# What is VertExmotion

VertExmotion是一个基于着色器的软体系统与程序动画系统相结合。

你可以很容易地动画部分你的网格，如头发，衣服，脂肪…在Unity编辑器!

所有元素都将以程序方式移动，所以不需要为所有东西添加骨骼!

因为它是基于着色器的，所以速度非常快!

因为你没有时间浪费，它超级容易使用!

•添加单个组件。

•画出你想看到移动的东西!

•添加传感器和设置运动属性

•点击播放并享受!

所有部分将跟随网格的运动!

兼容超过80个Unity内置着色器。

很容易包括在您的自定义着色器。

工作与静态网格或蒙皮网格。

测试在PC / MAC Android / iOS / / Webplayer

## How to use it ? (Tutorial)

Sensors settings

您可以将绘画数据导出为一个新的网格。这是非常有用的，因为你可以分享绘画

信息在不同的预制件之间，或保存不同的涂装模板。

作为一个新的网格导出还可以在实例之间共享网格并优化内存。

•输入一个模板名称:“capsule”

•点击“Save as new Mesh”

现在，新的网格参考被保存在另一个预制中。

绘画数据链接到这个资产，所以你不需要再次保存它

尝试改变设置并移动对象。

Parent:传感器变换的父元素(为SkinnedMeshRenderer设置最近的骨骼)。

Layer:传感器层(默认:all)

Distance power:传感器吸引功率(默认为1)

Amplitude multiplier:放大或缩小运动振幅。

Outer max distance :传感器方向上的最大顶点位移

Inner max distance:与传感器方向相反的最大顶点位移。

Inflate:从传感器位置充气顶点。

Damping:增加以稳定运动。

Bouncing:增加以放大弹跳。

T (s):改变曲线时间仅用于可视化



Gravity in/out:重力(Physics.gravity)应用于顶点。

Local offset:传感器空间中的平移偏移量。

World offset:世界空间中的平移偏移量。

Sensor’s link

每个传感器可以连接到列表中的下一个。链接行为通过按下激活/去激活

当两个传感器连接时，胶囊(双球)范围内的所有顶点都将被连接

接收运动数据。根据两者的距离合并变形信息传感器。

传感器顺序可以通过上下按钮进行修改

## How to setup the collision system ?

Collision zone

每个传感点击“碰撞器设置”

•按“添加碰撞器”

•设置物理层蒙版与碰撞。

•添加一个新的碰撞器区域。

•改变位置和半径。

•添加其他碰撞区域以适应网格表面。器都可以通过添加一些碰撞区域与物理进行交互。

如果网格已经有一个物理碰撞器

(SphereCollider BoxCollider…),碰撞区域必须大于对撞机的区域。

Collider settings

碰撞器设置定义当检测到碰撞时如何应用变形。

Layer mask:只有定义层中的物理碰撞器会被激活。

•Smooth collision:平滑的运动系统为更好的动画。

•No backward collisions:根据传感器方向禁用碰撞。

•Maximize collision:使用最大的碰撞区域。禁用此参数

只用半球面来探测碰撞。

•Wobble:添加碰撞后的程序动画。

•Friction:当检测到碰撞时，增加摩擦系数。

•Damping:增加阻尼以稳定运动。

•Bouncing:增加它来放大弹跳。

•Limit:最大弹跳值。

## How to setup the sensor's layers

在一个网格上有3层绘画数据，这提供了一种避免一些传感器重叠在不必要的网格部分。

你可以通过选择dropbox中的图层id来绘制一个单独的图层，或者通过选择' all '来同时绘制所有的图层。

当涂装信息完成后，可以将每个传感器分配到一层，或所有层。

在以下情况下，油漆信息在第三层是不可用的，传感器应该被分配到“1”或“所有”。

## How to import paint data from a map

有时在一个网格上绘画是非常困难的，导入一个纹理绘画数据是一个实时节省。

•创建一个漫反射贴图的副本。

•在你最喜欢的软件中打开它(photoshop, gimp…)

