

VR コンテンツのつくりかた 3

Hello, VTuber World!

youten 著

2018-04-22 版 発行

目次

第 1 章	はじめに	5
1.1	本書の賞味期限とリポジトリ公開について	5
第 2 章	バーチャル YouTuber とそのシステム	7
2.1	バーチャル YouTuber とは	7
2.2	バーチャル YouTuber を実現するシステムの方式	8
2.3	アバターシステム	9
2.3.1	2D アバターシステム	9
2.3.2	3D アバターシステム	10
2.3.3	VR 向け 3D アバター新フォーマット VRM	15
2.4	モーションキャプチャーシステム	17
2.4.1	ボディトラッキング	17
2.4.2	フェイストラッキング	20
第 3 章	Unity と HTC Vive で 3D アバター VTuber システムをつくってみる	22
3.1	基本方針	22
3.1.1	完成品と OSS リポジトリ	23
3.1.2	XR Mecanim IK Plus	24
3.1.3	Unity XR API	24
3.2	3D モデルの調達について	24
3.2.1	発注しよう…としたけど仕様が決められない	24
3.2.2	完成品のモデルを利用する	25
3.2.3	筆者の悪あがき事例紹介	26
3.3	機能の実装（基本編）	27
3.3.1	はじめに完成品の動作確認	28
3.3.2	3D モデルへの IK 適用設定	28
3.3.3	XR API	31

3.3.4	リップシンク	33
3.3.5	自動まばたき	36
3.3.6	コントローラによる表情の変更	36
3.3.7	コントローラによる手のポーズ制御	37
3.3.8	ビルド	38
3.4	撮影と編集	39
3.4.1	OBS (Open Broadcaster Software) Studio での撮影	39
3.4.2	AviUtl と拡張編集プラグインでの字幕つけ	40
3.5	その他の機能の実装例	43
3.5.1	グリーンバックによるアバターの切り抜き	43
3.5.2	UnityCam でカメラ出力	44
3.5.3	Tracker 対応と両足のトラッキング	47
3.6	まとめ	48
	あとがき	50

第 1 章

はじめに

はいどうもー。バーチャル YouTuber が大好きな youten と申します。

バーチャル YouTuber が VR コンテンツか？ というと直接的にはそうではないと思います。しかし、使われている技術や、地続きとなっている文化は VR のものでしょう。

というわけで、「VR コンテンツのつくりかた 3 Hello, VTuber World!」と題しまして、「バーチャル YouTuber のやり方」を本書ではお届けします。

本書は、以下のような方をターゲットにしています。

- バーチャル YouTuber に使われている技術が知りたい
- Unity+Vive ベースのバーチャル YouTuber を試してみたい
- Unity のプログラミングや 3D モデリングなどに興味、または経験があり、バーチャル YouTuber をやってみたい

1.1 本書の賞味期限とリポジトリ公開について

なお、本書は記載内容の賞味期限が切れてしまうことへのアップデート対応、あまり紙メディアが向いていない環境等のバージョンアップに追従するため、初版以降の PDF と関連ファイル一式を全て以下のリポジトリで公開予定です。あらかじめご了承ください。

- <https://github.com/youten/howto-create-vr-contents3>
 - Re:VIEW の素材一式、出力 PDF、関連プロジェクトのソースコードを全て含む想定です。

本書のうち、私 youten が著作権を有する範囲のライセンスについては、文章は CC-BY 4.0 ライセンス^{*1}を適用します。

本書籍は Re:VIEW で作成されており、その設定ファイル等について、MIT ライセンスに基づき「C89 初めての Re:VIEW v2」リポジトリ^{*2}で公開されているものを利用させていただいております。

^{*1} <https://creativecommons.org/licenses/by/4.0/legalcode.ja>
^{*2} <https://github.com/TechBooster/C89-FirstStepReVIEW-v2>

第2章

バーチャル YouTuber とそのシステム



▲図 2.1 キズナアイの YouTube チャンネル A.I.Channel

2.1 バーチャル YouTuber とは

バーチャル YouTuber とは、バーチャル（仮想的な、事実上の）YouTuber のことで、VTuber などと略されます。YouTube のチャンネル登録者数 100 万人超えのトップランナーであり、"親分"と称される「キズナアイ^{*1}」さんが自称したのが用語としての始まりと言えますが、明確な定義があるわけではなく、またその形態は様々であるため、実際

^{*1} キズナアイの Youtube チャンネル A.I.Channel <https://www.youtube.com/channel/UC4Ya0t1yT-ZeyB00mxHgolA>

にリアルの人間ではない仮装アバターを用いていればおおむね全てバーチャル YouTuber と呼ばれているのが現状です。

キズナアイさんの最初の YouTube 動画投稿は 2016/11/29 ですが、界隈としては 2011/06/13 に「First Vlog. Trying this out.」というタイトルでロンドンからのビデオメールを投稿した、「Ami Yamato^{*2}」さんなどが始祖であると言われています。

また、YouTube をメインターゲットとしない「にじさんじ^{*3}」の方々は「バーチャルライバー」、SHOWROOM での生放送がメインの「うたっておんぷっコ♪ 東雲めぐ^{*4}」さんは「バーチャル SHOWROOMER」と名乗られています。

2.2 バーチャル YouTuber を実現するシステムの方式

バーチャル YouTuber を実現するシステムは主な構成要素として、キャラクターを表現するアバターシステムと、そのキャラクターアバターをどのように動かすかというモーションキャプチャシステムの二つから成り立ちます。

システムの方式についてはアバターを 2D/3D どちらで実現するかという点で大きく 2 つに分かれ、その後、いわゆる中の人のモーションをどのようにキャプチャするか、あるいは擬似的に生成するか、その方式によって区分することができます。このアバターシステムとモーションキャプチャシステム以外の企画や脚本、音声収録（ボイスチェンジャーを含む）や動画編集については通常の YouTuber と変わりがありませんので、重要な要素ではあるのですが本章では取り扱いません。

^{*2} Ami Yamato - First Vlog. Trying this out. <https://www.youtube.com/watch?v=5pcilZ8ffPY>

^{*3} にじさんじ 公式バーチャルライバー <https://www.ichikara.co.jp/official>

^{*4} はびふり！ 東雲めぐちゃんのお部屋♪ <https://www.showroom-live.com/megu>

2.3 アバターシステム

2.3.1 2D アバターシステム

Live2D



▲図 2.2 Live2D Cubism

「Live2D^{*5} Cubism」は、Live2D 社の 2D イラスト絵のパート毎に動きの情報を指定することでアニメーションさせることができる技術です。

ソシャゲに多く採用されていますが、個人的に「バンドリ！ ガールズバンドパーティ！^{*6}」の Live2D 事例が随一だと思っています。CGWORLD の Craft Egg インタビュー^{*7}が技術的なポイントを掴みやすく面白いのでぜひ一読を。

Live2D Euclid と E-mote とその他

デファクトスタンダードとして知名度のある Live2D ですが、一番メジャーである、その 2D イラスト絵を平面上で立体的に動かす技術は正式には「Live2D Cubism」といいます。「Live2D Cubism」とは別にボディは 3D、顔はアニメ技法的な投影を駆使して 3D 動作を実現する「Live2D Euclid^{*8}」という技術・製品が別にあります。

その様子は（おそらく唯一の）Live2D Euclid を利用した VTuber である「ふえあり

^{*5} Live2D <http://www.live2d.com/ja/>

^{*6} バンドリ！ ガールズバンドパーティ！ <https://bang-dream.bushimo.jp/>

^{*7} 『バンドリ！ ガールズバンドパーティ！』の登場キャラを魅力的にした、Live2D によるモーション制作 | 特集 | CGWORLD.jp <https://cgworld.jp/feature/201709-cgw230HS-bang.html>

^{*8} Euclid | Live2D <http://www.live2d.com/ja/products/euclid>

す^{*9}」さんや、Live2D Euclid のサンプルキャラクターである風花ちゃん（青髪の子）を利用することができる iPhoneX 向けアプリの「パパ文字^{*10}」などにて確認することができます。

また、混同されやすいのですが Live2D とは別に、同様の 2D キャラクターアニメーションツールとして、エムツー社の「E-mote^{*11}」という技術・製品があり、VR にも対応した「E-mote VR」というバリエーションがあります。筆者が「E-mote VR」を VR 結婚式で体験した^{*12}感想としては、ビルボード感があるものの、元ゲームの 2D と VR 空間としての 3D が同一技術のため再現度という観点では不思議な没入感がありました。

新規 HTML5 プラットフォームで注目のシャニマスこと「アイドルマスター シャイニーカラーズ^{*13}」の 2D アバターシステムは独自実装とのことで、キャラクターゲームのビッグタイトル用途から個人 VTuber まで Live2D 一強の現状が、変わっていくこともあるのか注目どころです。

2.3.2 3D アバターシステム

Unity, UnrealEngine (ゲームエンジン)

「Unity, UnrealEngine (ゲームエンジン)」という表現はあまり適切ではないとは思いますが、現状、モーションキャプチャや映像・カメラ制御との連携、あるいは外部システムと連携の上、3D キャラクターを統括的に扱うことができるプラットフォームとして、ゲームエンジンが 3D アバターシステムの主流となっています。

「Oculus Rift や HTC Vive などの VR HMD に関して、開発初心者向けの情報が多い」「VRChat のアバター・ワールドの SDK が Unity (5.6.3p1) に限定されているため、それを前提とした初心者向け情報が現在進行形で増えている」という要素から、未経験者が自由度の高いプラットフォームに手をつけてみたい際には、Unity がオススメです。

と、書いてはみたものの、CGWORLD 2018 年 5 月号にてんちょーさんによる UE で VTuber する話も掲載されました^{*14}し、お好きな方でチャレンジすれば良いと思います。

*9 Fairys Channel - YouTube <https://www.youtube.com/channel/UC2Rr7mILebYLTjd38DNNUTw>

*10 パパ文字 | 株式会社 ViRD <http://vird.co.jp/product/puppemoji/>

*11 E-mote について | キャラクターアニメーションツール E-mote <http://emote.mtwo.co.jp/about/>

*12 リアル式場でゲームキャラと VR で結婚、誓いのキスも「新妻 LOVELY × CATION VR 結婚式」レポート - PANORA <http://panora.tokyo/32636/>

*13 アイドルマスター シャイニーカラーズ (シャニマス) | バンダイナムコエンターテインメント <https://idolmaster-shinycolors.bxd.co.jp/>

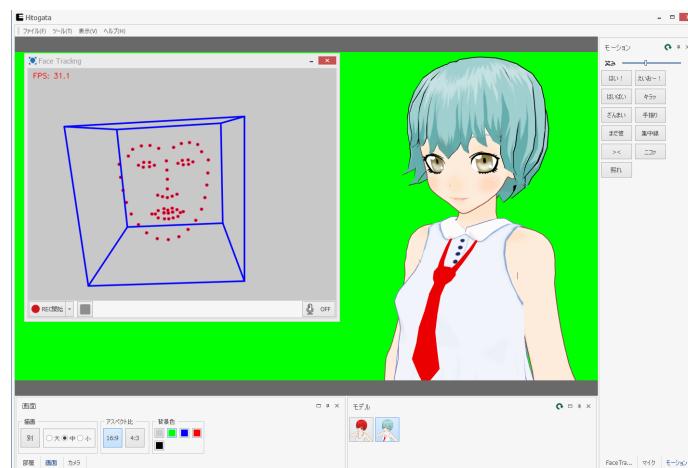
*14 【チュートリアル】Gray ちゃんになれる！ UE4 × Oculus でカンタンアバター | 特集 | CG-WORLD.jp <https://cgworld.jp/feature/201804-cgw237t1-gray.html>

