Proper Handling of Real Players in Serious Gaming Studies

Seth N. Hetu and Gary Tan
Dep. of Computer Science
National University of Singapore
Computing 1, Law Link, Singapore 117590

Keywords: MMOHILS; real-time simulation; serious gaming; validation

Abstract

Serious Gaming and Simulation Studies nearly always involve a human component. Unlike other components, the human is inaccurate, and prone to generate input which is wildly deviant from the structured distributions researchers have come to expect in the field of simulation. Techniques that apply to filtering simulation input can, however, still be used for serious games. In this paper, we will discuss a generalized approach to extracting valid user input from real humans in serious gaming studies. We will provide examples from an ongoing study where possible.

Terminology

The field of serious gaming forms a bridge between the fields of simulation and gaming. Unfortunately, the term itself is quite vague, and lends to multiple definitions. For the purpose of this paper, we shall consider the term a sensible combination of its composites:

- A ***model*** is a mathematical or statistical representation of a real-world phenomenon. A model can be a simple formula (e.g., `F = ma`), or it can be a collection of input distributions and service times —describing, for example, a factory in operation.
- A ***simulation*** involves the running of a model with various (possibly random) input parameters in order to generate output and reason about the nature of the model.
- A ***game*** is an interactive activity —with some element of play— that users take part in.
- We define a ***serious game*** to be a game with some ulterior educational motive. We focus on games used to generate input for simulations, but one might also proceed contrariwise, using a realistic game to educate users. We treat ***serious gaming*** as an all-encompassing term applied to these types of studies. Although both terms are to some degree buzz words, our work applies in a universal sense, and it requires a term to classify both game-heavy and simulation-heavy variants of this field of study. Where possible, we will use the terms ***game*** or ***simulation*** specifically.

1. Introduction

Serious games serve a variety of purposes. Some of them aim to educate players, while others are geared to study a particular phenomenon using real-life participants. (c.f. Leigh, 2007) Mixed-reality and social games can sometimes be loosely classified as serious. The game supervised by Cheok *et al* (2006) involved two teams playing Capture the Flag using GPS receivers and mobile phones. Virtual teammates helped to coordinate the outdoor members. Likewise, the *Big Urban Game* and *PacManhatten* —described by McGonigal (2006)— were designed "to explore what happens when games are… placed in the larger "real world" of street corners and cities". Both games involved overly large representations of classic gaming paraphernalia; inflatable player pieces, dice, gigantic game boards, and people dressed up as heroes and enemies. All of these may be considered serious games. Often, the variety of serious games seems so diverse that there could be no underlying element common to all types of game. However, that is not true: as all games involve ***play***, all games must therefore involve ***players***. Thus, an issue common to all serious games is how to gather the often sporadic input from real-life players and filter it into a game (physical or virtual) such that the game remains a valid representation of what was intended.

Many studies attempt to manage users under the abstraction of an agent. According to Russell & Norvig (1995) an intelligent agent is a unit of autonomous processing which perceives and modifies the world through sensors and effectors, and aims to maximize some performance measure. By definition, agents only have local knowledge of the world, and their operational instructions do not depend on other agents' resources or contributions. Thus, the analogue of humans to agents makes sense at face value. This approach is particularly advantageous for studies which mix virtual human players with software agents; i.e., it allows humans and agents to interact using the same functional specification. Unfortunately, the "human-as-agent" approach is a gross oversimplification. Agents have a ***limited*** and ***known*** set of actions, and the goal is to build these into complex "emergent" phenomena. With humans, the goal is to filter their ***unlimited*** and ***unknown*** actions into a behaviour set that allows them to interact properly with the virtual world. Moreover, while a software agent's behaviour in isolation is known, a human's behaviour in isolation is a black box.

Figure 1 — *In the* Big Urban Game*, players carried larger-than-life pieces around a gigantic game board.*

The frailty of this assumption becomes vividly apparent with the tiniest scrutiny. For example, Penn State began a study involving real users taking virtual classes in Second Life. Apparently at the behest of employers, several students sat in virtual classrooms and performed normal classroom activities. This was found to "encourage peer learning and more engagement among students". (Penn. State 2008) Although at first glance this might seem like a hallmark example of a serious game, overall confidence in the method is shaky. For example, how was it ensured that students were actually in front of their computers. Were webcams and microphones used? Perhaps students reported enjoying the experience because they were under less scrutiny and had more freedom. And how exactly was "more engagement" quantified? It seems unlikely that a virtual classroom could possibly be more engaging than a physical one, unless multimedia content was integrated into the lesson. This study is not necessarily invalid, but it would boost confidence in its assertions if input and output were clearly defined, and speculation was backed up by provably hypotheses.

