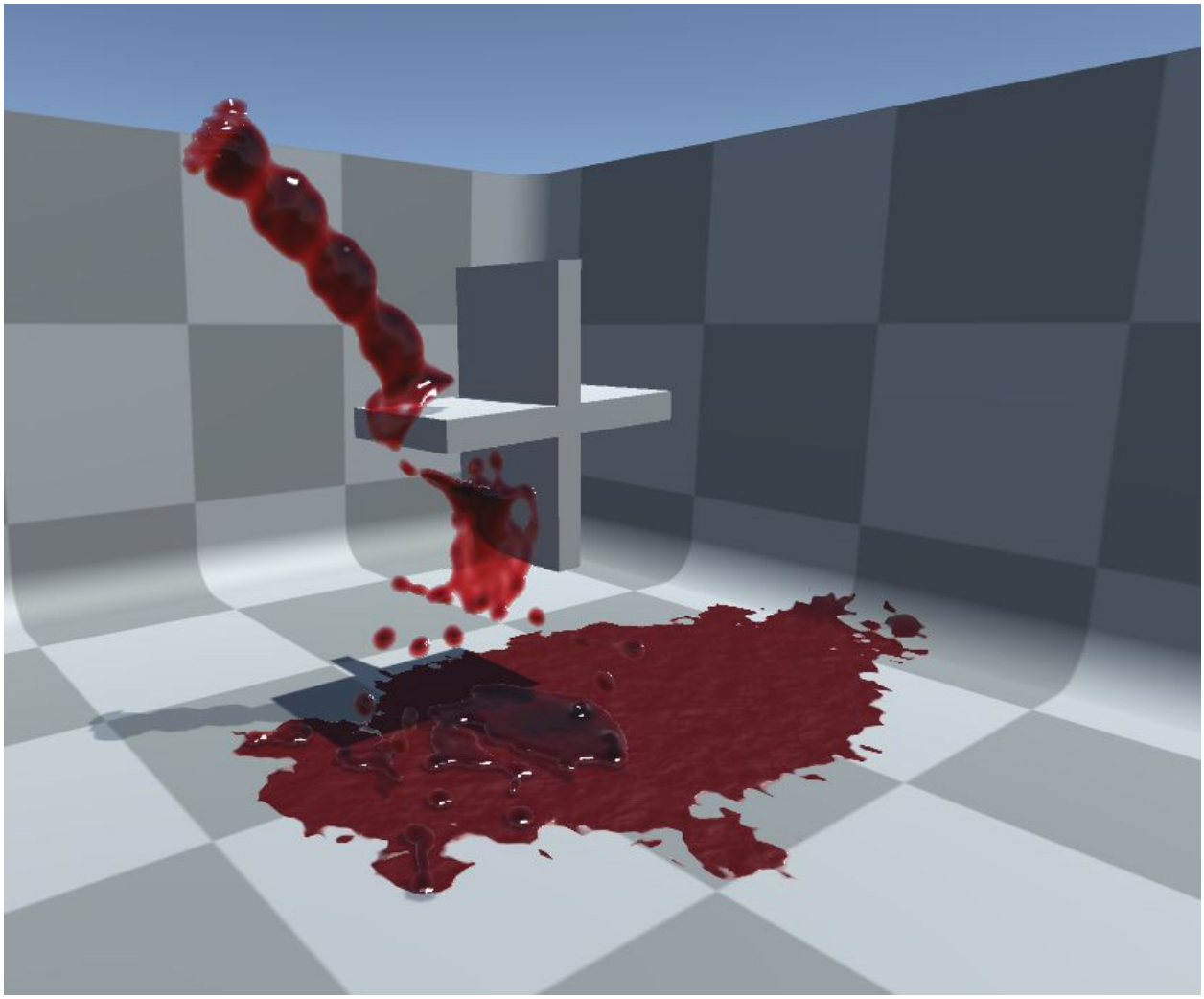


Step 2 - Add “Obi Particle Painter” to your Solver object

***IMPORTANT - Setup your Brush params, Note the Texture and Splats X/Y settings**



That's it, the Obi fluid should paint any valid paint surface.

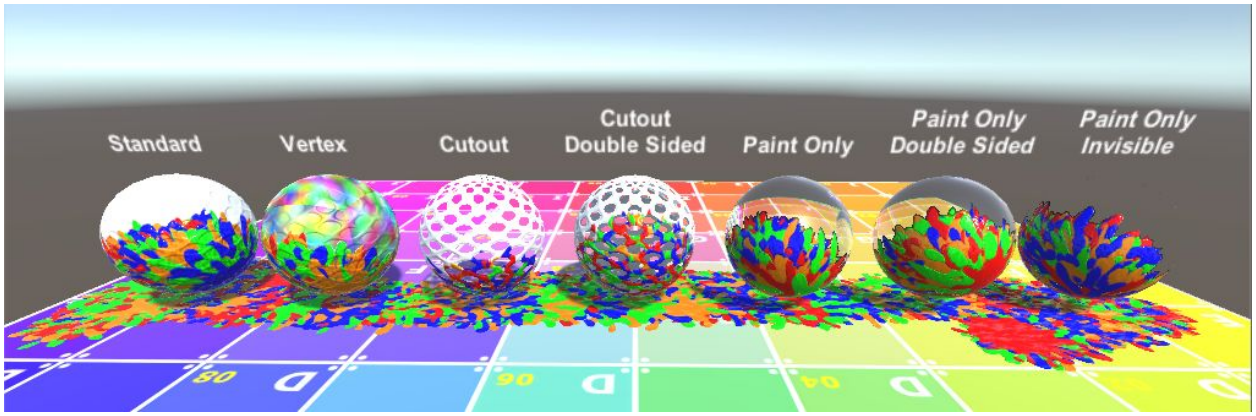


Version 1.5

New API:

```
public static Color CursorColor()
public static int CursorChannel()
public static Color RayColor(Ray ray)
public static int RayChannel(Ray ray)
```

New Shaders:



Version 1.7

Universal Pipeline Support
Obi Fluid Extension