THE INTERACTIVE GAMING SIMULATION LANGUAGE:
AN INTRODUCTION

Douglas E. Van Houweling
Interactive Gaming Simulation Group
Center for Urban Development Research
Cornell University, Ithaca, New York 14850

INTRODUCTION

The Interactive Gaming Simulation Language (I.G.S.L.) is being implemented at Cornell University because a "computer lag" has developed in the gaming simulation field. The large scale gaming simulations available today typically use computing tools that were available a decade ago, thereby failing to take full advantage of the computer's potential for enriching the gaming environment.

I.G.S.L. is designed to serve the same purpose for practitioners of interactive computer gaming that languages such as SIMSCRIPT(1), GPSS(2), and GASP(3) serve for practitioners of computer simulation. The need for functions not offered by the traditional simulation languages is a result of the place gaming simulations occupy in the spectrum of simulation techniques. All simulations can be placed on a continuum ranging from man simulation to computer simulation. A man simulation utilizes human beings to model the system which is being simulated. On the other end of the continuum are the computer simulations, in which the structure and activity in the model are entirely expressed in terms of a computer program. In between the man simulation and computer simulation ends of the continuum is a large area of activity called man-computer simulation. In these simulations, the computer is used to model part of the simulated system, and human participants model the remainder. Gaming simulation can be either man simulation or man-computer simulation, but must always include one or more human participants.

The specific application area of I.G.S.L., then, is man-computer simulation, which is also known as computerized gaming simulation. In addition to performing the functions carried out by the computer simulation languages, I.G.S.L. provides additional facilities to improve the interaction between the computer simulation and the human participants which together comprise a computerized gaming simulation.

This paper analyzes computer support of gaming simulations on the basis of the twin dimensions of responsiveness and centrality of computing in the game environment. On the basis of this analysis, the specific design requirements for computer support of large scale gaming simulations are outlined, and the language capabilities included in I.G.S.L. are presented.

This research was supported by the Exxon Education Foundation, the Alfred P. Sloan Foundation, the Meigs Endowment Fund, and the Cornell University Program on Science, Technology and Society.

DESIGN DIMENSIONS OF COMPUTER GAMING SIMULATIONS

An application of computing technology to any particular gaming simulation can be characterized in terms of two continua. The first is the user responsiveness of the computer system employed. The second is the degree to which the computer is central to the game environment. As will be shown below, more responsive and central computing allows the computer to enhance the game experience of the participants in ways which extend beyond simply modeling a part of the simulated system.

Responsiveness

The first continuum, responsiveness, refers to the contrast between so-called "batch" and interactive computing facilities. Almost all of the complex, multi-role gaming simulations that depend on computer support rely on "batch" computing. In this mode the players meet and make their decisions in consultation with each other and the game director(s). These decisions are then read into the computer after transfer from the player decision forms to punched cards. The computer model, on the basis of these decisions and data on the simulated system, prints out information on the status of the simulation for the next time period, thereby providing the basis for the participants' next round of decision-making.

On the other hand, almost all of the general systems for computer assisted instruction rely on interactive computing because of its recognized pedagogical superiority over "batch" computing. In this mode the participant is asked questions by the computer and immediately replies in layman's language on a typewriter-like keyboard. If the questions need further explanation, the participant can request help from the computer system. If the answers are not among the correct possibilities, the system notifies the user and provides additional help in arriving at an acceptable answer. Thus, in a fully interactive computing environment, the participant would make decisions in response to questions from the computer simulation, and the results of those decisions would be available directly from the computer terminal without the need for a delay to process a "batch" job through the computer with its attendant clerical costs.

Centrality

Centrality is determined by the extent to which the computer plays a crucial role in the creation and maintenance of the playing environment of the gaming simulation. Thus, for instance, the computer would not be central to the game if it were simply used as a calculating device by the game director. On the other hand, if the players made all decisions in response to computer stimuli and all of their interaction with the simulation and other participants in the game were controlled by the computer, the computing facility would be characterized as highly central to the gaming simulation.

DESIGN REQUIREMENTS OF COMPUTER GAMING SIMULATIONS

The nine major design characteristics of large scale interactive gaming simulations which require computer facilities are discussed below.

