

3 D モデルにアニメーションを付ける



Vroidでアバターを作る

サンプルモデルを選択

アバターを作成する Vroid https://hub.vroid.com/

モデルの修正をする場合は下記を参照しましょう(公式) https://vroid.pixiv.help/hc/ja/articles/4405597663385



VRMエクスポート

VRMにエクスポートを開始します



フォーマットの設定

利用するシステムで制限がある場合がありますので設定を変更します。

 ✓ VRoid Studio 1.22.0 - 名称未設定 ★ VRMエクスポート 		Clusterの場合 ①ポリゴンの数が32,000以下であること ②マテリアル数が8であること ③ボーンの数が128以下であること ③テクスチャアトラスを2048×2048			
		エクスボート情報 ボリゴン数:32847 マテリアル数:6 ボーン数:113	 > ボリゴンの削減 > マテリアルの削減 マテリアル数 8 テクスチャアトラス解像度 2048 × 2048 > ボーンの削減 髪(各グループの最低数1) <u>エクスポート</u>		

エクスポートデータ修正

再配布・改変に関する許諾範囲やアバターの人格に関する許諾範囲を それぞれのシステムに合わせて情報を修正しましょう



Unity Assetsの入手 VRoid → Unity → Animation

Unityを起動します

Unity プロジェクトを作成します。 VRoidやBlenderで作成した人型3Dモデルを取り込む為にUnityに、UniVRMをインストー

ルします。

3Dモデルが移動する為のステージを作成します。



UnityとUnity AssetsのID登録

モデルは、BlenderやUnityで作成しますが、Unity Assets(アセット)を利用する事ができます Unityと同じユーザーIDを利用する事で、Unityと連動して使用する事ができます



Assets

RPG用パック Simple Low Poly Nature Pack https://assetstore.unity.com/packages/3d/environments/landscapes/simple-low-poly-nature-pack-157552 モーションキット Dance Animations FREE https://assetstore.unity.com/packages/3d/animations/dance-animations-free-161313 **Basic Motions FREE** https://assetstore.unity.com/packages/3d/animations/basic-motions-free-154271 Unity $O \land y - y - + y +$ 一人称 Starter Assets - First Person Character Controller | URP https://assetstore.unity.com/packages/essentials/starter-assets-first-person-character-controller-urp-196525 三人称 Starter Assets - Third Person Character Controller https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-196526 地面用テクスチャ Terrain Textures Pack Free https://assetstore.unity.com/account/assets **Outdoor Ground Textures** https://assetstore.unity.com/packages/2d/textures-materials/floors/outdoor-ground-textures-12555

制服ユニティちゃん <u>https://unity-chan.com/download/releaseNote.php?id=TPK-Hmnd-Kohaku_B</u>

Simple Low Poly Nature Packを使う

https://assetstore.unity.com/packages/3d/environments/landscapes/simple-low-poly-nature-pack-157552



Simple Low Poly Nature Pack

https://assetstore.unity.com/packages/3d/environments/landscapes/simple-low-poly-nature-pack-157552



Basic Motions FREEを使う

Basic Motions FREE

https://assetstore.unity.com/packages/3d/animations/basic-motions-free-154271



Animation Assets

Unity Assetsからモーションのアセットをダウンロードして3Dモデルに割り当て動かします。

Dance Animations FREE

https://assetstore.unity.com/packages/3d/animations/dance-animations-free-161313

Basic Motions FREE

https://assetstore.unity.com/packages/3d/animations/basic-motions-free-154271



Unityでシーンを作る

シーンを取り込む



UniVRMインポート

UniVRM ダウンロード

Unity for VRMダウンロード

https://github.com/vrm-c/UniVRM/releases

VRM 0.x Import/Export がVRoid対応になります。



UniVRMインポート



x

NEW

Import Unity Package

UniVRM-0.110.0_3f7d

アバターをインポート

3 D モデルをインポート

VRMフォーマットのファイルをAssetsの中にドラック&ドロップするとインポートされます。 インポートされると、アニメーション等が付随したPrefubが作成されます。 作成された、PrefubをHierarchyにドラック&ドロップします



