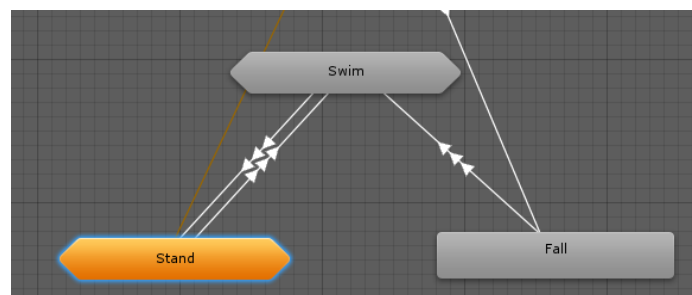




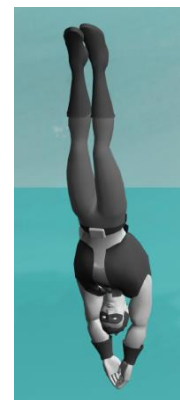
Overview:

Thank you for purchasing **Heroic Swimming by Animation Bandit**. *Please backup your work before importing this asset into your project for the first time.*

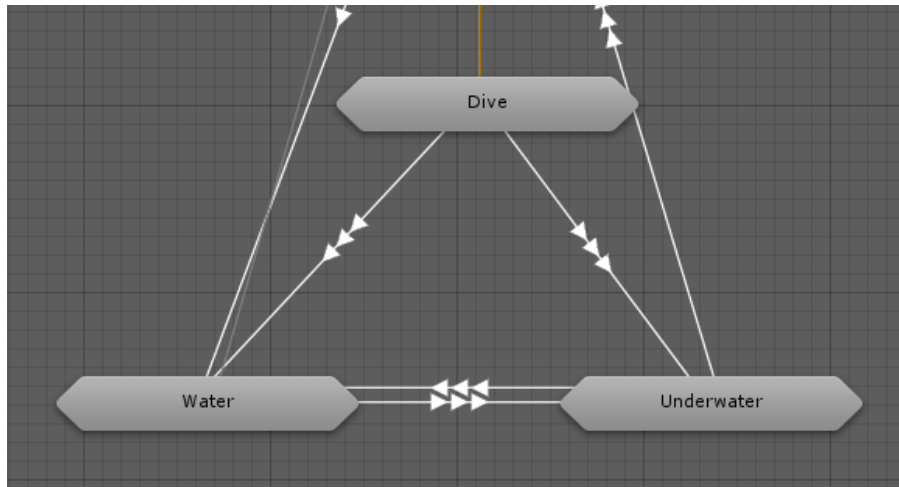
At the top level, the controller is divided into three states: **Swim**, **Stand**, and **Fall**



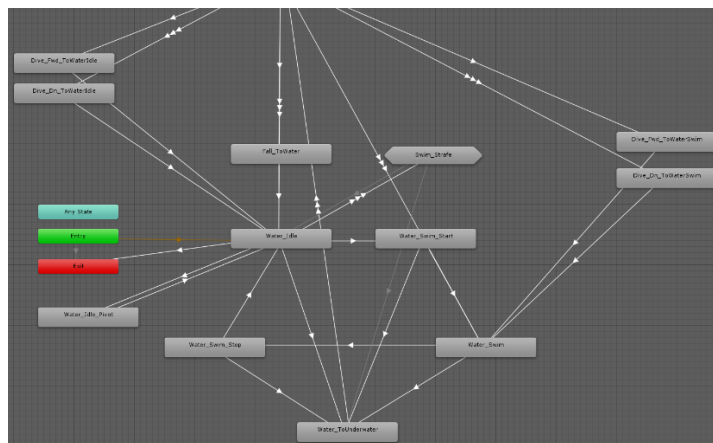
Nearly all the animation for this asset is contained within the state, **Swim**. Inside **Swim**, the animation is divided into three main states, **Underwater**, **Water**, and **Dive**.



Dive is used to transition from **Stand** and **Fall** to **Water** and **Underwater**:



Many animated actions and transitions are nested within each of these sub-states. Transition arrows show how each action blends to the next:



Parameters set the conditions required for the animated transitions between each state:

Has Exit Time

Settings

Timeline: 0:00, 1:00, 2:00, 3:00

Dive_Cycle

Dive_Dn_ToWaterIdle

Conditions

| | |
|------------|-------|
| Water | true |
| Underwater | false |
| Idle | true |
| Swim | false |

+ -

Parameters:

Though you may opt to add, remove or modify parameters for your own custom controller, reviewing the descriptions below will help clarify how each move is intended to function.

