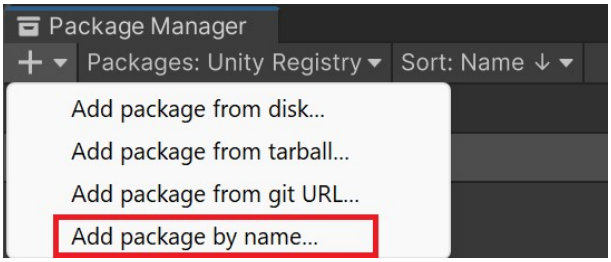
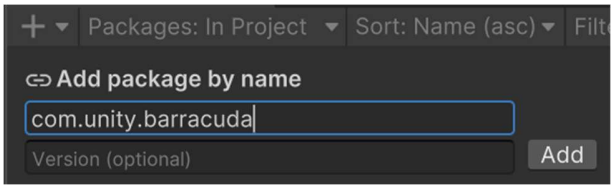
Documentation

Dependencies

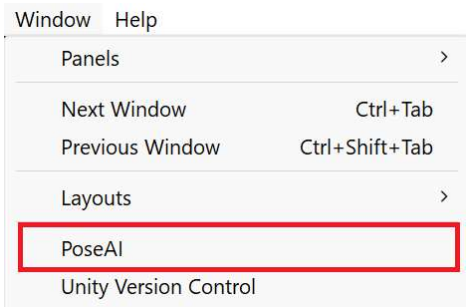
This asset requires the external package **Barracuda v3.0.0** to be installed.
Window > Package Manager > Add package by name > com.unity.barracuda



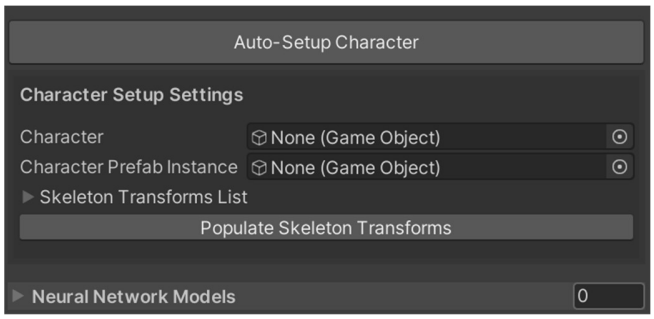


Getting Started

The PoseAI Editor is a generative AI tool designed for aiding animations and authoring poses. This editor uses learned inverse kinematics (IK) to predict and estimate body positions based on constraints applied to various joints.



Features



Auto-Setup Character

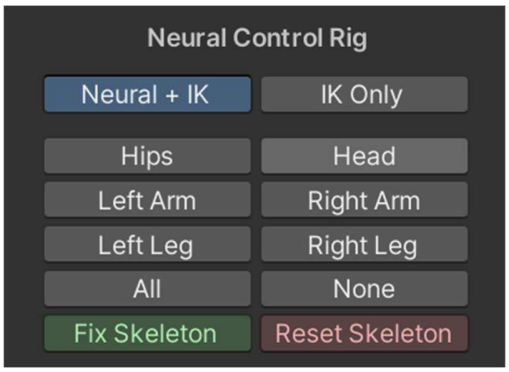
- Button: Auto-Setup Character
    - This button automatically sets up the character for animation using the AI and IK systems. It streamlines the initial setup process by configuring the character's necessary components and settings.

Character Setup Settings

- Character
    - Field: Character (Game Object)
        - Allows you to select the character game object that will be used for the animation.
- **Character Prefab Instance**
    - Field: Character Prefab Instance (Game Object)
        - Allows you to select a prefab instance of the character for consistent setup and testing.
- **Skeleton Transforms List**
    - Expandable Section: Skeleton Transforms List
        - Provides a detailed list of the character's skeleton transforms, allowing for precise control and adjustments. This is only for your to check if there are 52 transforms for your custom character.
- **Button: Populate Skeleton Transforms**
    - Populates the skeleton transforms list with the character's bone structure, making it easier to manage and edit the skeletal configuration. Populates the list based on naming. This is a naming sensitive auto populate method.

