-ZERIN LABS-

## Kubes

# CubeWorld dungeon KIT

Welcome!
...and thanks for buying this outstanding asset pack :)

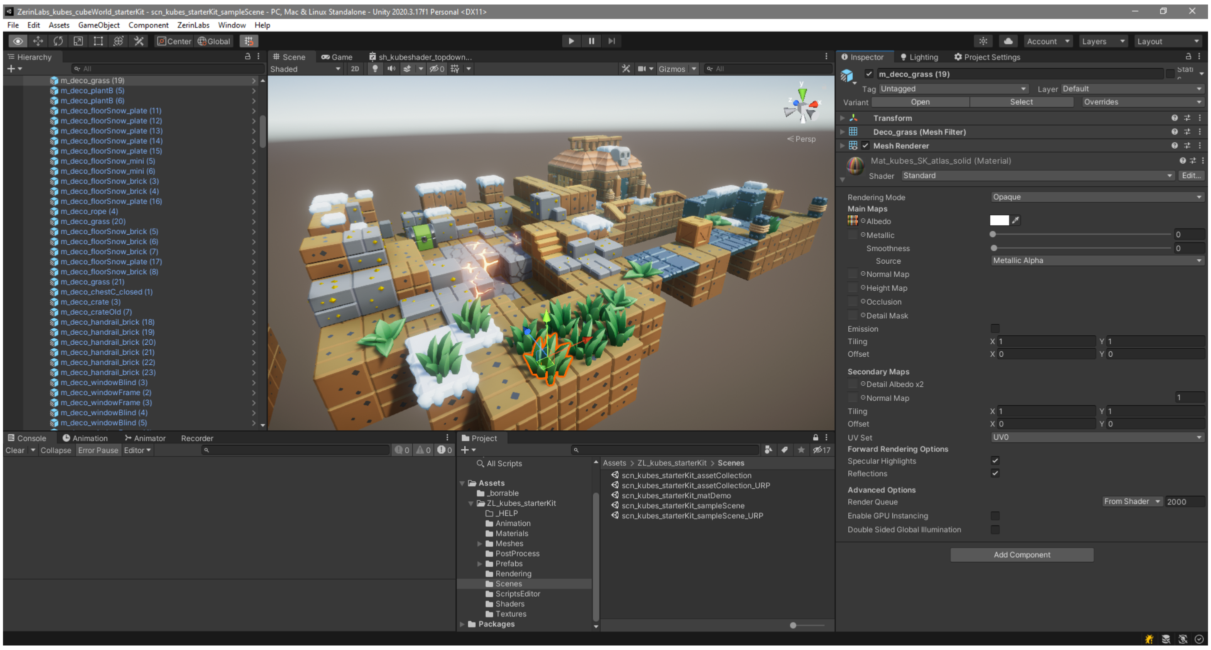
On this small tutorial you will find all the necessary details to understand how to maximise the versatility and capabilities of these assets. Besides we will explain you how to configure your meshes in case you want to use the included bonus scripts “**Object snap utility**” + “**Object transform randomizer**” and how to fine tune the included “**Kubeshader_topdown**” shader.

For any doubt feel free to contact us at: zerinlabs@gmail.com

# THE ASSET COLLECTION

The assets are divided in two major groups:

- **Kube assets** (AKA structural kube-like assets)
    - These are mainly the pieces you'll use to create the base structure of your levels
- **Deco assets** (AKA prop assets)
    - Plants, fences, pots, vessels, etc. and other assets you'll use to decorate your levels