On Ground:

Idle: If this is the only parameter that is checked, the character will play his standing idle.

Diving Into Water:

Fall: Interrupts any active state and plays looping fall cycle. Enable **Water** and **Idle** to land in the water. You may also interrupt a fall cycle by enabling **Dive** or **Flip**.

Flip: Trigger while falling to play a forward flip cycle. Enable **Dive** when you want the character to stop flipping. **Flip** is only available during **Fall**, not **Idle**.

Dive: While in **Idle** or **Fall**, activate this trigger to make the character perform a basic dive. Until you tell the character to enter the water, he will continue falling in a dive cycle. You can modify how the character dives via other parameters such as:

Dive_High: If checked, the character leaps with his arms outstretched before tucking into a dive pose. This animation is only available diving from **Idle**, not from **Fall**.

Dive_Twist: If checked, the character performs an acrobatic twist in air. This animation is only available diving from **Idle**, not from **Fall**.

*The first 24 frames of Dive_Twist have been trimmed for player responsiveness, but you may opt to start the animation clip at frame 0 if you want the character to play a running intro leading into the dive.

Water: If enabled the character will complete his dive and transition to the surface of the water. Enable **Idle** to make the character tread water, or enable **Swim** to make him swim forward.

Underwater: If enabled the character will complete his dive beneath the surface of the water. Enable **Idle** to make the character idle underwater, or enable **Swim** to make him swim forward underwater.

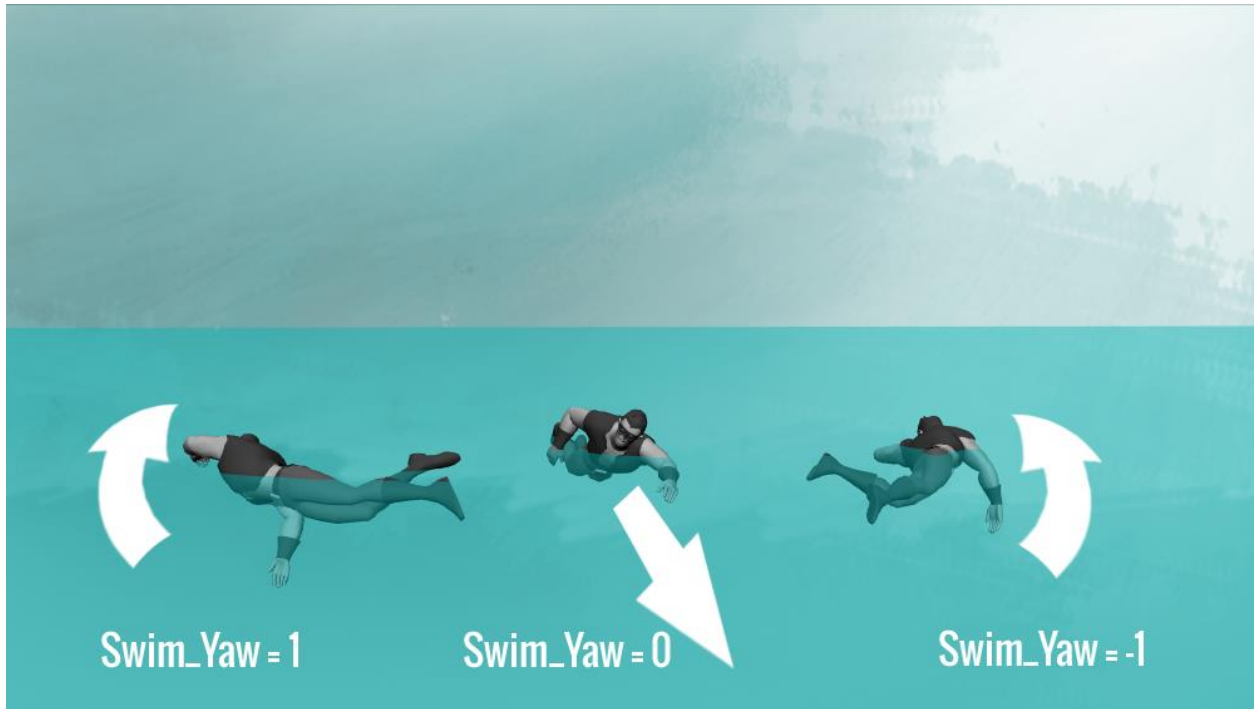
Swimming:

After diving into water, toggling between **Water** and **Underwater** will cause the character to submerge or swim to the surface. At any given time, only **Water** or **Underwater** may be active, not both. Disabling both **Water** and **Underwater** returns the character to his standing **Idle**.