MoveControllerの追加



MoveControllerを作成して、アバターの操作を行います。 WSADキーやアローキーでアバターの移動を行います。

Assets にScriptsフォルダーを作成
 3DObject→Create→C# Scriptでスクリプトを作成
 名前を"CameraScript"とします
 CameScript を Hierarchy内の MainCamera ヘドラッグ&ドロップします
 スクリプトのターゲットを【モデル名】にします



MoveController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

}

```
public class MoveController : MonoBehaviour {
    private Animator animator;
     // Start is called before the first frame update
    void Start() {
    animator = GetComponent<Animator>();
      }
     // Update is called once per frame
     void Update() {
          if (Input.GetKey(KeyCode.UpArrow) || Input.GetKey(KeyCode.W)) {
               animator.SetBool("Walk", true);
               transform.position += transform.forward * 0.01f;
          } else {
          animator.SetBool("Walk", false);
          if (Input.GetKey(KeyCode.DownArrow) || Input.GetKey(KeyCode.S)) {
               animator.SetBool("Back", true);
               transform.position -= transform.forward * 0.01f;
          } else {
               animator.SetBool("Back", false);
          if (Input.GetKey(KeyCode.RightArrow) || Input.GetKey(KeyCode.D)) {
               transform.Rotate(0.0f, 1.0f, 0.0f);
          if (Input.GetKey(KeyCode.LeftArrow) || Input.GetKey(KeyCode.A)) {
               transform.Rotate(0.0f, -1.0f, 0.0f);
          }
     }
```

Scriptをアタッチ



カメラ追従

アバター ヘカメラの追従

Assets にScriptsフォルダーを作成
 3DObject→Create→C# Scriptでスクリプトを作成
 A前を"CameraScript"とします
 CameScript を Hierarchy内の MainCamera ヘドラッグ&ドロップします
 スクリプトのターゲットを【モデル名]にします



Targetをモデルにする

カメラ移動用のScript「CameraScript.cs」を作成
 2)Cameraにスクリプトをアタッチ
 3)ターゲットをアバターに指定



アバター ヘカメラの追従

using System. Collections; using System. Collections. Generic; using UnityEngine;

}

```
public class CameraScript : MonoBehaviour
{
    public GameObject target;
    private Vector3 distance;
    // Start is called before the first frame update
    void Start()
    {
        distance = transform.position - target.transform.position;
    }
    // Update is called once per frame
    void Update()
    {
        transform.position = target.transform.position + distance;
    }
}
```

モーション追加

Animatorを選択する

Hierarchy からアバターを選択
 Animator の Controller の "◎"をクリック
 ウィンドウが開く それぞれのアニメーターをダブルクリック



🔻 🏱 🗹 Animator	0:	‡ i (
Controller	Non Clorpinte Arionator Gup trol	ler
Avatar	☆ vrmavatar	U
Apply Root Motion		
Update Mode	Normal	T
Culling Mode	Always Animate	T
Select RuntimeAnimatorContr	roller	
Select RuntimeAnimatorContr A Assets Scene	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Run	roller	
Select RuntimeAnimatorContr Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Spine BasicMotions@Spine BasicMotions@Strafe DeviceMotions@Taille	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Run BasicMotions@Spin BasicMotions@Spin BasicMotions@Strafe BasicMotions@Talk BasicMotions@Talk BasicMotions@Walk	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Sum BasicMotions@Strafe BasicMotions@Strafe BasicMotions@Talk BasicMotions@Talk DancePack@Dance01 DancePack@Dance02	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Run BasicMotions@Sprine BasicMotions@Strafe BasicMotions@Talk BasicMotions@Talk BasicMotions@Talk BasicMotions@Valk DancePack@Dance02 DancePack@Dance03 DancePack@Dance04	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Spins BasicMotions@Spins BasicMotions@Strafe BasicMotions@Talk BasicMotions@Talk BasicMotions@Talk DancePack@Dance01 DancePack@Dance02 DancePack@Dance03 DancePack@Dance04 DancePack@Dance05 DancePack@Dance06 DancePack@Dance06	roller	
Select RuntimeAnimatorContr Assets Scene BasicMotions@Jump BasicMotions@Jump BasicMotions@Sprint BasicMotions@Strafe BasicMotions@Talk BasicMotions@Talk BasicMotions@Talk DancePack@Dance01 DancePack@Dance02 DancePack@Dance05 DancePack@Dance05 DancePack@Dance05 DancePack@Dance05	roller	