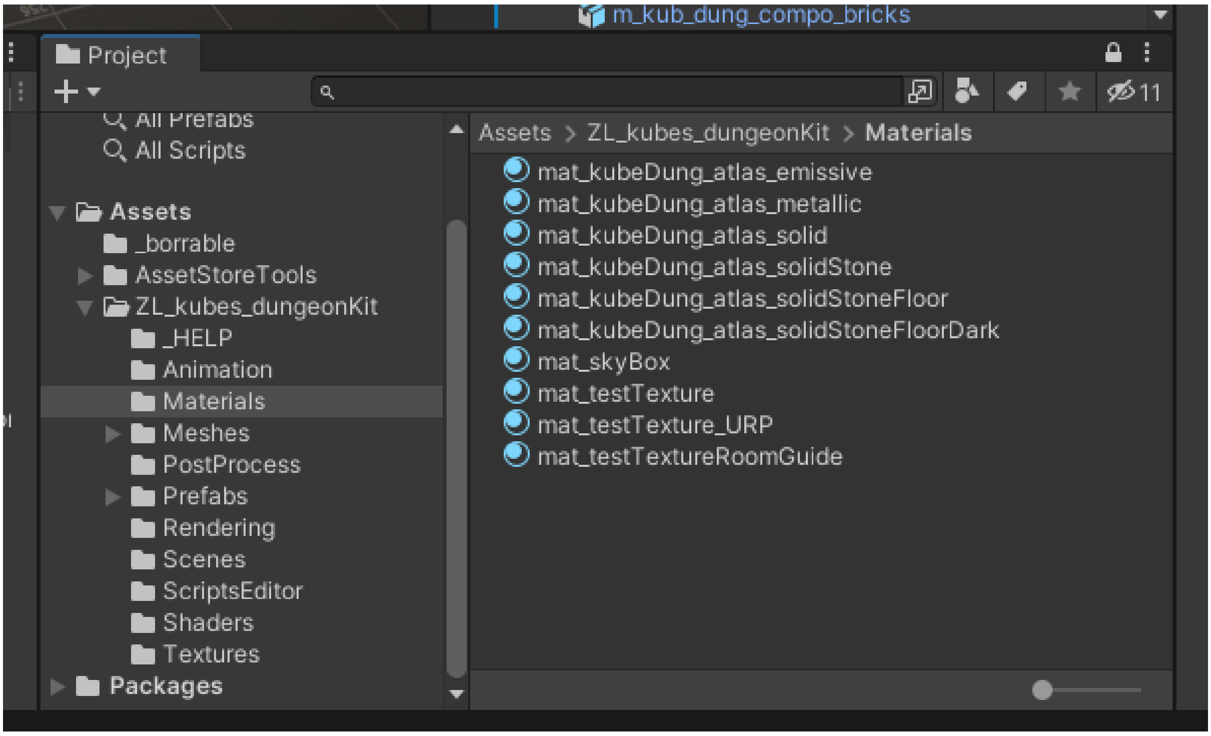

ZerinLabs_kubes_cubeWorld_starterKit - scn_kubes_starterKit_sampleScene - PC, Mac & Linux Standalone - Unity 2020.3.17f1 Personal <DX11>
File Edit Assets GameObject Component ZerinLabs Window Help
Center Global
Account Layers Layout
Hierarchy
m_deco_grass (19)
m_deco_plantB (5)
m_deco_plantB (6)
m_deco_floorSnow_plate (11)
m_deco_floorSnow_plate (12)
m_deco_floorSnow_plate (13)
m_deco_floorSnow_plate (14)
m_deco_floorSnow_plate (15)
m_deco_floorSnow_mini (5)
m_deco_floorSnow_mini (6)
m_deco_floorSnow_brick (3)
m_deco_floorSnow_brick (4)
m_deco_floorSnow_plate (16)
m_deco_rope (4)
m_deco_grass (20)
m_deco_floorSnow_brick (5)
m_deco_floorSnow_brick (6)
m_deco_floorSnow_brick (7)
m_deco_floorSnow_plate (17)
m_deco_floorSnow_brick (8)
m_deco_grass (21)
m_deco_chestC_closed (1)
m_deco_crate (3)
m_deco_crateOld (7)
m_deco_handrail_brick (18)
m_deco_handrail_brick (19)
m_deco_handrail_brick (20)
m_deco_handrail_brick (21)
m_deco_handrail_brick (22)
m_deco_handrail_brick (23)
m_deco_windowBlind (3)
m_deco_windowFrame (2)
m_deco_windowFrame (3)
m_deco_windowBlind (4)
m_deco_windowBlind (5)
Scene Game sh_kubeshader_topdown...
Shaded 2D Gizmos
Persp
Console Animation Animator Recorder
Clear Collapse Error Pause Editor
Project
All Scripts
Assets > ZL_kubes_starterKit > Scenes
Assets
_borrable
ZL_kubes_starterKit
_HELP
Animation
Materials
Meshes
PostProcess
Prefabs
Rendering
Scenes
ScriptsEditor
Shaders
Textures
Packages
scn_kubes_starterKit_assetCollection
scn_kubes_starterKit_assetCollection_URP
scn_kubes_starterKit_matDemo
scn_kubes_starterKit_sampleScene
scn_kubes_starterKit_sampleScene_URP
Inspector Lighting Project Settings
m_deco_grass (19)
Tag Untagged Layer Default
Variant Open Select Overrides
Transform
Deco_grass (Mesh Filter)
Mesh Renderer
Mat_kubes_SK_atlas_solid (Material)
Shader Standard Edit...
Rendering Mode Opaque
Main Maps
Albedo
Metallic 0
Smoothness 0
Source Metallic Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission
Tiling X 1 Y 1
Offset X 0 Y 0
Secondary Maps
Detail Albedo x2
Normal Map 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set UV0
Forward Rendering Options
Specular Highlights
Reflections
Advanced Options
Render Queue From Shader 2000
Enable GPU Instancing
Double Sided Global Illumination
Add Component