High Capacity for Detail

Since the primary objective of most gaming simulations is to provide a simulated experience with a large and complex segment of reality, a high level of detail must be reproduced to support the realism of the game. The more detailed a gaming simulation becomes, the more important the computer becomes, for without it the participants are quickly overwhelmed with the clerical and rule enforcement requirements of the game. Moreover, as the computer plays a larger role in the gaming simulation, it becomes more central to the game environment and needs to be more responsive.

Variable Level of Structure

Gaming-simulations range along a continuum from highly structured, or rigid, to loosely structured, or free-form. In highly structured games, all player options are designed into the game before play begins and attempts to move the game beyond these predetermined boundaries are not allowed. In loosely structured games, a control team evaluates each player decision as it is made and determines the appropriate response of the simulated environment on an ad hoc basis. Loosely structured games obviously provide a much more flexible game experience for the participants. Due to the fact that all player options must be anticipated in advance of the gaming session in computer gaming, gaming simulations which utilize the computer as a central element are usually highly structured. It would be desirable, however, to provide for a mix of rigid and free form play by providing the staff with capabilities for overriding the computer portion of the model in order to respond to unanticipated decisions on the part of the players.

Wide Disciplinary Scope

The disciplinary scope of a gaming simulation is not only influenced by the complexity of the system being modeled, but also by the level of detail the game designer wishes to include. As the disciplinary scope of the gaming simulation increases, the need for responsive and central computing also increases. This is a result both of the increasing detail which usually accompanies a broadening of disciplinary scope and also of the increased difficulty of assembling a multidisciplinary staff to direct the game. The more help the participant can obtain from the computer, the less dependent successful use of the game will be on the availability of staff.

If the gaming simulation is to have multidisciplinary coverage, the computer component must not only include submodels from the relevant disciplines, but provide for substantial interaction between those submodels. Because gaming simulations are frequently changed and augmented, it is important that changes to one submodel require minimum alterations in other submodels of the simulation, and that new submodels can be added to the simulation with minimum disruption of the pre-existent simulation.

Variable Number of Participants

Most games include more than one human participant. This is especially true of gaming simulations because the elements of interpersonal cooperation and competition are often central to the processes being modeled. Depending on the goals of the gaming simulation, however, it may be desirable to keep the number of participants to a minimum. Thus, for instance, if a game is designed to provide increased insight into the interaction of one role with its environment, it may be quite costly to provide that environment through a group of human participants. As an alternative, a responsive computer model can often provide a realistic simulation of the actions of other relevant roles in the game, thereby obviating the need for the expenditure of other participants' time to create the environment.

On the other hand, if the richness of a multiple participant game is desired, the computer can provide a communications and coordinating facility among the players. In order to support a game designed to include both the single- and multi-participant modes of operation, the computer must be programmed both to simulate role players when the roles are unrepresented by humans and to see that a slow participant does not create a bottleneck in the game which might hinder the activity of the other participants.

Capacity for Distributed Gaming

Most gaming simulations require that all of the participants be located at a single geographic location. While having all role players in face-to-face communication adds an important element of interpersonal communication to the game, it is often desirable, for either pedagogical or financial reasons, to have the players geographically distributed rather than centralized. Such distributed gaming is possible if the computer is responsive and central to the gaming environment. The computer can create the common simulated environment for all participants and serve as a message processor for communications among participants in the game. This basic communications capability is already present in a multi-role gaming simulation if the computer is a central component of the gaming environment. The addition of a teleconferencing capability provides the players with a structured system of communication without making it necessary to leave the gaming environment.

Flexibility of Scheduling

Short duration games have the important advantage of being easily integrated into a curriculum, but they are limited in richness and complexity by their total length. Games which require more than a few hours for their execution are usually played in several sessions over a period of time. Unfortunately, it is often difficult to adapt such games to the needs of courses and training exercises in which the game is not the central focus. If the computer is a central element in the game environment, some of the problems inherent in complex games designed for play over an extended period can be ameliorated. The computer can preserve the current status of the simulation when requested to do so by a participant or game director. At a subsequent session, the computer can present a brief summary of action in previous sessions as part of the initial scenario. Using this capability, gaming sessions can be of varying duration, depending on the schedule of the participants.

