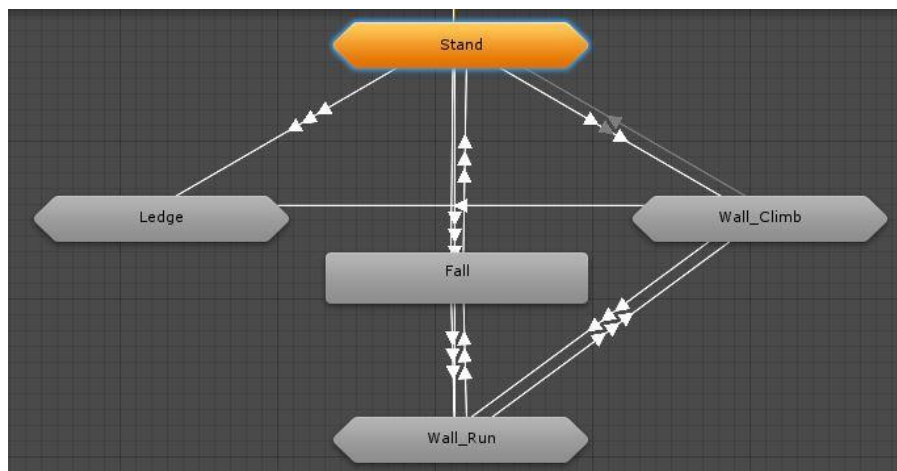




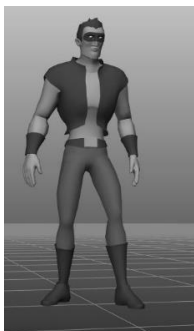
Overview:

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

The controller is divided into five primary states: **Stand**, **Ledge**, **Wall_Climb**, **Wall_Run**, and **Fall**



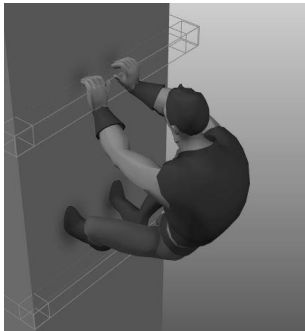
Stand:



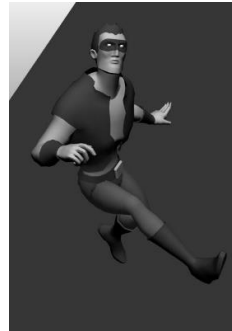
Ledge:



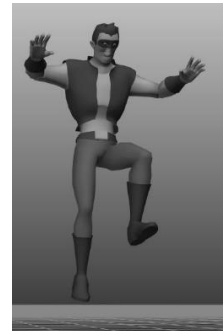
Wall_Climb:



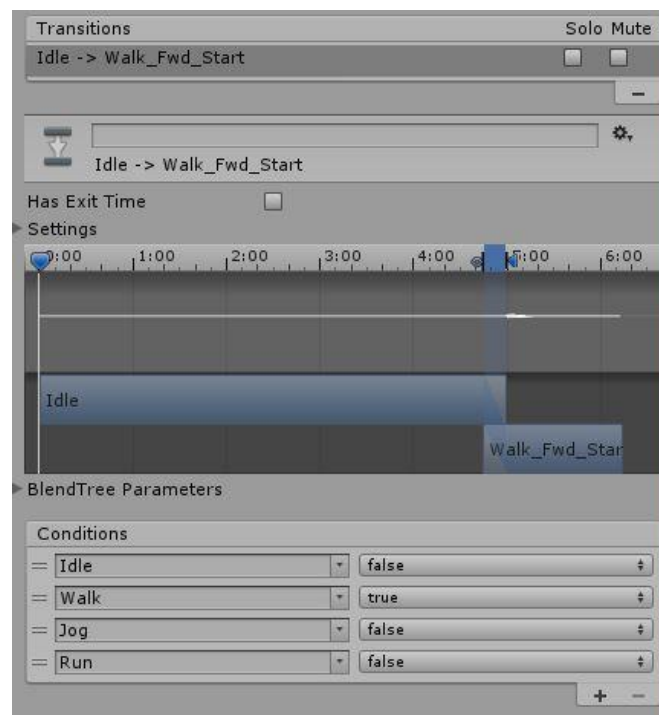
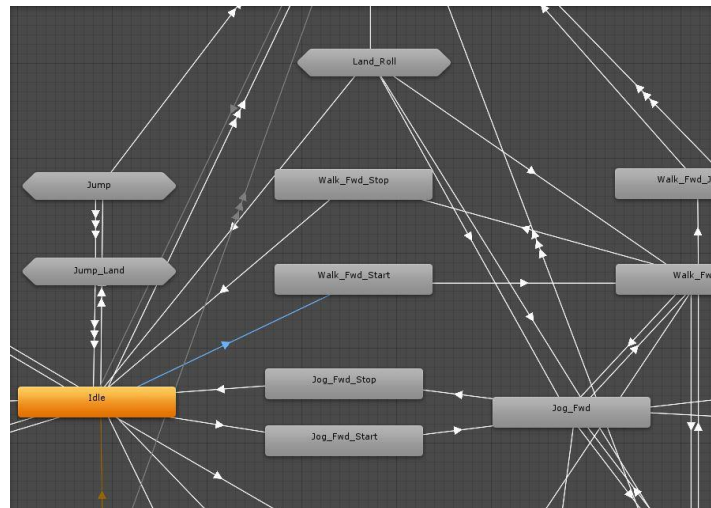
Wall_Run:



Fall:



There are many additional sub-states nested within each of these broad categories to create fluid transitions between each action. Here is a look inside **Stand**. Parameters set the conditions required for the animated transitions between each state.



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.

Idle, Walk, Jog, Run: Check these parameters to set the character's locomotion speed. Only one movement speed may be selected at any given time. **Direction:** -1 = lean left, 0 = forward, 1 = lean right.

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

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

Jump: May be triggered when idling, walking, jogging, running, wall running or wall climbing. When idle, you have the option to set the **Height** parameter to 0.5, 1, 2 or 3 meters.

Fall: Interrupts any active state and plays looping fall cycle.

Land: Interrupts the fall cycle and plays a roll to land animation.

Land_Heroic: If checked, **Land** trigger will play a three-point landing. Uncheck to return to stand.

Roll: Triggers a dive forward to roll animation. Used while idle, walking, jogging, or running.

Ledge: If enabled, the character will grab onto a ledge at the apex of his jump.

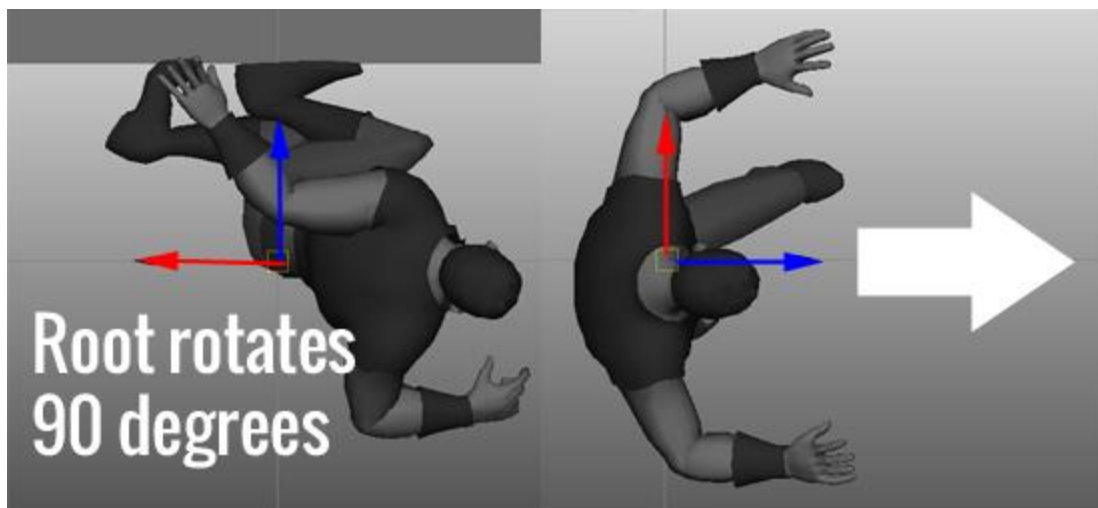
Wall: If enabled, the character will grab onto a wall at the apex of his jump.

