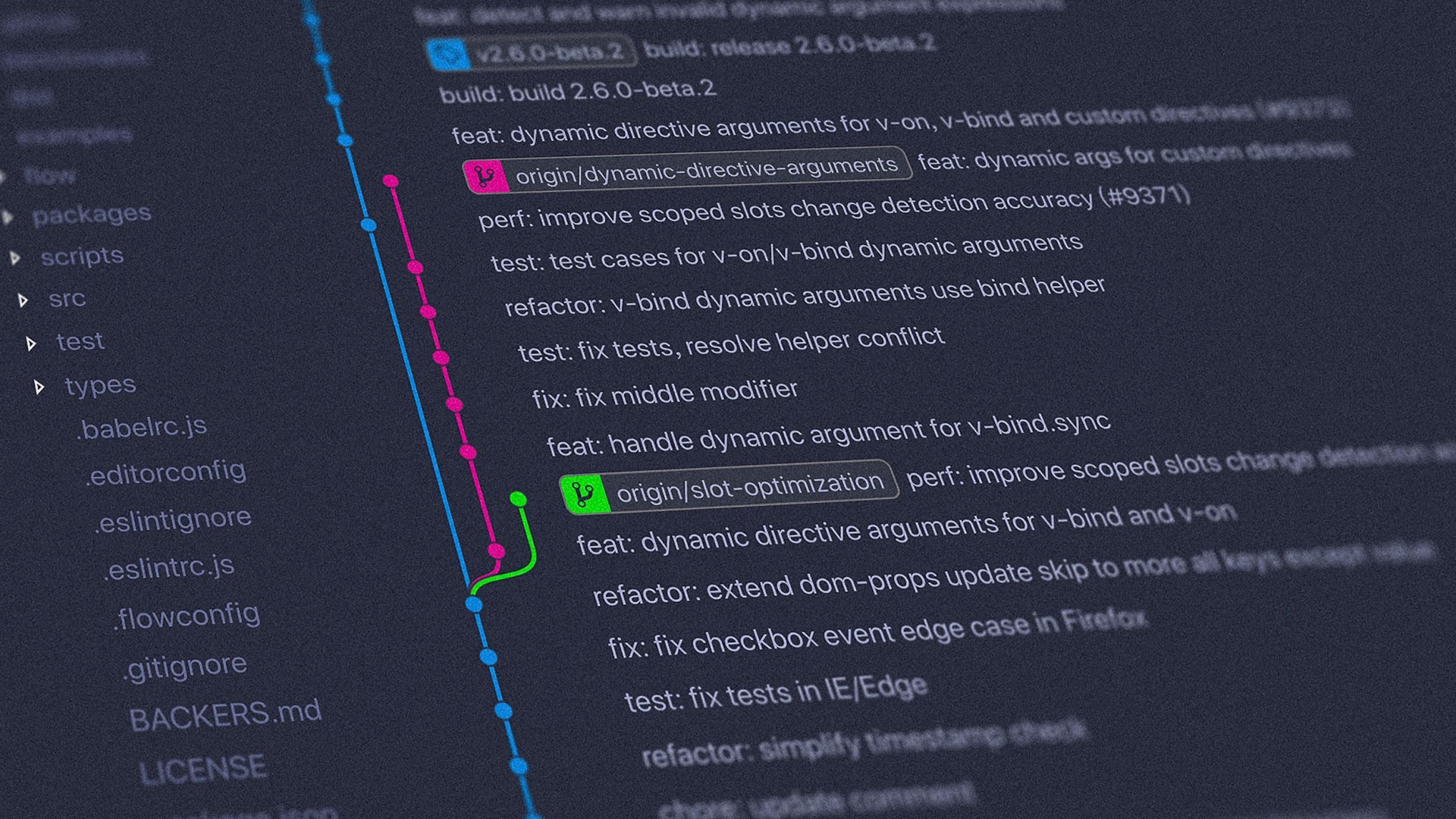
CIS367 - Computer Graphics

Git + Shader Graphs

Erik Fredericks - frederer@gvsu.edu

Based on material from: Unity tutorials: https://learn.unity.com/tutorial/introduction-to-shader-graph-1, https://learn.unity.com/tutorial/shader-graph-vertex-displacement, https://thoughtbot.com/blog/how-to-git-with-unity


v2.6.0-beta.2
build: release 2.6.0-beta.2
build: build 2.6.0-beta.2
feat: dynamic directive arguments for v-on, v-bind and custom directives
origin/dynamic-directive-arguments
feat: dynamic args for custom directives
perf: improve scoped slots change detection accuracy (#9371)
test: test cases for v-on/v-bind dynamic arguments
refactor: v-bind dynamic arguments use bind helper
test: fix tests, resolve helper conflict
fix: fix middle modifier
feat: handle dynamic argument for v-bind.sync
origin/slot-optimization
perf: improve scoped slots change detection
feat: dynamic directive arguments for v-bind and v-on
refactor: extend dom-props update skip to more
fix: fix checkbox event edge case in Firefox
test: fix tests in IE/Edge
refactor: simplify timestamp check
chore: update comment
flow
packages
scripts
src
test
types
.babelrc.js
.editorconfig
.eslintignore
.eslintrc.js
.flowconfig
.gitignore
BACKERS.md
LICENSE

I am going to assume you know Git

If you don't, there are resources such as https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/ out there to help get you started

Also if you don't, I **highly** recommend becoming familiar with it as you can:

1) Maintain version control over your files
2) Check out / contribute to open-source projects
3) Have a nice way to create a portfolio for marketing yourself!

Why are we talking about this?

Git by itself? Great
Unity by itself? Also great

Unity + Git? Problematic!

- Unity creates a **lot** of temporary files to manage
- Unity also deals with a **lot** of files that can be large in size (GitHub has a 100mb limit)

Solution?

- git-lfs

Enter git-lfs

https://thoughtbot.com/blog/how-to-git-with-unity

Basically, we want to:

- Filter out temporary files (they don't need to be tracked with Git)
- Add text links to large files
  - Host on some other file server

Like everything Unity-related in the past year or so...

this has changed a bit!

Both of these approaches will work

Approach 1: Use the GitHub package for Unity

https://assetstore.unity.com/packages/tools/version-control/github-for-unity-118069

Approach 2: Manually handle it!