Additionally, if the game director wants the participants' experience to focus on a particular period in simulated time, it is possible for the game director to play the game to the beginning of the desired period and, using the status saving feature, initialize the game for the students at the chosen point in simulated time. Finally, the capability to preserve the status of the simulation at will means that a participant can replay the game over selected portions of simulated time by saving the status of the game at the beginning of the selected period, playing the game forward to the end of the period, and then restarting the game at the beginning of the period for another trial.

Experimental Capability

When a participant is attempting to understand the behavior of the model or evaluate the impact of a particular decision on the simulated system, he needs to be able to analyze the output of the gaming simulation in a systematic fashion. This observational and analytic capability is more easily provided if the computer is central to the gaming simulation. If all participant decisions are reached and implemented through interaction with the computer, a complete record can be kept of those interactions. Furthermore, if the results of these decisions are calculated by the computer model before being communicated to the participants in the game, these results can also be made available for analysis. A responsive computing environment can facilitate the analysis of data collected during a simulation run if a rudimentary content analysis capability and a statistical package are integrated into the computing facility.

High Level of Transparency

Typically, as the level of detail and complexity of the gaming simulation increases, with a concommitant increase in the complexity of the computer component of the model, the simulation model becomes more difficult to understand. While the modularity of the model, the availability of observational roles, and the experimental capabilities available to the participant provide aids to understanding the simulation model, these facilities alone are not always adequate. A highly responsive and central computing capability can further alleviate the opacity of complex models by making documentation of the internal workings of the model available to the participant.

Graphic Capabilities

Most gaming simulations incorporate some type of graphic capability. In fact, most environments which are complex enough to warrant gaming as an educational or research tool are hard to deal with if graphic techniques are not used to supplement spoken and written communications. The more complex the gaming environment becomes, the more important good graphic techniques are to the success of the gaming simulation. If the computer is to be a central element in the gaming environment, graphic techniques must be a part of the panoply of media used for communication. Furthermore, these graphic capabilities should be integrated with the data analysis capabilities provided in the gaming simulation. Finally, the graphics should be responsive or interactive. The participant should be able to use graphic techniques to input decisions and use time-phased graphics to monitor change in the simulated environment over simulated time.

CAPABILITIES OF THE INTERACTIVE GAMING SIMULATION LANGUAGE

Responsive and central computing makes possible important advances in the design of gaming simulations, many of which were outlined in the section above. Even with today's computing technology, however, providing the computing services required to support the design features cited above is a complex and difficult task. The undertaking becomes much more tenable if the substantive design of the gaming simulation can be isolated from the computational aspects of the game implementation. This goal can be best accomplished through a programming system which includes the basic facilities needed to implement interactive gaming simulations.

The practitioners of computer simulation have developed a series of programming languages such as SIMSCRIPT(1), GPSS(2), and GASP(3) which eliminate many of the difficult and tedious programming tasks common to all simulations and allow the designer of a computer simulation to concentrate on the substance of the model. Probably because few large scale computer gaming simulations have been developed and because of a lack of communication between the practitioners of computer simulation and the gaming simulation community, no similar programming language has been developed to implement large gaming simulation models. However, on the basis of the special facilities needed to implement these models, it is obvious that such a language is needed if substantial progress is to be made in computer gaming simulation. The Interactive Gaming Simulation Language is an attempt at providing these capabilities, and is best described in terms of the facilities it is designed to provide.

Discrete-Continuous Simulation Capability

The basic task of the computer in a gaming simulation is to simulate those aspects of the overall model which are not simulated by player activities. While the computer is used for this purpose in most computerized gaming simulations, none of the well-known games utilize a simulation programming language. This gap between the computer simulation practitioners and the gaming simulation community has made large scale gaming simulations expensive to develop, maintain, and modify.

