

MakeHuman© - 1.0 – Alpha 1

User's Guide

Released 10th April 2009

Table of Contents

1. Introduction.....	4
1.1 The New GUI.....	5
1.2 The Mesh.....	5
1.3 Python Scripting.....	5
1.4 External Rendering.....	5
2. Installation of MakeHuman.....	6
2.1 Windows.....	6
2.2 Mac OS X.....	6
2.3 Linux.....	6
3. Graphical User Interface Controls.....	7
3.1 Default Display.....	7
3.2 Controlling the Display.....	7
3.3 Main Toolbar (Changing Modes).....	8
3.4 Mode-Specific Toolbars.....	9
4. Anatomical Controls.....	10
4.1 Granularity Selector.....	11
4.2 MacroDetails Tab.....	11
4.2.1 Tetrawidget Toolbar.....	11
4.2.2 Ethnic Mixer.....	12
4.3 Details Tab.....	12
4.3.1 Tetrawidget Toolbar.....	12
4.3.2 Move, Scale Body Parts.....	13
4.3.3 Symmetry Controls.....	13
4.4 MicroDetails Tab.....	13
4.4.1 Move, Scale Anatomical Features.....	13
4.4.2 Symmetry Controls.....	14
4.5 Body Part List.....	14
4.6 Individual Features List.....	14
5. Pose Controls.....	15
5.1 The Pose View.....	15
6. File Options.....	16
6.1 Load Button.....	16
6.2 Save Button.....	16
6.3 Export Button.....	17
6.4 Temporary Export Features.....	17
6.4.1 Obj, BVH Export Workarounds.....	17
6.4.2 POV-Ray Export Workaround.....	17
7. Rendering Options.....	18
7.1 Aqsis.....	18
7.2 Pixie.....	18
7.3 POV-Ray.....	18
8. Application Programming Interface.....	19
9. Plugins.....	22
10. The Mesh.....	23

11. Licensing.....	24
11.1 Overview.....	24
11.2 The MakeHuman Application.....	24
11.3 MakeHuman Source Code Files.....	24
11.4 Other MakeHuman Application Code.....	24
11.5 The MakeHuman Mesh.....	25
11.6 Documentation.....	26

1. Introduction

This Alpha 1 release is intended to provide users with a view of a subset of the features that will appear in the full 1.0 release later this year. **This is not a full-function release** and, most notably, does not provide functionality to allow the figure to be posed. It does illustrate the new mechanisms by which the figure can be modelled for ethnicity, gender and age. More complete releases are expected to follow soon. The purpose of this Alpha 1 release is to encourage feedback on the features and their useability via the MakeHuman community forums. This feedback will enable the team to improve the application. This is an Open Source project and depends entirely upon community members contributing their time and ideas for free.

MakeHuman© is a free interactive modelling tool for creating custom 3D human characters. These characters can be modelled very quickly and can then be exported and used with many other modelling and rendering programs to incorporate realistic human figures into computer generated images and animations. The export feature supports the 'obj' format (amongst others), which is widely compatible with other renderers and modelling software. Features that make this software unique include the tetra-parametric GUI© and the Natural Pose System©, for advanced muscular simulation.

The home page for MakeHuman© is at <http://www.makehuman.org/> which provides links through to download software and updates, to the MakeHuman community forums and other community resources. The MakeHuman project is an open source project hosted on sourceforge at <http://sourceforge.net/projects/makehuman/>. Documentation for the project, including the FAQ (Frequently Asked Questions list) is maintained on the MakeHuman Wiki at <http://makehuman.wiki.sourceforge.net/>.

This document is the **Users Guide** for the MakeHuman product and covers its installation and use. This document is maintained on the MakeHuman Wiki at http://makehuman.wiki.sourceforge.net/UG_Introduction. When accessing this site, be aware that the 'active' Wiki version of this document corresponds to the 'next' release of MakeHuman that is currently under development and may differ slightly from the Users Guide that corresponds to the version of MakeHuman that you currently have installed.

A 'Quick Start' document is also available at http://makehuman.wiki.sourceforge.net/Quick_Start, which covers many of the same topics as this Users Guide, but in a more succinct format, that may be more suitable for those accustomed to not reading documentation. MakeHuman is designed to be consistent and intuitive to use. Most people with prior experience of graphical editors and modellers have found the Quick Start document sufficient to enable them to use the product (simply dipping into this reference as required to answer specific technical questions).

In later releases standard installers will be available for Windows, Mac OS-X and Linux. For this Alpha 1 release a compressed archive is available for:

- Windows
- Linux (Ubuntu 8.10)

To install the software on any other platform you are likely to need to build a copy from the source files. This is not particularly complicated because the application has been written in a highly portable manner, but you will need a platform-dependant compiler and some understanding of how to compile and build applications. This process is not covered in this document, but is covered in the [Developers Guide](http://makehuman.wiki.sourceforge.net/DG_Introduction) which is available on the MakeHuman Wiki at http://makehuman.wiki.sourceforge.net/DG_Introduction.

1.1 The New GUI



The GUI incorporates modelling controls based upon Ethnicity, Gender, Age, Muscle Tone and Body Mass. It also provides tools to enable you to model key body parts and individual anatomical features. This Alpha 1 release does not allow you to pose the figure. The full release will support the posing of figures. This version does not include props such as clothing or hair and does not yet contain animation features.

The current Alpha 1 release supports the loading and saving of modelled figures and provides some temporary export functions. The export capabilities will be extended in later Alpha and Beta testing releases.

This GUI incorporates a small number of standardised 'intelligent' tools designed to minimise the learning curve for new users while providing powerful features to enable all users to rapidly model a character that meets their needs. Knowledge has been built into the tools so that, for example, if a female figure is being modelled (as defined by the Gender tool), then any body mass added to the model will accumulate fat in those areas of the body where a woman typically accumulates fat.

1.2 The Mesh



All MakeHuman humanoid figures are based on a single, highly optimized, light and professional mesh. Modelling of the mesh is performed by deforming the mesh rather than altering its topology. The mesh has been through a series of iterations to improve the structure so that deformations can be realistically applied while maintaining a low polygon count to minimise processing overheads. The mesh supports subdivision to enable higher density, smoothed meshes to be exported for high quality rendering. A considerable number of mesh deformation targets have been created by artists to provide you with a large number of realistic starting points from which to model particular ethnic, gender, age and body mass figures of your own design.

1.3 Python Scripting



The MakeHuman application has been structured to expose a great many of the program internals through the Application Programming Interface (API). This open structure has been documented and published to encourage the development of new scripts and plugin functionality that will help the application to develop and rapidly adapt to the needs of the user community.

1.4 External Rendering



MakeHuman incorporates a range of plugins to export a modelled and posed figure in 3D graphics formats supported by a very wide range of external modellers and rendering engines. These export functions have been written as Python plugins that can easily be extended and can serve as examples to enable other interfaces to be readily constructed. Support for the Renderman format is built into the MakeHuman application. Aqsis is the officially supported Renderman format renderer, but export to other formats and other renderers is also provided.

2. Installation of MakeHuman

In the full release of MakeHuman installers will be available for Windows, Mac OS X and Linux. These installers will not be available until later phases of testing. Until then, compressed archives will be available for Windows and Linux which will need to be unzipped.

You will need about 30 MB of disk space for the zipped archive and a further 100MB of disk space for the application code. Once installation is complete you can delete the zip file to recover the disk space that it occupies. It is recommended that you run MakeHuman on a machine with at least 512MB of memory and at least a 800MHz processor.

