VRM文件大小

由于 VRM 是基于 GLB 的格式，

`导出的VRM`文件大小 => `glb` 的大小

`glb` => `json + binary`

json 是文本格式，通常不会超过1MB。

其中，二进制文件中主要包括：

`image` 和 `mesh`。

以一个包含5万个顶点和5万个三角形的模型为例，我们来计算其大致容量。

Image

Image `Texture(纹理)`、缩略图（材质中，VRM元数据中，用于提供预览的图片）。这些图像以 PNG（JPG）的形式存储。

> 在 v0.56 中，模型中的大纹理（例如 4096x4096）无法导出为较小的纹理（例如通过纹理导入器设置将其设置为 1024x1024 -> MaxSize）。我们已在 v0.5

https://github.com/vrm-c/UniVRM/issues/502

Mesh

包括索引缓冲区（Index Buffer）和顶点缓冲区（Vertex Buffer）。

索引缓冲区（Index Buffer）

索引缓冲区使用整数（Int）数组。

对于一个含有5万个三角形的模型，

`50000 x 4（Int = 4字节） x 3（三角形的3个顶点）` => 使用0.6MB 的容量。

> 虽然GLTF规范中也支持无符号短整数（unsigned short），但UniVRM的导出器目前不支持。由于最多只能存储65536个顶点，因此无法存储50000个三角形。

顶点缓冲区（Vertex Buffer）

一个顶点包括以下内容：

```
{
    float3 Position; // 頂点位置 4(floatサイズ) x 3(xyz) => 12byte
    float3 Normal; // 頂点法線 4(floatサイズ) x 3(xyz) => 12byte
    float2 TEXCOORD_0; // 頂点UV 4(floatサイズ) x 2(xy) => 8byte
    short4 JOINTS_0; // 頂点BoneIndex 2(shortサイズ) x 4(4boneまで) => 8byte
    float4 WEIGHTS_0; // 頂点Weight 4(floatサイズ) x 4(4boneまで) => 16byte
}
```

> 某些模型可能包含顶点颜色或辅助 UV（不支持），因此所需大小可能会有所变化。

> 在 UniVRM 中，切线（float4）可以在 Unity 中计算，而不是存储在 GLTF 中。通过给定顶点法线和 UV，可以使用 MIKK T Space 算法获取切线。

对于一个含有5万个顶点的模型

`50000 x (12 + 12 + 8 + 8 + 16) => 2.8MB` 的使用容量。

基本大小

[`总图像大小` + [illegible] 50k 个顶 [illegible] [`3.4MB +` [illegible] BlendShape 的大小计算，它在某些情况下可能导致总大小爆炸。

BlendShape（MorphTarget）的容量

```
// ブレンドシェイプ頂点
{
    float3 Position; // 頂点位置 4 x 3 => 12byte. 必須
    float3 Normal; // 頂点法線 4 x 3 => 12byte. オプション
    float3 Tangent; // 頂点Tangent 4 x 3 => 12byte. 記録しない
}
```

每个BlendShap`50000 x (12 + 12) => 1.2MB`。

假设有20个BlendSh`50000 x (12 + 12) x 20 => 24MB`。

假设有60个BlendSh`50000 x (12 + 12) x 60 => 72MB`。

注意，在以下情况

- 当存在大量 BlendShape
- BlendShape 所在的位置与非 BlendShape 所在的位置未分割时

容量会变得很大。接下来是关于减少BlendShape容量的方法。

> 译者补充：
>
> 曾经看过小K直播姬官方提供的一个规范文档，里面就有提到过要把面部网格单独分开（不予其他网格合并在一起），虽然不分开也能使用，且不会有任何问题，但依然为了省资源做了规范。

减少**BlendShape**容量的方法

在导出对话框中，有几个与BlendShape大小优化相关的选项。

导出选项

为了减小 BlendSha`ReduceBlen dshape` [`ReduceBlendshapeClip` 是最安全的方法（nc`UseSparseAccessor` 来解决一些 VRM loaders的导入问题（UniVRM loader 没有问题）。如`OnlyBlendshapePosition` 本会导致导入错误。

ReduceBlendshape（减少BlendeShape）

未被BlendShapeClips引用的BlendShapes将不会被导出。可以减小文件大小。

ReduceBlendshapeClip（减少BlendeShapeClip）

属于Preset.Unknown的BlendShapeClip将不会被导出。与ReduceBlendshape结合使用。

UseSparseAccessor（使用稀疏访问器）

在GLTF中使用稀疏访问器功能：仅记录具有非零值的BlendShape顶点。

如果模型包含多个BlendShape，启用此选项可以帮助减小文件大小。

> 正在修复中：存在GLTF的兼容性问题，会导致非UniVRM的加载器出现错误。

```
// ブレンドシェイプ頂点
{
    int Index; // 有効なブレンドシェイプの index => 4
    float3 Position; // 頂点位置 4 x 3 => 12byte. 必須
    float3 Normal; // 頂点法線 4 x 3 => 12byte. オプション
    float3 Tangent; // 頂点Tangent 4 x 3 => 12byte. 記録しない
}
```

`BlendShape的有效顶点数 x (12 + 12 + 4) =>` ?MB

- 只是保存方式有所变化

OnlyBlendshapePosition（仅导出BlendeShape Position）

不导出 BlendShape 的 Normal 和 Tangent。可减小文件大小。

> 请注意，UniVRM-0.53 之前的版本在导入时会出现错误。

MESHUTILITY：拆分带/不带BlendShape的网格

例如，一个模型的网格包含50k个顶点。其中有10k个（带有BlendShape）在面部，40k个（不带BlendShape）在身体上。

分割后设置一个Blend `10000 x (12 + 12) => 0.24MB`

分割前设置一个Blend `50000 x (12 + 12) => 1.2MB`

> 运行时性能也将受益于这种网格分割。但是由于渲染网格的数量增加，Draw调用可能会增加。

Mesh Utility（面向开发，无翻译）

总结

如果你的 VRM 文件体积过大，首先请检查 BlendShape，其次是纹理图像和预览图。

Revision #2
Created 27 July 2024 02:17:13 by 北溟瞬息
Updated 29 August 2024 08:41:05 by 北溟瞬息