MikuMikuDance + MikuMikuCapture, Hitogata, FaceVTuber

VTuber のメインの最終ターゲットは動画なので、3DCG ムービー製作プラットフォームである MikuMikuDance（略称：MMD）はノウハウドキュメントを含めた既存資産も多く、有力なプラットフォームの一つです。

2017 年末時点では、全身の姿勢を取得することができるセンサデバイスである Kinect を用いた「MikuMikuCapture^{*15}」がほぼ唯一の選択肢でしたが、Kinect はすでに生産終了してしまっているという欠点があります。

そんな中、MMD 関連システムと言える流れで、2018 年に入って WebCam ベースの「Hitogata^{*16}」と「FaceVTuber^{*17}」が出てきました。



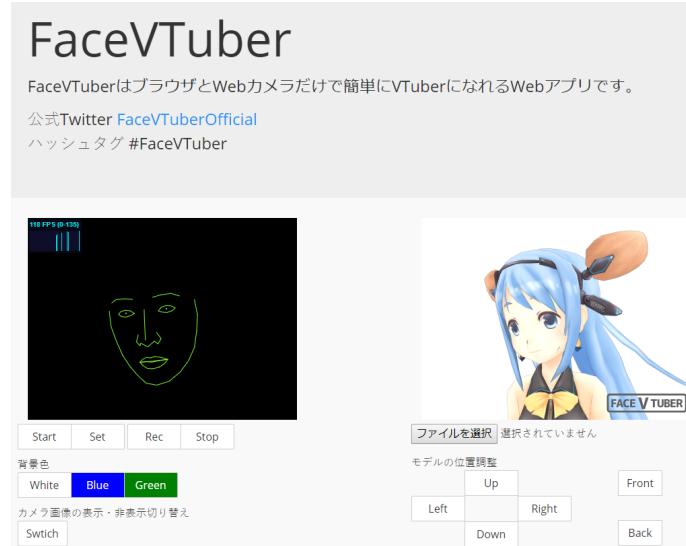
▲図 2.3 Hitogata beta2.1

「Hitogata」は MikuMikuMoving 作者の Mogg 氏によるアバターシステムで、キャラメイク機能が含まれており、単体で素材となる動画を出力できるシステムです。また、vmd エクスポート機能があり、「フェイス周りのモーションのみ Hitogata で生成したものを組み込んで MMD 動画をつくる」という使い方もできます。

^{*15} MikuMikuCapture <https://sites.google.com/site/mikumikucapture/>

^{*16} Hitogata <https://sites.google.com/site/vhitogata/>

^{*17} FaceVTuber <https://facevtuber.com/>



▲図 2.4 FaceVTuber

「FaceVTuber」はブラウザで完結して独自のモデルの読み込みに対応し、動画の録画と保存を実施することができるアプリです。インターフェイスからはわかりづらいですがライブラリを読み込む以外はローカルで動作し、モデル等のアップロードは行いません。Three.js の MMDLoaderあたりの機能を用いていると思われます。

パペ文字, ホロライブ



▲図 2.5 パペ文字（左）とホロライブ（右）

カメラ画像から顔の部位検出については、最近は機械学習でまた新たなフェーズを迎えてはいますが、昔から存在する定番の技術です。TrueDepth カメラの機能を利用したアプリをインストールした iPhoneX を貸し出している運用であるにじさんじ組に引き続き、カメラ画像からの入力でアバターシステムを実現するスマホアプリがこの 4 月頭、立て続けに 2 本、リリースされました。

「パパ文字^{*18}」は ViRD 社の iPhoneX 向けアバターシステムで、プリインの 3D キャラクター、Live2D Euclid のキャラクター、マスク画像などを切り替えて使うことができます。パパ文字はその他のモデルの個別対応の相談をリリース時から受け付けているあたりが気になるポイントです。

また、「ホロライブ^{*19}」は「ときのそら^{*20}」さんの運営元であるカバー社製のアバターシステムで、既存の AR キャラクターアプリをリニューアルする形でリリースされました。ホロライブは iPhoneX・iPhoneX 以外の iPhone・Android とプラットフォーム・ハードウェア特性にあわせて部位のトラッキング方式が違うことが特徴です。

元々、GOROMan さんが FaceRig + Live2D で実現した^{*21}時点から明らかではあるのですが、本質的には顔の向きとそれっぽい表情の切り替えができればアバターシステムとしての要件は満たしていると言えます。スマホアプリでライブ配信プラットフォームごとトータルサポートされたアプリが登場するのもそれほど遠くない未来だと思われます。

バーチャルキャスト

「あいえるチャンネル^{*22}」を有するインフィニットループと、遊び要素たっぷりの「バーチャルのじゃ口り狐娘 YouTuber おじさんことねこます^{*23}」さんやにじさんじの「月ノ美兎^{*24}」さんのニコ生を実施しているドワンゴという強力タッグによる、VR ライブ・コミュニケーションサービス「バーチャルキャスト^{*25}」がβリリースされました。VR キャラクターになりきってインターネット越しに音声を含むコミュニケーションを取ることが

*18 パパ文字 | 株式会社 ViRD <http://vird.co.jp/product/puppemoji/>

*19 「ホロライブ」を App Store で <https://itunes.apple.com/jp/app/id1288450594>

*20 SoraCh. ときのそらチャンネル - YouTube <https://www.youtube.com/channel/UCP6993wxpyDPHUpavvwDFqgg>

*21 中の人（二次元）になる方法 [FaceRig × Live2D × Unity × OBS × AVVoiceChanger × 気合] - Medium https://medium.com/@GOROMan_1661/%E4%B8%AD%E3%81%AE%E4%BA%BA-%E4%BA%8C%E6%AC%A1%E5%85%83-%E3%81%AB%E3%81%AA%E3%82%8B%E6%96%B9%E6%B3%95-facerig-live2d-unity-obs-avvoicechanger-%E6%B0%97%E5%90%88-d49c3d456ed6

*22 あいえるちゃんねる/株式会社インフィニットループ - YouTube https://www.youtube.com/channel/UC_IfjiHP6UnWEONNog_n_3A

*23 けもみみおーこく国営放送 - YouTube <https://www.youtube.com/channel/UCt8tmsv8kL9Nc1s xvCo9j4Q>

*24 月ノ美兎 - YouTube https://www.youtube.com/channel/UCD-miitqNY3nyukJ4Fnf4_A

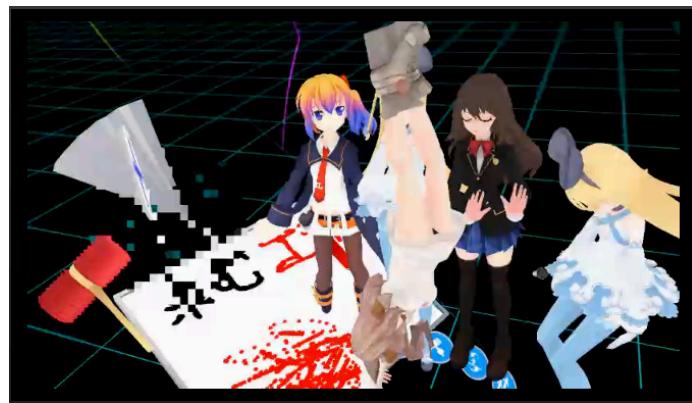
*25 バーチャルキャスト [Virtual Cast] <https://virtualcast.jp/>

できます。



▲図 2.6 バーチャルキャスト

「みゅみゅ^{*26}」さんがニコ生でやっていた VR 空間システムがベースになっているとのことですが、色々機能がてんこ盛りな上に VRM 形式のオリジナル 3D モデルのインポートにもすぐに対応予定^{*27}とのことで、今後が楽しみなサービスです。



▲図 2.7 Tracker の追加で逆立ちも可能

少しマニアックな観点でごいなと思ったところを挙げておきます。

*26 □。*みゅみゅちゃんねる♡*. □-ニコニコミュニティ <https://com.nicovideo.jp/community/co1774343>

*27 「VRM 形式の投稿」「バーチャルキャスト連携許可」に対応しました - ニコニ立体お知らせブログ <http://blog.nicovideo.jp/3d/2018/04/vrm.html>

- ・ 絶妙な広角めのカメラ一体型ミラーが標準配備。他アイテム同様配置を変更できるのであおりも俯瞰も自在ですぐに面白い画を作って試せます。
- ・ 見栄えもするダイナミックでわかりやすい UI。一人称向けの手元空中コンソールではなくて、外側からもメニュー表示が見えます。
- ・ ファーストリリースでニコ生コメント連携も対応、他の人のルームに参加する「凸」の対応とその手軽さ。きちんと凸許可設定と他の凸可能な人のリストがあつてすぐ凸できる IF がきっちり揃っていてうまいなと思いました。
- ・ Tracker 追加によるトラッキングポイントの増加に対応、キャリブレーション時に自動で認識、最大で Tracker を 7 つ追加して両肘・両膝・両足・腰の 10 点トラッキング対応（図 2.7）。VRChat のいわゆる「フルトラッキング勢」の装備がまさかの他サービスでも使えるようになるなんて、Tracker の品薄が続くこと間違いないしの衝撃です。

2.3.3 VR 向け 3D アバター新フォーマット VRM

2018 年 4 月 16 日、「オリジナルアバターを登録して利用することもできる」とアナウンスされたバーチャルキャスト β 版のリリースからわずか 3 日、ドワンゴ社から「VR 向け 3D アバター新フォーマット VRM^{*28}」が発表されました。

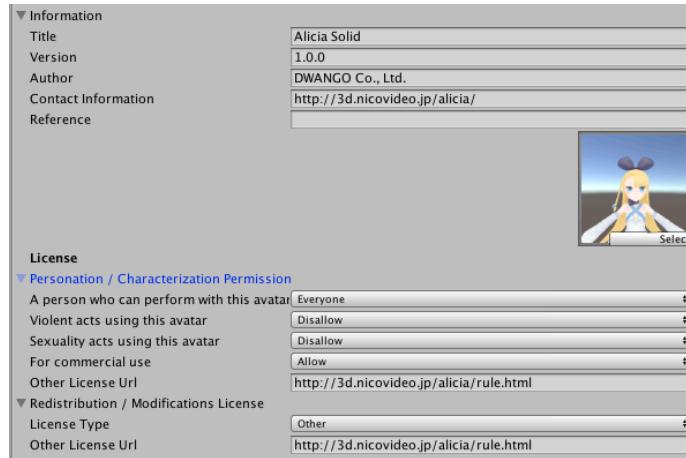
VRM のドキュメントを一通り眺めて、VRM を Unity で扱うための実装・ツールである UniVRM を触ってみたファーストインプレッションは筆者の Blog に掲載しています^{*29}のでそちらを参照願います。ここでは少し脱線して、筆者が面白いと感じたポイントについて述べさせていただきます。

メイン用途のアバターにフォーカスしつつ、別用途にも対応できるライセンス設定 UI の思想

まさに VTuber や VRChat のようなメタバースでの利用を想定した、「アバターの人格に関する許諾範囲 (Personation / Charaterization Permission)」という項目があることが注目されていますが、Unity 上でのその入力 UI から思想を感じました。