If you don't have Windows or Linux, you may still be able to build, run and use this application. The MakeHuman source code is available to enable you to perform a build of your own. This requires a bit more effort, additional knowledge and compilation tools. Considerable information on building a copy of MakeHuman can be found in the Developers Guide on the MakeHuman Wiki at http://makehuman.wiki.sourceforge.net/DG_Introduction .

2.1 Windows

The Windows version needs the Microsoft Visual C++ [2008 Redistributable Package](http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf) to be installed because MakeHuman is compiled with Python 2.6 and without these libraries the application can crash at startup. To check whether you already have this package installed you can open the Control Panel and select 'Add/Remove Programs' ('Programs and Features' in Vista) and look for Microsoft Visual C++ 2008 Redistributable. If you need to install this package you can download a free Microsoft installer from <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf>.

Earlier versions like the 2005 redistributable may also work, but if you get problems, update to 2008.

To install the MakeHuman files, download the MakeHuman Windows archive file and unzip it into a separate directory on your file system. Double click on *makehuman.exe* to start the application.

2.2 Mac OS X

The Alpha 1 code is not currently available as a compressed archive for Mac OS X due to lack of maintainers.

2.3 Linux

A zipped package is available for Ubuntu 8.10.

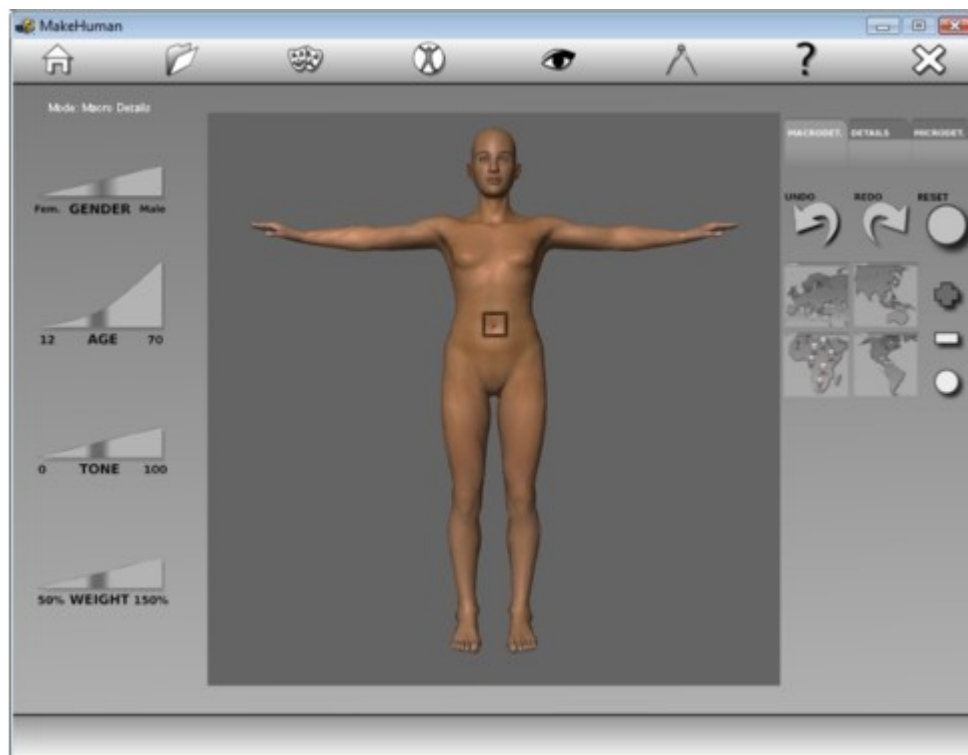
Download the archive and unzip it into a separate directory on your file system.

Double click on the makehuman binary to start the application.

3. Graphical User Interface Controls

3.1 Default Display

When you first launch the MakeHuman application you should see the default human figure facing you with outstretched arms. As illustrated in the screenshot on the right, the figure should appear in the centre of the window and a set of toolbar buttons should be displayed along the top of the window to enable you to change modes. This is the modelling mode screen and enables you to control the anatomy of the figure (see Section 4, 'Anatomical Controls' below). You can return to this screen at any time by pressing the Home button on the toolbar (far left).



Various other controls are displayed on the left hand side of the screen and there is a bottom tool bar where buttons specific to the current mode can be displayed (empty in this image).

The small square box symbol shown in the middle of the figure's stomach in the image above is the cursor and should follow your mouse movements.

3.2 Controlling the Display

The human figure is displayed in 3D. You will need to get familiar with how to navigate around in the 3D environment, turning the model, zooming the camera in and out and panning around the window because you'll be doing this a lot as you design and pose your human figure. These basic navigation functions can be performed using either hotkeys on your keyboard or by using your mouse (or graphics tablet). A 3 buttoned mouse is recommended.

The numeric keys, along with the '+', the '-' and the '.' keys and the up, down, left and right keys on your keyboard are used as hotkeys. The numeric key settings make most sense if you have a numeric pad on your keyboard. If you don't then you may still find it useful to use the ordinary numeric keys to perform operations that are not available using the mouse. For example resetting the display to fixed views.

Certain other hotkeys are available such as 'Ctrl Z' and 'Ctrl Y' for undo/redo.

Rotating the Figure:

- Left click the mouse and drag up, down, left, right or press 8, 2, 4 and 6 on the keyboard.

Zooming In and Out:

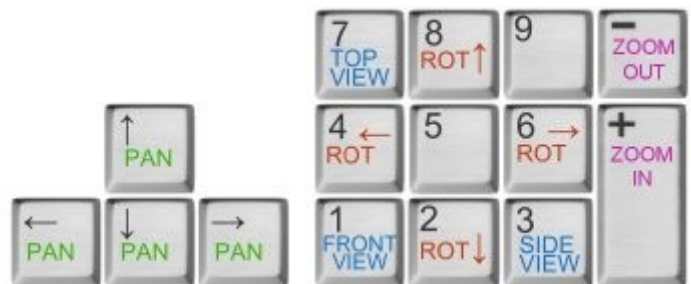
- If you have a central wheel on your mouse this can usually be configured to zoom in and out, otherwise you'll need to use the '+' and '-' keys on your keyboard. Pressing the '.' character resets the zoom (and removes any panning to point the camera at the centre of the figure).

Panning:

- Right click and drag or use the up, down, left and right keys to drag the contents of the viewing window around, effectively panning the camera. Pressing the '.' character returns the figure to the centre of the window (and resets any zoom).

Preset Positions:

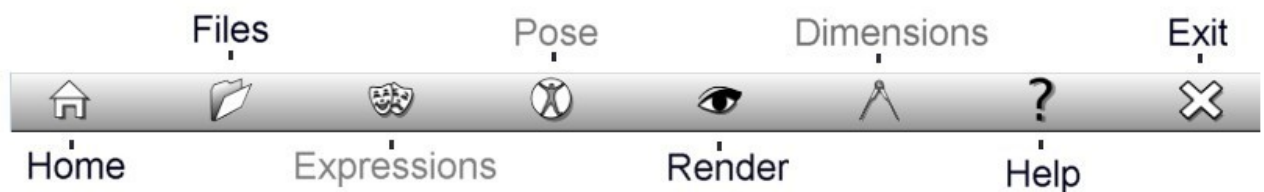
- Pressing '1' returns to the default front facing view, '3' displays a side view and '7' an arial or top view. These keys do not reset the zoom, or panning, so press '.' if you need re-centre the camera.