モーションの切り替え

Animetor Contolerの作成

Object Winodw > Create > Animetor Contoler



床との緩衝

床との干渉

アバターにRigidbody と Capsule Collider の値を下図のように設定する

Rigidbody:物理法則を与える Capsule Collider:カプセル型の干渉の形状

	🔻 🔁 Rigidbody		9 ∓≛ €
	Mass	1	
	Drag	0	
 Freeze Rotation 	Angular Drag	0.05	
X.Y.Z それぞれチェックを入れる	Use Gravity	~	
,,	Is Kinematic		
	Interpolate	None	-
Contor	Collision Detection	Discrete	-
X=U, Y=U. 85, X=U	Freeze Position	X Y Z	
• Height	Freeze Rotation	🖌 X 🖌 Y 🗹 Z	
1. /5	▶ Info		
	🔻 🔩 🗹 Capsule Collider		9 ∓ :
	Edit Collider	ふ	
	ls Trigger		
	Material	None (Physic Material)	\odot
	Center	X 0 Y 0.85 Z 0)
	Radius	0.5	
	Height	1.75	
	Direction	Y-Axis	•
完成





Unity 基礎

新規プロジェクト





Hierarchy (ヒエラルキー) :シーンに使うパーツを入れる場所 Project (プロジェクト) :プロジェクト全体の必要なパーツを入れておく場所 Inspector (インスペクター) :オブジェクトの属性を表示します。



シーン

 Main Cameraは ゲーム画面を映し出すカメラ
 Directional Lightは ゲーム内を照らす明かり





マウス 左 : オブジェクト選択 マウス 右 : 視点移動 マウス 中 : 画面移動 マウス ホイール:拡大・縮小



オブジェクトを置いてみる

ゲームオブジェクトを作成します。

Game Object \rightarrow 3D Object \rightarrow Cube

作成されたCuleをクリックします。

Inspector(インスペクター):3D Object のパラメータ



シーンとカメラ





オブジェクト編集

作成したCuleをクリックします。

Inspector(インスペクター)の3D Object のパラメータを変更する事で変更できます。



地面を作る

Game Object \rightarrow 3D Object \rightarrow Terrain

X: -500

Y: 0

Z: -500



オブジェクトを配置する

Cube X: 0 Y: 10 Z: 0



実行してみる





Add component \rightarrow 重力 (Sphere) \rightarrow Regidbody



色を付ける

マテリアル(Material)

1)Materialを保存するフォルダを作ります。
Projectタブの「Assets」フォルダを右クリックし、[Create] → [Folder]をクリックします。
2)Materialsフォルダの中に、新しくMaterialを作成する。
Materialsフォルダをダブルクリックして開きます。
Materialsフォルダの中で右クリック → [Create] → [Material] をクリックします。
3)InspectorのMain Mapsの色を青にします。
4)Materialの名前を変更します。
例:Material_Bule
4)青色のマテリアルをObjctに反映します。
Hierarchy上のCubeを選択してInspectorを開きます。
Blue_Materialをドラッグ&ドロップでAdd Component下の空間へ放り込みます。

色を付ける





VRMについて



UnityやBlenderで修正ができる3Dモデルを作成します。 アプリ毎に読み取れるファイル形式があります。 人型モデルは「VRM形式」を使用します。 参考:3Dフォーマット(一部)