Overcoming these difficulties is beneficial to the research community. Serious games require a level of validation which is not necessary in games of play. They compensate by allowing researchers to experiment with "live" users in a much larger worldscape. Various dangerous or expensive real-world events can be virtualized and studied so long as the human users in those studies are generating valid data.

1.1 Problem Formulation

Dealing with real users requires overcoming three hurdles:

- **The Misbehaviour Problem**. Whether out of ignorance or malice, players may act in ways which deteriorate the quality of a serious game. How can we avoid misbehaviour, or at least limit its impact on our data?
- **Input Acquisition**. What kind of input should we expect from users, and how do we create the artificial parts of our world so that they accurately reflect their real-world counterparts?
- **World Design**. What is the proper way to design a physical or virtual world space so as maximize the amount we can learn from a serious game?

Several techniques exist within the field of simulation which serious gaming can leverage to deal with such problems. We combine these techniques with our own to create a generalized framework for serious gaming studies involving real users.

In this study, we focus on serious games which follow the scientific method — specifically, those with a hypothesis to prove or disprove. We also have a bias towards virtual games instead of real-world games. However, this does not restrict our approach to a particular set of serious games; rather, all games which feature players can benefit from some or all of our techniques.

1.2 Related Work

If the goal of a serious game is formulated as a hypothesis, then the point of such a study is to answer the question, "Is this hypothesis correct?" This formulation mirrors some aspects of traditional simulation studies. (Unlike serious games, simulations deal with models rather than with the physical system.) The following steps are part of any simulation study. First, the modeller must **define the problem space**, clearly setting out the boundaries and objectives of the study. Next, the **conceptual model** can be defined. In particular, this step lists required input, generated output, and the algorithms used internally. Then, the modeller can **collect input data** and **construct the model**. Input data includes distributions of arrival times, properties regarding the environment, and other inputs to the system. (A serious game will require some of these inputs; the rest will be generated in real-time by the players.) The next step is to **verify and validate** the model. This involves checking that all parts of the model (algorithms, input data, etc.) are constructed correctly *and* accurately represent the real world system. This step is particularly important for serious games; the authors feel that this makes the difference between a truly **serious** game, and one that simply claims to be serious. After this, one must **design the experiments** which are expected to maximize information gain. Serious games may also benefit from repeated execution, as this increases confidence in their outputs. The **simulation run** occurs next, with **output data collected** so that one can later **analyze the output data** and **document his results**. At that point, one might choose to **expand the model** to cover new cases, or to re-apply it (which requires an **accreditation** stage in addition to the verification and validation one). (c.f. Smtih, 1998)

Serious games should adapt themselves to the simulation lifecycle so that they can benefit from the maturity of this method. For example, if input data is defined before the model is constructed, then the player's interface can be easily specified without leaving any room for ambiguity. Also, if verification and validation take place before experiment design, then the team will not waste any time designing experiments for an invalid model. Even a simple game can align itself with this approach and reap the benefits.

In addition to the general simulation approach, several specific techniques have been introduced which deal with acquiring data from human entities. Hopkinson and Sepulveda (1995), for example, provided a real-time layer of role validation for human-in-the-loop simulation studies. This approach uses case-based reasoning to determine if humans are violating the "role" that they were assigned upon entering the virtual world —if they are misbehaving, in other words. One of our previous papers leveraged this technique and discussed several others which could be used to gather and validate human-generated data from Massively Multiplayer Online Human In the Loop Simulations (MMOHILS). (c.f. Hetu and Tan 2008a)

1.3 Moving Forward With a Generalized Approach

Current techniques in the field of simulation are sufficient for producing valid serious games dealing with real users. However, these techniques are laden with domain knowledge from the field of simulation, and may be difficult to adapt properly to real-life serious games. Moreover, certain assumptions (such as the ability to repeat simulation runs at minimal cost) may not hold for games with real users. Hence, the remainder of this paper will focus on a generalized approach to gathering valid input from potentially misbehaving users in serious gaming studies. Two sections will follow, one on world design and one on general misbehaviour. Each section will present the theory behind its approach, and then give specific examples from an ongoing study of our own (c.f. Hetu and Tan 2008b). We will then conclude the paper with some final considerations.

