The Universe In Operation

A player views the **Universe** through the viewpoint module of his home system. Through the filter of a situation simulator, a player takes the role of some *character* in the **Universe**. The depth and importance of any given character is partially a result of the behavior and style of the player, partially a result of the constraints imposed by the situation simulator, and partially a result of the events in which the character is involved.

The **Universe** that a player's characters encounter is a large and diverse place containing various worlds, technologies, artifacts, institutions, and other characters. All this stuff comes from a variety of sources. Some of it is generated directly by software in different parts of the system. Some is generated by semi-automated processes at Universe Central which are guided by the people there — the Gamemasters (or *GMs* for short). Some of it is placed in the system by the GMs manually. Finally, some is a result of more or less direct interaction between the player and other human beings, either GMs or other players.

The key to the existence of the **Universe** itself is the database and associated software that reside at Universe Central. The protocols of the player access server define, in some sense, the physical laws of the **Universe**, while the information in the database defines most of the contents of the **Universe**. However, Universe Central is not merely a passive database that simply stores and retrieves information on command. Rather, with the assistance of the GMs and their phantom army of software helpers, it is an active simulation that goes on independent of any particular player.

The players participate in this simulation through the situation simulators in their home systems. The particular situation simulator that a player chooses defines the nature of that participation. This is best illustrated by a few examples:

An example

One possible role that a situation simulator could put you in would be ''Mercenary Space Fighter Pilot''. This would not be a particularly significant role in the historical perspective of the **Universe** as a whole but would be an opportunity to play fly-through-space-shoot'em-up. You would (in essence) wander down to the local Mercenary Guild hall and get assigned a mission by the Guild's job-shop computer. You'd then hop into your space fighter and fly off to participate in somebody else's battle. The battle itself would then take the form of an exciting but fairly conventional shoot'em-up video game. If you survive the mission you get to try again, in the tradition of video games everywhere. If not,*c'est la guerr e*.

Actually, your home machine would probably be presenting itself to the **Universe** as the Mercenary Guild chapter of some particular city or world. The reason for this is that a particular mercenary pilot character has a ''lifetime'' of perhaps a few missions (a single session at the machine), or less if he gets himself blown up. You (the *real* you) on the other hand, remain more or less the same person. Skills learned on one mission (even an unsuccessful one) carry over to the next, even if you fly the next mission as a ''different'' character. Universe Central keeps track of how well you do on the missions you fly. Thus you represent (to the **Universe**) a pool of available pilots with a particular set of skills and ability ratings — in short, a Mercenary Guild chapter. The more able such chapters will, of course, be more in demand by other players playing roles in which they, among other things, hire mercenaries. Thus, the more successful chapters get paid more, with the total value of a chapter's contracts perhaps forming a sort of score for the player.

What actually happens

Behind the scenes, what's happening is this: your computer calls up Universe Central and says, basically, ''Hi, I'm player number so-and-so [with appropriate identification and authorization procedures, of course] running the 'Mercenary Space Fighter Pilot' module; give me a mission.'' Universe Central looks through its list of raging battles (in such a big **Universe** you can be sure that somebody's at war with somebody somewhere — if not, Universe Central can concoct a skirmish somewhere for you...) and picks one based on various factors that our designers have deemed relevant: ''location'' of the battle vs. ''location'' of the player, the player's recorded skill level, the nature of the conflict (some kinds of missions being more difficult or more complex than others), and so on. It responds with a message that says, essentially, ''OK, here's a type X mission on planet Y and here are some magic numbers to plug into your simulator.'' The ''magic numbers'' are the appropriate parameters and random number seeds that define the particulars of

the mission. The two computers then hang up the phone. The whole exchange lasts maybe fifteen or twenty seconds. The player then plays fighter pilot for a while, with his local machine handling all of the graphics and interaction. The parameters given by Universe Central allow it to generate the appropriate terrain to fly around in and the correct set of opponents to vie with. Finally, the local system calls back to Universe Central with the results: ''Well, the first time out he got blown up by a foobar. The second flight he killed 17 fnords and then crashed. On the third sortie he wiped out all opposition and returned to base unscathed.'' Again, the exchange only takes a few seconds. Universe Central notes the results, adjusts the player's ratings accordingly, and factors the information into the eventual outcome of the battle in which the player participated.