Swim: Disable **Idle** and Enable **Swim** to make the character swim forward. Disable **Swim**, and enable **Idle** to make the character tread water.

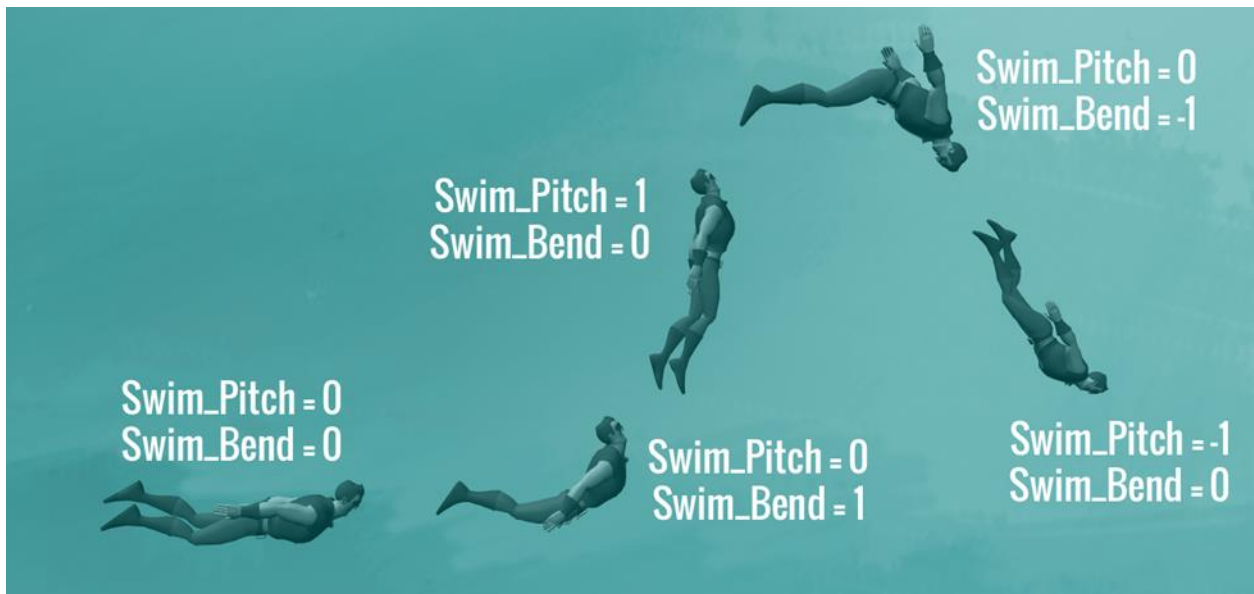
Swim_Kick: Enable while swimming underwater to play an alternate variation of the character's swimming cycle.

Swim_Yaw: Once the character is swimming forward, Swim_Yaw may be used to steer him to the left or right. (-1 = lean left, 0 =forward, 1= lean right).



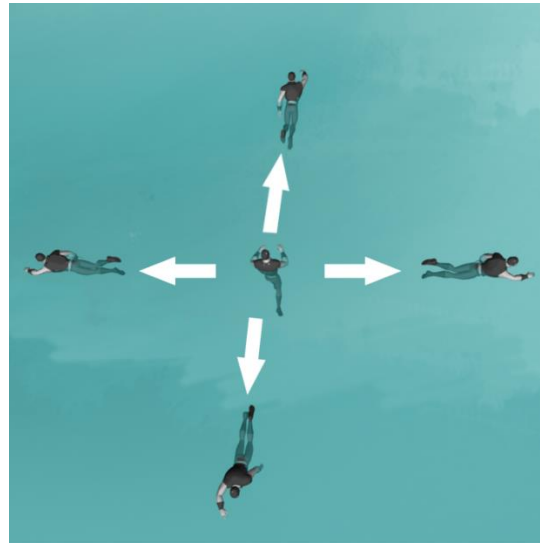
Swim_Pitch: This parameter is used exclusively while swimming underwater to steer up or down. (-1 = down, 0 =forward, 1 = up).