2. World Design

Both virtual and physical games require world design, although the latter to a lesser extent. World design consists both of creating the perceived game space and specifying rules regarding the players' interaction with that game space.

2.1 World Design —Theory

Generally, the first step taken when creating a serious game's world is to define the actors which take part in the game. Actors include human and software agents, objects, adversaries, and anything else that interacts within the game space. Actors have properties, such as their size and weight, which must either be measured (real games) or modelled as input distributions (virtual games). Defining actors also requires defining their basic actions. Real-world games require this step as truly as virtual games do; for example, real-world humans playing a maze-like game may be restricted from stepping over marked lines on the floor. Constructing large walls to physically limit movement is not necessary; we simply restrict the users' actions to avoid stepping over lines. We will deal later with the possibility of users misbehaving and stepping over the line.

We will use the term "players" to refer to the human users in our serious game, and "actors" to refer to the virtual representation of the players which take action within the game. In most real-world serious games, the actor and the player are the same entity. Other researchers may choose to use their own domain-specific terms to further categorize entities in serious games; for example, a graphical representation of the player might be called an "avatar" in some contexts.

Given actors and their associated actions, we then define a set of rules —a set of "if, then" pairs which clarify the users' interactions in a general fashion. Unlike user actions, which rigidly limit the behaviour of players in our world, rules define points of interest that should be logged. For example, instead of restricting the users' actions to avoid bumping into other actors, we might say "**if** the actor bumps into another actor **then** have both actors move backwards as though two perfectly elastic spheres collided".

Virtual worlds should define input as a set of rules to capitalize on the potential to log this input. For example, "if the player holds space bar, his actor moves forward". Input rules should be considered client code, and they can generalize where other rules are specific. For a complex online world, we might have an input rule like: "**If** the user presses space, **then** his avatar will try to activate the nearest object. If nothing is nearby, the user will jump". The corresponding rules regarding "interacting" and "jumping" will need to be more specific, of course.

With agents and rules defined, the next task is to identify the world and its rooms. The world is simply defined by its geometry. One might say "we operate on a flat plane" which necessitates either finding a suitable location (in the physical world) or designing it (in a virtual one). The world is made up of minimally-interacting spaces called "rooms", which can be of any size. A game may have one room or several, depending on its requirements. Rooms may represent inter-connected areas of interest (like buildings) or alternate areas to study (e.g., different mazes for a *Pacman*-style game). Rooms may be grouped together under a label (e.g., "floor 1, which may be referenced by rules ("when any user enters floor1, sound an alarm").

2.2 World Design — Practice

A proof-of-concept for a massively-multiplayer online human in the loop simulation (MMOHILS) was developed with the principles just described for serious gaming studies. The world was a large pedestrian movement exercise, featuring 20 to 30 players per room.

The actors are small human avatars. A brightly-colored player represents the current user, and dull-colored ones represent other players. The breadth and depth of the actors are defined according to Fruin's data (1987) as 0.58m and 0.33m respectively. The world's rooms are scaled accordingly.

Figure 2 — Three actors on a virutal playing field. The ellipses indicate the actors' collision space and heading. The current player's avatar is brightly-colored.

Actors can move around the virtual world, and collide with other actors and with the environment. The input rules they obey are as follows:

- If the mouse is moved, then a "footprint" cursor is moved to the current mouse's position.
- If the user presses the left mouse button, then the avatar moves in that direction.
- If the avatar is moving, then she rotates to face the direction she is moving.

The remaining rules are:

- If a player turns more than 90 degrees at one instant, then he will move at half speed. In addition, a case-based reasoned is used to determine if this wild rotation is considered bad behaviour. (see Section 3.2)
- If a player collides with a wall (modelled as a rectangle) or another player (modelled as an ellipse, as previously mentioned) while moving then his velocity vector will be shortened to the point where a collision does not occur.
    - Note that bounding boxes are used to improve performance of the collision-detection algorithm.
- If a player collides with a wall or another player while rotating, then he rotates in reverse until he reaches a collision-free location (at worst, rotating all the way back to his initial position before the move).
- If the game is just beginning, then all players are moved to their starting locations and given an initial heading.

