KAT karma authoring tool

Welcome to KAT!

In a nutshell, KAT, which stands for Karma Authoring Tool, is a tool for creating, editing, simulating and managing Karma assets. The term “asset” is so fundamental that it requires immediate explanation.

What is an asset?

A few definitions and examples:

- An asset is collection of low–level Karma objects, specifically geometry, parts and joints (constraints).
- An asset can be thought of as a “game object”. It is designed to be a standalone object that can be instanced multiple times in a game level.
- Examples of good assets include a ragdoll human, a crane or a car. An example of bad asset usage would be an asset that contains all the physics data for an entire game level.

Assets can be imported into KAT from modelling tools and exported either as KA files or as C code.

The file manager

When you work with KAT you will be using the File Manager to work with KA files. The File Manager is located on the File tab of the tab panel. It is your main entry point into KAT. From here you can:

- Manage workspaces
- Create and destroy assets
- Import and export assets in various formats

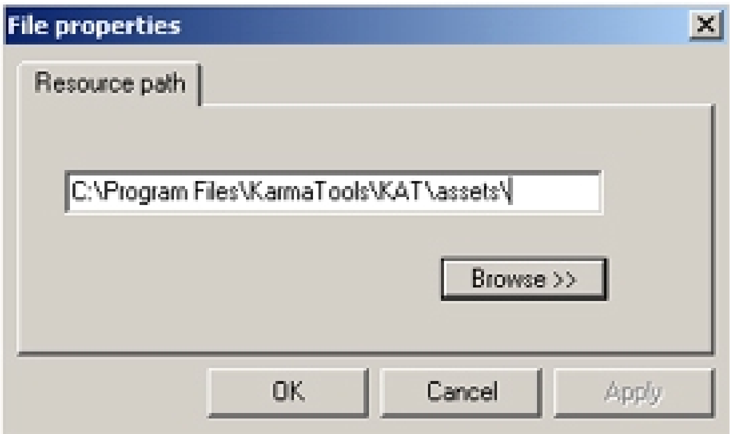
Workspaces

Workspaces are files with a KAW extension. KAT uses them to manage sets of KA files. Workspaces are very easy to use and are conceptually very similar to workspaces in Microsoft Visual C++ 6.0.

When you first start working with KAT you should create a workspace for your assets. This workspace should not be anywhere inside the KarmaTools directory – this is so that your data will remain safe when new releases of KAT are installed.

Now that you have a blank workspace you are ready to either insert existing KA files containing existing assets, or to create new KA files. To do either of these things right click on the newly created workspace root item. Almost all workspace functionality is available only on a context menu and you get different context menus depending on whether you click on the workspace root, a KA file or an asset. You will soon get used to the functionality provided.

Once you have a KA file in your workspace you are nearly ready to start working with assets. Before you do, however, you must ensure that you set the *resource path* correctly for the file. The resource path tells KAT where the artwork accompanying your assets is kept. Resource paths are maintained on a per KA file basis so bear this in mind when designing you directory tree. It is also worth noting that a KA file can contain multiple assets although they should be uniquely named to avoid confusion. To set the resource path, right click on a file in the workspace (not on an asset in the file) and choose Properties.



C:\Program Files\KarmaTools\KAT\assets\

It is up to you how you manage your assets and KA files. Workspaces are there to make this process easier. You can move assets between files using standard dragging functionality in the workspace tree. You can also rename files and assets.

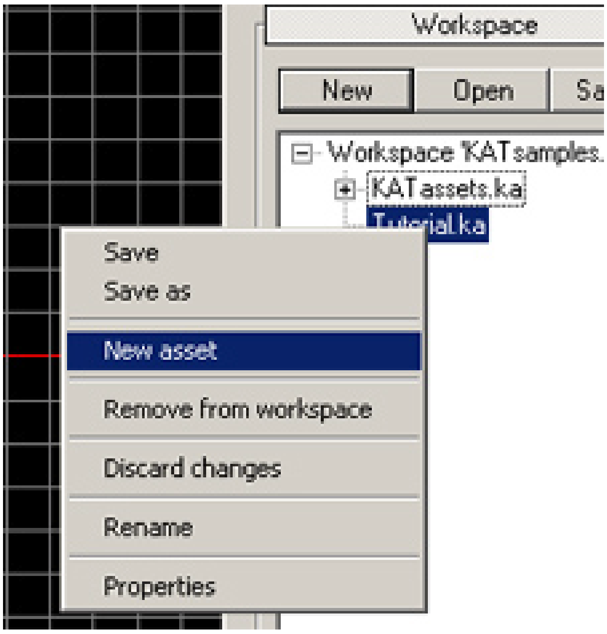
To create an empty asset you can right click on a KA file and select Create Asset. (Note that for skinned characters it would be much more usual to use an importer to create the initial asset, followed by hand editing).

Double clicking on the asset will set the asset active and bring you into edit mode. You will spend 95% of your time in edit mode. At any time you can tab back to the File Manager and select another asset for editing or perform any other workspace function on any file. KAT will maintain undo history on a per-asset basis so you are free to roam. A very useful feature is the discard changes feature which you can do from the workspace context menus at either workspace, file or asset level. You can also do it from the edit panel. Notice that an asterisk denotes unsaved data.

Creating a crane from scratch

This tutorial will take you through how to create a crane. Along the way it explains some useful theory too.

Step 1: Create an empty asset

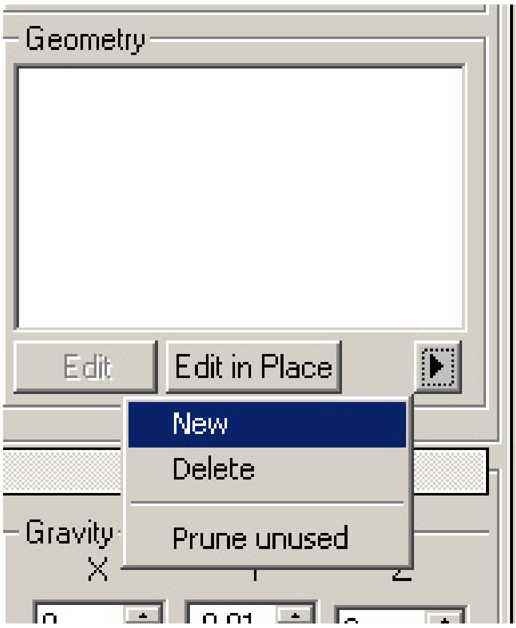


Create an empty asset from the File Manager. When you are prompted enter a name and choose a file. Leave the graphic blank as this is more appropriate for skinned characters where a single graphic is used for the entire asset. Here we are working with a non–skinned articulated system.

Remember to specify the file's resource path.

Double click the new asset in the workspace to make it active. You are now in edit mode.

Step 2: Create the crane base geometry



An asset consists of geometry, parts and joints. In the general case, a joint depends on two different parts and a part depends on a geometry. Geometry can be shared between multiple parts. This is the only referencing in KAT, everything else is by value.

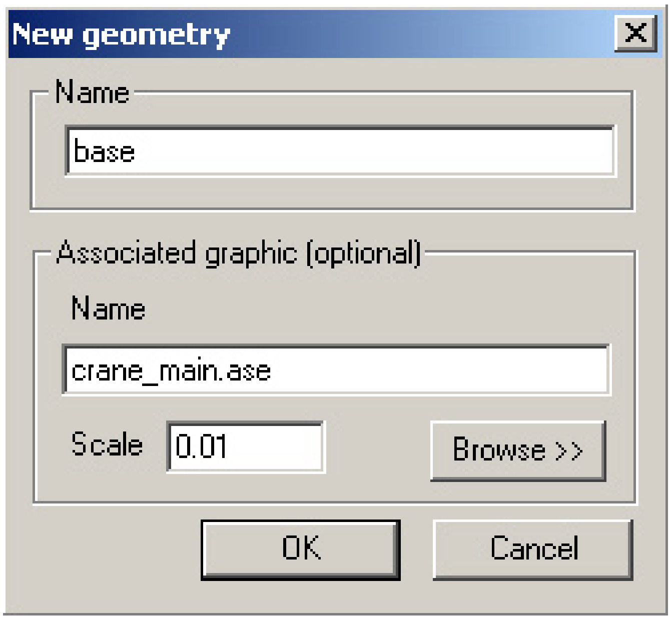
The upshot of this is that you must create a geometry before you can do anything else.

As you can see parts joints and geometry each have flyout menus where certain operations are performed.

Step 3: Create the crane base geometry (2)

You are now looking at the new geometry dialog. We will be creating the base of the crane first so specify the name as “base”. The crane graphics are stored as a series of .ASE files in the KAT/KATroot/assets directory. Choose the .ASE file entitled crane_main.ase and specify the scale as 0.01.

The scale factor maps between your graphics units and units that are sensible for physics simulation. This is normally determined by your game engine’s integration with Karma and is normally constant.


New geometry

Name: base

Associated graphic (optional)

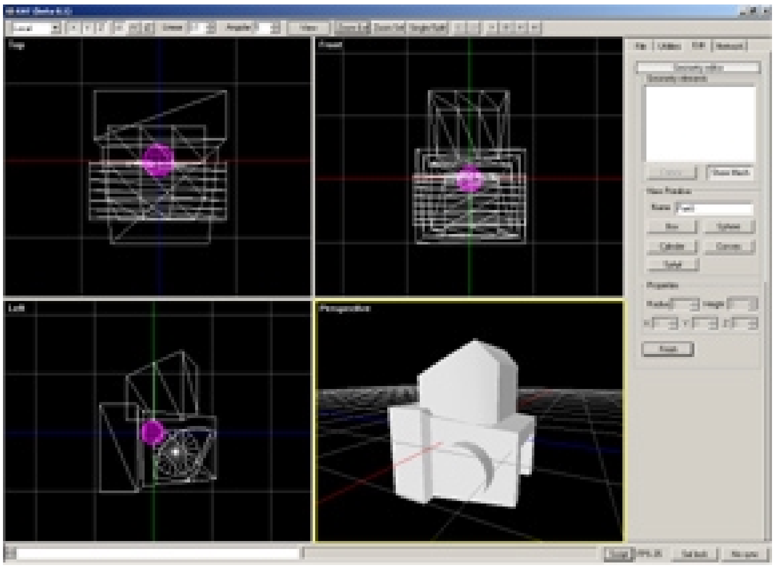
Name: crane_main.ase

Scale: 0.01

Step 4: Create the crane base geometry (3)

You are now in geometry editing mode. In this mode you can mark up a mesh with Karma collision primitives. As you can see the primitives available to you are sphere, box, cylinder, sphyl and convex.

You should now be looking at the mesh. The pink sphere is the geometry origin.



Click on the Zoom Ext button on the toolbar to zoom in for a better view.

Each geometry can be made up of multiple primitives and here we are going to use a box for the front and a convex mesh for the more detailed back end of the base.

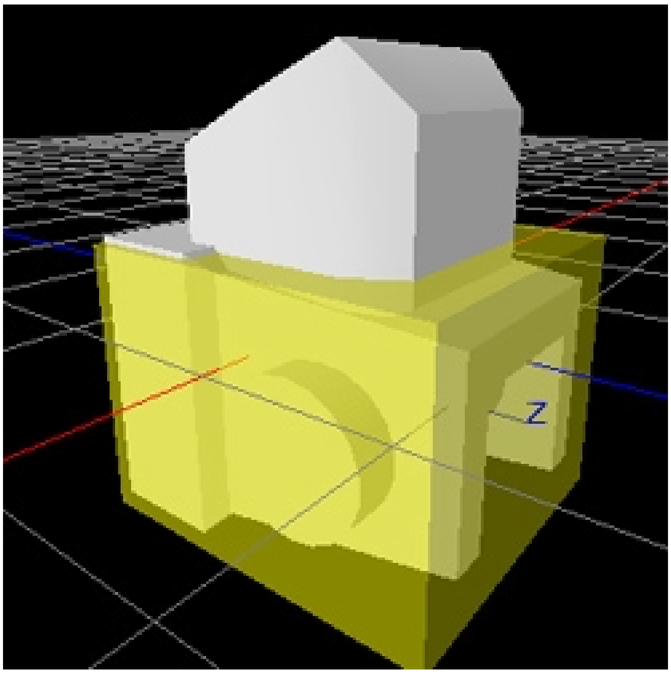
Step 5: Create the crane base geometry (4)

Click on “Box” to create a new box. The box appears in yellow and is unit sized. It is yellow because it is selected.

To change the size you must use the spinners. To change the position and rotation use the movement controls and snap settings on the toolbar.

This is a fairly arduous task and you will see in a minute how this can be greatly improved.

You should end up with something resembling the image on the right.



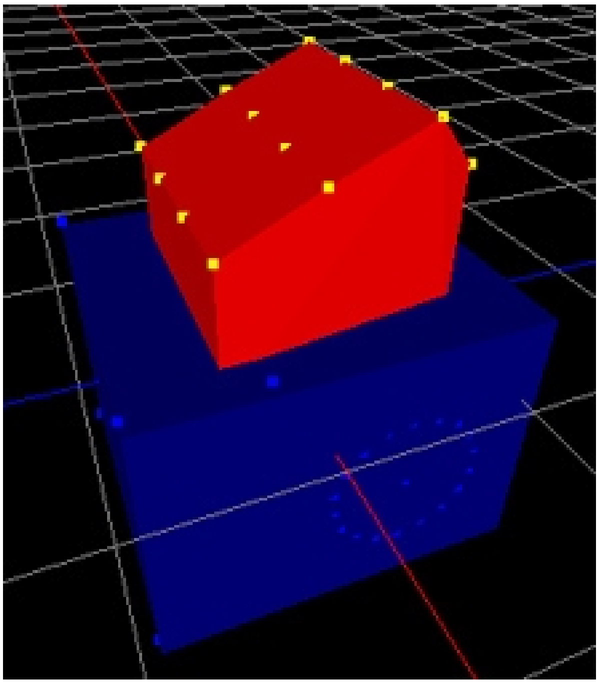
Step 6: Create the crane base geometry (5)

Click on “Convex” to enter convex hull mode. Your mesh will be displayed as a set of points, a subset of which you can select to create a convex hull collision geometry. Hold down CTRL to add to the set of points and SHIFT to remove.