I.G.S.L. offers most of the facilities for computer simulation provided in SIMSCRIPT and GASP IV. These facilities include provisions for continuous and discrete updating of the status of the simulated system, list processing, collection of statistics on the simulated system, and the generation of random variates. Additionally, since I.G.S.L. is implemented as an expansion of a general purpose programming language, it also incorporates all of the very substantial power of PL/I(4), its base language.

Modularity

I.G.S.L. requires the simulation model and program to be organized into subsystems, each of which possesses the following characteristics and capabilities: an independent data storage area inaccessible to procedures in other subsystems, ability to communicate with other subsystems through a global storage area under the control of a simple and carefully defined protocol, control over the advancement of simulated time in the subsystem, and facilities for defining and communicating with the game participants (or roles) in the subsystem. These facilities allow each subsystem in the overall gaming simulation to be developed and changed with minimal impact on the remainder of the simulation.

Participant-Simulation Coordination

Because I.G.S.L. is designed to support a gaming simulation which allows a large and varying number of participants to simultaneously interact in a common game, a very substantial problem of coordination between the activities of the participants and the simulation model as a whole arises. For example, the decision of one participant may be dependent on computer simulation results which are, in turn, dependent on the decisions of another player who may be located at a different site. The computer program must allow the first participant to proceed with all decisions that are not dependent on the actions of the second participant, while simultaneously notifying the second participant that a decision is necessary. Furthermore, the computer program should, insofar as possible, keep the participants moving through simulated time at about the same rate in order to minimize the frequency of cases in which one player's progress is retarded by another player.

I.G.S.L. contains facilities for specifying dependencies between subsystems and roles within subsystems. A provision is made to include simulations of non-playing roles and to utilize these simulations to provide time-critical decisions if a participant is unwilling to do so after being notified. Furthermore, a priority system is included which provides faster computer response to participants with a heavy decision agenda, and slower response to those with a light decision agenda, thereby attempting to move all participants through simulated time at similar rates.

Computer Simulation-Participant Communication

Highly detailed, multidisciplinary, many participant, long duration gaming simulations require the ability to store and manipulate a large amount of textual material to support a realistic and convenient flow of information between the computer simulation and the game participants. The preparation and maintenance of this material is one of the major tasks in the construction of a large scale gaming simulation; and the task can become impossible if the material is included in the computer program itself.

I.G.S.L. provides a facility for the manipulation of the textual information which does not require program modification. The programmer is also provided with a convenient and flexible system for sending messages to the participant and checking the responses. Special attention is given to making the message handling system easy for an inexperienced role player to use as well as freeing the programmer from concern with the management of computer-participant dialogue.

Participant-Participant Communication

If a gaming simulation is to be used in a distributed, or multi-site mode, provision must be made for the participants to communicate with each other as well as with the computer simulation. I.G.S.L. is designed to allow any participant in communication with the computer simulation model to communicate with other participants if the game director so allows.

Furthermore, it will be possible to store and classify all such communications by subject, simulated time of transmission, real time of transmission, participant name, and role name so the communications can be reviewed by interested parties as needed during and after the gaming simulation. This storage and classification system will also apply to messages transmitted between the computer simulation and the participants, thereby providing a complete record of the dialogue for review in a systematic fashion. Finally, since all communications will be classified by subject and participant role, the game director will be able to monitor selected interactions and modify the text before it is sent to its destination, if such modifications are required or desirable.

Simulation Status Recording

Long duration, multi-session gaming simulations which are to be used for both pedagogical and experimental purposes need a flexible capability to preserve the status of the simulation pending its future restart. I.G.S.L., through a combination of data structuring techniques and service procedures, provides the programmer with the option of storing all of the information required to reinitiate the simulation on a disk or tape through the execution of one program statement.

Interactive Graphics

Incorporated into the design of I.G.S.L. are syntax and facilities for convenient use of graphic capabilities which are so crucial to successful gaming simulation. The routines that manage the textual communication also will be capable of storing and displaying graphic communications, including displays which are partially preformatted and partially dynamic. Provision will be made for graphic response from the participants using a variety of graphic input devices such as a graphic tablet, joystick, or light pen. Finally, one participant will be able to send a graphic display to another participant. The recipient will then be able to modify the display while the sender observes the modification. Thus the graphic capability will provide for inter-participant communication as well as communication between the computer simulation and the participants.