There are several rooms in our world. One room is 8m square, and can accommodate up to four flows of traffic. Another is a corridor, 10m by 4m, that can accommodate 6 people abreast. The final two rooms are also corridors, but with the right half bottlenecked to 2m and 1m respectively. Each room is meant to be a study of a different pedestrian condition, under varying degrees of cross-flow. The perspective of each world is top-down. Users are free to wander each room for several minutes before each simulation run, and are then automatically moved to their starting positions and given a countdown.

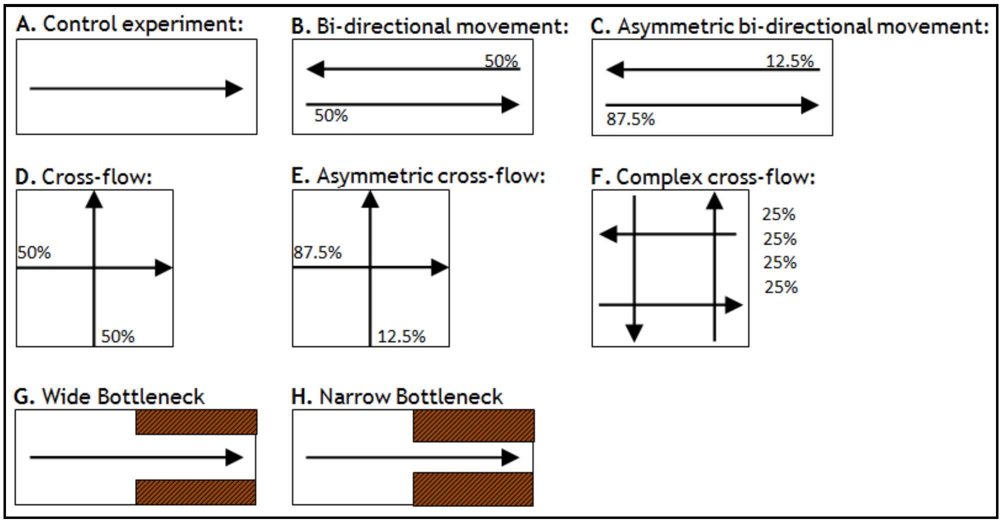

Figure 3 — Seven experiments plus a control, taking place in four different environments, each with various degrees of cross-flow.

3. Player Misbehaviour

Even the most well-intending players can misbehave in the context of a serious game. "Misbehaviour", in this discussion, means that a user is violating the role which was set for him during the course of one instance of the game. This can be a simple issue, such as failing to keep in line with his actions (e.g., stepping over the line), or it can be a human interface issue, such as a lawyer in a "courtroom" game who cannot type fast enough to raise objections and the like. It could also be a variety of meta-game cheating, such as implicit collusion between two players on different teams who are friends outside the game space, and agree to target each other last. Finally, it could involve very sophisticated hacking of the game client, bending of the rules when "no-one is looking", or maliciously trying to distort the data gathered from this serious game. A riot policewoman in a pedestrian control simulation who obstructs peaceful citizens and lets unruly ones roam free would fall into this last category.

3.1 Player Misbehaving — Theory

Although the various types of misbehaviour may seem daunting or even arbitrarily theoretical, these are real concerns within serious games, and as such each one must be dealt with summarily. In some cases, things like collusion may actually be part of the intended study. Or, if cheating cannot be stopped, it can at least be logged. Someone stepping over the line in a path-finding exercise may only invalidate his own data, not the entire study's data.

Misbehaviour arises from either players' ignorance or bad intention. That which arises from ignorance should be mitigated. A "help key" listing a player's current objectives can be assigned, or the user can be given written directions. Trial runs can be performed under full supervision, allowing supervisors to train players how to respond properly. Real-time alerts can be provided to users of virtual systems when they misbehave. Keeping all virtual players in the same computer lab for the duration of the game is also important. In terms of detecting misbehaviour, debriefing sessions may occur after each run, asking detailed questions which help to determine the players' state of mind during game play. (c.f. Hetu and Tan, 2008a)

The larger (and more expensive) a serious game gets, the more important it becomes to detect and correct errors while the game is running. Supervisors in a real-world game may pull players aside if they misbehave, and instruct them how to behave in the future. They may also remove the player from the game after a series of disruptive activities have occurred. Although this itself taints the data gathered by the game, it is quite likely to do less damage than letting a rogue player continue to act unchecked.

For virtual worlds, a form of case-based reasoning can be used to detect violations of role. (c.f. Hopkinson and Sepulveda, 2005) At that point, players can be prompted, automatically corrected, or removed from the game. An added benefit to running a virtual game is that all deviations from role can be logged, and the total error introduced by these can be quantified.