When you have finished, click Done.

You can now toggle the mesh on and off with the Show Mesh button to see how well it matches.

When you have finished editing press Finish to return to the asset editor.



Step 7: Create the crane base part

Now that you have a geometry you can create one or more parts that use it. In this case we only want one part.

Selecting New from the parts flyout brings up a dialog. Set the name to “base”. Although this is the same as the geometry it doesn’t matter as parts, joints and geometry names are all in separate namespaces.

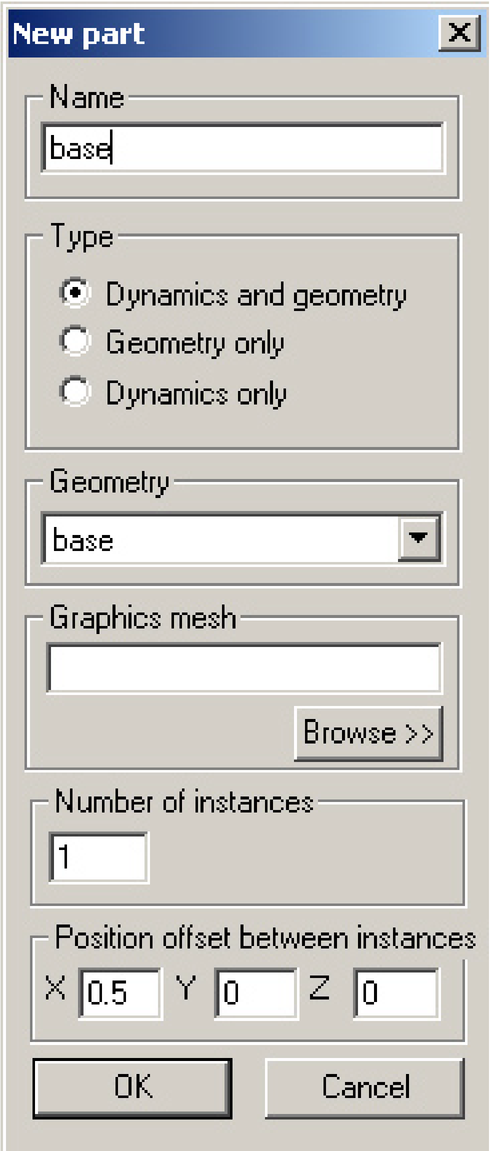
You now need to choose a type for the part. If you want the part to move it must have dynamics, if you want it to collide it must have geometry. Here we want both. As with all asset properties, you can easily change it at a later date.

Select the geometry you have just created from the geometry combo.

Because we want to use the graphic that is associated with the geometry you can leave the graphic mesh blank. Filling this in would override the existing graphic.

Leave the rest as the defaults and click OK.

Save your work using the Save button.



New part

Name: base

Type:
● Dynamics and geometry
○ Geometry only
○ Dynamics only

Geometry: base

Graphics mesh:

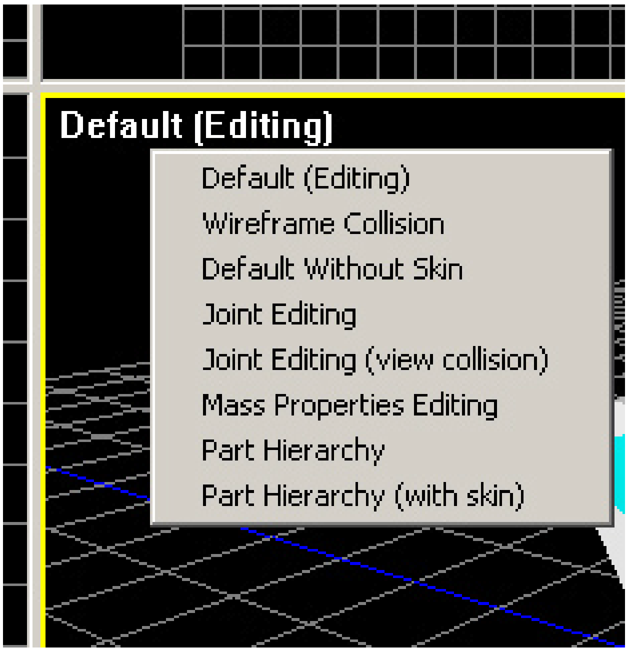
Number of instances: 1

Position offset between instances: X 0.5 Y 0 Z 0

Interlude: View modes

KAT’s main focus is not geometry editing but physics simulation. Physics simulation carries with it a lot of different types of visualizations. To help you manage these different visualizations you can use *view modes*.

View modes determine both what is seen and what is selectable. You should get used to changing view modes regularly (by right clicking at the top of a viewport). You can find good ways of working by having different viewports using different view modes.

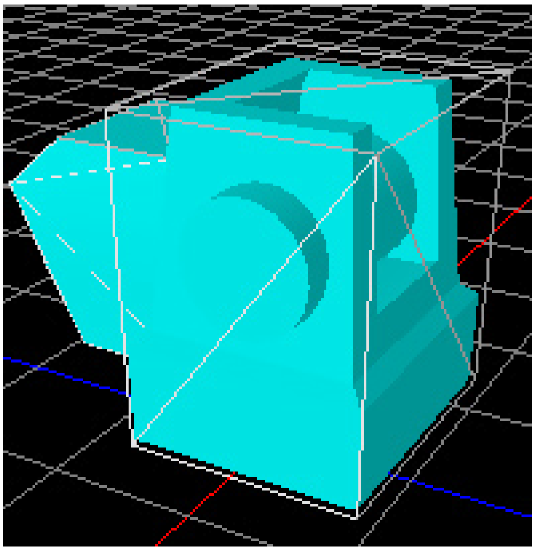


Step 8: Position the crane base part

The crane base’s default orientation is incorrect, so use the movement controls on the toolbar to position it correctly.

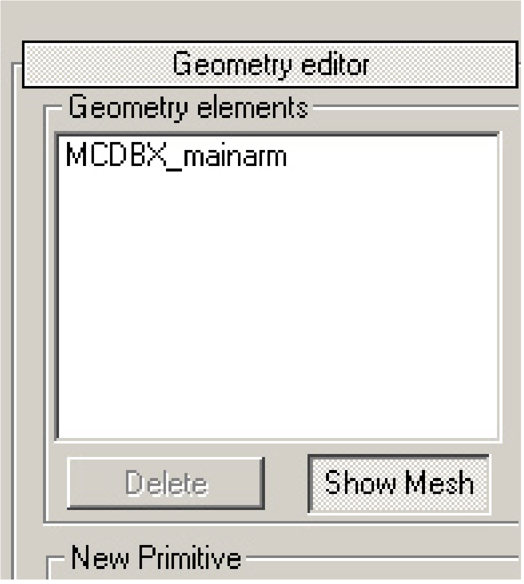
TIP: use + and – on the numeric keypad to alter the axes lengths. Pressing CTRL will do it in all viewports.

On the right is a shot of the perspective viewport in Wireframe Collision mode.



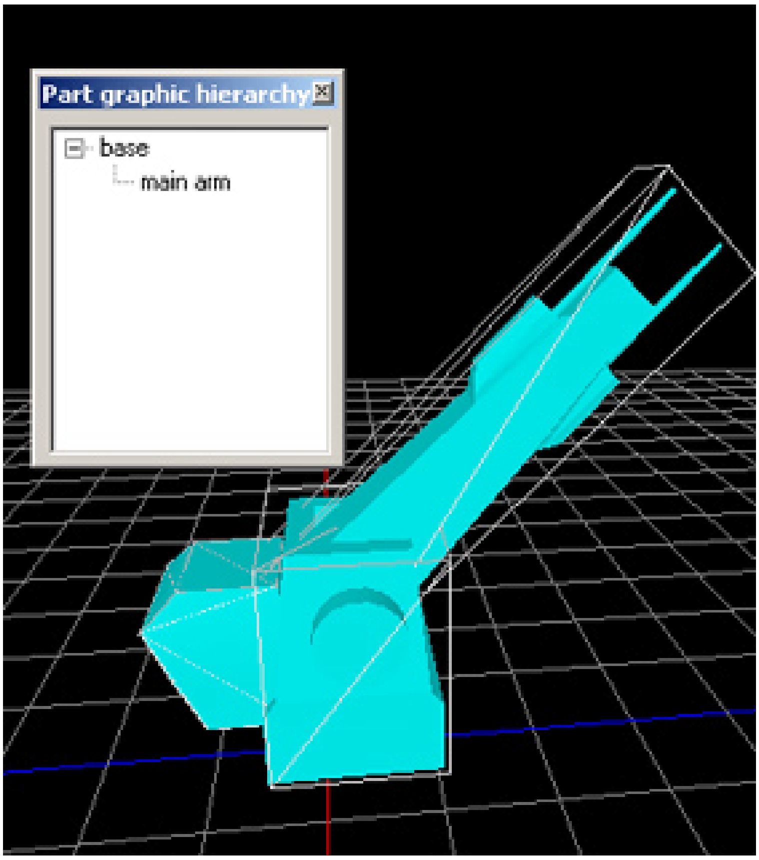
Step 9: Create the crane main arm geometry

Create a new geometry, this time using crane_main_arm.ase as the graphic and using the same scale.



This time you will notice that the geometry is already there. This is because it was previously marked up in 3DS Max, which if you are using Max, this is the recommended approach. You must be careful to adhere to the naming convention when naming collision geometry in Max. Use a MCDSP_ prefix for spheres, an MCDBX_ prefix for boxes, an MCDCY_ prefix for cylinders. You will have to use KAT for creating convex meshes.

Step 10: Create and position the main arm part



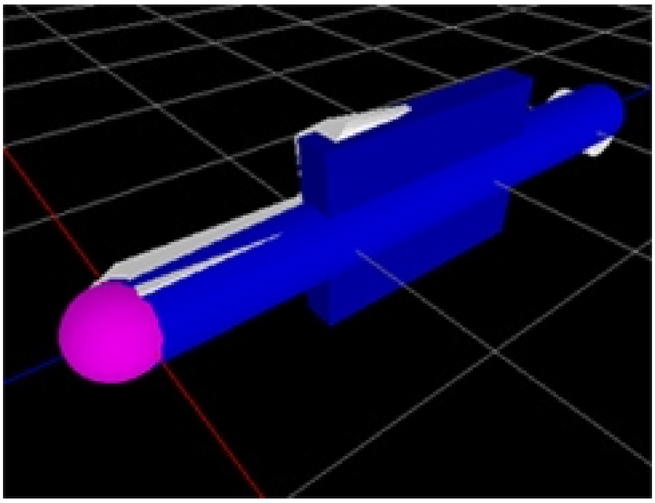
Using the movement controls make your crane look like the one on the right.

TIP: If you make a mistake use undo.

Once the crane is in place it is a good idea to make the main arm a child of the base in the graphics hierarchy. This is an aid for manipulating and positioning multiple parts and is unrelated to joints in the simulation.

To bring up the hierarchy dialog click on Hierarchy in the parts section. The hierarchy can be edited using drag and drop within the dialog. To make an item a root item drag it off the dialog.

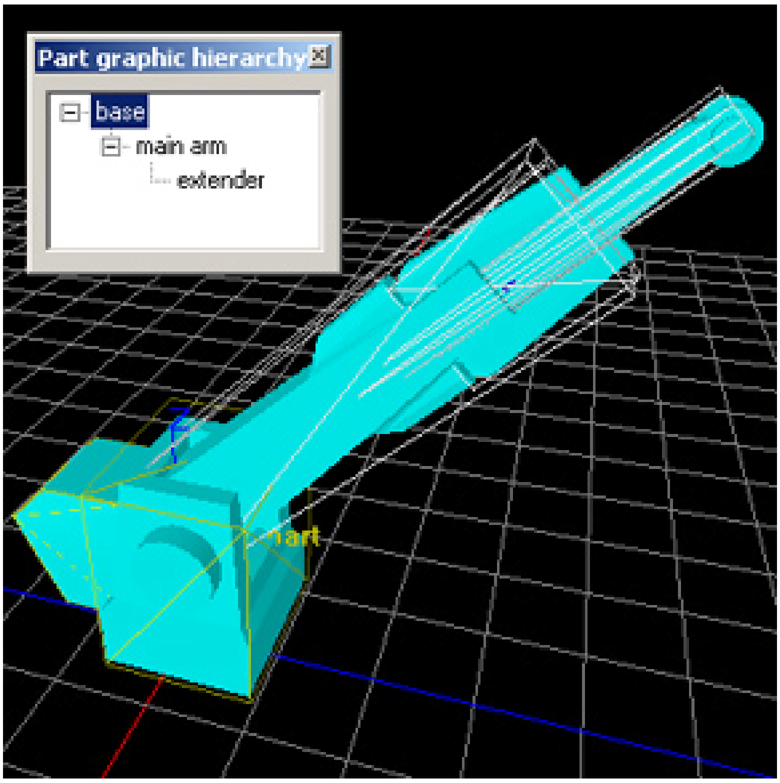
Step 11: Create the extending arm's geometry



Up until this point the geometry origins have been correct by default. However, when you create a geometry for the extender you must position the origin as shown. This is to ensure that its pivot point is correct. The joint attachment position is also at the pivot point.

TIP: To move the origin ALT left drag (but not on it). To toggle the origin ALT right click.

Step 12: Create the extender's part and arrange crane



Use what you have learned so far to add the crane's extending arm (crane_extender.ase).