1) Install git-lfs
   a) Varies per OS -- I had to enable a separate repository and `apt install git-lfs`
   b) If using a graphical client, simply follow the graphical installation procedure on the git-lfs website

1) Setup appropriate .gitignore and .gitattribute files
   b) Windows .gitignore: https://github.com/github/gitignore/blob/master/Global/Windows.gitignore
   c) .gitattributes (from thoughtbot link):

```
# 3D models
*.3dm filter=lfs diff=lfs merge=lfs -text
*.3ds filter=lfs diff=lfs merge=lfs -text
*.blend filter=lfs diff=lfs merge=lfs -text
...
```

   d) Add to project

I had to also filter out *.bc files in addition to the provided ones (add to .gitattributes file)

- However, they were already staged for commit
- Solution: do a reset

Error:

```
build.bc is 117.29 MB; this exceeds GitHub's file size limit of
100.00 MB
```

Solution:

```
git lfs migrate import --include="*.bc"
```

Gotchas

Assets don't seem to be included and will have to be manually resolved if downloading the ZIP file from the website:

DungeonBall-master - MainScene - WebGL - Unity 2019.3.2f1 Personal <DX11>

File Edit Assets GameObject Component Window Help

Pivot Global Collab Account Layers Layout

Hierarchy

- MainScene
  - Main Camera
  - Directional Light
  - Plane
  - Canvas
  - EventSystem
  - Player
  - TorchObject-Player
  - AmbientLight

Scene Game Asset Store

Shaded 2D Gizmos

Top

Inspector

Metal pattern 12

Shader Legacy Shaders/Self-Illumin

Normal mapped shader without a normal map. Consider using a non-normal mapped shader for performance.

Main Color

Specular Color

Shininess 0.208012

Base (RGB) Gloss (A) None (Texture) Select

Tiling X 3 Y 3

Offset X 0 Y 0

Illumin (A) None (Texture) Select

Tiling X 3 Y 3

Offset X 0 Y 0

Normalmap None (Texture) Select

Tiling X 3 Y 3

Offset X 0 Y 0

Emission (Lightmapper) 1

Render Queue From Shader 2000

Enable GPU Instancing

Double Sided Global Illumination

Global Illumination None

Metal pattern 12

AssetBundle None None

Project Console

Favorites

- All Materials
- All Models
- All Prefabs

Assets

- Dynamic Flickering
- Materials
  - Dungeon_stone_t
    - Materials
    - Textures
  - Metal textures pa
    - pattern 12
- PyroParticles
- Stylize Wood Te

Assets > Materials > Metal textures pack > pattern 12

diffuse glow Metal patt... normal preview

Assets/Materials/Metal textures pack/pattern 12/Metal pattern 12.mat

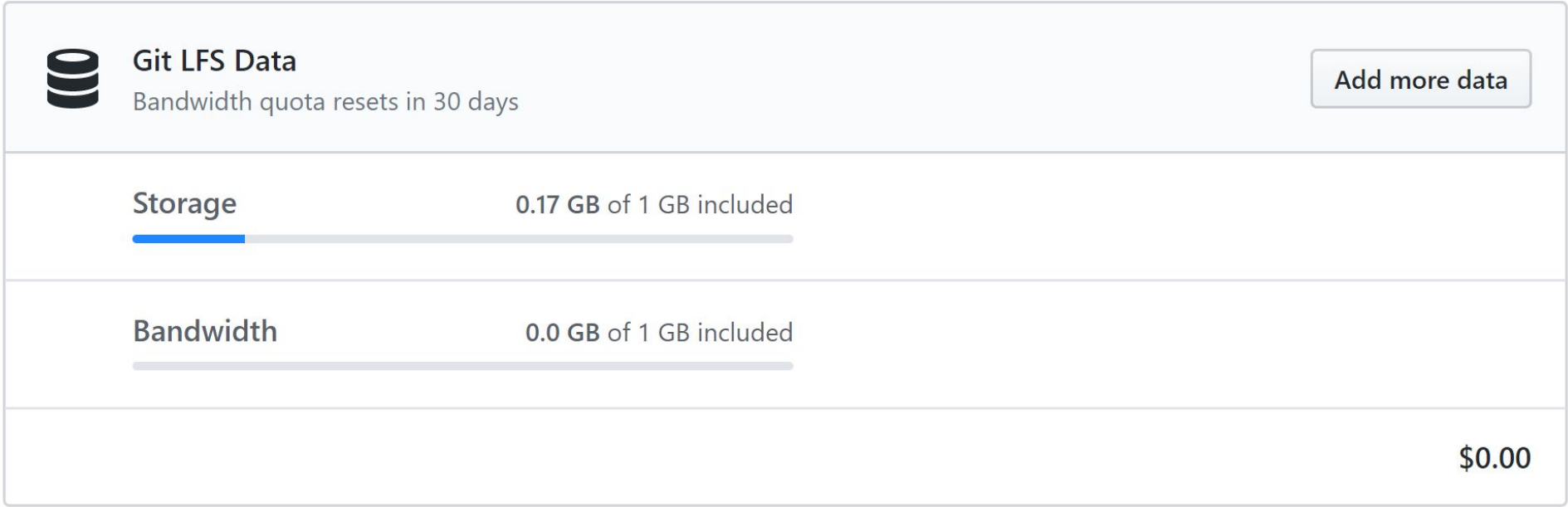
Width: 75 Height: 75 % Walls: 40

Auto Generate Lighting Off

Gotchas

1GB of storage/bandwidth per month in GitHub

- This was the result of 1 push



Cmd line:

You need to setup an SSH agent to automatically login to your git server for you, otherwise every single time it will ask you for your login credentials **per asset**

- You may notice that when you do a git push/clone/etc you will be asked for username and password
  - Consider this happening for every PNG and realize how annoying it can get
  - https://help.github.com/articles/caching-your-github-password-in-git/

Workaround:

Cache your assets elsewhere

- Public FTP site
- Shared folder
- etc.