3.2 Player Misbehaving — Practice

Several design strategies were used to manage player misbehaviour in the pedestrian MMOHILS mentioned earlier. First, to minimize misbehaviour due to ignorance, we allow the player to wander the map before a simulation begins, thus familiarizing them with the environment. When the simulation starts, we move the user to his starting location and suspend input. We then flash a countdown on the screen, so that all users are ready to start at the same time. Finally, we show the user's instructions (in the form of an arrow pointing in one of the cardinal directions) in the lower-left corner of the screen.

To limit misbehaviour due to malice, we constantly monitor users, and we run the experiment in one computer lab. Client programs are distributed at the start of the study, and may be obfuscated to prevent users from copying a client to external storage, modifying it, and then re-using it during the next session. We also monitor the user in real-time using a case-based reasoning system, and we flash a "warning" light on the user's screen if he turns too fast. (For our simple movement-based exercise, turning too fast was both the most common error and the most difficult to detect *post factum*.) Other misbehaviour like refusing to move at all is easier to spot manually; however, the use of a case base allows us to easily extend and add this case to the automatic reasoning module later, if it proves desirable to detect this automatically.

Collusion is minimized by using the same graphic for every "other player" on one's screen. This also helps to remove any prejudices involved in micro-lane formation — for example, in real life, certain people avoid following others if they judge (without actually observing) that they might walk slowly, or erratically.

4. Final Considerations

4.1 Future Work

Our current work on the model described in (Hetu and Tan, 2008b) culminates in a series of calibration experiments to prove the concept of our MMOHILS. We hope to continue research into the field of serious gaming by working to leverage the online gaming communities in games such as Second Life. The online communities of such games are a vast and untapped resource in the fields of serious gaming and simulation. We feel that with the proper level of real-time validation, large scale simulation studies can be carried out on these communities without affecting gameplay or enjoyment.

4.2 Conclusion

Through an appeal to existing and new techniques in the field of simulation, we have proposed a generalized framework and approach to retrieving reliable and consistent data from unreliable and inconsistent human users in serious gaming studies. This framework was designed for studies with a provable hypothesis, but its general tenets are flexible enough to be used in training exercises or general exploratory studies as well.

Author information

Seth Hetu is working towards his PhD in Computer Science at the National University of Singapore. His focus is in Crisis Management Simulation, and in Symbiotic Simulation.
E- mail: <seth.hetu@gmail.com>

Gary Tan is an Associate Professor at the Department of Computer Science, School of Computing, National University of Singapore. His current research interests are Parallel and Distributed Simulation, Model Composability, Crisis Management Simulation and Symbiotic Simulation.
E- mail: <gtan@comp.nus.edu.sg>

List of references

Cheok, A.D. Sreekumar, A. Lei, C. and Thang, L.N. (2006) Capture the flag: mixed-reality social gaming with smart phones. *Pervasive Computing, V5 I2*. pp. 62- 69

Fruin, J. (1987). *Pedestrian Planning and Design*, 2nd Edi-tion. Elevator World, Mobile AL.

Hetu, S. and Tan, G. (2008a). MMOHILS: A simpler approach to valid agents in human simulation studies. *Proceedings of the 2008 Winter Simulation Conference*.

Hetu, S. and Tan, G. (2008b). Real-Time Simulation in Java: A Feasibility Study. *ICSC 2008*.

Hopkinson, W. and Sepulveda, J. (1995). Real Time Validation of Man-in-the-Loop Simulations, *Proceedings of the 1995 Winter Simulation Conference*.

Leigh, E. (2007). Learning Through Simulations & Games. *Presentation to SSAGSG 31st March, 2007*. Available online at: http://www.ssagsg.org/doc/2007_ElyssebethPresentation.pdf

McGonigal, J. (2006). This Might Be a Game: Ubiquitous Play and Performance at the Turn of the Twenty-First Century. *Dissertatation, University of California at Berkeley*. Chapter 4.

Penn State. (2008). Second Life Panel: Problem Based Learning and Graduate Study in an Immersive Experiential Environment. Available online at: http://symposium.tlt.psu.edu/content/second-life-panel-problem-based-learning-and-graduate-study-immersive-experiential-environme

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey.

Smith, R. (1998). *Modelbenders Simulation Encyclopaedia Article*. Available online at: http://www.modelbenders.com/encyclopedia/encyclopedia.html