# MATERIALS

All the assets have been mapped with the same set of materials, so, editing one of them should should change all the materials of the asset group. Of course you can create and add a new materials if that is more convenient for you.

All materials have their standard and universal render pipeline versions. The URP version of the materials has the suffix _URP on their name

IMPORTANT NOTE: some of the included shaders & materials are both compatible with URP + built in RP, so you should be able to use them in both pipelines without problems



# THE COLLECTION

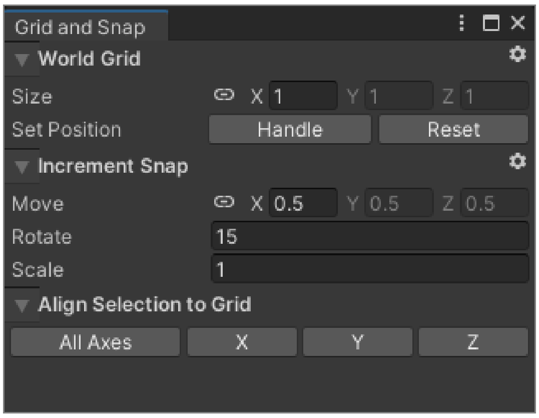
All the “kube set” is organized in two groups: Kubs + Decos.

Kubs are structrural pieces designed to create the “architetonical elements of the levels while Decos are mainly props meant to be used as decorative and detail elements

Each group has a number of modular pieces that can be combined and arranged to create more complex structures.

All the structural kube pieces have been designed to be used in harmony with the default unity snap and the provided script tools.

We recommend to use the unity grid & snap setup with the following configuration:



Grid and Snap

World Grid

Size X 1 Y 1 Z 1

Increment Snap

Move X 0.5 Y 0.5 Z 0.5

Rotate 15

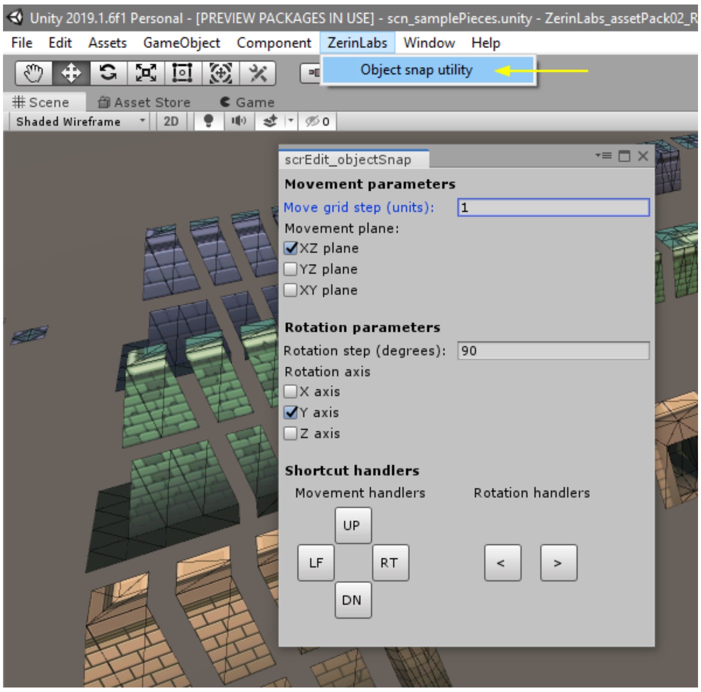
Scale 1

Align Selection to Grid

**PRO TIP**: Remember you can press CTRL while you’re moving or rotating an object to enable this settings

# C# EDITOR SCRIPT : Object Snap Utility

The pack includes a special dedicated editor script (named “**Object snap utility**”) that will assist you positioning and rotating all your assets. (You should be able to find this script on the **ZerinLabs** tray menu)



**Movement parameters**

Move grid step (units): 1

Movement plane:

☑ XZ plane

□ YZ plane

□ XY plane

**Rotation parameters**

Rotation step (degrees): 90

Rotation axis

□ X axis

☑ Y axis

□ Z axis

**Shortcut handlers**

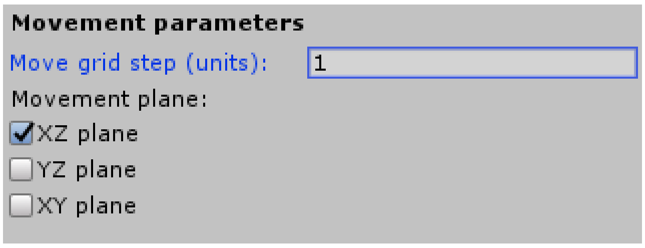
Movement handlers Rotation handlers

The script has 3 different parts:

- **Movement parameters**
- **Rotation parameters**
- **Shortcut handlers**

### Movement parameters

On this group you will be able to customize some parameters used to “position” the assets



**Movement parameters**

Move grid step (units): 1

Movement plane:

☑ XZ plane

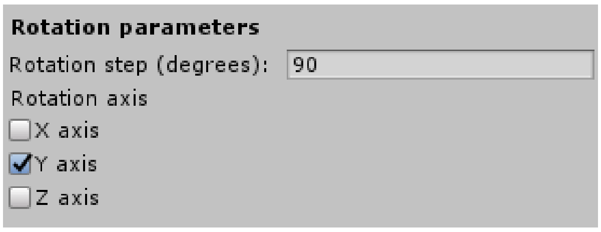
□ YZ plane

□ XY plane

- *Movement grid step:* This is the snamp distance any asset will move each time you use one of the “movement handlers” from the section below.
- *Movement plane:* Here you can select which is the “plane” (defined by 2 axis) where the object will be moved. By default this is the “horizontal plane XZ” (suitable for any top-down game) but you can choose any of the 3 main planes

**Rotation parameters**

On this group you will be able to customize some parameters use to “rotate” the assets



**Rotation parameters**

Rotation step (degrees): 90

Rotation axis

☐ X axis

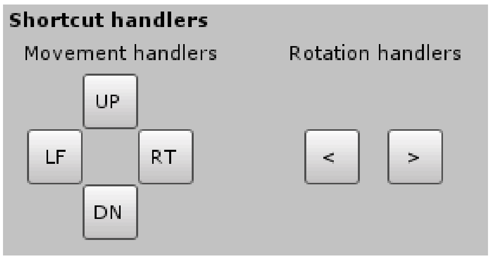
☑ Y axis

☐ Z axis

- *Rotation step*: This is the angle (degrees) that any selected asset will rotate if you use one of the “rotation handlers” from the section below.
- *Rotation axis:* You can select here the axis you want to use for the rotation too. Sameways as the “movement plane”, the rotation axis have been set by default to “Y”. This is the vertical axis (suitable for any top-down game) but you can choose any of the 3 main axis.