Undo/Redo:

- Pressing 'Ctrl Z' can be used to undo the previous operation (change to ethnicity, gender, age, etc.)
- Pressing 'Ctrl Y' redoes an operation undone by using 'Ctrl Z'.

3.3 Main Toolbar (Changing Modes)



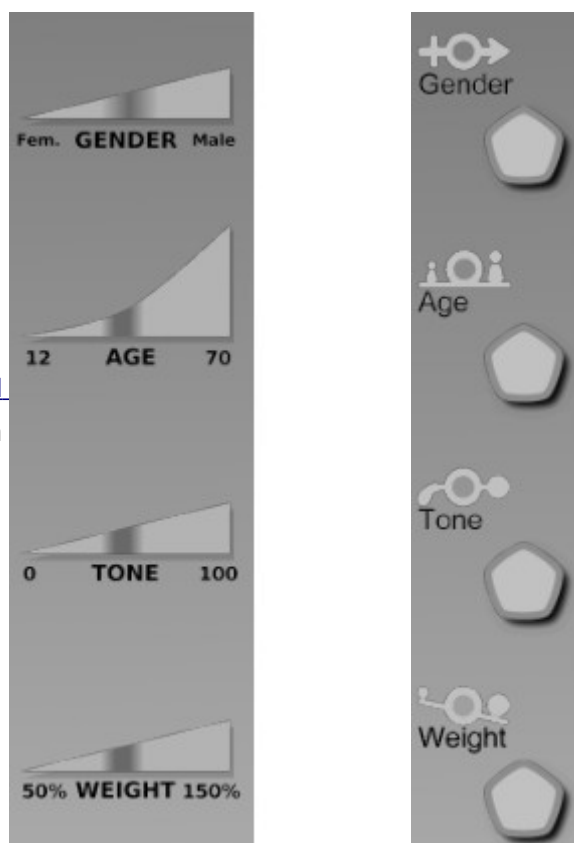
The main toolbar along the top of the MakeHuman window allows you to change mode. Each mode supports a particular category of function and provides controls and options specific to that function. The main functions and the controls and options that they provide are detailed in their own sections of this document. The following paragraphs on this page provide a very brief summary.

- The **Home** button returns you to Modelling Mode (see [Anatomical Controls](#)).
- The **File Mode** button changes to File Mode to enable file interactions (see [File Options](#)).
- *Expression Mode is not supported in this Alpha 1 release.*
- *Pose Mode is not supported in this Alpha 1 release.*
- The **Rendering Mode** button changes to Render mode to access external rendering applications (see [Rendering Options](#)).
- *Dimensions Mode is not supported in this Alpha 1 release.*
- The **Help** button displays a PDF copy of this document..
- The **Exit** button terminates the MakeHuman application.

3.4 Mode-Specific Toolbars

When you change modes, various toolbars containing buttons to perform functions specific to the mode you have selected are displayed.

Modelling Mode Tetrawidget (see [Anatomical Controls](#)) The specific tools displayed depend upon which 'level of detail' tab is selected.



File Mode Toolbar (see [File Options](#))



Expression Mode Toolbar (Not Supported in this Alpha 1 release)

Pose Mode Toolbar (Not Supported in this Alpha 1 release)

Rendering Mode Toolbar (see [Rendering Options](#))



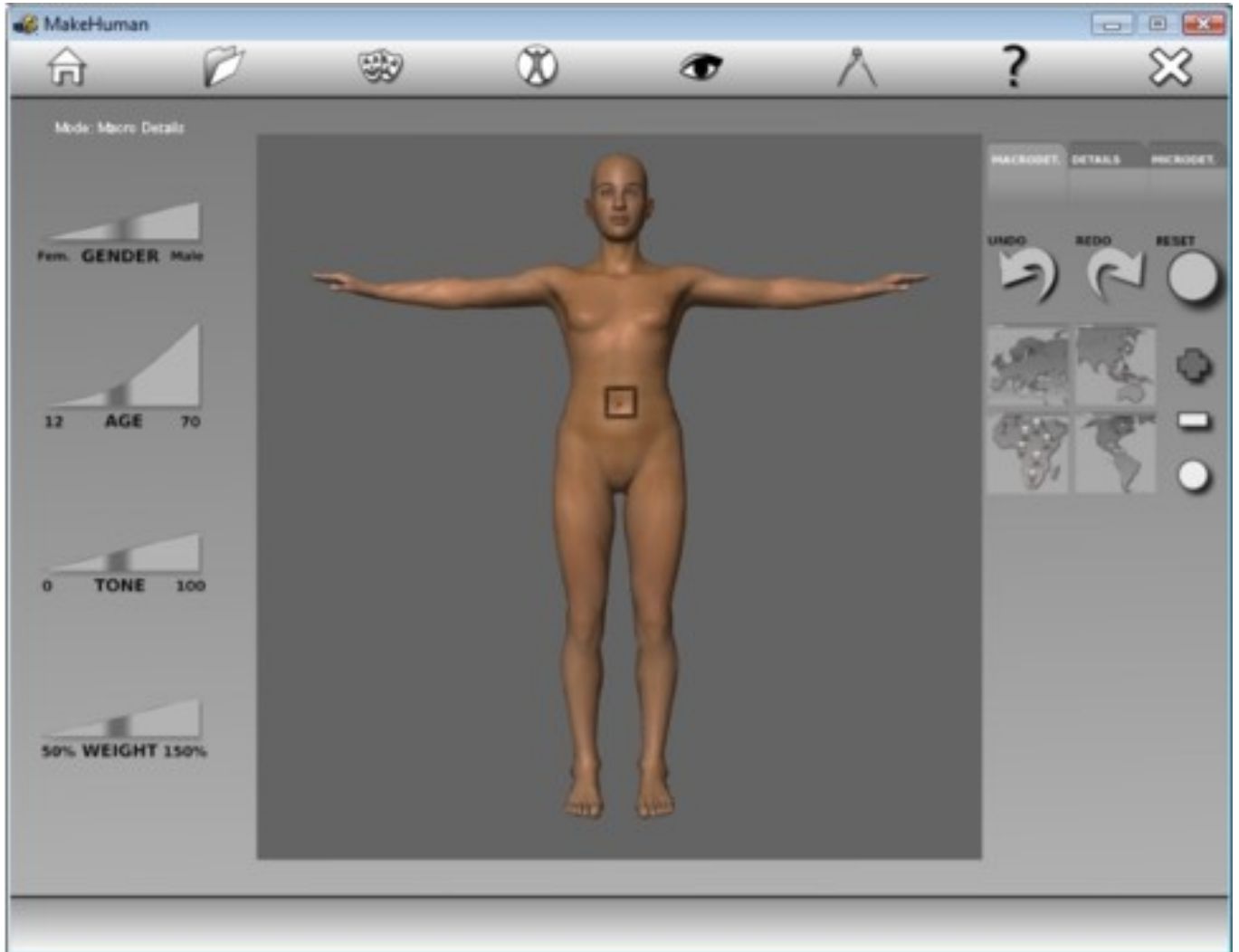
Dimensions Mode Toolbar (Not Supported in this Alpha 1 release)

Application **Help**

Opens separate PDF Reader

4. Anatomical Controls

The 'Home' button on the top toolbar takes you into Modelling Mode. This is the default mode that appears when the application starts. This mode is used to control the anatomy of the human figure. MakeHuman provides controls ranging from course-grain to fine-grain for defining anatomy.