Neural Network Models

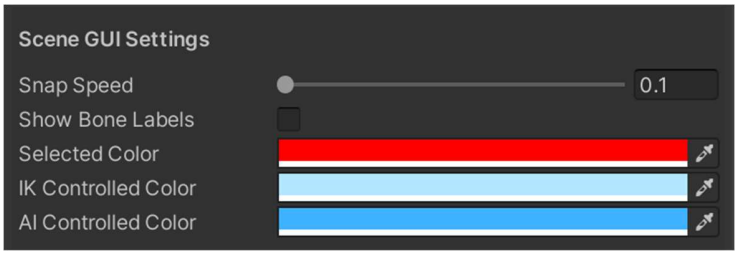
- **Neural Network Models**
    - Field: Neural Network Models (NN List)
        - Indicates the number of neural network models available for use. This can be adjusted based on the complexity and requirements of the animation.



Neural Control Rig

- **Neural Control Rig Options**
    - **Button: Neural + IK**
        - Activates the neural control rig with both neural network predictions and inverse kinematics for comprehensive animation control.
    - **Button: IK Only**
        - Activates the control rig using only inverse kinematics, without neural network predictions.
- **Joint Control Buttons**
    - Hips, Head, Left Arm, Right Arm, Left Leg, Right Leg
        - Individual buttons to toggle the neural and IK control for specific joints. This allows for granular control over the animation process, enabling or disabling AI-driven predictions for each joint independently. The feedback on what AI function is active can be also seen through the material update.
- **Global Control Buttons**
    - **Button: All**

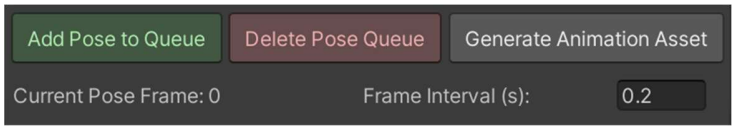
        - Enables neural and IK control for all joints simultaneously.
    - **Button: None**
        - Disables neural and IK control for all joints.
- **Skeleton Adjustment Buttons**
    - **Button: Fix Skeleton**
        - Applies fixes to the skeleton based on the neural and IK predictions to correct any misalignments or issues.
    - **Button: Reset Skeleton**
        - Resets the skeleton to its default state, removing any applied neural or IK adjustments.



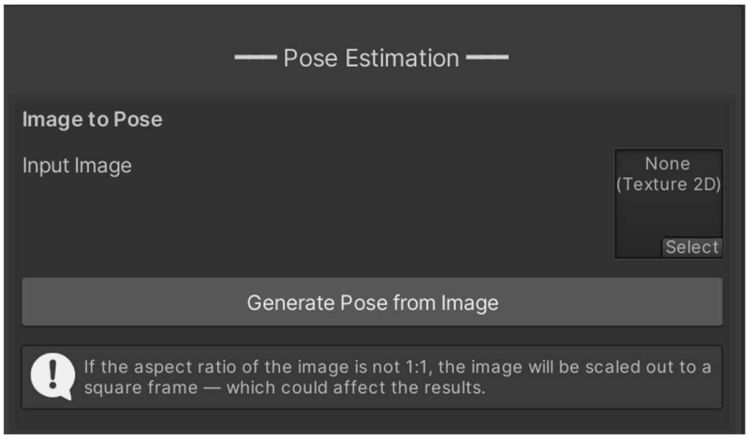
Scene GUI Settings

- **Snap Speed**
    - **Slider: Snap Speed**
        - Adjusts the speed at which snapping occurs during pose adjustments. A higher value means a faster snap. A lower value will SLERP the quaternions to their places slowly. If you have a low end graphics card, please increase the value of the snap.
- **Show Bone Labels**
    - **Checkbox: Show Bone Labels**
        - Toggles the display of bone labels in the scene for easier identification and manipulation.
- **Colors**
    - **Selected Color**

        - Color Picker: Selected Color
            - Allows you to change the color used to highlight selected bones or joints.
    - IK Controlled Color
        - Color Picker: IK Controlled Color
            - Allows you to change the color used to highlight IK-controlled bones or joints.
    - AI Controlled Color
        - Color Picker: AI Controlled Color
            - Allows you to change the color used to highlight AI-controlled bones or joints.



- Pose Queue Management
    - Button: Add Pose to Queue
        - Adds the current pose to the animation queue.
    - Button: Delete Pose Queue
        - Deletes all poses in the current animation queue.
    - Button: Generate Animation Asset
        - Generates an animation asset from the poses in the queue.
- Current Pose Frame and Frame Interval
    - Current Pose Frame
        - Displays the index of the current pose frame.
    - Frame Interval
        - Number Input: Frame Interval (s)
            - Sets the time interval between the capture of a pose in seconds.