VroidのサンプルアバターをVRMで出力します。



VRMについて

「VRM」はVRアプリケーション向けの人型3Dアバター(3Dモデル)データを扱うためのファイルフォーマットです。 gITF2.0をベースとしており、誰でも自由に利用することができます。

VRMは複数アプリケーション間で相互利用されることが想定される、VR時代の3Dアバターフォーマットです。たとえば 「ネットワークに相互接続されたアバターによるVRコミュニケーション」を考えた場合、 自分のアバターデータを「他の 人から自分の姿が見えるように」他のユーザへ送信する 、ということも行われることになります。 この場合、従来の考え方に基づく「再配布」規定だけでは対応できない場面が考えられます。そのため、VRMではモデル データ自体に対しての改変・再配布規定(Creative Commonsなどから設定可能)のほかにモデルデータを使用して「人格 を演じる」ことについての許諾規定を ファイルに設定できる など、VR時代を想定したフォーマットとなっています。

VRMは、Unity、VRChat、Cluster等に利用する事ができます。

アバターの人格に関する許諾範囲(Personation / Characterization Permission)

アバターの人格に関する許諾範囲(Personation / Characterization Permission) アバターに人格を与えることの許諾範囲(A person who can perform with this avatar) アバターを操作することはアバター作者にのみ許される(The avatar may only be manipulated by its creator)

明確に許可された人限定(Authorized only to explicitly specified person)

全員に許可(Authorized to anyone)

このアバターを用いて暴力表現を演じることの許可(Permission to perform violent acts with this avatar) 不許可(Disallow) 許可(Allow)

このアバターを用いて性的表現を演じることの許可(Permission to perform sexual acts with this avatar) 不許可(Disallow) 許可(Allow)

商用利用の許可(For commercial use) 不許可(Disallow) 許可(Allow)

その他のライセンス条件(Other License Url) 上記許諾条件以外のライセンス条件がある場合はそのライセンス文書へのURLを記述

再配布・改変に関する許諾範囲(Redistribution / Modifications License)

再配布・改変に関する許諾範囲(Redistribution / Modifications License) ライセンスタイプ(License Type) 再配布禁止(Redistribution Prohibited) 著作権放棄(CCO) Creative Commons CC BYライセンス(CC_BY) Creative Commons CC BY NCライセンス(CC_BY_NC) Creative Commons CC BY NCライセンス(CC_BY_NC) Creative Commons CC BY NC SAライセンス(CC_BY_NC_SA) Creative Commons CC BY NC SAライセンス(CC_BY_ND) Creative Commons CC BY NDライセンス(CC_BY_ND) Creative Commons CC BY NC NDライセンス(CC_BY_ND) その他(Other) その他ライセンス条件(Other License Url) 上記許諾条件以外のライセンス条件がある場合はそのライセンス文書へのURLを記述

参考:3Dフォーマット(一部)

ファイル形式	説明
. stl	3D System(3D システム)社によって開発されたファイル形式 3Dプリントをする上での標準形式となっており、多くのCADソフトウェアで読み込みや書き出しが 可能
.step .stp	ISO(国際標準化機構)により開発された3DCADデータのファイル形式 国際標準として普及が進んでおり、ほとんど全てのCADソフトウェアで対応している
. obj	Wavefront(ウェーブフロント)社が開発したファイル形式 表面材質やテクスチャ情報も保持できるがアニメーション情報は保持できない 多くの3Dモデリング、レンダリングソフトが対応している
. fbx	Autodeskに買収されたAlias(エイリアス)社がオープンソースとして開発したファイル形式 カラー情報やアニメーション情報も保持できるため、3DCGの標準フォーマットとして浸透している
.wrl	Web上で使用することを前提に設計されたファイル形式 カラーやアニメーション情報に加え、wrlファイルへのリンクURLも保持できる
.vrm	「VRM」はVRアプリケーション向けの人型3Dアバター(3Dモデル)データを扱うためのファイル フォーマットです。 glTF2.0をベースとしており、誰でも自由に利用することができます。 <u>https://vrm.dev/#</u>
. gb l	「gITF (GL Transmission Format)」とは、3Dモデルのファイル形式のひとつで、主にWebブラウ ザ上で動作するコンテンツを作る際に用いられるフォーマットです。 2000年にKhronos Groupによって開発・策定されました。 また、gITFをバイナリ形式に変換したものが「GLB」です。 palan3DではgITF/GLBをPC・スマートフォン共通の3Dモデル表示画面、AndroidのAR表示用に使用し ます。

