

FIMPOSSIBLE *CREATIONS*

RAGDOLL ANIMATOR 2

Helper Methods List

About Ragdoll Animator 2

- Ragdoll Animator 2 is a plugin which provides tools for **precisely adjusting ragdoll related components**, like **rigidbodies**, **colliders** and **configurable joints**. It provides tools for **detailed and easy control** over **physical body animation**.
- Plugin offers many ways to handle **ragdoll body behavior**, like animating physical bones **with currently played animation**, simulating **full body physics**, playing physical animation **when having attached body parts to the different joints**, adding **physical impacts** to the selected bones, **dragging** full ragdoll body by it's leg, and more.
- Ragdoll Animator 2 component is providing a **highly customized inspector window** (GUI) to help use it without confusion since there are a **lot of parameters to play with**.
- Package is providing many example scenes presenting different features, which can be **unpacked to project with** "Demo - Ragdoll Animator 2" unitypackage file.
- Ragdoll Animator 2 offers methods for custom usage through code. These methods are starting with "User_" like "**User_AddAllBonesImpact**".
- You can combine Ragdoll Animator 2 with my other packages, like Tail/Spine/Look, Animator. Check the manual pages for more details about it.

Contact and other links you will find in Readme.txt file

Under this file, you will find a list of helper methods of Ragdoll Animator 2.

Working with ragdolls requires a lot of possibilities to create your own unique physical behaviors. We prepared many methods, which you use in your project, to get desired results faster.

In the future there may be more helper methods.

When a package gets updates, this list may be outdated a bit, and fulfilled after a while.

To call the helper method, simply get reference to the Ragdoll Animator 2 component, and write `myRagdollAnimator.User_ ...`

All helper methods start with **User_** predicate.

[Tutorials on Youtube](#)

User_FadeMusclesPower : Transitioning all rigidbody muscles power to target value (RagdollHandler.MusclesPower)

User_FadeMusclesPowerMultiplier: Transitioning all rigidbody internal muscles power multiplier to target value (RagdollHandler.musclesPowerMultiplier) Should be used to transition to zero or one value.

User_DisableMecanimAfter: Disabling [Mecanim](#) after some time, useful for death behavior

(using **Mecanim** tech name, instead of **Animator** to not confuse it with Ragdoll **Animator** name)

User_GetNearestRagdollColliderToPosition: Returning nearest dummy bone collider, towards provided world point

User_GetNearestRagdollRigidbodyToPosition: Returning nearest ragdoll dummy bone rigidbody, towards the provided world point

User_GetNearestAnimatorTransformBoneToPosition: Returning nearest source bone transform, towards the provided world point

User_GetNearestPhysicalTransformBoneToPosition: Returning nearest dummy bone transform, towards the provided world point

User_GetNearestRagdollBoneControllerToPosition: Returning nearest bone setup, towards the provided world point

User_ChangeAllCollidersPhysicMaterial: Setting target physical material to all dummy bones (will be overwritten if using physical materials in the handler setup)

User_FreezeAndDestroyRagdollDummy: Call it if your character died and you want to keep it in the current lying pose, without possibility to make it ragdolled again. This method is disabling unity's Animator! - It's required to avoid playing standing animations, it would be resurrected.

[List<Rigidbody>](#) **User_GetAllRigidbody**: Generates new list of Rigidbodies belonging to the physical ragdoll dummy

[List<RagdollChainBone>](#) **User_GetAllRagdollDummyBoneSetups**: Generates new list of Ragdoll Bone Setups of this Ragdoll

User_UpdateRigidbodyParametersForAllBones: If you changed some ragdoll handler variables through code, you need to trigger settings refresh for the ragdoll dummy components

User_UpdateColliderParametersForAllBones: If you changed some ragdoll handler variables through code, you need to trigger settings refresh for the ragdoll dummy components

User_UpdatePhysicsParametersForAllBones: If you changed some ragdoll handler variables through code, you need to trigger settings refresh for the ragdoll dummy components

User_UpdateLayersAfterManualChanges: Call after changing layer settings in ragdoll animator (can't be executed in OnValidate for some reason, unity gives warning about that)

User_UpdateAllBonesParametersAfterManualChanges: If you changed some ragdoll handler variables through code, you need to trigger settings refresh for the ragdoll dummy components

RagdollChainBone User_GetBoneSetupByHumanoidBone: Returning reference to the ragdoll dummy setup, using HumanBodyBones unity enum

RagdollChainBone User_GetBoneSetupBySourceAnimatorBone: Returning reference to the ragdoll dummy setup, using source skeleton transform reference

RagdollChainBone User_GetBoneSetupByAnimatorBoneName: Returning reference to the ragdoll dummy setup, using source skeleton transform name

RagdollChainBone User_GetBoneSetupByDummyBone: Returning reference to the ragdoll dummy setup, using ragdoll dummy bone transform reference

Transform User_GetPhysicalBoneBySourceBone: Returning physical dummy bone transform which represents animator bone

Transform User_GetSourceBoneByPhysicalBone: Returning source animator bone transform basing on the physical dummy transform reference

Vector3 User_Teleport: Refreshing Ragdoll Animator object after teleporting or teleporting and refreshing rigidbodies of ragdoll animator

Vector3 User_GetStoredAnchorRootOffset: Defined offset from anchor (hips) towards character standing position (by standards - origin in feet middle)

Vector3 User_GetStoredAnchorRootOffsetRot: Defined offset from anchor (hips) towards character standing rotation

Vector3 User_BoneWorldForward: Computing target bone forward pointing direction in world space

Vector3 User_BoneWorldUp: Computing target bone up pointing direction in world space

Vector3 User_BoneWorldRight: Computing target bone right pointing direction in world space

Vector3 User_GetRagdollBonesStateBounds: Using collider bounding boxes volumes to define bounding box of current ragdoll pose

Vector3 User_GetPosition_BottomCenter: Calculating position based on the current state of physical bones. Using collider bounding boxes volumes

Vector3 User_GetPosition_Center: Calculating position based on the current state of physical bones. Using collider bounding boxes volumes.

