# Geppetto v.1.01

This guide will assist you in learning how to use the Geppetto toolset.

Contents

[Geppetto v.1.01 1](#_Toc297126512)

[The Geppetto toolset UI 2](#_Toc297126513)

[The Blueprint UI 3](#_Toc297126514)

[Install Character 5](#_Toc297126515)

[Setup Tools 6](#_Toc297126516)

[Module Maintenance 6](#_Toc297126517)

[Animation UI 7](#_Toc297126518)

[Workflow for Character Setup 9](#_Toc297126519)

# The Geppetto toolset UI

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | Use this button to access the help files and other materials related to Geppetto. | | This opens the Blueprint UI. The Blueprint UI is where you will layout the structure of your character. This is where you will layout joints, choose rotation orders. | | Character Install provides a list of all characters that are ready for use. Click the appropriate character to load it into your scene. | | Module Maintenance opens up the rigging tool. Animation components can be added on demand using this tool. After you have added the necessary animation components, click the maintenance button again to leave maintenance mode. | | Setup tools to assist the TD in preparing character assets for animation and export. Here you can find skinning tools, and tools to assist in creating an export file. | | The reference button will properly reference a character into your scen for animation or export. | | The animation UI contains a variety of tools to assist animators. Tools include the ability to blend between controls, change the look of animation components, and the ability to add space switching. | | The anim library will allow the saving and transfer of animations. | |

# The Blueprint UI

The Geppetto toolset eliminates the need for what we would traditionally consider character rigging; however it is still necessary to complete some processes by hand. Only a trained eye can determine the proper joint placement for a character. With that in mind, Geppetto attempts to make the job of joint layout a little easier. This is where the Blueprint UI comes into play.

The Blueprint UI houses a set of blueprints which can be thought of as joints. Blueprints can be translated, rotated, and to define the joint chains that will construct your characters skeleton. The UI provides additional functionality to assist in orienting and hooking together joints. Let’s run through the UI to get a better understanding of what everything does.

The UI consists of two tabs. The Modules tab is where you will find all the tools needed to build your character. The Templates tab is the template library where you can load and save pre-defined templates such as bipeds, quadrupeds, or anything else you might need on a regular basis.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | This are stores the blueprint modules. Blueprint modules define a particular limb or component. Modules are hooked together to form the characters layout. Each module contains  Additional modules can be written as needed and those new modules will automatically populate the UI. | | This are allows you to name your module and perform various hooking, grouping, and mirroring functions. | | Here you will find all the custom attributes for the selected module. Most modules will contain UI components for orientation, but other attributes can be added as well. | | The lock button converts your setup to joints for publishing. | | The publish button is the final step. This will save your character for animation. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | | --- | | This is a list of available templates. Templates will save time by providing a list of pre-defined rigs. You can edit existing templates and add your own.  Each template provides a description that will give you more information regarding that template. | | The Prepare for Template button will group all of your components and delete any nodes which are no longer needed. | | The Save Current as Template will save all the modules in your scene as a new template. | |

**Let us now go into more detail concerning some of the functions contained in the Blueprint UI.**

**Re-hook:**  Re-hook connects individual modules. This can be thought of as a parent constraint. To use re-hook simply select a module then select the module you want to hook and press the re-hook button. A visual representation will be drawn between modules. This tool will define your joint hierarchy.

**Snap Root > Hook:** This moves the hooked object into the exact position of its parent.

**Constrain/Unconstrain Root:** This will create a solid constraint between modules. If you move the parent object, the child will follow.

**Group Selected:** Group selected allows you to group multiple modules for easy manipulation. For example, you could group a hand and all of its associated fingers. Now you can move the entire hand and associated fingers by manipulating one control object.

**Ungroup:** This simply removes all the modules from the selected group.

**Mirror Module/Group:** This assists in creating a symmetrical setup and functions just like Maya’s mirror joint command.

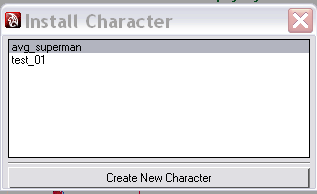
**Duplicate:** Makes a copy of the selected module.