**Shortcut handlers**

There are two button/handler groups:



*Movement handlers:*

- Here you have 4 buttons that will act just like a D-pad moving your assets on the “movement plane” defined above one “step” per click.

*Rotation handlers:*

- Here you have 2 buttons that will rotate your asset along the axis defined on the prior section. The “>” button will rotate the object “clockwise” meanwhile the “<” button will rotate the asset “counter-clockwise”.

# C# EDITOR SCRIPT : Object Transform Randomizer

The pack includes another special editor script (named “**Object transform randomizer**”) This script will help you randomize some transform parameters (position, rotation and/or scale) of your assets easily and in a very efficient and complete parametric way. (Same as with the previous script, you should be able to find this script on the **ZerinLabs** tray menu)

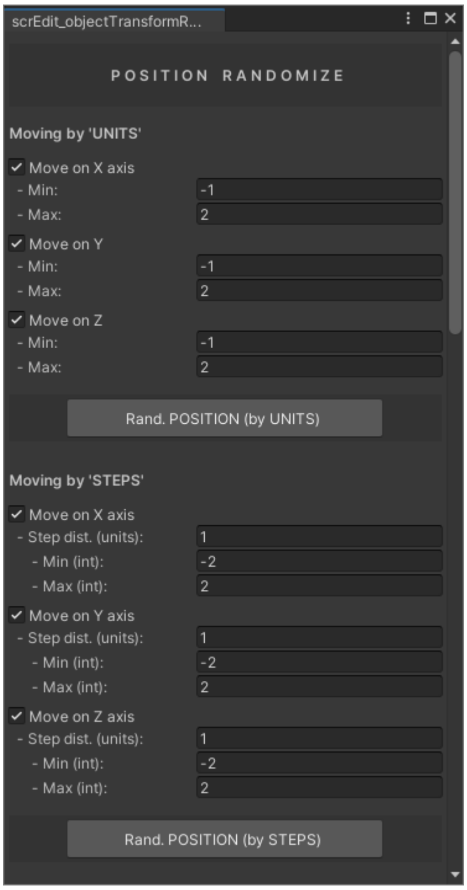
The script is divided in 3 sections

- Position randomize
  - Handles the POSITION randomization of the objects
- Rotation randomize
  - Handles the ROTATION randomization of the objects
- Scale randomize
  - Handles the SCALE randomization of the objects

Each of this sections is divided in two groups of randomization: randomization by units (or angles) and randomization by steps

Generally speaking the randomization by units changes the transform values of an object based on a formula that generates a random number inside a range defined by the user.

This is, for instance, a random scale number between 0.5 and 3 or a random rotation between 0 and 360º degrees


scrEdit_objectTransformR...
POSITION RANDOMIZE
Rand. POSITION (by UNITS)
Rand. POSITION (by STEPS)

Moving by 'UNITS'

☑ Move on X axis
- Min: -1
- Max: 2

☑ Move on Y
- Min: -1
- Max: 2

☑ Move on Z
- Min: -1
- Max: 2

Moving by 'STEPS'

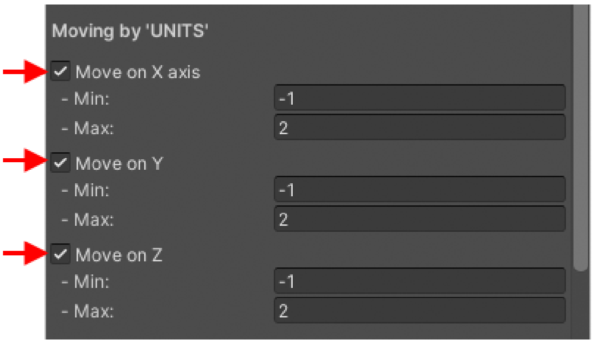
☑ Move on X axis
- Step dist. (units): 1
  - Min (int): -2
  - Max (int): 2

☑ Move on Y axis
- Step dist. (units): 1
  - Min (int): -2
  - Max (int): 2

☑ Move on Z axis
- Step dist. (units): 1
  - Min (int): -2
  - Max (int): 2

On the other side, the “step” method does a random number of “steps” (aka iterations of the variation) with a value defined by the user.