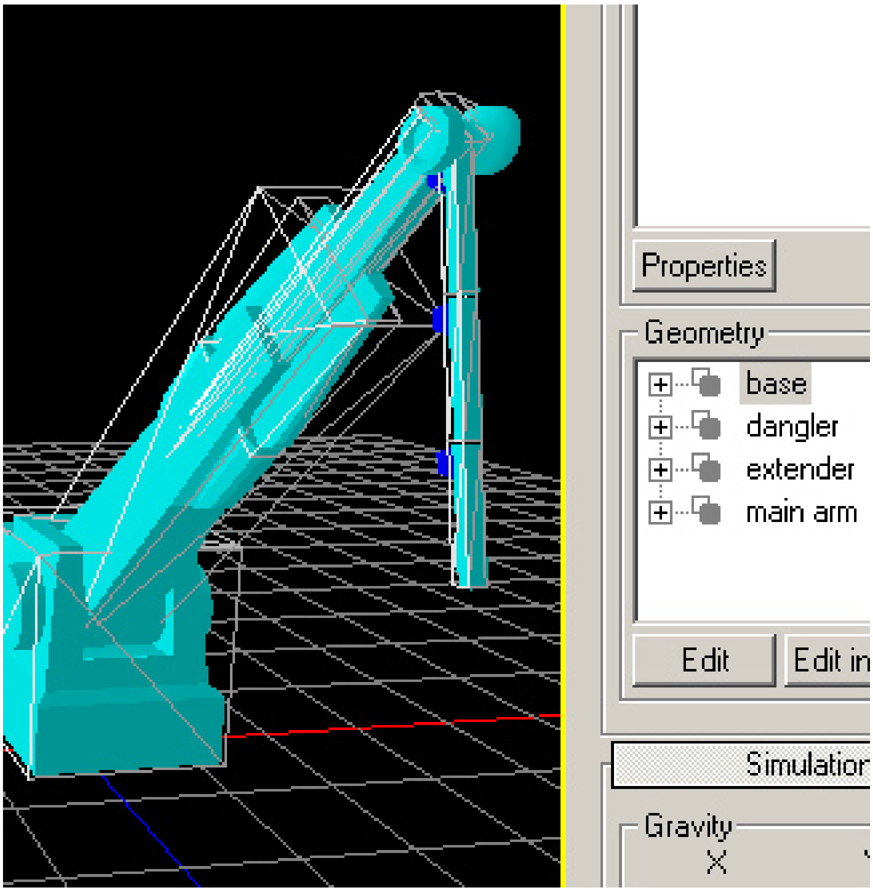
After you have done this the crane will look like the picture on the right.

TIP: If you are feeling adventurous you can cut and paste parts from the completed example.

Step 13: Add the three danglers to create a chain

Remember that geometry can be shared so you only need to create one dangler geometry (crane_dangler.ase).

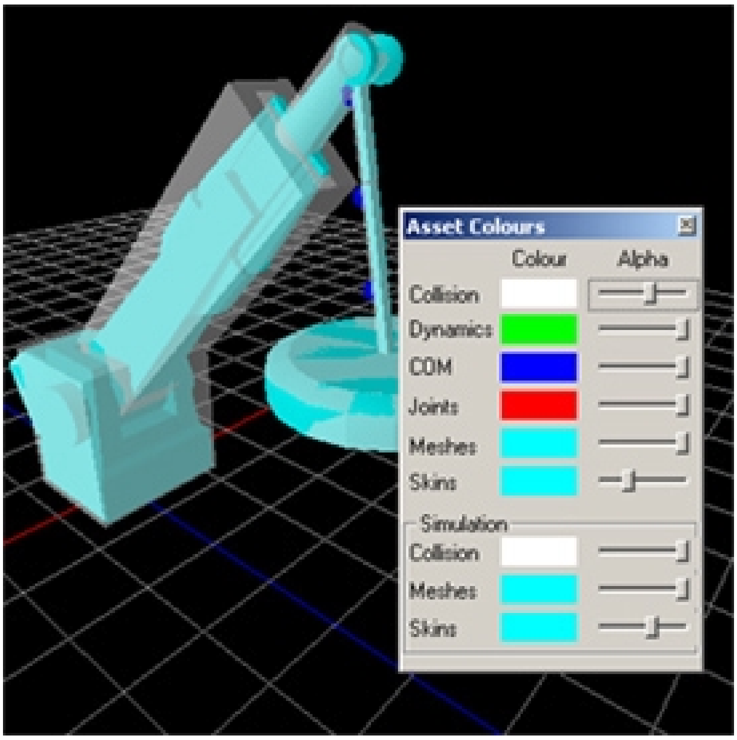
TIP: When a graphic will result in a fairly thin collision geometry, it is a good idea to err on the side of a thicker collision geometry than the graphic. This will result in better collision response for thin objects.



Step 14: Add the magnet to complete the crane

TIP: prefer convex mesh to cylinders as they simulate faster and in most cases give a more accurate shape.

The crane on the right is shown in default view mode with the asset colour properties dialog. This dialog is very useful for familiarizing yourself with the colour codes used in KAT (eg by default cyan is a mesh, white is collision etc).

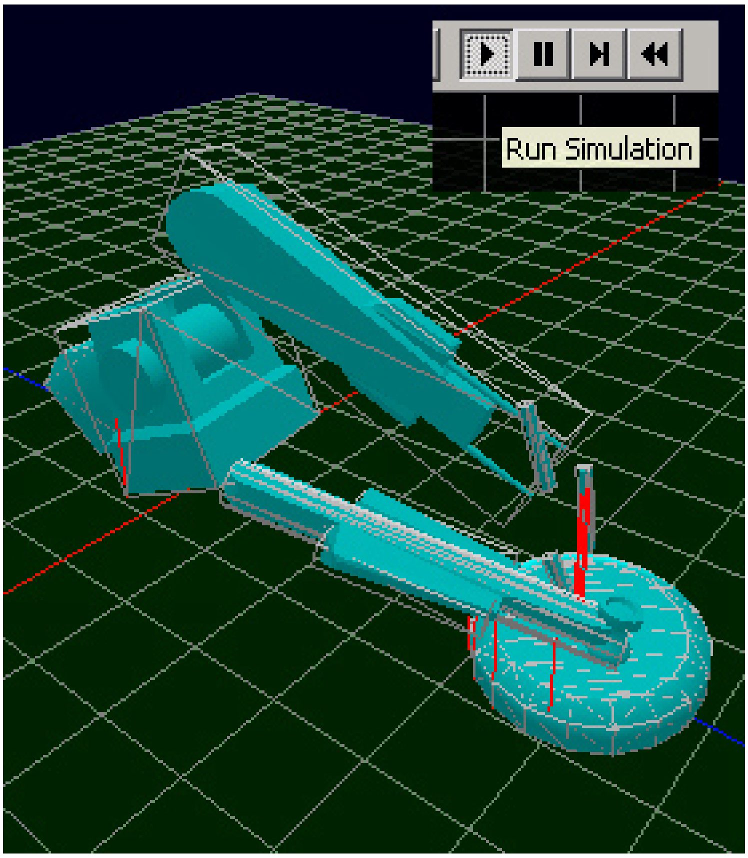


Step 15: Simulate!

Visually your crane looks perfect, but when you simulate it you will realize, if you haven't already, that there is a lot more to simulation than looks alone ☺

TIP: Drag pieces around to see how they interact. The red lines are contact points that Karma has calculated.

Experiment with the parameters in the simulation options rollup and try changing the RPRO strength in the mouse dialog found in the View menu, or by pressing F5.



The Relative Position Relative Orientation constraint is used to move objects when you drag them with the mouse. For more information consult the Karma Userguide.

You may want to see the simulation in more than one viewport at once. To do this CTRL right click to bring up Viewport options and select "Use for simulation". Depending on your machine you may find that your framerate appears to make the simulation run too slowly. Either run with a single viewport or enable the "No sync" option which is on the right hand side of the bottom toolbar. When this is enabled clicking on a viewport will instruct KAT only to render that view.

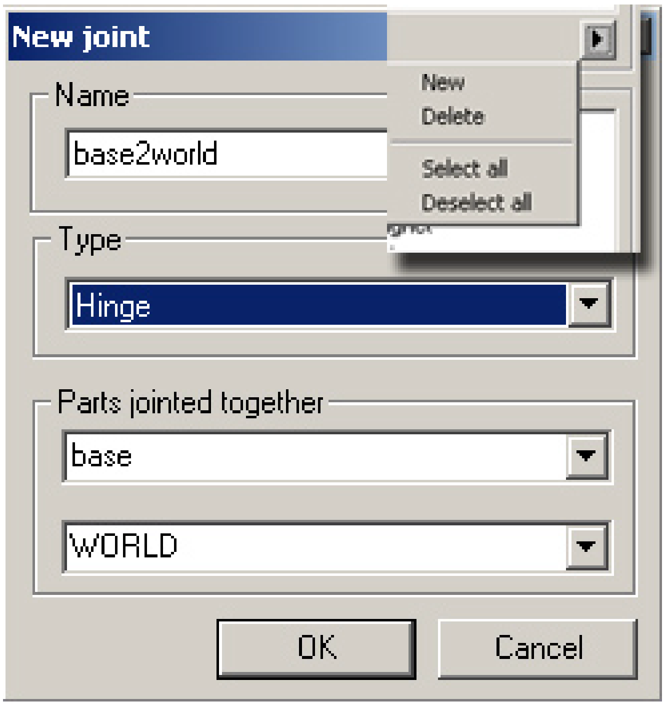
TIP: You can drag the splitter bars to customize your view or enter single viewport mode using the button on the toolbar or F4.

Step 16: Adding the base joint

The first joint we need is between the base and the world, so choose New Joint from the joint flyout. Type in a descriptive name and select the joint type “Hinge” because we want to be constrained to one axis of rotation. Set up the parts as shown. Notice that the second part is WORLD.

TIP: Selecting one part before creating a new joint will automatically fill in part1. Selecting two parts will fill in both parts.

TIP: For detailed joint documentation refer to the Karma user guide.


New joint

Name: base2world

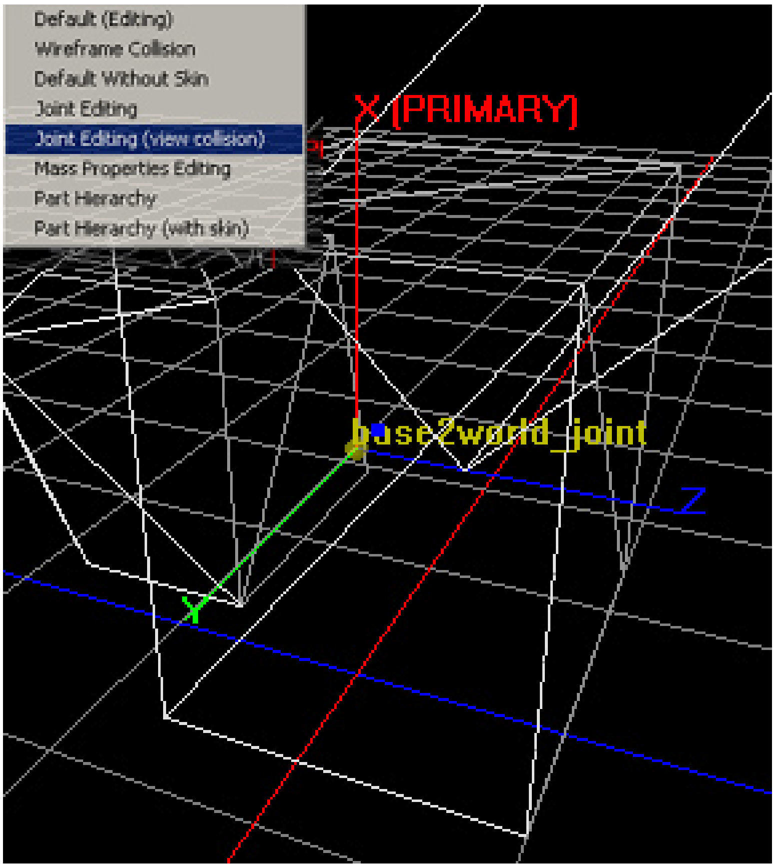
Type: Hinge

Parts jointed together: base, WORLD

Step 17: Orienting the base joint

If you simulate it now you will see that the base rotates around the wrong axis. To correct this you need to enter a joint editing mode. Select “Joint editing (view collision)”. This mode strips away extraneous visual information and lets you concentrate on the task at hand. Note that in this mode you cannot select parts.

Select the joint by either drag–select or by name in the joint list box and rotate 90 degrees around Z so that the primary hinge axis points upwards.

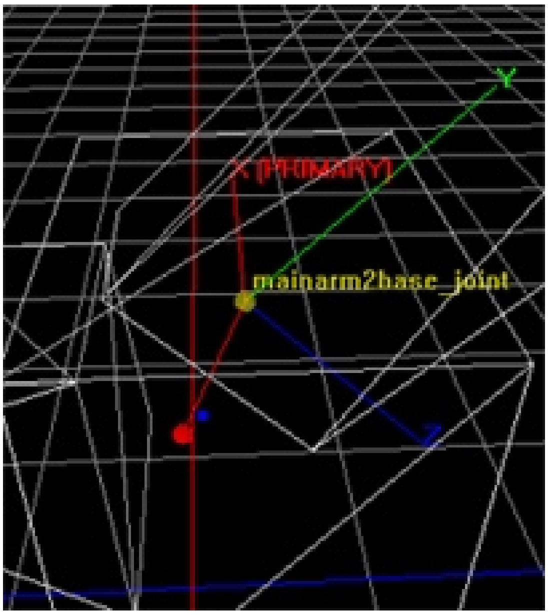


TIP: Use 90 degree rotational snap and constrain rotation to Z using movement controls on toolbar.

Try simulating it to see how it behaves.

TIP: In simulation the middle button can be used to rotate objects (in world space only). Try spinning the base around the Y axis.

Step 18: Add the main arm joint



Repeat the above procedure. Ensure that part1 of the joint is the main arm and that part2 is the base. This is because joints are set up in Karma child to parent.

This joint is also a hinge, but constraining a different axis.