Pose Estimation

- **Input Image**

  Please make sure the image you import is set to readable and NPOT.

  - **Field: Input Image (Texture 2D)**

    - Allows you to select an image to be used for pose estimation. This image can be of any resolution. Internally, this will be scaled to 1:1, the closer the aspect ratio to 1:1, the better the results.

- **Button: Generate Pose from Image**

  - Generates a pose based on the selected input image. Note that if the aspect ratio of the image is not 1:1, it will be scaled to fit a square frame, which may affect the results.

Pose Preview

- **Preview Animation**

  Works by selecting an anaimtion clip in the project and opening the PoseAI window.

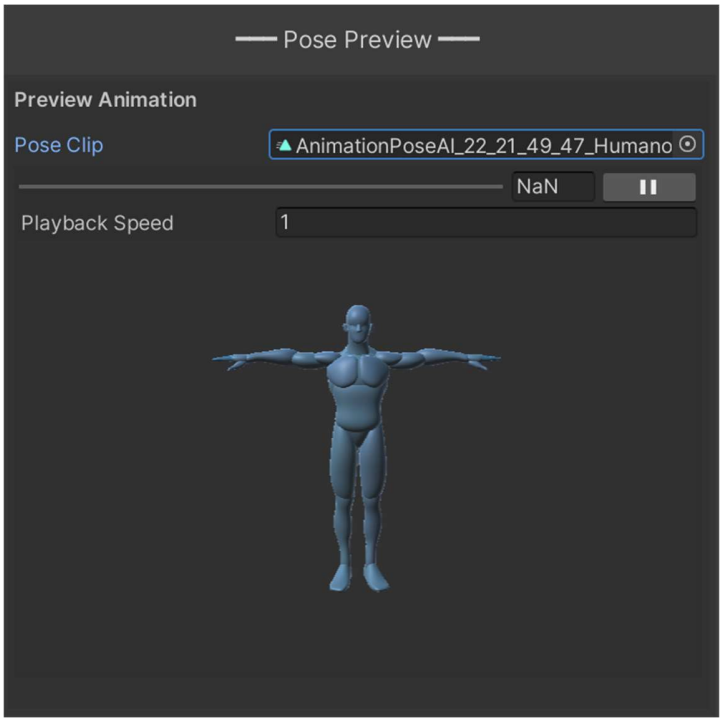
  - **Pose Clip**

    - Allows you to select the animation clip to be previewed.

  - **Playback Speed**

    - **Slider: Playback Speed**

      - Adjusts the speed at which the animation is played back in the preview window. Note, that this is not an accurate representation of the actual animation speed. This is calculated based on the time interval set between the frames internally.



- **Animation Preview Window**
  - Displays the selected animation clip, allowing you to visually assess the generated animation. This uses the prefab from the Models folder called RagdollAnimationPreview.fbx and changes its pose to display it in the editor.

Neural Network Architecture:

A feed forward network is designed for the prediction of limb position and a ResNet for the Pose Prediction. The Feed Forward neural net is fully connected (Gemm) layers interspersed with ReLU activation functions.

Here is the architecture of the neural network used:

Input Layer:

- The input to the network is passed into the first Gemm (General Matrix Multiplication) layer.

First Gemm Layer (Fully Connected Layer):

- This layer performs a matrix multiplication between the input and a weight matrix B of size 128×4 then adds a bias vector C of size 128.
- Weight of matrix B is transposed before multiplication.

First ReLU Activation Layer:

- The output of the first Gemm layer is passed through a ReLU (Rectified Linear Unit) activation function, which applies an element-wise nonlinearity.

Second Gemm Layer (Fully Connected Layer):

- The output from the ReLU activation is fed into another Gemm layer.
- This layer multiplies the input by a weight matrix B of size 128×128 and adds a bias vector C of size 128.
- Again, the weight matrix B is transposed before multiplication.

Second ReLU Activation Layer:

- The output of the second Gemm layer is passed through another ReLU activation function.

Output Gemm Layer (Fully Connected Layer):

- The output from the second ReLU activation is fed into the final Gemm layer.
- This layer performs a matrix multiplication with a weight matrix B of size 204×128 and adds a bias vector C of size 204.
- The weight matrix B is transposed before multiplication.

Output:

- The final output of the network has a fixed shape depending on the input and the neural networks are switched to coordinate the movements.

If you face any issues with the asset, have any questions or want to collaborate, please let us know at info@aikodex.com.

Happy Animating!

AiKodex