エラー処理

Prefab (プレハブ)に変換されない場合

Vroid のVRMは、VRM 0.X なので VRM 0.x Import/Export をインストールする必要があります。 下記のチェックをいれると、 VRM 0.x →VRM 1.x に変換されます。



画面がピンクの時

https://qiita.com/kazuma_f/items/4d265c5615f1d441ffe9#%E8%A3%9C%E8%B6%B3

Edit > Render Pipeline > Universal Render Pipeline > Upgrade Project Materials to Universal RP Materials



Render Pipeline Converter	: 🗆 ×
Built-in to URP The Render Pipeline Converter performs the following tasks: * Converts project elements from the Built-in Render Pipeline to * Upgrades assets from earlier URP versions to the current URP	ORP. Version.
Built-in to URP	r project before proceeding.
Rendering Settings This converter will look at creating Universal Render Pipeline as for the set of	6 items assets and respective Renderer Assets a builtin renderer. – 0 ▲ 0 ❶ 0 ↓6 ↓ @
Quality Level 1: Low	√ 0
 Quality Level 2: Medium 	√ 0
 Quality Level 3: High 	√ 0
✓ Quality Level 4: Very High	√
Material Upgrade	
Initialize Convert	Convert Asets

Starter Assets - Third Person Character Controller



VRoid Studio ダウンロード

下記URLより、VRoid Studio ダウンロードします インストールは、手順に従ってください

https://vroid.com/studio



VRoid Studio インストール



UnityChan!ダウンロード

Unity Assets Storeからアセットをダウンロードします。 Window→PakageManager→PakageMyassets→【Unity-Chan! Model】 →Download→Import



Unity-Chanモーションの取り込み

Uniy AssetsでUnity-Chan! Modelを入手

Unity-Chan! Model

https://assetstore.unity.com/packages/3d/characters/unity-chan-model-18705



UnityChan!ダウンロード

Unity Assets Storeからアセットをダウンロードします。 Window→PakageManager→PakageMyassets→【Unity-Chan! Model】 →Download→Import



モーションスクリプト追加

1) Hierarchy からアバターを選択 2) アバターのInspector からAdd Component をクリック 3) ウィンドウが開く Scripts を選択 4) UnityChan 5) UnityChanControlScriptWithRigidBodyを選択	Image: Com Image: Com Image: Com Image: Com
	Audio UniHumanoid
NM CompleSense Woodows May Linux, Units 2013 2 2161 Research (NV11)	UnityChan
File Edit Assets GameObject Component UniGLTF VRM0 Window Help	UnityEngine.EventSystems
	© Q Layers ▼ La Layout UnityEngine.TestTools.TestRunner.Ca
+ → exampleScenet :	Mesh UnityEngine.Timeline
Om Main Camera Image: Comparison of the second	Tag Untagged V Layer I Miscellaneous VRI
C Root O Face S Rody C C C C C C C C C C C C	Transform Navigation VRI
⊕ booy ⊕ Hair ⊕ secondary □ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Rotation x 0 y 0 Physics 2D CultyChan
G Directional Light	Scale © X 1 Y 1 Physics # Auto Blink
	VRM Humanoid Descripti Playables # Camera Controller
	Rendering # Face Update
	Script Sc
	First Person Offset Tilemap # IK Ctrl Right Hand
	x 0 ¥ 0.06 Z # Random Wind
■ Project ■ Console	▶ 🖬 🗸 VRM Look At Head (Script 🛛 ≠ : 🗖 🗸 VRM Look At Head (Script 🔍 ≠ :
+ + Q 20 20 20 20 20 20 20 20 20 20 20 20 20	Add Component # Spring Bone
► Scenes III III III III III III III III III I	# Spring Collider
	# Spring Manager
Autio Autosink Camerato Paceupoa idec.nanger incurringen kandomivi springsone springcoll	# Third Person Camera
Editor ▶ Prefabs ■ Scenes # # #	# Unity Chan Control Script With Roid B
Exripts SpringMan ThirdPerso UnityChan	
GVRM VRMRhandere File unitychan_RUN00_R' has animation import warnings. See Import Messages in Animation Import Settings for more details.	