A common modelling technique is to model generic attributes first, such as age, ethnicity and gender and then to work down into greater detail, modelling individual body parts and then specific detailed features. MakeHuman is structured to make it very easy to follow this pattern of working.

Internally the application uses a significant number of ethnic characteristics that have been modelled by hand with corresponding gender and age variants. When you use the MakeHuman application to model a specific figure, the application uses a combination of these hand-modelled variants to create a specific model that meets your specifications, allowing you to easily generate a unique character. You can also apply generic variants for muscle tone and for weight.

Changes made using the various anatomical controls can be reversed by using the *Undo* button on the right of the screen (or Ctrl-Z on your keyboard). Undone changes can be re-applied using the *Redo* button (or Ctrl-Y on your keyboard). Changes can be reset to their initial values using the *Reset* button.



4.1 Granularity Selector

The granularity selector provides 3 tabs located at the top of the right hand side of the Modelling page. These tabs enable you to switch from course-grain to fine-grain control and back again. Each of the 3 tabs contains controls and tools designed to manipulate the model at the level of granularity selected.



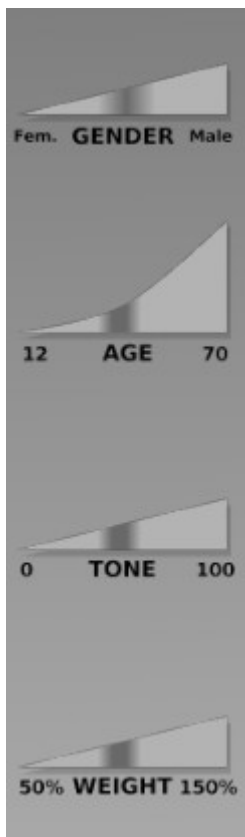
Initially you will find the *MacroDetails* tab selected, which applies changes to the whole body, but you can select the *Details* tab to apply changes to discrete body parts, such as the head, the upper arm, the forearm, etc. or the *MicroDetails* tab to apply changes to more detailed body parts corresponding to individual anatomical features, such as the nose tip, the left nostril, the right cheek, the lower mouth etc. Note that applying characteristics to the whole body takes a fair bit longer than applying changes to individual body parts and features, so you're likely to see the timer a bit more frequently.

A list of the selectable body parts is given below. A list of the individual anatomical features that can be selected using the *MicroDetails* tab is also given below. These individual features are also illustrated online at <http://makehuman.sourceforge.net/FaceGroups/HeadFaceGroups.html> as they correspond to the face groups used by the application.

4.2 MacroDetails Tab

The MacroDetails Tab is initially selected and allows you to manipulate the whole body of the figure.

4.2.1 Tetrawidget Toolbar



The controls on the left hand side of the Modelling page allow you to adjust the gender, age and the physique of the entire figure. Clicking on the slider for a control moves the setting to the position you click.

You can still use the mouse to rotate/pan/zoom, but in this Alpha 1 release you'll need to click the grey area around the figure to do this once a Tetrawidget tool has been selected.

Gender

The Gender button allows you to make the figure more or less masculine, adjusting the general form of the entire body, generally making body parts more angular for the masculine figure and smoother for the female figure.

Age

The Age button controls aspects of the anatomy that characterise age, with older characters tending to attract looser skin and more wrinkles.

Muscle Tone

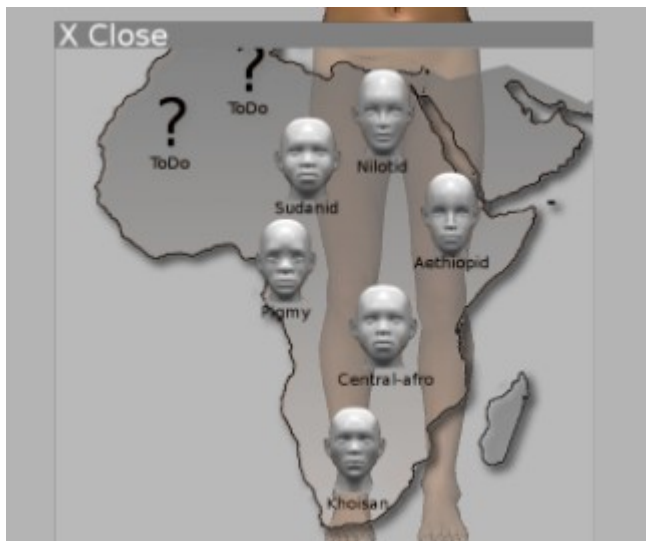
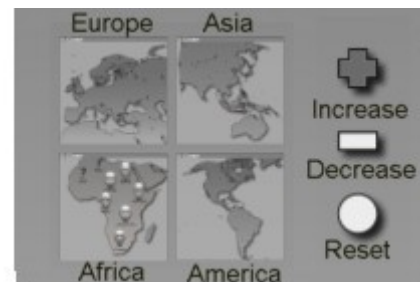
The Muscle Tone button controls how full and muscular the figure is, with the higher settings transforming the character into a more muscular individual.

Weight

The Weight button controls how lean or overweight the figure is.

4.2.2 Ethnic Mixer

The world map shown on the right hand side of Modelling Mode page enables you to select and mix together different ethnic characteristics. This *Alpha 1 release* only contains a small collection of the ethnic groups that will be available in the full release. The groups available in this *Alpha 1 release* are all African.



When you click one of the continents on the small graphic, a larger map of that region is displayed (remember that only the African map is activated in this *Alpha 1 release*). The larger map contains icons illustrating the facial characteristics of each of the different ethnic groups available from that region. The icons are arranged so that they correspond to the characteristics traditionally associated with that region.

The buttons to the right of the small graphic allow you to select whether you wish to increase or decrease the proportion of characteristics inherited from the ethnic group that you click. The reset button allows you to remove the influence from that ethnic group on the current figure. To adjust the proportion of the features inherited from each ethnic group first click the 'increase', 'decrease' or 'reset' button, then click the facial icons on the larger map as many times as you need to apply the amount of change you wish to apply.

To indicate that you have finished editing this characteristic press the '*Close*' button at the top left corner of the enlarged graphic. Once you have set the ethnicity in this way the areas that you have used to build the 'ancestry' of your character will be shown on the map using darker colors to indicate more influence. The message line near the top of the screen also gives you a numerical breakdown of the ethnic proportions of your figure.

4.3 Details Tab



The Details Tab allows you to manipulate discrete body parts, such as the head, the upper arm, the forearm, etc. The selected body part is shown using highlight colors.

4.3.1 Tetrawidget Toolbar

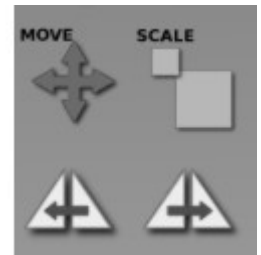
The controls on the left hand side of the Modelling page allow you to adjust the gender, age and the physique of individual body parts. Once you have selected a button you can click and drag the mouse on a particular body part to increment or decrement the selected feature. You can still use the mouse to rotate/pan/zoom, but you'll need to click the grey area around the figure to do this once a Tetrawidget tool has been selected.



4.3.2 Move, Scale Body Parts

There are buttons on the right of the screen that enable you to move or scale individual body parts.

To move or scale a body part, click the '*Move*' or '*Scale*' button then click and drag the mouse on a particular body part that you wish to resize or move. You can continue selecting different body parts, performing the same operation on them without selecting the '*Move*' or '*Scale*' buttons again unless you wish to change operations.