For instance, a 0 to 4 steps rotation with a step of 90º would iterate the rotation from 0 to 4 times with steps of 90 degrees. as a result, the object will be rotated by 0º 90º 180º or 270º All the sections are splitted by axis, in a way that the user can disable or enable the randomization over every axis independently via enabling or disabling the respective checkbox.



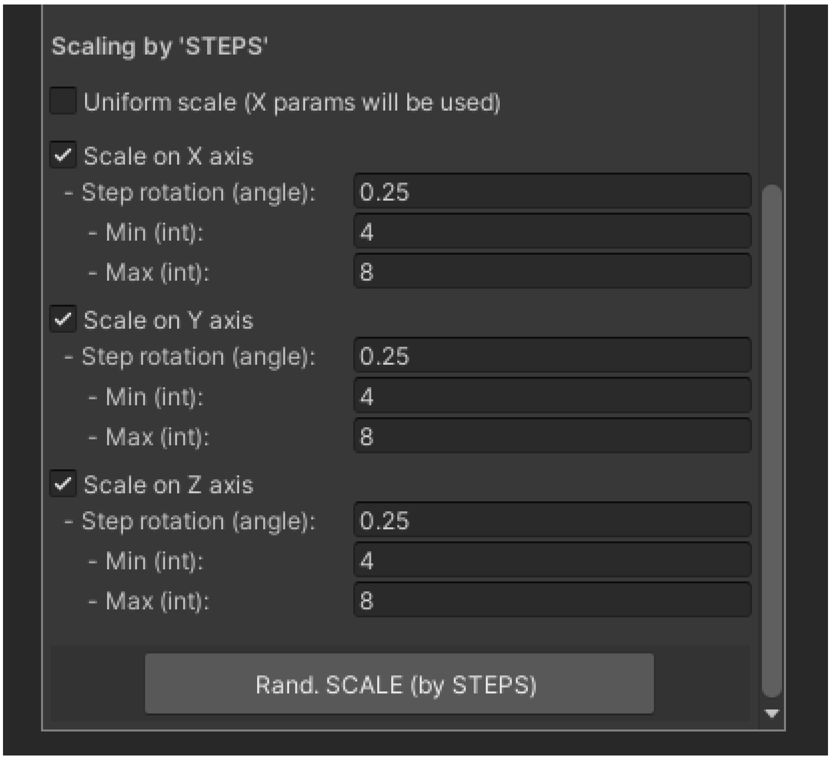
Moving by 'UNITS'

☑ Move on X axis
- Min: -1
- Max: 2

☑ Move on Y
- Min: -1
- Max: 2

☑ Move on Z
- Min: -1
- Max: 2

Last but not least, on the SCALE RANDOMIZE section you have another checkbox called “uniform scale”. Uniform uses the random values from the “X” axis and uses it on the rest of the axis so the object preserves a regular scale (all the axis have the same scale value)



Scaling by 'STEPS'

□ Uniform scale (X params will be used)