^{*28} VRM - dwango on GitHub <https://dwango.github.io/vrm/>

^{*29} VR 向け 3D アバター新フォーマット「VRM」発表、ドキュメントひととおり読んでみた話 - ReDo <http://greety.sakura.ne.jp/redo/2018/04/vr3dvrm.html>



▲図 2.8 UniVRM による、VRM メタ情報の入力 UI

まず、「アバターの人格に関する許諾範囲 (Personation / Charaterization Permission)」の設定項目について、まずメイン用途であるアバターとしての利用許可に関する設定が並んだ上で、それ以外に何か規定したい際には「別途サイトを用意して URL を記載してください」と並んでいます。続いて、「再配布・改変に関する許諾範囲 (Redistribution / Modifications License)」の設定項目について、「禁止」と「CC0 から CC ライセンスのバリエーション」が並んだ上で、それ以外に何か規定したい際には、同様に「別途サイトを用意して URL を記載してください」と並んでいます。

あくまでテンプレートとして CC などの運用コストの低いライセンスをまず選択式で用意した上で、その他もカバーしているこの並びは良いな、と思いました。

glTF 2.0 ベースの仕様に MIT ライセンスで統一された UniVRM のフルスクラッチ

VRM のフォーマットとは別に、Unity 向けの実装実例である UniVRM は、そのサブモジュールとともに全て MIT ライセンスで統一された OSS になっています^{*30}。

MIT に従う限り、誰もがロイヤリティも用途の制限も無く自由に使うことができ、UniVRM を含めたシステム・ライブラリも問題なく OSS にできます。独自拡張をサービスに閉じて実装することもできますし、そしてもちろん「伽藍とバザール」のバザール方式として VRM そのものの開発に参加することもできます。

仕様と関連ツールがセットで OSS なことは、VRChat SDK の「Final IK とか Dynamic

^{*30} dwango/UniVRM: Unity package that can import and export VRM format <https://github.com/dwango/UniVRM/>

Bone とか前提だ、Asset Store ライセンスにつっこむぞつかまれッ！」という大胆な方針^{*31}と比較してとてもクリーンに感じました。

2.4 モーションキャプチャシステム

モーションキャプチャシステムについては、身体の部位をトラッキングするものから、顔をトラッキングするもの、あるいは音声入力から唇の形を推測するリップシンクなどが含まれます。単体ではなく、複数のシステムを組み合わせて利用するケースもあります。また、「キャプチャ」とは言っていますが、擬似的に生成をしたり、手動で操作したり、編集フェーズで調整することも組み合わされます。

2.4.1 ボディトラッキング

HTC Vive + Tracker

VR ヘッドセットシステムである「HTC Vive」を用いる方法です。

VR ヘッドセットシステムは VR HMD の頭部と、コントローラの両手の空間座標をキャプチャすることができ、ここからキャラクターの姿勢を推定することができます。



▲図 2.9 HTC Vive での頭 (HMD)・両手 (コントローラ)・腰・両足 (Tracker) の 6 点トラッキング

また、HTC Vive は「Tracker」という、HMD とコントローラ以外にもトラッキング

^{*31} Supported Script Using Assets <https://docs.vrchat.com/docs/supported-assets>

ポイントを追加することができるデバイスが標準で提供されており、主には頭（HMD）・両手（コントローラ）に加え腰・両足などを追加し、Tracker でトラッキングすることにより全身のモーションキャプチャが可能となります。

IKinema 社の Orion^{*32}は 6~8箇所の Tracker を含めた Vive トラッキングシステムから全身のモーションキャプチャをするシステムで、100FPS オーダーのリアルタイムキャプチャの他、fbx へのエクスポートもサポートしています。

IK による姿勢の推定

VR ヘッドセットシステムにより、各トラッキングポイントの座標と向きが取得できますが、そこから骨格を IK (Inverse Kinematics の略、逆運動学) で求めることにより、姿勢を推定することができます。

定評のある IK ライブラリである、「Final IK^{*33}」を組み合わせるのが 3D VTuber システムの一つの定番です。Unity 標準の IK 機能^{*34}もあり、第3章ではこちらと XR API を用いて VTuber システムを構築した話を紹介します。

Oculus Rift

VR ヘッドセットシステムである「Oculus Rift」を用いる方法です。



▲図 2.10 Oculus Rift での頭（HMD）・両手（コントローラ）3 点トラッキング

HTC Vive と比較して「Tracker のような追加トラッキングユニットによる全身トラッキングには対応していない」というディスアドバンテージがありますが、逆にいようと全

^{*32} IKinema | Motion Capture, VR, Games - Project Orion <https://ikinema.com/orion>

^{*33} Final IK - Asset Store <https://assetstore.unity.com/packages/tools/animation/final-ik-14290?aid=10111Gbg>

^{*34} Unity - マニュアル: IK <https://docs.unity3d.com/jp/530/Manual/InverseKinematics.html>

身トラッキングが不要であるケースでは「価格面で 20,000 円ほどベースセットが安い」「Vive の棒状のコントローラと比較して、Oculus Touch の方が手の形を直感的に表現できる」という観点で Oculus Rift の方が優れています。

前述した東雲めぐさんは XVI 社の「AniCast^{*35}」を用いていますが、こちらは Oculus Rift ベースのシステムです。

個人的には HTC Vive と比較して「軽量」「ヘッドホンシステムが標準装備で取り回しがしやすい」とこと、「Rift Core 2.0 (β) による高パフォーマンスのデスクトップキャプチャが標準で使える」という利点があり、HTC Vive のルームスケールを活かせる広い部屋がない際など、Vive より Rift が VTuber システムのベースに向いてるケースもそれなりにあると思っています。

Perception Neuron

モーションキャプチャシステムの「Perception Neuron^{*36}」を用いる方法です。



▲図 2.11 Perception Neuron

この手のモーションキャプチャシステムとしては非常に安価（約 21 万円）というのが特徴で、これと光学式カメラによるフェイシャルキャプチャを組み合わせた仕組みが大手（？）VTuber では主流の方式となっています。

公表されてる範囲では、Perception Neuron ベースのキャラクターシステムとして先駆者である KiLA^{*37}が「虹川ラキ^{*38}」さんなどで採用されている他、「電腦少女シロ

^{*35} XVI Inc. | パーチャルキャラクター配信システム「AniCast」プレスリリース <http://www.xvi.co.jp/news/anicast/>

^{*36} Perception Neuron by Noitom | Perception Neuron motion capture for virtual reality, animation, sports, gaming and film <https://neuronmocap.com/ja>

^{*37} Kigurumi Live Animator [KiLA] <https://kila.amebaownd.com/>

^{*38} Laki Station ラキステーション - YouTube <https://www.youtube.com/channel/>

YouTuber シロ^{*39} さんや「ばあちゃんる^{*40}」さんは運用元であるアップランド株式会社の勉強会の「バーチャル美少女ねむ^{*41}」さんによるレポート^{*42}にて、Perception Neuron を使用していることがわかります。

その他のモーションキャプチャシステム

その他、大規模から小規模まで、ビジネス用途の製品から、研究目的のシステムまで、さまざまなモーションキャプチャが使われています。

設備が大きく少し高額ですが光学式マーカーシステムの自由度が高く、複数人数での利用にも強い OptiTrack^{*43}、生産終了とはなってしまいましたが Xbox プラットフォームでのゲームへの利用・応用事例の豊富な Kinect^{*44}、Kinect の後継として期待されている RealSense^{*45}が実際に VTuber に使われているのを確認できます。

また、現時点で画像ベースのトラッキングライブラリの雄である OpenPose^{*46}はライセンスの観点で VTuber では利用しづらいのですが、機械学習ベースの画像認識ライブラリそのものの進化は進んでおり、キャラクター設定によってはトラッキングの精度の低さが許容できる際など、「WebCam ひとつあればいい」という世界はそれなりに広がっていくと考えられます。

2.4.2 フェイストラッキング

フェイストラッキングは現状、主にカメラによる「画像認識からの顔の部位検出」とマイクによる「キャプチャ音声からの口の形を推測するリップシンク」の2種類の技術が主流で、これらの両方を組み合わせたりされています。

ただし、表情のトラッキングはせずにスクリプトで自動まばたきを行ったり、親しみやすい漫画・アニメ的表現である「><顔」や「ジト目」や、汗の漫符など、モーフを手動で切り替えてしまう方法も多くの VTuber が採用しています。

UCp77Qho-YHhnk1Rr-Dkj10w

*³⁹ Siro Channel - YouTube https://www.youtube.com/channel/UCLhUvJ_w09h0vv_yYENu4fQ

*⁴⁰ 【世界初?!】男性バーチャル YouTuber ばあちゃんる - YouTube <https://www.youtube.com/channel/UC6TyfKcsrPwBsBnx2QobVLQ>

*⁴¹ ねむちゃんねる【バーチャル美少女 YouTuber】 - YouTube https://www.youtube.com/channel/UC7xyzq_Hd72-IbJVoz4DSrQ

*⁴² バーチャル YouTuber だけど、シロちゃんの秘密を探りに「第1回バーチャル YouTuber 勉強会」に行ってきた！ | バーチャル美少女ねむの人類美少女計画 <https://www.nemchan.com/2018/01/VtuberWorkshop.html>

*⁴³ OptiTrack Japan, Ltd. | オプティトラック・ジャパン株式会社 <https://www.optitrack.co.jp/>

*⁴⁴ Kinect - Windows アプリの開発 <https://developer.microsoft.com/ja-jp/windows/kinect>

*⁴⁵ Intel ® RealSense Technology | Intel ® Software <https://software.intel.com/realsense>

*⁴⁶ OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

カメラ画像による部位検出について、VR HMD を装着してしまっている際には目の位置などはもちろん検出できなくなってしまうのですが、Tobii 社の視線追跡技術を HTC Vive に組み込んだもの^{*47}がある他、モバイル向けにも Qualcomm 社の Snapdragon 845 ベースの次世代スタンダードアローン VR ヘッドセット開発キット^{*48}にも同 Tobii 社の視線追跡技術が組み込まれています。

現世代の VR HMD にアイトラッキングが標準搭載されてはいませんが、その高解像度化・高 FPS 化していく進化「視線が向いている部分だけ高解像度化する」という Foveated Rendering が高パフォーマンスのためには必要と考えられており、その際にアイトラッキングの機構が必須となります。

人間の目は微細眼球運動と呼ばれる、無意識下で視線が揺れる運動を行うことがわかっていますが、これがゆえに視線は入力ポインタとしては扱うことが難しいことがわかつています。ただし、人間らしさの再現としてトラッキングできた眼球運動そのものをキャラクターに反映させることができると、よりリアリティの高い「生きた目」を実現することができるでしょう。

^{*47} Tobii Pro offers VR integration services <https://www.tobiipro.com/ja/product-listing/vr-integration/>

^{*48} ...Virtual Reality Development Kit <https://www.qualcomm.com/news/releases/2018/03/21/qualcomm-announces-support-next-generation-vr-experiences-new-snapdragon>

第 3 章

Unity と HTC Vive で 3D アバター VTuber システムをつくってみる

さて、バーチャル YouTuber システムの技術要素について理解できましたら、次は実際につくってみましょう。

この章では、Unity を用いて、VR ヘッドセットシステムである HTC Vive をベースとした 3D アバターの VTuber システムを作つてみた話について、要素をひとつひとつ説明します。

3.1 基本方針

前章のとおり、バーチャル YouTuber システムはその実現方式は多様です。本書で扱うバーチャル YouTuber システムはそのやり方を主に 2 つの方針に沿つて決めました。

