

Digital Games as Simulations

Katrin Becker & J. R. Parker
Digital Media Laboratory
Department of Computer Science
University of Calgary
1 403 220 6784
parker@cpsc.ucalgary.ca

Keywords: video games, discrete event simulation, methodology.

Abstract

On a computer, all games are simulations, but not all simulations are games.

With the recent and rapidly growing interest in digital games, it has become important to examine this claim in detail. In this paper, the authors will examine the dichotomy between modern digital games and the more traditional view of simulations, and what implications this has for the development of both ‘pure’ simulations as well as games.

INTRODUCTION

Isn't the connection between games and simulations obvious? For much of the simulation community it appears so, but other disciplines make other connections and use different terminology. What difference does it make, anyway? Well, in many instances it does not matter much or at all. And yet, if computer games are simulations, then tools and techniques are being developed in the games domain that would be of interest to simulation practitioners of all stripes, and we need to know how to find them. It may also be valuable to avoid a duplication of terminology, thus making the literature of both subjects more accessible.

The relationship of games to simulations is of particular interest in the area of “serious games”, which are games whose primary purpose is something other than pure entertainment. This area includes games created for training and other educative purposes, games used for advertising, games designed for health-related, medical, and military applications as well as games used as vehicles to effect social change. Several elements set these kinds of games apart from commercial entertainment games. This

includes, but is not limited to the diversity of experts involved in the development of these applications, as well as the degree of influence they hold over their design and implementation. This difference is one that makes a clear understanding of the connection between games and simulations crucial. Not only do different expert communities use different vocabulary to express different ideas; they sometimes use the *same* terminology to express different ideas[1], and this causes difficulties.

Simulation has been an active application area of computer technology almost since the very beginning of computers, yet with a very few exceptions, CS professionals still seem to have a view of simulation that is different from that of most other interest groups. One noteworthy exception comes from the military, and the relationship between simulations, games, and training has been a long and fruitful one for them. Other groups do not share the same comfort. With more and more attention being paid to the use of games in educational and training situations, the view of simulation held by educators [2] is bound to have a greater impact on how simulations will be developed and used as well as on research and development in simulation.

In education, many view simulations as necessarily involving the simulation of realistic and real-life systems, and make sweeping assumptions about how simulations get used. This includes notions that fidelity and realism in simulations are essential to effective learning, notions that learners do not use games in ways other than those intended (which, by implication makes them less effective than simulations), and that winning always takes precedence over exploration in games. These notions tend to limit the efficacy of all simulations including games, whereas a better understanding of how games are in fact merely a subset of a larger category of

simulations affords opportunities for exchange of knowledge and a broadening of the view of games as an area of focus for emerging methodologies and tools.

THE NATURE AND STRUCTURE OF A COMPUTER GAME

Although a significant fraction of the North American population has played a video game, very few of those individuals have any real appreciation of what kind of technology a game depends on, or even necessarily that a game is a software object.

Looked at from the perspective of a designer or developer, a computer game can be described as a real-time system that uses input from a person with a keyboard, mouse or special purpose device (E.G. joystick) to control a simulation and a graphical and audio display. A certain degree of (artificial) intelligence is needed to control independent graphical entities with which to interact (E.G. opponents) and/or there can be a network interface with which to communicate with other real entities (network opponents). There is usually a significant amount of multimedia data associated with a game, in the form of graphics, video, and sound. Finally, game consoles (E.G. Playstation 2) and recent PCs can use multiple processors, and so a measure of parallel computation is sometimes used to accelerate play. Whether that is so or not, it is always true that an efficient implementation is always required - games constantly push the available resources, both processor speed and memory, of modern PCs to the limit.

A distinction should be made at the outset between games that are implemented on the computer, such as chess and solitaire, and games that require a computer in order to be played, like *Tetris* and *Doom*. It is the latter category that will be referred to as computer games. It is this class of games that requires the significant technology of the modern computer, and that is the study of so much recent attention to define rational methods of design.