Only track your project/source files and copy over your Assets later

https://www.reddit.com/r/gamedev/comments/fx5zct/sharing_how_we_made_our_super_cute_toon_shader/

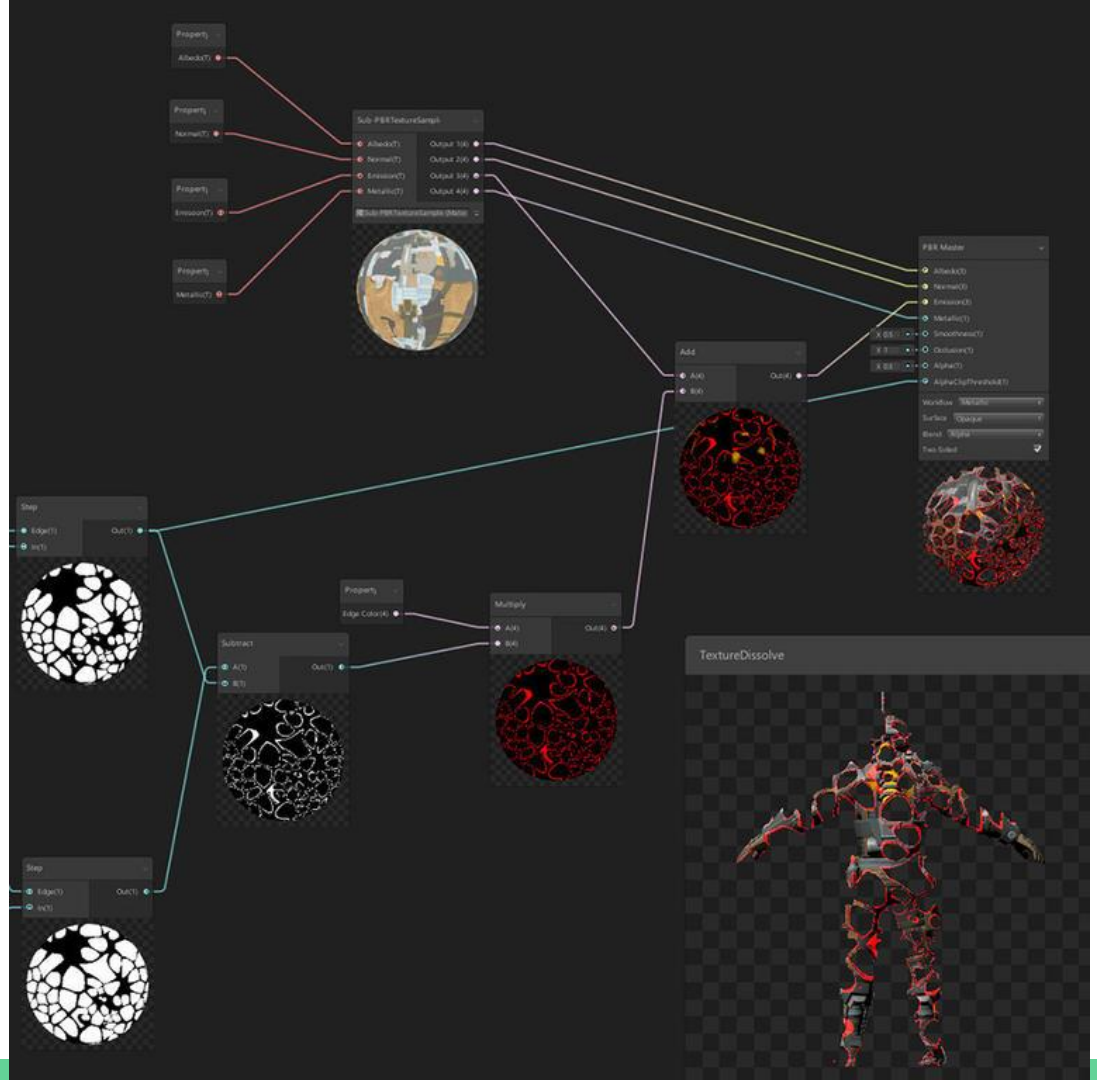
Unity Shader Graph

Node-based approach for building up (complex) materials

Remember the node editor for Blender?

- Similar concept!

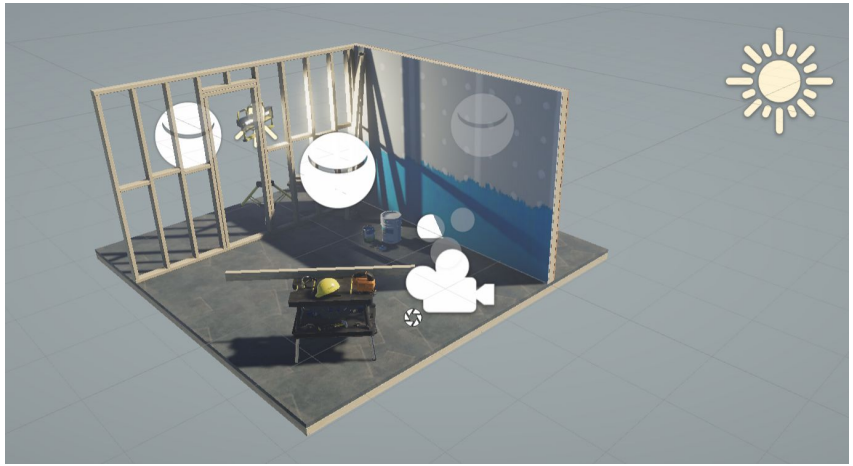
Enables us to edit materials **in real time**, rather than clicking 'Play' to 'see' what 'happens'



Shader Graph in Unity?

Either start from HDRP/URP or install via Package Manager (Window menu)

- Will probably need to install a slew of dependencies, tends to be easier to start fresh
- Note that both will start you with some sample assets

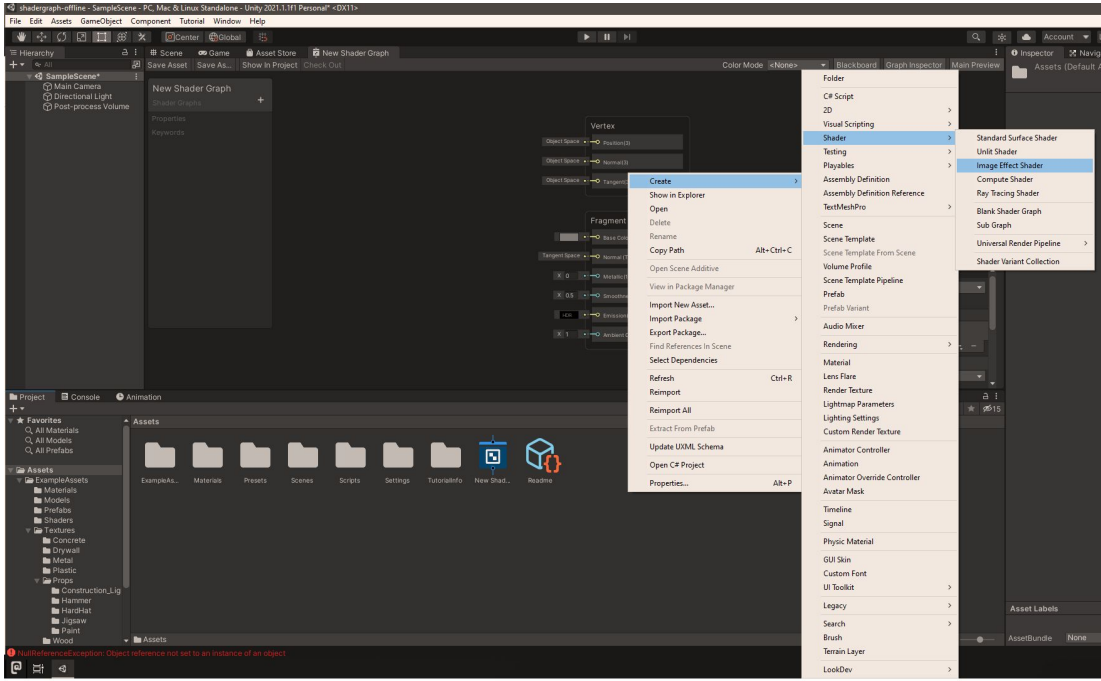


Shader graphs

Unlit : 'basic' graph for materials (among others)