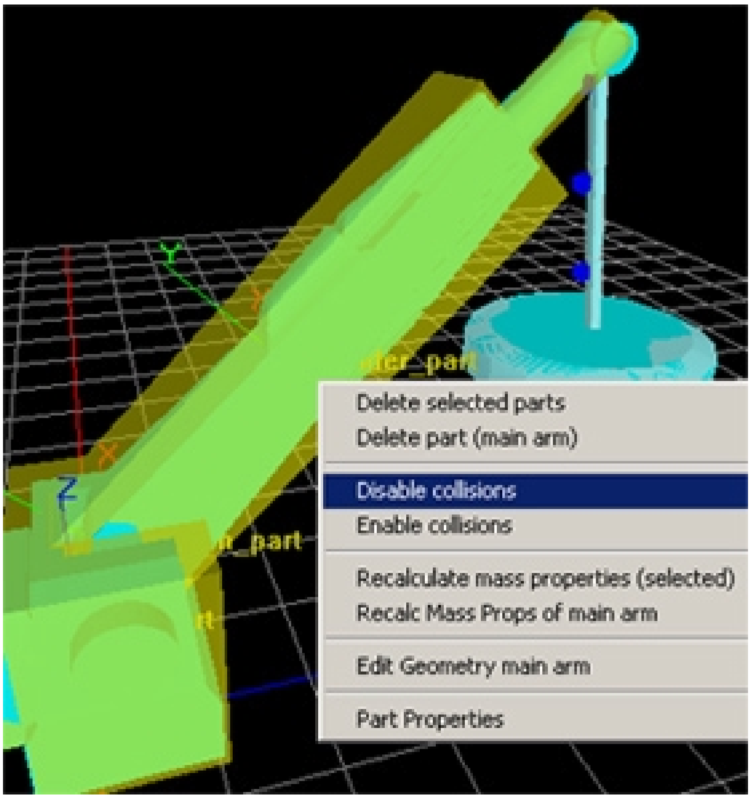
You will notice a red line appears to join the two joints. In fact if you experiment with moving the joints (you can always undo) you will see that red lines indicate which parts a joint connects. Lines join the part origins (pivot points) to the joint position.

Step 19: Add the extender joint

This time a prismatic joint is required which is like a telescope with an additional rotation constraint.

Once you have oriented the primary axis to lie along the extender, simulate it.

As usual the crane flies apart, but don't worry we will soon fix this. As you have probably guessed the reason it flies apart is because collisions are enabled between parts that start initially overlapped and touch during simulation.

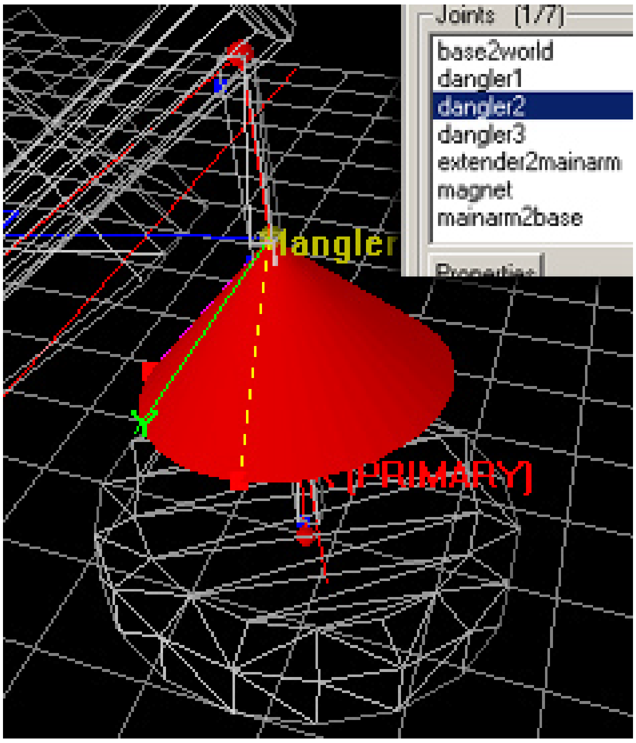


To rectify this, switch to the default editing view mode and select the base, the main arm and the extender. Right click and select "Disable collisions". Note that this disables all selected parts against each other so use with care.

Step 20: Add the dangler and magnet joints

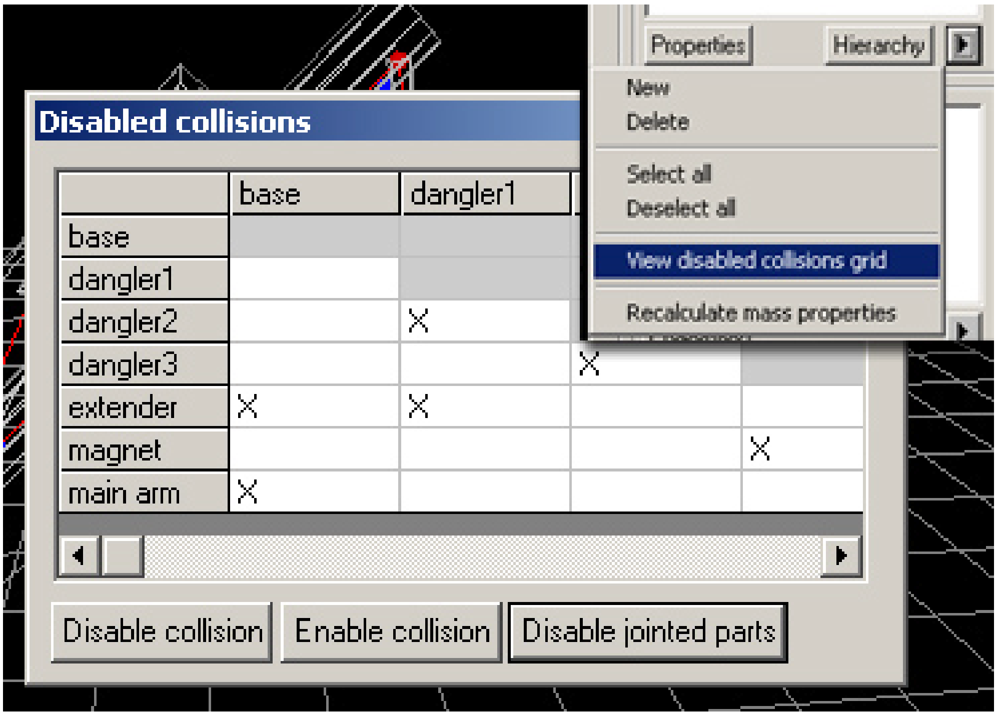
Create the remaining four joints, but this time make them of type “Skeletal”. Skeletal joints are so named because they constrain the child part to move within an elliptical cone. A twist constraint can also be optionally added. This is similar to a shoulder joint, but it works perfectly here too.

You may like to orient the cones as shown on the right because this is the most intuitive orientation. However, the constraint is set up such that the X axis wherever it is positioned, represents the original joint orientation and can move anywhere inside the cone. Thus with the skeletal joint, it’s own orientation is irrelevant.



Step 21: Finish disabling collisions

As well as the right click menu you can view all disabled collisions on a grid. From the parts flyout choose “View disabled collisions grid”.



Disabled collisions

	base	dangler1		
base				
dangler1				
dangler2		X		
dangler3			X	
extender	X	X		
magnet				X
main arm	X			

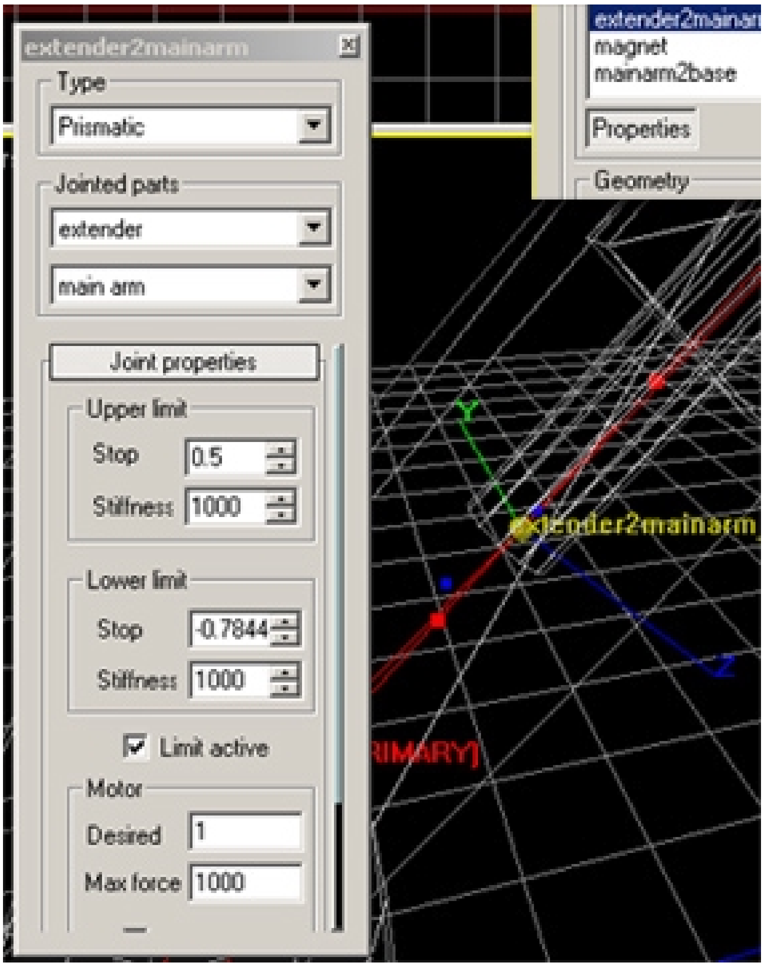
You can use the “Disabled jointed” button to quickly ensure that no two jointed parts can collide. This sorts out the collisions between the danglers and the magnet.

Step 22: Limiting the extender

Click “Properties” in the joint section and then select the extender joint. A good mode for this is “Joint editing (no collision)”.

TIP: Press space to lock the graphics selection (but remember to unlock again).

In the joint properties dialog check the “Limit active” checkbox. You will now see two red joint limit handles which can be dragged. Alternatively you can type these values in.



extender2mainarm

Type: Prismatic

Jointed parts: extender / main arm

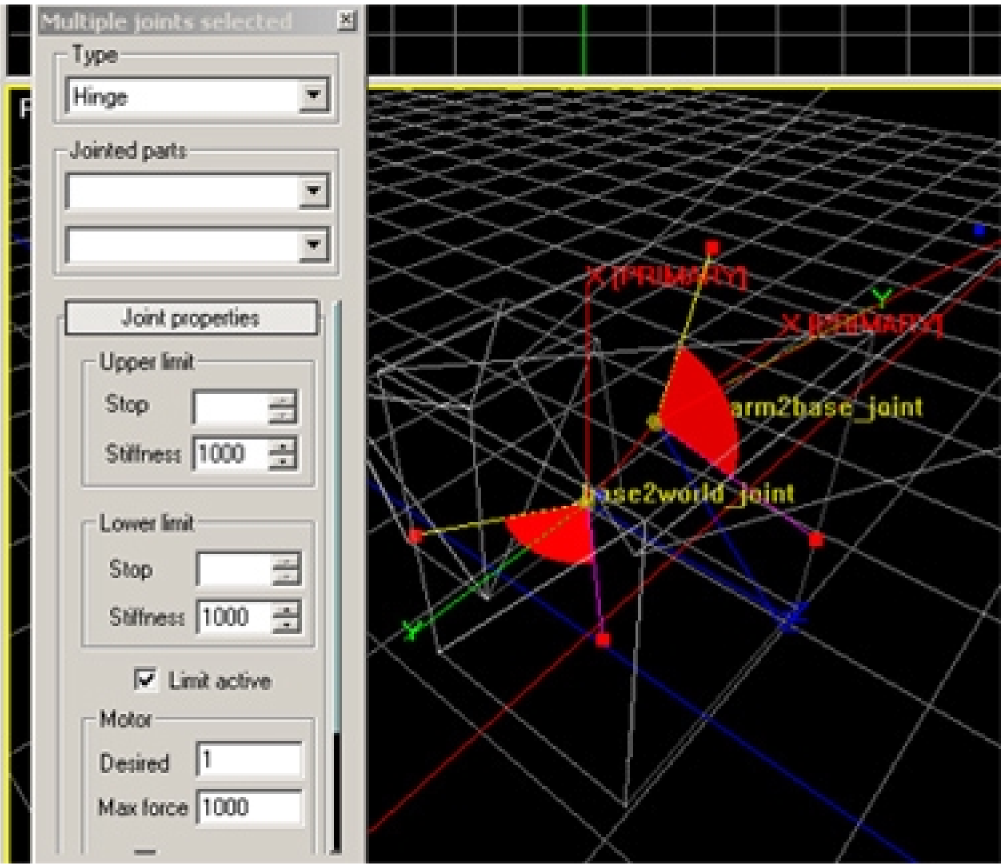
Upper limit
Stop: 0.5
Stiffness: 1000

Lower limit
Stop: -0.7844
Stiffness: 1000

☑ Limit active

Motor
Desired: 1
Max force: 1000

Step 23: Add limits to the base and main arm joints



Type: Hinge

Upper limit: Stop: ; Stiffness: 1000

Lower limit: Stop: ; Stiffness: 1000

☑ Limit active

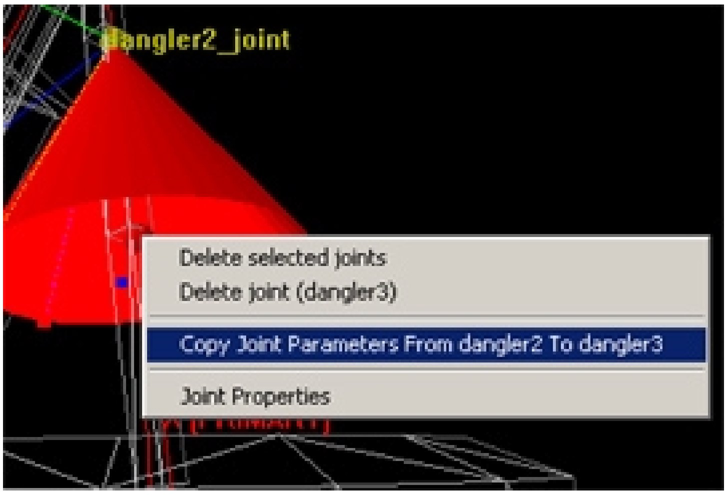
Motor: Desired: 1; Max force: 1000

Do the same for the base and main arm hinges. The same sort of situation to skeletal joints applies here too. In the case of hinges, rotation about the primary axis is irrelevant.