1 つは、「VR 技術を使った 3D アバターシステムである」ことです。こちらは筆者が VR 技術が好きで、トラッキングポイントが増やせたり、HMD の代わりに頭に装着することもできる Tracker が好きで、3D キャラクターを扱うシステムと仲良くするのが好きだからです。

もう 1 つは、本書などを参考に「やってみようと思った人が成功しやすいように、シンプルで、一通りの情報がまとまっている」ことです。



▲図 3.1 ReDo (筆者 Blog) - バーチャル YouTuber のやり方 #VTuber

本書の前身とも言えるのですが、2017 年度に Vive Tracker の登場からゆるゆると流行した、「VR コスプレ」の先駆者達の情報をかけあわせて、Unity+Vive+Final IK 式のシステムを組んでみて、その成果を筆者の Blog エントリ^{*1}にまとめました。

これを参考に VTuber システムを作ってみようと思うと、複数の情報を「いい感じに」繋げる必要があり、Unity と VR 開発経験がない人は正直脱落してしまっているんじゃないかな、と思いました。他人様のライブラリにおんぶにだっこではあるのですが、そのあたりをどうにかできるものを目指してみました。

3.1.1 完成品と OSS リポジトリ

本章で説明するバーチャル YouTuber システムは、YVTuber (Yume VTuber System)^{*2} という名前で OSS として GitHub に公開しています。

特徴としては Asset Store 配布のアセットを一切含んでおらず、ファイル一式をダウンロードして Unity で開くと、すぐ実行することができます。

また、このリポジトリには改造・再配布に制限のあるものを一切含んでおりません。詳細は各アセットフォルダ配下の readme や LICENSE ファイルを参照願います。

実際のところ、デファクトスタンダードである Final IK (VR IK) や、Steam VR SDK を素直に使った方が長い運用を考えるとお得だとは思うのですが、多彩な「俺の考えた最

^{*1} バーチャル YouTuber のやり方 #VTuber - ReDo <http://greety.sakura.ne.jp/redo/2018/01/how-to-vtuber.html>

^{*2} YVTuber <https://github.com/youten/YVTuber>

強のバーチャル YouTuber システム」がどんどん出てきてみんなで使い分けたり、OSS であれば自由に改造しあったりして、楽しめたらいいなと考えています。

3.1.2 XR Mecanim IK Plus

YVTuber は@chobi_luck さんの XR Mecanim IK Plus^{*3}をベースにしています。こちらは実のところ、XR Mecanim IK Plus は、ユニティちゃんモデルのアセットを追加することで、十分にバーチャル YouTuber システムとして成り立つ完成品です。これをベースに、ライセンス上自由度の高いモデル（後述します）を組み込み、全てのアセットをリポジトリに含む、別途一切のダウンロード・追加が必要ないものに組み立てました。

3.1.3 Unity XR API

Unity XR API は、Unity の標準機能として数多くの VR/AR/MR デバイスを扱えるようにした API です。

XR Mecanim IK Plus をベースにした副産物なのですが、Unity XR API をベースにしているため Steam VR プラグインが不要であり、Oculus Rift 環境でも一応動作します^{*4}。Steam VR 側が Oculus Rift や WinMR に対応している互換性については把握していたのですが、Unity 標準機能として徐々に整備されてきており、これはこれで十分だなと感じました。後述しますが Tracker にも対応しており、目立って非対応の機能というとバイブルーションぐらいのことです。

3.2 3D モデルの調達について

3D ベースの VTuber システムでは必須となる 3D モデルですが、その調達については悩みどころです。この節では、その 3D モデルの調達方式についての見解と、筆者が今回どうしたかということについて述べます。

3.2.1 発注しよう…としたけど仕様が決められない

餅は餅屋、最初に考えるのが誰かにお願いすることなのですが、1 体数十万から数百万と言われる 3D モデルを発注しようと考えた際に、すぐに「仕様が決められない」ことに気づきます。一般的なソフトウェアの世界に照らし合わせると、「発注範囲（工程）」と「機能要件」と「品質」について、どのような選択肢があるのかをわかっていないといけ

^{*3} XR Mecanim IK Plus http://chobi-glass.com/Unity/XR_MecanimIKPlus.html

^{*4} コントローラの形状が物理的に違うため、使いづらい機能があります

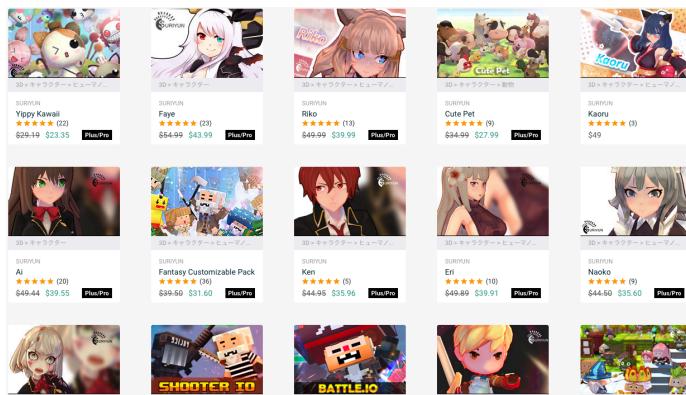
ません。まずは実際に使ってみたり、作ってみたりして勉強が必要だな、と考えました。

3.2.2 完成品のモデルを利用する

そんなわけで発注するにも勉強が必要なことが分かってしましたが、次に完成品のモデルを利用することを考えます。いくつかの具体例を挙げます。

Unity Asset Store で販売されているモデルを利用する

Unity の Asset Store で販売されているアセットは、無料・有料を問わず原則「改変可」「ロイヤリティ不要で商用利用可能」「実行形式に組み込んで容易に取り出せる形でなければ再配布可」と、あくまで Unity 上で動作させる際には特別な制限がありません^{*5}。



▲図 3.2 Unity Asset Store - SURIYUN

「カフェ野ゾンビ子^{*6}」さんは SURIYUN さんの「Zombie Girl01^{*7}」を、「雨下カイト^{*8}」さんは GAME ASSET STUDIO さんの「Taichi Character Pack^{*9}」を使われている模様です。

^{*5} アセットストアが選ばれる 5 つの特徴 <http://assetstore.info/howto/howto-point/>

^{*6} Zombi-Ko Channel - YouTube https://www.youtube.com/channel/UCiVRnULJjc8o-j_1G5BAzKw

^{*7} Zombie Girl01 - Asset Store <https://assetstore.unity.com/packages/3d/characters/humanoids/zombie-girl01-97923>

^{*8} Kite Channel - YouTube <https://www.youtube.com/c/KiteChannelOfficial>

^{*9} Taichi Character Pack - Asset Store <https://assetstore.unity.com/packages/3d/characters/taichi-character-pack-15667>

インターネットで配布されているモデルを利用する

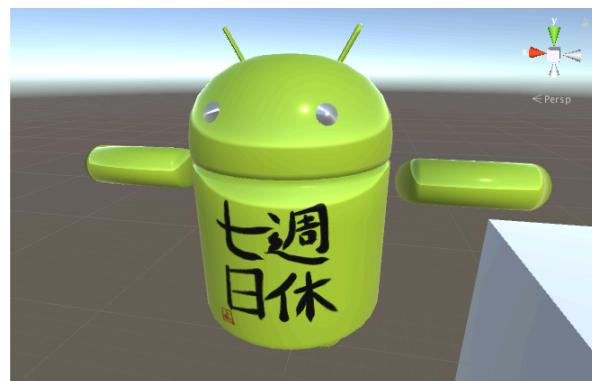
「みゅみゅ¹⁰」さんが「ニコニ立体ちゃんことアリシア・ソリッド¹¹」を使っている事例が一番有名だと思いますが、おなじみ「ユニティちゃん¹²」や「中野シスターズ¹³」など、いくつかライセンスの自由度が高いモデルが無償で配布されています。配布モデルそれぞれに利用規約がありますので、よく確認の上、遵守しましょう。

3.2.3 筆者の悪あがき事例紹介

結局のところ、モデラーさんが作られたモデルを使うのが正道なのですが、もう少し他の方法も探ってみました。

Blender での自作

自然な人型タイプではなく、各パーツが円柱や球でシンプルで、宙に浮いており、ロボットや不定形のキャラクターなら少々形が壊れても問題ないのでは、と考えました。筆者はジョイメカファイトが大好きです。



▲図 3.3 ドロイド君こと Android Robot

ちなみに、このドロイド君¹⁴をつくるのには、Blender の本をみながら 20 時間ぐらい

*10 みゅみゅ - YouTube https://www.youtube.com/channel/UC_tFMq_yvulj-yzQDqmH9SQ

*11 ニコニ立体ちゃん特設サイト - ニコニ立体 <http://3d.nicovideo.jp/alicia/>

*12 UNITY-CHAN! OFFICIAL WEBSITE <http://unity-chan.com/>

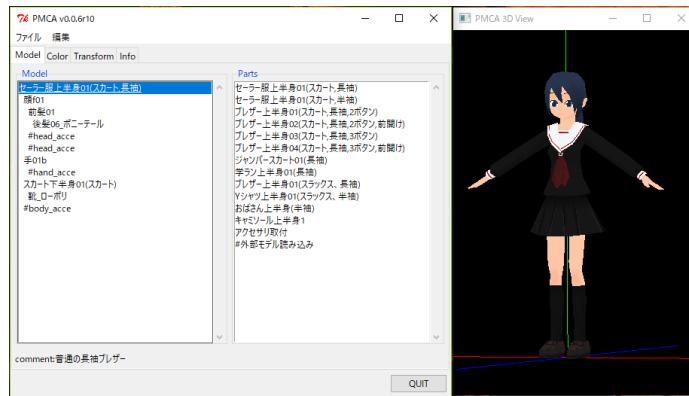
*13 中野シスターズ（ナカシス）公式サイト <http://nakasis.com/>

*14 youten/android-robot: Android Robot a.k.a. Droid-kun 3d model by blender <https://github.com/youten/android-robot>

勉強した後、6時間ほどかかりました。1回 undo できない間違った操作をしてしまい、ゼロからやりなおしました^{*15}。

PMCA での生成

実際にドロイド君を使ってみた感想として、「もう少し身長と高い頭身が欲しい」「手と指が欲しい」「モーフが欲しい」ということに気づきました。



▲図 3.4 PMCA (PMD Mob Character Assembler) v0.0.6r10

そこで、MMD モデル組み立てソフトである PMCA (PMD Mob Character Assembler)^{*16}を利用しました。このソフトはいわゆる「モブ子さん」を生成するツールなのですが、大胆なことにツールがまるっと Public Domain ライセンスと宣言されており、デフォルトで同梱されているパーツを使う限りにおいては、本ツールで生成されたモデルにも同ライセンスが適用されます。

これで生成した pmd を、Blender にて、mmd_tools^{*17}プラグインで fbx に変換したものを使っています。

3.3 機能の実装（基本編）

それでは、Unity 上で機能の実装を行なっていきましょう。

^{*15} やっていき Blender 第1回 週休七日ドロイド君をつくるまで - ReDo <http://greety.sakura.ne.jp/redo/2018/01/blender-1.html>