A Quick Walkthrough of a Game

When a computer game begins executing, it loads the initial set of media files (sounds, images) as it displays a set of *splash screens* that are either purely decorative or that are asking the user for input (selection of player parameters and so on). The next

step is sometimes to play a short video and perhaps to enter a tutorial that explains the rules and the operation of the interface.

When game play begins, simulation time begins as well. It's generally true that game/simulation time advances at the same rate as real time, and that the game proceeds whether the player is actively participating or not. For example, in a racing game once the race begins the player can choose to race, follow another car, stop, or drive backwards – the other cars will continue to race each other, and time will advance. The graphical display is updated quickly enough so that the screen appears to be a window into the simulated world, and object motion appears to be continuous. This requires a minimum display rate (*frame rate*) of 24 images per second, but faster is better. Moreover, the computation associated with the graphical display takes most of the time between each frame.

So, in each 1/24 of a second, here what has to be done:

- Determine where all of the objects should be. 20%
- Detect any collisions and their time and result.

- Determine how the objects ought to behave, given their current system state, and what the next step in their plan should be. 15%
- Start playing any sounds resulting from actions.

- Determine which frame in each animation should be next. 8%
- Determine the camera position.

- Render all objects from the current camera position 57%

The order matters – objects need to be positioned before being drawn, and position often depends on interactions with other objects. Also, it should be clear that a computer game is a discrete event simulation, in the sense that the state of the simulation is displayed at regular intervals, and continuous processes need only to be calculated at those times.

Essentials of Digital Simulation

Let's begin with a good definition of a computer (digital) simulation, one that is non-threatening and, with any luck, uncontroversial. Wikipedia defines:

A computer simulation or a computer model is a computer program that attempts to simulate an abstract model of a particular system. ... Traditionally, the formal modeling of systems has been via a mathematical model, which attempts to find analytical solutions to problems that enable the prediction of the behaviour of the system from a set of parameters and initial conditions.

This definition unfortunately uses the word *simulate*. What simulate means in that context is to offer a working analog, in software, of the larger original system or model that reacts in all practical ways like the original, but in an artificially provided environment. Consider a spring hanging from the ceiling with a weight attached to the lower end. Pulling the weight down a few inches and then letting go causes the weight to be pulled up by the spring to a maximum height, then falling again, and so on in an oscillating manner. That's the real world.

The *model* is an equation that will provide the position of the weight as a function of time given all of the parameters of the system: the stiffness of the spring, the magnitude of the weight, and the amount of the initial displacement (this ignores air resistance and other minor effects, and perhaps more aspects of the real situation).

The *simulation* is a computer program that codes the equation of the model and allows an experimenter to enter the parameters. The simulation then builds a virtual environment within which time appears to pass, and the model appears to govern the motion of the weight on the spring. There is no weight and no spring, only the model and perhaps a graphical version for display on a computer screen. The passage of time in the simulated virtual world is under programmer control, and bears no necessary connection to the passage of time in the real world. A virtual world has been defined in which physical laws are what we say they are, and in which objects behave

as we model them.

Fidelity

The term *fidelity* refers to the accuracy with which the simulation represents or models the situation being simulated. Sometimes the simulation is intended to be a model of the real world, in which case what we mean by fidelity is clear enough – it is how well the simulation resembles the real situation. A difficulty is encountered if we become too religious about what we mean by *real*. Common usage in the field of educational technology, for instance, has it that simulations must simulate the real world, or they are not actually simulations.

On the other hand, real is not always what a simulation is modeling. A pretty clear statement of this has been made by the Defense Modeling and Simulation Office (DMSO) who refer to the system being simulated as the *simuland*, and clearly make a distinction between a simuland and reality:

However, simulands are not necessarily the “real world,” because many simulations do not intend to represent situations found in current reality. For example, they may represent the performance of proposed weapon systems in a hypothetical battlefield.