The initial position is represented by the Y axis which can be thought to move in the red arc. Notice that you can select more than one joint at a time – the dialog only shows common properties.

Step 24: Adjust skeletal joints

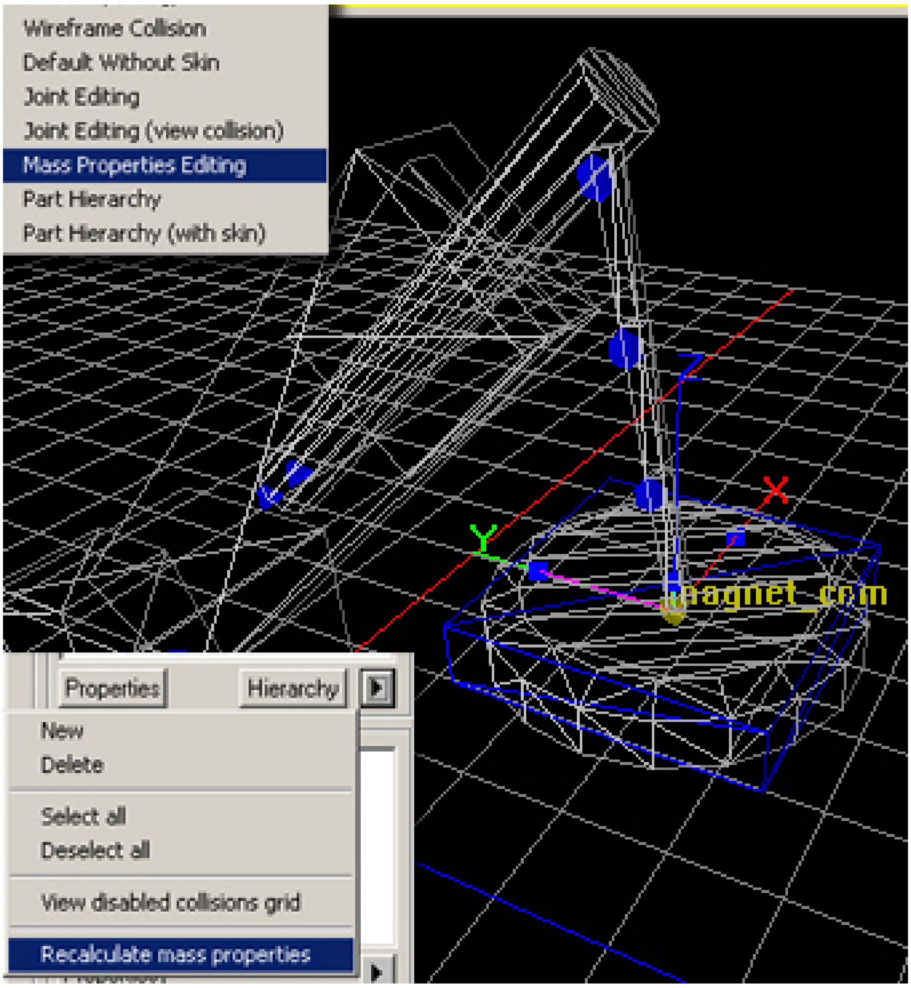
Select the joint between the extender and the first dangler link and selection lock it. Now you can drag the cone limit handles appropriately. Also, experiment with skeletal joint properties such as limiting the twist.



Assuming you want the same cone on the remaining three joints you can simply set up the first and copy the parameters. To do this select the source joint and right click on the destination joint. Choose “Copy Joint Parameters” from the context menu.

Step 25: Recalculate mass properties

Select the "Mass properties editing" view. This view shows centres of mass as blue shapes which when selected display a box representing the equivalent inertia tensor. Note that this visualization does not take into account off diagonal inertia components at the moment.

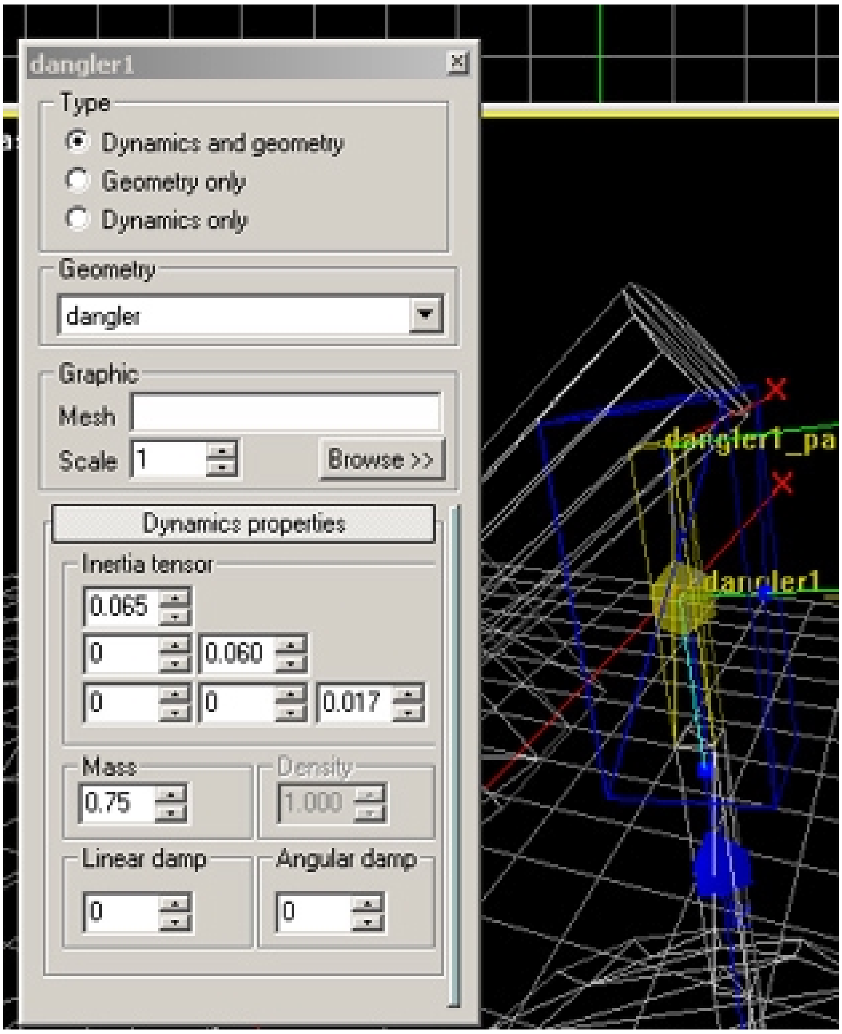


Use the part flyout to select all parts (this is not possible in this view mode), and then use it again to recalculate mass properties. Alternatively you can right click on each centre of mass to recalculate them on an individual basis.

Step 26: Change the mass and the inertia of the dangler parts for a better simulation

TIP: Don’t forget to save your work.

Simulate the asset as it now stands. You may notice a warning appear in the bottom warning panel indicating that the inertia of a physics body is below the lowest limit that Karma can reliably handle. This is coming from the danglers. To rectify this situation select the first dangler and bring up the part properties dialog. Increase the mass to 0.75.



dangler1

Type
- ● Dynamics and geometry
- ○ Geometry only
- ○ Dynamics only

Geometry: dangler

Graphic
Mesh:
Scale: 1

Dynamics properties

Inertia tensor
0.065
0 0.060
0 0 0.017

Mass: 0.75
Density: 1.000
Linear damp: 0
Angular damp: 0

If you now select the centre of mass of this part you will see that the inertia box has shrunk (or perhaps disappeared). You need to expand this manually either by entering the correct values in the dialog or by dragging the inertia box handles.

TIP: If the handles are inside the centre of mass shape you can still drag them using CTRL middle drag ☺

It is best to exaggerate the inertia as shown in the image. Note that this step may become unnecessary with later versions of Karma and KAT, but you will still need to do this for generating particular effects.

TIP: If you select one centre of mass and then right click on another you can copy the properties in a similar way to joints.

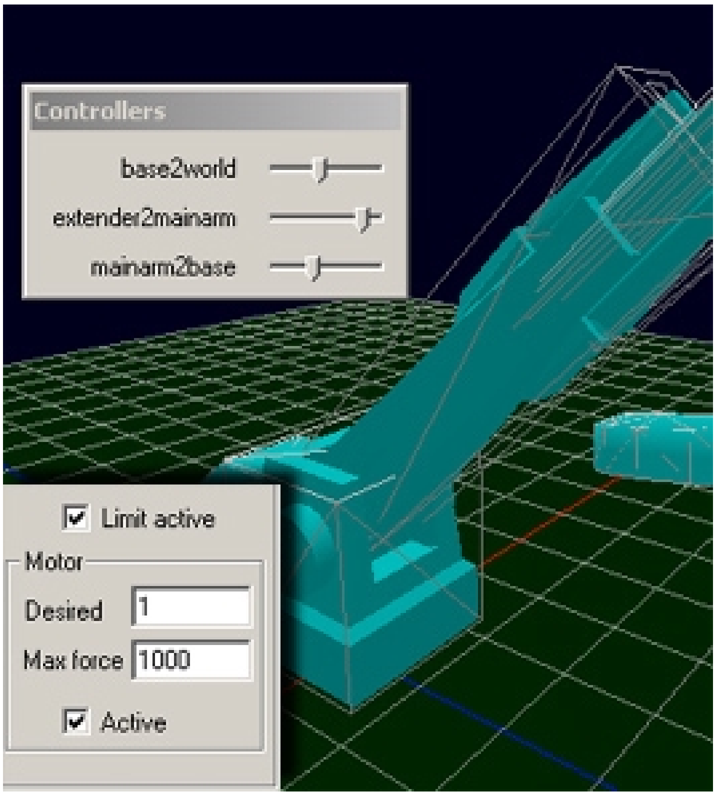
Step 27: Controllers

At the moment if you simulate the crane it just collapses under gravity.

TIP: You can change gravity in the simulation options rollup.

To make the crane behave like a real crane it needs some form of control.

Select the base and main arm joints and bring up the joint properties dialog. Check the “Motor active” checkbox. Repeat for the prismatic joint (the three joints are not of the same type so multiple select does not recognize commonality).



☑ Limit active

Desired: 1

Max force: 1000

☑ Active

When you run the simulation you now get a controller dialog to play with. Note that controllers only work with hinges and prismatics at the moment.

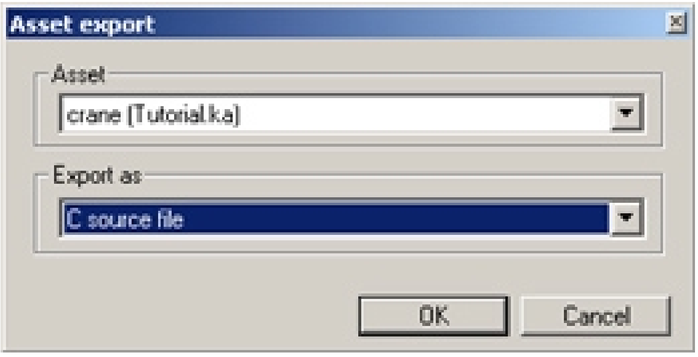
TIP: Bring up a another non-simulating viewport and tweak joint limits at the same time.

TIP: Try playing with the “Desired” and “Max force” properties.

You have now finished creating the crane asset!

Step 28: Export

Now that you’ve finished your asset you can export it either as a KA file or C code which can both be used with the Karma libraries. Just hit export, or use the file manager context menu.



Asset: crane (Tutorial.ka)

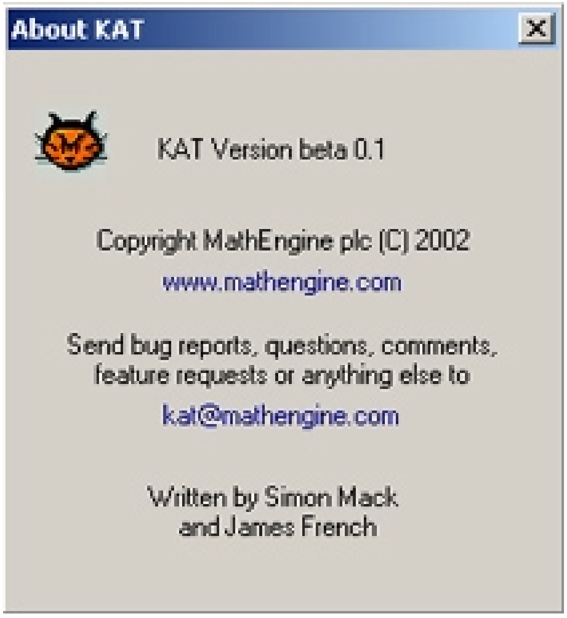
Export as: C source file

For your convenience there is simple MathEngine app wizard that works with Microsoft Visual C++ 6.0 that is designed to accept C code assets generated by KAT. Just run the wizard and copy over the asset.c file.

Step 29: Feedback

Please provide us with feedback both on this tutorial and on KAT so we can improve it for you ☺

Please send feedback to kat@mathengine.com and check our website at MathEngine for updates.