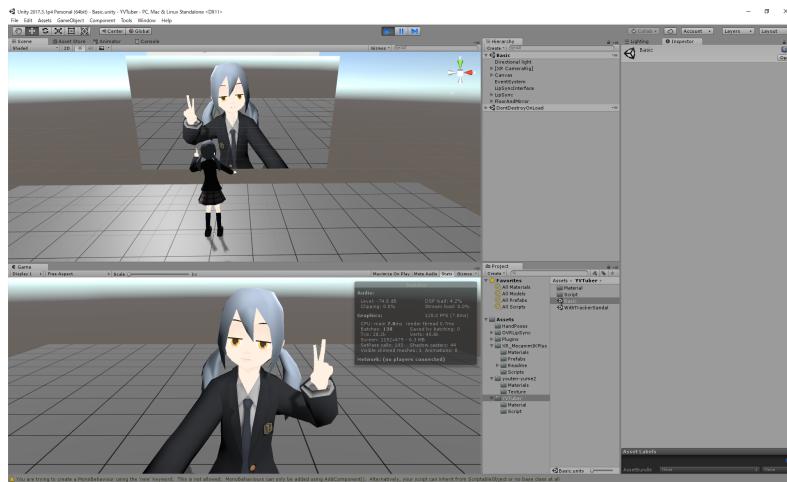
^{*16} PMCA (PMD Mob Character Assembler) v0.0.6 <http://matousus304.blog106.fc2.com/blog-entry-215.html>

^{*17} https://github.com/powroupi/blender_mmd_tools

3.3.1 はじめに完成品の動作確認

前述の YVTuber のリポジトリから、プロジェクト一式をダウンロードして、Unity で開いてください。筆者は 2017.3.1p4 で動作確認をしています。

「Assets」内、「YVTuber」>「Basic」シーンを開きます。SteamVR が起動しており、コントローラ 2 つも認識されていることを確認した上で、「Play」を実行します。正常に起動していれば、キャラクターの手や体がコントローラに追従することが、正面のミラーで確認できると思います（図 3.5）。



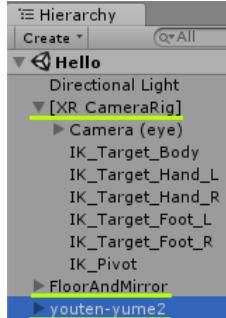
▲図 3.5 Basic シーンを動作させた様子

右コントローラのトラックパッドの上にある Menu ボタンで位置の Recenter ができます。シーン内にも表示されていますが、「YVTuber」>「Material」>「manual」に簡単な操作方法を示す画像が入っていますので確認してください。トラックパッドのクリックで手の形が、グリップ+トラックパッドのクリックで表情モーフを切り替えることができます。また、少々操作が難しいのですが、グリップで手と足を同期して動かすことができます。

以降、この「Basic」シーンをゼロから再現する手順と、その内容について説明していきます。

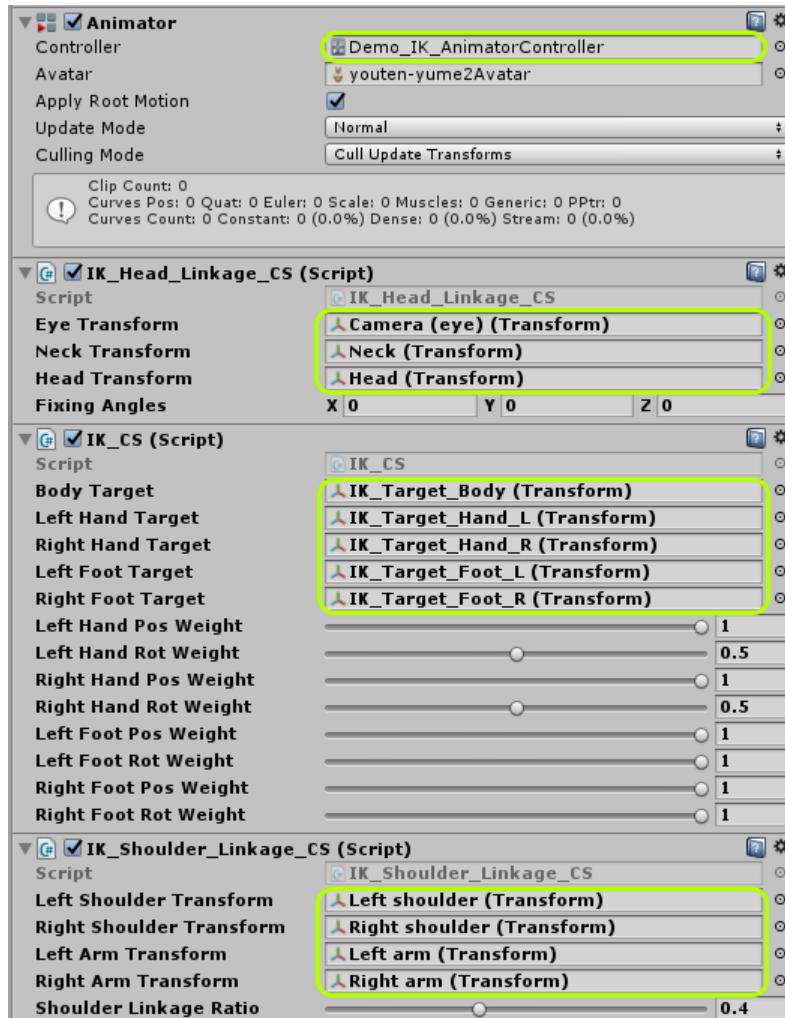
3.3.2 3D モデルへの IK 適用設定

まず、XR Mecanim IK Plus をモデルに適用します。

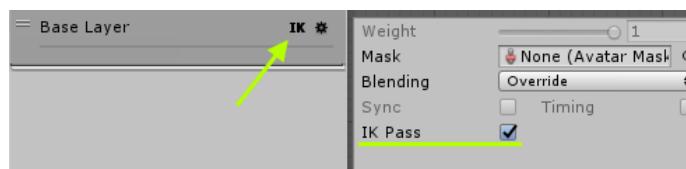


▲図 3.6 3つの Prefab を Hierarchy に追加

- メニューより、「File」「New Scene」を選択、"Hello"と入力し、新規シーンを作成します。
- Project タブより、「XR_MecanimIKPlus」>「Prefabs」配下の「[XR CameraRig]」と「FloorAndMirror」を Hierarchy に追加します。
- Project タブより、「youten-yume2」>「youten-yume2」を Hierarchy に追加し、「XR_MecanimIKPlus」配下の「Demo_IK_AnimatorController」Animator Controller と、「Scripts」配下の「IK_Head_Linkage_CS」「IK_CS」「IK_Shoulder_Linkage_CS」Script をアタッチします。
- 「IK_Head_Linkage_CS」の Eye Transform には [XR CameraRig] 配下の「Camera(eye)」を、Neck Transform には モデルのボーン構造 Armature 配下の「Neck」を、Head Transform には「Head」を設定します。
- 「IK_CS」の 5 つのターゲットには、[XR CameraRig] 配下の「IK_Target_Body」「IK_Target_Hand_L」「IK_Target_Hand_R」「IK_Target_Foot_L」「IK_Target_Foot_R」をそれぞれ設定します。
- 「IK_Shoulder_Linkage_CS」の 4 つのターゲットには モデル配下の「Left shoulder」「Right shoulder」「Left arm」「Right arm」をそれぞれ設定します。



▲図 3.7 Animator Controller と XR Mecanim IK Plus 3 つの Script の設定



▲図 3.8 Unity 標準 Mecanim IK の利用には、Animator Controller への IK 設定が必要

XR Mecanim IK Plus の本家サイトでも説明がありますが、設定している Animator Controller は空のものに、標準の Mecanim IK を有効にする「IK Pass」のチェックを加えただけのものですので、他 Animator Controller に差し替える際には、同様にチェックを追加してください（図 3.8）。

ここまで、3 つの Prefab（図 3.6）と、3 つの Script の設定（図 3.7）が終わって Unity エディタにて再生すると、頭と手を動かすことができます。また、グリップを組み合わせることにより、足も動かすことができます（図 3.9）。



▲図 3.9 XR Mecanim IK Plus とモデルの紐づけ完了まで

3.3.3 XR API

Unity 標準の XR API とは、VR/AR/MR のさまざまなデバイスを統一的に扱うために定義された API 群です。もともとは「VR API」という名前でしたが、Unity2017.2 から「XR API」に変更されました。ドキュメント中には「VR」という用語がそのまま残っているケースが多いです。

コントローラ入力

XR API はさまざまな XR デバイスに対応するために、最小限のプリミティブな API しか用意されておらず、そのコントローラ入力を取得するには、各プラットフォーム毎に設定を追加する必要があります。

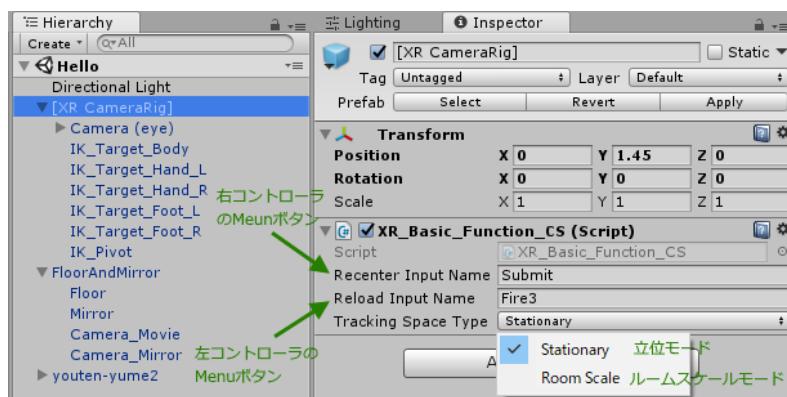
具体的にはラッパー Util クラスである「YVTuber」>「Script」>「XRVive」スクリプト内にコメントで記載していますが、Unity 公式の「Input for OpenVR controllers」

ドキュメント^{*18}を参考に、Input Manager に設定を追加しています。

例えば、左コントローラの Grip ボタン「Left Controller Grip Button」は「Unity Axis ID が 11」と記載されてていますので、「Edit」>「Project Settings」>「Input」の Input Manager 設定に以下のように「Name」に"Grip L"、「Type」に「Joysticks Axis」、「Axis」に「11th axis (Joysticks)」を設定します。

こうすると、「Grip ボタンが押されているか否か」は、「if (Input.GetAxis("Grip L") > 0)」といったコードで取得することができます。

トラッキングモード：立位とルームスケール



▲図 3.10 XR Basic Function Script のトラッキングモード設定

XR Mecanim IK Plus はデフォルトではトラッキングモードが「Stationary（立位モード）」になっています。これは（名前が直観的ではないのですが）椅子などに座って行うのに適した方法で、右コントローラの Menu ボタンに現在のヘッドセットの位置にあわせてリセットする「Recenter」が割り当てられています。

HTC Vive のルームスケールセットアップが有効であれば、これを「Room Scale（ルームスケール）」に変更することでルームスケールモードに切り替えることができます。このモードではカメラを配置した座標がルーム内中央の地面に設定されますので、[XR CameraRig] の Y 座標を 0 にする他、利用している部屋にあわせて各オブジェクトの配置を調整してください。また、ルームスケールモードでは Recenter は動作しません。

^{*18} Unity - Manual: Input for OpenVR controllers <https://docs.unity3d.com/Manual/OpenVRControllers.html>