☑ Scale on X axis
- Step rotation (angle): 0.25
- Min (int): 4
- Max (int): 8

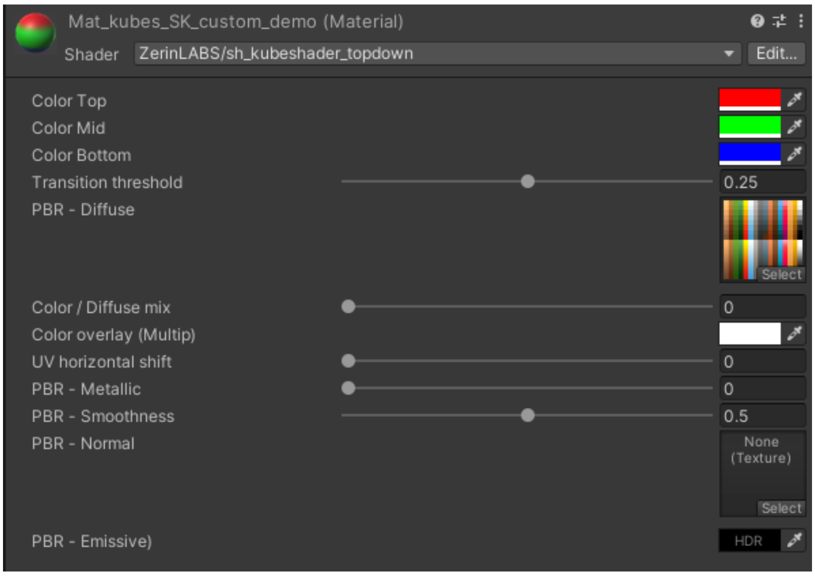
☑ Scale on Y axis
- Step rotation (angle): 0.25
- Min (int): 4
- Max (int): 8

☑ Scale on Z axis
- Step rotation (angle): 0.25
- Min (int): 4
- Max (int): 8

# SHADER : sh_kubeshader_topdown

This pack includes a powerful shader specially designed to maximize the flexibility and versatility of all the pieces included on this kit.

In the list below we will review all the shader functionalities and its possibilities.



Mat_kubes_SK_custom_demo (Material)

Shader ZerinLABS/sh_kubeshader_topdown Edit...

Color Top 🟥

Color Mid 🟩

Color Bottom 🟦

Transition threshold 0.25

PBR - Diffuse Select

Color / Diffuse mix 0

Color overlay (Multip) ⬜

UV horizontal shift 0

PBR - Metallic 0

PBR - Smoothness 0.5

PBR - Normal None (Texture) Select

PBR - Emissive) HDR

## Color Top/Mid/Bottom

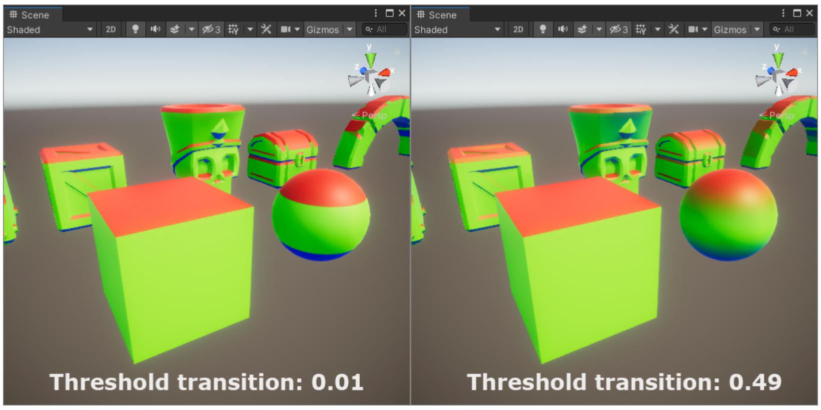
This 3 color pickers define the color of the object based on it’s mesh normals

- Normals pointing up (positive +Y) would be colored with the TOP color
- Normals pointing down (negative -Y) would be colored using the BOTTOM color
- Normals pointing in any direction of the XZ plane would be colored with the MID color