KAT Version beta 0.1

Copyright MathEngine plc (C) 2002
www.mathengine.com

Send bug reports, questions, comments, feature requests or anything else to
kat@mathengine.com

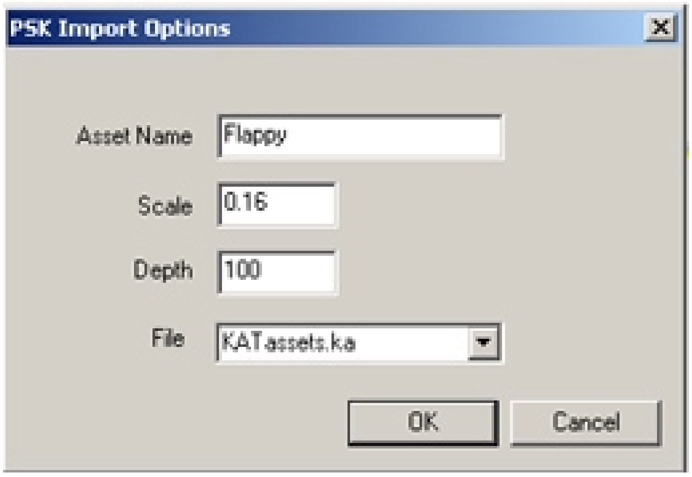
Written by Simon Mack
and James French

KAT Unreal Plugin

The Unreal plugin is designed to support the Epic/Karma Unreal integration. When the plugin is loaded, assets that specify a PSK file as their graphic will be skinned using that file. Any asset parts that have the same name as a bone in the PSK skeleton will be used to determine that bone’s position. The reference pose will be used for all other bones.

The plugin also adds a PSK importer, which can be used to automatically create an asset from a PSK file. This will generate sphyl and sphere collision models for the bones and ball-and-socket joints between them.

The plugin inserts an Unreal rollup into the FileManager tab, on which the “Import PSK” button can be found. After selecting the PSK file, the PSK Import Options dialog will be shown.

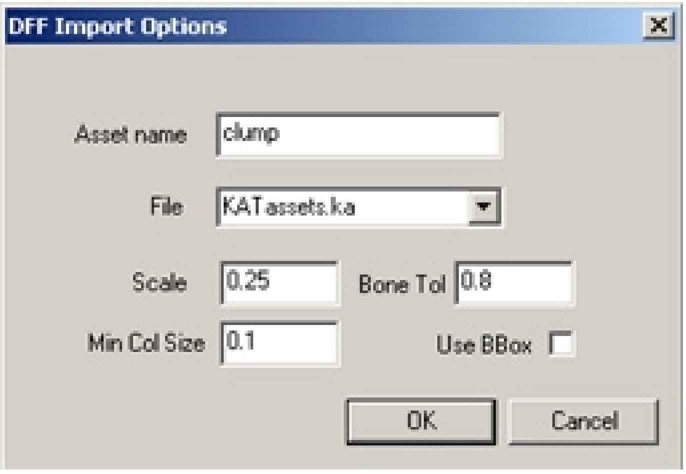


Asset Name: Flappy
Scale: 0.16
Depth: 100
File: KATassets.ka

The name of the new asset, and the file into which it is to be inserted must be specified. The scale parameter should reflect the scaling used between Unreal units and Karma units. Dimensions in the PSK will be multiplied by this value to create asset units. The depth value is an integer that determines the maximum depth from the root that a bone may be to be converted into an asset part by the importer.
Clicking OK will create the asset, and a confirmation dialog will be displayed indicating the number of parts in the new asset. This asset can then be found in the FileManager and edited like any other.

KAT RenderWare Plugin

KAT’s RenderWare plugin supports the display of DFF skins, the automatic creation of assets from a hierarchy in a DFF file, and the export of C++ code for use with the RenderWare Studio Karma integration.
In order for assets to be skinned with a DFF, they should have their asset graphic set to the DFF file, which must be in the KA file’s resource path. Additionally, there must be a corresponding ANM file, from which the reference pose of the skeleton is taken. Asset parts named “boneXXXX” where XXXX is the bone’s NodeID in the RpHAnimHierarchy loaded from the DFF will be treated as bones in the skin. All other bones will be fixed relative to their parent using the reference pose from the first frame in the ANM file.
To automatically create an asset from a skeleton specified by a DFF and ANM, select “Import DFF” from the RenderWare rollup which appears on the FileManager tab when the plugin is loaded. After selecting the DFF file, which must be in the same directory as a ANM file with the same name, the DFF Import Options dialog is displayed.



Asset name: clump
File: KATassets.ka
Scale: 0.25
Bone Tol: 0.8
Min Col Size: 0.1
Use BBox ☐

This dialog allows the user to specify the new asset’s name, and the file into which it should be inserted. The dimensions in the DFF are multiplied by the “Scale” value to produce the asset, and so should use the same scale factor as the target integration. The “Min Col Size” parameter specifies the minimum size (in asset units) for a part to be given collision geometry. Parts that are smaller than this will be created with dynamics only. By default, each bone is converted into an asset part, and given either sphyl or sphere collision geometry. The dimensions of the sphyl are in part calculated from the bounding box of the vertices that are influenced by the bone. The “Bone Tol” parameter specifies the minimum influence a bone must have on a vertex for that vertex to affect the bone’s bounding box. This must be a value between zero and one. The “Use BBox” checkbox allows the creation of an asset with box collision instead of sphyl and sphere geometries. This is intended as a debugging tool. After setting the options, clicking “OK” will build the asset and display a confirmation message indicating the number of parts created, which should match the number of bones in the skeleton. By default, bones are joined using hinges without limits.

The RenderWare plugin also adds an export option to KAT. When exporting an asset, either from FileManager or the Asset Editor, select “Renderware Studio C++ source file” from the “Export As” combo. After specifying the destination file, an options dialog will appear, allowing the class name and Karma toolkit version to be specified. The C++ code produced can be used with the RenderWare Studio Karma integration.

Appendix A – Mouse Input & Keyboard Shortcuts

KAT's Standard Mouse Handler

IMPORTANT: The middle mouse button plays an important role in KAT. It would be best to ensure that you have no drivers loaded that remap this button.

NB: Where Alt is specified, both Ctrl & Shift together will have the same effect

1> CAMERA Controls:
(NB: These are also implemented by the CameraMouseHandler)

In a perspective or ortho view:
Right-drag: Rotate view (around look-at point)
Shift-Right-Drag: Pan view (move look-at point)

In all other (fixed angle) views:
Right-drag: Pan view (move look-at point)

In all views:
Control-Right-Drag: Zoom view (move camera away from/towards look-at point)
Double-Left-Click on Object: Set look-at view to be origin of object
Control-Double-Left-Click on Object: As above, but for all views

MouseWheel: Zoom view
Ctrl-MouseWheel: Zoom view (large increments)
Shift-MouseWheel: Zoom view (small increments)

2> ORIGIN Controls:

The origin is displayed as a magenta sphere.
Alt-Right-Click: Toggle display of the origin
Alt-Left-Drag: Move the origin

3> SELECTION Controls:

Left-Click not on Object: Deselect all
Left-Click on Object: Select object (deselects all other objects)
Control-Left-Click on Object: Toggle object's selection state
Left-Drag not starting on selected Object: Area selection
Control-Left-Drag: Area add to selection
Shift-Left-Drag: Area remove from selection

4> MOVEMENT Controls:

Toolbar options set the constrained axes and reference frame for movement.

Left-Drag starting on selected Object: Move (translate) selected objects
Middle-Drag: Move (translate) selected objects
Alt-Left-Drag starting on selected Object: Rotate selected objects

Alt-Middle-Drag: Rotate selected objects
MouseWheel with Alt-Left Button down: Cycle rotation axes

Limit Handles: (These are not selected before movement)
Left-Drag on Limit Handle: Drag the handle
Ctrl-Middle-Drag on Handle: Drags a handle, even if it is behind/inside an object
Ctrl-Middle-Drag NOT on Limit Handle: Same as Middle-Drag

5> VIEWPORT Controls:

Right-Click: Display context menu
Ctrl-Right-Click: Display viewport options menu
Right-Click at top of viewport: Display View-Mode menu

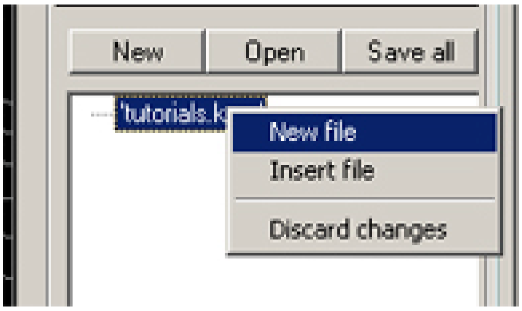
Keyboard Shortcuts

In the Asset Editor, the following shortcuts apply. Please note that in this version, these may only be recognized whilst a viewport has input focus.

Ctrl-Z	Undo
Ctrl-Y	Redo
Ctrl-S	Save
Ctrl-C	Copy
Ctrl-X	Cut
Ctrl-V	Paste
S	Toggle simulation
R	Reset simulation
A	Zoom extents all
Ctrl-A	Select All
Space	Selection Lock

Appendix B – Creating a ragdoll with the PSK importer

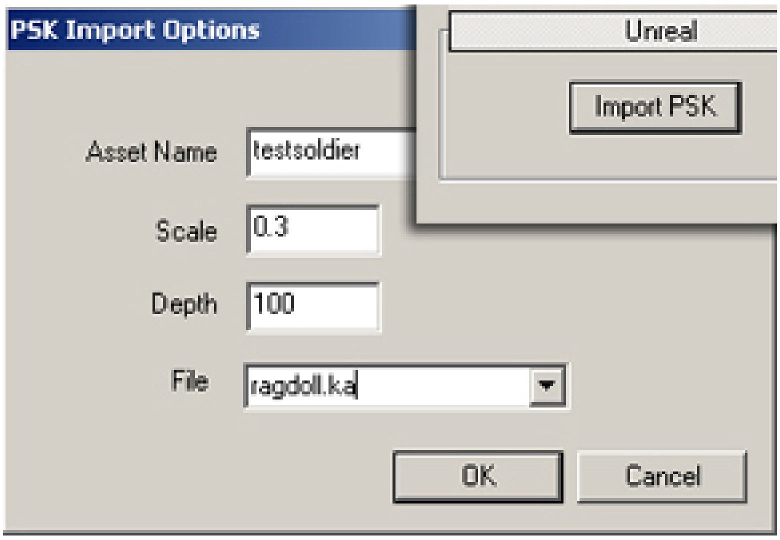
Step 1 : Create a new file



Ensure that you are working in your desired workspace – if necessary create a new one. Create a new file in the workspace by right clicking on the root item and selecting “New File”. Your imported asset will go into this file.

Ensure that the file’s resource path is set correctly so that the asset can locate its graphic.

Step 2 : Import the PSK



Asset Name: testsoldier

Scale: 0.3

Depth: 100

File: ragdoll.ka

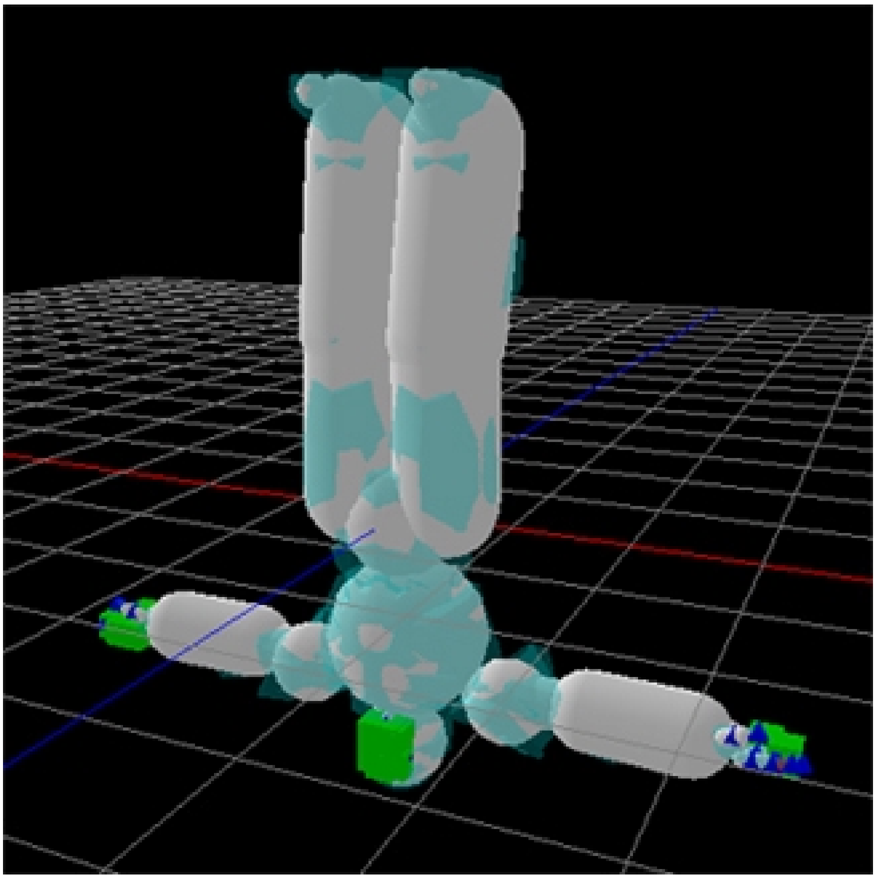
Click on the “Unreal” rollup to expand it and select “Import PSK”. For this tutorial we are going to use the TestSoldier.psk. After selecting the PSK you will get the Import Options dialog shown on the right.

Name the asset something like “testsoldier”.

The correct scaling factor for this PSK is 0.3. By setting the depth to 100 we are importing all the bones. See the section entitled “KAT Unreal Plugin” for details. Select the relevant file from the dropdown and click OK.