3.3.4 リップシンク

続けて、リップシンクの実装です。マイク入力音声から唇のモーフ制御を行うために、Oculus 製の「OVRLipSync^{*19}」というライブラリを使います。

OVRLipSync には手を入れています

配布されている OVRLipSync 1.16.0 をベースに、以下のとおり改造したものをリポジトリに含めています。

- マウス・キーボード入力によるデバッグ機能切り替えは誤爆回避のためコメントアウト。
- 実行ウィンドウにフォーカスがないとマイク入力を一時停止する機能は不要のためコメントアウト。
- 「OVRLipSync.cs」と「OVRLipSyncContext.cs」に初期化順番依存バグがあるため、「OVRLipSync.cs」の優先度を上げる
 - [DefaultExecutionOrder(-1)]属性をクラスに追加。
- 「LipSync_Demo」シーンの機能切り替え実装が煩雑で分かりづらいため、マイク入力によるモーフ制御のみに絞った「SimpleMicMorph_Demo」シーンを追加。
- 副産物として、「OVRLipSyncMicInput」「OVRLipSyncContext」「OVRLipSyncContextTextureFlip」「OVRLipContextMorphTarget」スクリプトと付属の唇モデル向け設定を含む「InputType_Microphone」Prefab を追加。

このあたりの補足説明として、空の Unity プロジェクトから無改造の OVRLipSync の使い方を一通り説明する動画を YouTube にアップロードしています^{*20}。

モデルへの OVRLipSync の適用

では、実際に OVRLipSync を適用していきましょう。

- Project タブより、「OVRLipSync」>「Prefabs」配下の「LipSyncInterface」と「InputType_Microphone」を Hierarchy に追加します。
- 「InputType_Micrphone」にアタッチされている「OVRLipContextMorphTarget」の「Skinned Mesh Renderer」に、モーフ制御対象のモデル配下の「Body」を設

^{*19} Oculus Lip Sync Unity Integration Guide <https://developer.oculus.com/documentation/audiosdk/latest/concepts/book-audio-ovrlipsync/>

^{*20} OVRLipSync 1.16.0 + Unity 2017.3.1p4 - YouTube <https://www.youtube.com/watch?v=0JwNNJuIBXw>

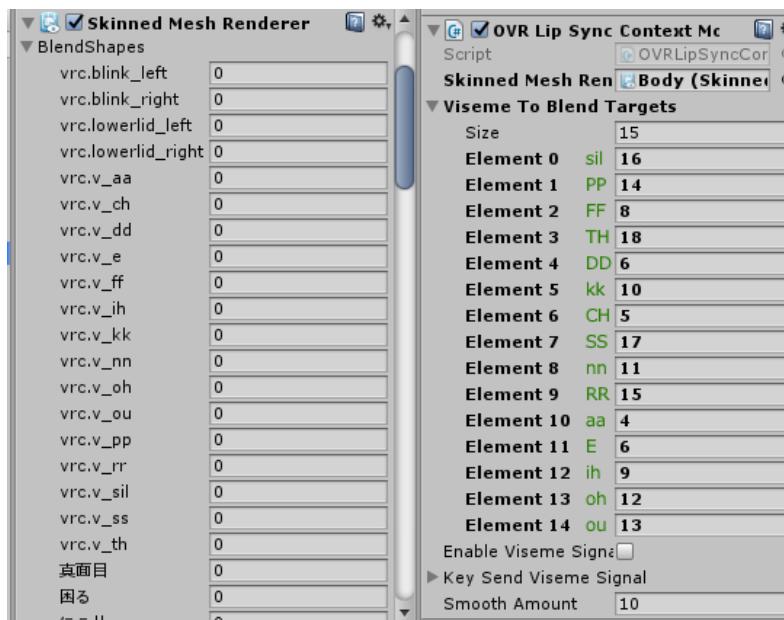
定します。

- Viseme To Blend Targets の配列に、15 種類の口唇の形状にあわせた、モデルのモーフの index を設定します。

15 種類の口唇の形状 (Viseme、ビゼーム) の仕様については、OVRLipSync 公式ドキュメントの「Overview^{*21}」に記載があります。詳細はリンクが貼られた「Viseme MPEG-4 Standard」の PDF 内に、顔写真・対応する英単語とセットで一覧表が掲載されていますので、そちらで確認する必要があります。

大雑把には「Element 0」から順番に、「sil（無音、口を結ぶ）, PP, FF, TH, DD, kk, CH, SS, nn, RR（このあたり無清音、"無音"か"う"を選択）, aa（あ）, E（え）, ih（い）, oh（お）, and ou（う）」のように設定します。

リポジトリに同梱している「youten-yume2」モデルは Blender での変換時に VRChat 仕様（おそらく OVRLipSync 仕様と同義）のモーフ設定がされていますので、そのまま Viseme 名に対応した index を設定します（図 3.11）。



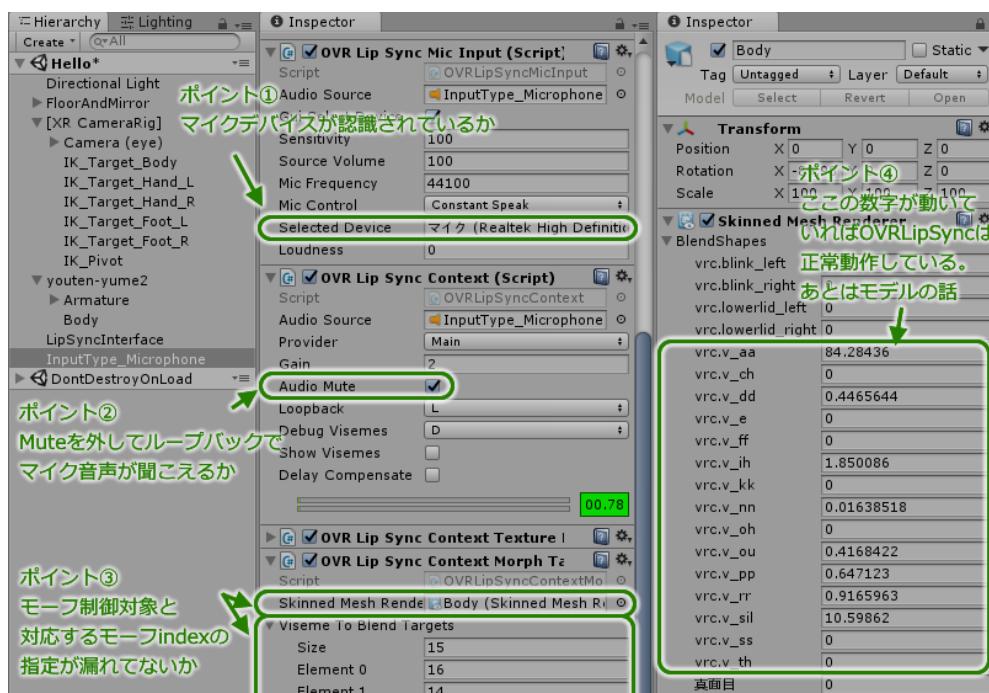
▲図 3.11 モデルのモーフと 15 種類の Viseme 設定の対応

^{*21} Overview <https://developer.oculus.com/documentation/audiosdk/latest/concepts/audio-ovrlipsync-overview/>

リップシンクがうまくいかない際の確認

OVRLipSync が動かない際には、実行中に以下のポイントを確認してみてください（図 3.12）。

- 「OVRLipSyncMicInput」でマイクデバイスが正しく認識されているか。複数マイクの際には選択するなどが必要です。
- 「OVRLipSyncContext」の Audio Mute チェックを外して、ループバック音声がきちんと聞こえるか。PC 側の録音デバイスの設定を見直してください。
- 「OVRLipSyncContextMorphTarget」に必要な設定が行われているか。
- ターゲットの「Skinned Mesh Renderer」でモーフ制御が行われている様子が見られるか。0 から 100 までの数字がきちんと変化しているのであれば、BlendShape が対応しているかどうかのモデル側の話になります。



▲図 3.12 OVRLipSync の動作確認ポイント

3.3.5 自動まばたき

キャラクターが目を開けっぱなしで無表情だと、お面のように感じられて、かわいくありません。自動でまばたきをするようにしましょう。

- ・「youten-yume2」に、「YVTuber」>「AutoBlink」スクリプトをアタッチします。

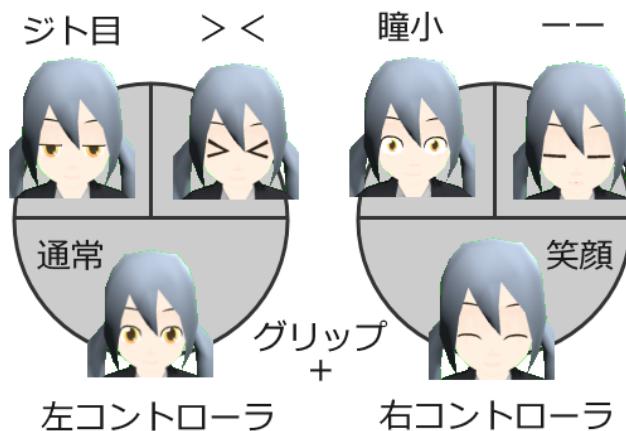
dksjal さんのスクリプト^{*22}をモデルに合わせて変更しています。他のモデルで使う際にはまばたきモーフの index と対象の Skinned Mesh Renderer を変更してください。

3.3.6 コントローラによる表情の変更

まばたきだけでもそれなりに人間っぽさが出ますが、表情をコントローラで制御できるようにしてみましょう。

- ・ Hierarchy より、「youten-yume2」に、「YVTuber」>「MorphController」スクリプトをアタッチします。

モデルにあわせたモーフの処理については「Morph」スクリプトを、コントローラ入力の取得については「XRVive」スクリプトを参照願います。



▲図 3.13 コントローラによるモーフ制御

^{*22}ユニティちゃんの瞬きを滑らかにする <https://gist.github.com/dskjal/17034ad3b97bc6596248881f02caaafe>

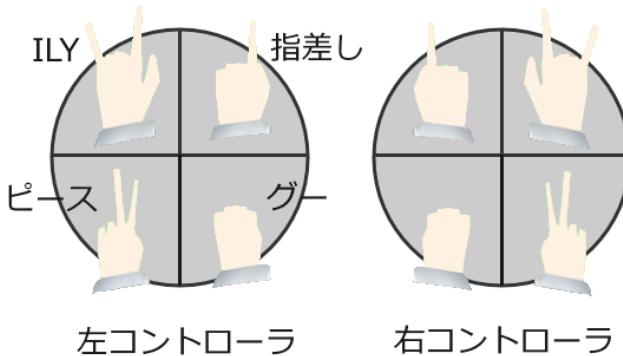
コントローラの後ろのトリガを押しながらトラックパッドを押し込むことでモーフを変化させることができます（図 3.13）。ちょっといじってみるとすぐ気づくと思いますが、今の実装はあまりよいものではありません。

そもそもその話として、顔の表情を手で操作するということが直感的ではありません。また、Vive コントローラのトラックパッドに少々癖があり、タップしてからスライドするとやっとタッチイベントが発生するようなことがあります。ジト目や笑顔目のようなふんだんに使っても自然であるモーフと、「><」目や瞳がキュツと小さくなるようなマンガ的な表現の特別なモーフを同じ操作体系に組み込んでしまっているため、誤爆しやすく、その際には違和感が強く出てしまいます。