WallRun: Trigger to make the character enter Wall_Run state. You must also enter a value for the

Direction parameter. Values less than zero run left. Values greater than zero run right.

* If **Idle** is not checked, the character continues to play his wall run cycle until directed to stop. Checking the **Idle** parameter while wall running causes the character to enter Wall_Climb state.

*The character's root always faces towards the wall when wall running, even though it appears he is running to the right. **When jumping off the wall**, you must set the **Direction** parameter so that the root rotates away from the wall to face the direction he is jumping.

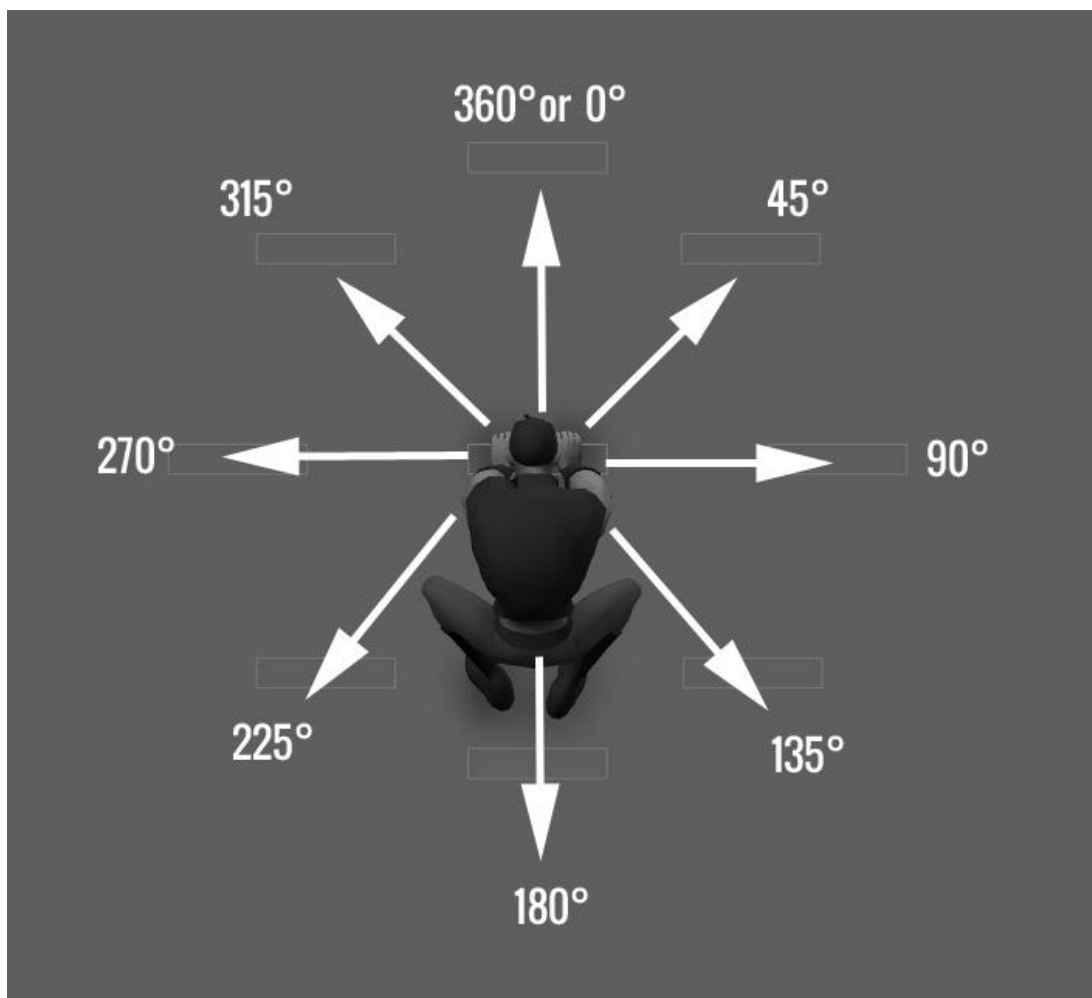


Pivot: Trigger to turn the character in place. When standing idle, set a **Direction** value where -180 turns 180 degrees to the left, and 180 turns 180 degrees to the right. The values in between are blended to allow pivoting in any direction.