Statistical Analysis Capabilities

A real world decision maker is usually able to gain information on which to base his decisions by collecting and analyzing relevant data. A gaming simulation should give the participants the same capability. Furthermore, the ability to subject the output of the simulation to statistical analysis is an important debugging and validation tool.

I.G.S.L. will allow the programmer, using the statistical capabilities of the language, to specify selected program variables as "monitored" variables. "Monitored" variables can be analyzed by statistical commands which are included in the computer-participant communication system. The participant thereby will be provided with a means to analyze data without leaving the simulated environment and to use the graphic display capabilities of I.G.S.L. to present the results of his analyses.

In addition, some of the most important information a real world decision maker has available is his record of the communications between the key actors in the system. Since the communications system of I.G.S.L. will record and classify all communications among participants and the computer model, it will be possible to retrieve and study these communications on a selective basis. Of course, in both the areas of statistical and textual data, a given participant should only be allowed access to certain data, and the system will allow the game designer to specify the range of access available to each participant.

Self-Documentation

One of the most severe criticisms of large scale models results from the inadequacy of the available documentation on their assumptions and operation. Typically, documentation efforts are neglected because of pressure to complete the simulation and funds are depleted before the documentation can be done. The result is a model difficult to use for either research or pedagogical purposes and nearly impossible to maintain and modify. Even if some documentation is produced, it is usually difficult to relate to the specific outcomes of a run of the gaming simulation because it is not cross-indexed adequately with the computer simulation programs.

I.G.S.L. is designed to attack the documentation problem by providing for documentation of the computer simulation element of a gaming simulation to be produced as the computer programs are written. If well-structured programs are produced with explanatory comments on each block, I.G.S.L. will provide a facility to make these comments available to game participants. During use of the gaming simulation, a user will be able to get specific documentation for any section of the computer program. The graphic display capability, in conjunction with an automatic flowcharting system, will serve to make this documentation comprehensible and clearly indicate the relationships among blocks in the program.

Distributed Computing

The increasing availability of inexpensive, powerful micro-computers results in a need for the incorporation of support for distributed computing, that is, a sharing of the computing activity by a number of computers. For instance, by using graphic display terminals which incorporate small computers, much of the extensive dialogue between participants and the simulation on the central computer can be condensed to transmission of control codes between computers. These short control messages recall a particular message or display from the small terminal computer's mass storage, display it to the participant, prompt the participant for a response, check the response for appropriateness, and transmit a code for the response back to the central computer. While this mode of operation does not eliminate a need for the capability to transmit rather lengthy messages in special cases, it does greatly reduce the level of communications traffic needed to support a distributed game and makes interactive graphics financially feasible.

I.G.S.L. is designed to support both programmable and non-programmable remote terminals through its communications package. The programmer will not have to be concerned with the distinction between distributed or centralized computing operation while implementing a program. If the computer is communicating through a non-programmable terminal, it will carry out the full communications function. On the other hand, if the communication is via a programmable terminal, a condensed communications format will be automatically used, and graphic display functions will be executed by the terminal rather than the central computer.

CONCLUSION

In view of the revolution in computing techniques which has taken place in the last decade, the potential usefulness of computers in gaming simulation extends beyond the restricted arena of calculation, record-keeping, and modeling. Indeed, the computer can now assist with the interface between the participants and the computer model in such a way as to greatly increase the effectiveness of man-computer simulation as a research and teaching tool. The Interactive Gaming Simulation Language is designed to facilitate such an advance.

REFERENCES

1. P.J. Kiviat, R. Villanueva, and H.M. Markowitz, SIMSCRIPT II.5 Programming Language, Los Angeles: C.A.C.I., Inc., 12011 San Vincente Blvd., 1973.

2. Stanley Greenberg, GPSS Primer, New York: John Wiley and Sons, Inc., 1972.

3. A.A.B. Pritsker, The GASP IV Simulation Language, New York: John Wiley and Sons, Inc., 1974.

4. OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, White Plains, New York: International Business Machines Corporation, Data Processing Division, 1133 Westchester Avenue, July 1974.