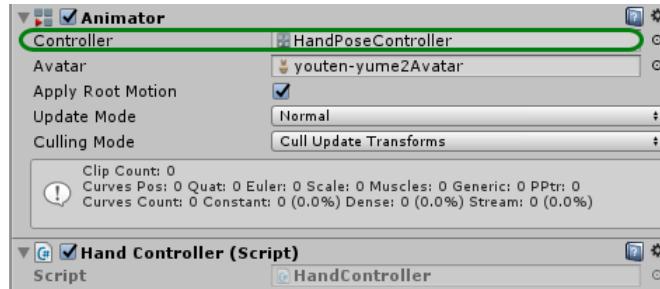
前章で紹介したアイトラッキング技術と、個人の目の大きさや笑顔の個人差をうまく吸収できるような機械学習ベースの仕組みに、マンガ的な大げさな、効果的な演出が重ねられるようになるまではこのモーフ制御というのは難しいジャンルなのかな、と思います。

3.3.7 コントローラによる手のポーズ制御

続けて、手のポーズをコントローラで制御できるようにしてみましょう。



▲図 3.14 手のポーズ。左右で鏡になる操作系になっている。



▲図 3.15 Animator Controller の設定変更

- Hierarchy 中の「youten-yume2」を選択し、「Animator」コンポーネントの「Controller」を「HandPoses」>「HandPosesController」を設定します。(図 3.15)
- Hierarchy 中の「youten-yume2」に、「YVTuber」>「HandController」スクリプトをアタッチします。

コントローラのトラックパッドをタッチすることで、手のポーズを変化させることができます(図 3.14)。色々な手の形を詰め込んだアニメーションパックは@m2wasabiさんの「HandPose 0.2.0」を使わせていただいています。

3.3.8 ビルド

ビルドについては、実のところ必須ではありません。結局のところ録画目的でも、配信目的でも、ウィンドウキャプチャが主流ですので、Unity エディタ上での再生のみでも問題ないケースがあります。

主なビルドのメリットはパフォーマンスの観点で優れていることと、配布がしやすくなることです。逆にデメリットは Unity エディタのように、実行中に Hierarchy やパラメータの操作はできないことです。



▲図 3.16 ビルドした上でウィンドウモードで起動した YV Tuber

ビルドして実行後、画面解像度を選択して起動し、エディタ上での再生と同じように各機能が動くことを確認します（図 3.16）。

3.4 撮影と編集

前節までの手順で、ビルドした exe にて VTuber らしき画が出せるようになりましたので、撮影と編集を行なって、動画を作ってみましょう。

3.4.1 OBS (Open Broadcaster Software) Studio での撮影

デスクトップ、特定のウィンドウ、フルスクリーンのゲームなど多様な画面キャプチャ機能を持ち、録画・エンコード・配信を一気に引き受ける、OBS (Open Broadcaster Software) Studio^{*23}というフリーソフトがあります。

Twitch や YouTubeLive 他、各配信サービスに対応しており、クロマキーなどのフィルタエフェクトを含む優れたレイヤー機能があり、大変人気のソフトです。そのため、インターネットで検索すると初心者向けから上級者向けまでたくさんの情報が見つかりますので、詳しい使い方はそちらを参照願います。

筆者が一つだけつまづいたのは一番最初のインストーラのダウンロードで、配布サーバがあまり強くないのかダウンロードできないケースがよくある模様です。BitTorrent で別経路からダウンロードしてうまくいきました。

^{*23} Open Broadcaster Software | ホーム <https://obsproject.com/ja>



▲図 3.17 OBS (Open Broadcaster Software) Studio

YVTuber を起動した状態で OBS を起動し、画面下部のソースから対象ウィンドウと切り抜き範囲などを適当に設定すると OBS 側で画が見えるようになります(図 3.17)。「録画開始」ボタンを押すと、録画が出力されはじめます。

後で編集することを想定している際には、劣化しない AVI 形式の動画が望ましいです。「UtVideo^{*24}」という可逆圧縮フォーマットのコーデックがありまして、無圧縮(=連続 BMP)の AVI とほぼ同等に扱えながら、ファイルサイズを大幅に削減できますのでオススメです。

3.4.2 AviUtl と拡張編集プラグインでの字幕つけ

それでは、動画編集を行なっていきましょう。テンポが良く、情報密度の高い動画を作るには、やはり字幕芸がてっとり早いです。

画像(文字)は音声と比較して情報を取得するのに必要な時間が短く、閲覧のコストを下げるよいものです。テレビのバラエティで当たり前のように浸透し、niconico・YouTuberなどの文化にてさらに映像ごと無音を詰めるスタイルも確立され、多くのバー

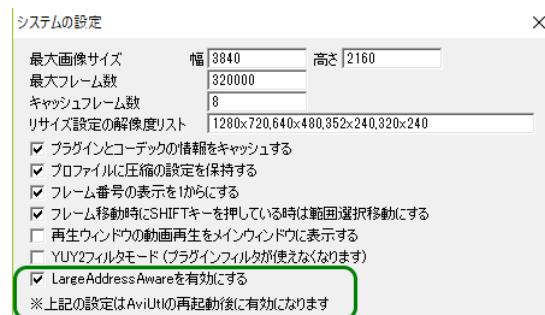
^{*24} 或るプログラマの一生 » Ut Video Codec Suite <http://umezawa.dyndns.info/wordpress/?cat=28>

チャル YouTuber の方々も、先駆者のキズナアイさんに倣って同様のスタイルを探っています。



▲図 3.18 AviUtl メイン画面

動画編集ソフトについては、キャプチャ・配信の OBS と比較すると選択肢が多く、良いものが複数ある印象です。「無料ながら高機能」「プラグイン形式での機能拡張と、その情報の手に入りやすさ」で一世代を築いた^{*25}「AviUtl^{*26}」が今でも現役なようで、これでよさそうです。



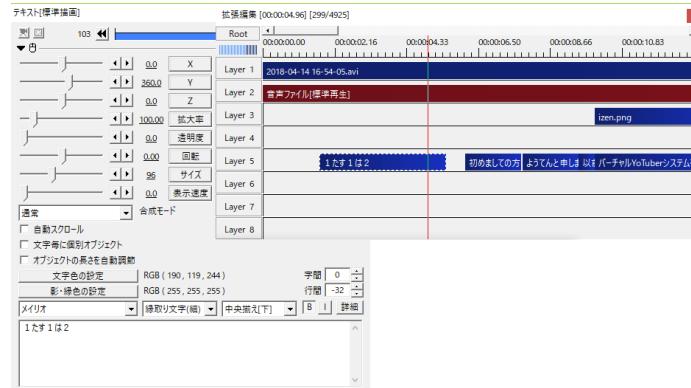
▲図 3.19 LargeAddressAware 設定による利用可能なメモリサイズ増加設定

AviUtl は残念ながら 32bit アプリのため、扱えるメモリサイズに限界があります。デ

^{*25} RO の GvG 動画を zoome に、FEZ の部隊訓練動画を stage6 に、MMD 動画を niconico に…

^{*26} AviUtl のお部屋 <http://spring-fragrance.mints.ne.jp/aviutl/>

フォルト 2GB の上限を 4GB まで引き上げる LargeAddressAware の有効化は最初にやつておいたほうがいいでしょう。根本的には解決していない問題のため、大きなプロジェクトを扱う際には工程やシーンの分割などの泥臭い対応が必要になります。



▲図 3.20 拡張編集のタイムラインとテキスト

拡張編集のタイムラインに AVI を読み込んで、ゴリゴリと字幕をつけていきます。タイムラインで「メディアオブジェクトの追加」から「テキスト」を選択してフォントサイズと縁取り設定をすると、それっぽい字幕をつけられます（図 3.20）。

実際に YVTuber システムで動画を撮って、OBS で撮影し、AviUtl で字幕をつけたサンプル動画を YouTube にアップロードしています^{*27}。エンコードには「x264guiEx^{*28}」を使っています。

こちらの約 1 分半の尺の動画で、撮影から YouTube への投稿完了まで 1 時間半程度かかりました。

コツというか、実際にやってみた感想なのですが、喋る内容の脚本を予め書いた上で、字幕や画像をどのように埋めるのかを予め想定しながら読み上げ練習をしておくと、録画の成功率が上がります。いわゆる「絵コンテの作成」や「読み合わせ」にあたるものです。

また、最初は 1 分程度の短い動画からチャレンジすることをオススメします^{*29}。実際に撮影する前に練っておくことで脱線を防ぐことができ、脚本のテキストがそのまま字幕に使えるようになり、動画の編集効率がよくなります。

^{*27} [YVTuber] VTuber システムを作った話の同人誌【サンプル動画】 - YouTube <https://www.youtube.com/watch?v=xFWbyAcOVKO>

^{*28} rigaya の日記兼メモ帳 x264guiEx 1.xx/2.xx 導入 <https://rigaya34589.blog.fc2.com/blog-entry-139.html>

^{*29} いきなり 5 分超えで収録したところ、編集に時間がかかりすぎで泣きました。本職の人マジ尊敬します…

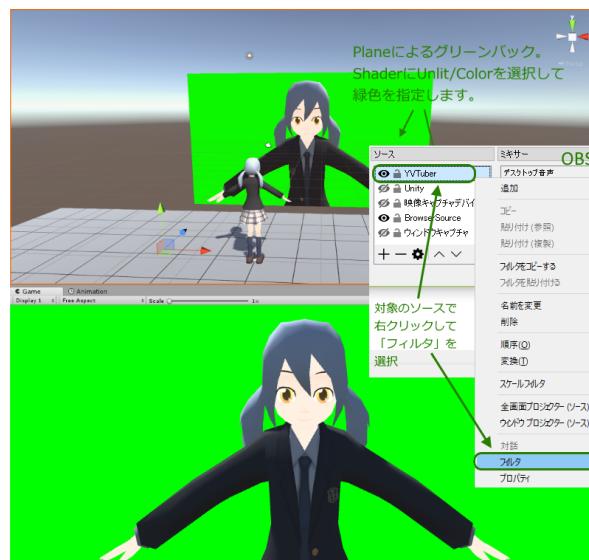
3.5 その他の機能の実装例

基本編は以上ということで、その他の機能の実装例を紹介します。

3.5.1 グリーンバックによるアバターの切り抜き

グリーンバックによるアバターカットも簡単です。

Unity 側でグリーンバックの Plane を準備して、OBS 側でクロマキー設定をしましょう。手順は次のとおりです。



▲図 3.21 Unity にてアバターの後ろにグリーンバックを設定し、OBS にてソースにエフェクトフィルタを追加

- Unity で Hierarchy タブにて、「Create」>「3D Object」>「Plane」で平面を用意します。"GreenBack"あたりの名前に変えておきましょう。
- Project タブにて、「Create」>「Material」で新規 Material を作成し、名前は"green"にします。Inspector タブから、「Shader」を「Unlit/Color」に変更し、「Main Color」を緑 (RGB = 0,255,0,255) に設定します。
- 作成した Plane に Material を設定し、適切な位置に回転して移動させてください (図 3.21)。

- 続けて OBS で、アバターの映っている YVTuber ウィンドウを右クリックし、フィルタを選択します（図 3.21）。
- 新しく開いたフィルタ設定ウィンドウにて、左下の「+」からクロマキーを追加します。グリーンバックと書きましたが、青やマゼンダで切り抜くこともできます。



▲図 3.22 クロマキー合成後の画

このように、だいたいどんなシーンでも自由に作ることができる Unity と、サイズ調整・配置・レイヤー制御・クロマキーなどのフィルタが揃った OBS を組み合わせると、録画・配信問わず自由度の高い画を作り出すことができます（図 3.22）。