***Pivot** may also be used when **hanging from a ledge**, **swinging on a ledge**, or **climbing on a wall**. In these cases it is always assumed the character is pivoting 180 degrees.

Climb: When standing idle, triggering **Climb** will cause the character to grab onto a wall. Once on the wall, triggering **Climb** will cause the character to move to an adjacent handhold.

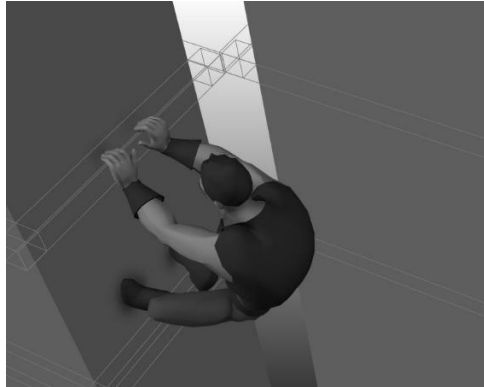
Climb_Direction: Set this value to determine which handhold is grabbed. The motion is blended so you can place handholds at any angle.



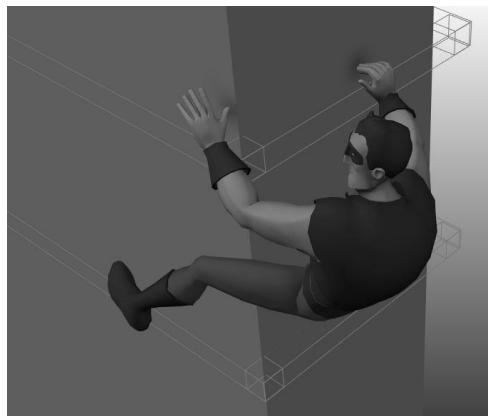
ClimbAim_Pitch: Adjust to make the character look up/down (value range = -90 to 90). This is only available when idle on a wall.

ClimbAim_Yaw: Adjust to make the character look left/right (value range = -180 to 180). This is only available when idle on a wall.

CornerIn: Climb across a 90 degree interior corner. Uses **Direction** (-1 to 1) to determine left/right.



CornerOut: Climb across a 90 degree outside corner. Uses **Direction** (-1 to 1) to determine left/right.



Mount: From stand, climb up and stand on top of an obstacle. Set **Height** to determine whether character takes a small step up onto a curb, or scrambles to the top of a massive wall. Supported heights are 0.5, 1, 2, and 3 meters.

Enable **Ledge** while mounting to transition to a hanging pose. A positive **Climb_Direction** value mounts up, and a negative value drops down to the ledge.

Dismount: From stand, step down from the top of an obstacle to the ground. Similar to **Mount**, this also uses the **Height** parameter. **Dismount** may also be used to transition from **Wall_Climb** or **Ledge** to **Stand**.

Shimmy: Set the **Shimmy_Direction**, and then use this trigger to move the character left/right when hanging from a ledge.