Uniy AssetsでUnity-Chan! Modelを入手

Unity-Chan! Model

https://assetstore.unity.com/packages/3d/characters/unity-chan-model-18705



Animator追加

- ① Hierarchy からアバターを選択
- ② Animator の Controller の "◎"をクリック
- ③ ウィンドウが開く UnityChanLocomotions を ダブルクリック



Select RuntimeAnimatorController

x
モーションスクリプト追加

1) Hierarchy からアバターを選択 2) アバターのInspector からAdd Component をクリック 3) ウィンドウが開く Scripts を選択 4) UnityChan 5) UnityChanControlScriptWithRigidBodyを選択	Image: Com Image: Com Image: Com Image: Com
	Audio UniHumanoid
NM CompleSense Woodows May Linux, Units 2013 2 2161 Research (NV11)	UnityChan
File Edit Assets GameObject Component UniGLTF VRM0 Window Help	UnityEngine.EventSystems
	© Q Layers ▼ La Layout UnityEngine.TestTools.TestRunner.Ca
+ → exampleScenet :	Mesh UnityEngine.Timeline
Om Main Camera Image: Comparison of the second	Tag Untagged V Layer I Miscellaneous VRI
C Root O Face S Rody C C C C C C C C C C C C	Transform Navigation VRI
⊕ booy ⊕ Hair ⊕ secondary □ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Rotation x 0 y 0 Physics 2D CultyChan
G Directional Light	Scale © X 1 Y 1 Physics # Auto Blink
	VRM Humanoid Descripti Playables # Camera Controller
	Rendering # Face Update
	Script Sc
	First Person Offset Tilemap # IK Ctrl Right Hand
	x 0 ¥ 0.06 Z # Random Wind
Project Console	▶ 🗄 🗸 VRM Look At Head (Script 🛛 ≠ : 🛱 VRM Look At Head (Script 🔍 ≠ :
+ + Q 20 20 20 20 20 20 20 20 20 20 20 20 20	Add Component # Spring Bone
Scenes UniGLTF III III III III III III III III III I	# Spring Collider
	# Spring Manager
Autio Autosink Camerato Paceupoa idec.nanger incurringen kandomivi springsone springcoll	# Third Person Camera
Editor ▶ Prefabs ■ Scenes # # #	# Unity Chan Control Script With Roid B
Exripts SpringMan ThirdPerso UnityChan	
GVRM VRMRhandere File unitychan_RUN00_R' has animation import warnings. See Import Messages in Animation Import Settings for more details.	

ゲームスタート

- ・▶(ゲーム実行)をクリック
- ・方向キーでアバターが動くようになりました
- 動かし終えたら、もう一度 ▶を選択し、プレイモードを終了



地形のアレンジ

地面にテクスチャ(模様)を付ける

 Unity Assets 入手 Terrain Textures Pack Free : <u>https://assetstore.unity.com/account/assets</u>
 Paint Terrain を クリック
 Paint Texture を 選択
 Edit Terrain Layers をクリックし、create Layer を選択
 Select Texture2D ウィンドウが開く ・GrassHillAlbedo をダブルクリック



地面に起伏を付ける

Paint Terrain をクリック
 Raise or Lower Terrain を選択
 ブラシタイプ、サイズを適当に調整
 Terrain 上でドラッグすると、ドラッグした部分が隆起します
 Shiftキーを押しながらドラッグすると逆に沈みます





