This text is a service of https://github.com/christoph-v/spark

MMF New

This hobby report describes how the "MMF paradigm" could evolve from a perspective at the beginning of step 0033.10 "Pieta".

This hobby report was a "snapshot" and should not be updated, but the version was upgraded to version 3.0 on February 18, 2018, to mark the beginning of the SPARK project.

Version 3.1 was created on March 30th, 2018, after translating to English language.

Version 4.3 is indicating the version at the first serious specification of the release "Arimathea" (step 0033.11), where the basic concepts of "Arimathea" seem to be stable now. No update done.

Note: the results of the present hobby report ("snapshot") were integrated to the appendix of the glossary of the Concepts' Descriptions of the SF project http://simulrr.sourceforge.net, where they are kept up to date.

1 The Question

As part of the "thorough review" that I started in the course of the activity "Rebase to MIB Core", I noticed that it is not possible with the current concepts to nest models together.

I wondered if it would not make sense to nest objects (ie MIDAS objects and models) in general without favoring or negating MIDAS objects or models.

The following flashes of thought came to this fundamental question:

1. I still have the concept of "Moving Modules" in my future concepts, which would make it unnecessary to nest models in one another, because then modules could be contained in models, which in turn contain models - this approach would not be "more straight forward"?

2. I still have the MIDAS object "Avatar Container", which can instantiate an avatar in a model. Are avatars not objects? There was still the idea of the "DM avatars".

All this I try now systematically to question and answer.

1.1 Basic assumptions about SMS models (MMF)

1. Models can be rendered (they are visible, audible, tactile, ...).

2. Models are always displayed relative to a module.

3. Either they have a "parent module" or they can not be rendered. In the second case, they are in the "detached" state.

4. To become multiuser, models use the so-called "MIDAS objects".

5. MIDAS objects are parts of the SMUOS framework that specialize in specific use cases (binary switch, carousel drive, ...) and help with the interactive animation and simulation of the models.

6. MIDAS objects can not be rendered, they make sense only in the context of models and / or modules (an exception is the idea of "astral" objects, which exist outside of modules).

This text is a service of https://github.com/christoph-v/spark

1.2 Basic assumptions about SMS modules (MMF)

1. A module models a (small) section of a virtual universe and represents the environment in which avatars and models can frolic.

2. A module spans a (quasi) Euclidean spacetime with a so-called "module coordinate system".

3. The relationships of these "module coordinate systems" to the world coordinate system of the Web3D Browser are FFS.
For the time being, we assume that the "module coordinate system" results from the world coordinate system by a constant translation and a constant rotation about the y-axis, and that the gravity acts in the (-y) direction of the world coordinate system.

4. Modules may include so-called "intrinsic models" that are implemented directly as part of the module, and may also use MIDAS objects.

5. Modules may contain so-called "static models" that have been provided separately from the module - and thus can be used by many different modules - but are simultaneously loaded, initialized, attached and disabled with the module.

6. Modules can contain so-called "dynamic models". These are models that can be loaded and unloaded independently of all modules and are each "assigned" to a module for a specific time. The "assignment" of dynamic models is a global process that applies to all scene instances of a multiuser session and is followed locally by an "attaching" the model instance to the module instance.

7. The change of a dynamic model from one module to another is called "handover".

8. Modules each contain one instance of the SMS module coordinator. This can not be rendered, but it coordinates the MIDAS objects of all models that are appropriate for this module.

1.3 Basic assumptions about the frame (MMF)

1. The frame integrates a number of one or more SMS modules into one VR / AR platform.

2. The frame makes the decision as to whether and when SMS modules will be registered, loaded and initialized or disabled and unloaded.

3. The frame is responsible for initializing the used multiuser system and provides the network connection used by the SMUOS framework.

4. The frame contains the central part of the SMUOS Framework, namely the "Simple Scene Controller".
Frames can also contain parts that are rendered, such as HUDs or CLIs.

1.4 Basic assumptions about classic avatars

1. Classic avatars are not SMS models.

2. Classic avatars have N instances, one pilot in the scene instance of the user represented by the avatar, and N-1 drones in the other

This text is a service of https://github.com/christoph-v/spark

1.5 Basic assumptions about DM avatars

1. DM avatars are dynamic SMS models.

2. DM avatars are models "with a special relationship to the user".

1.6 Basic assumptions about "Moving Modules"

1. With the help of a special MIDAS object - let's call it "Module Container" - it is possible to establish modules as parts of a model.

2. As a result, models can subsequently be modeled in models.

3. The MMF paradigm is therefore expanded to
    Frame -- 1: n -- Modules -- 1: n -- Model -- 1: n -- Modules -- 1: n -- Model -- ad. Inf.

2 Conclusions

The concept of Moving Modules is sufficient to allow models as part of models.

To the basic assumptions we will add the following assumption

1. Models do not build local coordinate systems, so you can not SMS models render relative to SMS models. You need a module between the two models, which actually builds the coordinate system.