Although **Swim_Pitch** angles the character up and down, you will notice that his body shape is rigid like a plank of wood. Combine **Swim_Pitch** with **Swim_Bend** to curve the shape of the spine. For best results, use **Swim_Bend** only when the player first changes pitch up or down, and then gradually blend this parameter to 0.



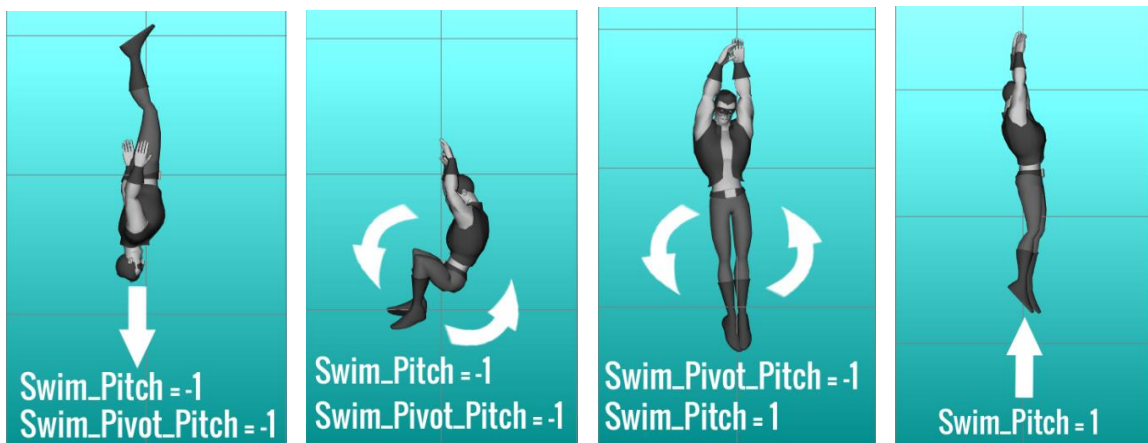
Swim_Pivot: Trigger to make the character rotate in place. While **Idle**, the direction he faces is determined by **Swim_Pivot_Yaw**. If **Swim_Pivot** is triggered during **Swim**, the character automatically flips 180 degrees in the opposite direction.

Swim_Pivot_Yaw: When transitioning from **Idle** to **Swim**, use this parameter to determine the character's start direction (-180 = left back, -90 = left, 0 = forward, 90 = right, 180 = right back).

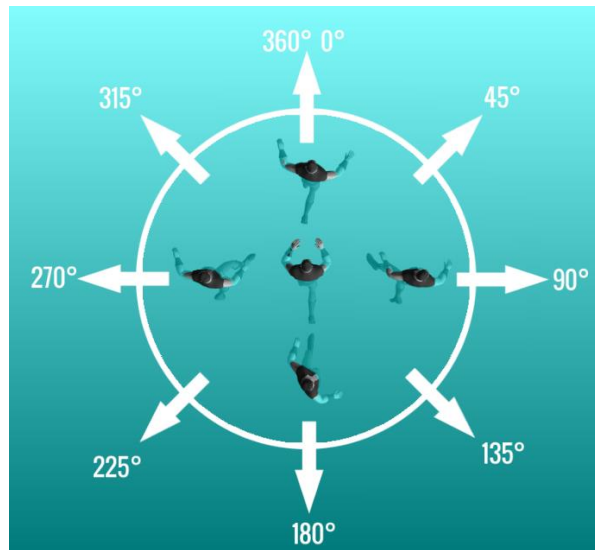


Swim_Pivot_Pitch: allows you to match the pitch of the pivot animation to the pitch of the underwater swim animation. This works as follows:

1. Trigger **Swim_Pivot**, and set the value of **Swim_Pivot_Pitch** to match the same value as **Swim_Pitch**. For example, if the character is swimming down, both parameters would be set to a value of -1.
2. The pivot animation will cause the character to flip 180 degrees in the opposite direction so that he is now swimming up. On completion of the pivot, change **Swim_Pitch** to a value of 1.



Swim_Strafe: When enabled, the character will move while treading water. Adjust the character's direction using the parameter, **Swim_Strafe_Direction** (clockwise rotation from 0° to 360°).



Drown: Trigger while underwater to make the character play his drowning animation. On complete, he will idle in his drowned state.

Revive: Trigger to return a drowned character to his default idle state.

Character Awareness:

These parameters adjust additive poses on layers that control where the character's head is aiming:

Aim_Pitch: Adjust to make the character look down/up (value range = -90 to 90). This is only used when standing idle.