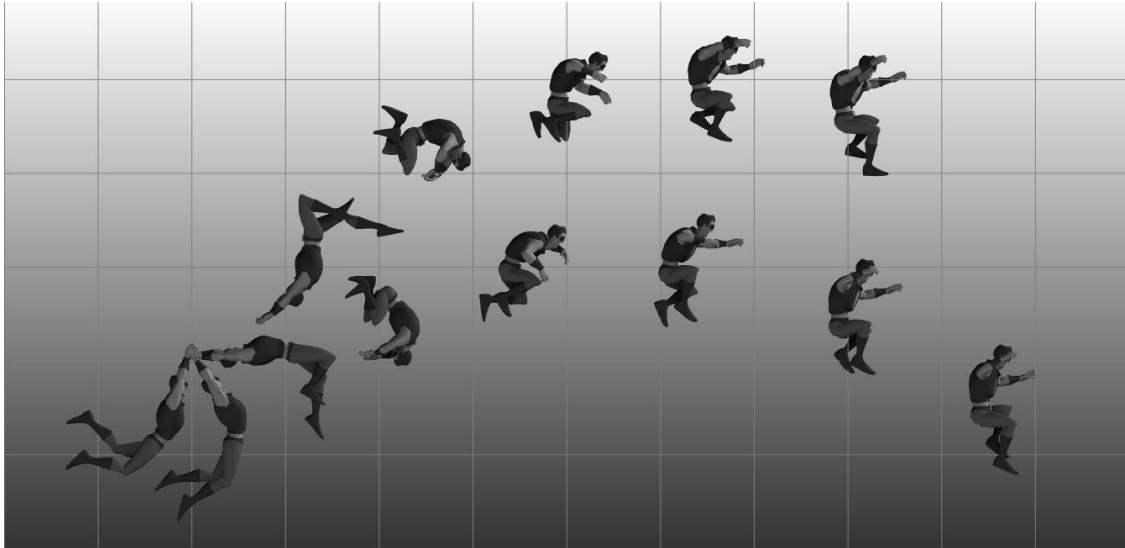
Swing: First confirm that **Ledge** is enabled, and then trigger **Swing** to make the character kick forward and generate momentum with his body. If **Idle** is enabled, the character will return to hanging on a

ledge. If **Idle** is disabled, the character will continue generating momentum with his body in preparation for a **SwingRelease**.

Swing360: If enabled, when **Swing** is triggered the character will make a 360 degree revolution around the beam.

SwingRelease: Leap forward away from the ledge. Must already be swinging prior to triggering release.

SwingRelease_Angle: (0 = release low, 45= release high)



Slide: Trigger to make character slide forward

Slide_Pitch: Adjust to set slope angle for slide (0 = flat surface, 45 = 45 degrees down).

Slide_Cycle: If enabled, character will continue sliding until unchecked.

Strafe: If enabled, character will side step and walk backwards. This is only available for walk and jog speeds at this time.

Strafe_Direction: Enter a value from 0 to 360 to determine which direction the character moves.

FAQ

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

We believe that the vast majority of users will save time and reduce cost by using **AnimationBandit**. 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 **AnimationBandit** 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 apply AnimationBandit’s motion to my own character?”

AnimationBandit 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”. If you would like to match the animated speed but prefer to control motion with code, use these settings:

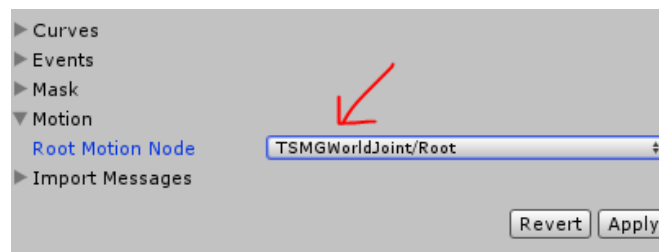
Walk = Z axis linear speed of 1.75862 meters/cycle (1.5 meters/second)

Jog = Z axis linear speed of 3.57143 meters/cycle (4.70779 meters/second)

Run = Z axis linear speed of 3.7512 meters/cycle (6.78645 meters/second)

“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 during run cycles for example).



If you want to bake root motion into the pose (to remove Y translation during a jump), 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.

“When climbing, why do my character’s hands and feet clip through the ledge?”

This asset assumes flat wall surfaces. If your character is proportioned differently or your terrain is uneven, you may have to use IK to fix clipping issues. Please consult the Unity website for tutorials.

“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 Traversal by Animation Bandit**. We value your feedback and questions. Feel free to send us a message on our website.

www.AnimationBandit.com