地面に木を植える

・Inspector から Terrain を選択

Paint Trees を クリック
 Edit Trees をクリックし、Add Tree を選択
 ③"◎"をクリック
 ④Broadeaf_Mobile を選択
 ⑤ Add をクリック

	Add Tree				×
Ζ	Tree Brofab		None (Game Object)		
$\mathbf{\hat{o}}$	Bend Factor	0	None (Game Object)		
	Please as	sign a tree			
	s	elect GameO	bject		×
	٩				
		Assets 🍞 BlobSha	dowProjector		Ø≥10
Inspector		BlockPr	ototype04x04x04		
🕥 🗹 Terrain		Box	0.00,000,000		
Tag Untagge	d	BoxPile	.11		
▼ 👃 Transform		😭 BoxUnit	yChan vChan		
Docition	X -50(Brick			
Rotation	Add Tree	Bas Broanie s	at Deskinn		x
Scale					
	Tree Prefab		Broadleaf Mobil	<u>م</u>	0
V 💑 🗸 Terrain	NavMesh LOD	Index	Last	<u> </u>	
Paint Trees					
Click to paint trees.					
Hold shift and click					
Hold Ctrl and click to					Add
Trees					Add
No trees defined.					
Mass Place Tree		🗘 Edi	t Trees Refresh		
Settings					
Brush Size		•	Add Tree		
Tree Density			Edit Tree		
Tree Height	Random?	 ✓ 	Remove Tree		
Lock Width to Hei	ght 🖌		-	00	
Tree Width	Random?				

オリジナルのアバターをオリジナルのステージで動かせるようになりました!!



Animator追加

- ① Hierarchy からアバターを選択
- ② Animator の Controller の "◎"をクリック
- ③ ウィンドウが開く UnityChanLocomotions を ダブルクリック



Select RuntimeAnimatorController

x

③ モーションを付ける

① Hierarchy から アバターを選択

'≡ Hierarchy		а	:
++			
⊤ 🛠 Sam	pleScene*		:
Эм Эрі Эт	ain Camera rectional Light rrain		
🔺 🏠 Vi			>



≻ 🗸 Animator	Q 7		
Controller	None (Runtime Animator Concoller	0	
Avatar	☆ VrmAvatar	0	
Apply Root Motion			
Update Mode	Normal	T	
Culling Mode	Always Animate	T	
Clip Count: 0 Curves Pos: 0 Quat: 0 Curves Count: 0 Const (0.0%)	Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr tant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0		

③ ウィンドウが開く UnityChanLocomotions を ダブルクリック

Select Ru	ntimeAnimatorControl	ler	
ব			
Assets	Scene		95 1
Nor	ie		
🗾 🔁 Spla	ashScreen		
🔄 🔁 Unit	tyChanActionCheck		
🖬 Unit	tyChanARPose		
Ti Unit	tyChanLocomotions		