Virtual Worlds

So, it appears that both games and simulations define a mathematical construct that bounds the possible properties and activities of the objects being described. This would be a fair description of a *virtual world*. What is possible in these worlds is not restricted by the bounds of the known universe: the gravitational constant could be varied, for instance, or a substance that is massless but very strong could be created. In the case of many simulations, the purpose is to predict the results of an experiment, to try to define the end state of a situation. In some cases the simulation is intended to determine which set of parameters will yield a particular outcome.

A game often has as its goal to entertain. People often find a game entertaining if it offers a virtual world that has many real aspects, probably because the real world is so familiar. If time moves too quickly or slowly it is unnerving, and if gravity is too strong then our instincts are all wrong. It may be useful at this point to examine a few of the major genres of

computer game and see how they are, in fact, simulations.

Video Game Genres

A *genre* is a *type* or *class* of an object, and with respect to computer games there are many recognized classes. The essential ones are:

- ² **Action games**, including first person shooters (FPS) and fighting games.
- ² **Role-playing games**, including real time and turn based versions.
- ² **Simulators**, within which the sports genre, flight simulators, and driving games reside.
- ² **Strategy games**, including military games.
- ² **Puzzle games**, including pinball and tetris type games.
- ² **Adventure games**, which sometimes are text based.
- ² **Activity based games** such as Dance, Dance Revolution, that are based on external sensors that detect motions of the player.

Genre Activities

Shooters and fighting games require that the model universe represent reality in the sense that blows and weapons have the effect that you would expect, and physical activities including collisions must behave as they would in the actual world. They are in that sense a simulation of a restricted part of the real world.

Role-playing games can be turn based, and so might not have a real time aspect. Other role-playing games have the characteristic of shooters, in that the perspective is that of the individual, and the goals are local and tactical at the moment but support an overall global goal that results in winning the game. Like shooters, these require a fair model of the physical universe in order to be entertaining, or at least a consistent model of a fictional universe.

Sports games and driving games are the most obvious simulations in the group. Physics takes a huge role in these games, as collisions are a key feature and real-time activity is almost always essential. Flight simulators are amongst the oldest in this type. They have been used to teach flight skills for a long time, in some form, and the simulation (for example, a Link

trainer) predates the video game. Flight sims can be made into first person shooters quite easily (E.G. *Crimson Skies*).

Strategy games are in many ways a more abstract representation of the world, sometimes showing terrain or military units as icons. In this way they can be thought to be a high level simulation, giving a more strategic view than a tactical one. Indeed, this is the source of the name: *strategy* game. Their value is that the details associated with individual objects is sacrificed in favour of a higher level perspective. Frequently the issues are those of logistics, position, and general strength levels and goal selection rather than specific individual strengths or skills. The simulation is at a higher level than is the shooter, but has the same characteristics: the world must be a consistent model of a universe in order to be effective.

Puzzle games fall into a few categories. Some are clearly simulations of a physical game, like the computer pinball games, and so might not fall into the video game definition as defined here at all. Others, such as the famous Tetris, contain isolated parts that are simulations: geometry in the abstract and gravity, for instance. This class of game may well be the most interesting, and is certainly the hardest to get a firm grasp on. If there is a counter example to the 'games are simulations' theorem it is in this class.

Tetris is a good example of what I would call a *pure* video game. There can't even be an approximation of Tetris without use of real-time interactive technology. Kids play shooting games with water guns, and do role-playing with each other; strategy games have an origin in certain kinds of board games. However, the nearest non-computer game I know to Tetris is *Soma*, which was a 3D geometric puzzle that became popular in the late 1960's and 1970's (a computer version can be found on the internet at <http://yavsoft.com/soma/>).

Adventure games consist of a model of the physical environment and how that interacts with the player. In that sense, they are obviously simulations, but the degree of interaction and the resolution of the display have a large amount of variation, from simple text descriptions to graphics and positional sound.

