

自己紹介

Voyage Voyage

VRCHAT民 since 2020/10

プレイ時間：3000時間以上

VRCHAT向けの自作

- Quick Avatar Setup (Godot Engine + Unity Editor - 自分用)
 - Blendshapes Animationsで表情をアバターに付けるツール
- World Lock Autosetup (Unity Editor - リリース)
 - アバターにワールドで固定できるアイテムを装備するツール

その他、色々 . . .

(画像)

このプレゼンテーションの流れ

1. つい最近リリースしたBlenderのAddonの作る理由と作る苦労を語る。
2. 出来た自分用ツールから、Boothでリリース出来るちゃんとしている皆が使えるツールするまでの道を語る。
そして、なぜ色んな人がその道を通らないのことも。

Part 1

今回の話題

自作Blenderアドオン：

- My bones merger (Blender - リリース)
複数のボーンとそのウェイトポイントを結合ためのBlenderアドオン

作ろうと思った理由

BlenderでVRCHAT用モデルを最適化するために、そのボーンを簡単に減らすツールを作りました。

「Excellent」と言うVRCHATのアバターランキングを狙いたいなら、モデルのボーン数を85以下にしないとだめです。

問題は、色んなモデルはPhysics Bonesのため、100ボーンを超えていますから、困りますね。

ボーンの仕組み

どうして動かないの？

ボーンを動かすと、関連している頂点が割り振れたウェイトによって動きますね。

Blenderでは無暗にボーンを消すと、モデルがグニャグニャになりますね。

理由は削除したボーンに対応する頂点がもう動きません。

Blenderでは、ボーン・ウェイト・頂点の関係はこんな感じで決めることができます：
(ほかの方法が存在します)

(Armature) ボーン <- (Mesh) Armature Modifier, バインド先: 頂点グループ -> 同名
頂点グループ <- ウェイト

つまり、BodyのArmature Modifierが、Armatureの各ボーンをMeshの同名頂点グループに繋がります。

例えば、Headのボーンは
Headの頂点グループに繋がって、
そのグループの頂点に割り振ったウェイトを使います。

ですから、ボーンを減らすためには、
まずは頂点グループ合成みたいな機能を探しましたね。

そして、ぴったりのModifierがありますね：
「頂点ウェイト合成」 Modifier。

お陰で、消したボーンの名頂点グループの頂点情報を
他のボーンに移動することが出来ますね。

使い方はこんな感じです：

- あるボーンAを消します
- Meshに「頂点ウェイト合成」を付けます
- Aの頂点グループを決めます
- 残っているボーンBの同名頂点グループを決めます
- OperationをAddにします
- Modifierを適用します

お陰で、Aの同名頂点グループの頂点情報を、Bのボーンの名義頂点グループに追加して、
グニャグニャの問題を解決できます！

とはいえ、すで他の問題が現れますね：

120ボーンモデルを85以下にしたいなら、その手順を35回をくりかえすことが必要ですね。

はい。

他の方法を見つかりませんでしたから、一回やったんですが。
二度とやりたくなかったから、自動化スクリプトを作っていました。

Blenderでは「Scripting」タブがあります。ですから、そこでスクリプトを開発して始めました。

自動化スクリプトを作って初めて、目標をこんな感じに決めました：

- [] 消すボーンを決める
- [] ウェイト情報を受けるターゲットボーンを決める
- [] それぞれの頂点グループを見つける
- [] 消しているボーンの頂点グループの頂点をターゲット・ボーンの頂点グループに追加する

そして、一番の相当難しかった所は：

消しているボーンの頂点グループの頂点をターゲット・ボーンの頂点グループに追加する

つまりリストAをリストBに積算する
ってことです。

原因はBlenderの頂点グループのAPIです：

VertexGroup

メソッド

- `add(index, weight, type)` : グループに頂点を追加できる。
- `remove(index)` : グループから頂点を消すことができる。
- `weight(index)` : グループに含んでいる頂点のウェイト情報を確認できる。

おわかりいただけたでしょうか？

グループに追加されている頂点のリストを受け取るメソッドがありません！

結構狂いましたね。

そして、Web検索の結果によると、グループが含んでいる頂点のリストを簡単に取れません。

むしろ、そのリストを自分で作らないといけません。

方法は：

- 各頂点をスキャンして、
- その頂点の「groups」の情報を使って、狙っているグループに含んでいるのかどうか確認できます

「まじかよ」って感じになって、目標に一つのステップを追加していました：

- [x] 消すボーンを決める
- [x] ウェイトペイントを受けるターゲットボーンを決める
- [] 頂点グループの頂点リストを簡単に取れるようにする
- [x] それぞれの頂点グループを見つける
- [] 消しているボーンの頂点グループの頂点をターゲット・ボーンの頂点グループに追加する

「頂点グループの頂点リストを簡単に取れるようにする」のために、選んだやり方はこうでしたね：

- リストを作って
- そのリストのサイズを頂点グループの数に決定して
- 各添え字で、Setを用意して
- 各頂点をスキャンして、見つかったグループの添え字つつ、その頂点を関連するSetに追加します。