2 移動のスクリプト(命令)を付ける

① Hierarchy から アバターを選択



② アバターのInspector からAdd Component をクリック

-9-			Layer					
Inspector								
		🖈 Vrm	nAvatar					
Apply Root Mot								
		Norma	al					
		Alway		nate				
Clip Count Curves Po Curves Co (0.0%)	: 0 s: 0 Quat: 0 sunt: 0 Const	Euler: 0 tant: 0 () Scale: ((0.0%) D	0 Muscle ense: 0 i	is: 0 Gei (0.0%) S	neric: Strean	0 PPI 1: 0	
🔻 🗰 🖍 VRM BI	end Shape	Proxy	/ (Scrip	ot)		6		
Blend Shape Av	vatar	GBler	ndShar	be (Bler	nd Shai	be Av	atar	n) 💿
Enable wh								
🔻 🗰 🛛 VRM Fir	rst Person	(Scrip	ot)			6		
		VRN						
		,,J_B	ip_C_H	ead (Tr	ansfor	m)		
		X O		Y 0.06		ΖO		
Renderers								
🔻 🗰 🗹 VRM Lo	ok At Hea	d (Scr	ipt)			6) #	
		~						
Update Type		Updat	te					
		None	(Trans	form)				
		None	(Trans	form)				
Debug								
🔻 🗰 🖌 VRM Lo	ok At Bon	e Appl	lyer (S	cript)		6		
Draw Gizmo								
⊳ Left Eye								
▶ Right Eye								
Degree Mappi	ng er							
 Horizontal Inne Vertical Down 								
 Horizontal Oute Horizontal Inne Vertical Down Vertical Up 								
 Horizontal Oute Horizontal Inne Vertical Down Vertical Up Runtime 	r e Gitf Insta	ance (S	Script)			6		
 Horizontal Oute Horizontal Inne Vertical Down Vertical Up # Runtime Script 	r e Gitf Insta	ance (S	Script) ntimeGl			6		
 Horizontal Oute Horizontal Inne Vertical Down Vertical Up # Runtime Script 	e Gitf Insta	ince (S Run id Com	Script) ntimeGI nponen			6		
Horizontal Oute Horizontal Inne Vertical Down Vertical Up Runtime Script Vita	r e Gitf Insta Ac	ince (S Run id Com	Script) ntimeGl nponen			6		

③ ウィンドウが開く Scripts を選択

<	Component		
	Audio	=	
ŧ	Effects		
	Event		
	Layout		
>	Mesh		
3	Miscellaneous		
-	Navigation		
	Physics 2D		
	Physics		
	Playables		
	Rendering		
-	Scripts		
	Tilemap		5

④ UnityChan を選択



④ UnityChanControlScriptWithRigidBodyを選択

9	
	UnityChan
#	Auto Blink
#	Camera Controller
#	Face Update
#	Idle Changer
#	IK Ctrl Right Hand
#	Random Wind
#	Splash Screen
#	Spring Bone
#	Spring Collider
#	Spring Manager
#	Third Person Camera
#	Unity Chan Control Script With Rgid B

床との干渉

アバターのInspector から、Rigidbody と Capsule Collider の値を下図のように設定する

Freeze Rotation
 X, Y, Z それぞれチェックを入れる

 Center X=0, Y=0. 85, X=0
 Height 1. 75

🔻 😪 🛛 Ri	igidbody				Ø	ᅶ	:
Mass		1					
Drag		0					
Angular [Drag	0.05					
Use Grav	ity	~					
ls Kinema	atic						
Interpolat	te	None					•
Collision	Detection	Discrete					•
▼ Constrair Freeze Freeze ▶ Info	nts e Position e Rotation	×	z J z				
🔻 😓 🔽 c	apsule Collider				Ø	ᅶ	:
Edit Colli	der	ሌ					
Is Trigger							
Material		None (Physi	c Material)				\odot
Center		X O	Y 0.85	Z	0		
Radius		0.5					
Height		1.75					
Direction		Y-Axis					•

・アバターのInspector から、Rigidbody と Capsule Collider の値 を下図のように設定する

	- 😪	Rigidbody		9		Ξ
	Mass		1			
	Drag		0			
・Freeze Rotation には	Angul	lar Drag	0.05			
X.Y.7 それぞれ	Use G	Gravity	~			
チェックを入れる	ls Kin	ematic				
「エックを八礼る」	Interp	olate	None		_	-
	Collis	ion Detection	Discrete			•
	▼ Const	traints				
	Fre	eze Position				
	⊢re ⊾ Info	eeze Rotation	✓ X ✓ Y ✓ Z			
	P INIO					
	▽ 🐣	Capsule Collider		9	ᆤ	
	Edit C	Collider	ሌ			
	ls Tric	aar				
	Mater	rial	None (Physic Material)			$\overline{\circ}$
	Cente	er	X 0 Y 0.85	Z O		
	Radiu	IS	0.5			
	Heigh	nt	1.75			
	Direc	tion	Y-Axis			•