Finally, activity based games are often direct measurements of game-like activities that would take place in real life. These games always require extra input devices: microphones, cameras, pressure sensors, and so on, so as to create an input from user

motions and activities. This is almost an identity simulation, with the game platform relegated to keeping track of performance.

WHAT CAN SIMULATIONS TAKE FROM GAMES?

Games constitute the single biggest influence on the advancement of PC hardware in the last decade, and this includes processor speeds, storage capacities, transfer rates, graphics capabilities and sound. The majority of after-purchase hardware upgrades are used to enhance gameplay [7]. With the ever increasing popularity of games also comes a higher profile for the role of simulations in general. Whether we wish to admit it or not, computer science owes much to the consumers' ever-increasing demands for more realistic, immersive game experiences. Nearly half of all freshman entering the field cite an interest in games development as a significant factor in their choice of major [8].

Until recently, the games industry has been largely discounted by the more 'serious' and rigorous scientific community. The data reliability and real-world performance essential in scientific visualization and visual simulations [9] and the kinds of environments implemented in games were too disparate to suggest that the simulation community at large could benefit from game technology, at least, in terms of software and design. However, the demands of present-day high end simulation and the kind of environments supported by games like *Elder Scrolls: Oblivion* are converging to the point where dual purpose games like *Full Spectrum Warrior* have emerged. This game, which was first released in 2003, has been produced in two versions: a scaled-down version produced for commercial release, and a substantially more involved one designed for and used by the U.S. Department of Defense as part of its training program. Other military groups, such as the Singapore Armed Forces are also actively exploring the viability of using commercial off the shelf (COTS) games for training [10].

State of the art technology in games can now be used to inform design in more serious simulations, from user interfaces and navigation schemes to file formats. Perhaps as important as the technology of games is the knowledge of the psychological issues surrounding them. Especially when simulations are used in training applications, the concepts of *flow* [11]

and *intrinsic motivation* [12] are important in understanding how to create interactive software that accomplishes its stated goals. Flow is a state of human perception, one in which the person is completely immersed or absorbed in their activity. It has been studied more fully since the advent of video games, mainly because game developers seem to understand how to create that state in players using the relatively limited multimedia tools at their disposal. Intrinsic motivation has been described as a motivation to participate in activities that maintain or agree with a person's self-concept; in more plain terms it refers to a desire to engage in an activity for its own sake. People who are intrinsically motivated work on tasks because they find them enjoyable, and not necessarily for another kind of reward.

Are these theories the reason that some players find video games so addicting? Perhaps. However, there is a recent theory that the brain desires sensory information because it craves the natural opiates that the sensations create [13]. This would imply that interactive game-based methods are a better way to encourage a greater participation in the defined simulation activity in some groups of people. The latter theory may explain sense of pleasure that many players receive from game play. Training sims can learn much from this if they are to continue to be successful.

REFERENCES

- [1] Shaw, Mildred and B. Gaines, *Comparing Conceptual Structures: Consensus, Conflict, Correspondence and Contrast*. 1989.
- [2] Alessi, S.M. and S.R. Trollip, *Multimedia for learning: methods and development*. 3rd ed. 2001, Boston: Allyn and Bacon. xi, 580.
- [3] Parker, J.R., Loose, K., and Verheyde, N., Objectives and Outcomes of a Senior Course in Computer Game Programming, *Western Canadian Conference on Computing Education (WCCCE)*, Mt. Royal College, Calgary, Alberta, Canada. May 4-6, 2006.
- [4] Parker, J.R., *Start Your Engines - Developing Driving and Racing Games*, Paraglyph Press, 2005.
- [5] Parker, J.R., *Cyber GameSpace: Simulating the Unreal, Simulation and Other Re-enactments: Modeling the Unseen Symposium*, Banff, Alberta.
- [6] J.R. Parker, Sonny Chan and Jeff Howell, *OceanQuest: A University-Based Serious Game Project*, Digital Games Research Association Conference (DiGRA 2005) Vancouver, BC. June 16-20, 2005.