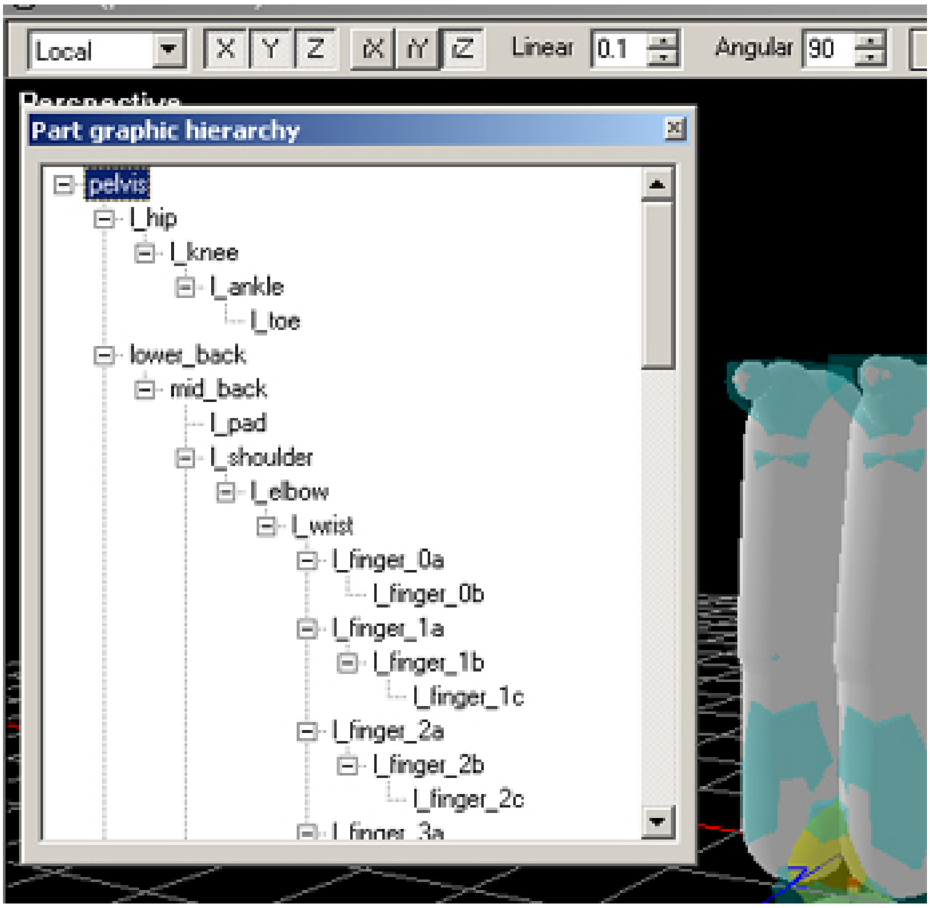
Step 3 : Begin editing

This is what the asset will look like. Briefly, the light blue is the skin, the white is the auto-generated collision geometry, the green boxes are dynamics only objects and the dark blue are centres of mass. The dynamics objects are dynamics only because they were considered too small to assign a geometry to.



Double click the testsoldier asset in the file manager to enter editing mode.

Step 4 : Apply correct rotation



Local
Linear 0.1
Angular 90

As you can see the asset is the wrong way up. To rectify this the root part needs rotating by 180 degrees about its local Z axis. Rotating the root will rotate the whole asset. To determine which is the root part click the “Hierarchy” button and select the root item which is the pelvis. This automatically selects the pelvis part in the viewport. Next, ensure that the movement options on the toolbar are set correctly (as in the image above).

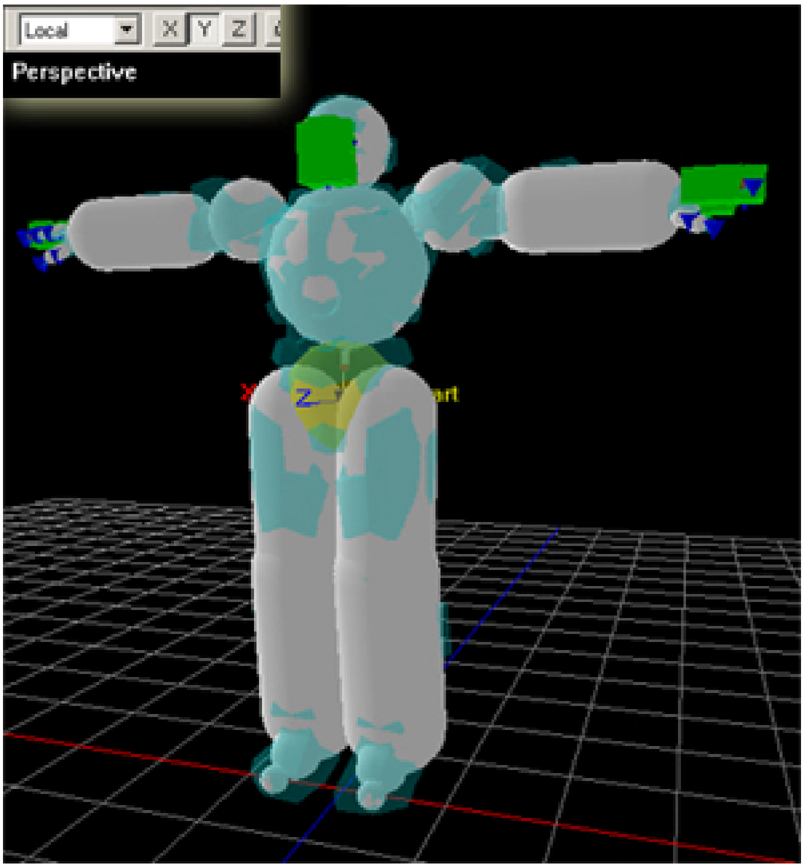
To perform the rotation you have a couple of options. The easiest way to do it is to hold down the middle button while holding down ALT. Dragging the mouse left to right will cause 90 degree rotation steps. Note that when using the middle button the cursor does not need to be over the selected item. The other, less convenient way, is to position the cursor over the selected item and hold down the left button and the ALT key – this way only works when the cursor is **over** the selection.

You may want to save your work at this point.

Step 5 : Apply correct translation

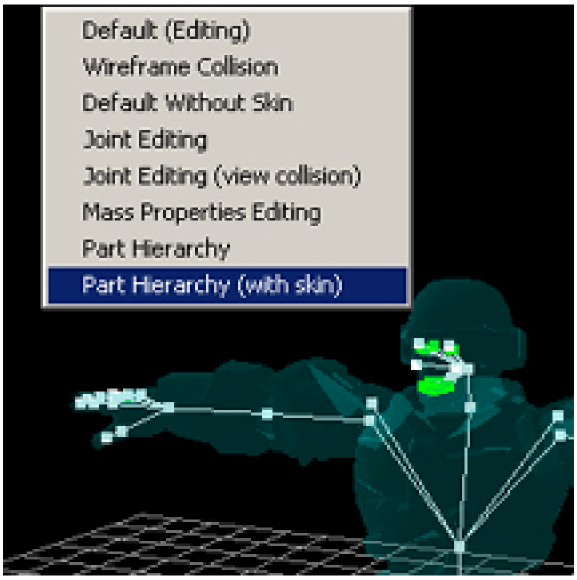
In much the same way as the previous step, apply a local (or world) translation to the root part so that the asset looks something like this.

Note that it is necessary to move the asset above the ground plane (or move the ground plane below the asset) because starting a simulation with a large amount of penetration is probably not what you want.

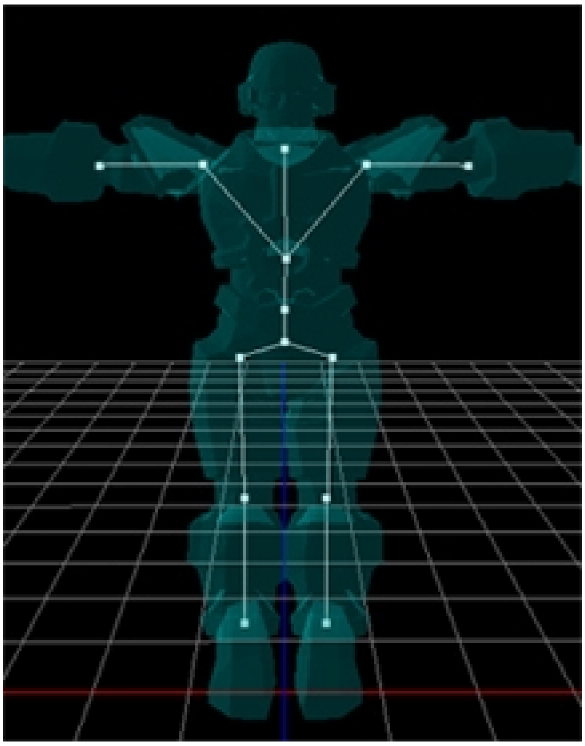


Step 6 : Reduce complexity

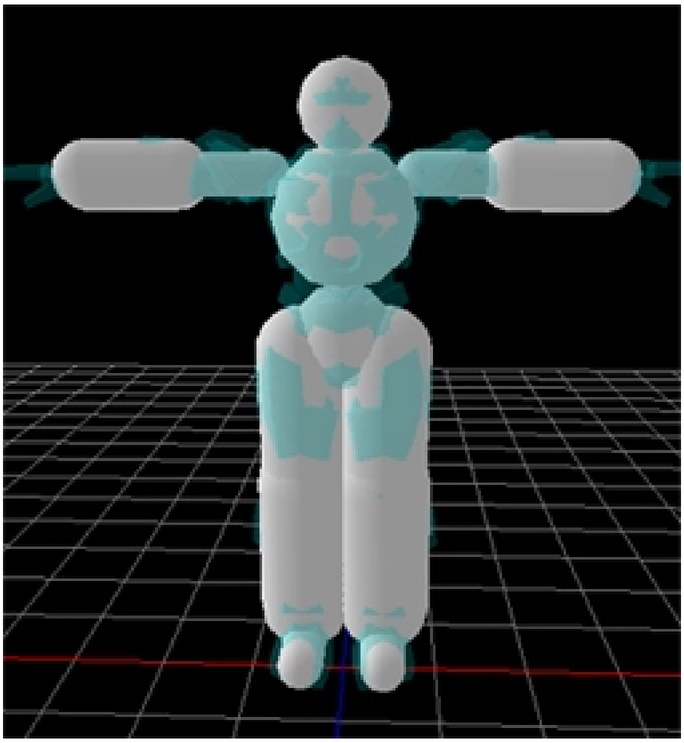
Right click at the top of the perspective viewport and select the “Part Hierarchy (with skin)” view mode. Drag–select over parts to select them and press DELETE to remove them. Remove all the fingers, the wrists, all the bones in the face and remove the head bone as we will use the neck for that part. Also remove the left and right shoulder pads and the toe parts. Additionally, using the part list or the part hierarchy dialog, select weapon_bone and delete it. This will get it down to an appropriate level of detail.



The asset should now look like this.



Step 7 : Adjust the geometry

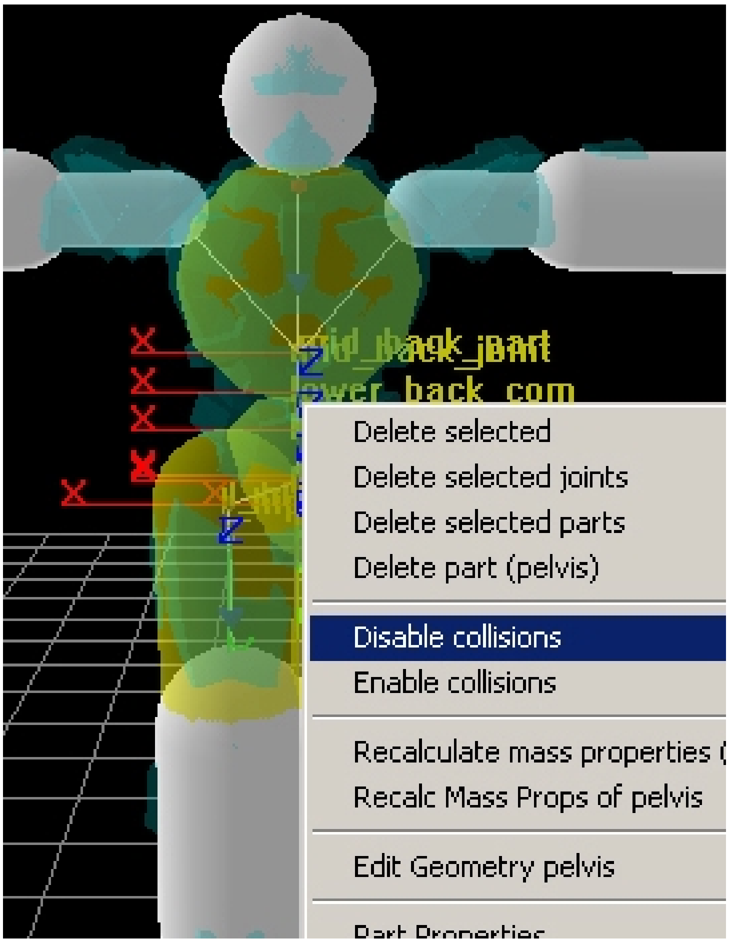


Switch back to the default view mode. You will see that the collision geometry for the head and feet are now inappropriate. Right click on the head collision part and select “Edit geometry neck”. Referring to the previous tutorial, replace the sphyl with a sphere that matches the head size. A radius of about 0.45 is fine. After pressing Finish, repeat the operation for the geometry on the left foot (l_ankle). You may want to rotate the sphyl and make it longer. Remember to adjust your angular snap value and axis constraint as appropriate.

You can use this same geometry for the other foot (r_ankle). To do this, select the geometry you have just edited and right click on the other one. Choose “Use geometry l_ankle for part r_ankle”. Once you have done all this it should look like the image on the right.