Vector3 User_GetPosition_AnchorBottom: Calculating position based on the current state of the physical anchor bone collider.

Vector3 User_GetPosition_HipsToFoot: Calculating position based on the current state of physical bones. Trying to define a character object stand position with origin in feet. Using collider bounding boxes volumes.

Vector3 User_GetPosition_FeetMiddle: Calculating position basing on the leg chains last bones middle position

Quaternion User_GetMappedRotationHipsToLegsMiddle: Getting hips direction rotation, prioritized with up vector, dedicated for get up orientation, defining direction using middle feet position and anchor bone (hips) position

Quaternion User_GetRotation_Mapped: Getting hips direction rotation, prioritized with up vector

Quaternion User_GetRotation_MappedFor: Getting hips direction rotation, prioritized with up vector, prepared for certain get up type

Vector3 User_GetAverageDirectionOf: Summing and normalizing all chain bones target world directions

Bool User_GetUpByRotationPossible: Checking state for ragdoll get-up possibility case, basing just on the rotation of the anchor bone.

Bool User_IsOnBack: Basing on the orientation of the anchor bone, defining if character is currently lying on its back

Float User_CoreLowTranslationFactor: Calculating multiplier (multiplier which makes value lower 0-1) for average angular velocity, to lower get up angular velocity threshold when body is not moving

ERagdollGetUpType User_CanGetUpByRotation: Checking state for ragdoll get-up possibility case, based just on the rotation of the anchor bone.

ERagdollGetUpType User_LayingOnSide: Basing on the orientation of the anchor bone, defining if character is currently lying on its side

RaycastHit User_ProbeGroundBelowAnchorBone: Checking ground raycast below anchor/hips bone

User_TransitionToStandingMode: Few operations which are helpful for animating get up animation.

User_SwitchFallState: Switching ragdoll mode to fall / standing mode

User_FallImpact: Switching ragdoll mode to fall, applying strong impact to the selected bone and weaker impact to all bones

User_AddBoneImpact: Adding physical push impact to single bone's rigidbody

User_AddRigidbodyImpact: Adding physical push impact to the single provided rigidbody object

User_AddChainImpact: Adding physical push impact to whole chain bones rigidbodies

User_AddAllBonesImpact: Applying physical push to all rigidbodies of the ragdoll dummy

User_SetAllBonesVelocity: Assigning velocity to all rigidbodies of the ragdoll dummy

User_SetAllKinematic: Setting all ragdoll bones rigidbodies kinematic or non kinematic

User_SwitchAllBonesUseGravity: Setting all ragdoll bones rigidbodies Use Gravity switch

User_SwitchAllBonesMaxVelocity: Switch all dummy bones rigidbodies parameter

User_ChangeAllRigidbodyDrag: Switch all dummy bones rigidbodies parameter

User_SetAllAngularSpeedLimit: Setting all ragdoll bones rigidbodies angular speed limit (by default unity restricts it very tightly). You can set it using ragdoll handler max angular velocity in the inspector view or changing it through code and calling UpdateAllAfterManualChanges

User_SetAllInterpolation: Setting all ragdoll bones rigidbodies interpolation mode. You can set it using ragdoll handler interpolate mode in the inspector view or changing it through code and calling UpdateAllAfterManualChanges

User_SetPhysicalTorqueOnRigidbody: Applying torque impact on the selected rigidbody.

User_SetAllPhysicalTorque: Applying torque impact to all bones of the ragdoll..

Vector3 User_GetAllBonesMaxVelocity: Returning velocity of the bone with greatest velocity magnitude

Vector3 User_GetChainBonesAverageTranslation: Computing average translation of ragdoll chain bones. Average translation is calculated only when calling this method!
- To avoid calculating it all the time.

Float User_GetChainBonesAverageAngularVelocity: Computing average translation of ragdoll chain bones. Average angular velocity is calculated only when calling this method! - To avoid calculating it all the time.

Vector3 User_GetChainBonesVelocity: Computing average velocity of ragdoll chain bones

Vector3 User_GetChainAngularVelocity: Computing average angular velocity of ragdoll chain bones

RagdollAnimator2.Handler.User_OverrideMusclesPower: Overriding target muscles power slider, no matter what is driving this value in the current state.

RagdollAnimator2.Handler.User_GetAllDummyColliders: Generating new list including all ragdoll dummy colliders.

RagdollAnimator2.Handler.User_GetDummyRigidbody: Generating new list including all ragdoll dummy rigidbodies.

RagdollAnimator2.Handler.User_FindAllCollidersInsideAndIgnoreTheirCollisionWith DummyColliders: Using Physics.Ignore to trigger colliders ignore for all colliders within provided transform and ragdoll dummy colliders

If you like this package please visit my [asset store page](#) for more or write a review for this asset ;)