**Delete:**  Deletes the selected module.

**Symmetry Move:** Allows you to move a module and its mirror at the same time and in a symmetrical fashion.

# Install Character

The Install Character UI is simply a pick able list of all characters that have been published from the Blueprint UI. Installing a character brings in a set of smart joints that are module aware. This means each joint knows what animation modules can be installed on it.



# Setup Tools

|  |  |
| --- | --- |
|  | This is a WIP with more features planned in the future.  Export Game Assets will create a rig and a setup file which are referenced into a scene for export or animation.  Add Holding Locations adds a set of standard holding locs.  Bind Skin will bind to only the required game joints. Bind skin also sets up the bind options and puts the geometry in a designated group.  More information on these tools and more will be found in the workflow outline. |

# Module Maintenance

|  |  |
| --- | --- |
|  | Module maintenance puts the character into an edit mode where new animation modules can be added. Each joint is aware of what animation modules can be installed on it, and what modules have already been installed.  Module maintenance mode will be the first step when an animator is assigned to create an animation. The animator can use this UI to setup the character in a way that suits the shot or their particular animation style.  Once the character is setup, that file can be saved out to form the basis of future animations. |

At some point I will provide a UI for this functionality. When the desired modules have been added, simply press the module maintenance button again to leave maintenance mode. Module maintenance mode can be initiated at any time during the animation process.

Note: Module Maintenance Mode is the backbone of why this type of rigging and animation system rocks! This is a non-destructive way to rig. If an animator should need a new type of control, that control can be added and used by the whole team without disrupting previously created animations.

# Animation UI

As mentioned earlier, the animation UI contains the tools needed to work with animation components. Here is a brief intro to the UI followed by a more detailed description of some of the key functions.

|  |  |
| --- | --- |
|  | First off you will see a couple of lists. On the left side is a list displaying all of the modules that makeup the character. When you select a module, you will see all the animation modules that have been applied to that module. You can effectively navigate your entire character through this UI.  The active module dropdown makes a module active, meaning that all the setting in the lower windows will affect that module when they are changed.  The Module Weights area of the UI is where you will do your switching. You can blend between any numbers of animation modules that have been applied to that particular part of your character.  Space Switching: You can switch any control in your scene to that of any other object or control in the scene.  Module LOD: Controls the visibility of the selected control object.  All the keyable attributes are displayed in the UI  Preferences: You can change the size and color of the selected control object. |

Please note that you will not be able to dynamically setup hook objects in this version of GEPPETTO. If you need a new hook target setup, you will need to do it in the base rig.

The next page provides a more detailed explanation of the components found in the animation UI.

|  |  |
| --- | --- |
|  | This UI component contains a list of all the blueprint components associated with the character. If you click on blueprint (body part), the list on the right will show all the animation modules installed on that blueprint. Selecting a module in the list will populate the other UI elements with information pertaining to that module. |
|  | Active Module: This dropdown list contains the animation modules installed on the selected blueprint. Select an animation module from this list to make it active. In this case “active” means that you can animate the module. Think of it like IK/FK switching.  Key All : This is like keying the IK/FK switch  Graph Weights will open an instance of the hypergraph with the space switch keys framed. |
|  | Module weights: This element is linked to the active module. This UI element allows you to blend between controls or to blend back to the “bind pose”. |
|  | This is like IK/FK matching. |

# Workflow for Character Setup

This section will provide a step by step overview of how to prepare a character for animation and export using Geppetto.

Gepetto requires that the art tree be formatted in a particular way. This formatting can be changed on a per project basis. Please refer to the Hendrix art tree for an example of how to structure your character folders. [\\corp-nas01\DC\dc\_art\character](file:///\\corp-nas01\DC\dc_art\character)

Each root character folder should have a directory named GEPPETTO. This is where the tool will store data such as blueprint templates.

The first step in setting up a new character will be to launch the Geppetto tool. This can be found in Maya under Turbine Art Platform/Geppetto. This will open the Geppetto Toolset UI.

From the UI you can launch the blueprint UI. Most of the setup work will be done here.

You should also import the mesh you will be rigging so you have a reference for joint placement.

This process is a little bloated and will undergo some serious streamlining in future versions.