これ以降については、Unity や動画編集の技術としてたくさんの情報がインターネットや書籍で手に入りますので、そちらを参照願います。

3.5.2 UnityCam でカメラ出力

続いて、ウィンドウへの出力ではなく、カメラデバイスとしての画面出力機能を追加してみましょう。これができると、Hangout などのカメラデバイスが選択できるテレビ電話アプリで、バーチャルアバターを使うことができます。

「UnityCam^{*30}」を使うと、仮想のカメラデバイスに対して Unity 内のカメラ映像を流し込むことができます。

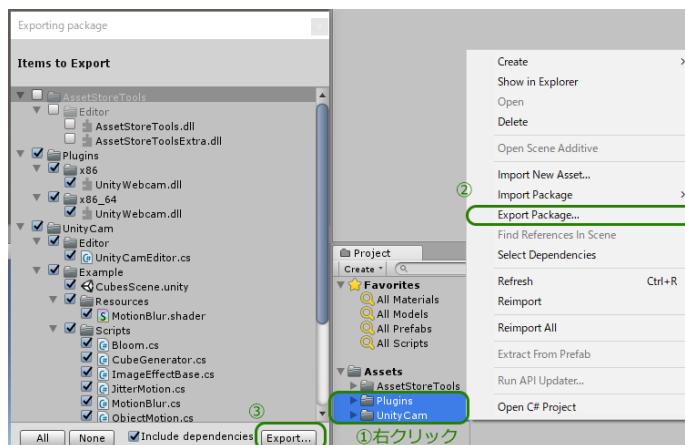
仮想カメラデバイスのセットアップ

「UnityCam/RunMe First」配下に「UnityCameraService.dll」という仮想カメラデバイスがありますので、「register.bat」を実行して登録しましょう。カメラドライバはインストールすれば Windows システム全体で共通して利用できるため、このフォルダは Unity プロジェクト配下ではなく、どこか PC 上の別のパスに配置の上 register した方が良いと思われます。

必要なファイル一式を **export** して目的のプロジェクトで **import**

UnityCam を本書で作成した VTuber システムのシーンに適用してみましょう。

次のとおりの手順となります。



▲図 3.23 UnityCam プロジェクトから必要なファイル一式を export

- 「UnityCam」リポジトリ一式をダウンロードし、「RunMe First」配下の dll 登録バッチファイルを実行します。
- 「UnityCam/UnitySample」プロジェクトを開き、「UnityCam」と「Plugins」を右クリックして「Export Package...」を選択、依存関係含め一式が選択されているのを確認して、unitypackage として「Export」します（図 3.23）。

^{*30} mrayy/UnityCam: Unity3D Virtual webcam plugin, streams unity viewport contents to other applications as virtual camera <https://github.com/mrayy/UnityCam>

- 適用する YVTuber プロジェクトに戻って、export した unitypackage を import します。
- 出力対象のカメラ、ここでは Hierarchy より「FloorAndMirror」>「Camera_Movie」に「UnityCam」>「Scripts」>「UnityCam」スクリプトをアタッチします。



▲図 3.24 Hangout プレビューでの UnityCam による YVTuber の出力

Unity を実行すると、仮想カメラデバイスから UnityCam で流した画が表示されているはずです。カメラ画像のプレビューなどが確認できるアプリを動かしてみましょう。他の Web カメラと同様のしくみで動いていますので、ブラウザで動くテレビ電話アプリである Hangout でもこのようにバーチャルアバターを使うことができます。

UnityCam のデメリットと OBS-VirtualCam

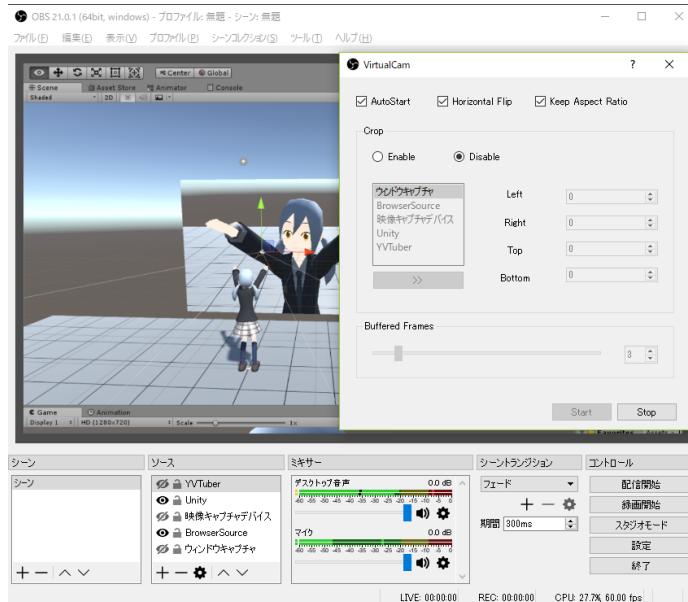
カメラが扱えるアプリであればアバターの画が取れるようになり、汎用性が上がる UnityCam の導入ですが、どうも解像度と FPS の上限が dll 側で決められているようです (*³¹)。また、それなりに負荷が高いことが知られており、とくに録画目的では OBS などのウィンドウキャプチャの方が向いていると言われています。

では、Unity 側ではなく、OBS 側で合成した画像を仮想カメラデバイスに流す方法がないのかと調べてみたところ、「OBS-VirtualCam^{*32}」というそのまんまの名前のプラグ

*³¹ <https://github.com/mratty/UnityCam/blob/master/UnityWebcam/UnityCamService/CaptureSource.cpp>

*³² OBS-VirtualCam | Open Broadcaster Software <https://obsproject.com/forum/resources/obs-virtualcam.539/>

インがありました。



▲図 3.25 OBS-VirtualCam

インストールして、OBS のメニュー「ツール」>「VirtualCam」で動作させることができます。間に挟むアプリが増えるので手間は増えてしまいますが、OBS 側のレイヤー合成結果ができることが最大のメリットといえるでしょう。

結論としてはどちらにもメリットがあるため、用途にあわせて使い分けるのがよさそうです。

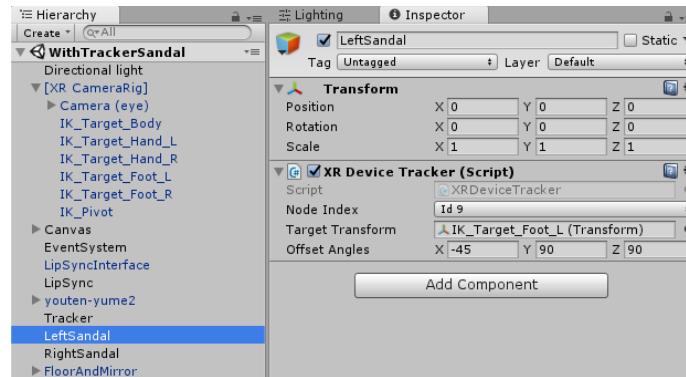
3.5.3 Tracker 対応と両足のトラッキング

XR Mecanim IK Plus には HTC Vive コントローラの Grip ボタンによる手と足を同期させて両方動かす機能があるのであるのですが、その代わりに HTC Vive Tracker の座標を設定してやれば、両足も動かせるのでは、と試してみました。

作りかけのサンダル

「YVTuber」直下に「WithTrackerSandal」という作りかけ（すいません）のシーンがありますのでそちらを参照願います。高さやスクリーンの座標調整が不十分であったり、Unity エディタ内で起動後に Tracker の id を手動で設定してやる必要がありますが、一

応最低限は動作します。



▲図 3.26 XRDeviceTracker スクリプトとアタッチ・設定例

Hierarchy 中にそれぞれ「裸の Tracker」・「Tracker を取り付けたサンダル左足」・「Tracker を取り付けたサンダル右足」に対応する、「Tracker」・「LeftSandal」・「RightSandal」というオブジェクトを配置しています。ここにアタッチされた「YVTuber」>「Script」>「XRDeviceTracker」スクリプトが API `InputTracking.GetNodeStates(List<XR.XRNodeState>)`によって Tracker の現実空間上の座標を取得し、対象の Transform へ座標を転送します（図 3.26）。

XR API とトラッキングモードの注意

`InputTracking.GetNodeStates(List<XR.XRNodeState>)` は、原点位置からの相対座標を返すため、本章の頭の方で述べたトラッキングモードがルームスケールでない際には正しく動きません。きちんと対応するのであれば、立位モードの際には Recenter 操作のタイミングで、リセットした座標を差し引く必要があると思われます。

本プロジェクトをベースに Tracker を扱う際には、「[XR CameraRig]」にアタッチされた「XR_Basic_Function_CS」中の「Tracing Space Type」を「Room Scale」に設定した上でこれ試してみてください。

3.6 まとめ

本誌のメインであるこの章ですが、「Hello, VTuber World!」の副題の通り、あくまで Hello World として、とくに難しいことはしておりません。ここから先は、他の VTuber の方々が実施されているように、世界観を作り込んでいく作業が丸々と残っています。

モデル・音声・脚本などのキャラクターを構成する要素は仮のものですし、Asset Store の豊富なアセットは背景、小物、エフェクト含め一切使っておらず、いわゆる「演出」がゼロの状態です。完成品としての作品やサービスを創り上げた経験がほとんどない筆者の現時点での限界でもありますし、今後はそのあたりをきちんと盛って、きちんと「VTuber」になれたら面白いのでがんばってみたいな、と考えています。

あとがき

バーチャル YouTuber（と VRChat）の昨今の流れ、Oculus Rift DK2 以来の VR 開発者増加ブームを感じています。

現時点での「VR」はビジネス観点ではゲーム・エンタメ用途でなんとか立ち上がったところで、「AI」や「ブロックチェーン」あたりと比較してメインストリームにはなれないな気配です。これは、ユーザが「VR」で生活を便利にするためには新しいハードウェアが普及する必要があるのですが、コモディティ化が不十分で、サイズ・性能・価格、様々な観点でもう一段階ブレイクスルーが必要で、「ギーク向けおもちゃ」を超えられないのです。（Vive Focus/Oculus GO/Mirage Solo はそれなりにそのあたりを改善してくれると思いますが。）

そんなふんい気の中で登場した「バーチャル YouTuber」というものは、VR 技術を使ってコンテンツをつくりますが、それを消費するユーザは必ずしも VR デバイスを必要としないのです。しかも動画という PC/スマホで閲覧できる入り口から、実際に VR デバイスを使った体験まで連続性があるのです。これは画期的！

さらに言うと、その「作る側が使う VR 技術」も現時点で揃ってるもので（赤外線や磁気などと戦いながら神チューニングをする必要がありますが）十分なのです。なんてこった、海の向こうのデジタル本国側の研究者やエコシステムの成果を待たなくていい！ 全力ダッシュだ！

「ボカラ以来のビジネスまで地続きの双方向 UGC 文化」「新しい芸能ビジネスの一分野」としてあっという間に収束して、そこに使われる VR 技術の守備範囲としてはすみやかに「あたりまえ」に溶けるんじゃないかな、と思っています。

いやほんと無職でよかった、流れ速すぎる。

Twitter: @youten_redo

Web: ReDo -Refrigerator Door- れいぞうこのドア <http://greety.sakura.ne.jp/redo/>

VR コンテンツのつくりかた 3 Hello, VTuber World!

2018 年 4 月 22 日 技術書典 4 版 v0.9.0

著 者 youten

(C) 2018 ReDo れいぞうこのドア