4.3.3 Symmetry Controls

The triangular buttons with the arrows on them allow you to apply right-to-left or left-to-right symmetry to the Move or Scale settings of the selected body part. That is to say that the settings from the corresponding body part on one half of the body are mirrored across to the body part on the other half of the body. These operations necessarily break the general convention of only modifying the highlighted body part as they can be used to modify a body part on the opposite side of the body.

4.4 MicroDetails Tab

The *MicroDetails* tab allows you to manipulate individual anatomical features, such as the nose tip, the left nostril, the right cheek, the lower mouth etc. The selected body part is shown using highlight colors.

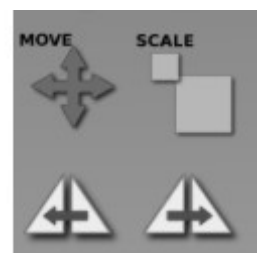
Notice that when you select an individual feature it may initially appear very similar to when you select a body part using the controls in the Details Tab. For example, if you click on the upper left leg using the standard front view then the regions selected seem virtually identical. However, if you turn the model you will see that only the front of the leg is selected when using the MicroDetails Tab modelling tools.

One consequence is that, when you try to scale the selected anatomical feature the system responds differently. When using the Details Tab vertically scaling the upper left leg changes the length of the whole leg, moving the lower left leg down to make space for it. When using the MicroDetails Tab to scale just the front of the upper left leg, vertical scaling is constrained by the length of the back of the upper left leg.

4.4.1 Move, Scale Anatomical Features

There are buttons on the right of the screen that enable you to move or scale individual anatomical features. Be aware that the amount by which you can move or scale those features may be tightly constrained by the size and position of adjoining anatomical features.

To move or scale a feature, click the '*Move*' or '*Scale*' button then click and drag the mouse on a particular anatomical feature that you wish to resize or move. You can continue selecting different anatomical features, performing the same operation on them without selecting the '*Move*' or '*Scale*' buttons again unless you wish to change operations.



4.4.2 Symmetry Controls

The triangular buttons with the arrows on them allow you to apply right-to-left or left-to-right symmetry to the Move or Scale settings of the selected anatomical feature. That is to say that the settings from the corresponding anatomical feature on one half of the body are mirrored across to the anatomical feature on the other half of the body. These operations necessarily break the general convention of only modifying the highlighted anatomical feature as they can be used to modify an anatomical feature on the opposite side of the body.

4.5 Body Part List

This *Alpha 1 release* supports the selection and manipulation of the following body parts which can be independently selected when using 'Body Parts' selection in Modelling Mode:

- **Head** - Eyes, Jaw, Nose, Ears, Mouth, The rest of the Head
- **Neck** - Whole Neck
- **Torso** - Hips, Pelvis, Upper Torso including Shoulders
- **Arms** - Left Upper Arm, Right Upper Arm, Left Forearm, Right Forearm, Left Hand, Right Hand
- **Legs** - Left Thigh, Right Thigh, Left Calf, Right Calf, Left Foot, Right Foot

4.6 Individual Features List

Certain detailed anatomical features can be selected and individually manipulated in this *Alpha 1 release* when using 'Individual Features' selection in Modelling Mode. The following list of individual anatomical features are likely to be selectable in the full MakeHuman Version 1.0 Release:

- **Head** - a large number of individual muscle groups on the face can be selected. There are 4 separate zones around the scalp that can be independently selected. The ears consist of 3 independently selectable sections.
- **Eyes** - Each eye can be selected.
- **Mouth** - Each of the teeth in the upper and lower jaw can be selected and the tongue can be selected.
- **Neck** - The upper and lower neck, and the Adam's apple can be selected.
- **Torso** - Various muscle groups and regions of the torso can be independently selected.
- **Arms** - Individual muscle groups on each arm can be selected.
- **Hands** - The palms and each segment of each finger and thumb can be individually selected on either hand, as can individual finger nails.
- **Legs** - Individual muscle groups on each leg can be selected.
- **Feet** - The heels, cores and each segment of each toe can be individually selected on either foot, as can individual toe nails.

5. Pose Controls

(Not Supported in this Alpha 1 Release)

A key part of the functionality of the MakeHuman application will be to support the posing of characters (comparable with the sort of posing of characters available in the earlier MakeHuman 0.9.1 release candidate). This functionality is not available in this *Alpha 1 release*.

The default position has the human figure standing facing straight towards the camera with arms outstretched. Currently this is the only pose supported. In the final release it is anticipated that there will be a library of predefined poses which can be loaded to cause the character that you have specified to adopt that pose. The Pose Mode screen will then allow you to select individual joints and adjust the settings appropriate to that joint. Once you have defined a pose you can save it and subsequently apply the same pose to other figures.

5.1 The Pose View

To enter the 'Pose' view, you will be able to select the 'Pose' icon from the toolbar. This view will contain a 'map' of the body enabling you to select a joint to work on. Once a joint is selected a TetraWidget specific to that body part will appear, enabling the settings for that joint to be manipulated.

Head

Neck

Left and Right Collar

Left and Right Upper Arm

Left and Right Lower Arm

Left and Right Hand

Pivot

Left and Right Upper Leg

Left and Right Lower Leg

Left and Right Foot

Fingers and Thumbs

Toes

Eyes

Mouth

6. File Options

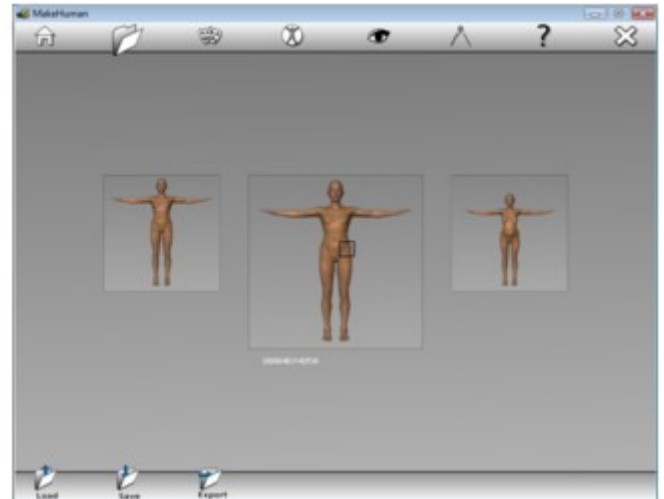
The File Mode screen provides buttons to Load, Save and Export files. Although the *Export* button does not do anything in this Alpha 1 Release, various workaround options for exporting files are present within the application and are described below.



6.1 Load Button

When you load a new model into MakeHuman, unsaved changes to the current model are lost. No 'unsaved changes' warnings are displayed, so it's up to you to make sure that you save any changes before using this function.

When you press the 'Load' button a model browser is displayed, showing model files stored in the 'models' directory of your MakeHuman installation. These are usually files that you have previously saved.