•为软体部分画一个白色面具。

•将图层设置为100% alpha(只有蒙版可见)

•将其保存在unity项目中

•在项目面板中选择图像文件

•在检查面板中设置“纹理类型”为“高级”

•检查“读写启用”

•选择顶点运动对象

•打开油漆面板

•拖放油漆地图

•选择您的UV通道(默认为UV1)

•单击“从地图上油漆”

Normal correction

当顶点移动时，法线可以根据新的人脸方向改变。

为了启用正常校正，你必须将校正参数设置为1，如果有一些视觉伪影，你可以降低值来修复它们。

平滑参数-根据与传感器位置的距离定义平滑法线的阈值

How to share sensors between mesh ?

如果你想在不同的网格之间共享一个传感器，你可以添加一个现有的传感器，而不是创建一个新的。

Vertexmotion组件将共享传感器设置。

这是有用的，如果你想同步身体变形的衣服。

How to include VertExmotion in my custom shader ?

首先，复制你的着色器在另一个文件。

修改着色器的名称为“VertExmotion/shadername”以保证编辑器的兼容性。

•对于表面着色器，你必须在你的着色器中修改这些:

#pragma surface surf Lambert alpha vertex:vert addshadow

#include“Assets/VertExmotion/Shaders/VertExmotion.cginc”

void vert(inout appdata\_full v){VertExmotion(v);}

•如果你的着色器已经有一个顶点函数，添加这些行:

#include "Assets/VertExmotion/Shaders/VertExmotion.cginc"

void vert (inout appdata\_full v) {VertExmotion(v);

•如果顶点函数不使用appdata\_full，添加以下行:

#include "Assets/VertExmotion/Shaders/VertExmotion.cginc"

void vert (inout appdata v) {v.vertex = VertExmotion(v.vertex, v.color);}

How to include VertExmotion in Unity Shader Graph

VertExmotion包包括内置的Unity Shader Graph节点。

解压文件'VertExmotion/Addon/VertExmotion\_ShaderGraph\_X\_Nodes'。

ShaderGraph 3。

-添加节点'VertExmotion'和'VertExmotion Init'

ShaderGraph 4。

拖放VertExmotion Subraph (VertExmotion/Editor/ShaderGraph/)

How to include VertExmotion in Amplify Shader Editor

VertExmotion包包含用于ASE的内置节点。

-解压文件'VertExmotion/Addon/VertExmotion\_AmplifyShaderEditorNodes'

在ASE窗口中打开你的着色器。

-在通用部分设置顶点输出为“相对”。

-在“附加指令”部分添加VertExmotion包含文件

-新的节点类别“VertExmotion”可用。

-拖放VertExmotion节点。

-连接到“局部顶点偏移”输入

如果你正在使用镶嵌或其他节点来修改“局部顶点偏移”，你可以使用advanced的版本。

LWRP和HDRP的正态校正节点要求世界正态和世界正切。

How to use the HDRP/Lit shader

这个包在“Addon”文件夹中包含一个兼容版本的“HDRP/Lit”着色器。

选择与你的Unity版本兼容的最新版本并解压包。

现在你可以使用“修复材质”按钮更新材质的着色器，或者手动修改材质(VertExmotion/HDRP/Lit)。

材质需要一个特定的位移模式:

-选择材质

-在检查器中选择“表面选项”面板

-设置位移模式为“顶点位移”

How to include VertExmotion in Shader forge

在shaderForge编辑器中打开着色器。

-在shader设置，设置路径为'VertExmotion/ShaderName'(例如:'VertExmotion/ShaderForge/test1')

-添加一个新的CG包括线:'Assets/VertExmotion/Shaders/VertExmotion.cginc'

-添加一个代码节点'VertExmotion' -复制代码:

-输出设置为“Float3”添加一个输入“Float3”:xyz

-添加一个输入“浮动”:w

-添加一个输入“Float3”:c

-添加世界pos节点插入某某输出xyz

-塞w功率w

-添加顶点颜色节点插入RGB输出c

-塞VertExmotion节点的顶点偏移量

How to include VertExmotion to Alloy

Alloy 1.3.6包含了对VertExmotion的官方支持。

-打开Alloy/Shaders/Config文件。

cginc'

-取消注释:

#define A\_USE\_VERTEX\_MOTION

#include "Assets/VertExmotion/Shaders/VertExmotion. "

重新导入“Alloy/Shaders”文件夹

How to convert a complex shader

一些着色器比其他的更复杂，转换它们的方法总是相同的:

-创建一个着色器文件的副本

-添加路径'VertExmotion/'的名称

-找到顶点程序或创建它，如果没有找到。

顶点程序可以在表面着色器中找到，通过查看

#pragma surface…vert:VertExFunctionName…

如果着色器是一个顶点/片段程序，看看这条线

#pragma vertex VertExFunctionName

在复杂的着色器中，顶点函数可以包含在 a.cginc文件中，

在这种情况下，你必须:

-复制cginc文件(例如:shaderCore.cginc - > shadercore-VM.cginc)

-添加VertExmotion.Cginc在文件的顶部

-在所有顶点程序中包含VertExmotion函数

-在着色器中重命名包含文件('#include " shaderCore-VM').cginc”)

-添加一行“#include”Assets/VertExmotion/Shaders/VertExmotion。

在顶点函数的顶部或者在cginc文件的顶部。

-插入顶点程序的顶点运动函数。

对着色器的所有顶点程序都这样做。

Tutorials for complex shaders

---Preintegrated Skin Shader 预先集成的皮肤材质

找到在着色器中使用的vertex程序，

看这一行:

# pragma vertex pss\_vert

…

# include“./ PreIntegratedSkinShaderCore.cginc”

---顶点程序位于“PreIntegratesSkinShaderCore.cginc”(pss\_vert)

复制文件到“PreIntegratesSkinShaderCore-VM”，并添加了vertExmotion函数

PSS\_V2F pss\_vert(PSS\_VIN v) {

PSS\_V2F o;

VertExmotion (v);

UNITY\_INITIALIZE\_OUTPUT(PSS\_V2F,o);

---在cginc文件的顶部添加include文件。

# include“Assets/ VertExmotion /着色器/ VertExmotion.cginc”

注意:如果你想移动VertExmotion文件夹位置，你可以在着色器中复制'VertExmotion.Cginc’

到'#include "VertExmotion.cginc"'

---复制shader文件到PreIntegratedSkinShaderStandard-VM.shader 和 添加“VertExmotion/”在名字的开头

---改变“PreIntegratedSkinShaderStandard.shader”的输入结构，

VertExmotion函数需要'appdata\_full'结构

/ / # define PSS\_VIN appdata\_tan

#定义PSS\_VIN appdata\_full

---更改所有的#include "./PreIntegratedSkinShaderCore.cginc"

--->“# include“./ PreIntegratedSkinShaderCore-VM.cginc””

---在shader文件的末尾修改阴影通道

使用VertexMotion通道来启用阴影变形

/ /传递的影子。

//SubShader {uspass "VertexLit/SHADOWCASTER"}

SubShader{uspass "VertExmotion/Standard/SHADOWCASTER"}

How to use the VertExmotion API

您可以访问自定义脚本中的所有VertExmotion设置。您将能够为您的特定需求创建自定义FX。

为了在自定义脚本中使用VertExmotion，你必须添加名称空间:'Kalagaan'。

这里有一个例子:

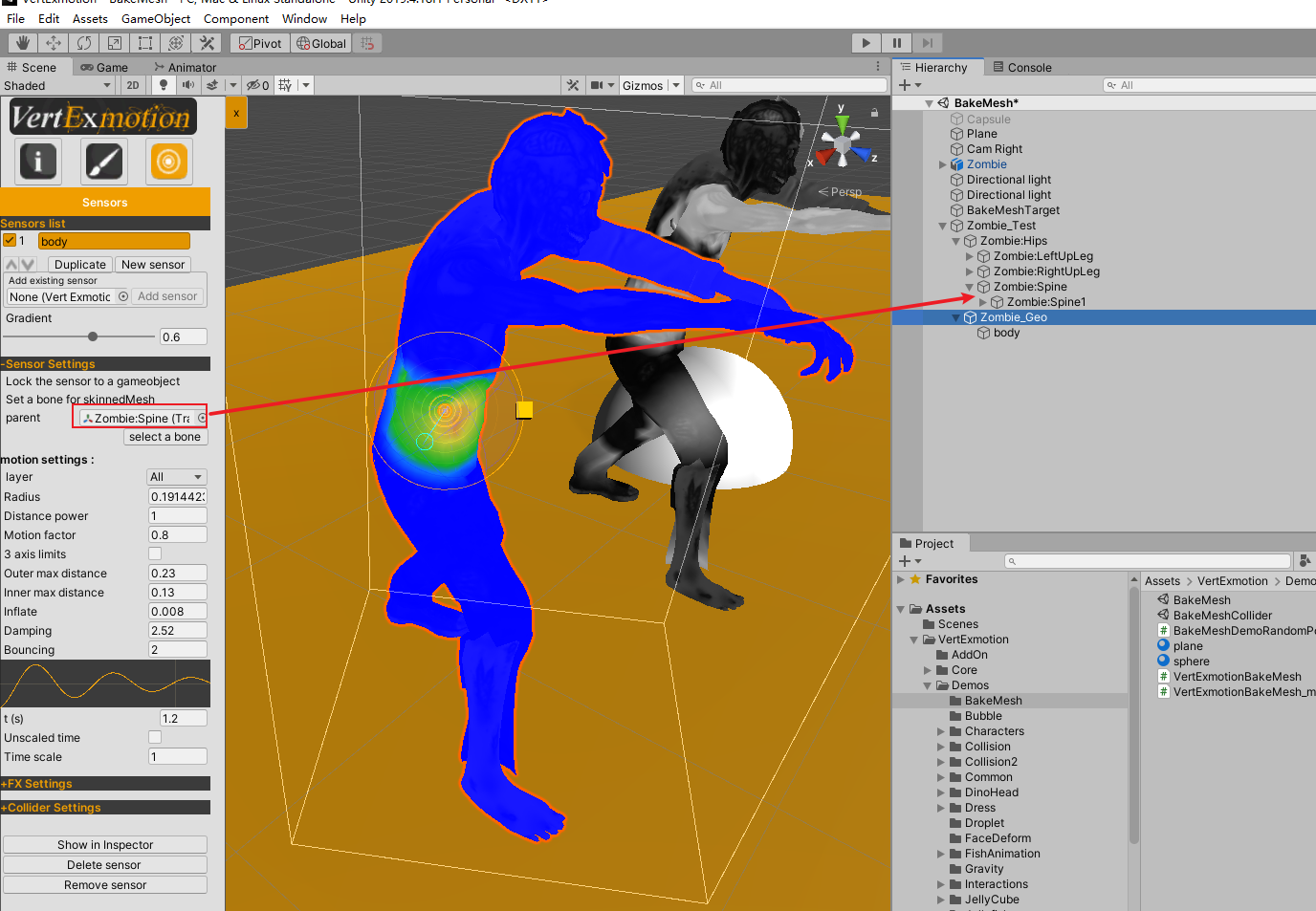
看看包中包含的演示，其中一些使用自定义脚本。

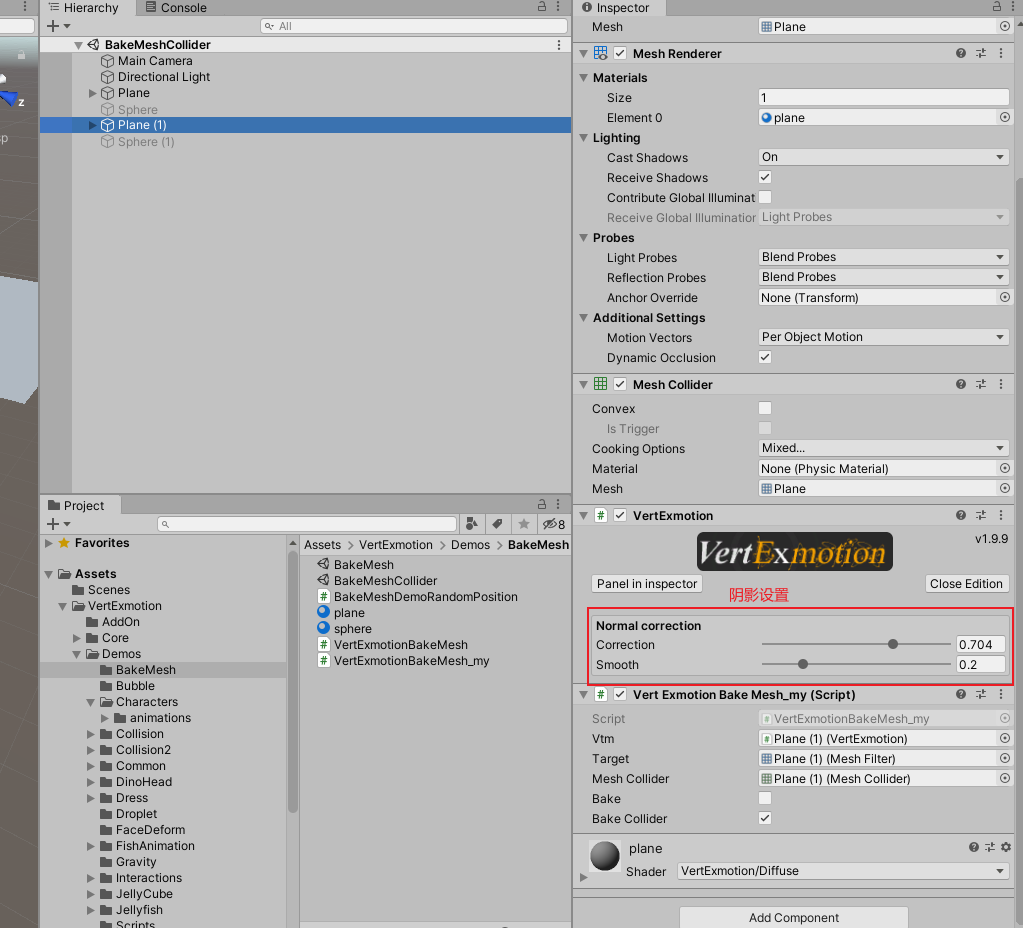
完整的API可以在这里获得:www.kalagaan.com/VertExmotion/API

Third party compatible assets

Demo:

BakeMesh Mesh烘焙 可以进行多实例 以及Collider应用 强制改变网格顶点 基础操作





Bubble 透明球 动态生成多个VertExmotion 并改变网格顶点

Characters 多人物 展示用

Collision 基本碰撞

Collision2 进阶碰撞

DinoHead 弯腰龙 多研究

Dress 裙子 多研究

Droplet 下落水滴 多研究

FaceDeform 脸部变形

FishAnimation 鱼动画

JellyCube Q弹方块

Jellyfish 水母

SplashDemo 一滩死水

Sprite2D 2D图片

TextMesh 文本Mesh变化

TireDemo 轮胎