PBR : physically-based rendering

➜ now found under Lit Shader Graph



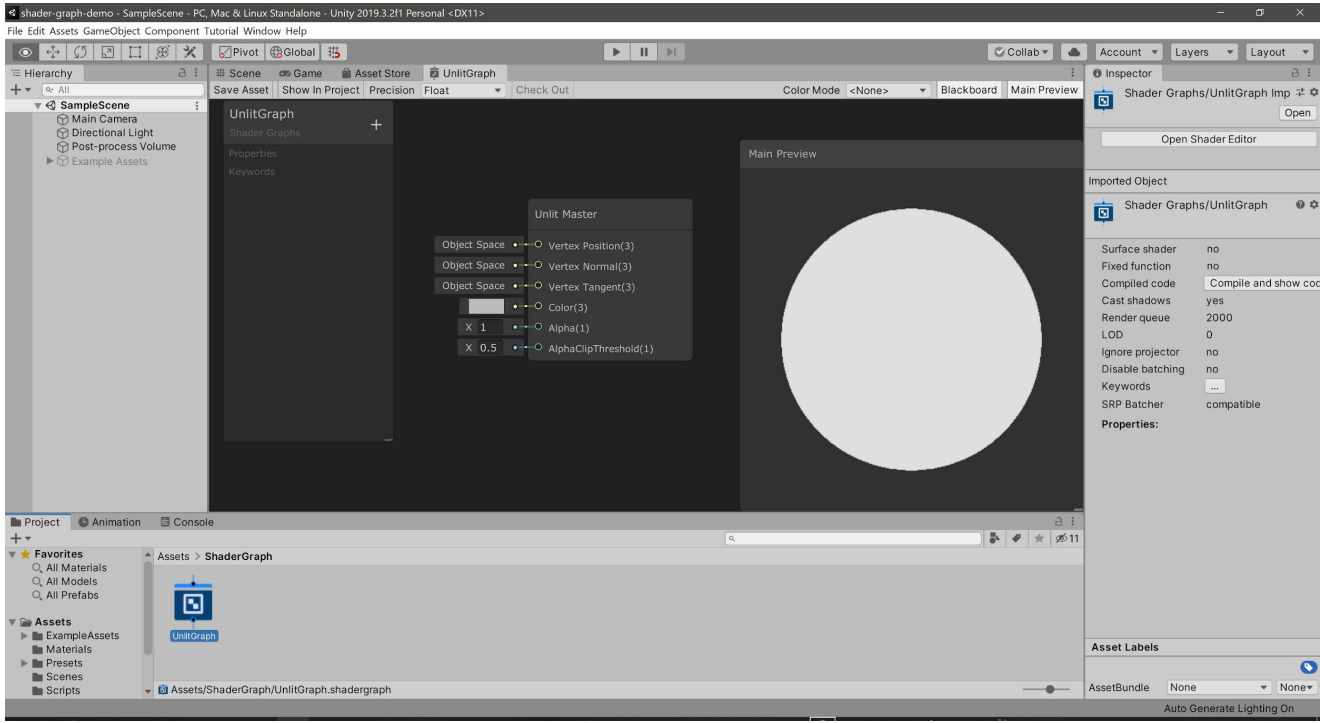
Basic shader graph (Universal Render Pipeline!)

(Assets ➔) Create ➔ Shader ➔ URP ➔ Lit Shader Graph

Double-click on graph asset to open up the node editor

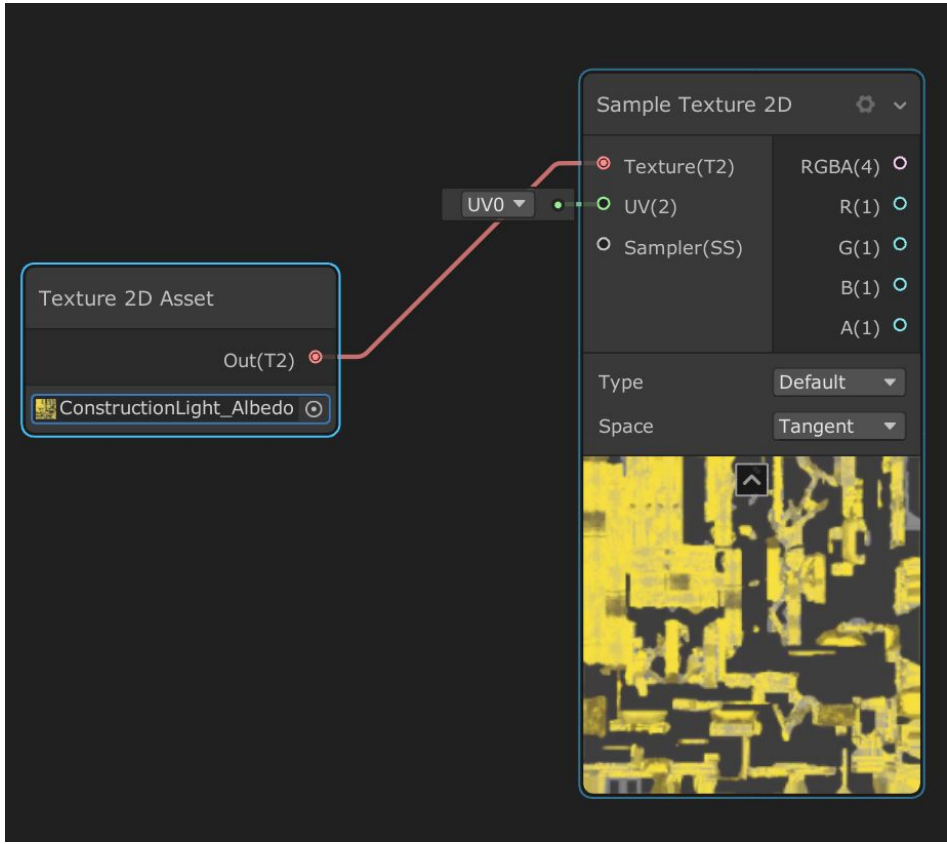
OH NOOOO

(**Shader Graph** + **Scriptable Build Pipeline** packages to install)