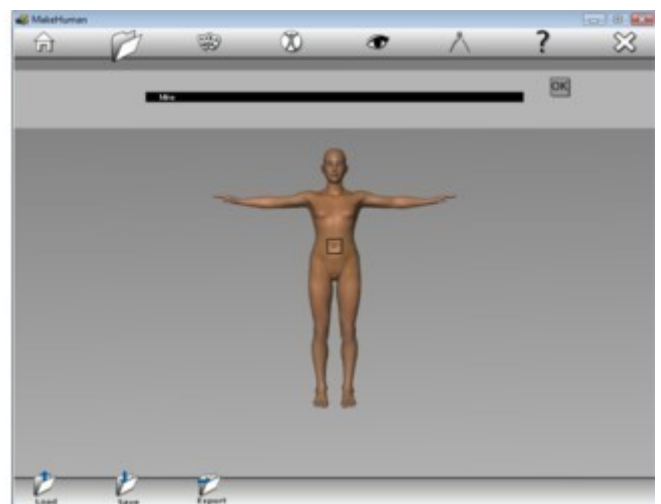
The file browser initially shows the start of a chain of model files with the first file in the centre and the next file in the chain off to the right. Pressing 'Enter' or clicking on the currently proposed model in the centre loads that model into MakeHuman so that you can work with it. Pressing the right arrow or clicking on the model file to the right moves that one into the centre and the previous file off to the left. Pressing the left arrow or clicking on the model file to the left moves you along the chain in the opposite direction.

Holding down the left or right arrow keys scrolls through the chain.

6.2 Save Button

When you hit the Save button, the currently loaded model is displayed, above which is a black text field into which you can type a file name and tagging information to help you to subsequently find your model.

Note that the first word that you enter will be used as the file name. All of the words will be added as tags into the file itself enabling you to perform tag-based searches (currently only through external text search functions). When satisfied with the tagging information you need to either press 'Enter' or click on the 'OK' button next to this field to actually save the file.



MakeHuman saves the anatomy and pose settings (this Alpha1 Release doesn't support Pose settings) of the current figure into a MakeHuman format file in the 'models' directory on your file system. This 'models' directory is located in your MakeHuman installation directory.

The file name uses the 'mhm' extension.

6.3 *Export Button*

The Export button is not active in this Alpha 1 Release, but certain workarounds for exporting files are described below (see Temporary Export Features).

Exporting a model generates a file in a different format suitable for use with many other modellers and rendering engines. Note that files generated in other formats using these export features can lose important MakeHuman contextual information that is not supported by those formats, so usually cannot be read back into MakeHuman. It is therefore important to always also save your work in MakeHuman format so that you can subsequently modify the figure.

Further information on export features is available on the [Export Options](#) page.

6.4 *Temporary Export Features*

This Alpha 1 release incorporates certain export options that are available through temporary means until the Export feature on the File Mode screen becomes available:

6.4.1 *Obj, BVH Export Workarounds*

This *Alpha 1 release* includes an experimental 'obj' export capability that can be run from any mode by pressing the "e" key on your keyboard. This exports a wavefront object (and the corresponding material layer). It also exports a BVH file (a BioVision Hierarchy file) that can be imported on Blender to build a skeleton. All of these files are saved using hard-coded file names in the same folder as the MakeHuman executable is stored.

6.4.2 *POV-Ray Export Workaround*

The POV-Ray rendering feature on the rendering screen exports files to render them. See the Wiki page at http://makehuman.wiki.sourceforge.net/UG_POVRay_Export.

7. Rendering Options

Three rendering options are provided by a toolbar at the bottom left of the 'Rendering' mode screen.



7.1 Aqsis

Aqsis is an Open Source rendering application that implements the [Renderman®](http://www.renderman.com/) standard. You can see examples of its capabilities and download the software from <http://wiki.aqsis.org/> . MakeHuman provides an option to directly render an Aqsis compatible Renderman format file containing a posed and morphed humanoid model.

The files generated by using this render option are written into the 'renderman_output' directory on your file system.

7.2 Pixie

Pixie is an Open Source RenderMan renderer for generating photorealistic images released on a GNU Lesser General Public License ([LGPL](http://www.gnu.org/licenses/lgpl.html)). You can see examples of its capabilities and download the software from <http://www.renderpixie.com/> . MakeHuman provides an option to directly render a Pixie compatible Renderman format file containing a posed and morphed humanoid model.

The files generated by using this render option are written into the 'renderman_output' directory on your file system.

7.3 POV-Ray

The Persistence of Vision Raytracer (POV-Ray) is a popular and freely available rendering application capable of generating stunning images and animations. You can see some of the images that it is able to generate and download the software from <http://www.povray.org> .

MakeHuman provides an option to directly render a POV-Ray 'include' file containing a posed and morphed humanoid model. This option also exports a pigment map and generates a sample 'scene' file that contains a series of examples illustrating a range of ways of using the model. These examples and further options for exporting and rendering figures are explained in more detail on the Wiki page at http://makehuman.wiki.sourceforge.net/UG_POVRay_Export.

Two Tutorials are available to walk you through the POV-Ray rendering and export features:

Introductory Tutorial: http://makehuman.wiki.sourceforge.net/POVRay_Tutorial1

Simple Clothing Tutorial: http://makehuman.wiki.sourceforge.net/POVRay_Tutorial2

The files generated by using this render option are written into the 'pov_output' directory on your file system.

8. Application Programming Interface

The core OpenGL 3D engine for MakeHuman is written in C, but the large majority of the functional behaviour is implemented using Python, which is an Open Source Object Oriented programming language.

Python is an extremely 'readable' language and lends itself to the development of enhancements and plugins to extend the capabilities of MakeHuman.

The MakeHuman installation applications for Windows, Mac OSX and Linux incorporate everything you need to run the MakeHuman application, create 3D human models and run Python plugins. To develop Python plugins or other code that interfaces with MakeHuman you will need to be familiar with the Python language and Python development techniques. You may also choose to install a full Python development environment, possibly including an Interactive Development Environment. The Python home page can be found at <http://www.python.org/> where you can download the full free Python development environment and access Python community resources and documentation.



The MakeHuman application itself contains a Python interpreter. If you write a Python script called 'foo.py' you can invoke it with the command "makehuman foo.py".

The MakeHuman Python Application Programming Interface (API) is fully documented at <http://makehuman.sourceforge.net/API/> . The following table presents a brief description of the functionality that the main MakeHuman Python scripts expose as an API. The application data structures are also contained within the class definitions for the Python code. Information about the evolution and design of the MakeHuman mesh is contained on the Wiki at http://makehuman.wiki.sourceforge.net/The_Humanoid_Mesh . Each face on the MakeHuman mesh is assigned to a face group which is used for the selection of different anatomical features within the MakeHuman GUI. Images illustrating each of these face groups can be seen in the face group documentation at <http://makehuman.sourceforge.net/FaceGroups/index.html> .

Module/Application	Description
main.py	The <i>Main Application</i> handling the interaction with the user and calling functionality appropriate to the actions initiated by the user. A <i>Python Module</i> in the 'mh_core' directory containing all of the base classes needed to manage the 3D data structures at runtime. This includes the data structures themselves as well as methods to handle the manipulation in memory of those data structures. For example: the Vert class defines the data structures to hold information about mesh vertices objects, the Face class defines data structures to hold information about mesh face objects. These base classes implement a nested heirarchical structure for the objects that go to make up the scene that is shown to the user. For example, a FaceGroup object contains groups of mesh face objects as defined by the Face class. An Object3D object contains all of the FaceGroup objects that go to make up a particular discrete object, such as the eyeball or the head. The Scene3D object contains all of the Object3D objects that go to make up the entire scene.
module3d.py	