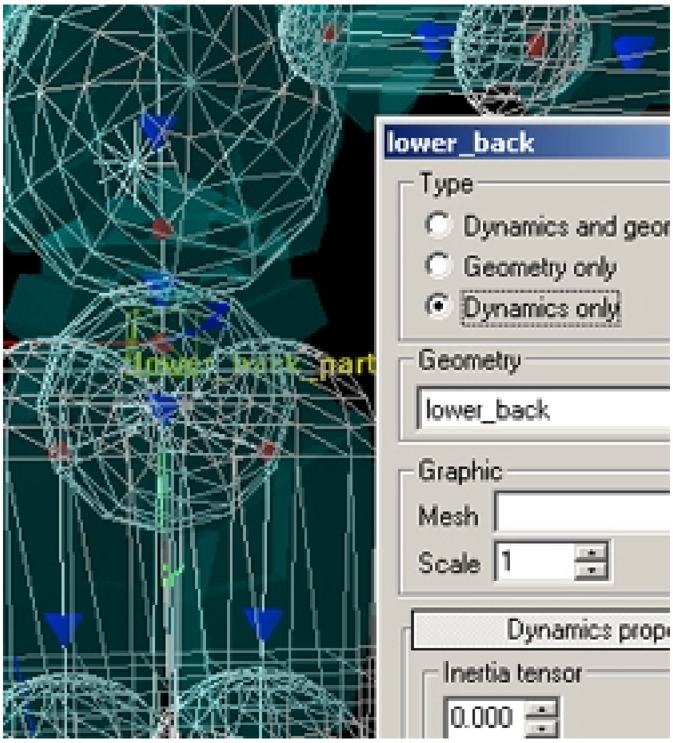
Step 8 : Disable collisions

Go through the model selecting pairs of parts or groups of parts, right click on them and choose “Disable collisions” from the context menu. The lower legs may start colliding from the reference pose, but you probably **do** want these to collide, so either move them slightly apart, or reduce their geomertry slightly, or just leave them initially colliding.



Step 9 : Make lower back dynamics only

Using the “Wireframe collision” view mode you will see that there is a part with collision geometry that is entirely inside another part (lower_back). Select it (using part list) and use part properties dialog to make it “Dynamics only”. It turns into a green box to show this. It is now worth recalculating the mass properties of the whole asset. Select all and choose “Recalc Mass Props” from the context menu. See step 25 of the crane tutorial for more information on this.



lower_back

Type

○ Dynamics and geor
○ Geometry only
● Dynamics only

Geometry: lower_back

Graphic

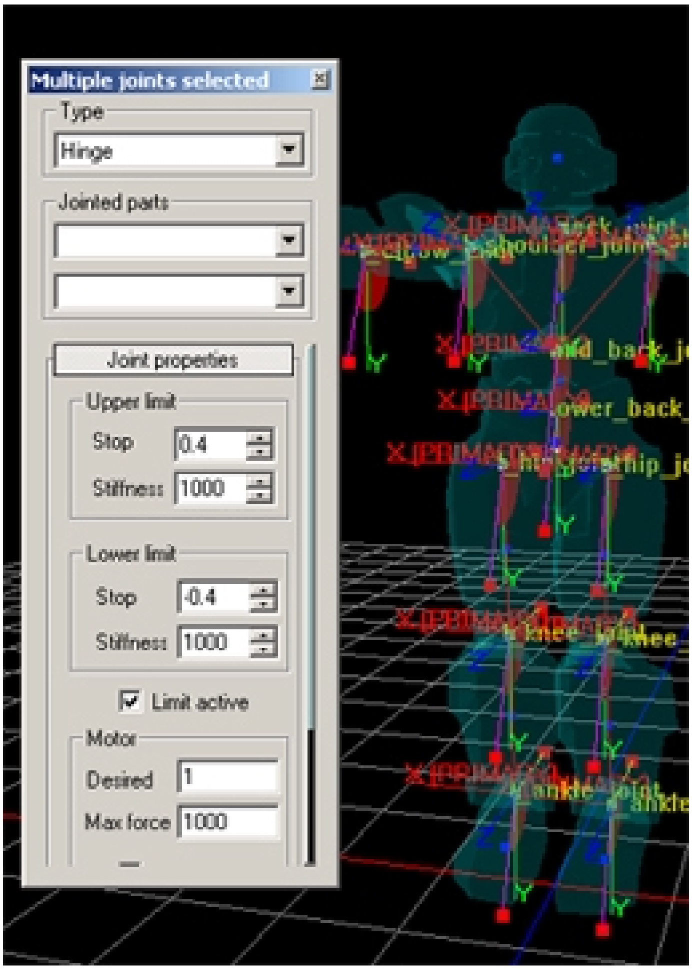
Mesh:

Scale: 1

Inertia tensor: 0.000

Step 10 : Change joint types

If you simulate the asset you will see that there is reasonable behaviour, but all the joints are ball and sockets, and unlimited. To begin with we will change all the joints to hinges. Switch to “Joint Editing” view mode, select all joints and display the joint properties dialog. Change the type to Hinge, enable the limits and set them to 0.4 and –0.4 as a reasonable starting point.



Multiple joints selected

Type: Hinge

Jointed parts:

Joint properties

Upper limit
Stop: 0.4
Stiffness: 1000

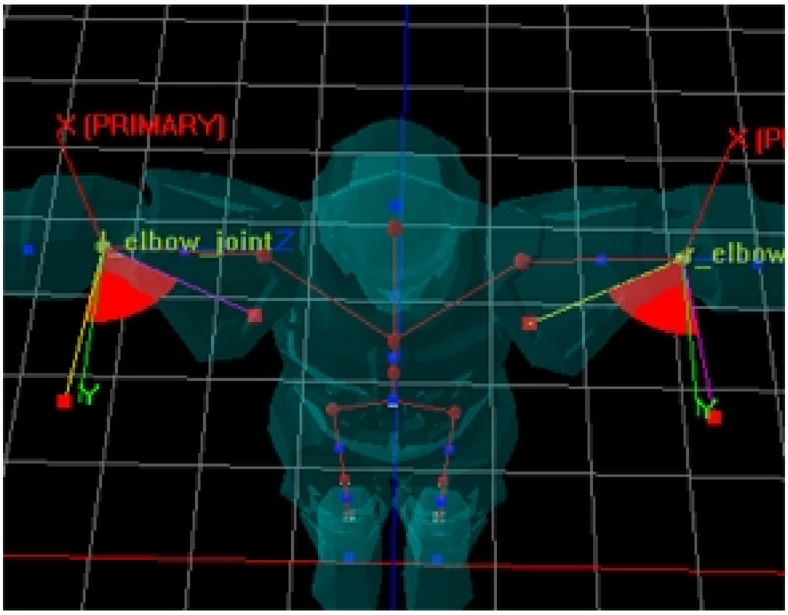
Lower limit
Stop: -0.4
Stiffness: 1000

☑ Limit active

Motor
Desired: 1
Max force: 1000

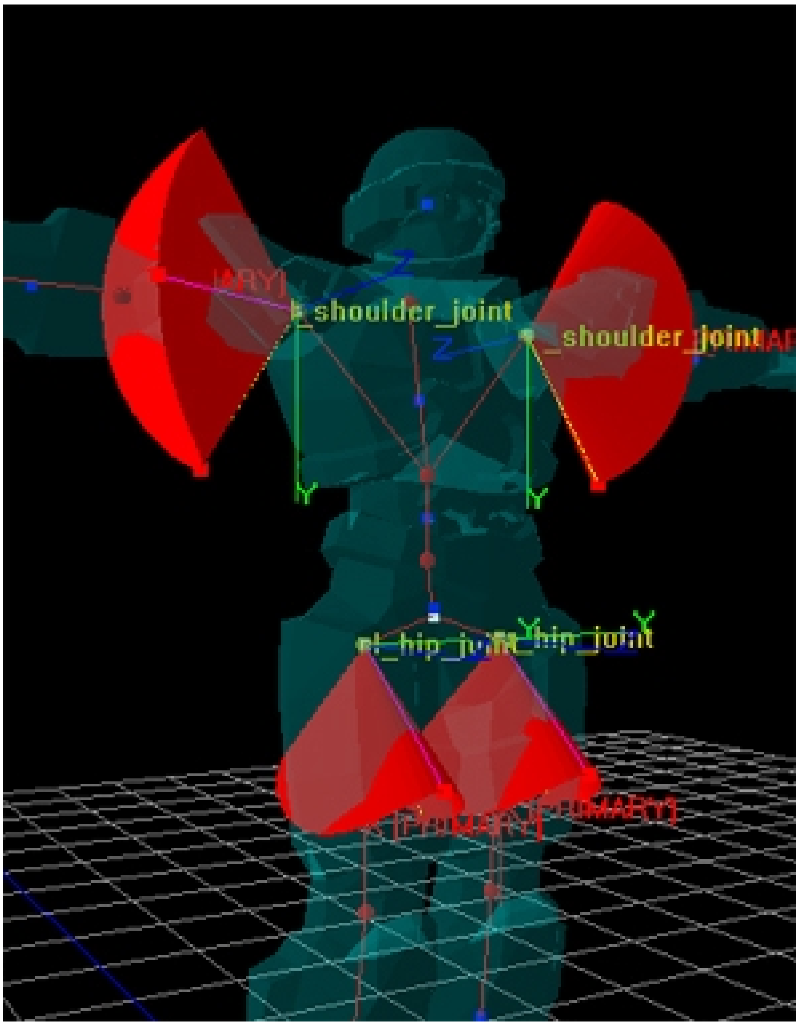
Step 11 : Adjust hinge axes and limits

Remembering to use the angular snap facility, rotate the elbow joints so that the primary axes are as shown in the picture. Also adjust the knee joints to give a sensible range of motion. Remember that Y indicates the initial position of the joint.



Step 12 : Change shoulders and hips to skeletal joints

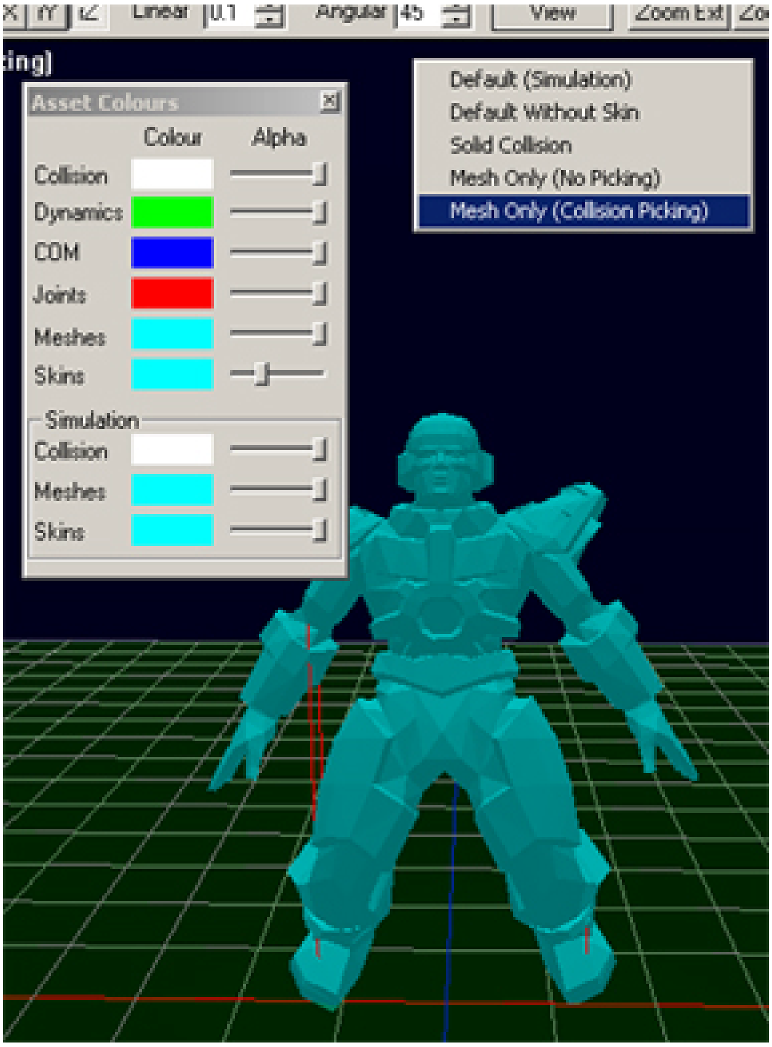
Select the shoulder joints, change them to Skeletal joints and rotate and adjust limits as shown. Do the same for the two hips.



Step 13

Simulate the asset. You may want to switch the view mode to “Mesh only (collision picking)” and fade up the simulation skin alpha using the asset colour dialog, to get a better view.

You now have a good ragdoll asset.



Additional things to try

- Add a bit of linear and angular damping to the parts by selecting all the parts and using the part properties dialog.
- Play with the simulation options.
- Try moving the parts to change the initial pose.

Appendix C – Known issues

- Remember that KAT is beta software. Report problems to kat@mathengine.com.
- There is a hard part limit of 256 per asset.
- KAT caches graphics that it loads. Some of these caches are not cleared during runtime.
- There are some minor File Manager and workspace wrinkles. If the workspace is not loading properly, edit the KAW file in a text editor.
- There may be viewmode bugs that occasionally cause nothing to be rendered in a viewport. This is often caused by the viewport configuration dialog. To rectify just change viewmode.
- The geometry editor does not support undo, however any complete geometry edit can be undone in the usual way.
- Geometry editor should be considered alpha and will be substantially reworked.
- Not all ASE options are recognized by the importer. Contact us if you have specific problems.
- The RPRO strength is not auto–calculated. If nothing appears to happen when you drag in simulation it may be too weak. Press F5 to edit the strength.
- The RPRO rotation is only in world space, not local.
- Dragging a hinge limit past 180 degrees makes it apparently disappear. However it has just snapped onto the other limit. You can easily drag it back.
- Stiffness and damping parameters sometimes produce garbage in edit boxes.
- Simulation options are not set on a per asset basis or persisted. However, the defaults can be edited in defaults.kat.
- Lighting normals for the DFF rendering are suspect.
- Keyboard shortcuts are only reliably intercepted when a viewport has input focus.
- The frame per second lock may not be accurate on some machines, especially Pentium 4s. Experiment with desired FPS in the Performance options utility rollup.
- Origin graphic not sensibly sized or reset.
- No checking to see if simulation is set up so badly that a frame will take arbitrary time to evolve. Can’t stop simulation until the frame has finished.
- Some dialog validation results in two message boxes
- Some dialog validation missing.
- Save As does not change the current asset to the newly saved one