Another example

A more significant role to play would be ''World Leader''. The ''World Leader'' situation simulator would present a view that would be something like a graphically interactive version of *Empire*. This viewpoint module would let you represent the government of some world in the **Universe**. Depending on how the fantasy is set up, this could be either a single individual character (*Phearless Phred the Phirst, Emperor of Graustark*, let's say) or it could be a collection of characters representing various officials of the bureaucracy or the legislature (*senatus populusque Romanus*).

As ''World Leader'' you would have various displays showing the resources and forces available to you as well as ''maps'' of the local region of the **Universe**. Various commands would then be available to you to organize your economy, allocate resources, deploy military strength, and so on. In addition, you would be able to exchange messages with other players and interact with various individuals and organizations that can do things for (or to) you. Some of these other entities would be controlled by other players running situation simulators corresponding to the appropriate roles. For example, you might hire a starship to carry some trade goods to a distant market on your behalf. This starship might be operated by someone playing the character of the starship's captain, running, of course, the ''Starship Captain'' situation simulator on his home system. Other of these external entities would be controlled or mediated by GMs or software at Universe Central. For example, you might need to hire some mercenary space fighter pilots to help defend your world from an invasion. To do this you would contact the Mercenaries' Guild (an organization operated as part of Universe Central) which, for a price, would furnish you with the required hired guns, which you could then deploy and direct as your own. The mercenaries you would receive through such a transaction would be a mixture of software simulated characters and real players playing the mercenary role (such as yourself in the previous example). Ideally, of course, real players would be used entirely, but in reality there will probably always be a mismatch between the supply and the demand. The system is designed to rely on the use of software simulated characters in cases such as this while using the limited number of player directed characters to add a leavening of realism and human unpredictability to the mix.

How It Works

The essential organizing mechanism here is *multiple levels of abstraction*. Simulating an entire universe at the highest level of detail available in the game would swamp the capacity of any plausibly conceivable computer system, as well as taxing the faculties of players interested in anything other than the lowest level roles.

Consider traditional (board) wargames. A strategic level wargame never has the players maneuvering individual soldiers and pieces of equipment. Rather, the players deal with whole armies and the game mechanics abstract the lower levels of detail. Conversely, in a tactical level wargame the players *do* deal with individual soldiers and pieces of equipment but rarely do the players manage more than a relatively small number of such individual units at a time. Intermediate level games have intermediate levels of detail and abstraction. The complexity of the model that the player deals with stays more or less constant, while the balancing point between detail and abstraction varies. Similar analysis should apply to computer games. Experience tells us that it is neither practical nor necessary to model a whole universe in great detail.

What we must do is model the big picture at a fairly abstract level, while retaining the capability to model *pieces* of it at greater resolution. To do this we define a hierarchy of levels ranging from the highly broad and abstract to the highly focused and detailed. For each level there is a natural granularity of detail and a corresponding set of natural structural units. Each structural unit of a given level comes in two

different forms: one form is an abstraction of the level below while the other is actually built out of lower level units. This notion is illustrated (abstractly!) by figure 3.

The structural units at one level of abstraction are connected to the structural units of the levels immediately above and below it by standardized interfaces, so that the component software of one level is insulated from the internal mechanics of the levels beneath it. Thus a piece of software can perform the function of some structural entity either by actually calling lower level units and modeling the interaction between them or by approximating the lower level units itself (e.g., by stochastic methods). As illustrated by figure 3, an abstract structural unit in essence ''pretends'' to the units above it to be an entire sub-tree of the structure.

Actually, there probably won't be a clear dichotomy between the abstract structural units which fake their sub-units and the detailed structural units which use their sub-units *in toto*. It may be useful to construct a sort of ''semi-abstract'' structural unit that uses some sub-units and fakes others. For example, in the ''Mercenary Space Fighter'' scenario described above, the sub-units representing tactical combat at the individual space fighter level are offloaded onto the players' systems. The higher level unit simulating a battle uses the simulations running in player machines for some of the sub-units of the battle model while generating the rest using a more abstract stochastic process. At any given time the proportion of the individual combats making up a battle that are run directly (in player machines) and the proportion that are simulated in aggregate is a function of the number of players who, at any given time, are playing the ''Mercenary Space Fighter'' role on their home machines. This can range anywhere from 100% player simulation (the completely detailed case) to 100% use of the simplified stochastic model (the completely abstract case).