files3d.py	<p>A <i>Python Module</i> in the '<i>mh_core</i>' directory that handles 3D file formats supported by MakeHuman. This module will support integration with functions to handle most common 3D file formats. This includes functions to transpose imported 3D data into a standard internal format for each 3D file format supported by the import functionality and functions to generate 3D data structures that correspond to 3D file formats supported by the export functionality. It also includes generic transformation utilities such as the <code>dataTo3Dobject()</code> function which takes an object defined in the standard internal format and makes it visible to the user.</p>
algos3d.py	<p>A <i>Python Module</i> in the '<i>mh_core</i>' directory containing all of the algorithms used to perform high-level 3D transformations on the mesh that is used to represent the human figure in the MakeHuman application. These include morphing for anatomical variations and pose deformations, collision deformations, etc. This module also contains a number of functions used during the development cycle to test and to visualize the mesh.</p>
aljabr.py	<p>A <i>Python Module</i> in the '<i>mh_core</i>' directory containing a set of the most common 3D algebraic operations used in MakeHuman. These are mostly the vector and matrix operations core to any 3D application.</p>
textures3d.py	<p>A <i>Python Module</i> in the '<i>mh_core</i>' directory containing a series of functions to perform standard processes on bitmaps. These functions provide simple higher level functionality to other functions, such as the 3D algorithms in <code>algos3d.py</code>.</p>
widgets3d.py	<p>A <i>Python Module</i> in the '<i>mh_core</i>' directory containing a series of classes that implement GUI utility functions.</p>
guitoolbar.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the application toolbars that provide access to the various MakeHuman operating modes and to the functionality available in each mode.</p>
guicommon.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the 'guicommon' class structures and methods to support functions common to the different operational modes made available through the MakeHuman GUI toolbar.</p>
guimodelling.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the modelling functionality available in Modelling Mode (the default mode and the Home option on the toolbar).</p>
guirender.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the rendering functionality available in Render Mode. This functionality connects through to the various external rendering applications supported by MakeHuman.</p>
guifiles.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the file handling functionality available in Files Mode. This functionality supports the loading of texture and morph target files.</p>
guifileselector.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the file selector GUI object used in Files Mode.</p>
halfedges.py	<p>A <i>Python Module</i> in the '<i>mh_plugins</i>' directory that implements the data structures and functions to support vertex/face analysis using HalfEdges. This technique introduces two HalfEdges for each edge, each with opposing orientations aligned along the length of the edge. The module provides high-level functions to developers to support validation of data structures and interrogation of the mesh such as support for tracking around the edges of a single face or a hole in a mesh object and the identification of all polygonal faces sharing a particular vertex. These functions are particularly useful with the type of surface mesh used by MakeHuman where no more than two polygons should share an edge.</p>

metadataengine.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that implements a metadata search engine to rapidly locate records within library files.
subdivision.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that handles the subdivision of surfaces. Subdivision can be performed before exporting or rendering the model to smooth the mesh and improve the appearance of the finished image.
blendersaveobj.py	A <i>Python Utility</i> in the ' <i>utils</i> ' directory that saves a human figure in a blender compatible object file.
blendersavevertscolor.py	A <i>Python Utility</i> in the ' <i>utils</i> ' directory that saves MakeHuman vertex colour information in a blender compatible format.
mh2renderman.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that implements a function to export a human model in Renderman format and to invoke the Aqsis or Pixie rendering engine to render the Renderman format files.
mh2obj.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that implements a function to export a human model in Wavefront obj format.
mh2povray.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that implements a function to export a human model in POV-Ray format. POV-Ray is a Raytracing application (a renderer) that is free to download and use.
mh2bvh.py	A <i>Python Module</i> in the ' <i>mh_plugins</i> ' directory that implements a function to export a Biovision motion capture file in BVH format.

The MakeHuman Developers Guide at http://makehuman.wiki.sourceforge.net/DG_Introduction contains a description of how the application works and an API section that contains links to details of both the Python and the C application components.

9. Plugins



Each of the different application modes, along with the export and rendering functions to integrate with different renderers are coded as plugins and can serve as an example to plugin developers to illustrate how extra plugins can be implemented.

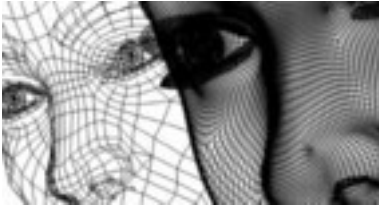
Further information about how to develop plugins to work with MakeHuman can be found on the *Plugin Development* section of the *MakeHuman Developers Guide* at http://makehuman.wiki.sourceforge.net/DG_Plugin_Development.

It is anticipated that future development will include conversion utilities to integrate with an extended range of modeller and rendering applications, feature enhancements, such as animation features, clothing, hair and prop integration and tools/utilities.

There are plugins under development for extended characters based upon alternative meshes, for example a 'toon' mesh for cartoon-like characters with exaggerated features. It is also possible to envisage plugins to support the modelling of other types of creature.

The Wiki version of this page at http://makehuman.wiki.sourceforge.net/UG_Plugins will be used as a reference page to accumulate information about plugins as they are developed.

10. The Mesh



The principal aim of the MakeHuman project is to develop an Open Source application capable of realistically modelling a very wide variety of human anatomical forms in the full range of natural human poses from a single, universal mesh.

Central to this is the design of a 3D humanoid mesh that can readily be parametrically manipulated and deformed to represent alternative anatomical characteristics while retaining and respecting a common structural skeleton that permits poses and the corresponding deformations to also be parametrically manipulated. This objective has been pursued to afford the artist the maximum degree of experimental freedom when using the software. It frees the artist from the artificial constraints that are inherent to a model that has pre-established gender or age.

By pursuing this aim the MakeHuman Team have developed a model that can combine different anatomical parameters to transition smoothly from the infant to the elderly, from man to woman and from fat to slim. The vast wealth of potential combinations provides the artist with an extraordinarily broad range of possibilities for artistic expression but presents many interesting problems to the development team. In particular it adds to the classical problems of 3D modelling (number of polygons, square or triangular faces, etc.) the problems of constructing a super mesh that can be transformed into any form of human while being sufficiently optimised to be able to be manipulated on desktop machines, yet still producing a professional quality of output. By contrast, most modellers only have to produce a single model that can be created and adapted for a single project.

The current MakeHuman mesh has evolved through successive iterations of the MakeHuman project, incorporating lessons learned, community feedback and the results of considerable amounts of study and experimentation. No generic mesh is perfect and this mesh has inevitably been subject to some compromise and will undoubtedly continue to be refined in future releases. Nevertheless, the current mesh represents a remarkable achievement and is a great source of pride for the MakeHuman team. The current iteration, known as the 'Zmesh' comprises a state of the art universal humanoid model. A paper describing the characteristics and capabilities of the mesh along with a brief history and discussions about potential future enhancements can be found on the MakeHuman Wiki at http://makehuman.wiki.sourceforge.net/The_Humanoid_Mesh.

11. Licensing

11.1 Overview

Licensing is a complex matter, particularly for software and data intended to be free and widely reusable, such as is the case with MakeHuman©. Most Copyright law is oriented heavily towards protecting commercial interests, so using this law to give you rights while protecting you from rights acquired by others is a delicate balancing act. The intention of the MakeHuman team is to license the reuse of its Copyright materials, free or charge, both commercially and non-commercially. Standard time-tested licenses have been used to attempt to bring clarity to this complex legal issue.

The MakeHuman Mesh licensing paragraph below has been particularly delicately worded so as to try to afford you extensive and wide ranging authority over creative works that you produce while attempting to avoid that the use of MakeHuman by others places unnecessary restrictions upon you. MakeHuman uses a standardised mesh topology which is deformed using standard deformation algorithms (morph targets). Files exported by different artists therefore often bear a great many similarities. It is the intention of the MakeHuman team that you should be able to copyright your own, sufficiently unique or distinct artistic works, without overly constraining other MakeHuman users who will be using very similar meshes.

