

How to make your *humanoid* into a *PROTO*type

When using STEP on the example humanoids, such YT and Raven, there is a problem in the case that you want to use multiple instances. This problem is due to the fact that the names of the joints and limbs as defined by H-Anim 1.1 are declared by using DEF. Only the use of PROTOs may help here. So what do you need to do? Simply, you must (1) incorporate the names used by STEP in the interface of a PROTO for this particular humanoid, (2) make sure that these names are 'activated' (as explained below), and (3) announce to the STEP library that you use the humanoid interface instead of the hardwired names. However, first we will look at the original file.

the original file

The file containing the definition of your humanoid will look more or less as follows. See for example the *raven* file.

the original

```
WorldInfo { }
PROTO Humanoid [ ... ] { ... }
PROTO Joint    [ ... ] { ... }
PROTO Segment [ ... ] { ... }
PROTO Site     [ ... ] { ... }

DEF Humanoid Humanoid { ... }
```

The first part contains (apart from the WorldInfo) the PROTOs needed to construct the actual humanoid. The humanoid is created in the last line of this fragment by declaring an instance of the Humanoid PROTO, which represents the rest of the file. Within this part the H-Anim names are defined by using DEF, such as for example *hanim_HumanoidRoot*. As a remark, in STEP the prefix *hanim* is automatically prefixed by the STEP *ontology* component.

(1) create and modify the interface

The PROTO for your humanoid is easily created. Let's say you make a PROTO for *raven*. Then it suffices to do something like:

the modified version

```
WorldInfo { }
PROTO Humanoid [ ... ] { ... }
PROTO Joint    [ ... ] { ... }
PROTO Segment [ ... ] { ... }
PROTO Site     [ ... ] { ... }

PROTO Humanoid-raven [
  (a) interface of Humanoid
  (b) interface for STEP
```

```

] { # # body
DEF Humanoid Humanoid {
...
  (c) script to export H-Anim names
} # # end Humanoid
} # # end PROTO

```

Let me explain. As indicated in (a), you just copy the interface from the original Humanoid PROTO. After all your PROTO, which has as a name say *Humanoid-raven*, must offer the same capabilities as the original Humanoid PROTO.

Then, to make the PROTO STEP-compatible, you must include the interface declarations introducing the H-Anim definitions. And finally, you must include a *script* to connect the original H-Anim DEFs to the STEP part of the PROTO interface.

The STEP part of the PROTO interface looks as follows:

interface for STEP

```

eventOut SFNode Lshoulder
eventOut SFNode r_shoulder
eventOut SFNode Lelbow
eventOut SFNode r_elbow
eventOut SFNode Lhip
eventOut SFNode r_hip
eventOut SFNode Lankle
eventOut SFNode r_ankle
eventOut SFNode Lknee
eventOut SFNode r_knee
eventOut SFNode Lwrist
eventOut SFNode r_wrist
eventOut SFNode vl5
eventOut SFNode skullbase
eventOut SFNode humanoidroot
eventOut SFNode sacroiliac

```

In the online version, you follow links to templates file containing the fragments needed, so that you can adapt your own humanoid simply by *copy & paste*.

(2) activate the STEP names

To activate the STEP names, you must include the script given below.

script to export STEP names

```

Script {
  mustEvaluate TRUE
  directOutput TRUE

```

```

eventOut SFNode Lshoulder IS Lshoulder
eventOut SFNode r_shoulder IS r_shoulder
eventOut SFNode Lelbow IS Lelbow
eventOut SFNode r_elbow IS r_elbow
eventOut SFNode Lhip IS Lhip
eventOut SFNode r_hip IS r_hip
eventOut SFNode Lankle IS Lankle
eventOut SFNode r_ankle IS r_ankle
eventOut SFNode Lknee IS Lknee
eventOut SFNode r_knee IS r_knee
eventOut SFNode Lwrist IS Lwrist
eventOut SFNode r_wrist IS r_wrist
eventOut SFNode vl5 IS vl5
eventOut SFNode skullbase IS skullbase
eventOut SFNode humanoidroot IS humanoidroot
eventOut SFNode sacroiliac IS sacroiliac

```

```

field SFNode hanim_Lshoulder USE hanim_Lshoulder
field SFNode hanim_r_shoulder USE hanim_r_shoulder
field SFNode hanim_Lelbow USE hanim_Lelbow
field SFNode hanim_r_elbow USE hanim_r_elbow
field SFNode hanim_Lhip USE hanim_Lhip
field SFNode hanim_r_hip USE hanim_r_hip
field SFNode hanim_Lankle USE hanim_Lankle
field SFNode hanim_r_ankle USE hanim_r_ankle
field SFNode hanim_Lknee USE hanim_Lknee
field SFNode hanim_r_knee USE hanim_r_knee
field SFNode hanim_Lwrist USE hanim_Lwrist
field SFNode hanim_r_wrist USE hanim_r_wrist
field SFNode hanim_vl5 USE hanim_vl5
field SFNode hanim_skullbase USE hanim_skullbase
field SFNode hanim_humanoidroot USE hanim_HumanoidRoot
field SFNode hanim_sacroiliac USE hanim_sacroiliac

```

```

url "javascript:
function initialize() {
Lshoulder = hanim_Lshoulder;
r_shoulder = hanim_r_shoulder;
Lelbow = hanim_Lelbow;
r_elbow = hanim_r_elbow;
Lhip = hanim_Lhip;
r_hip = hanim_r_hip;
Lankle = hanim_Lankle;
r_ankle = hanim_r_ankle;
Lknee = hanim_Lknee;
r_knee = hanim_r_knee;
Lwrist = hanim_Lwrist;
r_wrist = hanim_r_wrist;

```

```

vl5 = hanim_vl5;
skullbase = hanim_skullbase;
humanoidroot = hanim_humanoidroot;
sacroiliac = hanim_sacroiliac;
}
"
} # # end script

```

The idea is simple, although the code is lengthy. First declare the STEP names as fields for the script and connect them to the interface using IS. Then declare the *H-Anim* names, and connect them to the appropriate DEFs using USE. And finally, export the STEP names by assigning the H-Anim component to the STEP-related field of the script. As a note, it appeared to be necessary to use *eventOuts* for the STEP names. Don't change this!

(3) usage of STEP

Now, finally, you must modify the *stepsetting.pl* file (in the older version the *step_kernel* object), by including the line:

```

step_kernel
humanoid_objects(true).

```

If there is already another fact with this name, but value false, then simply replace this line.

example

As an example, look at the *modified raven* file. Actually, you will see there the inclusion of respectively an extended interface and an extended script. These extensions concern the inclusion of limbs and joint declarations for the hand, to allow for manual gestures.

You might even go further and wrap the new *Humanoid* into another PROTO, as illustrated in the template I use for my virtual presenters. But for the moment, you are strongly discouraged to do that!

A. Eliëns