Linking up a texture

- Right-click (or click plus) and add a Texture 2D asset and Sample Texture2D
- On Texture2D, click the radio button and select an asset
- Connect Texture2D with Sample Texture2D on the (T) I/O



Select both and duplicate (Ctrl+D)

- In secondary Texture2D, apply a gradient texture (or something that would give it a different texture)

Now add a 'Tiling and Offset' node

- Drag its output to the UV input of the (initial) Sample2D node
- Set tiling to '3'

And blend

Create a Divide node

- Drag A(1) output from initial Sample2D into the Divide node
- Set its (divide) X input to 20

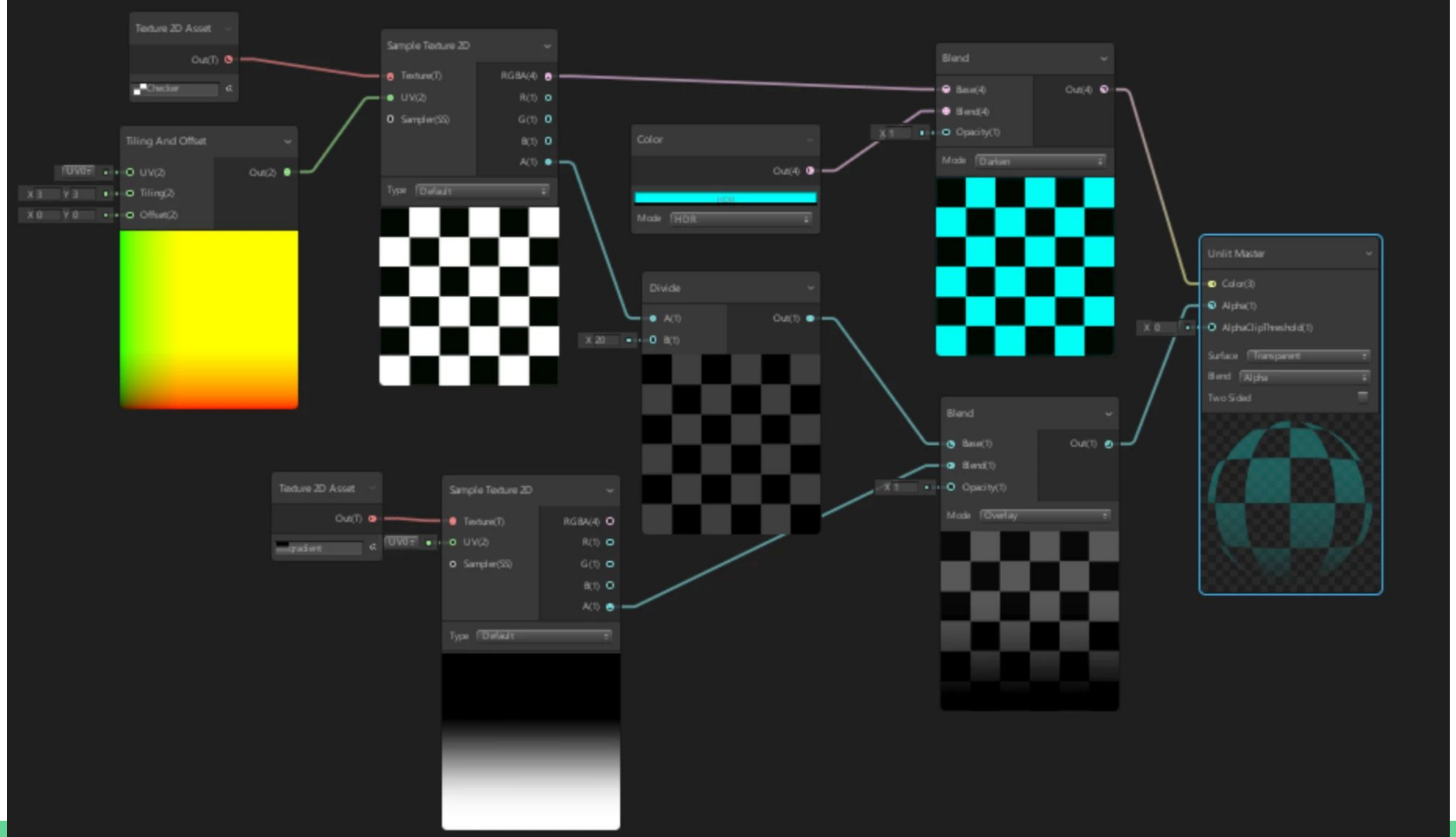
Create a Blend node

- Connect Divide out to first Blend input (Base (1))
- Connect second Sample2D to second Blend input (Blend (1))