Aim_Yaw: Adjust to make the character look left/right (value range = -180 to 180). This is only used when standing idle.

Water_Aim_Pitch: Same as above, but used when idle on the surface of the water

Water_Aim_Yaw: Same as above, but used when idle on the surface of the water

Underwater_Aim_Pitch: Same as above, but used when idle underwater

Underwater_Aim_Yaw: Same as above, but used when idle underwater

FAQ

“What is the advantage of purchasing Animation Bandit vs. hiring an animator or using motion capture?”

We believe that the vast majority of users will save time and reduce cost by using **Animation Bandit**. How much depends on the complexity of your project, the cost and skill of the animator you hire, and many other variables unique to your individual situation.

In general, a highly skilled animator working on AAA titles charges hundreds of dollars to produce about 3 to 4 seconds of quality work per day. Animating a complete character from scratch often takes months of labor to complete at great expense.

If you use motion capture you will spend many thousands of dollars per day renting a sound stage, and must hire an actor and a team of professionals to solve that data to your character rig. You will still need to hire an animator to clean up your raw data and turn it into game-ready motion cycles.

You may opt to supplement this asset with your own animation, but using **Animation Bandit** as your foundation lets you focus on the specific actions that truly make your game unique.

“What do I need to know before using this asset?”

This asset assumes users have a basic understanding of Unity’s animation tools, Mecanim, state machines, blending, etc. If this is unfamiliar to you, please consult tutorials on the Unity website.

“How do I make my character walk, run, jump, climb, and strafe?”

This asset is dedicated exclusively to swimming and diving in water. The standing idle animation is included only to demonstrate how this asset transitions into the dive state. If your game requires additional locomotion, consider purchasing **Heroic Traversal**, sold separately here:

<http://u3d.as/xui>



“How do I apply Animation Bandit’s motion to my own character?”

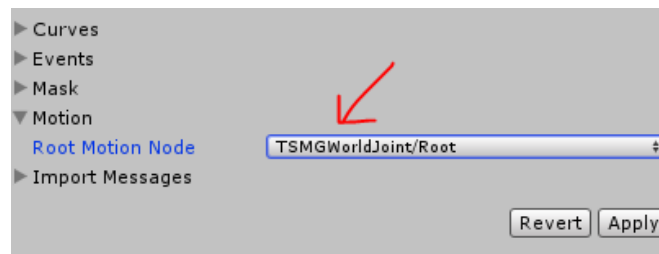
Animation Bandit is compatible with Unity’s Mecanim system which allows for retargeting to other Humanoid characters. Please consult Unity’s website for guidelines on how to properly rig your character. Animation will look best on characters similarly proportioned to the one provided with this asset.

“What if I want my code to translate the character as opposed to root motion?”

FBX files contain two versions of each animated clip. Use the version labeled “InPlace”.

“Where are the options to bake root motion into the pose?”

By default, Animation Bandit is set to follow an animated root joint. This provides the highest quality motion since the root is authored to cycle perfectly (no drift on the X axis for example).



If you want to bake root motion into the pose (to remove Y translation for example), you can do that by changing the root motion node to “none”. If you do this, please understand that Unity is estimating the position of the root based on the character’s center of mass, and this approximation is not always correct. This sometimes results in unwanted motion drifting even though there is no drift in the animation itself. In that case you are better off either using the authored root motion from the root joint, or by setting the character position in code using “InPlace” cycles.

“How do I create a playable character, add AI, sound effects, interactions with environment geometry etc.?”

All of the above require custom code, and falls outside the scope of this asset which is merely a collection of animated actions. Unity provides a wealth of tutorials on their website with instructions on how to add programmed behaviors to your game.

“I want the character to perform an action not included in this asset. Will you create additional custom content for my game?”

No promises on when/if we will create additional content, but your feedback and suggestions for future assets are always welcome.

“How do I adjust the animation to fit the custom needs of my project?”

Maya and MotionBuilder both use a Human IK system that allows you to retarget FBX animation to your character rig for further editing. Please consult documentation on how to use these tools for your preferred animation software.

“How do I share this with others for free, or sell it as a part of my asset pack?”

This asset may be used by the purchaser as part of a compiled game, but it is not meant to be shared or redistributed for any other purposes. We rely on the support of our customers to continue making new content. Thanks for helping us help you make great games.

Thanks again for purchasing **Heroic Swimming by Animation Bandit**. We value your feedback and questions. Feel free to send us a message on our website.

www.AnimationBandit.com