## Transition threshold

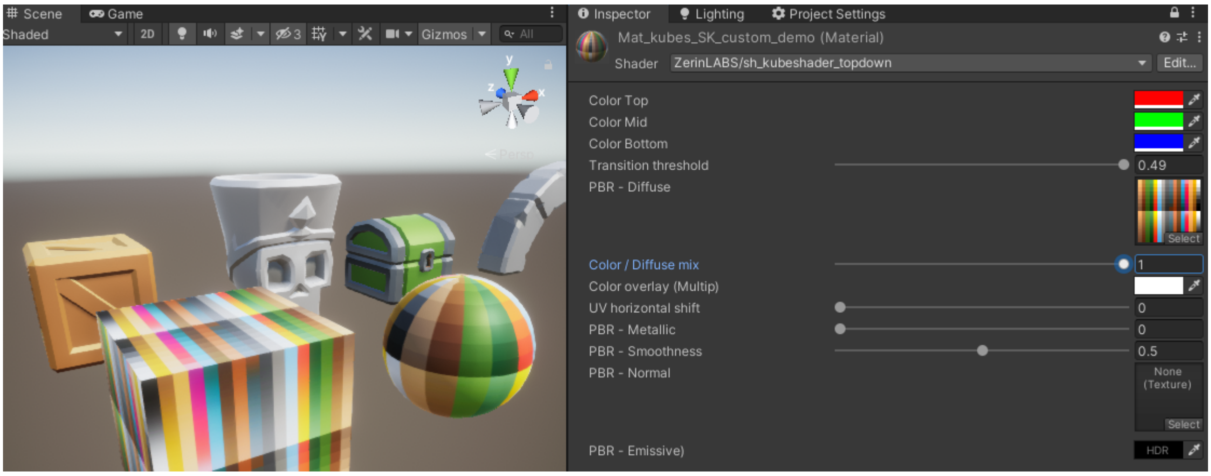
Controls the softness of the transition between the 3 aforementioned colors

Lower values create a harsher transition while higher values create a softer transition between colors



## PBR - Diffuse & Color/Diffuse Mix

The diffuse texture slot sets the diffuse texture used by the material while the “color/diffuse mix” slider mixes the previous 3 colors (Top/Mid/Bottom) with the diffuse texture



Mat_kubes_SK_custom_demo (Material)
Shader ZerinLABS/sh_kubeshader_topdown

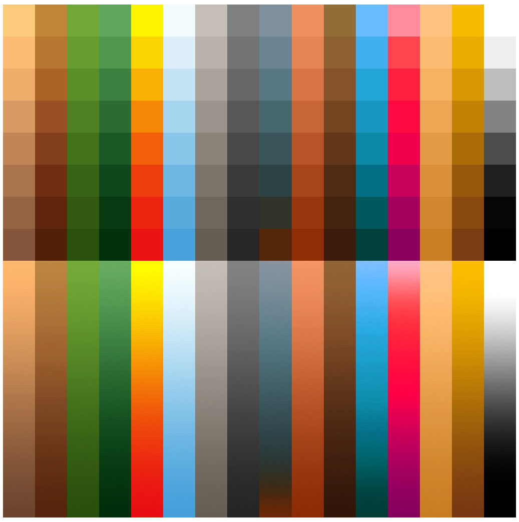
Color Top 🟥
Color Mid 🟩
Color Bottom 🟦
Transition threshold 0.49
PBR - Diffuse
Color / Diffuse mix 1
Color overlay (Multip) ⬜
UV horizontal shift 0
PBR - Metallic 0
PBR - Smoothness 0.5
PBR - Normal None (Texture)
PBR - Emissive) HDR

## Color overlay (Multiply)

This color will be multiplied over the resulting mix of the previous parameter. If you don’t want any color tint you can keep the color in pure white.

## UV horizontal shift

This slider shifts the diffuse texture on its X axis. The cube meshes are UV mapped in a way that are “compatible” with a “Lazy UVs” atlas in a way that offsetting its UV values will force them to use a different color combination of the atlas palette



-the lazy UV texture atlas-

### PBR - Metallic & Smothness

This slider can be used to fine tune the metallic and smoothness parameters of the shader as if it was a regular standard PBR shader

### PBR - Normal

This texture slot allows the inclusion of a normal texture as if it was set on any regular standard material

### PBR - Emissive

HDR color picker that allows to give some emissiveness to the overall material

### Demo video

For more details please check the following tutorial video about the shader:

https://youtu.be/vxHF5gq0nwA

## CONTACT

Feel free to contact us if you have any doubt or question.
And don't forget to follow us on Twitter!! ;)

- Mail
  - zerinlabs@gmail.com
- Twitter
  - @zerinlabs
- Site
  - https://zerinlabs.blogspot.com/
- Youtube channel
  - https://www.youtube.com/channel/UC-u0QyXSJUS60hAfc-UnF-A