And output

Create a color node and another blend node

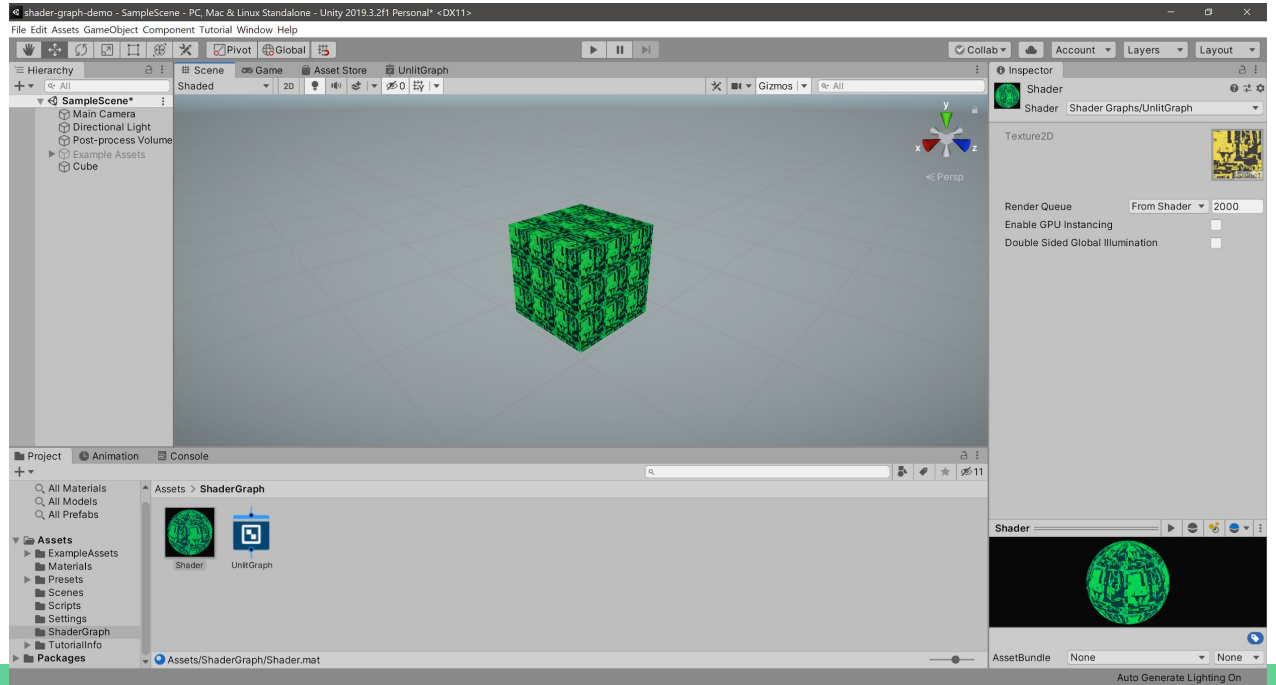
- Set color and change mode to HDR
- Connect RGBA (4) output of first Sample2D to first Base(4) of blend
- Connect Color node Out(4) to blend's second Base(4)
- Set blend mode to Darken
- Connect new blend Out(4) to Unlit Master Color(3) input
- Connect gradient (or whatever) Texture2D Out(4) to Alpha(1) input of Unlit Master
- Unlit Master Surface ➔ Transparent
- Save Asset


Texture 2D Asset
Out(T)
Checker
Sample Texture 2D
Texture(T)
UV(2)
Sampler(SS)
RGBA(4)
R(1)
G(1)
B(1)
A(1)
Type
Default
Tiling And Offset
UV0
X 3
Y 3
X 0
Y 0
UV(2)
Tiling(2)
Offset(2)
Out(2)
Color
Out(4)
Mode
HDR
Blend
Base(4)
Blend(4)
Opacity(1)
Out(4)
X 1
Mode
Darken
Divide
A(1)
B(1)
Out(1)
X 20
Unlit Master
Color(3)
Alpha(1)
AlphaClipThreshold(1)
X 0
Surface
Transparent
Blend
Alpha
Two Sided
Texture 2D Asset
Out(T)
gradient
UV0
Sample Texture 2D
Texture(T)
UV(2)
Sampler(SS)
RGBA(4)
R(1)
G(1)
B(1)
A(1)
Type
Default
Blend
Base(1)
Blend(1)
Opacity(1)
Out(1)
X 1
Mode
Overlay

How to apply?

Create a material and add the Shader Graph to that material

- Then add to your GameObject



Can also use it for vertex displacement!

https://learn.unity.com/tutorial/shader-graph-vertex-displacement#5e0e2223edbc2a167bfbc52f

Vertex displacement time (from prior slide's URL)

Create a new Lit Shader Graph ~~PBR graph~~ (we'll follow the naming scheme this time)

- Name VertexDisplacementSG

Create a new material

- Name VertexDisplacementMaterial

Select the material and add VertexDisplacementSG in the Inspector

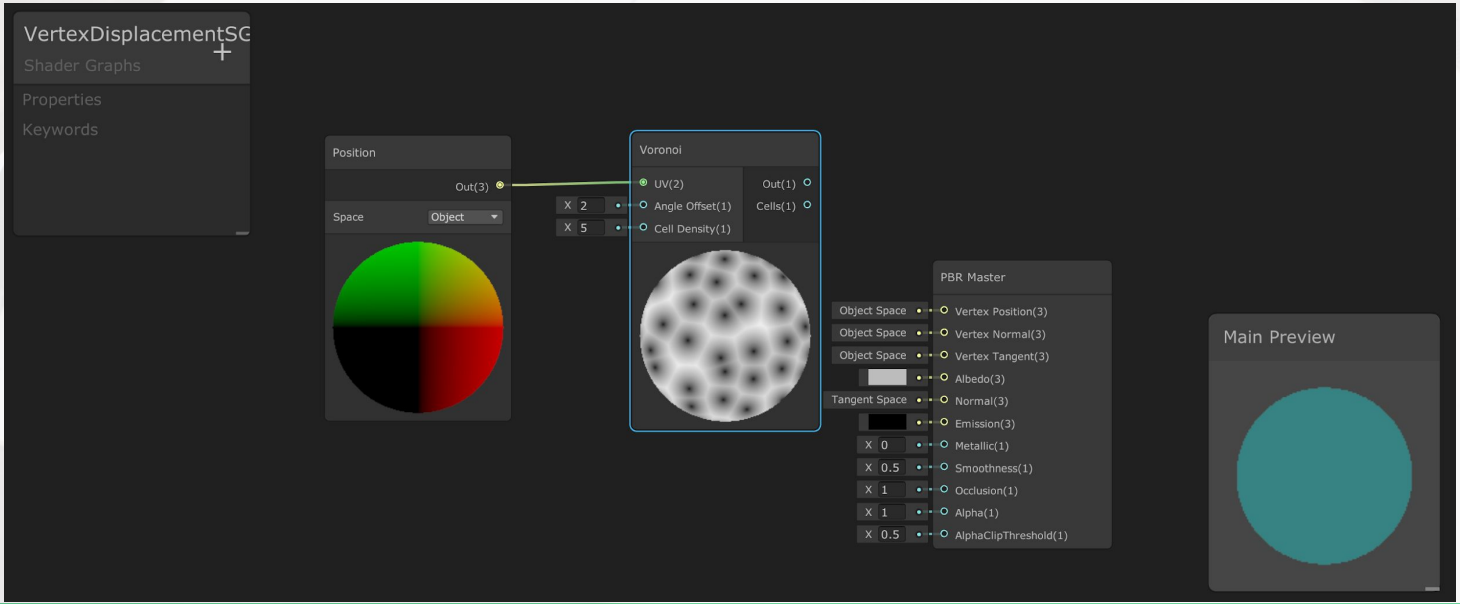
- In the Shader dropdown

Add a sphere and apply the mat to the sphere

Now open up the graph

Add a new position (Input ➔ Geometry ➔ Position // Space ➔ Object)

Then some noise (Procedural ➔ Noise ➔ Voronoi)
➔ Link up the **`Position`** node's output with the **`Noise`** node's UV(2) input