### Bluprint UI

1. I suggest starting with a template as this will greatly simplify things. To access the templates, simply hit the templates tab and click on an appropriate template. You can add or remove modules from this new template instance if your character requires it.
2. If an appropriate template does not exist, you can begin building one from blueprint modules. Refer to the previous section on the Blueprint UI to learn more about hooking, unhooking, symmetry move, and module mirroring.
3. Position the modules as you would a joint. Each module has a variety of visual helpers that will assist in visualizing the joints orientation. Make sure you name your bp modules using the UI.
4. Once you have your modules placed the way you want, I suggest saving a new template for later use. I also suggest backing up this working file just in case.
5. Now you can lock and publish your new character. I suggest deleting the mesh you imported before locking. The locking process should clean up any useless nodes, but I suggest the extra step just in case. You may wonder why lock and publish are two separate processes. The tool was built this way in anticipation of a future feature.

Notes:

At this time module naming must follow a specific format. Ex: “arm1\_\_li1”. This convention indicates that this will be arm 1 of x, and that it is the first instance of that arm on the left side. Please reference [the Geppetto Naming Conventions Doc](file:///\\corp-nas01\TAP\docs\GEPPETTO\Geppetto_NamingConventions.docx) for more information.

Character Install  
Now it is time to install the character, setup the rig, and weight the mesh.

1. Make sure you have a clean character mesh in “characters/chr\_someguy/mesh/”
2. Click the Character Install Button in the Geppetto Toolset UI.
3. Choose the new character from the list and click Create Character.
4. You should now see your character mesh in the scene. The mesh should already be bound to the appropriate joints. If you open the outliner you will see two container nodes. The character\_container holds the “rig”. Setup holds all the engine specific joints, holding locations, and the mesh. If all went well your bones should be marked, and holding locators created.
5. Save this file in “character/chr\_someguy/rig/wip/” You will be able to come back to this file later if you need to make changes to weighting, controls, or anything else relating to this character.

### Module Maintenance

Use the module maintenance button to start setting up animation controls. We were able to do this process dynamically during animation, but that needed to change to support referencing. We will work that feature back into the system when time allows.

This process is also being automated; however the tech is still a WIP. Look for its release later in the year.

1. Click Module Maintenance to put the character in maintenance mode.
2. Select the joint chain you want to rig.
3. A list of available animation modules will populate the Module Maintenance window. Select a module and install.
4. Continue selecting joints and installing modules until you have the rig you want.

### Animation UI

As I said earlier, much of the setup process was intended to be handled by animators. As such, some of the tools we need to finish off the rig are embedded in the animation UI. You will only need to be concerned with hooking and control size and color.

1. Select an animation control and press the Animation UI button.
2. By default each animation control lives in the space of its direct parent. We will want to give the animators more options as to where each control can be parented. I suggest that the ik arm and leg controls should be hooked to the parent\_control. I have been naming this hook “world”. Please refer to the naming convention documentation for more info.
3. Try this example. Select the ik arm control, and then select the parent control. In the animation UI, hit the space switch button. Make sure maintain offset is checked on and set key frame is off then accept.
4. Repeat the previous with any controls that require space switch targets.
5. Go through all of the animation controls and adjust the color and scale of those controls.
6. Paint skin weights. I suggest switching all the controls to an fk type.

Notes:

It is not always possible to zero an animation control. This is due to the intended dynamic nature of the system. Fk controls can always be zeroed out.

You can expect several changes to this process in future releases. I acknowledge that the process is not a smooth or automated as it could be. Consider this version an exploration of where we want the tool to go and some possible ways to get there.

### Setup

The final task in character creation is to crank out the two file which will be referenced in by animation and export files.

1. Click the Setup button.
2. You can specify a new name for the setup. This is useful if you are creating a variant.
3. Press “Export Game Assets”.
4. Feel the pride of a job well done.

You will now have two new files. “character/rig/setup/chr\_someguy.ma” and “character/rig/someguy.ma”

### Reference UI

Press the Reference button. This will give you a list of characters available for referencing. Choose a character and click “reference character”. You now get to choose a setup in case you want to load a variant. Choose the setup and click “reference character” . Start animating or create your final export file.

## More Geppetto Resources

[Naming conventions.](file:///\\corp-nas01\TAP\docs\GEPPETTO\Geppetto_NamingConventions.docx)

[Geppetto Feature and Bug Tracking](file:///\\corp-nas01\TAP\docs\GEPPETTO\GeppettoFeatureAndBugTracking.xlsx)