そして、今回は一番難しかったのは：

- そのリストのサイズを頂点グループの数に決定して
- 各添え字で、Setを用意して

理由はPython語です！

Python語では、配列に書く時でも、使う添え字がリストのサイズを超えたらExceptionがでますね。

```
>>> a = []  
>>> a[0] = 1  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
IndexError: list assignment index out of range
```

問題はPythonでは、`new Set[max_vertex_groups]` みたいなシンタックスがないんですから、

Webで、どうやって配列のサイズを伸ばせるお勧めの方法を探しました。

そして、見つかったのはこちらです：

```
cached_groups = [set()] * max_vertex_groups
```


立派な罠です。あの方法を使うと
すべてのインデックスのSetオブジェクトのリファレンスは同じになります。

```
>>> cached_groups[0] == cached_groups[n]
True
>>> cached_groups[0].add(1)
>>> cached_groups[n]
{1}
```

そのせいで、かなり時間を失いましたね。

後は友達と問題を説明したら、他の方法を説明してもらいました：

```
cached_groups = [set() for _ in range(max_vertex_groups)]
```

意外と、その方法を使うと、かくインデックスのSetリファレンスは別々になりますね。

にしても、Python語を触ってない人：このシンタックスは読めますか？

おれは読めませんね。そして、テストしないと結果を想像できないシンタックスだと思いますね。

まあ、にしても、結局そのBlenderAPIの問題とPython語の問題を超えたら、ツールを出来上がりましたね！

イエイ！

使い方は、アーマチュアの編集時に:

- 選んだボーンを「消したいボーン」に決めて、
- アクティブ選択を「ターゲットボーン」に決めて、
- 右クリックメニューの追加したアイテム「Merge With Active」を選ぶと、結合が行います。

「結合」のオペレーションは
消したいボーンの名頂点グループの情報をターゲットボーンのと積算して、
要らないボーンと頂点グループを消します。

じゃあ、完成だ！！！！

イエイ！！！！

(喜びの舞)

じゃあ、Commitして、GithubにPushして、そしてVRCHATしようぜ。

明日は他のプロジェクトに集中しようか

Part 2

ん？ え？ Boothでリリースしないですか？

いやあ、それから、Boothをリリースできるまでの道は結構長いんですよ！

あの時はね、そのスクリプトの機能を使うために、
「Scripting」タブで中身をコピーして、「起動」ボタンを押す必要でした。

モデルの最適化は一か月で3回ぐらいやっていますから、
こういう時間稼ぎにあんまり気にしません。

でも、本気にリリースしたいなら、それぐらいをやらないとだめだと思います：

- すべての面倒くさい所を抜いて、円滑に操作出来るにする
 - 馬鹿避け
 - UIを考えずに使えるにする
- 簡単なインストールできるようにする
- テストして、頼れるようにする (毎日で使う)
 - 問題が起こったら、簡単に直せるにする
- 文章を作る
- 言語に対応する
 - ソフトも
 - 文章も
- 紹介動画を取って、編集して、字幕を追加する
- 宣伝する

リリースするまでの時間を計れない！

まずは、なにをすればいいを解っていても、
どうやってできるのか決まっていません。

そして、やり方をなんとなく把握していても、
想定外の問題をいくつに遭える可能性があります！

さっきに見せた通り、コアー機能を開発していた時には、そうなりましからね！

やりたいことがいっぱい！

後はね、やりたいことがいっぱいですね！

つまらない物をリリースするより、
時間をやりたいことに集中するほうがよさそうです！

つまり、リリースするモチベーションがない！

「俺にとっては便利ですから、それを便利と思う人がいるはずですよ」は正しいんですが、ちゃんと宣伝しないと、そのソフトの存在を解る人がほとんどいません。宣伝するために、色々やらないとだめです。

なぜなら

中途半端はだめです！

中途半端なソフトを宣伝すると、来るユーザーがだいたい：

- そのツールを使うことを止めます。
- 偶には、不具合報告を出す。
- そして、最悪時には、Twitterで遭った問題を語って、そのツールを使わないと報告して始まりますね。

ですから、円滑な使い方、数秒で解る文章、そういう物を用意しないと、逆効果になりますね！

では、逆に、どうしてBoothでリリースしました

まずは、ある友達が自分のアバターも最適化したかったんですから、そのスクリプトを渡しました。

渡しましたが・・・

その友達はスクリプトを使えるまでめっちゃ苦労していましたね。

それを見て「あれ渡すこと程度ではない。どんなBlender・ユーザーがちゃんとした使えるツールにしないとだめだ」と言う恥を感じて、リリースすることを考えました。

そして、後は、英語版のリリースを用意して、
Githubリポジトリの綺麗にして、
Twitterで宣伝したら
使いたくなる人が居た見たいですから、
ちゃんとBoothでリリースして続きました。

そして、一か月後、出来ましたね！

イエイ！

(喜びの舞)

質問コーナー