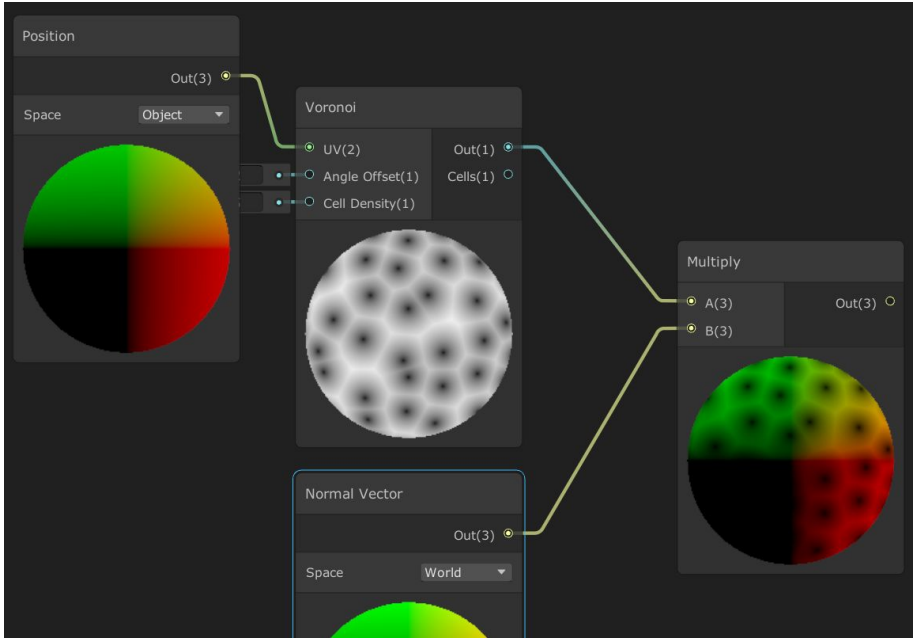


And now...

New nodes:

- Input ➔ Geometry ➔ Normal Vector
- Math ➔ Basic ➔ Multiply

Connect Out of **`Normal vector`** and **`Voronoi Noise`** to inputs (A and B) of **`Multiply`**



And then...

Math → Basic → Add

Connect:

- Outputs of **`Position/Multiply`** nodes into **`Add`**
- Output of **`Add`** to Vertex Position input of ~~**`PBR Master`**~~ **`Vertex`**

Click Save Asset

Object v. World Space

Move the "sphere" around in the Scene view

- Peaks don't seem to change at all!
- "Consistent displacement"

Change the **`Position`** space attribute to Absolute World

- *Don't forget to Save Asset!*
- Exit and move around again

What's happening here is that the world position is adding to itself!

- *Noise pattern depends on position*

Let's update that

Delete connection between **`Voronoi`** and **`Position`**

Add new Position

- Set Space to World
- Connect to UV input of **`Voronoi`**
- (Calculate based on vertex world position)

Set original **`Position`** node to Object

Save and check

- Still "rolls," but Transform gizmo never gets lost!

Let's buff it up (control its strength)

i.e., let's make a slider

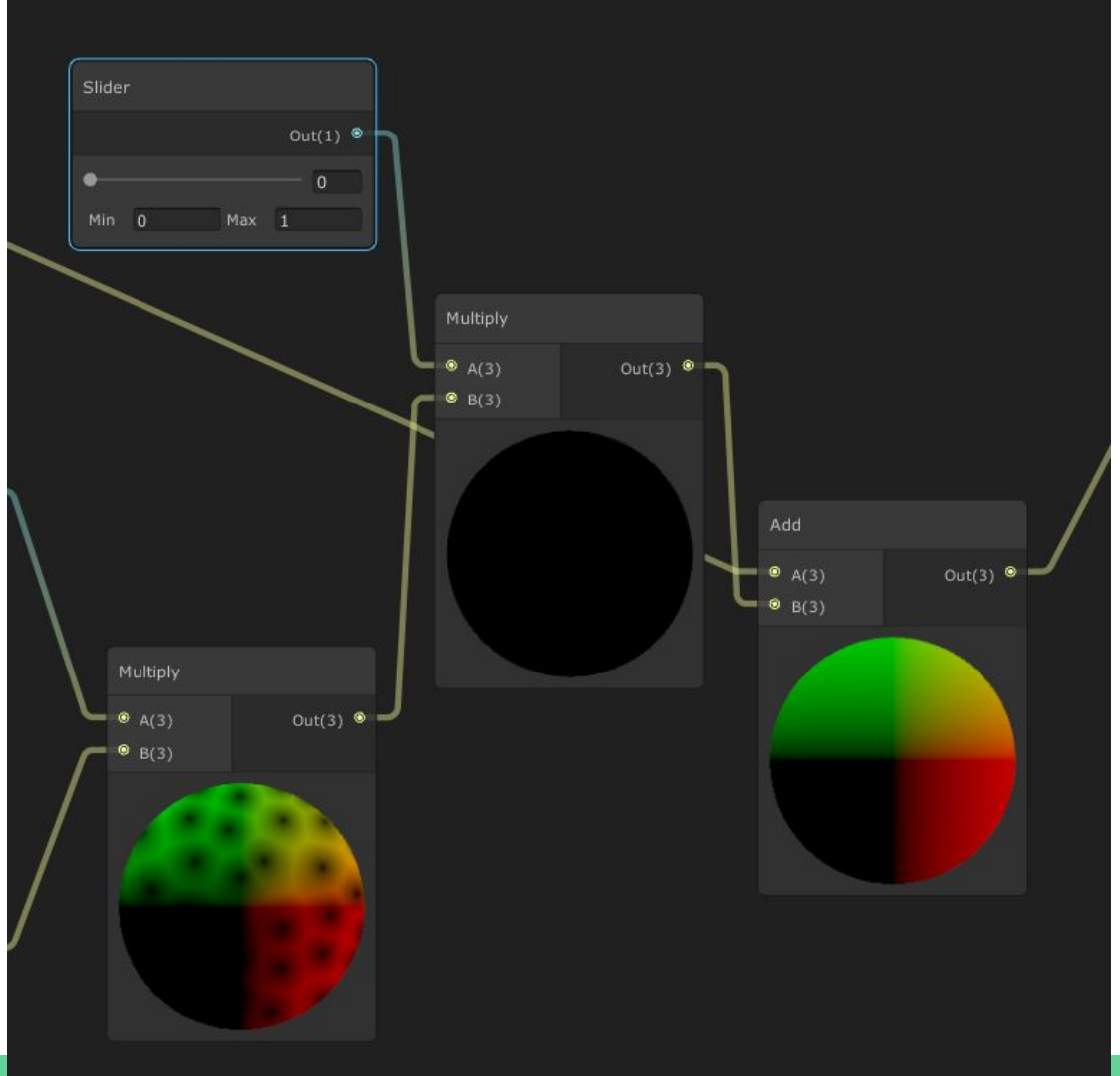
Create:

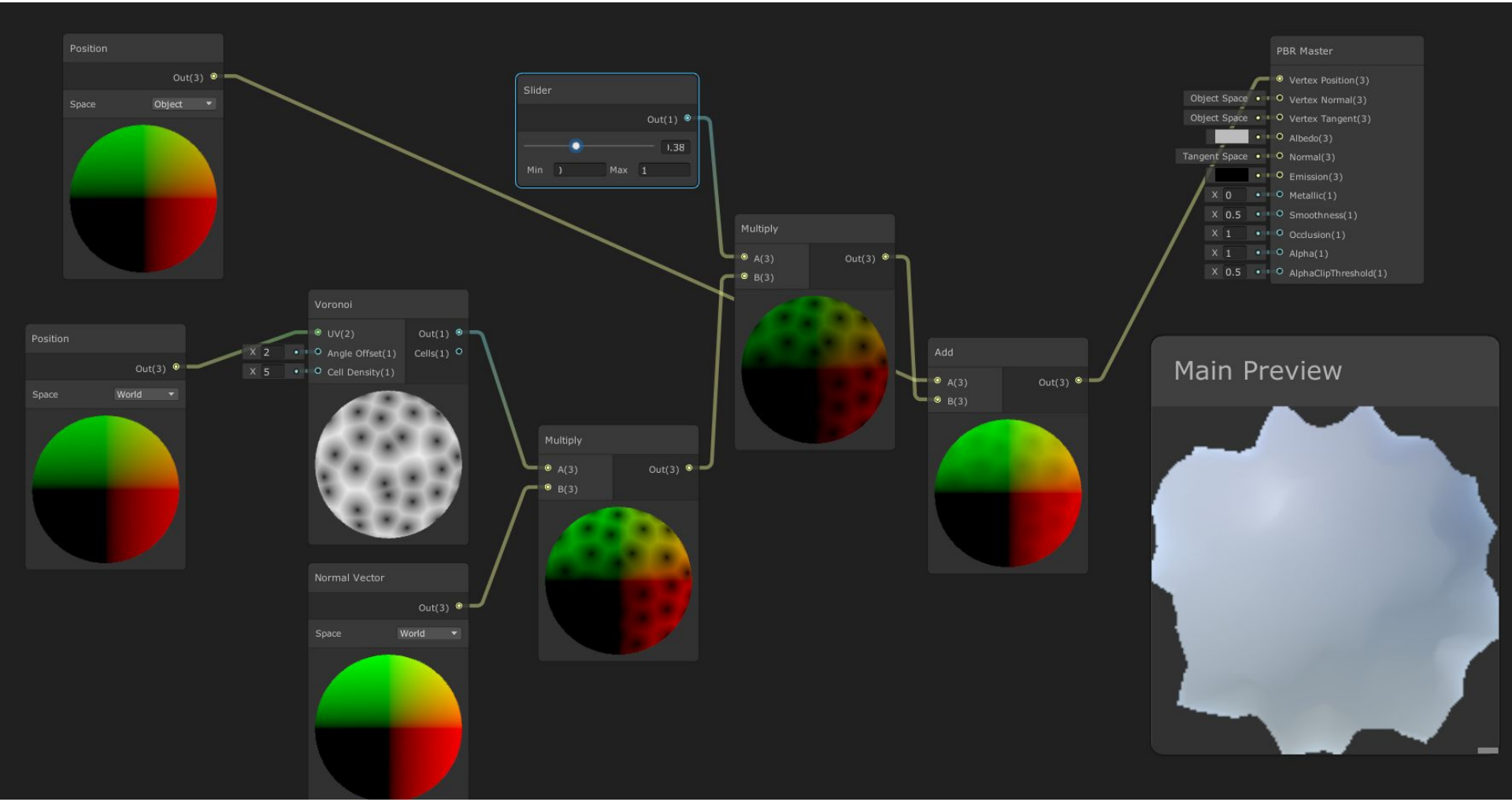
- Input → Basic → Slider
- Math → Basic → Multiply

Connect:

- Output of **`Slider`** to new **`Multiply`**
- Output of *original* **`Multiply`** to new **`Multiply`**
- New **`Multiply`** node to **`Add`** node (replacing old **`Multiply`**)

Save!


Slider
Out(1)
0
Min
0
Max
1
Multiply
A(3)
B(3)
Out(3)
Add
A(3)
B(3)
Out(3)
Multiply
A(3)
B(3)
Out(3)


Position
Out(3)
Space
Object
Slider
Out(1)
1.38
Min
Max
1
PBR Master
Vertex Position(3)
Object Space
Vertex Normal(3)
Object Space
Vertex Tangent(3)
Albedo(3)
Tangent Space
Normal(3)
Emission(3)
X 0
Metallic(1)
X 0.5
Smoothness(1)
X 1
Occlusion(1)
X 1
Alpha(1)
X 0.5
AlphaClipThreshold(1)
Multiply
A(3)
B(3)
Out(3)
Position
Out(3)
Space
World
X 2
X 5
Voronoi
UV(2)
Angle Offset(1)
Cell Density(1)
Out(1)
Cells(1)
Add
A(3)
B(3)
Out(3)
Main Preview
Multiply
A(3)
B(3)
Out(3)
Normal Vector
Out(3)
Space
World

Some fun?

Add a Rigidbody and then a force update:

```
private Rigidbody rb;

void Start() {
  rb = GetComponent<Rigidbody>();
}

void FixedUpdate() {
  rb.AddForce(new Vector3(1.0f, 0.0f, 0.0f));
}
```

Other resources

Older Unity (use at your own risk):
https://www.raywenderlich.com/3744978-shader-graph-in-unity-for-beginners

https://learn.unity.com/tutorial/introduction-to-shader-graph#

2021 and beyond:
https://danielilett.com/2021-05-20-every-shader-graph-node/

https://www.cyanilux.com/tutorials/intro-to-shader-graph/

https://area51.protokoll-studio.com/unity-water-with-shader-graph-tutorial/

https://www.youtube.com/watch?v=vje0x1BNpp8

https://area51.protokoll-studio.com/unity-water-with-shader-graph-tutorial/


THANK
YOU