11.2 The MakeHuman Application

MakeHuman© is an Open Source application that you can use and redistribute in original or modified form under the terms of the GNU General Public License as published by the Free Software Foundation.

Copyright© 2001-2009 is retained by the MakeHuman Team (makehuman.org) who grant you permissions to use released code under the conditions and terms of the [GNU General Public License 3.0](#) or (at your option) any later version of the GNU GPL.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

11.3 MakeHuman Source Code Files

Source code for both released versions of the MakeHuman application and for versions currently under development is also governed by the terms of the GNU General Public License as published by the Free Software Foundation. These source files are made available in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

11.4 Other MakeHuman Application Code

From time to time the MakeHuman Team may release other application code in various forms. Unless otherwise explicitly stated this application code is also covered by the GNU GPL. For example, the MakeTarget scripts for use with the Blender Modelling application are distributed under the GNU GPL and the MakeHuman morph target files

that they generate are covered by this license. Meshes generated or converted using any such tools are covered by the Mesh licensing statements below. Such application code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

11.5 *The MakeHuman Mesh*

The MakeHuman© application uses a 3D mesh to represent human figures. This mesh was developed in a number of successive iterations (see [The Humanoid Mesh](#)) over a number of years to support a diverse range of physiological variants and pose-based deformations. The application uses a mesh with a standard topology (how the vertices, edges and faces relate to one another), which is manipulated by applying standard morph targets (mesh deformations).

The specific data files used to define and deform the mesh, distributed with the MakeHuman application are covered by the terms of the [GNU General Public License 3.0](#) and you are granted permission to use and redistribute those files in original or modified form under those terms. These data files can also be manipulated using the Blender MakeTarget script, but they, and derivative works remain under the terms of the GNU GPL.

When you export, save, or in any way convert a MakeHuman data file, you may store any deformations that you have applied/created, along with the topology of the base mesh and any positional information that you have not materially changed. You are the Copyright holder of changes that you have made. The MakeHuman Team retains Copyright over the topology and the base mesh, but grants you permission to copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the generated materials over which they hold Copyright, including the topology and positional information constituting the base mesh provided that this does not prevent others from doing the same.

You do not acquire exclusive Copyright over the topology or positional information constituting the base mesh, so others can, in perpetuity also create works derived from the topology and positional information constituting the base mesh and the morph target data used to deform the base MakeHuman mesh. Note that you may be able to establish copyright over works that you produce using the MakeHuman application if your work is sufficiently unique or distinct. The definition of how unique and distinct a work needs to be to be copyrightable is jurisdiction dependant and can vary over time as legal systems evolve and adapt to new legal challenges.

You own any 2D images that you create using exported MakeHuman mesh data and any copyright that applicable law affords you over those materials. Note that the definition of how unique and distinct a work needs to be to be copyrightable is jurisdiction dependant and can vary over time as legal systems evolve and adapt to new legal challenges.

THE MESH IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE MESH OR THE USE OR OTHER DEALINGS IN THE MESH.

11.6 *Documentation*

In general the documentation (including the text on this Wiki) produced by the MakeHuman© Team is Copyright© 2001-2009 by the MakeHuman Team (makehuman.org) who grant you permission to use and redistribute in original or modified form. Note that certain names and terms are individually copyrighted or are trademarks, so separate restrictions on the use of those names or terms may exist. For example, this authorisation does not entitle you to produce documents claiming to be the MakeHuman Team or claiming to produce an application called MakeHuman.

The Documentation Archive on this Wiki may also contain documents or links to documents that were or were not produced by the MakeHuman Team and/or may contain their own separate Copyright© notices, which may differ from this generic documentation licensing statement. Any licensing statements within individual documents take precedence over this licensing statement.

Appendix 1 - POV-Ray Export

The Persistence of Vision Raytracer is a popular and freely available rendering application capable of generating stunning images and animations. You can see some of the images that it is able to generate and you can download the software from <http://www.povray.org>. The application is well supported on the newsgroups at <http://news.povray.org>. The application renders images from text files that describe the scene using a markup language called the Scene Description Language (SDL). It does not incorporate a modeller, but SDL components exported from modellers can be incorporated into POV-Ray scenes by editing the SDL with a text editor.

MakeHuman provides options to directly export and optionally automatically render a POV-Ray 'include' file containing a posed and morphed humanoid model. This option also exports a pigment map and generates a sample 'scene' file that contains examples illustrating how to use the model. You can incorporate MakeHuman figures into your own POV-Ray scene files using a few short pieces of SDL that you type or cut and paste into those text files.

Export/Render Options

The generation and rendering options can be controlled using settings stored in a short Python script. The file '*mh2povray_ini.py*' can be found in the '*mh_plugins*' directory and contains a small number of variables that you can edit. You can change these settings while MakeHuman is running (you don't need to close and restart MakeHuman).

Output File Path

The 'outputpath' variable holds the directory name and file name of the include file that you wish to generate. The default setting is:

```
outputpath    = "pov_output/makehuman.inc"
```

A sample scene file of the same name is written into the same directory, along with a few other files you may find useful.

Include File Format

You can generate the exported figure in two different formats using the '*format*' variable:

1. **"mesh2"** - A simple POV-Ray mesh2 object that can be incorporated directly into your scene. This is quick to render but is not particularly flexible (it doesn't provide you with many configuration options).
2. **"array"** - Arrays of coordinates and normals plus a set of macros to generate objects from those arrays, along with other useful settings and utilities. This is far more flexible, but takes longer to render.

The default setting is:

```
format        = "array"
```

Render/Export

The '*action*' variable is used to control whether MakeHuman will attempt to automatically launch POV-Ray to render the exported files:

1. **"export"** - Just export the files.
2. **"render"** - Export the current model then invoke POV-Ray to render a scene file.

The default setting is:

```
action    = "export"
```

To use the "*render*" option you need to make sure that POV-Ray is correctly installed and that the directory setting '*povray_path*' points to the POV-Ray executable you wish to use (see below).

Scene File to Render

When the '*action*' variable is set to "*render*" the MakeHuman application will attempt to launch POV-Ray and render a scene file. The '*renderscenefile*' variable is used to specify which scene file to render. If set to an empty string then MakeHuman will render the generated sample scene file (e.g. makehuman.pov). If set to anything else then MakeHuman will pass the string you specify into POV-Ray as the file to render.

For example, you could specify:

```
renderscenefile    = "myfile.pov"
```

to tell MakeHuman to render this file once it's performed the export of the figure. **Note:** It is completely down to you to make sure that the scene file specified includes the figure that you generate.

POV-Ray Executable

If you set the '*action*' variable to "*render*" you will need to make sure that POV-Ray is correctly installed and that the directory setting '*povray_path*' points to the POV-Ray executable you wish to use. For example:

```
povray_path    = "C:/Users/Chris/AppData/Roaming/POV-Ray/v3.7/bin/pvengine-sse2.exe"
```

tells MakeHuman to use the multi-processor binary from the POV-Ray 3.7 distribution. The '*mh2povray_ini.py*' file contains other examples, but the name and location of this executable varies between different POV-Ray versions and between operating systems. On Windows you can locate this path from a desktop, file system or quick start icon by right clicking the icon and selecting 'properties'. From here you can cut and paste the path into the '*mh2povray_ini.py*' file.

Further Details

Fuller documentation of the POV-Ray Export functionality, including examples is contained on the Wiki at http://makehuman.wiki.sourceforge.net/UG_POVRay_Export.